

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра информационных и управляющих систем

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ
«ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ»**

основной образовательной программы по специальности 230201.65 – информационные системы и технологии

Благовещенск 2012

УМКД разработан старшим преподавателем, Назаренко Натальей Викторовной

Рассмотрен и рекомендован на заседании кафедры

Протокол заседания кафедры от «___»___2012 г., № ___

Зав. кафедрой _____ / А.В. Бушманов /
(подпись) И.О. Фамилия)

УТВЕРЖДЕН

Протокол заседания УМСС специальности 230201.65 – информационные системы и технологии

от «___»_____2012 г., №___

Председатель _____ / _____ /
(подпись) (И.О. Фамилия)

СОДЕРЖАНИЕ

I. Рабочая программа	4
II. Краткое изложение программного материала	10
III. Методические указания (рекомендации)	32
1. Методические указания по изучению дисциплины	32
2. Методические указания к практическим занятиям	32
3. Методические указания по самостоятельной работе студентов	33
IV. Контроль знаний	34
1. Текущий контроль знаний	34
2. Итоговый контроль знаний	36
V. Интерактивные технологии и инновационные методы, используемые в образовательном процесс	36

I. РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Курс – 3
Зачет – 6
Семестр – 6
Лекции – 18 (час.)
Лабораторные работы – 18 (час.)
Самостоятельная работа – 36 (час.)
Общая трудоемкость дисциплины – 72 (час.)

1 ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины – получение теоретических знаний и практического опыта по представлению и обработке знаний, проектированию и использованию экспертных систем и баз знаний.

Задачи дисциплины:

- дать основные понятия о моделях представления знаний в новых информационных технологиях, об алгоритмах логического вывода в различных моделях знаний, о подходах к построению моделей предметной области или выделенной сферы деятельности с точки зрения систем ИИ.

- дать представление о новых технологиях обработки информации, основанных на знаниях, а также о системах, объединяемых понятием системы искусственного интеллекта.

В результате изучения курса студенты должны:

знать основные понятия и направления исследований в области искусственного интеллекта, основные средства проектирования и разработки программ современными интеллектуальными методами, модели представления и получения знаний, методы инженерии знаний. Владеть языками логического и объектно-ориентированного программирования для решения интеллектуальных задач.

уметь применять необходимые методы искусственного интеллекта при разработке различных прикладных задач, строить модели простых предметных областей в логических исчислениях высказываний и предикатов 1-го порядка, а также в продукционных и фреймовых представлениях; Практически использовать ПРОЛОГ, объектно-ориентированные и алгоритмические языки для разработки интеллектуальных задач.

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Дисциплина «Представление знаний в информационных системах» относится к блоку общепрофессиональных дисциплин ООП по специальности 230201.65 «Информационные системы и технологии» (квалификация «инженер»).

Изучение данной дисциплины требует от студентов предварительного усвоения таких дисциплин как «Информатика», «Информационные технологии», «Алгоритмические языки и программирование», «Технология программирования», «Управление данными» в объеме государственного образовательного стандарта высшего профессионального образования.

Изучение данной дисциплины требует от студентов предварительного усвоения таких дисциплин как «Информатика», «Информационные технологии», «Алгоритмические языки и программирование», «Технология программирования», «Управление данными» в объеме государственного образовательного стандарта высшего профессионального образования.

3 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 72 часа.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость в часах				Формы текущего контроля успеваемости Форма промежуточной аттестации
				Лек	Пр	Лаб	Сам	
1	Информационный процесс представления знаний	6	1	2	-	2	6	Защита лаб. работы
2	Модели представления знаний и вывод на знаниях	6	3-5	4	-	4	6	Защита лаб. работы
3	Теория и техника приобретения знаний	6	7-9	4	-	4	6	Защита лаб. работы
4	Особенности разработки и использования экспертных систем.	6	11-13	4	-	4	6	Защита лаб. работы
5	Нечеткие знания и способы их обработки	6	15	2	-	2	6	Защита лаб. работы
6	Программный инструментальный разработки систем основанных на знаниях.	6	17	2	-	2	6	Защита лаб. работы
7	Всего по разделам		1-18	18	-	18	36	Зачет

4 СОДЕРЖАНИЕ РАЗДЕЛОВ И ТЕМ ДИСЦИПЛИНЫ

Лекции:

Тема 1: Информационный процесс представления знаний.

Этапы развития СИИ, эволюция представления знаний. Свойства и типы знаний. Декларативные и процедурные знания. Классификация систем основанных на знаниях. Типичные модели представления знаний. традиционные способы обработки знаний.

Тема 2: Модели представления знаний и вывод на знаниях

Логическая модель представления знаний и правила вывода; теоретические основы; пример спецификации и вычисления, продукционные модели представления знаний и правила их обработки, теоретическая модель продукционной системы (ПС). Структура ПС. Рабочая база данных. База правил. Механизм сопоставления. Достоинства и недостатки ПС. Реляционные модели представления знаний. Фрейм как структура представления знаний. Структура фрейма. Типы фреймов. Механизм присоединенных процедур, механизм наследования. Достоинства и недостатки фреймовых представлений. Семантические сети (СС). Виды СС: однородные, функциональные, сценарные, иерархические. Основные типы отношений в СС. Механизм вывода на семантических сетях. Достоинства и недостатки СС.

Тема 3: Теория и техника приобретения знаний.

Принципы приобретения знаний. Теоретические аспекты извлечения знаний. Стратегии получения знаний. Структурирование знаний. Классификация методов практического извлечения знаний. Коммуникативные методы. Текстологические методы. Состояние и перспективы автоматизированного приобретения знаний.

Тема 4: Особенности разработки и использования экспертных систем.

Назначение и особенности методов искусственного интеллекта для разработки экспертных систем. Определение экспертной системы. Структура экспертных систем и основные понятия. Общие сведения об архитектуре экспертных систем. Пользовательский интер-

фейс, механизм вывода и база знаний. Классификация экспертных систем. Классификация инструментальных средств. Методология разработки экспертных систем. Этапы разработки экспертных систем. Основные направления в изучении и практическом использовании. Экспертные системы - инструмент автоматизированных обучающих систем. Роли эксперта, инженера знаний и пользователя.

Тема 5: Нечеткие знания и способы их обработки

Понятие о нечетких множествах и их связь с теорией построения экспертных систем. Виды нечеткости знаний. Операции с нечеткими знаниями.

Тема 6: Программный инструментарий разработки систем основанных на знаниях.

Языки программирования для ИИ и языки представления знаний. Логическое программирование и экспертные системы. Применение языка Пролог.

Лабораторные работы:

1. Работа с простейшими программами в системе Турбо-Пролог
2. Управление ходом выполнения программ в системе Турбо-Пролог
3. Создание отношений в Прологе
4. Рекурсия и рекурсивные процедуры в прологе
5. Списки и процедуры их обработки
6. Пролог-программы как простейшие базы данных и знаний
7. Способы представления баз данных в Пролог-программах

5 САМОСТОЯТЕЛЬНАЯ РАБОТА

В качестве самостоятельной работы выполняется одно задание, которое состоит из 6-ти задач.

1. Построение модели предметной области для данной сферы деятельности человека.
2. Построение продукционной модели заданной предметной области в виде базы фактов (данных) и базы знаний и имитация работы этой модели в течение 5 тактов.
3. Построение фрагмента модели заданной предметной области на языке исчисления высказываний.
4. Построение фрагмента модели заданной предметной области на языке исчисления предикатов путём соответствующего усложнения.
5. Построение фрагмента модели предметной области в виде семантической сети.
6. Построение фрагмента предметной области в виде сети фреймов.

Темы заданий

Выбирается одна из приведенных ниже тем, и на ее основе выполняются все 6 вышеперечисленных задач. Указанные темы не являются обязательными. Студент вправе предложить свою, близкую ему, тему.

1. Сборка арки с помощью робота. Арка состоит из 2-х вертикальных и одной горизонтальных балок. Первоначально все балки находятся в различных точках стройплощадки.
2. Создание нового текстового файла в редакторе Word. Исходное состояние – компьютер выключен.
3. Запуск в рабочее состояние автомобиля. Начальное состояние – водитель рядом с машиной.
4. Заправка бензобака автомобиля у бензоколонки. Начальное состояние – автомобиль у бензоколонки.
5. Замена картриджа у принтера. Начальное состояние – картридж лежит рядом с принтером.

6. Прием–передача информации на пейджер с помощью пейджинговой компании. В процессе участвуют три человека: 1-й отправляет сообщение, 2-й принимает сообщение, 3-й – оператор – передает сообщение.
7. Установка автомашины в гараж. В начальный момент времени машина стоит у гаража, гараж закрыт.
8. Автоматизированный комплекс (конвейер) проверки наполнения бутылок квасом (водой, молоком и т.п.). В случае обнаружения недолива – доливает необходимое количество жидкости и поочередно снимает бутылки и ставит их в ящик.
9. Диагностика неисправностей блока питания компьютера.
10. Диагностика неисправностей компьютерного дисковод.
11. Ввод документа в компьютер с дискеты и его распечатка.
12. Передача документа по факсимильной связи.
13. Принятие решения о сертификации товара. Решение принимается на основе установления соответствия реквизитов товара (имя, № накладной, фирма, стоимость) реквизитам товаросопроводительных документов. При наличии несоответствий принимается решение об отправке на экспертизу. Если экспертиза подтверждает нарушение, то следует отказ в выдаче сертификата.
14. Диагностика неисправностей компьютера при его включении.
15. Управление лифтом (6 этажей, из двух или более нажатых кнопок выполняется та, которая меньше по номеру, если лифт идет вверх, и та, которая больше по номеру, если лифт идет вниз).
16. Заварка чая по-японски (по-русски, по-китайски и т.п.).
17. Приготовление борща.
18. Управление движением автомобилей на Т-образном перекрестке. Движение допускается в обе стороны. Решения: включить красный светофор, включить зеленый.
19. Управление потоком заданий на компьютер. Задания имеют ранг: 1, 2, 3; время решения соответственно: t_1 , t_2 , t_3 .
20. Обслуживание абонента в библиотеке.
21. Уход за комнатным цветком.
22. Обработка детали на станке, съём и замена детали.
23. Формирование каталога Интернет-ресурсов. Решение о регистрации принимается в случае соответствия заявленных данных о фирме поданным документам (тип деятельности, платежеспособность, оплата сервера и т.п.).
24. Диагностика неисправностей в работе принтера.
25. Форматирование дискеты и запись двух новых файлов.
26. Отправка посылки через почтовое отделение.
27. Ведение и обеспечение функционирования базы данных.
28. Ремонт копировального аппарата.
29. Осуществление модемной связи между двумя компьютерами. Предполагается, что один модем (ожидающий) ждет звонка, второй (иницирующий) осуществляет дозвон.
30. Рыбалка: смоделировать процесс рыбной ловли.
31. Сортировка поездов на сортировочной станции.
32. Уход за цветком в теплице.
33. Управление взлетом самолетов в аэропорту (одна взлетно-посадочная полоса).
34. Игра в "крестики-нолики".
35. Перевозка через реку волка, козы и капусты.
36. Распределение вакансий на бирже труда.
37. Заключение договора на поставку оборудования.

6 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Результаты освоения дисциплины достигаются за счет использования в процессе обу-

чения современных инструментальных средств: лекционные занятия с применением технологий мультимедиа.

Защита лабораторных работ происходит в виде устной беседы по выполненным студентом индивидуальным заданиям и проверкой программ, написанных студентами.

7 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Для организации текущего контроля полученных студентами знаний по данной дисциплине используются контрольные работы и тесты. Текущий контроль знаний осуществляется на лабораторных занятиях. По результатам текущего контроля знаний и защиты отчета по самостоятельной работе формируется допуск студента к зачету.

Вопросы к зачету

1. Основные направления исследований в ИИ.
2. Языки представления знаний.
3. Методы представления и использования знаний в ИИ.
4. Правила продукций.
5. Структура продукционной системы.
6. Нечеткая логика для продукционных систем.
7. Опишите на примере механизм работы базы правил.
8. Опишите механизм вывода в ПС с использованием прямой (обратной) стратегии. Приведите примеры.
9. Достоинства и недостатки продукционной модели.
10. Логика предикатов.
11. Правила выводов логики предикатов.
12. Метод резолюций в логике предикатов.
13. Семантические сети.
14. Фреймовые системы.
15. Типы фреймов.
16. Представление и использование нечетких знаний.
17. Операции с нечеткими знаниями.
18. Стратегии получения знаний.
19. Психологический аспект извлечения знаний.
20. Лингвистический аспект извлечения знаний
21. Гносеологический аспект извлечения знаний
22. Классификация методов извлечения знаний.
23. Активные методы извлечения знаний.
24. Пассивные методы извлечения знаний.
25. Текстологические методы извлечения знаний.
26. Методология структурирования знаний.
27. Стадии и методы структурирования знаний.
28. Состояние и перспективы автоматизированного приобретения знаний.
29. Эволюция систем приобретения знаний.
30. Классификация ЭС.
31. Структура и режимы экспертных систем.
32. Общие сведения об архитектуре ЭС.
33. Назначение и особенности методов ИИ для разработки ЭС.
34. Структура и режимы ЭС.
35. Характеристики ЭС.
36. Классификация инструментальных средств ЭС.

37. Методология разработки ЭС.
38. Этапы разработки ЭС.
39. Взаимодействия инженера по знаниям с экспертом.
40. Трудности разработки ЭС.
41. Проблемы и перспективы ЭС.
42. Возможности представления знаний на базе языка HTML.
43. Как работает механизм наследования?
44. Зачем нужны присоединенные процедуры?
45. Опишите механизм вывода на фреймах.
46. Укажите достоинства и недостатки фреймовых систем.
47. Дайте определение семантической сети.
48. Семантическая сеть с одним типом отношений.
49. Функции экспертных систем.
50. Роль базы знаний в экспертных системах.
51. Из каких функциональных блоков состоит ЭС?

8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

- 1 Гаскаров, Д.В. Интеллектуальные информационные системы: учеб.: Рек. Мин. обр. РФ / Д.В.Гаскаров. - М. : Высш. шк., 2003. - 432 с.
- 2 Глухих, И. Н. Интеллектуальные информационные системы: учеб. пособие / И. Н. Глухих. - М.: Академия, 2010. - 110 с.
- 3 Сергиевский Г.М., Волченков Н.Г. Функциональное и логическое программирование: учеб. пособие: М.: Академия, 2010. – 320 с.

б) дополнительная литература:

- 1 Андрейчиков, А.В. Интеллектуальные информационные системы: учебник. / А.В. Андрейчиков, О.Н. Андрейчикова – М.: Финансы и статистика, 2004. – 424 с.
- 2 Гаврилова, Т.А. Базы знаний интеллектуальных систем : Учеб.: Доп. Мин. обр. РФ / Т.А. Гаврилова, В.Ф. Хорошевский . - СПб. : Питер, 2000. - 382 с.
- 3 Гладких В.В. Идеи и решения фундаментальных проблем науки и техники / В.В. Гладких, П.В. Гладких, В.П. Гладких. – СПб.: БХВ-Петербург, 2010. – 176 с.
- 4 Голицына О.Л. Информационные системы : учеб. пособие : рек. УМО/ О.Л. Голицына, Н.В. Максимов, И.И. Попов. -М.: ФОРУМ: ИНФРА-М, 2009. -496 с.
- 5 Достоверный и правдоподобный вывод в интеллектуальных системах: учеб. пособие: Рек. Мин. обр. РФ / В. Н. Вагин [и др.]. - М.: Физматлит, 2004. - 704 с.
- 6 Зубов А.В. Основы искусственного интеллекта для лингвистов : учеб. пособие: рек. УМО/ А. В. Зубов, И. И. Зубова. -М.: Логос, 2007. -320 с.
- 7 Матвеев М.Г. Модели и методы искусственного интеллекта: учеб. пособие/ М.Г.Матвеев, А.С.Свиридов, Н.А. Алейникова. – М.: Финансы и статистика: ИНФРА-М, 2008. – 448 с.
- 8 Новак, В. Математические принципы нечеткой логики / В. Новак, И. Перфильева, И. Мочкорж ; пер. с англ., под ред. А. Н. Аверкина. - М. : Физматлит, 2006. - 348 с.
- 9 Рыбина Г.В. Основы построения интеллектуальных систем: учеб. пособие / Г.В.Рыбина. – М.: Финансы и статистика: ИНФРА-М, 2010. – 432 с.
- 10 Ясницкий, Л.Н. Введение в искусственный интеллект: учеб. пособие: Рек. УМС по математике и механике УМО: спец. 010100-Математика / Л. Н. Ясницкий. - М.: Академия, 2005. - 176 с.

в) справочные издания:

- 1 Искусственный интеллект: справ.: в 3 кн. / под ред. Э. В. Попова. - М.: Радио и

связь, 1990 - Кн. 1: Системы общения и экспертные системы. - 1990. - 464 с.

2 Искусственный интеллект: справ.: в 3 кн. / Ред. Д. А. Поспелов. - М.: Радио и связь, 1990 - Кн. 2: Модели и методы. - 1990. - 304 с.

3 Искусственный интеллект: справ.: в 3 кн. - М.: Радио и связь, 1990 - Кн. 3: Программные и аппаратные средства / под ред. В.Н. Захарова, В. Ф. Хорошевского. - 1990. - 368 с.

г) **периодические издания:**

Информационные технологии и вычислительные системы

Мир ПК

Программные продукты и системы

Программирование

Информационные системы и технологии

Информационное общество

PC magazine Персональный компьютер сегодня

д) **Интернет-ресурсы:**

№	Наименование ресурса	Краткая характеристика
1	http://www.intuit.ru/	Интернет университет информационных технологий, содержит бесплатные учебные курсы, учебники и методические пособия по всем направлениям подготовки ИТ-специалистов
2	Электронная система «Университетская библиотека - online» http://www.biblioclub.ru	ЭБС по тематике охватывает всю область гуманитарных знаний и предназначена для использования в процессе обучения в высшей школе, как студентами и преподавателями, так и специалистами-гуманитариями
3	http://www.iqlib.ru	Интернет библиотека образовательных изданий, в которой собраны электронные учебники, справочные и учебные пособия. Удобный поиск по ключевым словам, отдельным темам и отраслям знания
4	http://www.aiportal.ru/	Портал искусственного интеллекта. Содержит статьи и файлы по основным направлениям исследований в области искусственного интеллекта.

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекции проводятся в аудитории, оборудованной проектором, экраном, интерактивной доской, ноутбуком.

Для проведения лабораторных работ необходим компьютерный класс на 12-14 посадочных рабочих мест пользователей. В классе должен быть установлен язык логического программирования PROLOG.

II. КРАТКОЕ ИЗЛОЖЕНИЕ ПРОГРАММНОГО МАТЕРИАЛА

Тема 1: Информационный процесс представления знаний.

План лекции:

1. Этапы развития систем искусственного интеллекта.
2. Представление данных и знаний. Свойства и типы знаний.
3. Декларативные и процедурные знания. Классификация систем основанных на знаниях. Типичные модели представления знаний. традиционные способы обработки знаний.

1.1. История развития систем искусственного интеллекта

Идея создания искусственного подобия человека для решения сложных задач и моделирования человеческого разума витала в воздухе еще в древнейшие времена. Так, в древнем

Египте была создана «оживающая» механическая статуя бога Амона. У Гомера в «Илиаде» бог Гефест ковал человекоподобные существа-автоматы. В литературе эта идея обыгрывалась многократно: от Галатеи Пигмалиона до Буратино папы Карло. Однако родоначальником искусственного интеллекта считается средневековый испанский философ, математик и поэт Раймонд Луллий, который еще в XIII веке попытался создать механическую машину для решения различных задач, на основе разработанной им всеобщей классификации понятий.

В XVIII веке Лейбниц и Декарт независимо друг от друга продолжили эту идею, предложив универсальные языки классификации всех наук. Эти работы можно считать первыми теоретическими работами в области искусственного интеллекта.

Окончательное рождение искусственного интеллекта как научного направления произошло только после создания ЭВМ в 40-х годах XX века. В это же время Нор-берт Винер создал свои основополагающие работы по новой науке - кибернетике.

Термин «искусственный интеллект» - ИИ - (AI - artificial intelligence) был предложен в 1956 г. на семинаре с аналогичным названием в Дартсмутском колледже (США). Семинар был посвящен разработке методов решения логических, а не вычислительных задач. В английском языке данное словосочетание не имеет той слегка фантастической антропоморфной окраски, которую оно приобрело в довольно неудачном русском переводе. Слово intelligence означает «умение рассуждать разумно», а вовсе не «интеллект», для которого есть термин intellect.

Вскоре после признания искусственного интеллекта отдельной областью науки произошло разделение его на два направления: нейрокибернетика и «кибернетика черного ящика». Эти направления развиваются практически независимо, существенно различаясь как в методологии, так и в технологии. И только в настоящее время стали заметны тенденции к объединению этих частей вновь в единое целое.

1.2. Данные и знания

Приведем определения основных понятий изучаемой дисциплины и рассмотрим различие между понятиями «данные» и «знания».

Информация – совокупность сведений, воспринимаемых из окружающей среды, выдаваемых в окружающую среду либо сохраняемых внутри информационной системы.

Данные – представленная в формализованном виде конкретная информация об объектах предметной области, их свойствах и взаимосвязях, отражающая события и ситуации в этой области.

Данные представляются в виде, позволяющем автоматизировать их сбор, хранение и дальнейшую обработку человеком или информационным средством. Данные – это запись в соответствующем коде наблюдения, акта, объекта, песни, текста и т.д., пригодная для коммуникации, интерпретации, передачи, обработки и получения новой информации.

Информация получается из данных в результате решения некоторой задачи. Однако большая часть информации не может быть выведена из данных. Так, практически еще невозможен автоматический перевод художественных произведений с одного языка на другой. Трудно рассчитывать и на то, что в ближайшем будущем компьютер-переводчик сможет донести до нас тонкие оттенки юмора, чувств и т.п.

Информация, с которой имеет дело ЭВМ, разделяется на процедурную и декларативную. *Процедурная* информация представляется программами, которые выполняются в процессе решения задач, а *декларативная* – данными, которые обрабатывают эти программы.

Обычно как человек, так и ИС имеют дело с данными и знаниями из некоторой предметной области, например, математики, медицины, экономики и пр. Под *предметной областью* понимают информацию об объектах и связях между ними из некоторой области знаний.

Знание – это обобщенная и формализованная информация о свойствах и законах предметной области, с помощью которой реализуются процессы решения задач, преобразования данных и самих знаний, и которая используется в процессе логического вывода.

Логический вывод – это генерирование новых утверждений (суждений) на основе исходных фактов, аксиом и правил вывода.

Знания с точки зрения решения задач в некоторой предметной области удобно разделить на две большие категории – факты и эвристики.

Под *фактами* обычно понимают общеизвестные в данной предметной области истинны, обстоятельства. *Эвристики* – это эмпирические алгоритмы, основанные на неформальных соображениях, которые ограничивают разнообразие и обеспечивают целенаправленность поведения решающей системы, не гарантируя, однако, получения наилучшего решения. Такие знания основываются на опыте специалиста (эксперта) в данной предметной области

Рассмотрим особенности знаний, в которых заключается их отличие от данных.

1. Интерпретация.
2. Структурированность.
3. Связность.
4. Семантическая метрика.
5. Активность.

Перечисленные пять особенностей определяют ту грань, за которой данные превращаются в знания, а БД – в БЗ. Совокупность средств, обеспечивающих работу со знаниями, образует СУБЗ.

1.3. Классификация систем основанных на знаниях.

Основным направлением, реализующим идеи ИИ, является разработка систем, основанных на знаниях. Центральный объект изучения ИИ – знания могут быть представлены в виде некоторой совокупности сведений (фактов, правил), процессов, явлений, а также способов решения задач данной предметной области. Специалисты, занимающиеся извлечением знаний, их формализацией и структурированием для обработки в компьютерных системах, называются *инженерами по знаниям* или *инженерами знаний*.

Для формализации знаний, представленных в текстовом, графическом виде, в виде документов и т.д., требуется наличие или, возможно, разработка методов, позволяющих преобразовать исходные знания к виду, пригодному для обработки в ИИС. Знания, полученные из различных источников, требуется интегрировать в связную и непротиворечивую систему.

Полученные от экспертов знания нужно оценить с точки зрения уже имеющихся в системе знаний, согласовать с последующими и выделить несовместные или противоречивые знания.

Для представления знаний используются логические модели, продукционные правила, таблицы принятия решений, семантические сети, фреймы и др. Для ввода, хранения, обработки, вывода знаний разработаны системы управления базами знаний, которые включают языки описания и манипулирования знаниями, а также программные процедуры.

Особое место в манипулировании знаниями занимает вывод на знаниях, заключающийся в получении новых знаний на основе уже имеющихся в системе. Это одно из наиболее проблематичных с точки зрения реализации направлений в ИИС. Большой интерес в выводе знаний представляет манипулирование человеческими рассуждениями: аргументация на основе имеющихся знаний, рассуждения по аналогии и многое другое, чем люди пользуются в своей практике.

Тема 2: Модели представления знаний и вывод на знаниях

1. Логические модели
2. Продукционные модели
3. Семантические сети
4. Фреймы
5. Вывод на знаниях

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

ческие модели, продукционные модели, семантические сети, фреймы.

2.1 Логические модели

Основная идея подхода при построении *логических моделей* представления знаний состоит в том, что вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода.

В основе логических моделей представления знаний лежит понятие формальной теории, задаваемое четверкой: $S = \langle B, F, A, R \rangle$, где B — счетное множество базовых символов (алфавит), F — множество, называемое формулами, A — выделенное подмножество априори истинных формул (аксиом), R — конечное множество отношений между формулами, называемое правилами вывода.

Достоинства логических моделей представления знаний:

1. В качестве «фундамента» здесь используется классический аппарат математической логики, методы которой достаточно хорошо изучены и формально обоснованы.
2. Существуют достаточно эффективные процедуры вывода, в том числе реализованные в языке логического программирования Пролог.
3. В базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода.

Однако действительность не укладывается в рамки классической логики. Так называемая человеческая логика, применяемая при работе с неструктурированными знаниями, — это интеллектуальная модель с нечеткой структурой, и в этом ее отличие от «старой» (классической) логики. Таким образом, логики, адекватно отражающей человеческое мышление, к настоящему времени еще не создано.

2.2 Продукционные модели

Психологические исследования процессов принятия решений человеком показали, что рассуждая и принимая решения, человек использует правила продукций, или продукционные правила (от англ. *Production* — правило вывода, порождающее правило).

При использовании таких моделей у систем основанных на знаниях имеется возможность:

- применение простого и точного механизма использования знаний;
- представление знаний с высокой однородностью, описываемых по единому синтаксису.

Эти две отличительные черты и определили широкое распространение методов представления знаний правилами. Впервые продукционные системы были предложены Постом в 1941 году.

Продукция в системе Поста имеет следующую схему: $\frac{t_1, t_2, \dots, t_n}{t}$, где t_1, \dots, t_n — посылки; t — заключение.

Применение схемы Поста основывается на подстановке цепочек знаков вместо переменных, причем вместо вхождения одной и той же переменной представляется одна и та же цепочка.

Общим для систем продукций является то, что они состоят из трех элементов:

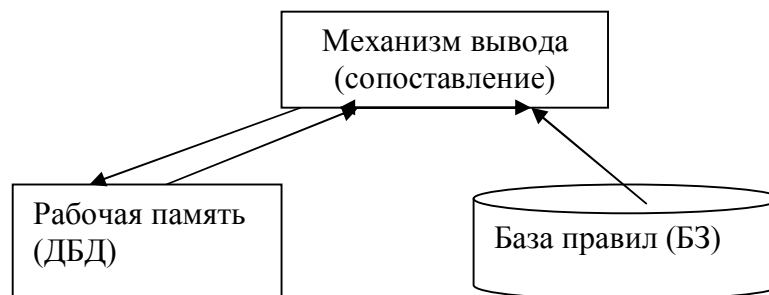
Набор правил, используемых как БЗ, его еще называют базой правил;

Рабочая память, где хранятся предпосылки, касающиеся отдельных задач, а также результаты выводов, получаемых на основе этих предпосылок (динамическая база данных ДБД);

Механизм логического вывода, использующий правила в соответствии с содержанием рабочей памяти.

Конфигурацию систем продукции упрощено можно представить в следующем виде:

Суть использования правил продукции для представления знаний состоит в том, что левой части ставится в соответствие некоторое условие, а правой части — действие: **ЕСЛИ <перечень условия>, ТО <перечень действий>**. В такой интерпретации левая часть правил оценивается по отношению к базе данных (известному набору фактов) системы, и если эта оценка в определенном смысле соответствует логическому значению «ИСТИНА», то выполняется действие, заданное в правой части продукции.



В общем случае продукционное правило можно представить в следующем виде: $i: S; L; A \rightarrow B; Q$,

где i — индивидуальный номер продукции; S — описание класса ситуаций, в котором данная структура может использоваться; L — условие, при котором продукция активизируется; $A \rightarrow B$ — ядро продукции, например: «ЕСЛИ A_1, A_2, \dots, A_n ТО B ». Такая запись означает, что «если все условия от A_1 , до A_n являются истиной, то B также истина» или же «когда все условия от A_1 , до A_n становятся истиной, то следует выполнить действие B »; Q — постусловие продукционного правила, описывает операции и действия (процедуры), которые необходимо выполнить после выполнения B . Например, внести изменения в данные либо в саму продукцию.

В общем случае под условием (антецедентом) понимается некоторое предложение - образец, по которому осуществляют поиск в базе знаний, а под действием (консеквентом) - действия, выполняемые при успешном исходе поиска, - это могут быть реальные действия, если система управляющая, или заключение - вывод, представляющий собой новое (фактуальное) знание, или некоторая цель.

При использовании продукционной модели база знаний состоит из набора правил. Программа, управляющая перебором правил, называется *машиной вывода*. Механизм выводов связывает знания воедино, а затем выводит из последовательности знаний заключение.

В продукционных системах, основанных на знаниях, процесс обработки информации может осуществляться двумя способами. Первый предполагает обработку информации в прямом направлении (метод сопоставления). При втором подходе обработка информации осуществляется в обратном направлении - метод «генерации» или выдвижения гипотезы и ее проверки (стратегия «от цели к данным»).

Таким образом, продукционные правила могут применяться к описанию состояния и описывать новые состояния (гипотезы) или же, напротив, использовать целевое состояние задачи как базу, когда система работает в обратном направлении. При этом продукционные правила применяются к целевому описанию для порождения подцелей (образуют систему редукций).

Свойства продукционных моделей:

- 1) Модульность
- 2) Каждое продукционное правило — самостоятельный элемент знаний
- 3) Простота интерпретации
- 4) Естественность

Недостатки продукционных систем проявляются тогда, когда число правил становится большим и возникают непредсказуемые побочные эффекты от изменения старого и добавления нового правила. Кроме того, затруднительна оценка целостного образа знаний, содержащихся в системе.

2.3 Семантические сети

Способ представления знаний с помощью *сетевых моделей* наиболее близок к тому, как они представлены в текстах на естественном языке. В его основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек (arb) , где a и b - объекты или понятия, а r - бинарное отношение между ними.

Формально сетевые модели представления знаний могут быть заданы в виде $H = \langle I, C_1, \dots, C_n, \Gamma \rangle$, где I — множество информационных единиц, C_1, \dots, C_n - множество типов связей между элементами I , отображение Γ задает между информационными единицами, входящими в I , связи из заданного набора типов связей $\{C_i\}$.

В зависимости от типов связей $\{C_i\}$ различают: классифицирующие сети, функциональные сети, сценарии.

Если в сетевой модели допускаются связи различного типа, то ее называют семантической сетью.

Семантическая сеть — это модель, основой для которой является формализация знаний в виде ориентированного графа с размеченными вершинами и дугами. Вершинам соответствуют объекты, понятия или ситуации, а дугам — отношения между ними. В качестве понятий обычно выступают абстрактные или конкретные объекты, а отношения — это связи типа: «это» («АКО» — A-Kind-Of), «is»), «имеет частью» («has part»), «принадлежит», «любит». Это наиболее общая модель представления знаний, так как в ней имеются средства реализации всех характерных для знаний свойств: внутренней интерпретации, структурированности, семантической метрики и активности.

Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс – элемент класса (цветок – роза);
- свойство – значение (цвет – желтый);
- пример элемента класса (роза – чайная).

Можно предложить несколько классификаций семантических сетей, связанных с типами отношений между понятиями.

По количеству типов отношений:

- однородные (с единственным типом отношений);
- неоднородные (с различными типами отношений).

По типам отношений:

- бинарные (в которых отношения связывают два объекта);
- N-арные (в которых есть специальные отношения связывающие более двух понятий).

Наиболее часто в семантических сетях используются следующие отношения:

- связи типа «часть-целое» («класс-подкласс», «элемент-множество», и т.п.);
- функциональные связи (определяемые обычно глаголами «производит», «влияет»...);
- количественные (больше, меньше, равно...);
- пространственные (далеко от, близко от, за, под, над...);
- временные (раньше, позже, в течение...);
- атрибутивные связи (иметь свойство, иметь значение);
- логические (И, ИЛИ, НЕ);
- лингвистические связи и др.

Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети, отражающей поставленный запрос к базе.

Данная модель представления знаний была предложена американским психологом Куиллианом. Основным ее преимуществом является то, что она более других соответствует современным представлениям об организации долговременной памяти человека.

Недостатком этой модели является сложность организации процедуры поиска вывода на семантической сети.

Достоинства сетевых моделей: большие выразительные возможности; наглядность системы знаний, представленной графически; близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке; соответствие современным представлениям об организации долговременной памяти человека.

Недостатки: сетевая модель не дает (точнее, не содержит) ясного представления о структуре



Рис. Семантическая сеть

предметной области, которая ей соответствует, поэтому формирование и модификация такой модели затруднительны; сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода и планирования. Проблема поиска решения в базе знаний типа семантической сети сводится к задаче поиска фрагмента сети, соответствующего некоторой подсети поставленной задачи. Это, в свою очередь, обуславливает еще один недостаток модели — сложность поиска вывода на семантических сетях.

2.4 Фреймовые модели

Термин *фрейм* (frame — каркас, рамка) предложен М. Минским в 70-е годы для обозначения структуры знаний для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование. Под фреймом понимается абстрактный образ или ситуация. В психологии и философии известно понятие абстрактного образа. Например, слово «комната» вызывает образ комнаты — «жилое помещение с четырьмя стенами, полом, потолком, окнами и дверью». Из этого описания ничего нельзя убрать, например, убрав окна, мы получим уже чулан, а не комнату. Но в нем есть «слоты», или «щели», — незаполненные значения некоторых атрибутов — количество окон, цвет стен, высота потолка, покрытие пола и др. Такой образ и называется фреймом (фреймом минимального описания). Фреймом называется также и формализованная модель этого образа.

Фреймовая модель, основанная на теории М. Минского, представляет собой систематизированную в виде единой теории технологическую модель памяти человека и его сознания. Важным элементом в этой теории является понятие фрейма-структуры данных для представления некоторого концептуального объекта. Информация, относящаяся к этому фрейму, содержится в составляющих фрейма — слотах. В отличие от моделей других типов, во фреймовых моделях фиксируется жесткая структура, которая называется протофреймом (фреймом-прототипом, или образцом). В общем случае фрейм определяется следующим образом:

где f - имя фрейма; v_i - значение слота, или

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

(имя n-го слота: значение n-го слота)).

Значением слота может быть практически что угодно (числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма). В качестве значения слота может выступать набор слотов более низкого уровня, что позволяет во фреймовых представлениях реализовать «принцип матрешки».

В качестве значения слота может выступать имя другого фрейма; так образуются сети фрейм-

мов. Все фреймы взаимосвязаны и образуют единую фреймовую структуру, в которой органически объединены декларативные и процедурные знания. Это дает возможность достаточно естественно производить композицию и декомпозицию информационных структур аналогично тому, как это делал бы человек при описании структуры своих знаний.

Кроме фреймов-образцов, или прототипов, хранящихся в базе знаний, различают *фреймы-экземпляры*, которые создаются для отображения реальных ситуаций на основе поступающих данных. При конкретизации (означивании) фрейма ему и слотам присваиваются конкретные имена, и происходит заполнение слотов. Таким образом, из протофреймов получают фреймы-экземпляры.

Если в качестве значений слотов использовать реальные данные из таблицы, то получится фрейм-экземпляр.

Важнейшим свойством фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (от *A Kind Of* = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся значения аналогичных слотов, причем наследование свойств может быть частичным.

Фреймовые модели являются достаточно универсальными, поскольку позволяют отобразить все многообразие знаний о мире:

- через фреймы-структуры для обозначений объектов и понятий (заем, залог, вексель);
- фреймы-роли (менеджер, кассир, клиент);
- фреймы-сценарии (банкротство, собрание акционеров, празднование именин);
- фреймы-ситуации (тревога, авария, рабочий режим устройства и т. д.).

Основными достоинствами модели фреймов как модели представления знаний являются способность отражать концептуальную основу организации памяти человека, а также естественность, наглядность представления, модульность, поддержка возможности использования значений слотов по умолчанию. Однако фрейм-представление является не конкретным языком представления знаний, а некоторой идеологической концепцией, реализуемой по-разному в различных языках.

Основным недостатком фреймовых моделей является отсутствие механизмов управления выводом. Отчасти этот недостаток устраняется при помощи присоединенных процедур, реализуемых силами пользователя системы. Рассмотренные модели представления знаний во многом близки между собой. По сути, они обладают одинаковыми возможностями описывать и представлять знания. Разница состоит лишь в том, насколько удобно и естественно представлять те или иные знания в виде логических формул, семантических сетей, фреймов или продукций.

Обобщая анализ моделей представления знаний, можно сделать два основных вывода:

Невозможно дать универсальные рекомендации по выбору модели. Выбор конкретной модели определяется возможностью и удобством представления исследуемой проблемной области с учетом необходимости не только представления, но и использования знаний. Однако чаще используются эвристические, а не логические модели представления знаний.

Наиболее мощными оказываются смешанные представления.

2.5. Вывод на знаниях

Несмотря на все недостатки, наибольшее распространение получила продукционная модель представления знаний. При использовании продукционной модели база знаний состоит из набора правил. *Программа, управляющая перебором правил, называется машиной вывода.*

Машина вывода (интерпретатор правил) выполняет две функции: во-первых, просмотр существующих фактов из рабочей памяти (базы данных) и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов и, во-вторых, определение порядка просмотра и применения правил. Этот механизм управляет процессом консультации, сохраняя для пользователя информацию о полученных заключениях, и запрашивает у него информацию, когда для срабатывания очередного правила в рабочей памяти оказывается недостаточно данных.

В подавляющем большинстве систем, основанных на знаниях, механизм вывода пред-

ставляет собой небольшую по объему программу и включает два компонента — один реализует собственно вывод, другой управляет этим процессом. Действие компонента вывода основано на применении правила, называемого *modus ponens*.

Правило *modus ponens*. Если известно, что истинно утверждение А и существует правило вида «если А, то В», тогда утверждение В также истинно.

Правила срабатывают, когда находятся факты, удовлетворяющие их левой части: если истинна посылка, то должно быть истинно и заключение.

Компонент вывода должен функционировать даже при недостатке информации. Полученное решение может и не быть точным, однако система не должна останавливаться из-за того, что отсутствует какая-либо часть входной информации

Управляющий компонент определяет порядок применения правил и выполняет четыре функции.

1. Сопоставление
2. Выбор
3. Срабатывание
4. Действие

Интерпретатор продукций работает циклически. В каждом цикле он просматривает все правила, чтобы выявить те, посылки которых совпадают с известными на данный момент фактами из рабочей памяти. После выбора правило срабатывает, его заключение заносится в рабочую память, и затем цикл повторяется сначала.

В одном цикле может работать только одно правило. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию единственного правила, которое срабатывает в данном цикле. Цикл работы интерпретатора схематически представлен на рисунке 2.1.

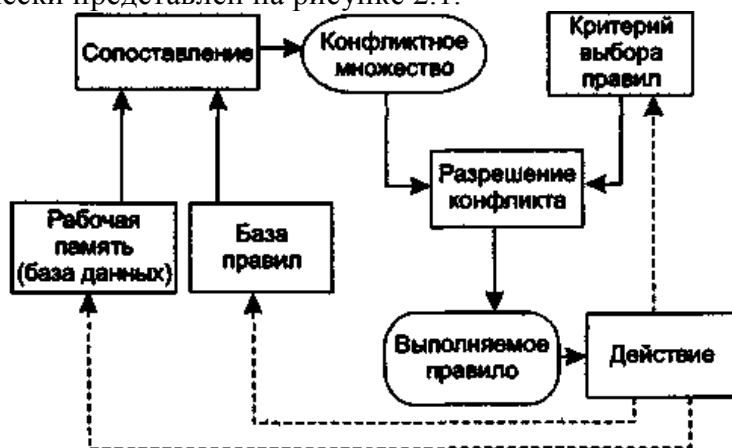


Рисунок 2.1 - Цикл работы интерпретатора

От выбранного метода поиска, то есть стратегии вывода, будет зависеть порядок срабатывания правил. Процедура выбора сводится к определению поиска и способа его осуществления. Процедуры, реализующие поиск, обычно «защиты» в механизм вывода, поэтому в большинстве систем инженеры знаний не имеют к ним доступа и, следовательно, не могут в них ничего изменять по своему желанию.

При разработке стратегии управления выводом важно определить два вопроса:

Какую точку в пространстве состояний принять в качестве исходной? От выбора этой точки зависит и метод осуществления поиска - в прямом или обратном направлении.

Какими методами можно повысить эффективность поиска решения? Эти методы определяются выбранной стратегией перебора - глубину, в ширину, по подзадачам или иначе.

Тема 3: Теория и техника приобретения знаний

1. Теоретические аспекты извлечения знаний
2. Эволюция систем приобретения знаний

3. Современное состояние автоматизированных систем приобретения знаний

3.1 Теоретические аспекты извлечения знаний

Приобретением знаний называется выявление знаний из источников и преобразование их в нужную форму, а также перенос в базу знаний ИнС. Источниками знаний могут быть:

а) книги, архивные документы, содержимое других баз знаний и т.п., т.е. некоторые объективизированные знания, приведенные к форме, которая делает их доступными для потребителя;

б) экспертные знания, которые имеются у специалистов, но не зафиксированы во внешних по отношению к ним хранилищах (экспертные знания являются субъективными)';

в) эмпирические знания (также субъективный вид знаний), которые получаются путем наблюдения за окружающей средой (если у ИнС есть средства наблюдения).

Ввод в БЗ объективизированных знаний не представляет особой проблемы, выявление и ввод субъективных и особенно экспертных знаний достаточно трудны. Чтобы разработать методологию приобретения субъективных знаний, получаемых от эксперта, надо четко различать две формы репрезентации (представления) знаний.

Одна форма связана с тем, как и в каких моделях хранятся эти знания у человека – эксперта. При этом эксперт не всегда осознает полностью, как репрезентированы у него знания. Другая форма связана с тем, как инженер по знаниям (когнитолог), проектирующий ИС, собирает их описывать и представлять. От степени согласованности этих двух форм репрезентации между собой зависит эффективность инженера по знаниям. В когнитивной психологии изучаются формы репрезентации знаний – когнитивные структуры знаний, характерные для человека. Примерами могут служить:

– представление класса понятий через его элементы (например, понятие «птица» репрезентируется рядом «чайка, воробей, скворец,...»)

птица = <чайка, воробей, скворец,...>;

– представление понятий класса с помощью базового прототипа, отражающего наиболее типичные свойства объектов класса (например, понятие «птица» репрезентируется прототипом «нечто с крыльями, клювом, летает,...»)

птица = <нечто с крыльями, с клювом, летает,...>;

– представление с помощью признаков (для понятия «птица», например, наличие крыльев, клюва, двух лап, перьев, ...)

птица = <крылья, клюв, две лапы, перья,...>.

Кроме понятий репрезентируются и отношения между ними. Как правило, отношения между понятиями определяются процедурным способом, а отношения между составляющими понятиями (определяющими структуру понятия) – декларативным способом. Наличие двух видов описаний заставляет в моделях представления знаний одновременно иметь обе компоненты, например семантическую сеть и продукционную систему.

Проблема приобретения знаний изучается в рамках инженерии знаний. Схема приобретения знаний может быть представлена следующим образом:

Носитель информации → Посредник → Модель знания

Приобретением знаний называют процесс получения знаний от эксперта или каких-либо других источников и передача их в ИнС. Однако наряду с термином «приобретение» сегодня широко используются и другие термины для обозначения этого процесса, например: «извлечение», «получение», «добыча», «формирование знаний». В англоязычной литературе по ИнС используются в основном два термина: acquisition (приобретение) и elicitation (извлечение, выявление, установление).

3.2 Эволюция систем приобретения знаний

Первое поколение таких систем появилось в середине 80-х – это так называемые системы приобретения знаний (СПЗ) (TEIRESIAS [Davis, 1982], SIMER+MIR [Осипов, 1988], АРИАДНА [Моргоев, 1988]). Это средства наполнения так называемых «пустых» ЭС, то есть систем, из БЗ которых изъяты знания (например, ЕМYSIN – EMPTY MYCIN, опустошенная медицинская ЭС MYCIN со специальной диалоговой системой заполнения базы знаний

TEIRESIAS). Их авторы считали, что прямой диалог эксперта с компьютером через СПЗ может сократить жизненный цикл разработки. Однако опыт создания и внедрения СПЗ продемонстрировал несовершенство такого подхода.

Основные недостатки СПЗ I поколения:

- Слабая проработка методов извлечения и структурирования знаний.
- Жесткость модели представления знаний, встроенной в СПЗ и связанной с привязкой к программной реализации.
- Ограничения на предметную область.

Таким образом, традиционная схема разработки СПЗ I поколения:

создание конкретной ЭС → опустошение БЗ → разработка СПЗ для новых наполненных БЗ → формирование новой БЗ для другой ЭС

оказалась несостоятельной для промышленного применения.

Второе поколение СПЗ появилось в конце 80-х и было ориентировано на более широкий модельный подход с акцентом на предварительном детальном анализе предметной области. Так, в Европе широкое применение получила методология KADS (Knowledge Acquisition and Documentation Structuring), в основе которой лежит понятие интерпретационной модели, позволяющей процессы извлечения, структурирования и формализации знаний рассматривать как «интерпретацию» лингвистических знаний в другие представления и структуры.

3.3 Современное состояние автоматизированных систем приобретения знаний

Анализ современного состояния программных средств приобретения знаний и поддержки деятельности инженера по знаниям позволяет выявить две группы проблем, характерных для существующих СПЗ:

- методологические проблемы;
- технологические проблемы.

А. Методологические проблемы

Основная проблема, встающая перед разработчиками, – отсутствие теоретического базиса процесса извлечения и структурирования знаний – порождает дочерние более узкие вопросы и казусы на всех этапах создания интеллектуальных систем. Даже тщательно проработанная методология KADS, страдает громоздкостью и явной избыточностью. Ниже перечислены наиболее общие из возникающих проблем в последовательности, соответствующей стадиям жизненного цикла:

- размытость критериев выбора подходящей задачи;
- слабая проработанность теоретических аспектов процессов извлечения знаний (философские, лингвистические, психологические, педагогические, дидактические и другие аспекты), а также отсутствие обоснованной классификации методов извлечения знаний и разброс терминологии;
- отсутствие единого теоретического базиса процедуры структурирования знаний;
- жесткость моделей представления знаний, заставляющая разработчиков обеднять и урезать реальные знания экспертов;
- несовершенство математического базиса моделей представления знаний (дескриптивный, а не конструктивный характер большинства имеющихся математических моделей);
- эмпиричность процедуры выбора программного инструментария и процесса тестирования (отсутствие критериев, разрозненные классификации, etc.).

Б. Технологические проблемы

Большая часть технологических проблем является естественным следствием методологических и порождена ими. Наиболее серьезными из технологических проблем являются:

- отсутствие концептуальной целостности и согласованности между отдельными приемами и методами инженерии знаний;
- недостаток или отсутствие квалифицированных специалистов в области инженерии знаний;

- отсутствие технико-экономических показателей оценки эффективности ЭС (в России);
- несмотря на обилие методов извлечения знаний (фактически более 200), практическая недоступность методических материалов по практике проведения сеансов извлечения знаний;
- явная неполнота и недостаточность имеющихся методов структурирования знаний, отсутствие классификаций и рекомендаций по выбору подходящего метода;
- несмотря на обилие рынка программных средств, недостаток промышленных систем поддержки разработки и их узкая направленность (зависимость от платформы, языка реализации, ограничений предметной области), разрыв между ЯПЗ и языками, встроенными в «оболочки» ЭС;
- жесткость программных средств, их низкая адаптивность, отсутствие индивидуальной настройки на пользователя и предметную область;
- слабые графические возможности программных средств, недостаточный учет когнитивных и эргономических факторов;
- сложность внедрения ЭС, обусловленная психологическими проблемами персонала и неприятия новой технологии решения задач.

Тема 4: Особенности разработки и использования экспертных систем.

1. Назначение экспертных систем.
2. Структура ЭС
3. Этапы разработки ЭС

4.1 Назначение экспертных систем

В начале восьмидесятых годов в исследованиях по искусственному интеллекту сформировалось самостоятельное направление, получившее название "экспертные системы" (ЭС). Цель исследований по ЭС состоит в разработке программ, которые при решении задач, трудных для эксперта-человека, получают результаты, не уступающие по качеству и эффективности решениям, получаемым экспертом. Исследователи в области ЭС для названия своей дисциплины часто используют также термин «инженерия знаний», введенный Е.Фейгенбаумом как "привнесение принципов и инструментария исследований из области искусственного интеллекта в решение трудных прикладных проблем, требующих знаний экспертов".

Программные средства (ПС), базирующиеся на технологии экспертных систем, или инженерии знаний (в дальнейшем будем использовать их как синонимы), получили значительное распространение в мире. Важность экспертных систем состоит в следующем:

- технология экспертных систем существенно расширяет круг практически значимых задач, решаемых на компьютерах, решение которых приносит значительный экономический эффект;
- технология ЭС является важнейшим средством в решении глобальных проблем традиционного программирования: длительность и, следовательно, высокая стоимость разработки сложных приложений;
- высокая стоимость сопровождения сложных систем, которая часто в несколько раз превосходит стоимость их разработки; низкий уровень повторной используемости программ и т.п.;
- объединение технологии ЭС с технологией традиционного программирования добавляет новые качества к программным продуктам за счет: обеспечения динамичной модификации приложений пользователем, а не программистом; большей "прозрачности" приложения (например, знания хранятся на ограниченном ЕЯ, что не требует комментариев к знаниям, упрощает обучение и сопровождение); лучшей графики; интерфейса и взаимодействия.

По мнению ведущих специалистов, в недалекой перспективе ЭС найдут следующее применение:

– ЭС будут играть ведущую роль во всех фазах проектирования, разработки, производства, распределения, продажи, поддержки и оказания услуг;

– технология ЭС, получившая коммерческое распространение, обеспечит революционный прорыв в интеграции приложений из готовых интеллектуально-взаимодействующих модулей.

ЭС предназначены для так называемых неформализованных задач, т.е. ЭС не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач.

Неформализованные задачи обычно обладают следующими особенностями:

– ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;

– ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;

– большой размерностью пространства решения, т.е. перебор при поиске решения весьма велик;

– динамически изменяющимися данными и знаниями.

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

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

Экспертные системы применяются для решения только трудных практических (не игрушечных) задач. По качеству и эффективности решения экспертные системы не уступают решениям эксперта-человека. Решения экспертных систем обладают *"прозрачностью"*, т.е. могут быть объяснены пользователю на качественном уровне. Это качество экспертных систем обеспечивается их способностью рассуждать о своих знаниях и умозаключениях. Экспертные системы способны пополнять свои знания в ходе взаимодействия с экспертом. Необходимо отметить, что в настоящее время технология экспертных систем используется для решения различных типов задач (интерпретация, предсказание, диагностика, планирование, конструирование, контроль, отладка, инструктаж, управление) в самых разнообразных проблемных областях, таких, как финансы, нефтяная и газовая промышленность, энергетика, транспорт, фармацевтическое производство, космос, металлургия, горное дело, химия, образование, целлюлозно-бумажная промышленность, телекоммуникации и связь и др.

Коммерческие успехи к фирмам-разработчикам систем искусственного интеллекта (СИИ) пришли не сразу. На протяжении 1960–1985 гг. успехи ИИ касались в основном исследовательских разработок, которые демонстрировали пригодность СИИ для практического использования. Начиная примерно с 1985 г. (в массовом масштабе с 1988–1990 гг.), в первую очередь ЭС, а в последние годы системы, воспринимающие естественный язык (ЕЯ-системы), и нейронные сети (НС) стали активно использоваться в коммерческих приложениях.

Следует обратить внимание на то, что некоторые специалисты (как правило, специалисты в программировании, а не в ИИ) продолжают утверждать, что ЭС и СИИ не оправдали возлагавшихся на них ожиданий и умерли. Причины таких заблуждений состоят в том, что эти авторы рассматривали ЭС как альтернативу традиционному программированию, т.е. они исходили из того, что ЭС в одиночестве (в изоляции от других программных средств) полностью решают задачи, стоящие перед заказчиком. Надо отметить, что на заре появления ЭС специфика используемых в них языков, технологии разработки приложений и используемого оборудования (например, Lisp-машины) давала основания предполагать, что интеграция ЭС с традиционными, программными системами является сложной и, возможно, невыполнимой задачей при ограничениях, накладываемых реальными приложениями. Однако в настоящее

время коммерческие инструментальные средства (ИС) для создания ЭС разрабатываются в полном соответствии с современными технологическими тенденциями традиционного программирования, что снимает проблемы, возникающие при создании интегрированных приложений.

Причины, приведшие СИИ к коммерческому успеху, следующие.

Интегрированность. Разработаны инструментальные средства искусственного интеллекта (ИС ИИ), легко интегрирующиеся с другими информационными технологиями и средствами (с CASE, СУБД, контроллерами, концентраторами данных и т.п.).

Открытость и переносимость. ИС ИИ разрабатываются с соблюдением стандартов, обеспечивающих открытость и переносимость.

Использование языков традиционного программирования и рабочих станций. Переход от ИС ИИ, реализованных на языках ИИ (Lisp, Prolog и т.п.), к ИС ИИ, реализованным на языках традиционного программирования (С, С++ и т.п.), упростил обеспечение интегрированности, снизил требования приложений ИИ к быстродействию ЭВМ и объемам оперативной памяти. Использование рабочих станций (вместо ПК) резко увеличило круг приложений, которые могут быть выполнены на ЭВМ с использованием ИС ИИ.

Архитектура клиент-сервер. Разработаны ИС ИИ, поддерживающие распределенные вычисления по архитектуре клиент-сервер, что позволило: снизить стоимость оборудования, используемого в приложениях, децентрализовать приложения, повысить надежность и общую производительность (так как сокращается количество информации, пересылаемой между ЭВМ, и каждый модуль приложения выполняется на адекватном ему оборудовании).

Проблемно/предметно-ориентированные ИС ИИ. Переход от разработок ИС ИИ общего назначения (хотя они не утратили свое значение как средство для создания ориентированных ИС) к проблемно/предметно-ориентированным ИС ИИ [9] обеспечивает: сокращение сроков разработки приложений; увеличение эффективности использования ИС; упрощение и ускорение работы эксперта; повторную используемость информационного и программного обеспечения (объекты, классы, правила, процедуры).

4.2 Структура экспертных систем

Типичная статическая ЭС состоит из следующих основных компонентов:

- решателя (интерпретатора);
- рабочей памяти (РП), называемой также базой данных (БД);
- базы знаний (БЗ);
- компонентов приобретения знаний;
- объяснительного компонента;
- диалогового компонента.

База данных (рабочая память) предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, используемым в информационно-поисковых системах (ИПС) и системах управления базами данных (СУБД) для обозначения всех данных (в первую очередь долгосрочных), хранимых в системе.

База знаний (БЗ) в ЭС предназначена для хранения долгосрочных данных, описывающих рассматриваемую область (а не текущих данных), и правил, описывающих целесообразные преобразования данных этой области.

Решатель, используя исходные данные из рабочей памяти и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи.

Компонент приобретения знаний автоматизирует процесс наполнения ЭС знаниями, осуществляемый пользователем-экспертом.

Объяснительный компонент объясняет, как система получила решение задачи (или почему она не получила решение) и какие знания она при этом использовала, что облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату.

Диалоговый компонент ориентирован на организацию дружественного общения с пользователем как в ходе решения задач, так и в процессе приобретения знаний и объяснения результатов работы.

В разработке ЭС участвуют представители следующих специальностей:

эксперт в проблемной области, задачи которой будет решать ЭС;

инженер по знаниям - специалист по разработке ЭС (используемые им технологию, методы называют технологией (методами) инженерии знаний);

программист по разработке инструментальных средств (ИС), предназначенных для ускорения разработки ЭС.

Необходимо отметить, что отсутствие среди участников разработки инженеров по знаниям (т. е. их замена программистами) либо приводит к неудаче процесс создания ЭС, либо значительно удлиняет его.

Эксперт определяет знания (данные и правила), характеризующие проблемную область, обеспечивает полноту и правильность введенных в ЭС знаний.

Инженер по знаниям помогает эксперту выявить и структурировать знания, необходимые для работы ЭС; осуществляет выбор того ИС, которое наиболее подходит для данной проблемной области, и определяет способ представления знаний в этом ИС; выделяет и программирует (традиционными средствами) стандартные функции (типичные для данной проблемной области), которые будут использоваться в правилах, вводимых экспертом.

Программист разрабатывает ИС (если ИС разрабатывается заново), содержащее в пределе все основные компоненты ЭС, и осуществляет его сопряжение с той средой, в которой оно будет использовано.

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

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

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

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

4.3 Этапы разработки экспертных систем

Разработка ЭС имеет существенные отличия от разработки обычного программного

продукта. Опыт создания ЭС показал, что использование при их разработке методологии, принятой в традиционном программировании, либо чрезмерно затягивает процесс создания ЭС, либо вообще приводит к отрицательному результату.

Использовать ЭС следует только тогда, когда разработка ЭС *возможна, оправдана* и методы инженерии знаний *соответствуют* решаемой задаче. Чтобы разработка ЭС была *возможной* для данного приложения, необходимо одновременное выполнение по крайней мере следующих требований:

- существуют эксперты в данной области, которые решают задачу значительно лучше, чем начинающие специалисты;
- эксперты сходятся в оценке предлагаемого решения, иначе нельзя будет оценить качество разработанной ЭС;
- эксперты способны вербализовать (выразить на естественном языке) и объяснить используемые ими методы, в противном случае трудно рассчитывать на то, что знания экспертов будут "извлечены" и вложены в ЭС;
- решение задачи требует только рассуждений, а не действий;
- задача не должна быть слишком трудной (т.е. ее решение должно занимать у эксперта несколько часов или дней, а не недель);
- задача хотя и не должна быть выражена в формальном виде, но все же должна относиться к достаточно "понятной" и структурированной области, т.е. должны быть выделены основные понятия, отношения и известные (хотя бы эксперту) способы получения решения задачи;
- решение задачи не должно в значительной степени использовать "здравый смысл" (т.е. широкий спектр общих сведений о мире и о способе его функционирования, которые знает и умеет использовать любой нормальный человек), так как подобные знания пока не удается (в достаточном количестве) вложить в системы искусственного интеллекта.

Использование ЭС в данном приложении может быть возможно, но не оправдано. Применение ЭС может быть *оправдано* одним из следующих факторов:

- решение задачи принесет значительный эффект, например экономический;
- использование человека-эксперта невозможно либо из-за недостаточного количества экспертов, либо из-за необходимости выполнять экспертизу одновременно в различных местах;
- использование ЭС целесообразно в тех случаях, когда при передаче информации эксперту происходит недопустимая потеря времени или информации;
- использование ЭС целесообразно при необходимости решать задачу в окружении, враждебном для человека.

Приложение *соответствует* методам ЭС, если решаемая задача обладает совокупностью следующих характеристик:

- задача может быть естественным образом решена посредством манипуляции с символами (т.е. с помощью символических рассуждений), а не манипуляций с числами, как принято в математических методах и в традиционном программировании;
- задача должна иметь эвристическую, а не алгоритмическую природу, т.е. ее решение должно требовать применения эвристических правил. Задачи, которые могут быть гарантированно решены (с соблюдением заданных ограничений) с помощью некоторых формальных процедур, не подходят для применения ЭС;
- задача должна быть достаточно сложна, чтобы оправдать затраты на разработку ЭС. Однако она не должна быть чрезмерно сложной (решение занимает у эксперта часы, а не недели), чтобы ЭС могла ее решать;
- задача должна быть достаточно узкой, чтобы решаться методами ЭС, и практически значимой.

При разработке ЭС, как правило, используется концепция "быстрого прототипа". Суть этой концепции состоит в том, что разработчики не пытаются сразу построить конечный

продукт. На начальном этапе они создают прототип (прототипы) ЭС. Прототипы должны удовлетворять двум противоречивым требованиям: с одной стороны, они должны решать типичные задачи конкретного приложения, а с другой - время и трудоемкость их разработки должны быть весьма незначительны, чтобы можно было максимально запараллелить процесс накопления и отладки знаний (осуществляемый экспертом) с процессом выбора (разработки) программных средств (осуществляемый инженером по знаниям и программистом). Для удовлетворения указанным требованиям, как правило, при создании прототипа используются разнообразные средства, ускоряющие процесс проектирования.

Прототип должен продемонстрировать пригодность методов инженерии знаний для данного приложения. В случае успеха эксперт с помощью инженера по знаниям расширяет знания прототипа о проблемной области. При неудаче может потребоваться разработка нового прототипа или разработчики могут прийти к выводу о непригодности методов ЭС для данного приложения. По мере увеличения знаний прототип может достигнуть такого состояния, когда он успешно решает все задачи данного приложения. Преобразование прототипа ЭС в конечный продукт обычно приводит к перепрограммированию ЭС на языках низкого уровня, обеспечивающих как увеличение быстродействия ЭС, так и уменьшение требуемой памяти. Трудоемкость и время создания ЭС в значительной степени зависят от типа используемого инструментария.

В ходе работ по созданию ЭС сложилась определенная технология их разработки, включающая шесть следующих этапов: идентификацию, концептуализацию, формализацию, выполнение, тестирование, опытную эксплуатацию.

На этапе *идентификации* определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей.

На этапе *концептуализации* проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач.

На этапе *формализации* выбираются ИС и определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями.

На этапе *выполнения* осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и наиболее трудоемким этапом разработки ЭС. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

Тема 5: Нечеткие знания и способы их обработки

1. Основы теории нечетких множеств
2. Виды нечеткости знаний.
3. Операции с нечеткими знаниями.

Основы теории нечетких множеств

При попытке формализовать человеческие знания исследователи вскоре столкнулись с проблемой, затруднявшей использование традиционного математического аппарата для их описания. Существует целый класс описаний, оперирующих качественными характеристиками объектов (*много, мало, сильный, очень сильный* и т.п.). Эти характеристики обычно размыты и не могут быть однозначно интерпретированы, однако содержат важную информацию (например, «Одним из возможных признаков гриппа является *высокая* температура»).

Кроме того, в задачах, решаемых интеллектуальными системами, часто приходится пользоваться неточными знаниями, которые не могут быть интерпретированы как полностью истинные или ложные (логические true/false или 0/1). Существуют знания, достовер-

ность которых выражается некоторой промежуточной цифрой, например 0.7.

Как, не разрушая свойства размытости и неточности, представлять подобные знания формально? Для разрешения таких проблем в начале 70-х американский математик *Лотфи Заде* предложил формальный аппарат нечеткой (fuzzy) алгебры и нечеткой логики. Позднее это направление получило широкое распространение и положило начало одной из ветвей ИИ под названием – *мягкие вычисления* (soft computing).

Л. Заде ввел одно из главных понятий в нечеткой логике – понятие лингвистической переменной.

Лингвистическая переменная (ЛП) – это переменная, значение которой определяется набором вербальных (то есть словесных) характеристик некоторого свойства.

Например, ЛП «рост» определяется через набор {карликовый, низкий, средний, высокий, очень высокий}.

Значения лингвистической переменной (ЛП) определяются через так называемые *нечеткие множества* (НМ), которые в свою очередь определены на некотором базовом наборе значений или базовой числовой шкале, имеющей размерность. Каждое значение ЛП определяется как нечеткое множество (например, НМ «низкий рост»).

Нечеткое множество определяется через некоторую базовую шкалу В и функцию принадлежности НМ – $\mu(x)$, $x \in V$ принимающую значения на интервале [0...1]. Таким образом, нечеткое множество В — это совокупность пар вида $(x, \mu(x))$, где $x \in V$. Часто встречается и такая запись:

$$V = \sum_{i=1}^n \frac{x_i}{\mu(x_i)}$$

где x_i – i -е значение базовой шкалы.

Функция принадлежности определяет субъективную *степень уверенности* эксперта в том, что данное конкретное значение базовой шкалы соответствует определяемому НМ. Эту функцию не стоит путать с вероятностью, носящей объективный характер и подчиняющейся другим математическим зависимостям

Например, для двух экспертов определение НМ «высокая» для ЛП «цена автомобиля» в условных единицах может существенно отличаться в зависимости от их социального и финансового положения.

«Высокая_цена_автомобиля_1» - {50000/1 + 25000/0.8 + 10000/0.6 + 5000/0.4} «Высокая_цена_автомобиля_2» - {25000/1 + 10000/0.8 + 5000/0.7 + 3000/0.4}

Пусть перед нами стоит задача интерпретации значений ЛП «возраст», таких как – «молодой» возраст, «преклонный» возраст или «переходный» возраст. Определим «возраст» как ЛП (рис. 5.1). Тогда «молодой», «преклонный», «переходный» будут значениями этой лингвистической переменной более полно, базовый набор значений ЛП «возраст» следующий В – {младенческий, детский, юный, молодой, зрелый, преклонный, старший}

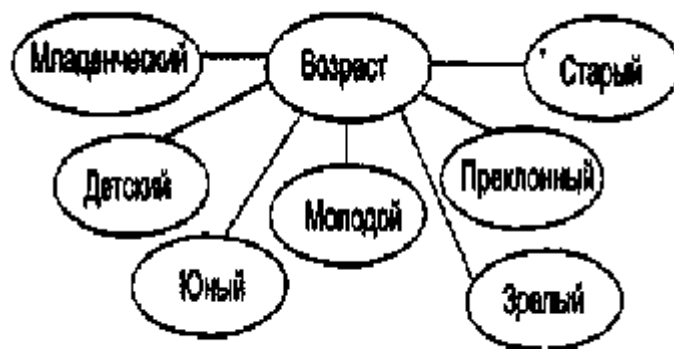


Рисунок 5.1 – Лингвистическая переменная «возраст» и нечеткие множества, определяющие ее значения

Для ЛП «возраст» базовая шкала – это числовая шкала от 0 до 120, обозначающая количество прожитых лет, а функция принадлежности определяет, насколько мы уверены в том, что данное количество лет можно отнести к данной категории возраста на рис. 5.2 отражено, как одни и те же значения базовой шкалы могут участвовать в определении различных НМ.

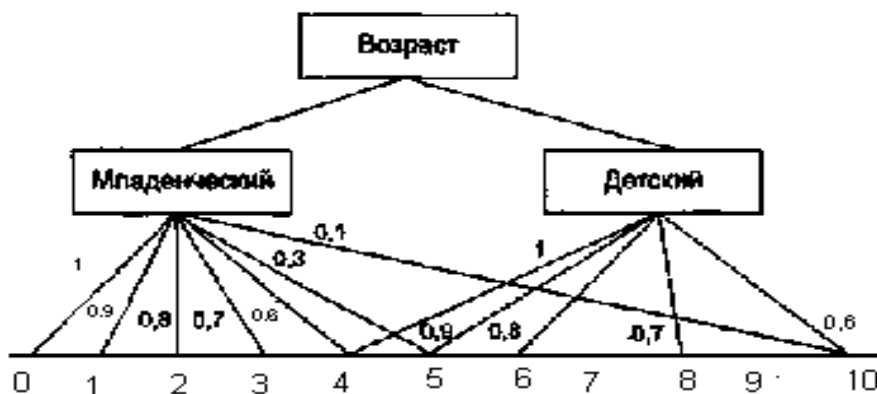


Рисунок 5.2 – Формирование нечетких множеств

Например, определить значение НМ «младенческий возраст» можно так

$$\text{"младенческий"} = \left\{ \frac{0,5}{1} + \frac{1}{0,9} + \frac{2}{0,8} + \frac{3}{0,7} + \frac{4}{0,6} + \frac{5}{0,3} + \frac{10}{0,1} \right\}$$

5.2 Виды нечеткости знаний

Некорректные задачи существуют практически в любой проблемной области. Там, где есть некорректные и слабоструктурированные задачи, можно ожидать эффекта от применения интеллектуальных систем. В области некорректных задач точные знания о проблеме получить невозможно или нельзя это сделать сразу, поэтому там применяется подход, суть которого состоит в постепенном приближении к полному набору необходимых знаний. При этом используются методы представления нечетких знаний и механизмы вывода, работающие в их среде.

Смысл термина нечеткость многозначен. Трудно претендовать на исчерпывающее определение этого понятия, поэтому рассмотрим лишь основные его компоненты, к которым относятся следующие:

- недетерминированность выводов;
- многозначность;
- ненадежность;
- неполнота;
- неточность.

5.3 Операции с нечеткими знаниями

Для операций с нечеткими знаниями, выраженными при помощи лингвистических переменных, существует много различных способов. Эти способы являются в основном эвристиками.

Операция «ИЛИ» часто задается так:

$$\mu(x) = \max(\mu_1(x), \mu_2(x))$$

$$\text{или так: } \mu(x) = \mu_1(x) + \mu_2(x) - \mu_1(x) \cdot \mu_2(x)$$

(вероятностный подход).

Усиление или ослабление лингвистических понятий достигается введением специальных квантификаторов. Например, если понятие «старческий возраст» определяется так:

$$\left\{ \frac{60}{0,6} + \frac{70}{0,8} + \frac{80}{0,9} + \frac{90}{1} \right\}$$

то понятие «очень старческий возраст» определяется так

$$\text{con}(A) = A^2 = \sum_i \frac{x_i}{\mu_i^2}$$
$$\left\{ \frac{60}{0,36} + \frac{70}{0,64} + \frac{80}{0,81} + \frac{90}{1} \right\}$$

Для вывода на нечетких множествах используются специальные отношения и операции над ними.

Одним из первых применений теории НМ стало использование коэффициентов уверенности для вывода рекомендаций медицинской системы MYCIN. Этот метод использует несколько эвристических приемов. Он стал примером обработки нечетких знаний, повлиявших на последующие системы.

В настоящее время в большинство инструментальных средств разработки систем, основанных на знаниях, включены элементы работы с НМ, кроме того, разработаны специальные программные средства реализации так называемого нечеткого, вывода, например «оболочка» FuzzyCLIPS.

Тема 6: Программный инструментарий разработки систем основанных на знаниях.

1. Языки программирования для ИИ и языки представления знаний.
2. Инструментальные пакеты для ИИ
3. Применение языка ПРОЛОГ для представления знаний

6.1 Языки программирования для ИИ и языки представления знаний

Необходимость использования средств автоматизации программирования прикладных систем, ориентированных на знания, и в частности ЭС, была осознана разработчиками этого класса программного обеспечения ЭВМ уже давно. По существу, средства поддержки разработки интеллектуальных систем в своем развитии прошли основные стадии, характерные для систем автоматизации программирования.

Оценивая данный процесс с сегодняшних позиций, можно указать в этой области две тенденции. Первая из них как бы повторяет «классический» путь развития средств автоматизации программирования: автокоды – языки высокого уровня – языки сверхвысокого уровня – языки спецификаций. Условно эту тенденцию можно назвать восходящей стратегией в области создания средств автоматизации разработки интеллектуальных систем. Вторая тенденция, нисходящая, связывается со специальными средствами, уже изначально ориентированными на определенные классы задач и методов ИИ. В конце концов, обе эти тенденции, взаимно обогатив друг друга, должны привести к созданию мощного и гибкого инструментария интеллектуального программирования. Но для настоящего этапа в этой области характерна концентрация усилий в следующих направлениях:

1. Разработка систем представления знаний (СПЗ) путем прямого использования широко распространенных языков обработки символьной информации и, все чаще, языков программирования общего назначения.
2. Расширение базисных языков ИИ до систем представления знаний за счет специализированных библиотек и пакетов.
3. Создание языков представления знаний (ЯПЗ), специально ориентированных на поддержку определенных формализмов, и реализация соответствующих трансляторов с этих языков.

На начальном этапе развития ИИ языков и систем, ориентированных специально на создание прикладных систем, основанных на знаниях, не существовало. С одной стороны, в то время еще не оформился сам подход, в котором центральное место отводилось бы изложению теории в форме программ, а с другой – сама область ИИ только зарождалась как научное направление. Немаловажным было и то, что появившиеся к тому времени универсальные языки программирования высокого уровня казались адекватным инструментом для соз-

дания любых, в том числе и интеллектуальных, систем. Однако сложность и трудоемкость разработки здесь настолько велики, что практически полезные интеллектуальные системы становятся недоступными для реализации. Учитывая вышесказанное, были разработаны языки и системы обработки символьной информации, которые на несколько десятилетий стали основным инструментом программирования интеллектуальных систем.

До недавнего времени наиболее популярным базовым языком реализации систем ИИ вообще и ЭС, в частности, был ЛИСП.

6.2. Инструментальные пакеты для ИИ

Развитые среды автоматизации программирования на базе языков символьной обработки являются необходимым технологическим уровнем систем поддержки разработки прикладных интеллектуальных систем. Как правило, такие среды покрывают (и то частично) подсистему автоматизации проектирования и программирования. Вот почему следующим этапом в развитии инструментальных средств стала ориентация на среды поддержки разработок интеллектуальных систем.

Анализ существующих инструментальных систем показывает, что сначала в области ИИ более активно велись работы по созданию интеллектуальных систем автоматизированного синтеза исполнительных программ. И это естественно, если иметь в виду, что инструментарий ИИ является, по существу, эволюционным развитием систем автоматизации программирования. При этом основная доля мощности и интеллектуальности такого инструментария связывалась не с его архитектурой, а с функциональными возможностями отдельных компонентов той или иной технологической среды. Большое значение при разработке инструментария для ИИ уделялось и удобству сопряжения отдельных компонентов. Пожалуй, именно здесь были получены впечатляющие результаты и именно здесь наиболее широко использовались последние достижения теории и практики программирования, такие, например, как синтаксически-ориентированное редактирование и инкрементная компиляция. Вместе с тем подавляющее большинство современных инструментальных систем «не знают», что проектирует и реализует с их помощью пользователь. И с этой точки зрения можно сказать, что все такие системы являются не более чем «сундучками» с инструментами, успех использования которых определяется искусством работающего с ними мастера. Примерами подобных сред служит подавляющее большинство инструментальных пакетов и систем-оболочек для создания экспертных систем типа EXSYS, GURU и др. Однако не они определяют на сегодняшний день уровень достижений в этой области. К первому эшелону большинство специалистов относит системы ART, KEE и Knowledge Craft. Заметим, что в последнее время в класс самых мощных и развитых систем вошла и среда G2. Все эти системы, во-первых, отличаются тем, что это, безусловно, интегрированные среды поддержки разработки интеллектуальных (в первую очередь, экспертных) систем. И вместе с тем для этих систем характерно не эклектичное объединение различных полезных блоков, но тщательно сбалансированный их отбор, что позволило сделать первые шаги от автоматизации программирования систем ИИ к технологическим системам поддержки проектирования сначала экспертных, а затем и других интеллектуальных систем.

6.3 Применение языка ПРОЛОГ для представления знаний

В отличие от подавляющего большинства других языков Пролог (Prolog) обычно рассматривается в одном контексте с понятием "логическое программирование". По прогнозам, в скором будущем решая задачу, человек не будет задавать последовательность команд на компьютер-ориентированном языке, а будет описывать ее в совершенно абстрактных логических терминах, не оперирующих определениями "байт" или "указатель". То есть он будет создавать модель анализируемой проблемы и пытаться получить положительные или отрицательные результаты этого анализа. Результатом такого анализа может быть, например, оптимизированная программа на процедурном языке (например, C++).

Строго говоря, Пролог не является языком программирования в чистом виде. С одной стороны, это оболочка экспертной системы, с другой - интеллектуальная база данных, что самое важное, не реляционная. Математическая модель, лежащая в основе Пролога, доволь-

но сложна, и по мощности системы формирования запросов к базе с этим языком не сравнится ни одна из коммерческих СУБД.

Фактически Пролог является не процедурным, а декларативным языком. И относится к группе пост объектно-ориентированных языков – функциональным языкам. Человек лишь описывает структуру задачи, а Пролог-машина вывода сама ищет решение. Более того, здесь вообще не существует понятия последовательности команд, все это скрыто в математической модели языка. Однако справедливо заметим, что хотя в идеальной модели Пролог-машины последовательность целей не играет роли, в ее реализациях от порядка следования целей зависит скорость работы ЭС, а порой и ее работоспособность.

Математическая модель Пролога основана на теории исчисления предикатов, в частности, на процедурной интерпретации Хорновых дизъюнктов, содержащих не более одного заключения, Роберта Ковальского из Эдинбурга. Алан Колмероз, автор языка Пролог, начал работы над полноценной компьютерной реализацией трудов Ковальского с 1972 года во французском университете Марсель-Экс. Он составил алгоритм формального способа интерпретации процесса логического вывода и разработал систему автоматического доказательства теорем, которая была написана на Фортране. Она-то и послужила прообразом Пролога. Название его произошло от *Programmation en Logique* - ЛОГическое ПРОграммирование. Вскоре появились первые компиляторы с этого языка, в частности, прекрасная реализация Дэвида Уоррена для компьютера DEC-10 в Эдинбурге, ставшая своего рода стандартом вплоть до сегодняшнего дня. Эффективность этой версии заставила специалистов по искусственному интеллекту по-новому взглянуть на Пролог. В некоторых приложениях, типичных для Лиспа, таких как обработка списков, Пролог уже не уступал своему конкуренту, что и послужило в дальнейшем стимулом для ряда специалистов по логическому программированию к переходу на этот язык.

В качестве типовых данных Пролог использует элементарные единицы данных, так называемые атомы - строки символов и числа. Из атомов составляются списки и бинарные деревья. Сама "программа" строится из последовательности фактов и правил, и затем формулируется утверждение, которое Пролог будет пытаться доказать с помощью введенных правил. Таким способом можно описывать очень сложные проблемы, которые будут решаться самим Прологом *автоматически*. Это происходит с помощью метода сопоставления и рекурсивного поиска. Рекурсия играет в Прологе большую роль.

С распространением Пролога появилась возможность создавать интеллектуальные нереляционные базы знаний с иерархической структурой на основе стандартного механизма с гибкой организацией очень сложных запросов. Были написаны эффективные программы для решения переборных задач, в частности, из области молекулярной биологии и проектирования СБИС, где требовалось учитывать либо сложные внутренние структуры, либо большое число правил, описывающих организацию объекта. Пролог хорошо зарекомендовал себя в качестве экспертной оболочки и при решении задач грамматического разбора. Что весьма характерно, первый высокопроизводительный компилятор этого языка (Эдинбургская версия) был написан на самом Прологе. И немудрено, ведь все формальные синтаксические описания грамматик в Бэкус-форме прекрасно записываются в терминах Пролога.

Долгое время среди разработчиков этого языка шла напряженная борьба между сторонниками оригинальной семантики Пролога и специалистами, стремившимися пожертвовать ясностью структуры языка ради повышения эффективности реализации. В частности, стала играть роль последовательность правил в базе данных. Дело в том, что нередко для получения быстрого ответа оптимально использовать сначала, например, наиболее простые правила, или наиболее эффективные с точки зрения человека. Программа на Прологе постепенно стала приближаться к обычным процедурным - последовательным языкам. Немалую роль в этом сыграло и искусственно введенное понятие отсечения, своего рода аналог столь нелюбимого Дейкстры *goto*. Теперь программист мог по своему усмотрению динамически отсекал бесплодные, по его мнению, ветви деревьев перебора, что приводило к многократному (на два-три порядка) повышению скорости работы программ, но при этом сильно наруша-

лась ясность ее структуры и возникало множество проблем, связанных с отладкой.

К счастью, развитие вычислительной техники в сочетании с уникальной структурой языка дало свои результаты. При появлении первых параллельных компьютеров люди, программирующие на Прологе, быстро осознали пагубность различных "нововведений" типа оператора отсечения, лишивших язык оригинальной чистоты, и вернулись к первоначальной версии языка. Пресловутая последовательность правил перестала играть роль, так как появилась возможность вычислять их параллельно, а в силу того, что математическая теория Пролога не накладывает никаких требований на упорядоченность фактов и правил в базе, то скорость работы программы стала линейно пропорциональной числу процессоров. Имеются бесплатные и коммерческие версии Пролога для реализаций на параллельных компьютерах, например, Densitron CS Prolog для транспьютеров, или Paralagic. В японском проекте компьютера пятого поколения все программное обеспечение создавалось на Пролог подобном языке.

III. МЕТОДИЧЕСКИЕ УКАЗАНИЯ (РЕКОМЕНДАЦИИ)

1 Методические указания по изучению дисциплины

Процесс по освоению всей совокупности теоретического и практического материала по дисциплине должен быть реализован в течение одного семестра и, проходить в соответствии с предложенным выше планом.

Теоретический курс изучается студентом в процессе работы на лекциях, при этом студент конспектирует излагаемый преподавателем материал, отвечает на вопросы, которые ставит преподаватель в процессе чтения лекций, а также в процессе самостоятельной работы с рекомендуемой литературой в библиотеке университета.

План лекций составлен в соответствии с Государственным образовательным стандартом, учебным планом и программой курса. Лекции раскрывают основное содержание изучаемой дисциплины, являясь основой для более глубокого изучения курса.

В ходе лекций раскрываются основные теоретические вопросы программы дисциплины, делаются акценты на наиболее сложные и интересные положения изучаемого материала. Это становится возможным благодаря тому, что студенты могут заранее распечатать слайды лекции в качестве основы конспекта (презентация лекции высылается на почтовый ящик студентам за день до лекции), а также за счет применения на лекциях мультимедийных технологий. Материалы лекций являются базовыми для подготовки к зачету.

2 Методические указания к лабораторным занятиям

Задачей преподавателя при проведении лабораторных работ является грамотное и доступное разъяснение принципов и правил проведения работ, побуждение студентов к самостоятельной работе, определения места изучаемой дисциплины в дальнейшей профессиональной работе будущего специалиста.

Цель лабораторной работы – научить студентов самостоятельно производить необходимые действия для достижения желаемого результата.

Прежде чем приступить к выполнению лабораторной работы, студенту необходимо ознакомиться с теоретическим материалом, соответствующим данной теме.

Выполнение лабораторной работы целесообразно разделить на несколько этапов:

- формулировка и обоснование цели работы;
- определение теоретического аппарата, применительно к данной теме;
- выполнение заданий;
- анализ результата;
- выводы.

Индивидуальные задания для лабораторных работ должны быть представлены конкретно-практическими и творческими задачами.

На первой ступени изучения темы выполняются конкретно-практические задачи, при решении которых формируется минимальный набор умений. Преподаватель опосредованно руководит познавательной деятельностью студентов, консультирует и подробно разбирает со студентами возникшие затруднения в ходе решения задачи, обращает внимание группы на возможные ошибки.

Вторая ступень изучения темы дифференцируется в зависимости от степени усвоения его обязательного уровня. Студенты, усвоив содержание типовых методов и приемов решения задач, приступают к решению творческих задач. Если уровень знаний и умений, демонстрируемых студентом при контрольном обследовании, не соответствует установленным требованиям, студент вновь возвращается к стандартным упражнениям, но под более пристальным наблюдением преподавателя.

После изучения отдельной темы курса дисциплины, каждый студент получает оценку по результатам выполнения лабораторных работ.

Студенты, пропустившие лабораторные занятия, должны их выполнить во внеаудиторное время и отчитаться до начала зачетно-экзаменационной сессии.

Рекомендации для организации рабочего места студента: для проведения лабораторных работ требуется компьютерный класс с установленным программным обеспечением.

Правила техники безопасности в компьютерном классе:

1. Находиться в компьютерном классе без разрешения преподавателя.
2. Включать без разрешения оборудование.
3. Трогать разъемы соединительных кабелей и проводов.
4. Прикасаться к питающим проводам и устройствам заземления.
5. Включать и выключать аппаратуру без указания преподавателя.
6. Работать в верхней одежде и влажными руками.
7. Класть диски, книги, тетради и другие предметы на монитор и клавиатуру.
8. При появлении запаха гари немедленно прекратите работу, выключите аппаратуру и сообщите об этом преподавателю.

При подготовке к лабораторным занятиям рекомендуется использовать методическое пособие: Основы логического программирования с использованием языка Пролог /Акилова И.М., Назаренко Н.В. Благовещенск, Режим доступа: Библиотечный сервер АмГУ. – 2011. – 120 с. File://10.4.1.254/ Digitallibrary/ AmurSU_Edition/1872.pdf

Выполняя задание, студент пользуется теоретическим материалом приведенном перед заданием к лабораторной работе и учебной литературой.

Выполненную лабораторную работу студент должен продемонстрировать преподавателю на экране компьютера и ответить на контрольные вопросы.

3. Методические указания по самостоятельной работе студентов

Для закрепления полученных теоретических и практических знаний студентам в течение всего учебного года предлагаются индивидуальные задания для самостоятельной работы. Консультирование по выполнению индивидуальных заданий проводится как непосредственно в компьютерных классах (во время консультаций), так и через электронный обмен сообщениями, посредством Интернет. Защита индивидуальных заданий по темам может проводиться в виде Круглого стола, когда каждый студент выступает с презентацией выполненной работы, а преподаватель вместе с остальными студентами оценивает работу. Задания по темам также могут быть выданы студентам в качестве домашних заданий в виде электронных файлов. Контроль выполненных заданий осуществляется либо непосредственно на занятиях, либо на консультациях.

В рабочей программе п.5 представлены виды самостоятельной работы по каждой теме дисциплины и трудоемкость в часах.

Содержание работы	Количество часов	Сроки сдачи	Форма контроля
--------------------------	-------------------------	--------------------	-----------------------

1. Выбор темы	1	1 неделя	Письменный отчет в конце семестра
2. Построение модели предметной области (ПО)	3	2-3 неделя	
3. Построение продукционной модели предметной области	4	4-5 неделя	
4. Построение модели ПО на языке исчисления высказываний.	6	6-7 неделя	
5. Построение модели ПО на языке исчисления предикатов	6	8-9 неделя	
6. Построение модели ПО в виде семантической сети.	6	10-11 неделя	
7. Построение модели ПО в виде сети фреймов.	6	12-13неделя	
8. Оформление отчета	2	14 неделя	
Защита отчета	2	15 неделя	

Студенты выбирают одну из приведенных ниже тем, и на ее основе выполняют все 6 ниже перечисленных задач. Указанные темы не являются обязательными. Студент вправе предложить свою, близкую ему, тему.

Отчет по самостоятельной работе студенты сдают в письменном виде, оформленный с учетом требований к оформлению письменных работ. В конце работы необходимо привести перечень использованной литературы и других источников (ссылки на Internet и пр.).

Индивидуальная самостоятельная работа включает одно задание, которое состоит из 6-ти задач.

1. Построение модели предметной области для данной сферы деятельности человека.
 2. Построение продукционной модели заданной предметной области в виде базы фактов (данных) и базы знаний и имитация работы этой модели в течение 5 тактов.
 3. Построение фрагмента модели заданной предметной области на языке исчисления высказываний.
 4. Построение фрагмента модели заданной предметной области на языке исчисления предикатов путём соответствующего усложнения.
 5. Построение фрагмента модели предметной области в виде семантической сети.
 6. Построение фрагмента предметной области в виде сети фреймов.
- Защита индивидуальных заданий будет происходить на лабораторном занятии в аудитории. Каждое выступление должно сопровождаться презентацией.

IV. КОНТРОЛЬ ЗНАНИЙ

В течении семестра знания студентов оцениваются с использованием рейтинговой системы, которая складывается из оценки за работу в семестре – максимально 60 баллов и экзаменационной оценки – максимально 40 баллов. Максимально возможное количество равно 100. Баллы по разным модулям приведены в рабочей программе п.12.

Минимальное количество баллов в семестре, необходимое для получения студентом допуска на зачет – 40 баллов.

Минимальное количество баллов за выполнение экзаменационной работы, необходимое для получения оценки: «удовлетворительно» - 15 баллов, «хорошо» - 20 баллов, «отлично» - 30 баллов.

Соответствие итогового рейтинга студента и традиционных оценок устанавливается по следующей шкале: «неудовлетворительно» - 0-50 баллов, «удовлетворительно» - 51-75 баллов, «хорошо» - 76-90 баллов, «отлично» - 91-100 баллов.

В качестве основных средств текущего контроля используется тестирование. В качестве дополнительной формы текущего контроля предлагаются аудиторные и внеаудиторные письменные задания (контрольные работы).

1 Текущий контроль знаний

Текущий контроль за аудиторной и самостоятельной работой обучаемых осуществляется во время проведения аудиторных занятий посредством устного опроса, проведения контрольных работ или осуществления лекции в форме диалога. Промежуточный контроль осуществляется два раза в семестр в виде анализа итоговых отчетов на аттестационные вопросы.

В качестве основных средств текущего контроля используется тестирование. В качестве дополнительной формы текущего контроля предлагаются аудиторные и внеаудиторные письменные задания (контрольные работы).

Образец тестовых заданий

1. Проблемами создания программ, с помощью которых можно решать те задачи, решение которых до этого считалось прерогативой человека, занимается:

- а) бионическое направление;
- б) программно-прагматическое направление;
- в) нейрофизиологическое направление.

2. Что такое экстенционал понятия:

- а) набор близких по смыслу понятий;
- б) расширение данного понятия;
- в) набор конкретных фактов соответствующих данному понятию.

3. Получение инженером по знаниям наиболее полного из возможных представлений о предметной области и способах принятия решения в ней:

- а) извлечение знаний;
- б) приобретение знаний;
- в) формирование знаний.

4. В семантической сети объектам, концепциям, событиям или понятиям обычно соответствуют:

- а) отношения;
- б) узлы;
- в) дуги.

5. Метод представления знаний, в котором свойства описываются в терминах атрибутов и их значений и связываются с вершинами, представляющими концепции или объекты:

- а) фрейм;
- б) семантическая сеть;
- в) логическая модель.

6. Раздел программы на ПРОЛОГе предназначенный для определения пользовательских типов данных:

- а) `domeins`;
- б) `clauses`;
- в) `domains`.

7. Выполнение этой программы приведет к:

```
clauses
road(«Cansas»,«Tampa»).
road(«Tampa»,«Cansas»).
road(«Tampa»,«Houston»).
```

goal:

```
road(_,Y)
```

- а) синтаксической ошибке;
- б) присвоению атому Y значения «Тампа»;
- в) получению списка («Тампа», («Cansas», «Houston».

8. Декларативный смысл Пролог-программы определяет:

- а) что должно быть результатом программы;
- б) как результат был достигнут;
- в) как результат будет достигнут.

9. X является элементом списка, если в соответствии с первым предложением X – это голова списка, или в соответствии со вторым предложением X – это элемент хвоста:

- а) `member(X,[X|_]).`
`member(X,[_|Y]):- member(X,Y).`
- б) `member(X,[]).`
`member(X,[X|Y]):- member(X,Y).`
- в) `member(_,[_|Y]).`
`member(X,[X|_]):- member(X,Y).`

2 Итоговый контроль знаний

В рабочей программе п.7 представлены вопросы к зачету.

V. ИНТЕРАКТИВНЫЕ ТЕХНОЛОГИИ И ИННОВАЦИОННЫЕ МЕТОДЫ, ИСПОЛЗУЕМЫЕ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Образовательный процесс по дисциплине строится на основе комбинации следующих образовательных технологий.

Интегральную модель образовательного процесса по дисциплине формируют технологии методологического уровня: модульно-рейтинговое обучение, технология поэтапного формирования умственных действий, технология развивающего обучения, элементы технологии развития критического мышления.

Рекомендуется использование информационных технологий при организации коммуникации со студентами для представления информации, выдачи рекомендаций и консультирования по оперативным вопросам (электронная почта), использование мультимедиа-средств при проведении лекционных и лабораторных занятий.

Игровые имитационные методы:

Мозговой штурм – наиболее свободная форма дискуссии, позволяющей быстро включить в работу всех членов учебной группы. Используется там, где требуется генерация разнообразных идей, их отбор и критическая оценка. Этапы продуцирования идей и их анализа намеренно разделены: во время выдвижения идей запрещается их критика.

Круглый стол – это метод активного обучения, одна из организационных форм познавательной деятельности учащихся, позволяющая закрепить полученные ранее знания, восполнить недостающую информацию, сформировать умения решать проблемы, укрепить позиции, научить культуре ведения дискуссии.

Дискуссия – это всестороннее обсуждение спорного вопроса в публичном собрании, в частной беседе, споре.

Деловая игра – форма воссоздания предметного и социального содержания профессиональной деятельности, моделирования систем отношений, разнообразных условий профессиональной деятельности, характерных для данного вида практики.

Метод анализа конкретной ситуации (ситуационный анализ, анализ конкретных ситуаций, case-study) – это педагогическая технология, основанная на моделировании ситуации или использования реальной ситуации в целях анализа данного случая, выявления проблем, поиска альтернативных решений и принятия оптимального решения проблем.

Мастер-класс – это главное средство передачи концептуальной новой идеи своей (авторской) педагогической системы. Преподаватель как профессионал на протяжении ряда лет вырабатывает индивидуальную (авторскую) методическую систему, включающую целеполагание, проектирование, использование последовательности ряда известных дидактических и воспитательных методик, занятий, мероприятий, собственные «ноу-хау», учитывает реальные условия работы с различными категориями учащихся и т.п.