


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

УТВЕРЖДЕНО
решением Ученого совета ИЭиБ
от « 22 » июня 2023 г., протокол № 09 / 261
Председатель  И.Б.Романова
« 22 » июня 2023 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	Прикладные системы искусственного интеллекта
Факультет	Экономики
Кафедра	Цифровая экономика
Курс	1


Направление: 38.04.01 Экономика
 Направленность (профиль): «Искусственный интеллект в финансово-экономических системах»
 Форма обучения: очная


Дата введения в учебный процесс УлГУ: « 01 » сентября 2023 г.

Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20 ____ г.
 Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20 ____ г.
 Программа актуализирована на заседании кафедры: протокол № _____ от _____ 20 ____ г.

Сведения о разработчиках:

ФИО	Кафедра	Должность, ученая степень, звание
Сковиков А.Г.	Цифровой экономики	к.т.н., доцент

СОГЛАСОВАНО
Заведующий выпускающей кафедрой цифровой экономики
 / Лутошкин И.В. /
Подпись ФИО
«21» июня 2023 г.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

ВВЕДЕНИЕ

Актуальность представленной образовательной программы "Прикладные системы искусственного интеллекта" в рамках направления магистерской подготовки 38.04.01 Экономика обусловлена возросшей потребностью в специалистах, способных к решению профессиональных задач по модификации, внедрению и использованию интеллектуальных систем для решения задач информационного поиска, извлечения знаний и поддержки принятия решений.

Направленность курса «Прикладные системы искусственного интеллекта» отвечает целям Федерального проекта «Кадры для цифровой экономики» Национальной программы «Цифровая экономика РФ».

Программа составлена в соответствии с требованиями Федеральных государственных образовательными стандартов высшего образования.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины является изучение теоретических основ искусственного интеллекта и проектирования систем, основанных на знаниях, областей использования интеллектуальных систем, их возможностей и ограничений; углубленное изучение теории и практики методов и средств представления и обработки знаний в системах искусственного интеллекта.

Задачи дисциплины:


1. освоение базовых знаний в области искусственного интеллекта и проектирования систем, основанных на знаниях;
2. приобретение теоретических знаний в части представления и обработки знаний в практически значимых предметных областях;
3. проведение собственных теоретических и экспериментальных исследований в области искусственного интеллекта и проектирования прикладных интеллектуальных систем;
4. приобретение навыков работы с инструментальными средствами представления и обработки знаний, а также с прикладными интеллектуальными системами в Интернет.

В результате изучения курса обучающиеся должны освоить способы самоорганизации и саморазвития на основе комплексного представления о том, как новые технологии изменяют нашу жизнь и жизнь будущих поколений, как они преобразуют экономическую, социальную, культурную и гуманитарную среду нашего обитания.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Прикладные системы искусственного интеллекта» относится к дисциплинам части, формируемой участниками образовательных отношений в системе подготовки магистров по направлению 38.04.01 Экономика, профиль «Искусственный интеллект в финансово-экономических системах».

Дисциплина рассчитана на обучающихся, имеющих начальную подготовку в области информационных технологий и систем, глобальных сетей, организации и инфраструктуры предпринимательской деятельности, коммерции, алгебры, теории вероятности, знакомых с основными понятиями физики, комбинаторики, информатики. Помимо этого, для успешного освоения данного курса магистранту необходимы навыки самостоятельной работы с различными источниками информации (интернет, печатные издания), умение обобщать информацию, полученную из разных источников, умение представлять результаты своих исследований.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


Изучаемая дисциплина является основой для продолжения формирования указанных компетенций в следующих практиках: Производственная практика. Практика по профилю профессиональной деятельности, Преддипломная практика и процедуре государственной итоговой аттестации: Подготовка к процедуре защиты и защита выпускной квалификационной работы.

Дисциплина направлена на изучение основных теоретических положений и методов, формирование умений и привитие навыков применения теоретических знаний для решения прикладных задач, а также развитие новых подходов к применению интеллектуальных технологий в сфере экономики.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения дисциплины формируются следующие компетенции:

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ПК-7 Способен руководить проектами по созданию комплексных систем на основе аналитики больших данных в различных отраслях со стороны заказчика	Знать методологию и принципы руководства проектами по созданию, поддержке и использованию комплексных систем на основе аналитики больших данных со стороны заказчика; специфику сфер и отраслей, для которых реализуется проект по аналитике больших данных Уметь решать задачи по руководству коллективной проектной деятельностью для создания, поддержки и использования комплексных систем на основе аналитики больших данных со стороны заказчика; выявлять небольшие по масштабу проекты аналитики, которые потенциально могут представлять интерес для ряда подразделений, служб или для организации в целом; выявлять области деловой деятельности, которые потенциально могут получить отдачу от аналитики Владеть способностью осуществлять руководство проектами по построению комплексных систем на основе аналитики больших данных в различных отраслях со стороны заказчика

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачетных единицах (всего) 5 ЗЕТ

4.2. Объем дисциплины по видам учебной работы (в часах)


Вид учебной работы	Количество часов (форма обучения – очная)	
	Всего по плану	в т.ч. по семестрам
		2
Контактная работа обучающихся с преподавателем	54*	54*
Аудиторные занятия:		
лекции	18*	18*
практические и семинарские занятия	36*	36*
лабораторные работы, практикумы	-	-
Самостоятельная работа	126	126
Форма текущего контроля знаний и контроля самостоятельной работы	Тестирование, устный опрос, решение кейсов	Тестирование, устный опрос, решение кейсов
Виды промежуточной аттестации	Диф.зачет	Диф.зачет
Всего часов по дисциплине	180	180

*В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, указанные часы работы ППС с обучающимися проводятся в дистанционном формате с применением электронного обучения

4.3. Содержание дисциплины (модуля.) Распределение часов по темам и видам учебной работы:

Форма обучения: очная

Название и разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		лекции	практические занятия, семинары	лабораторные работы			
1	2	3	4	5	6	7	
Тема 1. Введение в прикладные системы искусственного интеллекта.	36	2	4		2	30	проверка тестовых заданий, устный опрос, проверка кейсов
Тема 2. Системы, основанные на знаниях	57	9	16		4	32	проверка тестовых заданий, устный опрос, проверка кейсов, оценивание рефератов

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Тема 3. Прикладные системы искусственного интеллекта	87	7	16		6	64	проверка тестовых заданий, устный опрос, проверка кейсов, оценивание рефератов
Итого	180	18	36		12	126	

5. СОДЕРЖАНИЕ КУРСА

Тема 1. Введение в прикладные системы искусственного интеллекта


1. Исторический обзор применения искусственного интеллекта. Обзор современных приложений искусственного интеллекта.
2. Задачи, решаемые прикладными системами искусственного интеллекта (ПСИИ). Классификация ПСИИ. Критерии выбора ПСИИ.

Тема 2. Системы, основанные на знаниях

1. Системы основанные на знаниях. Понятие инженерии знаний. Этапы получения знаний о предметной области. Методы получения знаний о предметной области.
2. Инструмент Microsoft Analysis Services в процессе анализа данных
3. Экспертные системы. Классификация экспертных систем. Архитектура экспертных систем. Применение экспертных систем в профессиональной деятельности.
4. Этапы и методы создания и применения экспертных систем. Оболочки экспертных систем. Использование оболочек экспертных систем при создании конкретных экспертных систем.
5. Применение оболочки для создания экспертных систем CLIPS 6.31
6. Знания. Методы представлений знаний. Формализация базы знаний. Обработка знаний. Извлечение знаний из данных. Применение баз знаний в профессиональной деятельности.
7. Онтологические модели представления знаний о предметной области. Применение онтологий в решении профессиональных задач.
8. Semantic Web (Семантическая паутина). Поиск знаний в семантической паутине.
9. Применение редактора Protege 5.5.0 для создания и наполнения онтологий.

Тема 3. Прикладные системы искусственного интеллекта

1. Понимание естественного языка. Обработка текстов на естественном языке. Применение NLP технологий в профессиональной деятельности.
2. Автоматический машинный перевод.
3. Обзор NLP библиотеки spaCy. Применение ПО Natural Language Toolkit и ПО General Architecture for Text Engineering.
4. Автоматическое доказательство теорем.
5. Интеллектуальные игры.
6. Искусственный интеллект в промышленных системах.
7. Интеллектуальные САПР и АСУ.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

Цель проведения семинарских и практических занятий заключается в закреплении полученных теоретических знаний на лекциях и в процессе самостоятельного изучения студентами специальной литературы. Основной формой проведения семинарских и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам.

Часть практических занятий проводится в интерактивной форме: обсуждение проблемных вопросов на круглых столах.

В обязанности преподавателя входят оказание методической помощи и консультирование магистрантов по соответствующим темам курса.

№ п/п	№ темы	Тематика практических занятий	Кол-во часов
1	1	Практическое занятие. Исследование применяемых в профессиональной деятельности прикладных систем искусственного интеллекта.	4
2	2	Практическое занятие. Создание базы знаний о предметной области	4
3	2	Практическое занятие. Создание и применение экспертной системы в оболочке CLIPS 6.3	6
4	2	Практическое занятие. Создание спецификации знаний о предметной области в виде онтологии в редакторе Protege 5.6.2	6
5	3	Практическое занятие. Анализ и выбор соответствующих инструментов и средств, основанных на современных интеллектуальных технологиях для решения конкретной профессиональной задачи.	6
6	3	Практическое занятие. Решение профессиональной задачи путём использования современных интеллектуальных технологий.	10
		Итого:	36

Тема 1. Введение в прикладные системы искусственного интеллекта


Практическое занятие. Исследование применяемых в профессиональной деятельности прикладных систем искусственного интеллекта.

Вопросы для рассмотрения на семинарских занятиях. Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.

Форма проведения занятия – практическое занятие.

Вопросы для самоконтроля и текущего контроля

1. Понятие искусственного интеллекта.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

2. Три волны искусственного интеллекта
3. Направления ИИ
4. Что достигнуто в ИИ к настоящему времени?
5. Почему вокруг ИИ такой ажиотаж (ожидания)
6. Рынок ИИ
7. Основные международные программы и центры разработки
8. Состояние работ по ИИ в РФ
9. Основные направления работ в области ИИ
10. Тенденции использования систем ИИ в области государственного управления
11. Ключевые вызовы и угрозы развития систем ИИ

Прикладные системы искусственного интеллекта

Проблемой человеческого интеллекта исследователи занимаются давно, хотя попытки структурировать это явление психики, или хотя бы четко отделить его от других явлений и процессов (например, мышления) не всегда результативны. На сегодняшний день в психологии принято считать, что существует некий общий интеллект как универсальная психическая способность познания, в основе которой может лежать генетически обусловленное свойство нервной системы перерабатывать информацию с определенной скоростью и точностью. Считается, что многие виды умственной деятельности человека, например программирование, занятие математикой, проведение публичных лекций, или чтение книг, подобных этой, требуют интеллекта. Мы видим, что понятие «интеллект» для человека определяется косвенно, через предметную область, по некоторым результатам деятельности, а не напрямую указывается, что собой представляет и как реализуется этот аспект психики человека. Прямые определения, которые можно найти в энциклопедических изданиях, как правило, дают толкование, используя синонимические слова. Например, энциклопедический словарь дает следующее краткое определение.


Интеллект (от лат. intellectus — познание, понимание, рассудок) — способность мышления, рационального познания.

Толковый психологический словарь близок, но отличается большей детализацией в определении.

Интеллект (от лат. intellectus — понимание, познание) — способность к осуществлению процесса познания и к эффективному решению задач, в частности при овладении новым кругом жизненных проблем.

Другими словами, определения ничего не определяют, но могут иногда стимулировать верные ассоциации.

Предметом этого курса является прикладной искусственный интеллект (ИИ). Это понятие созвучно понятию интеллекта естественного, но имеет с ним, на самом деле, мало общего. Важно понимать, что ИИ не является моделью человеческого мышления¹, так же как, например, резина не является моделью каучука, хотя обладает рядом схожих свойств. Действительно, программа с искусственным интеллектом использует некоторые алгоритмы, свойственные так же и человеческому мышлению (в конце концов, любая программа написана людьми). Но Однако в реальной жизни человек будет решать ту же задачу *иначе*, чем это делает программа (производя другие операции, используя другие алгоритмы, задействуя структуры психики, пока что недостаточно изученные, а потому не поддающиеся переводу в математические формулы и программный код). Лобовой подход — изложение «человеческих» алгоритмов «машинным» языком, как мы увидим в дальнейшем, малоэффективен. Другими словами, если мы относим задачу к интеллектуальным, то любое ее решение, по нашему определению, свидетельствует о некотором интеллекте. Если решение добывается программным путем, то эта программа содержит элементы интеллекта.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Программа, умеющая решать задачи в предметной области, которую традиция относит к интеллектуальным, называется *прикладной системой с элементами² искусственного интеллекта (ПСИИ)*

Другое определение: ПСИИ — это программа, алгоритм которой нельзя найти в вузовском учебнике. Например, на заре программирования, программы, которые могли строить другие программы (первые компиляторы), считались «интеллектуальными». Современные компиляторы не считаются интеллектуальными программами, хотя «знают» про составление программ значительно больше и применяют весьма изощренные алгоритмы решения задач компиляции и оптимизации. Можно приводить и другие примеры. В каждом отдельном случае конкретная ПСИИ — это ноу-хау создателей такого программно-обеспечения, но каждый раз любая новая идея в конкретной предметной области дает существенное продвижение в ПСИИ в целом. При этом, по мере прогресса ПСИИ, меняются и традиции отнесения предметных областей к интеллектуальной сфере и снижается оценка степени интеллектуальности многих вещей. Например, расчет конструкции Эйфелевой башни в конце XIX века потребовал огромного труда, недюжинного интеллекта и изобретательности, а сейчас подобный расчет может быть выполнен рядовым инженером-конструктором с помощью готового пакета программ быстро, надежно и без особых умственных усилий.

Для подкрепления этого тезиса и конкретизации основного определения ПСИИ рассмотрим некоторые из предметных областей, типичных для приложений искусственного интеллекта.

Обзор приложений ИИ

Некоторыми типичными приложениями, использующие элементы ИИ, являются:

- понимание естественного языка;
- машинный перевод;
- интеллектуальные базы данных;
- экспертные системы;
- автоматическое доказательство теорем;
- планирование действий робота;
- распознавание образов;
- интеллектуальные игры.


Этот список не является ни исчерпывающим, ни ортогональным. Это просто набор наших любимых примеров.

Понимание естественного языка

Процесс *понимания естественного языка (ЕЯ)* — это процесс передачи знаний от одного носителя языка к другому.

Несмотря на всю свою кажущуюся простоту и привычность, этот процесс чрезвычайно сложен. Он подразумевает наличие общего контекста понимания у обоих партнеров и согласованных механизмов представления знаний в виде данных, а также наличие обратного механизма интерпретации данных.

Для процесса понимания естественного языка требуются мощные вычислительные ресурсы, связанные с интеллектом. Рассмотрим, в чем здесь состоит проблема. Пусть у нас есть два интеллектуальных агента, для простоты на первых порах пусть это будут люди. Первый является источником знаний, назовем его говорящий, а второй – приемником знаний, назовем его слушающий. В процессе понимания задействовано два механизма. Во-первых, у говорящего есть алгоритм, позволяющий преобразовывать гипотетические знания в данные, сигнал, (текст, речь или другие формы выражения естественного языка³) — *механизм представления знаний*. Во-вторых, у слушающего есть механизм, который позволяет преобразовывать получаемый сигнал обратно в знания — *механизм интерпретации*. Если принять, что эти механизмы являются алгоритмами⁴, то понятно, что у каждого из них есть область определения и область значений, вход и выход. Понимание происходит тогда, когда выход алгоритма представления знания и вход алгоритма интерпретации данных, по меньшей мере, пересекаются; а полное понимание достигается тогда, когда эти области совпадают. Если человек-источник информации говорит на языке, которым

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

приемник информации не владеет, к примеру, если бы эта книга была написана по-китайски, то происходит следующее: механизм представления знаний работает, механизм интерпретации работает, но понимания не происходит.

Общий контекст понимания — это совокупность согласованных механизмов (алгоритмов) представления (вывода) знаний в виде данных одним и последующей интерпретации (ввода) данных другим партнером.

Мудрецу достаточно сказать одно слово, для того, чтобы другой мудрец его понял. Иными словами, для мудрецов общий объем передаваемых данных может быть значительно меньше, чем у среднестатистических людей, за счет более мощного механизма представления знаний и интерпретации данных.

Обратимся теперь к компьютерам. В компьютерах механизмы представления знаний и интерпретации данных тривиальны, понимание сводится к передаче данных. Поэтому если компьютеры соединены каналом связи и форматы данных согласованы, то понимание достигается легко, а в противном случае, понимания нельзя достичь никоим образом. Критическим случаем, находящимся в фокусе нашего внимания, является взаимопонимание человека и компьютера. Существует два принципиально разных способа общения человека с компьютером. Первый из них — когда человек способен понимать компьютер. Это не заслуга компьютера, а способность человека, которого все окружающие называют программистом. Второй — когда компьютер способен понимать человека. Такой компьютер обязательно должен обладать элементами ИИ. Поскольку в аппаратном обеспечении обычного⁵ компьютера трудно разместить интеллект, даже искусственный, ясно, что понимание должно обеспечиваться программно. И такие программы были созданы уже сравнительно давно.

ПРЕКРАСНАЯ ELIZA, КОТОРАЯ ГОВОРИТ ПО-АНГЛИЙСКИ


Одной из первых ПСИИ, понимающих естественный язык, явилась программа ELIZA, созданная в 1966 году Дж. Вейценбаумом.

Позднее Терри Виноград (Terry Winograd) написал книгу «Программа, понимающая естественный язык», описывающую программу ELIZA. Структура программы ELIZA была чрезвычайно проста и включала в себя всего несколько десятков эвристических правил, предназначенных для распознавания грамматических конструкций и преобразования поступающих вопросов в «естественные» ответы. В результате эта программа могла вести вполне правдоподобный диалог, зачастую, при этом, отвечая вопросом на вопрос. Пример диалога с программой ELIZA:

- ELIZA, ты компьютерная программа.
- Почему Вы думаете, что я компьютерная программа?
- Потому что тебя составил Вейценбаум.
- Расскажите мне о Вейценбауме.

Несмотря на всю свою простоту, эта программа находит применение в психотерапии. Разговор психотерапевта с пациентом может продолжаться довольно длительное время, прежде чем достигается терапевтический эффект. ELIZA успешно имитировала диалог с роджерсианским психологом в течении многих часов (суть этого метода заключается в том, что терапевт выслушивает пациента и задает наводящие вопросы, в результате чего психическое состояние последнего улучшается). Причем многие пациенты отмечали, что эти сеансы оказались крайне полезны для них, и отказывались верить, что с ними разговаривала компьютерная программы, а не профессиональный психолог-человек.

Сам Вейценбаум, шокированный поведением своего детища, вскоре пришел к выводу, что «основная идеология развития искусственного интеллекта — искусственный разум — безнравственна».

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Машинный перевод



Важный класс примеров, связанных с пониманием естественного языка, образуют системы автоматического перевода текстов с одного естественного языка на другой.

Перевод с одного естественного языка на другой, несомненно, является интеллектуальной задачей. Автоматический перевод привлек внимание исследователей едва ли не раньше других предметных областей, ввиду явной практической востребованности этой задачи.

Важно, что, как и в других случаях ИИ, в классе систем машинного перевода наблюдается целый ряд систем все возрастающей сложности:

- электронный словарь;
- подстрочник технических текстов;
- художественный перевод поэзии;
- синхронный перевод речи.

Ясно, что эти задачи не могут быть решены единообразно, а значит, их решения появляются в разное время и опираются на различные технологии.


Для каждой новой области применения ИИ наблюдается одна и та же характерная картина, которую мы проиллюстрируем на примере истории развития машинного перевода.

После появления новой идеи к ней проявляется повышенный интерес, возникают завышенные ожидания, пользователи ждут чуда. Но чуда не происходит — методы ИИ трудоемки, сложны, и их развитие требует немалого естественного интеллекта. Наступает разочарование, новое направление уходит из сферы ажиотажного интереса и иногда подвергается даже необоснованной обструкции. Если идея была плодотворна, то, несмотря на падения интереса в средствах массовой информации, подспудная кропотливая работа в академических кругах продолжается, и через некоторое время (не сразу!) начинает давать практические результаты. Интерес вновь появляется, но уже не праздный, а деловой и обоснованный.

В марте 1947 г. Уоррен Уивер, директор отделения естественных наук Рокфеллеровского фонда, в переписке с Эдью Бутом (Andrew D. Booth) и Норбертом Винером впервые сформулировал концепцию машинного перевода, которую несколько позже (в 1949 г.) развил в своем меморандуме, адресованном Фонду.

В 1952 г. состоялась первая конференция по машинному переводу (МП) в Массачусетском технологическом институте, а в 1954 г в Нью-Йорке была представлена первая система МП — IBM Mark II, разработанная компанией IBM совместно с Джорджтаунским университетом (это событие вошло в историю как «Джорджтаунский эксперимент»). Была представлена очень ограниченная в своих возможностях программа (она имела словарь в 250 единиц и 6 грамматических правил), осуществлявшая перевод с русского языка на английский. После этого подобные разработки получили широкое распространение в Италии, Франции, Англии, Японии, а также был создан международный специализированный журнал.

В 1959 г. философ Й. Бар-Хиллел (Yohoshua Bar-Hillel) выступил с утверждением, что высококачественный полностью автоматический МП (FANQMT) не может быть достигнут в принципе. Это выступление самым неблагоприятным образом отразилось на развитии МП в США.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

В 1966 был отмечен мощный спад, связанный с отчетом специально созданной Национальной Академией наук США комиссией ALPAC (Automatic Language Processing Advisory Committee). Она пришла к заключению, что машинный перевод нерентабелен: соотношение стоимости и качества машинного перевода было явно не в пользу последнего, а для нужд перевода технических и научных текстов было достаточно человеческих ресурсов. Поэтому Министерство обороны решило прекратить финансирование такого рода проектов. Тема перестала быть модной, и работу продолжили только квалифицированные энтузиасты и профессионалы.

Следующие десять лет разработка систем МП осуществлялась в США университетом Brigham Young University.

С развитием вычислительной техники в конце 70-х годов (появление микрокомпьютеров, развитие сетей, увеличение ресурсов памяти) машинный перевод вошел в эпоху "Ренессанса". При этом несколько сместились акценты: исследователи теперь ставили целью развитие "реалистических" систем МП, предполагавших участие человека на различных стадиях процесса перевода. В результате в 80-х годах на рынке появились первые промышленные системы машинного перевода, ориентированные на массовое использование.

В 1990 г., когда системы машинного перевода снова стали одним из приоритетных направлений развития компьютерной отрасли и вышли на новый качественный уровень, пройдя непростой этап переосмысления и взаимной интеграции, Ларри Чайлдс предложил их классификацию. Он разделил все «электронные переводчики» на три группы:

- **FAMT (Fully-automated machine translation)** — инструменты полностью автоматизированного машинного перевода.

входной текст ® ввод в компьютер ® анализ / синтез текста ® выходной текст

- **HAMT (Human-assisted machine translation)** — приложения для автоматизированного машинного перевода текстов, выполняемого при участии человека.

входной текст ® редактор ® анализ / синтез промежуточного текста ® редактор ® выходной текст

- **MANТ (Machine-Assisted Human Translation)** — вспомогательные средства для выполнения перевода человеком с использованием компьютера.

входной текст ® переводчик + компьютер ® сырой перевод ® редактор ® выходной (переведенный) текст


Но жизненная практика свидетельствует о том, что в функционировании всех существующих на сегодняшний день систем МП в большей или меньшей степени необходимо участие человека-редактора.

Современные системы машинного перевода имеют развитую архитектуру, значительный объем, используют огромные информационные ресурсы и весьма изощренные алгоритмы.

Впрочем, мечты, с которыми род людской взялся полвека назад за задачу машинного перевода, в значительной мере остаются мечтами — высококачественный («художественный») перевод текстов широкой тематики по-прежнему недостижим. Однако несомненным является ускорение работы переводчика при использовании систем машинного перевода: по современным оценкам на 50-70%.

По мере появления все новых и новых алгоритмов, методик и программных решений, нацеленных на компьютерный перевод текстовой информации, достоверными фактами является наличие у таких систем целого ряда преимуществ перед обычным «рукотворным» переводом:

- высокая скорость перевода (использование МП позволяет значительно сократить время, требуемое для перевода больших объемов текста, по современным оценкам на 50-70%);
- снижение его стоимости (с помощью МП можно сэкономить на оплате услуг профессиональных переводчиков);
- конфиденциальность переводимой информации (участие компьютера в обработке конфиденциальных документов в гораздо большей степени гарантирует нераспространение их содержания, чем участие переводчика);

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- универсализацию процесса перевода (даже самый квалифицированный переводчик, как правило, специализируется в определенных областях знаний и тематических направлениях, тогда как системы МП достаточно универсальны, и их работа зависит только от проведения необходимых настроек — подключения нужных электронных словарей и т. п.).

В настоящее время существует множество коммерческих проектов машинного перевода: Systran, IBM, L&N (Lernout & Hauspie), Language Engineering Corporation, Atril, Trados, Caterpillar Co., LingoWare. В России большой вклад в развитие машинного перевода внесла группа под руководством проф. Р. Г. Пиотровского (Российский государственный педагогический университет им. Герцена, Санкт-Петербург). Сейчас заслуженное признание получил автоматический переводчик, разработанный компанией ПРОМТ (PROgrammer's Machine Translation), который удовлетворяет ожиданиям большинства пользователей.

Интеллектуальные базы данных

На заре использования компьютеров форматы хранения данных и средства для манипулирования ими изобретались программистами индивидуально для каждого случая. Это неудобно и неэффективно, и вскоре были предложены унифицированные способы хранения данных (модели данных) и разработаны унифицированные системы доступа к данным (системы управления базами данных – СУБД). Рассматривая концепцию базы данных (БД) с самой общей точки зрения, можно отметить, что БД поддерживает три основных группы операций: занесение данных в базу, поиск данных в базе и извлечение данных из базы, причем извлекается те данные, которые до этого были занесены⁶. Были предложены и используются несколько идей, к которым это ограничение неприменимо. Одной из них является идея интеллектуальной базы данных.

Интеллектуальная базы данных (Intelligent Database) предоставляет эффективный способ хранения, поиска и извлечения большего числа фактов, чем те, которые были изначально загружены в базу.

Рассмотрим простой случай. Реляционная база данных (см. База данных, снабженная такими правилами, является интеллектуальной.): таблица, строки которой соответствуют объектам, а столбцы – каким-то атрибутам объектов. Все объекты однотипны, и различаются только значениями атрибутов. Такая вещь позволяет отвечать на вопросы фактографического характера. Например, можно спросить: «В каком подразделении работает Новиков?», и получить ответ «На кафедре «Прикладная математика»». Но если добавить туда информацию, представленную не как таблица, а как правило, можно будет отвечать на вопросы, ответы на которые не содержатся в базе данных. Идея состоит в том, что кроме обычных реляционных таблиц в базе данных хранятся и некоторые правила, например, такое:


if (x.Должность = “Заведующий” &
(x.Подразделение = y.Подразделение)
then x начальник y

База данных, снабженная такими правилами, является *интеллектуальной*.

Таблица базы данных.

ФИО	Должность	Подразделение
Новиков	Доцент	Кафедра «Прикладная математика»
Клавдиев	Заведующий	Кафедра «Прикладная математика»

Действительно, если поставить вопрос «Кто начальник Новикова?», то несложный встроенный механизм логического вывода легко найдет ответ: «Клавдиев». Приведенный пример позволяет выяснить отношения на уровне начальник-подчиненный. В данном случае, очевидно, что Клавдиев является начальником не только Новикова, но и любого другого сотрудника кафедры «Прикладная математика». Таким образом, интеллектуальная база данных может в качестве ответа выдать факт, который в ней не хранится. Более того, такие базы позволяют отвечать на вопросы, требующие дедуктивного рассуждения, состоящего не из одного применения правила, а из многих применений, так что извлекаемые факты могут быть не так очевидны и весьма далеки от первоначально введенных.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

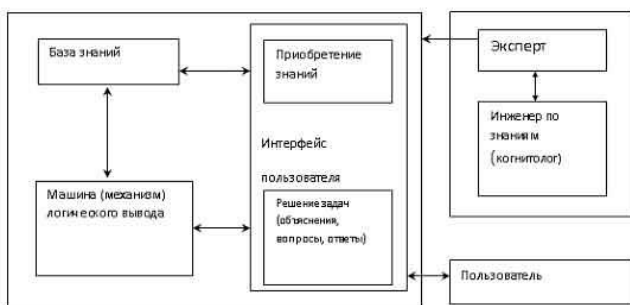
Экспертные системы

Методы интеллектуальных баз данных получили дальнейшее развитие при разработке автоматических консультирующих систем. Такие экспертные системы (Expert Systems) способны предоставить компетентный ответ пользователю на конкретный вопрос. Первые экспертные системы были созданы для применения, например, в медицине (диагностика заболеваний) и в геологии (оценка месторождений).

Одна из самых главных проблем в этой области заключается в том, как следует представлять и использовать знания экспертов, которые по своей сути являются в достаточной степени разнородными и противоречивыми данными.

Экспертные системы — системы искусственного интеллекта, включающие в себя знания об определенной слабо структурированной и трудно формализуемой узкой предметной области, и способные предлагать и объяснять пользователю разумные решения.

В настоящее время считается общепринятым, что структура экспертных систем (ЭС) состоит из пяти модулей. Совокупность трех модулей — Правила, Модель ситуации и Вывод — по сути, является интеллектуальной базой данных. Модули, позволяющие извлекать знания и объяснять выводы, превращают интеллектуальную базу данных в экспертную систему.



Архитектура экспертной системы

Сейчас чаще всего база знаний для ЭС — это набор правил. Простейший случай, когда правила имеют детерминированный характер и позволяют из имеющихся фактов выводить новые факты, как в интеллектуальной базе данных. Такой вариант детально рассматривается во второй лекции про представление знаний системами продукций.

Блок объяснений вывода фактически представляет машинный вывод результата в человеко-читаемой форме, так что пользователь может проследить, какие шаги рассуждений применяла экспертная система при решении задачи. Тем самым доверие пользователя к ответу возрастает.


Наиболее важным и сложным является блок извлечения знаний. Результат работы этого блока понятен — это пополнение и изменение базы знаний. А вот исходные данные могут быть различными. Не следует надеяться на то, что эксперт в предметной области просто так "из головы" сформулирует эффективные правила рассуждений, причем сразу в том формате, который требуется в системе. Это маловероятное чудо. Одним из наиболее надежных считается метод постепенного пополнения и изменения базы знаний, причем поводом к изменению является каждый случай, когда система не смогла решить задачу или решила ее неправильно. В таком случае эксперт анализирует с помощью блока объяснения ход машинных рассуждений, определяет, какое правило было неудачным или какого правила не хватает, и добавляет его в базу знаний посредством блока извлечения знаний. Можно сказать, что экспертная система, подобно человеку, учится на ошибках. Довольно часто встречается ситуация, когда правила имеют не детерминированный, а вероятностный характер:

если ..., то ... с вероятностью ...

Большинство выводов, сделанных ЭС по вероятностным правилам, имеют вероятностный характер. На свой вопрос пользователь получает несколько различных ответов, каждому из которых приписана своя вероятность правильности.

Автоматическое доказательство теорем

Доказательство теорем безусловно предполагает затрату интеллектуальных усилий, требующих не только дедуктивных навыков, но и в значительной степени интуитивных догадок (например, какие

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

леммы необходимы для той или иной теоремы). Другими словами, доказательство теорем подразумевает, в том числе, поиск уже доказанных теорем и решенных подзадач, которые могут оказаться полезны для достижения конечной цели. К настоящему времени разработано несколько программ с элементами ИИ, способных работать в этом направлении, то есть программ *автоматического доказательства теорем* (Automatic Theorem Proving).

При первом взгляде на предмет мы сразу отмечаем обстоятельства, известные из курса математической логики.

- Язык исчисления предикатов первого порядка, обогащенный такими вещами, как равенство и внелогические предикаты и функции, является универсальным средством для аксиоматизации любого разумного случая (то есть любое утверждение может быть сформулировано как формула прикладного исчисления предикатов, см [Новиков Ф.А. Дискретная математика для программистов. – СПб, Питер, 2009]).
- Алгоритм Тарского перечисления правильно построенных формул позволяет генерировать гипотезы исчерпывающим образом (то есть все правильно построенные формулы могут быть перечислены).
- Метод резолюций является полным (то есть вывод любой доказуемой формулы может быть автоматически найден).


Возникает следующая идея, подкупающая своей незатейливостью и простотой. Запишем аксиомы предметной области на языке предикатов первого порядка. Запустим алгоритм типа алгоритма Тарского, который будет генерировать новые гипотезы одну за другой. Каждую гипотезу будем пытаться автоматически доказывать методом резолюций, и если получится, то компьютер сам будет добавлять все новые и новые знания. Процесс можно запустить на 24 часа в сутки, 7 дней в неделю, 365 дней в году, и не на одном компьютере, а на всех незанятых в данный момент. Кстати, сейчас в среднем 90% времени компьютеры простаивают,

Таким образом, автоматическое доказательство теорем (АДТ) возможно, и его нетрудно запрограммировать, но такой прямолинейный подход оказывается безнадежно неэффективным. Сложные, интересные теоремы автоматически доказать не удастся, потому что пространство перебора столь велико, что даже самые быстродействующие компьютеры не в состоянии ничего сделать за разумное время.

Однако существуют примеры, когда в очень узком классе математических теорий, применяя очень специфические методы, удавалось автоматически находить и доказывать нетривиальные содержательные теоремы, которые были ранее неизвестны.

Этому примеру уже 10 лет. Член-корреспондент РАН В.М. Матросов придумал удивительный способ перечисления и доказательства математических фактов в очень узкой области, относящейся к качественной теории дифференциальных уравнений Основоположник метода векторных функций Ляпунова в теории устойчивости. Коллеги-программисты составили программу, которая генерировала гипотезы и автоматически строила к ним доказательства. Таким образом были доказаны некоторые действительно новые, нетривиальные теоремы. Опыт подтверждается следующим: не внося никаких изменений в машинно-сгенерированные гипотезы и тексты, Матросов отправлял их в самый престижный в этой области американский журнал, который публиковал эти данные, как новые научные результаты, полученные «В.М. Матросовом и соавторами». Мировое сообщество пока не знает, как отнестись к этому факту, однако прецедент есть.

Следует подчеркнуть, что АДТ является основным инструментом всех ПСИИ, которые в той или иной степени используют формулы логических исчислений для представления знаний. Формулы исчисления предикатов используются практически всегда – это основной способ формальной записи любых утверждений, известный человечеству в настоящее время. Поэтому можно смело утверждать, что автоматическое доказательство теорем – это фундаментальная тема ИИ. В следующих лекциях показано, что все рассматриваемые нами методы ИИ прямо или косвенно сводятся к автоматическому доказательству теорем.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


Автоматическое управление роботом

Роботы чаще всего используются при выполнении работ в опасных для жизни условиях или при относительной недоступности объекта. Все наслышаны об успехах беспилотных луно- и марсоходов, военных роботов, обеспечивающих разведку и обезвреживание взрывоопасных предметов. Недавно в Японии прошла международная выставка роботов, на которой большинством моделей были представлены бытовые и медицинские роботы. К примеру, вниманию публики был представлен «робот-пациент» для обучения дантистов, который способен реагировать на боль. Другой образец — «робот-полицейский», способный запоминать лица людей, которых разыскивает полиция. При установлении личности робот немедленно предупреждает об этом по радиосвязи сотрудников охранной службы. Все большую популярность приобретают роботы, созданные для развлечения. На той же выставке был представлен «робот-партнерша по танцам». Это «женщина» в человеческий рост, которая при помощи специальных датчиков чутко улавливает, в какую сторону движется танцор, и перемещается в такт с ним. Она может исполнить и короткий сольный танец, плавно перемещаясь и двигая руками, головой и корпусом. Начинает казаться, что мечты о будущем роботов, описанные фантастами XX века, начинают сбываться.

На самом деле, *управление роботом* (Robot Planning) — задача чрезвычайно сложная. Казалось бы, нет ничего проще, чем передвигаться на двух ногах. Развитие любого человеческого существа начинается со способности видеть и ходить. При изучении временной шкалы появляется мысль, что эти навыки вроде бы и не требуют значительных интеллектуальных усилий. На первый взгляд может показаться, что говорить, рассуждать и программировать значительно сложнее. Однако если взглянуть на развитие роботов и робототехники, можно легко убедиться в обратном. Роботы сначала научились программировать и рассуждать, потом говорить, затем ходить, и до сих пор не научились видеть как следует.

Успешное хождение двуногого робота — достаточно сложная задача даже для современного уровня развития науки и техники. Дело в том, что когда мы ходим, центр тяжести нашего тела почти никогда не находится над площадью опоры. С точки зрения механики, мы не ходим, а непрерывно падаем, но не совсем, а удачно подставляем ногу в нужное место и продолжаем движение. Лобовой подход в стиле XIX века — составить механическую модель с нужным количеством степеней свободы и на каждом шаге решать соответствующую систему дифференциальных уравнений, — не работает. Даже самый быстродействующий современный компьютер не успевает проводить нужные вычисления в реальном времени. Совершенно ясно, что мы, когда ходим, не решаем в уме систему уравнений. Также и для реализации в хождения роботов применяются совсем другие приемы. Каждое новое достижение в этой области рассматривается как значительный успех, что видно из перечисленного выше состава экспонатов международной выставки. Что касается быстрого бега и прыжков (на уровне профессиональных спортсменов), то двуногий быстро бегающий и прыгающий робот — все еще почти нереальная задача на сегодняшний день⁷.

До сих пор не создано адекватного аппарата, позволяющего роботу «видеть» окружающий мир и реагировать на «визуальные» изменения в нем также хорошо, как это делает человек. То, как видим мир мы, — результат длительного обучения, основанного на интеллектуальной активности. Известно, что уже на третьей неделе развития ребенок обладает цветным стереоскопическим зрением. В результате взаимодействия с предметами и накопления опыта наблюдений у ребенка на основе цветного стереоскопического зрения формируется новая способность — предметное восприятие. Важно подчеркнуть, что эта способность объясняется не физиологическими, а интеллектуальными причинами⁸. С другой стороны, исследователи установили, что если поместить человека в совершенно непривычную искусственную среду (в которой ненормально устроено освещение, привычные предметы имеют непривычные размеры, процессы имеют ненормальный темп и т.д.), то способность к адекватному восприятию трехмерной действительности человеком утрачивается. Таким образом, предметное восприятие — это способность психики, а не глаза. На самом деле, мы видим модель, которую очень быстро, за десятые доли секунды, строит мозг по информации, полученной из глаза. Иногда эта модель может несколько расходиться с реальностью (см. например Error: Reference source not found и Error: Reference source not found в разделе 1.1.7). Проблема технического зрения заключается не в разрешающей способности камер и не в их

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

количестве, а в сложности алгоритмов, адекватно восстанавливающих трехмерную модель по совокупности плоских изображений, полученных от камер. Для конкретных условий и специальных задач есть неплохие технические решения, например, роботы, распознающие человеческие лица в охранных системах, или роботы, распознающие положение и ориентацию мелких деталей на конвейере часового производства, однако эти системы не взаимозаменяемы и не совместимы. Универсальной системы технического зрения, сопоставимой по широте возможностей с человеческими, пока еще не сконструировано⁹.

Распознавание образов

Распознавание образов (Pattern recognition) — раздел кибернетики, развивающий методы классификации и идентификации предметов, процессов и сигналов, которые характеризуются конечным набором свойств и признаков.

Например, задача о переходе улицы по сигналам светофора. Распознавание цвета загоревшейся лампы светофора и знание правил дорожного движения позволяет принять правильное решение о том, можно или нельзя переходить улицу в данный момент.

Одной из первых искусственных нейронных сетей, способных к перцепции (восприятию) и формированию реакции на воспринятый стимул был *персептрон* Фрэнка Розенблатта. Персептрон рассматривался его автором не как конкретное техническое вычислительное устройство, а как модель работы мозга. Нужно заметить, что после нескольких десятилетий исследований современные работы по искусственным нейронным сетям редко преследуют такую цель.

На примере распознавания образов мы опять наблюдаем характерную кривую развития направления ИИ. После появления пионерской работы Розенблатта наблюдался резкий подъем интереса к этой тематике, затем последовал холодный душ в виде книги Минского и Пейперта. Казалось, перспективы утрачены, но к настоящему времени первоначальная идея трансформировалась в нейронные сети и нейрокомпьютеры, которые в текущий момент более чем успешны. Мы рассматриваем эту тему в последней лекции.

Один из серьезных успехов в распознавании образов связан с созданием в 1982 г фирмой Logica системы распознавания отпечатков пальцев, которая нашла широкое применение не только в криминалистике, но и в разнообразных системах защиты, например в ноутбуках.


Другими примерами применения ПСИИ является распознавание речи, распознавание сканированного текста (OCR), распознавание автомобильных номеров или мониторинг среды по аэрофотоснимкам.

Необходимо хорошо понимать, что синтез объекта всегда значительно проще, чем анализ того же самого объекта. Наглядный пример — рукописный текст. Его очень легко написать, но прочитать его иногда невероятно трудно. Всем нам, надо полагать, знакома ситуация, когда в Интернет при регистрации на сайте, отправке сообщения и тому подобных операциях необходимо «расшифровать» код на рисунке. Это считается достаточно надежным тестом, отличающим реального пользователя от робота. Современные программы распознавания не могут проанализировать сложное изображение, где контуры букв и цифр не замкнуты, линии причудливо искажены и присутствует большое количество «шума», а человеку это удастся практически без усилий.

Интеллектуальные игры

Одно из самых известных и популярных применений ПСИИ — это разнообразные интеллектуальные игры (Intelligent games). Почти наверняка каждый сталкивался с искусственным интеллектом, использующимся именно в этой области. Шашки, шахматы, нарды встречаются и в мобильном телефоне, и в качестве более серьезных программ для достаточно мощных компьютеров.

Исторически случилось так, что из всех интеллектуальных игр внимание программистов-исследователей более всего привлекали шахматы. На данный момент лучшие компьютерные шахматные программы уже превосходят в игре известных гроссмейстеров мира. Можно сказать, что в шахматы компьютер переиграл человека. При этом следует подчеркнуть, что никаких чудес в шахматных программах нет. В них используются хорошо известные и тщательно реализованные алгоритмы, подобные тем, что рассматриваются в лекции 3. Тем не менее, если мы принимаем

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

утверждение, что игра в шахматы на гроссмейстерском уровне требует интеллекта, то мы вынуждены признать, что лучшие шахматные программы обладают интеллектом. При этом компьютерные шахматы — это не только развлечение, но и способ обучения новых шахматистов, создание шахматных баз данных, анализ окончаний, изменение правил¹⁰. Но самое главное, в любом случае, это один из наиболее интересных способов представления знаний.

Задача представления знаний в интеллектуальных играх является вызовом для естественного интеллекта программистов. Дело в том, что некоторые из игр чрезвычайно сложны с количественной точки зрения. Например, шахматы имеют около 10^{40} возможных позиций, и около 10^{120} возможных партий. Это огромные числа, большие, чем число атомов во Вселенной (по разным оценкам от 10^{67} до 10^{80}) и чем число секунд, прошедших от начала существования Вселенной¹¹. Никакие информационные устройства не могут и никогда не смогут просмотреть или запомнить все позиции и разыграть все партии – это физически невозможно. Так что у разработчиков программ для интеллектуальных игр впереди непочтатый край работы!

Отечественное шахматное программирование

Огромный вклад в алгоритмическое понимание шахмат внес М. М. Ботвинник. Успешно совмещая шахматную деятельность с научной работой в области электротехники (он успешно защитил докторскую диссертацию), Ботвинник является автором ряда изобретений, запатентованных во многих странах. Закончив в 1970 спортивные выступления («матч века» сборная СССР — сборная мира), Ботвинник занимался проблемами искусственного интеллекта, работал над компьютерной шахматной программой «Пионер». Идея создания "электронного шахматиста" возникла у него еще в 1958 году. В 1968 году вышла книга Ботвинника, в которой он сформулировал метод моделирования мышления шахматного мастера при выборе хода, построенного на позиционной оценке, отмечающей все побочные и ненужные ходы и варианты. К сожалению, программа «Пионер» так и не была реализована, но ценные идеи и решения Ботвинника остались в теории шахмат и нашли свое применение в позднейших разработках отечественных специалистов по ИИ. Уже несколько десятилетий проводятся регулярные чемпионаты мира среди шахматных программ. В настоящее время Чемпионат проводится Международной ассоциацией компьютерных игр и открыт для всех типов компьютеров, у которых не более восьми ядер (то есть исключая суперкомпьютеры и большие кластеры). Отрадно отметить, что на первых чемпионатах побеждала отечественная программа «Каисса», разработанная под руководством Г.М. Адельсона-Вельского и М.В. Донского. Чемпион последних трех лет — программа «Рыбка» («Rybka»), разработанная чешско-американским шахматистом Васиком Райлихом (Vasik Rajlich).

Шахматное программирование достигло очень высокого уровня. Если провести турнир между обычной студенческой группой и обычной шахматной программой, например, ChessMaster, то окажется, что в среднем группа играет хуже, чем ChessMaster. О чем это говорит?


Место представления знаний в искусственном интеллекте

В интеллектуальной области невозможно создать универсальный алгоритм, подходящий для решения любой задачи. Этим интеллектуальные области отличаются от неинтеллектуальных, в которых существуют алгоритмы "делай раз, делай два, делай три...". Метод решения задачи в интеллектуальной области — это всегда поиск новых подходов, новых путей, в конце концов, если ничего не удастся придумать — банальный перебор всех возможностей в поисках решения.

Итеративный характер решения задач

Общая схема решения проблемы с помощью компьютера изображена на Рис. Стрелочки сверху вниз — детерминированный процесс, стрелочки снизу вверх — недетерминированный. Решение проблемы всегда носит, в целом, недетерминированный характер. Это метод проб и ошибок, метод поиска, в результате которого часто приходится возвращаться на предыдущие шаги.

Если обратная связь в таком процессе (стрелочки снизу вверх) реализуется через человека (а это происходит достаточно часто), то такой процесс не может называться обработкой знаний. Для того чтобы можно было говорить именно о компьютерной обработке знаний, каждая петля обратной связи должна замыкаться, соответственно, через компьютер. И тогда данные, с которыми работает такая программа, вправе называться знаниями.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

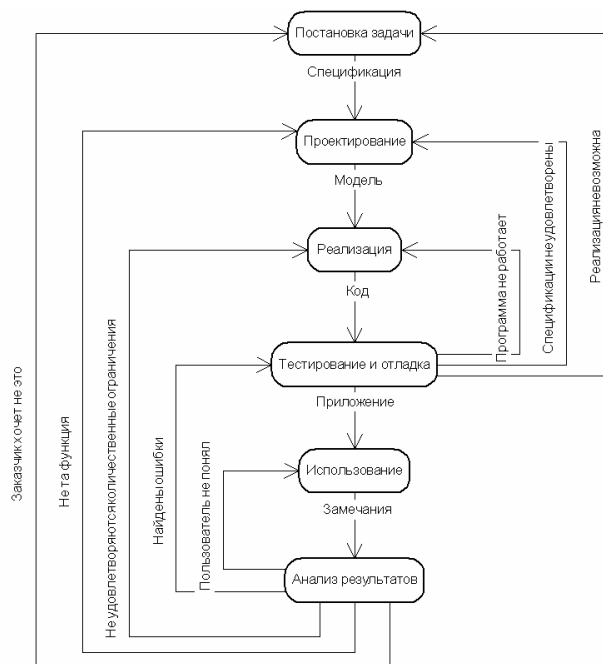


Схема процесса решения задачи на компьютере

При традиционном процедурном программировании (сверху вниз), решение целиком зависит от задачи. Каждая задача имеет свое решение, и решение одной задачи не может быть непосредственно, без дополнительной работы, использовано для решения другой. В ИИ (непроцедурное программирование) один и тот же метод поиска может применяться к целому набору проблем.

Наше резюме таково. Если задача не имеет прямого алгоритма решения, решается перебором, и возвраты осуществляются не через голову программиста, а через саму программу, то это прикладная система с элементами искусственного интеллекта. Если же задача имеет прямой алгоритм решения, который находит программист (может быть, за несколько попыток), но не сама программа, то искусственный интеллект отсутствует (возможно, присутствует естественный интеллект программиста). Но, фактически, непроходимой границы между этими случаями нет.

Знание и незнание

Явно выраженную границу между знанием и незнанием указать невозможно. На самом деле точной границы для представления знаний в компьютере нет — всякая программа и всякие данные представляют какие-то знания. С другой стороны, самые тонкие философские соображения можно (на естественном языке) записать в компьютер, и они будут там представлены бессмысленной последовательностью байтов. Байты присутствовать в компьютере будут, но философские знания?... Точнее говоря, ничто в компьютере не есть знания без нашей собственной интерпретации, и почти из всего можно извлечь знание, если как следует постараться. Рассмотрим это на модельном примере.


Допустим, мы хотим, чтобы компьютер отсортировал набор чисел в порядке их возрастания. Мы, люди, несомненно, знаем, как это сделать. Рассмотрим, можем ли мы научить это делать компьютер, то есть, можем ли мы представить необходимые знания в компьютере? Пусть нам заранее известно, что этих чисел три и они равны 3, 5 и 7. Можно научить компьютер решать данную задачу, написав всего один оператор:

Write (3, 5, 7)

Таким образом, мы представили в компьютер знания, достаточные для решения этой отдельной частной задачи. Знания представлены в форме одного оператора языка программирования.

Усложним задачу. Пусть чисел по-прежнему три, но при этом значение их произвольное. Составим процедуру, способную отсортировать эти числа по возрастанию:

proc Sort3 (x, y, z)

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

```

if x > y then x := y end if
      if y > z then y := z end if
if x > y then x := y end if
Write (x, y, z)
end proc

```

Таким образом, мы научили компьютер решать довольно большое множество задач, связанных с сортировкой трех чисел по возрастанию, не слишком удлинив при этом запись. Во всяком случае, данная запись намного меньше, чем все возможные операторы Write со всеми возможными комбинациями аргументов¹². Знания представлены в форме процедуры с параметрами. Другими словами, механизм процедур в языках программирования — это уже метод представления знаний. Просто это достаточно старый метод, мы к нему привыкли, и не воспринимаем как механизм представления знаний.

Возникает еще один вопрос: откуда взялась процедура Sort3?

Можно сказать, что, как обычно, ее придумал программист. Но мы дадим не совсем привычный ответ, показав, как можно представить в программе знания, достаточные для того, чтобы компьютер сам придумал процедуру, подобную Sort3.

Так как в процедуре Sort3 есть фрагменты кода, похожие друг на друга, и их больше двух, следует применить специальный *рефакторинг* (преобразование текста программы, не меняющее выполняемую программой функцию), который называется *запроцедуривание*. *Запроцедуривание* — это такой рефакторинг, при котором повторяющиеся или похожие фрагменты программы заменяются вызовом вновь введенной процедуры. Переписав процедуру Sort3, введя процедуру Sort2, получаем процедуру Sort3'.

```

proc Sort3' (x, y, z)
    Sort2(x, y)
Sort2(y, z)
Sort2(x, y)
Write (x, y, z)
end proc
proc Sort2(u, v)
if u > v then u := v end if
end proc

```

Но процедуру Sort2 нетрудно специфицировать формально:

x, y { Sort2(x, y) } x < y

Здесь перед заголовком процедуры (который заключен в фигурные скобки) написано предусловие, а после заголовка — постусловие. Процедура Sort2 очень проста и очевидно полезна во многих случаях. Можно допустить, что эта процедура вместе со спецификацией (предусловие и постусловие) содержится в библиотеке готовых процедур, доступной транслятору.

Предусловие — формальное утверждение о входных параметрах процедуры, которое должно быть выполнено до начала вызова процедуры для того, чтобы процедура работала правильно.
Постусловие — формальное утверждение о входных и выходных параметрах процедуры, выполнение которого гарантируется после окончания вызова процедуры в случае, если было выполнено предусловие.


Тогда, используя эту спецификацию, транслятор может логически вывести процедуру Sort3':

```

{x, y, z}
Sort2(x, y) {x < y, z}
Sort2(y, z) {y < z, x < z}
Sort2(x, y) {x < y < z}.

```

Таким образом, мы немного усложнили механизм транслятора (допустив возможность логического вывода при трансляции) и получили систему автоматического синтеза программ. Другими словами, мы компактно записали не одну процедуру Sort3, а целый класс процедур. В этом случае знания представлены в виде специфицированной процедуры и алгоритма логического вывода, встроенного в транслятор.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Кажется, что понятия «знание» и «процедура» ортогональны. Если есть процедура, никакие знания не нужны — делай, как сказано в процедуре, и задача будет решена. Приведенный пример показывает, что понятия «знание» и «процедура» (или, лучше сказать, алгоритм) не только не противоположны, но даже, можно сказать, неразделимы. Любой алгоритм обязательно содержит знания, а любые знания представляются, в конечном счете, алгоритмами.

Алгоритмы поиска решения и представление знаний

Первый этап развития ИИ был связан с разработкой методов решения задач — алгоритмов поиска решения и построения логического вывода. В настоящее время это уже достаточно проработанный и изученный вопрос. Основные классические результаты изложены в третьей лекции.

УНИВЕРСАЛЬНЫЙ РЕШАТЕЛЬ ЗАДАЧ

Первые исследования, относимые к искусственному интеллекту были предприняты почти сразу же после появления первых вычислительных машин.

В 1954 году американский исследователь А. Ньюэлл (A. Newel) решил написать программу для игры в шахматы. К работе была привлечена группа голландских психологов под руководством А. Де Гроота (A. De Groot), изучавших стили игры выдающихся шахматистов. Через два года совместной работы этим коллективом был создан язык программирования IPL (Information Processing Language) – по-видимому, первый символьный язык обработки списков. Вскоре была написана и первая программа, которую можно отнести к достижениям в области искусственного интеллекта. Эта была программа "Логик-Теоретик" (1956 г.), предназначенная для автоматического доказательства теорем в исчислении высказываний.


Собственно же программа для игры в шахматы, NSS, была завершена в 1957 г. В основе ее работы лежали так называемые эвристики (правила, которые позволяют сделать выбор при отсутствии точных теоретических оснований) и описания целей. Управляющий алгоритм пытался уменьшить различия между оценками текущей ситуации и оценками цели или одной из подцелей. Шахматная сила программы была невелика.

В 1960 г. той же группой, на основе принципов, использованных в NSS, была написана программа, которую ее создатели назвали GPS (General Problem Solver) – *универсальный решатель задач*. Программа GPS могла справляться с рядом головоломок, вычислять неопределенные интегралы, решать некоторые другие задачи.

Название «Универсальный решатель задач» обусловлено тем фактом, что это была едва ли не первая система для решения задач (problem-solving system), в которой общие методы решения были отделены от знаний, определяющих конкретную задачу. Часть программы, осуществлявшая поиск решения, не обладала информацией о конкретной задаче, с которой она работала. Специфичные для конкретной задачи знания организовывались в отдельные структуры: объекты и операторы преобразования объектов. Задача для системы GPS ставилась в виде пары, состоящей из начального объекта и целевого объекта, в который начальный объект должен быть преобразован.

Методология поиска, использованная в GPS, отличалась от предшествующих методов поиска в пространстве состояний тем, что она динамически выбирала путь, по которому продолжить поиск. Т.е. система пыталась искать решение в первую очередь в «наиболее перспективных» ветвях поиска. Такая методология была названа «анализ средств и целей» (means-ends analysis). Ее суть заключалась в том, что сначала отыскивалось различие между текущим объектом и объектом, который мы хотим получить. Это различие относилось к одному из ряда классов различий. С каждым классом был сопоставлен набор операторов, способных уменьшить различие между текущим и целевым объектами.

На каждом шаге поиска программа GPS определяла различие объектов и выбирала один из релевантных операторов, который и пыталась затем применить к текущему объекту. Поиск подходящей последовательности операторов выполнялся в глубину до тех пор, пока операторы оказывались применимы, а ветвь поиска «выглядела перспективной». Если ветвь поиска была бесперспективной, то выполнялся возврат. Важной особенностью анализа средств и целей является то, что всегда выбирается релевантный оператор, уменьшающий различие объектов, даже если он и не применим к текущему объекту. Вместо того чтобы отказаться от неприменимого к текущему

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

объекту оператора, программа GPS пыталась преобразовать текущий объект в объект, пригодный для применения выбранного оператора.

Наилучшим алгоритмом для поиска оптимальных путей в различных пространствах является A* (читается как "А-звездочка"). Этот алгоритм был впервые описан в 1968 году Питером Хартом, Нильсом Нильсоном и Бертрамом Рафаэлем. В их работе он упоминается как «алгоритм А». Этот эвристический поиск сортирует все узлы по приближению наилучшего маршрута, идущего через этот узел. Типичная формула эвристики выражается в виде:

$$f(n) = g(n) + h(n)$$

где:

$f(n)$ значение оценки, назначенное узлу n

$g(n)$ наименьшая стоимость прибытия в узел n из точки старта

$h(n)$ эвристическое приближение стоимости пути к цели от узла n


Таким образом, этот алгоритм сочетает в себе учет длины предыдущего пути из алгоритма Дейкстры с эвристикой из алгоритма "лучший-первый". Алгоритм хорошо отражен в листинге 3. Так как некоторые узлы могут обрабатываться повторно (для поиска оптимальных путей к ним позднее) необходимо ввести новый список Closed для их отслеживания.

A* имеет множество интересных свойств. Он гарантированно находит кратчайший путь, до тех пор пока эвристическое приближение $h(n)$ является допустимым, то есть он никогда не превышает действительного оставшегося расстояния до цели. Этот алгоритм наилучшим образом использует эвристику: ни один другой алгоритм не раскроет меньшее число узлов, не учитывая узлов с одинаковой стоимостью.

Однако для нетривиальных случаев трудоемкость поиска решения очень велика. Трудоемкость поиска решения — это быстрорастущая функция (в большинстве случаев растущая как экспонента или быстрее), аргументом которой является общий объем базы знаний. Рассматривая ниже конкретные алгоритмы, мы *доказываем* точные оценки, но здесь, на неформальном уровне, хотим *убедить* читателя в справедливости этого замечания. Пусть искомое решение является комбинацией конечного числа заранее заданных элементов, которых n . Тогда всего существует $2^n - 1$ непустых комбинаций¹³, если брать элементы без повторений, а если с повторениями, то еще больше. Заранее мы не знаем, какая комбинация является решением, в этом и состоит поисковый характер алгоритма, а значит, *в худшем случае* нам придется перебрать все комбинации!

Один из способов уменьшения трудоемкости поиска решения является изменение алгоритмов поиска с целью повышения их внутренней эффективности, не зависящей от решаемой задачи. Однако для очень многих задач, типичных для ИИ, доказана их NP-полнота¹⁴. Построение эффективного (более эффективного, чем экспонента) алгоритма для них совершенно невероятно. Если для какой-то еще не исследованной задачи ИИ вдруг обнаруживается эффективный (скажем, полиномиальный) алгоритм, то задачу перестают квалифицировать как задачу ИИ, и переносят в область обычного программирования. Таким образом, в настоящий момент существенного улучшения в повышении внутренней эффективности алгоритмов поиска ожидать трудно.

Гораздо более продуктивным методом, позволяющим уменьшить трудоемкость поиска решения, является повышение эффективности представление знаний. Если выбор изоциренного представления позволяет уменьшить объем базы знаний, то тот же стандартный алгоритм поиска становится значительно эффективнее в том смысле, что расширяются границы его применимости. Поясним это модельным числовым примером, сравнивающим два гипотетических способа представления знаний. Пусть одна и та же задача в первом представлении имеет размер n , а во втором — $n/2$. Пусть переборный алгоритм один и тот же, и имеет трудоёмкость 2^k , где k — размер задачи. Пусть, при этом, более компактное представление дается не даром: каждый шаг перебора выполняется в десять раз дольше и еще 100 единиц времени безвозвратно теряется на предварительную обработку. Фактически, мы сравниваем две функции: 2^n и $100 + 10 \cdot 2^{n/2}$. При малых n первая функция меньше, но после $n=8$ вторая функция становится меньше, и ее преимущество только возрастает с ростом n . Даже небольшой выигрыш в значении аргумента показательной функции перевешивает все другие проигрыши!

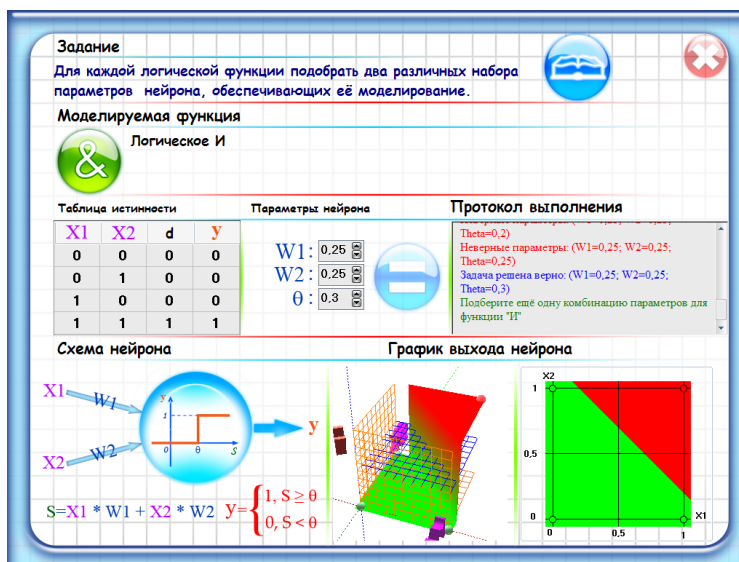
Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Компактность базы знаний — определяющий фактор эффективности системы и, следовательно, границ применимости ПСИИ. Именно это обстоятельство определяет структуру данного курса. Курс структурирован по методам представления знаний, а не по методам поиска решения. При решении любой задачи в интеллектуальной области сначала следует выбрать представление знаний, потом подобрать подходящий алгоритм поиска решения. Дело в том, что если выбрано неудачное представление знаний, то никакие ухищрения с алгоритмами не помогут. Если же изобретено особо удачное представление знаний, то, скорее всего, самый обычный алгоритм поиска решения даст вполне удовлетворительные результаты.

Задание на самостоятельную работу

1. Используя нейросимулятор выполнить следующее задание. путем подбора синаптических весов и порога чувствительности математического нейрона заставить его моделировать логические функции: «И», «ИЛИ» и др.

В распоряжении обучающихся имеется теоретический материал (открывается путем нажатия на кнопку в виде развернутой книги), таблицы истинности логических функций (слева по центру), схема математического нейрона с формулами его работы (слева внизу), графическое изображение работы нейрона (снизу по центру и справа).



Задание
Для каждой логической функции подобрать два различных набора параметров нейрона, обеспечивающих её моделирование.

Моделируемая функция
Логическое И

X1	X2	d	y
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

Параметры нейрона
 $W1: 0.25$
 $W2: 0.25$
 $\theta: 0.3$

Протокол выполнения
 $\theta=0.2$
 Неверные параметры: ($W1=0.25$, $W2=0.25$, $\theta=0.25$)
 Задача решена верно: ($W1=0.25$, $W2=0.25$, $\theta=0.3$)
 Подберите ещё одну комбинацию параметров для функции "И"


Схема нейрона
 $S = X1 * W1 + X2 * W2$
 $y = \begin{cases} 1, S \geq \theta \\ 0, S < \theta \end{cases}$

График выхода нейрона
 A 2D plot showing the output y as a function of inputs X1 and X2. The plot is a square with a diagonal line from (0,1) to (1,0). The region above and to the left of this line is green (y=1), and the region below and to the right is red (y=0).

2. Ответьте на следующие вопросы:
 1. Дайте определение искусственного интеллекта.
 2. Опишите особенности каждой волны искусственного интеллекта
 3. Какие ожидания от ИИ существуют на сегодняшний день?
 4. В каком состоянии находятся работы по исследованию ИИ в России на сегодняшний день?
 5. Перечислите основные направления работ в области ИИ.
 6. Перечислите основные угрозы и проблемы развития систем ИИ, существующие в настоящее время.

Задание для самостоятельной работы:

- Проработать лекционный материал, основную и дополнительную литературу с целью подготовки ответов на представленный перечень вопросов для устного опроса.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Тема 2. Системы, основанные на знаниях

Практическое занятие. Создание базы знаний о предметной области

Вопросы для рассмотрения на семинарских занятиях. *Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.*

Форма проведения занятия – практическое занятие.

Вопросы для самоконтроля и текущего контроля

1. Системы основанных на знаниях.
2. Понятие инженерии знаний.
3. Этапы получения знаний о предметной области.
4. Методы получения знаний о предметной области

Понятия интеллектуальной и экспертной систем

Согласно *принципу обратной связи*, сформулированному Норбертом Винером, любая искусственная система, претендующая на разумность (интеллектуальность), как и все живое, должна обладать способностью преследовать определенные цели и приспосабливаться, т.е. обучаться. Данный принцип является основополагающим при создании систем, целевое назначение которых во многом связано с кругом задач (проблем), алгоритмы (сценарии) решения которых заранее не известны. Это проблемы творческого (интеллектуального) плана, возникающие на различных этапах принятия решений при проектировании сложных объектов, контроле и управлении сложными системами, прогнозировании социальных, экономических и политических процессов, планировании финансовых операций и т.д.

Вышесказанное тесно связано с понятиями *интеллектуальной системы* и *экспертной системы*.

Под *интеллектуальной системой* (ИС, или IS — Intellectual System) принято понимать программно-техническую систему, способную решать задачи, традиционно считающиеся творческими, принадлежащие конкретной предметной области, знания о которой хранятся в памяти интеллектуальной системы.


Экспертная система (ЭС, или ES — Expert System) – система, объединяющая возможности компьютера со знаниями и опытом эксперта в такой форме, что система может предложить «разумный совет» или осуществить «разумное решение» поставленной задачи. При этом система способна пояснить «ход своих рассуждений» в понятной для пользователя форме. Это определение ЭС принято Комитетом группы специалистов по экспертным системам Британского компьютерного общества.

Согласно данным определениям, ЭС и ИС можно рассматривать как синонимы в определенном смысле. Они отражают лишь функциональное назначение систем, не акцентируя внимания на способах их организации и условиях применения.

Вместе с тем следует заметить, что первые *экспертные системы* (именно под этим названием) относились к классу *автономных экспертных систем*, которые были призваны оказывать консультации и использовали в основном эвристические приемы решения задач, не привлекая формальные методы моделирования, анализа и синтеза.

Позднее в различных предметных (проблемных) областях появились *средства интеллектуальной поддержки процессов принятия решений*. При реализации данных процессов использовались пакеты прикладных программ различного назначения. Такие средства могли быть организованы в виде интеллектуальной надстройки над прикладными программами, а могли и интегрироваться с последними, придавая им интеллектуальность.

Как первые (ЭС), так и вторые (средства интеллектуальной поддержки), несомненно, относятся к категории интеллектуальных систем (ИС), поэтому понятие ИС следует рассматривать как более широкое. В то же время средства интеллектуальной поддержки часто называют экспертными системами.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

ЭС обычно рождается как результат детального анализа (если такой удастся провести) и компьютерной реализации последовательности действий эксперта в определенной предметной (проблемной) области. При этом принимаются во внимание следующие возможности человека-эксперта:

- способность применять свои знания и опыт для оптимального решения задач, делая при этом достоверные выводы на основании неполных и ненадежных данных;
- способность объяснять и обосновывать свои действия;
- способность приобретать новые знания из общения с другими экспертами;
- способность заново систематизировать свои знания (выстраивать новую систему чанков);
- способность «нарушать» правила, т.е. подходить к их выполнению не формально, а с учетом условий возникшей ситуации (например, не сразу человек идет на зеленый сигнал светофора);
- способность плавно оценивать свою компетентность в конкретных ситуациях (почувствовав неуверенность, эксперт не отказывается от проведения экспертизы).

Современные экспертные системы способны имитировать первые три из перечисленных возможностей человека-эксперта. При этом они могут использовать как *глубинные*, так и *поверхностные* представления знаний.

Глубинные представления включают причинные модели, категории, абстракции и аналогии, отображающие структуру и природу процессов, протекающих в предметной области. Они дают широкие возможности для их трактовки, объясняют происходящие явления и обеспечивают возможность прогнозирования поведения объектов.

Поверхностные представления не дают явного отображения причинности, а только подразумевают ее существование. Это либо эмпирические взаимосвязи, либо свернутые формы субъективного понимания структуры и назначения конкретных знаний.

Очевидно, что без глубинных представлений невозможно построение ЭС, реализующих последние три из перечисленных возможностей человека-эксперта. Однако, поверхностные представления, как правило, оказываются дешевле в реализации, и их целесообразно использовать, когда на передний план выходят: процесс решения задачи, эмпирические ассоциации и свернутые формы понимания ситуаций. Есть предметные области (например, медицинская диагностика, геологоразведка полезных ископаемых), знания в которых в основном базируются на эмпирических ассоциациях.


Системы, основанные на знаниях

По определению Эдварда Фейгенбаума (Станфордский университет), *система, основанная на знаниях* (СОЗ, или на английском knowledge-based system), – это интеллектуальная компьютерная программа, использующая знания и процедуру вывода для решения проблем, которые настолько сложны, что требуют привлечения эксперта.

С точки зрения функционального назначения, понятия СОЗ, ИС и ЭС можно рассматривать как синонимичные. Однако в определении СОЗ явно прослеживается базовый принцип организации системы *декларативного* (не предписывающего) типа. Именно этот принцип, предполагающий четкое отделение друг от друга базы знаний и механизма вывода, максимально обеспечивает модульный принцип построения, открытость системы, возможность создания оболочек экспертных систем (empty expert systems), настраиваемых через формализм базы знаний на различные предметные области. В остальном СОЗ присущи все особенности, характерные для ИС (ЭС):

- ограниченность определенной областью экспертизы;
- качественный характер выходных результатов;
- способность рассуждать при сомнительных, неполных данных;
- способность объяснять ход своих рассуждений понятным пользователю способом;
- способность самообучаться и адаптироваться к конкретным условиям применения и многое другое.

Среди областей применения СОЗ можно выделить следующие довольно обширные области, каждая из которых объединяет в себе множество предметно- или проблемно-ориентированных областей конкретного применения.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Диагностика. Спектр задач, решаемых в данной области, направлен на установление связей между нарушениями функционирования сложной системы (живой или искусственной) и их возможными причинами. К этой области относятся диагностика неисправностей сложных технических систем, медицинская диагностика и т.п.

Планирование. Эта широкая область связана с решением проблем, направленных на достижение конкретных целей в задачах с большим числом переменных. Наряду с традиционным планированием экономики, производства и т.п. к этой области следует отнести и проектирование сложных объектов.

Прогнозирование. Задачи, характерные для данной области, связаны с предсказанием возможных результатов, событий на основе данных о текущем состоянии исследуемого *объекта* (в широком понимании этого слова). Это может быть прогнозирование (предсказание) развития ситуации на рынке ценных бумаг, предсказание извержения вулкана, прогноз погоды и т.д.

Контроль и управление. Эта область объединяет в себе проблемы принятия решений на основании анализа данных из нескольких источников.

Интерпретация. Имеется в виду получение (на основе результатов наблюдений) заключений, определяющих смысловое содержание наблюдаемых явлений, объектов, ситуаций.

Мониторинг. Это непрерывная интерпретация данных наблюдений в реальном масштабе времени с информированием о возникающих нештатных ситуациях. Например, мониторинг воздушного пространства службой управления полетами авиации, мониторинг состояния оборудования атомной электростанции и т.д.

Обучение. Применение СОЗ в данной проблемной области направлено на развитие концепции интеллектуальной обучающей компьютерной среды как системы, основанной на знаниях и нейросетевых технологиях репрезентации и интерпретации знаний. Такая среда способна адаптироваться под конкретного обучаемого, генерируя для него индивидуальный рабочий сценарий обучения.

Представленные области применения СОЗ не следует рассматривать как изолированные друг от друга. Задачи, характерные для одной проблемной области, могут возникать и в других проблемных областях. Например, задачи мониторинга возникают в области контроля и управления, задачи прогнозирования – в области планирования и т.д.

Целесообразность создания и использования СОЗ в конкретной предметной (проблемной) области определяется следующими критериями:


- отсутствие строгих алгоритмов и существование эвристических методов (приемов) решения задач;
- наличие эксперта, способного решать задачи и объяснять ход их решения;
- надежность и статичность имеющихся знаний (статичность понимается в том плане, что есть сложившийся, устоявшийся костяк знаний);
- сомнительный характер доступных данных;
- в основе решения задач лежит метод логических рассуждений.

С точки зрения интеграции с другими программными средствами СОЗ могут быть *автономными* и *неавтономными*.

Автономная СОЗ традиционно призвана оказывать консультации и использует в основном эвристические приемы решения задач, не привлекая формальные методы моделирования, анализа и синтеза.

Неавтономная СОЗ выступает как *средство интеллектуальной поддержки процессов принятия решений* в различных предметных (проблемных) областях. Она может быть организована в виде интеллектуальной надстройки над прикладными программами, а может и интегрироваться с последними, придавая им интеллектуальность.

Основой для создания как автономных, так и неавтономных экспертных систем может служить *оболочка СОЗ* (empty knowledge-based system), реализующая определенные формы представления знаний, механизмы их приобретения и интерпретации с использованием процедурного анализа и метапроцедур, лежащих в основе интеллектуальной деятельности человека: *дедукции, индукции и абдукции*. Последние можно рассматривать как три взаимно дополняющие друг друга формы

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

рассуждения (формы логического вывода), т.е. построения последовательности аргументов, приводящих к некоторому утверждению – цели рассуждения.

Дедуция – это логический вывод, базирующийся на точных знаниях общего плана и позволяющий по исходным посылкам (аксиомам) получать достоверные заключения. При этом речь может идти как о формальных рассуждениях на уровне абстрактных законов, так и об интерпретированных конкретных рассуждениях.

Индукция представляет собой механизм обобщения, реализующий построение некоторого общего правила на основании конечного множества наблюдаемых фактов (конкретных примеров). Это форма правдоподобного вывода от частного к общему, используемая для формирования эмпирических знаний интеллектуальной системы.

Другой формой правдоподобного вывода является *абдукция* — механизм формирования гипотезы, объясняющей наблюдаемые факты на основе существующих теоретических положений (законов). Это вывод от частного к частному, труднее всего поддающийся алгоритмизации и использующий для верификации (оценки) выдвигаемых гипотез (объяснительных гипотез) методы индукции.

Указанные формы рассуждения можно проиллюстрировать на примерах следующих трех силлогизмов.

1. Все автомобили, сошедшие с конвейера 1 апреля, имеют дефект в рулевом управлении.

Эти автомобили сошли с конвейера 1 апреля.

Эти автомобили имеют дефект в рулевом управлении.

2. Эти автомобили сошли с конвейера 1 апреля.

Эти автомобили имеют дефект в рулевом управлении.

Все автомобили, сошедшие с конвейера 1 апреля, имеют дефект в рулевом управлении.

3. Все автомобили, сошедшие с конвейера 1 апреля, имеют дефект в

рулевом управлении.

Эти автомобили имеют дефект в рулевом управлении.

Эти автомобили сошли с конвейера 1 апреля.

Первый силлогизм представляет дедуктивное рассуждение, второй - индуктивное обобщение, третий – абдуктивное рассуждение.


Таким образом, указанные механизмы призваны обеспечить:

- аналитический (дедуктивный) логический вывод, направленный на решение задач;
- синтетический (индуктивный и абдуктивный) вывод, обеспечивающий развитие базы знаний и расширение возможностей дедуктивного метода за счет отказа от четко фиксированного множества аргументов в цепочках рассуждений, использования стратегий и правил правдоподобного вывода, применения металогических средств управления выводом.

Традиционно в составе СОЗ присутствуют пять основных компонент (рис.2): база знаний, механизм вывода (интерпретации знаний), подсистема объяснения, подсистема приобретения знаний и интеллектуальный интерфейс.

База знаний обычно включает две составляющие. Одна из них объединяет в себе *долговременные знания* о предметной области, которые могут быть представлены в виде набора *продукционных правил*, иерархических структур *фреймов*, *семантических сетей* или других информационных структур, комбинирующих упомянутые и, возможно, иные формы представления знаний. Другая составляющая базы знаний представляет ее динамическую часть, в которой хранятся факты (оперативные данные), описывающие текущую ситуацию (например, состояние процесса проектирования). В системах продукционного типа эту составляющую называют *рабочей памятью*, а долговременные знания *базой правил*.

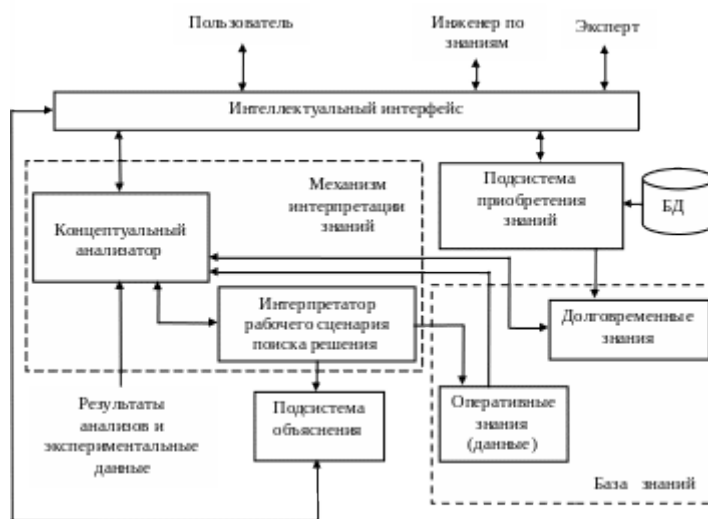
Механизм интерпретации знаний в общем случае объединяет в себе *концептуальный анализатор* и *интерпретатор рабочего сценария поиска решения*. Концептуальный анализатор по оперативным данным на основе долговременных знаний прогнозирует действия, востребованные текущей ситуацией, планируя шаг за шагом рабочий сценарий поиска решения. Интерпретатор реализует

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

выполнение действий и внесение, в зависимости от их результатов, изменений в информационные структуры, характеризующие состояние процесса поиска.

Подсистема объяснения призвана показать, в случае необходимости, в понятной для пользователя форме ход «рассуждений» (рабочий сценарий) механизма вывода для обоснования принятого им решения.

Подсистема приобретения знаний предназначена для выявления долговременных знаний из возможных источников (у эксперта; из накапливаемого опыта решения конкретных задач; из баз данных, хранящих готовые базы знаний; из фоновых рассуждений системы в виде доказательства теорем на уровне знаний общего плана и т.д.) и приведения их к формату, воспринимаемому механизмом вывода. Одно из основных требований, предъявляемых к этой подсистеме, заключается в обеспечении открытости базы знаний и максимальной комфортности в плане возможности ее модификации (замены, удаления или добавления новых фрагментов знаний) в оперативном режиме без глобальных преобразований ее структуры.



Интеллектуальный интерфейс объединяет в себе лингвистические, информационные и программные средства взаимодействия пользователя, инженера по знаниям (аналитика) и эксперта с соответствующими компонентами инструментария. Они должны быть ориентированы на неподготовленного пользователя, обладать способностью настраиваться на его терминологию и создавать, по возможности, комфортные условия для работы в системе.

Конкретное содержание, с точки зрения выполняемых функций, принципов реализации и методики использования, рассмотренные компоненты приобретают после выбора модели представления и формализации знаний.

Модели представления знаний

Данные и знания как категории информационного обеспечения задач

Данные и знания – это две категории информационного обеспечения задач, принципиально отличающиеся друг от друга ролью, выполняемой ими в вычислительном процессе.


Данные можно определить как информацию, полученную в результате наблюдений или измерений отдельных свойств объектов предметной области. Они используются при интерпретации *процедурных знаний* (знаний, заложенных в программах) и имеют удобную для этого структуру (простую или сложную).

Знания – это информация, отражающая полученные эмпирическим путем связи и закономерности предметной области, обеспечивающие возможность решения возникающих в ней задач.

Знания можно рассматривать как *данные о данных*, несущие информацию о том, каким образом последние следует интерпретировать.

Можно выделить ряд базовых свойств, отличающих знания от обычных данных.

Внутренняя интерпретируемость. Это свойство знания отображать в своей структуре всю информацию, необходимую для его содержательной интерпретации, т.е. семантику знания.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Структурированность и связность. Данное свойство предполагает возможность представления информационных единиц знания в виде иерархических (рекурсивно вложенных) структур и установления между ними внешних связей, выражающих различные типы отношений (отношения структуризации, каузальные отношения, пространственные и временные отношения, отношения ситуационной близости и другие). Обозначенные связи позволяют эффективно решать проблемы наследования информации, использования обобщенных процедур, сужения пространства поиска знаний, релевантных рассматриваемой ситуации, и т.д.

Шкальная и ассоциативная упорядоченность. Для взаимного упорядочения информационных единиц могут использоваться метрические (абсолютные и относительные) шкалы, порядковые шкалы (например, шкала оценок успеваемости студента), нечеткие порядковые шкалы (например, шкала «никогда – всегда» с рядом промежуточных значений: «почти никогда», «редко», «часто», «почти всегда»), оппозиционные шкалы (например, шкала «добрый – злой» с третьим, нейтральным, значением «не добрый — не злой»).

Помимо шкал для упорядочения сведений могут быть задействованы ассоциативные связи, выделяющие в структуре знаний типовые ситуации (*сцены*) и определяющие степень близости (отношение релевантности) к ним конкретных информационных единиц.

Частота возникновения тех или иных ситуаций или проявления в них конкретных элементов знания может служить основой для выделения информационных структур-заготовок для быстрого реагирования системы в условиях дефицита времени. Например, в ситуации «надо забить гвоздь» по умолчанию оперативно всплывает структура-заготовка «молоток».

Активность. Это свойство определяет способность знания активизировать систему на выполнение тех или иных действий. Знания предписывающего типа активизируют систему посредством программ, в которых они заложены. Декларативные знания активизируют СОЗ через программное ядро, основой которого являются механизм вывода и подсистема приобретения знаний, инициализация которых может быть вызвана появлением в базе знаний (ее оперативной части) исходных фактов для рассуждений или описаний новых фрагментов знаний.

Отмеченные свойства знаний, планируемые для использования механизмы вывода и стратегии управления выводом – все это должно приниматься во внимание при разработке концептуальной модели представления знаний как интеллектуальной основы создаваемой системы. На первом этапе решения этой проблемы важно правильно определиться с выбором тех или иных формализмов описания знаний, известных из теоретических работ в области искусственного интеллекта.

Модель представления знаний надежной и гибкой ЭС должна объединять в себе глубинные и поверхностные, качественные и количественные, приближенные и точные, конкретные и общие, описательные (декларативные) и предписывающие знания. Системы, использующие различные виды представления знаний, называются *многоуровневыми*. Они предоставляют экспертам и аналитикам широкие возможности для систематизации знаний с целью эффективного их использования в ходе решения конкретных прикладных задач.

Логические модели представления знаний, основанные на исчислениях

Одним из формализмов представления знаний являются *логические модели*. В их основе лежит понятие формальной системы, заданной четверкой

$$M = \langle T, P, A, F \rangle,$$

где T – множество базовых элементов системы;


P – множество синтаксических правил построения из элементов множества T синтаксически правильных выражений;

A – множество аксиом (априорно истинных выражений);

F – семантические правила (правила вывода), позволяющие получать из аксиом другие истинные в рамках данной системы выражения.

По специфике применения правил вывода формальные системы могут быть трех классов:

- исчисления;
- продукционные системы;
- алгоритмы.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Исчисление позволяет применять любое из предусмотренных в нем правил вывода к любой уже выведенной формуле, если последняя допускает применение этого правила.

В продукционной системе каждое правило вывода (*продукционное правило*) имеет свои условия применимости (*условия активации*) в виде набора фактов (атрибутов и их значений), характеризующих состояние процесса вывода, при котором правило претендует на срабатывание. В ходе рассуждений некоторые из фактов могут менять свои значения из-за появления новой информации. Это характерно для систем с *немонотонным выводом*.

Что касается алгоритма как формальной системы, то последовательность применения правил вывода в нем определена однозначно.

В СОЗ, как системах декларативного типа, логические модели представления знаний базируются на исчислениях и (или) продукциях.

Классическими примерами формальных систем, используемых для представления знаний в моделях дедуктивного вывода, являются исчисление высказываний и исчисление предикатов первого порядка. Надо заметить, что вторая система является более предпочтительной по сравнению с первой при использовании средств интеллектуальной поддержки в составе прикладных программ для придания им интеллектуальности. Ибо исчисление предикатов обеспечивает большую гибкость и органичность связей интеллектуальной составляющей с проблемной областью через предметные переменные предикатов.

В исчислении предикатов знания о предметной области описываются множеством общезначимых (истинных во всех интерпретациях) формул (аксиом). Цель конкретной поставленной проблемы (задачи) представляется синтаксически правильной формулой, а процесс ее решения сводится к доказательству общезначимости этой формулы на основании аксиом, посылок и правил вывода формальной системы (другими словами, к доказательству теоремы).

Модели представления и интерпретации знаний, основанные на исчислении предикатов первого порядка, отличаются высокой степенью формализации, универсальностью подхода. Однако они оказываются громоздкими для принятия решений в обширных пространствах поиска реальных предметных областей. Это объясняется их неспособностью учитывать должным образом семантику предметной области, применять эвристические процедуры для управления процессом вывода с целью придания ему направленного, рационального для данной предметной области характера. Это послужило причиной перехода к так называемым *семиотическим системам*, использующим формализм, задаваемый восьмеркой

$$M = \langle T, P, A, F, Q(T), Q(P), Q(A), Q(F) \rangle,$$

в которой к четверке обычной формальной системы добавлены еще четыре компонента — $Q(T), Q(P), Q(A), Q(F)$, определяющие правила управления первой четверкой в ходе накопления, обобщения и обновления знаний о предметной области и опыта функционирования интеллектуальной системы. По сути это двухуровневая система, нижний уровень которой представляет обычную формальную систему, а верхний – модель ее адаптации к реальным условиям применения.


Сетевые модели представления знаний

Наиболее полно (в явном виде) отразить семантику предметной области позволяют *сетевые модели* представления знаний. С этой формой представления связано понятие *семантической сети* как структуры, отображающей совокупность объектов предметной области и отношений между ними. При этом объектам соответствуют вершины (узлы) сети, а отношениям – дуги между ними.

Объектами могут быть обобщенные понятия, события, действия. Свойства указанных объектов также представляются вершинами. Совокупность объектов, включаемых в сеть, определяется содержанием предметной области и рассматриваемым кругом задач.

В отличие от *однородных сетей* (с одинаковыми отношениями между вершинами) *неоднородные сети* могут содержать дуги различных типов, имеющие различный смысл. Наиболее распространенными являются следующие типы отношений:

- *быть элементом класса;*
- *обладать свойством;*
- *являться следствием;*

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- *иметь значение* (определяет значения свойств объекта) и другие.

Вершины семантической сети могут иметь свою (внутреннюю) сетевую структуру. Тогда ее называют *сетью иерархического типа* (в отличие от *сети простого типа*).

Как разновидность сетевых моделей можно рассматривать *иерархические структуры фреймов*.

Фрейм (по определению М.Минского) – это структура для описания стереотипной ситуации, состоящая из характеристик этой ситуации (*слов*) и их значений (*заполнителей слов*).

Совокупность фреймов, моделирующая определенную предметную область, имеет иерархическую структуру. Каждый фрейм соответствует некоторому объекту предметной области. Слоты содержат описывающие этот объект данные. На верхнем уровне иерархии находится фрейм, отражающий наиболее общую информацию, присущую фреймам более низкого уровня. В качестве заполнителей слотов могут указываться:

- одно значение;
- несколько значений;
- фасет (диапазон или перечень возможных значений);
- правило, согласно которому определяется заполнитель слота;
- имя процедуры, реализующей алгоритм вычисления заполнителя слота.

Значения характеристик фреймов могут передаваться по умолчанию фреймам более низкой ступени иерархии, если те не содержат собственных значений данных характеристик. Различают *статические системы фреймов*, не меняющиеся в процессе решения задачи, и *динамические*.

К достоинствам сетевых, в частности фреймовых, моделей представления знаний следует отнести:

- явное представление иерархических связей;
- наследование информации;
- возможность вычисления значения любого слота с помощью процедур или эвристик.

Недостаток сетевых моделей состоит в сложности их реализации и внесения изменений в базу знаний.

Представление знаний в виде набора продукционных правил

Анализ действий экспертов, связанных с диагностикой состояний сложных объектов и систем, показывает, что профессионал проводит экспертизу вполне целенаправленно, придерживаясь определенной стратегии. При этом он руководствуется множеством правил (эвристик), которые могут быть представлены в форме

ЕСЛИ <посылка> **ТО** <заключение>,

называемой *продукционным правилом*.

Первая часть правила (<посылка>) называется *антецедентом*, вторая (<заключение>) – *консеквентом*. Антецедент состоит из элементарных предложений (высказываний), соединенных логическими связками **И** и выражающих условия срабатывания правила. Консеквент включает одно или несколько предложений, описывающих выдаваемое правилом заключение в виде некоторых фактов и/или ссылки на определенное действие.


Правило может быть задано с *коэффициентом уверенности*, определяющим положительным значением меньше единицы степень адекватности вывода правила (консеквента) условиям его срабатывания (антецеденту). Другими словами, коэффициент уверенности правила определяет степень близости его к точной логической конструкции, для которой его значение принимается равным единице. Коэффициент уверенности может быть задан в скобках в конце правила.

С коэффициентами уверенности могут задаваться и факты антецедента, в достоверности которых существует определенная доля сомнения.

При организации базы знаний антецеденты и консеквенты продукционных правил могут представляться совокупностью пар <атрибут, значение> или троек <атрибут, объект, значение>. Это приводит к экономии числа задействованных переменных. Однако надо заметить, что использование троек усложняет процесс формирования правил и алгоритм вывода.

Представление знаний продукционными правилами обладает следующими преимуществами:

- модульность;
- единообразии структуры (возможность построения и использования оболочек);

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- естественность (имитация рассуждений эксперта);
- гибкость иерархии понятий с точки зрения внесения изменений.

Вместе с тем данному представлению присущи и некоторые недостатки:

- громоздкость процесса вывода, связанная с проверкой условий применимости правил;
- сложность управления процессом вывода;
- отсутствие наглядности представления иерархии понятий.

Если с точки зрения отмеченных достоинств и недостатков сопоставить продукционные правила с другими формами представления знаний, например, такими как иерархические структуры фреймов и семантические сети, то можно сделать следующие выводы.

Отмеченные преимущества продукционных правил неоспоримы, хотя создавать и настраивать на конкретные предметные области можно и оболочки СОЗ, использующие фреймовые структуры или семантические сети. Но это оказывается сложнее с точки зрения программной реализации, а главное, затрудняет процесс формализации конкретных знаний на инфологическом уровне. В продукционной системе (системе, базирующейся на продукционных правилах) для этого удобно используется *дерево решений*, которое к тому же существенно компенсирует недостаток продукционных правил, связанный с отсутствием наглядности представления иерархии понятий.

Что касается громоздкости процесса вывода на продукционных правилах и сложности управления им, то эти проблемы также решаются, например, путем учета статистики срабатывания правил, организации распределенной структуры базы правил и использования других доступных приемов. Труднее компенсировать отсутствие у продукционных систем явного компактного представления иерархических связей понятий и наследования по умолчанию значений слотов путем ссылок на прототипы, характерных для фреймовых структур и семантических сетей.

В пользу продукционной модели представления знаний говорит и то, что она хорошо согласуется с представлением задач *в пространстве состояний*, характерным для целевого назначения многих интеллектуальных систем.

В системе интеллектуальной поддержки продукционного типа процесс поиска решений реализуется как логический вывод, основанный на сопоставлении по образцу. При этом состояние процесса вывода (процесса поиска решения) определяется содержимым *рабочей памяти* базы знаний, а роль операторов, переводящих систему из одного состояния в другое, выполняют продукционные правила.

Построение базы знаний

Многогранность классификации знаний, сложность и специфичность их выявления в различных предметных областях обуславливают плохую структурированность и цикличность процесса построения базы знаний. Тем не менее можно выделить нечто общее, определяющее методологию построения базы знаний в целом.

Циклический процесс построения базы знаний складывается из следующих этапов:

- исследование и описание предметной области;
- организация модели представления и формализация знаний;
- приобретение знаний.


5.1. Исследование и описание предметной области

На данном этапе выполняется следующее:

- определяется класс решаемых задач (целевое назначение системы);
- выделяются объекты предметной области и устанавливаются связи между ними;
- выявляются особенности предметной области.

Целевое назначение системы должно быть определено точно, достаточно полно и непротиворечиво. Например, от системы диагностики неисправности автомобиля можно потребовать получение ответов на следующие вопросы:

1. Исправен ли автомобиль? Если нет, то, что именно неисправно?
2. Как устранить неисправность?
3. Следует ли обратиться для устранения неисправности на станцию технического обслуживания?
4. Как предупредить подобную неисправность в будущем?

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Целевое назначение системы обычно детализируется в виде иерархической структуры: задачи, подзадачи, подподзадачи и т.д. Цели каждого уровня иерархии представляют собой часть некоторой общей цели или одно из возможных направлений поиска. Например, исходную задачу «Исправен ли автомобиль?» можно разбить на подзадачи:

1. Исправна ли система питания?
2. Исправна ли система зажигания?
3. Исправна ли система охлаждения?
4. Исправна ли тормозная система?

и т.д. Это нисходящий метод описания — от абстрактного (общего) к конкретному.

При определении и структуризации целевого назначения системы используют то или иное представление задач, исходя, как правило, из двух основных критериев:

- достаточно точное отражение реальности (адекватность модели);
- удобство работы с данным представлением.

Среди известных типов представления задач можно выделить следующие.

Перечисляющее представление. Базируется на рассмотрении множества возможных решений с использованием определенного правила их оценки. Это представление основано на методе слепого порождения пробных решений до тех пор, пока не будет получено приемлемое решение. Каждый последующий шаг не зависит от результатов предыдущего и не отслеживает направление на цель. В качестве примера можно привести перечисляющее представление оптимизационной задачи, решаемой методом Монте-Карло.

Представление в пространстве состояний. Предполагает существование счетного множества S состояний и множества O операторов, отображающих состояния множества S в себя. Процесс поиска решения представляет собой передвижение в пространстве состояний с целью достижения желаемого множества состояний. Задача считается решенной, если найдена последовательность операторов

$$O^* = o_1, o_2, \dots, o_k,$$

$$\text{порождающая } s^* = o_k(o_{k-1}(\dots(o_2(o_1(s^0)))) \dots),$$

где s^* — некоторое состояние из множества целевых состояний;


s^0 — некоторое состояние из множества начальных состояний.

Удобно графовое представление задач в пространстве состояний, в котором узлам (вершинам) графа соответствуют состояния, а ребрам (дугам) — операторы. Оно отличается наглядностью, естественностью обобщения задачи на случай учета стоимости выполнения операторов (взвешенный граф), а также возможностью во многих случаях выбирать каждый последующий шаг как результат сравнения целевого и текущего состояний.

В качестве примера опять можно взять оптимизационную задачу, но решаемую с помощью последовательного алгоритма поисковой оптимизации. Множеству состояний в этом случае соответствует множество точек пространства варьируемых переменных. Функции операторов, переводящих процесс поиска из одного состояния в другое, выполняют базовые оптимизационные процедуры, заложенные в алгоритме.

Иерархическое представление. Оперирует *деревом задачи*, которое каждой подзадаче определенного уровня иерархии ставит в соответствие совокупность ее И-подзадач либо ИЛИ-подзадач следующего (более низкого) уровня иерархии. Это означает, что для решения исходной подзадачи необходимо решить все ее И-подзадачи либо одну из ИЛИ-подзадач. В процессе поиска решения каждый узел дерева, начиная с листьев, помечается как решенная задача. Узлы, для которых какая-нибудь ИЛИ-подзадача либо все И-подзадачи решены, помечаются как решенные. Процесс поиска продолжается до тех пор, пока не будет помечен корень дерева, т.е. решена исходная задача.

Комбинированное представление. Использует различные типы представлений. Например, иерархическое представление задачи может быть полезно для представления ее глобальных аспектов. Для решения более специфических задач целесообразно использовать пространство состояний или перечисляющее представление.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Для систем продукционного типа характерно представление задач в пространстве состояний. Вместе с тем при структуризации задач, например с целью создания распределенной базы правил, может быть использовано и иерархическое представление.

Следующим шагом после определения и структуризации целевого назначения системы является выделение объектов предметной области и установление связей между ними. Эта фаза исследования и описания предметной области в наибольшей степени зависит от содержания предметной области и специфики рассматриваемых для нее задач. В качестве способов определения базовых понятий предметной области можно использовать введение их посредством *интенционала* или *экстенционала*.

Интенционал понятия реализует аналитический подход к его определению через абстракции более высокого уровня с указанием специфических свойств. Например, интенционал понятия «дерево» — «многолетнее растение с твердым стволом и отходящими от него ветвями, образующими крону».

В отличие от интенционала, *экстенционал* использует синтетический метод определения понятия путем перечисления его конкретных примеров, т.е. понятий более низкого уровня абстракции. Например, экстенционал того же понятия «дерево» — «береза, дуб, сосна, рябина, яблоня и т.д.».

Совокупность выделенных объектов (понятий) предметной области должна быть достаточно полной с точки зрения класса рассматриваемых задач и непротиворечивой. Связи между объектами должны быть выявлены по возможности все.

В качестве особенностей предметной области можно рассматривать:

- перекрытие некоторых признаков (фактов) другими;
- скачкообразное изменение отдельных признаков (в зависимости от внешних условий);
- трудность или невозможность анализа ряда признаков;
- вероятность сбоев используемого оборудования.

Организация модели представления и формализация знаний

Любое «осмысленное» поведение искусственной системы в условиях реального мира требует наличия у нее определенным образом организованной модели этого мира. В надежной и гибкой ЭС в основе такой модели, очевидно, должны лежать различные виды представления знаний: продукционные правила, фреймовые структуры, семантические сети.


Процесс формализации знаний может быть упрощен путем разбиения их на классы в соответствии с иерархией целей (задач). При этом способ организации знаний, состав классов, вид структуры зависят от того, с какой точки зрения рассматривается предметная область. Можно выделить три вида иерархии в зависимости от точки зрения на предметную область:

- структурная иерархия;
- причинно-следственная иерархия;
- функциональная иерархия.

С точки зрения *структурной иерархии*, например, автомобиль рассматривается как совокупность иерархически организованных физических компонентов (агрегатов, узлов, деталей). При этом поиск неисправности автомобиля можно осуществлять путем последовательной замены агрегатов, узлов, деталей. Такой метод локализации неисправности не требует никаких специальных знаний об устройстве и функционировании автомобиля. Он базируется на поверхностных знаниях (знаниях только о том, как разобрать и собрать автомобиль).

Причинно-следственная иерархия лежит в основе модели поведения системы. Например, поворот ключа в замке зажигания автомобиля должен привести к включению определенных контрольных приборов. Если этого не происходит, необходимо перейти к следующему уровню иерархии и проверить исправность аккумулятора, приборов и т.д. Очевидно, что причинно-следственная иерархия базируется на знаниях более глубокого уровня.

Функциональная иерархия связана с самым высоким уровнем абстракции, когда объект рассматривается как совокупность систем, подсистем и модулей, выполняющих определенные функции. Например, в функциональную иерархию автомобиля входят система зажигания, система питания, система охлаждения, тормозная система и т.д. В свою очередь, система зажигания включает подсистему (цепь) низкого напряжения и высоковольтную цепь и т.д. Очевидно, что функциональная иерархия базируется на глубинных знаниях.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Модель представления знаний необходима для перехода от содержательного описания предметной области к форме, допускающей включение этого описания в базу знаний.

Приобретение знаний

Знания системы могут формироваться на основе:

- рекомендаций эксперта;
- аналогий;
- примеров;
- наблюдений, открытий и экспериментов;
- умозаключений на базе глубинных представлений.

Неавтоматизированный сбор знаний экспертов – это трудоемкий процесс, плохо структурированный и носящий циклический характер. Одна его итерация состоит в получении от эксперта максимального количества знаний о каком-либо фрагменте предметной области и анализ их на непротиворечивость.

Для ускорения процесса накопления знаний и придания ему комфортного характера необходима разработка специального программного интерфейса приобретения знаний от эксперта. При создании такого интерфейса важно правильно определиться с языком описания знаний на инфологическом (содержательном) уровне. С одной стороны, этот язык должен быть достаточно формален, нагляден и удобен для эксперта; он должен позволять эксперту использовать свою терминологию на естественном языке. С другой стороны, этот язык не должен создавать проблем компьютеру с его распознаванием и интерпретацией, как, например, естественный язык.

Указанным требованиям при описании знаний продукционного типа на инфологическом уровне в достаточной степени удовлетворяет традиционное *дерево решений*. Это бинарная древовидная структура с вершинами двух типов:

- *вершины условий* (типа «овал»), из которых могут исходить только две ветви (одна – по «Да», другая – по «Нет»);
- *вершины выводов* (типа «прямоугольник»), имеющие единственный выход (обязательно на вершину условия) для промежуточных выводов и не имеющие выхода (не продолжающие логику рассуждений) в случае окончательного вывода.

Каждая вершина условия содержит некоторое высказывание, которое может принимать значения «Истина» или «Ложь» (соответствует выходам по «Да» или «Нет»). Вершина вывода содержит одно или несколько предложений, описывающих некоторое промежуточное или окончательное заключение в виде набора фактов или ссылок на определенные действия. Корень дерева обозначается как вершина условия (овал), содержащая высказывание, с которого начинается процесс логических рассуждений.

Рассмотренная структура дерева решений обладает следующими недостатками. Во-первых, она не ориентирована на описание нечеткой логики рассуждений эксперта. Во-вторых, заставляет дублировать фрагменты дерева решений в случаях прихода к ним с разных направлений.


Избавиться от указанных недостатков позволяет введение понятия *графа решений*.

Граф решений представляет собой ориентированный помеченный граф с вершинами двух типов:

- *вершины условий* (типа «овал»), из которых могут исходить несколько ребер (по «Да» или «Нет» со своими коэффициентами уверенности, помечающими эти ребра);
- *вершины выводов* (типа «прямоугольник»), имеющие один вход и один выход (обязательно на вершину условия) для промежуточных выводов и не имеющие выхода (не продолжающие логику рассуждений) в случае окончательного вывода.

Каждая вершина условия (вершина-овал) содержит некоторое высказывание, которое может принимать значения «Истина» или «Ложь» с различными коэффициентами уверенности (каждому коэффициенту уверенности соответствует помеченный им выход).

Вершина вывода содержит одно или несколько предложений, описывающих некоторое промежуточное или окончательное заключение в виде набора фактов или ссылок на определенные действия. Вершина вывода помечается (например, в нижнем правом углу) *коэффициентом уверенности*, определяющим положительным значением меньше единицы степень адекватности (правдоподобия) вывода условиям его активации.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Граф имеет одну корневую вершину-овал, содержащую высказывание, с которого начинается описание логики рассуждений. Некоторые из вершин-овалов могут иметь несколько входных ребер с разных направлений, не приводящих к появлению обратных связей (исключением является корневая вершина-овал, не имеющая ни одного входного ребра).

Таким образом, граф решений позволяет описать более глубокую, нечеткую логику рассуждений эксперта в предметной области. Для описания структурированных (с целью сужения пространства поиска) знаний может использоваться не один, а несколько графов решений.

Для реализации механизмов приобретения знаний на основе аналогий, примеров, наблюдений и экспериментов скорее всего целесообразно использовать нейросетевые технологии.

Умозаключения на базе глубинных представлений в виде логических моделей, основанных на исчислениях, можно организовать посредством доказательства теорем с применением метода резолюций Робинсона.

Задание на самостоятельную работу


1. Создайте базу знаний «Знатоки», которая состоит из следующих фактов и правил, выраженных предложениями на естественном языке:

- ☐ Мария знает английский язык.
- ☐ Мария знает информатику.
- ☐ Мария знает музыку.
- ↪ Иван знает то же, что и Мария.

Первые три предложения - это факты. Четвертое – правило.

2. Ответьте на следующие вопросы:

1. Назовите основные признаки, по которым можно провести четкую грань между понятиями ИС, ЭС и СОЗ.
2. Перечислите основные способности (возможности), которыми обладает человек-эксперт.
3. Какие возможности заложены в современных экспертных системах?
4. Что отличает глубинные знания от поверхностных?
5. В чем состоит суть концепции СОЗ?
6. Назовите основные области применения СОЗ.
7. По каким критериям определяется целесообразность создания и использования СОЗ в конкретной предметной (проблемной) области?
8. Чем отличается автономная СОЗ от неавтономной?
9. Что Вы понимаете под оболочкой СОЗ?
10. Что понимается под метапроцедурами, лежащими в основе интеллектуальной деятельности человека?
11. Перечислите три известных формы рассуждения и определите суть каждой из них.
12. Назовите пять основных компонент, традиционно присутствующих в составе СОЗ, и определите функциональное назначение каждой из них.
13. В чем состоит принципиальное отличие знаний от данных?
14. Какие основные свойства знаний отличают их от данных?
15. Перечислите традиционные формы представления знаний.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

16. Как Вы оцениваете логические модели представления и интерпретации знаний, основанные на исчислениях, с точки зрения их применения для решения практических задач?
17. В чем состоит суть семиотической формальной системы?
18. Что представляет собой семантическая сеть?
19. Что представляет собой фрейм?
20. Каковы достоинства и недостатки сетевых моделей представления знаний?
21. Что представляет собой продукционное правило?
22. Можно ли с помощью продукционных правил описывать нечеткую логику? Если можно, то как?
23. Дайте сравнительную оценку моделей продукционного типа и сетевых моделей представления знаний.
24. Из каких основных этапов складывается циклический процесс построения базы знаний?
25. Какая последовательность действий выполняется на этапе исследования и описания предметной области?
26. Какими основными критериями руководствуются при выборе того или иного представления задач, на которые ориентируется система?
27. Какие типы представления задач Вы знаете? В чем состоит их суть?
28. При выделении объектов предметной области и установлении связей между ними необходимые понятия могут вводиться посредством интенционала или экстенционала. Чем различаются эти два способа определения понятий?
29. Что может рассматриваться в качестве особенностей предметной области?
30. Какие виды иерархий могут приниматься за основу при структуризации целевого назначения системы и организации модели представления знаний? В чем суть этих иерархий?
31. Какие источники приобретения знаний Вы можете назвать?
32. Что собой представляет дерево решений как формализм описания знаний на инфологическом уровне?
33. Изложите основные принципы трансформации дерева решений в эквивалентный набор продукционных правил.
34. В чем состоит суть понятия графа решений как обобщения понятия дерева решений?


Задание для самостоятельной работы:

- *Проработать лекционный материал, основную и дополнительную литературу с целью подготовки ответов на представленный перечень вопросов для устного опроса.*

Тема 2. Системы, основанные на знаниях

Практическое занятие. Создание и применение экспертной системы в оболочке CLIPS 6.3

Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Форма проведения занятия – практическое занятие.

Вопросы для самоконтроля и текущего контроля

1. Краткая история и характеристика CLIPS
2. Основные элементы программирования
3. Факты
4. Правила
5. Дополнительные средства CLIPS

Практическая разработка экспертных систем в среде CLIPS

Постановка задачи.

Создание *экспертной системы управления технологическим процессом* (ТП) может значительно ускорить процесс разработки сложной системы управления ТП, повысить качество решения задачи и дать экономию ресурсов за счет эффективного *распределения функций* центрального управления и локальных измерительных и управляющих подсистем. Такой эффект достигается за счет открытости *системы представления знаний* об объекте управления, *адаптивности системы* к условиям функционирования, автоматической коррекции управляющих воздействий при изменении существенных параметров в процессе функционирования.


ЭС *CLIPS* рассматривается в лекции как инструментальное средство для разработки. Выбор *CLIPS* обусловлен двумя причинами: во-первых, эта ЭС, разработанная *NASA*, доказала свою эффективность и **свободно распространяется через Internet**; во-вторых, реализация *CLIPS* на языке C++ позволяет переносить конкретные ЭС на различные типы операционных систем. Кроме того, может быть обеспечена возможность работы в реальном масштабе времени, когда *реакция* системы на возмущения должна не превышать нескольких миллисекунд.

В качестве ТП рассмотрим создание деталей сложной формы, например, вытачивание лопаток турбины. В данном случае ИРС осуществляет управление технологическим процессом через систему управления высшего уровня, способную к самостоятельному функционированию и обеспечивающую выполнение всех основных функций по управлению сбором и анализом информации и принятию оперативных решений по ходу процесса на основе разрабатываемой ЭС. В состав ИРС входит ряд локальных управляющих подсистем нижнего уровня, каждая из которых осуществляет управление одним из компонентов ТП по жесткому алгоритму в реальном времени. ЭС управления ТП, разрабатываемая в рамках данной лекции, обеспечивает организацию сбора информации об *управляемом процессе* от локальных управляющих подсистем, управление режимами их функционирования и принятие оперативных управляющих решений на основе информации, поступившей от систем управления нижнего уровня. В общем случае управление ТП может осуществляться полностью автоматически.

Приступим к *формализации знаний* экспертов по управлению ТП создания деталей сложной формы. Выделим множество информативных (существенных) параметров, влияющих на ТП и позволяющих управлять ТП с некоторой достоверностью. Одновременно для выбранных параметров выделим информативные значения или информативные диапазоны значений. Указанные параметры и их значения представлены в [табл. 7.1](#).

Таблица 7.1. Информативные параметры ТП

№№ п/п	Обозначение параметра	Название параметра	Единица измерения	Диапазон значений
1	Vr	Скорость резания	об/мин	А, В, С
2	Vm	Подача	мм/с	10-180 с шагом 10
3	T	Виды траектории		Круговая (T=0), по участкам 1(T=1), ..., по участкам 6(T=6)

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

4	I	Инструмент		алмазный (I=1), на бакелитовой основе (I=2), на эльборовом основе (I=3)
5	G	Геометрические параметры инструмента		тор (G=tor), линия (G=line), макс. радиус вращения Rm, угол контакта инструмента и детали J, угол заточки S
6	Ds	Размер детали	мм	10,300,800
7	Dm	Материал детали		Титан1 (Dm=1), Титан2 (Dm=2), Жаропрочная сталь (Dm=3)
8	Da	Требования к детали по точности		1, 2, 3
9	Dar	Достигнутая точность детали		1, 2, 3


Запишем со слов экспертов информационные образы управляющих решений в алфавите значений информационных параметров. В [таблице 7.2](#) представлена база знаний (база правил) нашей экспертной системы управления технологическим процессом. Здесь достоверность это уверенность эксперта, что такое воздействие позволит достичь заданных параметров обработки *Ds*, *Dm*, *Da*, *Dar* на основе данного воздействия.

№№ п/п	<i>Ds</i>	<i>Dm</i>	<i>Da</i>	<i>Dar</i>	Управляющее воздействие	Достоверность
1	10	1	1		$Vr=A, Vm=10, T=0, I=1, G=tor$	0,98
2	10	2	2		$Vr=B, Vm=10, T=1, I=1, G=line, Rm=40, J=80, S=60$	0,95
3	300	2			$Vr=B, Vm=20, T=2, I=1, G=tor$	0,92
4	300		3		$Vr=C, Vm=40, T=3, I=2, G=line, Rm=50, J=75, S=75$	0,97
5	800	2	2	1	$Vr=B, Vm=60, T=4, I=2, G=line, Rm=60, J=70, S=70$	0,94
6	800			< 3	$Vr=C, Vm=80, T=6, I=3, G=line, Rm=60, J=60, S=75$	0,90
7	800			3	$Vr=B, Vm=40, T=6, I=3, G=line, Rm=60, J=60, S=75$	0,90

Достоверность правильности управляющего воздействия должна автоматически корректироваться по результатам изготовления детали. В [табл. 7.2](#) приведен учебный пример базы знаний, упрощенный для целей реализации. Здесь не сформулированы задачи работы с базой данных. База целей (конфликтное множество правил) является внутренним для CLIPS механизмом. В общем случае, в процессе обработки производится измерение параметров, и управляющие воздействия задаются в зависимости от результатов измерений и БЗ управляющих воздействий. Например в данном примере, пока *точность* детали *Dar* < 3, работает строка 6 [таблицы 7.2](#), как только *Dar* достигло значения 3, начинает работать строка 7. Это и есть простейший пример работы ЭС в реальном времени.

Основы программирования в системе CLIPS.

CLIPS (C Language Integrated Production System) начала разрабатываться в космическом центре Джонсона NASA в 1984 году. Сейчас CLIPS и документация на этот инструмент свободно распространяется через интернет (<http://www.ghg.net/clips/CLIPS.html>). Язык CLIPS свободен от недостатков предыдущих инструментальных средств для создания ЭС, основанных на языке LISP. Язык CLIPS получил большое распространение в государственных организациях и учебных заведениях благодаря низкой стоимости, мощности, эффективности и переносимости с платформы на платформу. Например, даже Web-ориентированный инструментальный JESS (Java Expert System Shell), использующий язык представления знаний CLIPS, приобрел достаточную известность в настоящее время.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Следует отметить, что несмотря на многочисленные преимущества функционального программирования, некоторые задачи лучше решать в терминах объектно-ориентированного программирования (ООП), для которого характерны три основные возможности: *ИНКАПСУЛЯЦИЯ* (работа с классами), *ПОЛИМОРФИЗМ* (работа с родовыми функциями, поддерживающими различное поведение функции в зависимости от типа аргументов), *НАСЛЕДОВАНИЕ* (*поддержка абстрактных классов*). ООП поддерживает многие языки, в том числе *Smalltalk*, *C++*, *Java*, *Common LISP Object System (CLOS)*. Язык *CLIPS*, в свою очередь, вобрал в себя основные преимущества *C++* и *CLOS*.

Читатель может познакомиться с языком *CLIPS*, получив через *Интернет* полный комплект документации на английском языке, или прочитав изданную на русском языке книгу. В данном разделе лекции дается краткое неформальное введение в *CLIPS*, необходимое для программирования учебных задач.

Отличительной особенностью *CLIPS* являются конструкторы для создания баз знаний (БЗ):

defrule	определение правил;
deffacts	определение фактов;
deftemplate	определение шаблона факта;
defglobal	определение глобальных переменных;
deffunction	определение функций;
defmodule	определение модулей (совокупности правил);
defclass	определение классов;
definstances	определение объектов по шаблону, заданному defclass ;
defmessagehandler	определение сообщений для объектов;
defgeneric	создание заголовка родовой функции;
defmethod	определение метода родовой функции.

Конструкторы не возвращают никаких значений, в отличие от функций, например:

```
(deftemplate person
 (slot name)
 (slot age)
 (multislot friends))
(deffacts people
 (person (name Joe) (age 20))
 (person (name Bob) (age 20))
 (person (name Joe) (age 34))
 (person (name Sue) (age 34))
 (person (name Sue) (age 20)))
```


Пример функции:

```
(deffunction factorial (?a)
  (if (or (not (integerp ? a)) (< ? a0)) then
    (printout t "Factorial Error!" crlf)
  else
    (if (= ? a0) then
      1
    else
      (* ? a (factorial ($-$ ? a1))))))
```

Правила в *CLIPS* состоят из предпосылок и следствия. Предпосылки также называют ЕСЛИ-частью правила, левой частью правила или LHS правила (*left-hand side of rule*). Следствие называют ТО-частью правила, правой частью правила или RHS правила (*right-hand side of rule*).

Пример правила представлен ниже:

```
(deftemplate data (slot x) (slot y))
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

```
(defrule twice
(data (x ? x) (y =(*2 ? x)))
=>)
(assert (data (x2) (y4)); f-0
        (data (x3) (y9))); f-1
```

Здесь самая распространенная в *CLIPS* функция *assert* добавляет новые факты в *список* правил. В противоположность *assert* функция *retract* удаляет факты из списка фактов, например:

```
(defrule vis11
?doors < — (fit ? wdfit)
(test (eq ? wdfit no))
=>
(assert (EVIDENCE OF MAJOR ACCIDENT))
(retract ? doors))
```

В этом правиле проверяется наличие факта *doors* и в случае его отсутствия факт *doors* удаляется из списка фактов задачи.

Функция *modify* является также весьма распространенной. Она позволяет в определенном факте поменять *значение* слота, например,

```
(deftemplate age (slot value))
(assert (age (value young)))
(modify 0 (value old))
```

Следующий пример описывает *представление* данных в виде фактов, объектов и глобальных переменных. Примеры фактов:

```
(voltage is 220 volt)
(meeting (subject "AI") (chief "Kuzin") (Room "3240"))
```

В первой строке приведен упорядоченный факт, во второй - неупорядоченный, в котором порядок слотов не важен.

CLIPS поддерживает следующие типы данных: *integer*, *float*, *string*, *symbol*, *external-address*, *fact-address*, *instance-name*, *instance-address*.

Пример integer: 594 23 +51 -17

Пример float: 594e² 23.45 +51.0 -17.5e⁻⁵

String — это *строка символов*, заключенная в двойные кавычки.


Пример *string*: "expert", "Phil Blake", "состояние \$-0\$", "quote=\"

CLIPS поддерживает следующие процедурные функции, реализующие возможности ветвления, организации циклов в программах и т.п.:

If	оператор ветвления;
While	цикл с предусловием;
loop-for-count	итеративный цикл;
prong	объединение действий в одной <i>логической команде</i> ;
prong\$	выполнение набора действий над каждым элементом поля;
return	прерывание функции, цикла, правила и т.д.;
break	то же, что и return , но без возвращения параметров;
switch	оператор множественного ветвления;
bind	создание и связывание переменных.

Функции *CLIPS* описываются в книгах. Среди **логических функций** (возвращающих значения **true** или **false**) следует выделить следующие группы:

- функции булевой логики: **and**, **or**, **not**
- функции сравнения чисел: =, \neq , >, \geq , <, \leq

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


- предикативные функции для проверки принадлежности проверяемому типу: **integerp**, **floatp**, **stringp**, **symbolp**, **pointerp** (относится ли аргумент к **xternal-address**), **numberp** (относится ли аргумент к **integer** или **float**), **lexemerp** (относится ли аргумент к **string** или **symbol**), **evenp** (проверка целого на четность), **oddp** (проверка целого на нечетность), **multifildp** (является ли аргумент составным полем).
 - *Функции сравнения* по типу и по значению: **eq**, **neq**
- Среди **математических функций** следует выделить следующие группы:
- Стандартные: **+**, **-**, *****, **/**, **max**, **min**, **div** (целочисленное деление), **abs** (абсолютное значение), **float** (преобразование в тип **float**), **integer** (преобразование в тип **integer**)
 - Расширенные: **sqrt** (извлечение корня), **round** (*округление числа*), **mod** (вычисление остатка от деления)
 - Тригонометрические: **sin**, **sinh**, **cos**, **cosh**, **tan**, **tanh**, **acos**, **acosh**, **acot**, **acoth**, **acsc**, **acsch**, **asec**, **asech**, **asin**, **asinh**, **atan**, **atanh**, **cot**, **coth**, **csc**, **csch**, **sec**, **sech**, **deg-grad** (преобразование из градусов в секторы), **deg-rad** (преобразование из градусов в радианы), **grad-deg** (преобразование из секторов в градусы), **rad-deg** (преобразование из радиан в градусы)
 - Логарифмические: **log**, **log10**, **exp**, **pi**

Среди **функций работы со строками** следует назвать функции:

str-cat	объединение строк,
sym-cat	объединение строк в значение типа symbol ,
sub-string	<i>выделение подстроки</i> ,
str-index	поиск подстроки,
eval	выполнение строки в качестве команды CLIPS ,
build	выполнение строки в качестве конструктора CLIPS ,
upcase	преобразование символов в символы верхнего регистра,
lowcase	преобразование символов в символы нижнего регистра,
str-compare	сравнение строк,
str-length	определение длины строки,
check-syntax	проверка синтаксиса строки,
string-to-field	возвращение первого поля строки.

Функции работы с составными величинами являются одной из отличительных особенностей языка **CLIPS**. В их число входят:

create\$	создание составной величины,
nth\$	получение элемента составной величины,
members	поиск элемента составной величины,
subset\$	проверка одной величины на подмножество другой,
delete\$	удаление элемента составной величины,
explode\$	создание составной величины из строки,
implode\$	<i>создание строки</i> из составной величины,
subseq\$	извлечение подпоследовательности из составной величины,
replace\$	замена элемента составной величины,
insert\$	добавление новых элементов в составную величину,
first\$	получение первого элемента составной величины,
rest\$	получение остатка составной величины,
length\$	определение числа элементов составной величины,
delete-member\$	удаление элементов составной величины,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

`replace-member$` замена элементов составной величины.

Функции ввода-вывода используют следующие логические имена устройств:

<code>stdin</code>	устройство ввода,
<code>stdout</code>	устройство вывода,
<code>wclips</code>	устройство, используемое как справочное,
<code>wdialog</code>	устройство для отправки пользователю сообщений,
<code>wdisplay</code>	устройство для отображения правил, фактов и т.,п.,
<code>werror</code>	устройство вывода сообщений об ошибках,
<code>wwarning</code>	устройство для вывода предупреждений,
<code>wtrase</code>	устройство для вывода отладочной информации,

Собственно *функции ввода-вывода* следующие:

<code>open</code>	открытие файла (виды доступа <code>r</code> , <code>w</code> , <code>r+</code> , <code>a</code> , <code>wb</code>),
<code>close</code>	закрытие файла,
<code>printout</code>	вывод информации на заданное устройство,
<code>read</code>	ввод данных с заданного устройства,
<code>readline</code>	ввод строки с заданного устройства,
<code>format</code>	форматированный вывод на заданное устройство,
<code>rename</code>	переименование файла,
<code>remove</code>	удаление файла.

Среди двух десятков команд *CLIPS* следует назвать основные команды при работе со средой *CLIPS*:

<code>load</code>	загрузка конструкторов из текстового файла,
<code>load+</code>	загрузка конструкторов из текстового файла без отображения,
<code>reset</code>	сброс рабочей памяти системы <i>CLIPS</i> ,
<code>clear</code>	очистка рабочей памяти системы,
<code>run</code>	выполнение загруженных конструкторов,
<code>save</code>	сохранение созданных конструкторов в текстовый файл,
<code>exit</code>	
	выход из <i>CLIPS</i> .


В рамках нашего краткого описания опустим *список* функций для работы с методами родовых функций и *список* функций для работы с классами, объектами, слотами, обработчиками сообщений. С этим можно ознакомиться по документации.

В завершение следует иметь в виду, что *CLIPS* может не удовлетворительно работать в реальном времени, когда потребуется *время реакции* менее 0,1 сек. В этом случае надо исследовать на разработанном прототипе *механизмы* вывода для всего *множества* правил *предметной области* на различных по производительности компьютерах. Как правило, современные мощные компьютеры Intel обеспечивают работу с производционными системами объемом 1000--2000 правил в реальном времени. Веб-ориентированные средства на базе *JAVA* (системы Exsys Corvid, JESS) являются более медленными, чем, например, *CLIPS* 6.0 или *OPS-2000*. Поэтому *CLIPS* - лучший на сегодня выбор для работы в реальном времени среди распространяемых свободно оболочек ЭС, разработанных на C++.

Программирование в *CLIPS* экспертной системы управления технологическим процессом.

Программа ЭС управления ТП по обработке деталей сложной формы, разработанная на основе [табл. 7.1](#) и [табл. 7.2](#), выглядит следующим образом.

```
;;;
;;; =====
;;; Control Expert System of technological process
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

```

;;;
;;; This expert system administers technological process
;;; of creations of details of the complex form

```

```

;;;
;;; CLIPS Version 6.0 Example

```

```

;;; To execute, merely load, reset and run.

```

```

=====

```

```

(defacts initial-state
  (Ds 800)
  (Dm 2)
  (Da 2)
(Dar 1))

```

```

(defrule rule1
  (declare (salience 9098))
  (Ds 10)
  (Dm 1)
  (Da 1)
=>
  (printout t "Rule1: Vr=A, Vm=10, T=0, I=1, G=tor " crlf))

```

```

(defrule rule2
  (declare (salience 9095))
  (Ds 10)
  (Dm 2)
  (Da 2)
=>
  (printout t "Rule2: Vr=B, Vm=10, T=1, I=1, G=line, Rm=40, J=80, S=60 " crlf))

```

```

(defrule rule3
  (declare (salience 9092))
  (Ds 300)
  (Dm 2)
=>
  (printout t "Rule3: Vr=B, Vm=20, T=2, I=1, G=tor " crlf))

```

```


(defrule rule4
  (declare (salience 9097))
  (Ds 300)
  (Da 3)
=>
  (printout t "Rule4: Vr=C, Vm=40, T=3, I=2, G= line, Rm=50, J=75, S=75" crlf))

```

```

(defrule rule5
  (declare (salience 9094))
  (Ds 800)
  (Dm 2)
  (Da 2)
  (Dar 1)
=>

```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

```
(printout t "Rule5: Vr=B, Vm=60, T=4, I=2, G= line, Rm=60, J=70, S=70 " crlf)
```

```
(defrule rule6_7
(declare (salience 9090))
(Ds 800)
(Dar ?num)
=>
(if (< ?num 3)
then
(printout t "Rule6: Vr=B, Vm=40, T=6, I=3, G= line, Rm=60, J=60, S=75 " crlf)
else
(printout t "Rule7: Vr=C, Vm=80, T=6, I=3, G= line, Rm=60, J=60, S=75 " crlf)))
```

Листинг 7.1. Программа ЭС управления ТП по обработке деталей сложной формы.

Эта программа сохраняется в виде файла с именем, например, *robot.clp*, далее в среде *CLIPS* выполняются команды: *clear; load robot.clp; reset* и *run*. Эта программа начинает работать. Входные воздействия заданы в данном примере через *deffacts initial-state*.

Активизация правил БЗ для конкретных воздействий, заданных в программе, дает конфликтное множество (базу целей) - правила *rule5* и *rule6_7*, а затем по критерию максимальной достоверности первым выбирается управляющее воздействие на систему низшего уровня:

Rule5: Vr=B, Vm=60, T=4, I=2, G= line, Rm=60, J=70, S=70

В реальной жизни входные воздействия поступают через оператор *read* (ввод данных с заданного устройства), например, следующим образом:


```
(defrule Dar_parameter
(declare (salience 9101))
(Dar ? num)
=>
(printout t "Ds parameter has value "crlf"
1) 10 "crlf" 2) 300 "crlf" 3) 800 "crlf" Choose 1—3 =>")
(assert (Dar =(read))))
```

Задание на самостоятельную работу

1. Реализовать ИС, основанную на знаниях по следующей дорожной карте.
- Сформулировать цель создания ИС и согласовать её с преподавателем.
Предпочтительно, чтобы цель ИС (решаемая интеллектуальная задача) соответствовала предметной области из списка:

1. Разработка ПО (ИС)
2. Анализ требований к ПО
3. Проектирование архитектуры ПО
4. Программная реализация ПО
5. Тестирование и отладка ПО
6. Внедрение и поддержка (сопровождение) ПО
7. Проектирование компьютерных интерфейсов
8. CASE-средства
9. Контроль качества ПО
10. Моделирование бизнес-процессов

По согласованию с преподавателем, студенты могут предложить свой вариант предметной области.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- *Описать входную и выходную информацию для ИС, уровни значений входной информации.*
- *Также необходимо разработать 2-3 примера работы системы (корректных сочетаний входной и выходной информации) и согласовать их с преподавателем.*
- *Реализовать в среде CLIPSWindows прототип ИС, включающий в себя:*
 - *Организацию ввода информации (опрос системой пользователя).*
 - *Обработку информации при помощи правил и функций, для достижения цели создания ИС. Общее количество правил и функций должно составлять не менее 10 (без учёта ввода-вывода данных).*
 - *Структуру функций системы (2-3 реально работающих функции, для остальных – описание работы на естественном языке).*
 - *Организацию вывода информации на экран и в файл.*

Реализуемая ИС должна основываться на знаниях, представленных в виде фактов или иных структур данных.

2. Ответьте на следующие вопросы:
 1. В чем заключается отличие переменных, фактов и экземпляров классов в CLIPS?
 2. Какие команды используются для создания и удаления фактов в CLIPS?
 3. Каким образом определяется очередность выполнения правил в CLIPS и для чего она может использоваться?
 4. Что может находиться в LHS правил в CLIPS?

Задание для самостоятельной работы:

- *Проработать лекционный материал, основную и дополнительную литературу с целью подготовки ответов на представленный перечень вопросов для устного опроса.*

Тема 2. Системы, основанные на знаниях

Практическое занятие. Создание спецификации знаний о предметной области в виде онтологии в редакторе Protege 5.6.2

Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.

Форма проведения занятия – практическое занятие.

Вопросы для самоконтроля и текущего контроля


1. Основные понятия
2. Определение понятия «онтология»
3. Структура онтологии
4. Классификация онтологий
5. Методы построения онтологий
6. Применение онтологий
7. Языки представления (Semantic web)
8. Существующие онтологические ресурсы

Определения онтологии

Слово "онтология" имеет два значения:

Форма А

Страница 45 из 67

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- **Онтология 1** - философская дисциплина, которая изучает наиболее общие характеристики бытия и сущностей;
- **Онтология 2** - это артефакт, структура, описывающая значения элементов некоторой системы.

Настоящий курс посвящен способам разработки и использования в приложениях онтологий как артефактов (*Онтология 2*).

Неформально *онтология* представляет собой некоторое описание взгляда на мир применительно к конкретной области интересов. Это описание состоит из терминов и правил использования этих терминов, ограничивающих их значения в рамках конкретной области.

На формальном уровне *онтология* - это система, состоящая из набора понятий и набора утверждений об этих понятиях, на основе которых можно описывать классы, отношения, функции и индивиды.

Одно из самых известных определений онтологии дал Том Грубер, звучит оно следующим образом: *Онтология* - это точная спецификация *концептуализации*.

Концептуализация - это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации.

Например, если мы рассматриваем простую предметную область, описывающую кубики на столе, то *концептуализацией* является набор возможных положений кубиков, а не конкретное их расположение в текущий момент времени.

Более поздней модификацией определения Грубера является такое *определение*: *Онтология* - это *формальная спецификация согласованной концептуализации*. Под *согласованной концептуализацией* подразумевается, что данная *концептуализация* не есть частное мнение, а является общей для некоторой группы людей.

Сформулировано еще достаточно много разных определений онтологии. Например, Никола Гуарино определяет онтологию следующим образом: *Онтология* - это формальная теория, ограничивающая возможные *концептуализации* мира.

Некоторые определения отражают способы, которыми авторы строят и используют онтологии, например: *Онтология* - это иерархически структурированное множество терминов, описывающих предметную область, которое может быть использовано как исходная структура для базы знаний.

Содержание онтологии


Основными компонентами онтологии могут являться:

- классы (или понятия),
- отношения (или свойства, атрибуты),
- функции,
- аксиомы,
- экземпляры (или индивиды).

Классы или **понятия** используются в широком смысле. Понятием может быть любая сущность, о которой может быть дана какая-либо информация. Классы - это абстрактные группы, коллекции или наборы объектов. Они могут включать в себя экземпляры, другие классы, либо же сочетания и того, и другого. Классы в онтологиях обычно организованы в **таксономию** - иерархическую классификацию понятий по отношению включения. Например, классы **Мужчина** и **Женщина** являются подклассами класса **Человек**, который в свою очередь включен в класс **Млекопитающие**.

Отношения представляют тип взаимодействия между понятиями предметной области. Формально *n*-арные отношения определяются как *подмножество* произведения *n* множеств: $R \subseteq C_1 \times C_2 \times \dots \times C_n$. Пример бинарного отношения - отношение **ЧАСТЬ-ЦЕЛОЕ**. Отношения тоже могут быть организованы в таксономию по включению; например, отношения **быть_отцом_для** и **быть_матерью_для** на множестве людей содержатся в отношении **быть_родителем_для**, которое в свою очередь содержится в отношении **быть_предком_для**.

Функции - это специальный случай отношений, в которых *n*-й элемент отношения однозначно определяется *n-1* предшествующими элементами. Формально функции определяются следующим образом: $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$. Примерами функциональных отношений являются отношения

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

быть матерью для на множестве людей, или **цена подержанного автомобиля**, которая вычисляется в зависимости от модели автомобиля, даты изготовления и пробега.

Аксиомы используются, чтобы записать высказывания, которые всегда истинны. Они могут быть включены в онтологию для разных целей, например, для определения комплексных ограничений на значения атрибутов, аргументы отношений, для проверки корректности информации, описанной в онтологии, или для вывода новой информации.

В качестве примера того, что в рамках онтологий понимается под аксиомами, можно привести следующее положение и его формальную *запись* на языке исчисления предикатов первого порядка:

Работник, являющийся руководителем проекта, работает в проекте.

Вводятся переменные **E (работник)** и **P (руководитель проекта)**. Тогда *аксиома* записывается следующим образом:

forall (E,P) Employee(E) and Head-Of-Project(E,P)

=> Works-At-Project(E,P)

Цели создания онтологий

В последние годы разработка онтологий - явное формальное описание терминов *предметной области* и отношений между ними - переходит из мира лабораторий по искусственному интеллекту на рабочие столы экспертов по предметным областям. Во всемирной паутине WWW онтологии стали обычным явлением. Онтологии в сети варьируются от больших таксономий, категоризирующих веб-сайты (как на сайте Yahoo!), до категоризаций продаваемых товаров и их характеристик (как на сайте Amazon.com). Во многих дисциплинах сейчас разрабатываются стандартные онтологии, которые могут использоваться экспертами по предметным областям для совместного использования и аннотирования информации в своей области.

Например, в области медицины созданы большие стандартные, структурированные словари, такие как SNOMED и *семантическая сеть Системы Унифицированного Медицинского Языка (Unified Medical Language System, UMLS)*. Также появляются обширные общецелевые онтологии. Например, *Программа ООН по развитию (the United Nations Development Program)* и компания Dun & Bradstreet объединили усилия для разработки онтологии UNSPSC, которая предоставляет терминологию товаров и услуг (unspsc.org).


Онтология определяет общий словарь для ученых, которым нужно совместно использовать информацию в *предметной области*. Она включает машинно-интерпретируемые формулировки основных понятий *предметной области* и отношения между ними.

Почему возникает потребность в разработке онтологии? Вот некоторые причины, которые ниже будут рассмотрены подробнее:

- для совместного использования людьми или программными агентами общего понимания структуры информации;
- для возможности повторного использования знаний в предметной области;
- для того чтобы сделать допущения в предметной области явными;
- для отделения знаний в предметной области от оперативных знаний;
- для анализа знаний в предметной области.

Совместное использование людьми или программными агентами общего понимания структуры информации является одной из наиболее общих целей разработки онтологий. К примеру, пусть несколько различных веб-сайтов содержат информацию по медицине или предоставляют информацию о платных медицинских услугах, оплачиваемых через *Интернет*. Если эти веб-сайты совместно используют и публикуют одну и ту же базовую онтологию терминов, которыми они все пользуются, то компьютерные агенты могут извлекать информацию из этих различных сайтов и накапливать ее. Агенты могут использовать накопленную информацию для ответов на запросы пользователей или как *входные данные* для других приложений.

Обеспечение возможности использования знаний *предметной области* стало одной из движущих сил недавнего всплеска в изучении онтологий. Например, для моделей многих различных предметных областей необходимо сформулировать понятие времени. Это *представление* включает понятие временных интервалов, моментов времени, относительных мер времени и т.д. Если одна *группа* ученых детально разработает такую онтологию, то другие могут просто повторно

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

использовать ее в своих предметных областях. Кроме того, если нам нужно создать большую онтологию, мы можем интегрировать несколько существующих онтологий, описывающих части большой *предметной области*. Мы также можем повторно использовать основную онтологию, такую как UNSPSC, и расширить ее для описания интересующей нас *предметной области*.

Создание явных допущений в *предметной области*, лежащих в основе реализации, дает возможность легко изменить эти допущения при изменении наших знаний о *предметной области*. Жесткое кодирование предположений о мире на языке программирования приводит к тому, что эти предположения не только сложно найти и понять, но и также сложно изменить, особенно непрограммисту. Кроме того, явные спецификации знаний в *предметной области* полезны для новых пользователей, которые должны узнать значения терминов *предметной области*.

Отделение знаний *предметной области* от оперативных знаний - это еще один вариант общего применения онтологий. Мы можем описать задачу конфигурирования продукта из его компонентов в соответствии с требуемой спецификацией и внедрить программу, которая делает эту конфигурацию независимой от продукта и самих компонентов. После этого мы можем разработать онтологию компонентов и характеристик ЭВМ и применить этот *алгоритм* для конфигурирования нестандартных ЭВМ. Мы также можем использовать тот же *алгоритм* для конфигурирования лифтов, если мы предоставим ему онтологию компонентов лифта.

Анализ знаний в *предметной области* возможен, когда имеется декларативная спецификация терминов. Формальный *анализ* терминов чрезвычайно ценен как при попытке повторного использования существующих онтологий, так и при их расширении.

Часто *онтология предметной области* сама по себе не является целью. Разработка онтологии сродни определению набора данных и их структуры для использования другими программами. Методы решения задач, доменно-независимые приложения и программные агенты используют в качестве данных онтологии и базы знаний, построенные на основе этих онтологий.

Редакторы онтологий

При создании онтологий (как и при проектировании программного обеспечения или написании электронного документа) целесообразно пользоваться подходящими инструментами. Будем называть инструментальные программные средства, созданные специально для проектирования, редактирования и анализа онтологий, редакторами онтологий.

Основная функция любого редактора онтологий состоит в поддержке процесса формализации знаний и представлении онтологии как спецификации (точного и полного описания).

В большинстве своем современные редакторы онтологий предоставляют средства "кодирования" (в смысле "описания") формальной модели в том или ином виде. Некоторые дают дополнительные возможности по анализу онтологии, используют механизм логического вывода.

Поддерживаемые редактором формализмы и форматы представления

Под формализмом понимается теоретический базис, лежащий в основе способа представления онтологических знаний. Примерами формализмов могут служить логика предикатов (First Order Logic - FOL), дескриптивная логика, фреймовые модели (Frames), концептуальные графы и т.п. Формализм, используемый редактором, может не только существенно влиять на внутренние структуры данных, но и определять формат представления или даже пользовательский интерфейс.

Формат представления онтологии задает вид хранения и способ передачи онтологических описаний.

Под форматами подразумеваются языки представления онтологий: RDF, OWL, KIF, SCL.

Таким образом, некоторая формальная модель представляется в формализме FOL и может быть выражена средствами языка KIF.


Редакторы онтологий обычно поддерживают работу с несколькими формализмами и форматами представления, но часто только один формализм является "родным" (native) для данного редактора.

Функциональность редактора онтологий

Важной характеристикой является функциональность редактора, т.е. множество сценариев его использования.

Базовый набор функций обеспечивает:

- работу с одним или более проектами;
- сохранение проекта в нужном формализме и формате (экспорт);

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- открытие проекта;
- импорт из внешнего формата;
- редактирование метаданных проекта (в широком смысле: от настройки форм редактирования и представления данных до поддержки версий проекта);
- редактирование онтологии. Набор возможных действий обычно включает создание, редактирование, удаление понятий, отношений, аксиом и прочих структурных элементов онтологии, редактирование таксономии.

К дополнительным возможностям редакторов относят поддержку языка запросов (для поиска нетривиальных утверждений), анализ целостности, использование механизма логического вывода, поддержку многопользовательского режима, поддержку удаленного доступа через Интернет.

Сложные инструментальные средства

Эти средства нужны для того, чтобы не только вводить и редактировать онтологическую информацию, но и анализировать ее, выполняя типичные операции над онтологиями, например:

- выравнивание (alignment) онтологий - установка различного вида соответствий между двумя онтологиями для того, чтобы они могли использовать информацию друг друга;
- отображение (mapping) одной онтологии на другую - нахождение семантических связей между подобными элементами разных онтологий;
- объединение (merging) онтологий - операция, которая по двум онтологиям генерирует третью, объединяющую информацию из первых двух.

Редактор Protege

С момента его создания Protege многие годы использовался экспертами в основном для концептуального моделирования в области медицины. В последнее время его стали применять в других предметных областях - в частности, при создании онтологий для Semantic Web.

Используемые формализмы и форматы

Изначально единственной моделью знаний, поддерживаемой Protege, была фреймовая модель. Этот формализм сейчас является "родным" для редактора, но не единственным.

Protege имеет открытую, легко расширяемую архитектуру и помимо фреймов поддерживает все наиболее распространенные языки представления знаний (SHOE, XOL, DAML+OIL, RDF/RDFS, OWL). Protege поддерживает модули расширения функциональности (plug-in). Расширить Protege для использования нового языка проще, чем создавать редактор этого языка "с нуля".

Protege основан на модели представления знаний ОКВС (Open Knowledge Base Connectivity). Основными элементами являются классы, экземпляры, слоты (представляющие свойства классов и экземпляров) и фасеты (задающие дополнительную информацию о слотах).

Пользовательский интерфейс

Пользовательский интерфейс состоит из главного меню и нескольких вкладок для редактирования различных частей базы знаний и ее структуры. Набор и названия вкладок зависят от типа проекта (языка представления) и могут быть настроены вручную. Обычно имеются следующие основные вкладки: Классы, Слоты (или Свойства для OWL), Экземпляры, Метаданные.

Назначение основных вкладок - предоставить набор форм для заполнения базы знаний.

Вкладка "Классы"

Функции: создание классов, слотов для данного класса, отображение иерархии классов, добавление текстовых примечаний к классам, поиск класса по шаблону.

Вкладка "Слоты"


Функции: создание слотов, назначение домена и диапазона для данного слота, отображение иерархии и свойств слотов, добавление текстовых описаний слотов, поиск слота по шаблону, задание ограничений на значения слота.

Вкладка "Экземпляры"

Функции: создание экземпляров данного класса, отображение и редактирование свойств экземпляра, отображение иерархии классов, связывание экземпляров слотами, добавление текстовых описаний экземпляров, поиск слота по шаблону, задание ограничений на значения слота.

Практическая разработка онтологии включает:

определение классов в онтологии;

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- расположение классов в таксономическую иерархию (ПОДКЛАСС-НАДКЛАСС);
- определение слотов и описание их допустимых значений;
- заполнение значений слотов экземпляров.

После этого можно создать базу знаний, определив отдельные экземпляры этих классов, введя в определенный слот значение и дополнительные ограничения для слота.

Выделим некоторые фундаментальные правила разработки онтологии. Они выглядят довольно категоричными, но во многих случаях помогут принять верные проектные решения.

Не существует единственно правильного способа моделирования предметной области - всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.

Разработка онтологии - это обязательно итеративный процесс.

Понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в интересующей предметной области. Скорее всего, это существительные (объекты) или глаголы (отношения) в предложениях, которые описывают предметную область.

Знание того, для чего предполагается использовать онтологию, и того, насколько детальной или общей она будет, может повлиять на многие решения, касающиеся моделирования. Нужно определить, какая из альтернатив поможет лучше решить поставленную задачу и будет более наглядной, более расширяемой и более простой в обслуживании. Следует помнить, что онтология - это модель реального мира, и понятия в онтологии должны отражать эту реальность.

После того как определена начальная версия онтологии, мы можем оценить и отладить ее, используя ее в каких-то приложениях и/или обсудив ее с экспертами предметной области. В результате начальную онтологию скорее всего нужно будет пересмотреть. И этот процесс итеративного проектирования будет продолжаться в течение всего жизненного цикла онтологии.

Повторное использование существующих онтологий может быть необходимым, если системе нужно взаимодействовать с другими приложениями, которые уже вошли в отдельные онтологии или контролируемые словари. Многие полезные онтологии уже доступны в электронном виде и могут быть импортированы. Существуют библиотеки повторно используемых онтологий, например, Ontolingua или DAML.

Задания на самостоятельную работу

1. Создание онтологию с помощью редактора Protege

Спроектировать онтологию для выбранной предметной области:


Предпочтительно, чтобы цель ИС (решаемая интеллектуальная задача) соответствовала предметной области из списка:

1. Разработка ПО (ИС)
2. Анализ требований к ПО
3. Проектирование архитектуры ПО
4. Программная реализация ПО
5. Тестирование и отладка ПО
6. Внедрение и поддержка (сопровождение) ПО
7. Проектирование компьютерных интерфейсов
8. CASE-средства
9. Контроль качества ПО
10. Моделирование бизнес-процессов

По согласованию с преподавателем, студенты могут предложить свой вариант

Создать онтологию в редакторе Protégé-Frames:

В соответствии с описанным выше процессом создания онтологии, реализуйте в Protégé-Frames прикладную онтологию задач для выбранной предметной области и

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

цели создания ИС. Создаваемая онтология должна содержать не менее 50 основных фреймов (классов, слотов) и не менее 20 фреймов-экземпляров.

2. Ответьте на следующие вопросы:

1. Поясните, каким образом цель создания онтологии определяет её структуру?
2. В чём вы видите отличие онтологии от базы знаний и от интеллектуальной (экспертной) системы?
3. Какие модели представления знаний (помимо фреймов) вы знаете?
4. Какие типичные отношения между концептами в онтологиях вы знаете?
5. В чём отличие иерархических и ассоциативных связей между концептами?
6. Объясните зачем необходимо создавать онтологии?
7. Перечислите из чего состоит онтология?
8. Опишите методологию инженерии знаний
9. Дайте определение классу и понятию в онтологии
10. Опишите назначение иерархии классов в онтологиях
11. Объясните для чего применяются свойства в онтологиях
12. Перечислите рекомендации именования понятий в онтологиях
13. Укажите какие есть ресурсы для создания онтологий помимо Protege.
14. Опишите как применяются онтологии при решении прикладных задач.

Тема 3. Прикладные системы искусственного интеллекта

Практическое занятие. Анализ и выбор соответствующих инструментов и средств, основанных на современных интеллектуальных технологиях для решения конкретной профессиональной задачи.

Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.

Форма проведения занятия – практическое занятие.

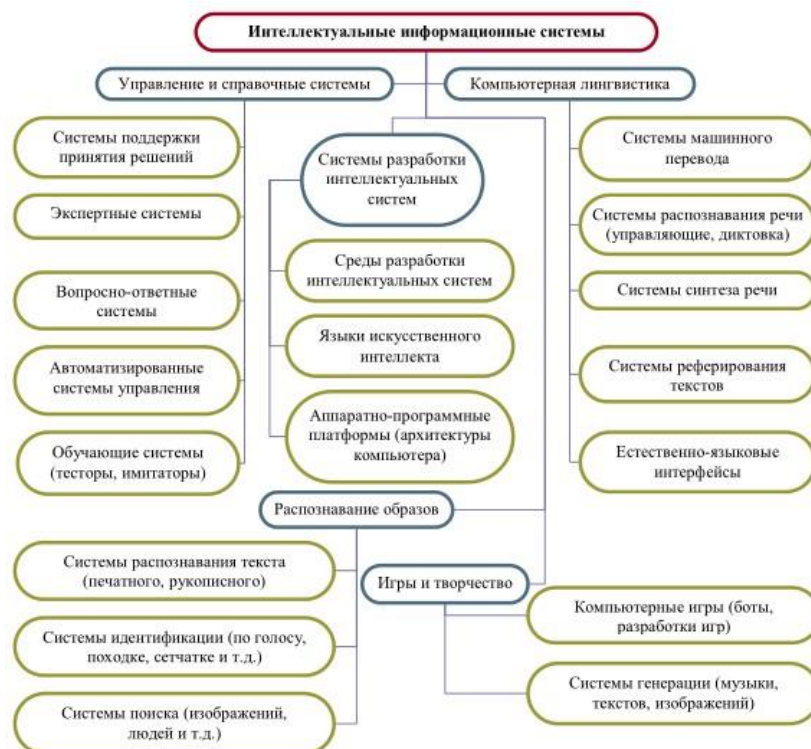
Вопросы для самоконтроля и текущего контроля

1. Искусственный интеллект (AI), интеллектуальный анализ (DM) и машинное обучение (ML).
2. Модель искусственного нейрона.
3. Обучение искусственных нейронных сетей.
4. Алгоритм обратного распространения ошибки.


Классификация интеллектуальных информационных систем по типам систем



Классификация интеллектуальных информационных систем по решаемым задачам



Классификация интеллектуальных информационных систем по методам

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		



Классификация интеллектуальных систем по назначению



Задания на самостоятельную работу


1. Ответьте на следующие вопросы:

1. Перечислите направления применения современных интеллектуальных систем.
2. Дайте определение интеллектуальной информационной системы.
3. К каким интеллектуальным системам относится система, использующая генетические вычисления и базы данных?
4. Какие системы являются системами общего назначения?
5. Какие системы относятся к самоорганизующимся системам?
6. Какие системы относятся к системам компьютерной лингвистики?
7. Перечислите критерии оценки интеллектуальных систем.

Задание для самостоятельной работы:

- Проработать лекционный материал, основную и дополнительную литературу с целью подготовки ответов на представленный перечень вопросов для устного опроса.

Тема 3. Прикладные системы искусственного интеллекта

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Практическое занятие. Решение профессиональной задачи путём использования современных интеллектуальных технологий.

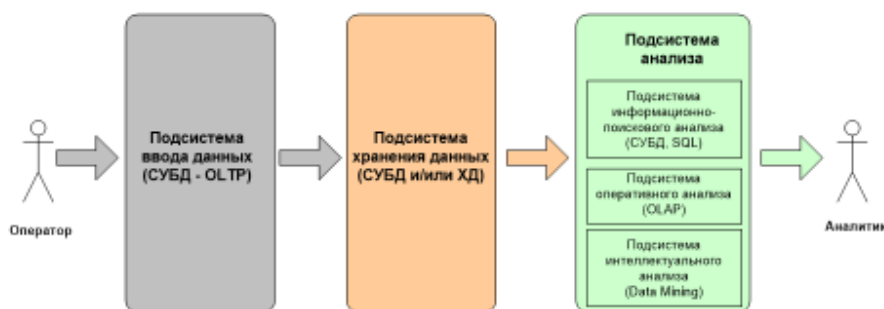
Изучение нижеперечисленных вопросов будет производиться на примере конкретных ситуаций, с целью формирования и развития профессиональных навыков обучающихся.

Форма проведения занятия – практическое занятие.

Вопросы для самоконтроля и текущего контроля

1. Инструментальные среды для моделирования алгоритмов искусственного интеллекта.
2. Классификации СППР.
3. Аналитическая low-code платформа Loginom.
4. Функции Loginom.
5. Решение практических задач.
6. Многомерный анализ данных (OLAP).
7. Выявление скрытых закономерностей (Data Mining).

Архитектура СППР




Задания на самостоятельную работу

1. Выполните аппроксимацию данных временных рядов с помощью low-code платформы Loginom.

Задача аппроксимации данных временных рядов заключается в поиске аномальных значений относительно временного ряда и их сглаживание. Решение данной задачи было реализовано двумя методами: при помощи компонента Редактирование выбросов и с помощью компонента ARIMAX.

Модели ARIMA применяются для решения задач, в которых требуется построить прогноз на основе имеющихся данных, то есть вычислить последующие значения ряда на основе предыдущих. Временным рядом могут быть любые данные в разрезе времени, например, продажи товаров, количество заказов, поток клиентов, значения окружающей температуры и т.д. Кроме того, на основе построения прогноза автоматически корректируются аномальные значения, соответственно, прогноз — это выборка текущих значений с учетом корректировки выбросов.

2. Выполните оценку стоимости недвижимости (Нейросеть) с помощью low-code платформы Loginom.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Особенностью процесса оценки стоимости недвижимости является его рыночный характер. Данный процесс не ограничивается учетом одних только затрат на создание или приобретение оцениваемого объекта собственности — необходим учет совокупности рыночных факторов, экономических особенностей оцениваемого объекта, а также макро- и микроэкономического окружения. Кроме того, рынок недвижимости очень динамичный, поэтому требуется периодическая переоценка объектов собственности. Создание моделей на основе искусственных нейронных сетей для оценки стоимости недвижимости может существенно повысить эффективность работы организаций, занимающихся риэлтерской деятельностью.

3. Выполните Анализ рыночной корзины на примере розничной сети, занимающейся продажей бытовой химии.

Анализ рыночной корзины в Data Mining — это поиск устойчивых групп событий, происходящих совместно, в некоторой предметной области. В его основе лежит поиск ассоциативных правил, позволяющий находить закономерности между связанными событиями.


Анализ рыночной корзины используется в розничной торговле для выявления устойчивых наборов товаров, приобретаемых клиентами в супермаркете по одному чеку (например, «картошка, лук и салат», «макаронны и кетчуп», «пиво и чипсы», «чай и выпечка» и т.д.), что позволяет оптимизировать товарный ассортимент и размещение товаров в торговом зале с целью «спровоцировать» покупателя на покупку нескольких товаров из набора.

Метод успешно используется и в других областях, например, для анализа посещений веб-страниц, в анализе и прогнозировании сбоев телекоммуникационного оборудования, в медицине и т.д.

Согласуйте исходные данные с преподавателем.

4. Ответьте на следующие вопросы:

1. Определите понятие «Система поддержки принятия решений».
2. Основные компоненты СППР. Какие подсистемы входят в системы поддержки принятия решений?
3. Как можно классифицировать системы поддержки принятия решений?
4. Каковы функции систем поддержки принятия решений в оценке существующих и гипотетических ситуаций, в которых функционирует предприятие?
5. Какие системы поддержки принятия решений позволяют модифицировать решения системы, опирающиеся на большие объемы данных из разных источников?
6. Какие бывают архитектуры систем поддержки принятия решений?
7. Укажите принципы загрузки, верификации и очистки данных.
8. Определите основные причины низкого качества данных в СППР.
9. Укажите методы и средства повышения качества исходных данных.
10. Охарактеризуйте возможные условия, в которых менеджеру приходится принимать решения: определенность, риск, неопределенность.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

Задание для самостоятельной работы:

- *Проработать лекционный материал, основную и дополнительную литературу с целью подготовки ответов на представленный перечень вопросов для устного опроса.*

Практические (семинарские занятия) представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают основные разделы.

Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях. В обязанности преподавателя входят: оказание методической помощи и консультирование обучающихся по соответствующим темам курса.

На семинаре каждый его участник должен быть готовым к выступлению по всем поставленным в плане вопросам, проявлять максимальную активность при их рассмотрении. Выступление должно строиться свободно, убедительно и аргументировано. Преподаватель следит, чтобы выступление не сводилось к репродуктивному уровню (простому воспроизведению текста), не допускается и простое чтение конспекта. Необходимо, чтобы выступающий проявлял собственное отношение к тому, о чем он говорит, высказывал свое личное мнение, понимание, обосновывал его и мог сделать правильные выводы из сказанного. При этом студент может обращаться к записям конспекта и лекций, непосредственно к первоисточникам, использовать знание художественной литературы и искусства, факты и наблюдения современной жизни и т. д. Вокруг такого выступления могут разгореться споры, дискуссии, к участию в которых должен стремиться каждый. Преподавателю необходимо внимательно и критически слушать, подмечать особенное в суждениях студентов, улавливать недостатки и ошибки, корректировать их знания, и, если нужно, выступить в роли рефери, обратить внимание на то, что еще не было сказано, или поддержать и развить интересную мысль, высказанную выступающим студентом. В заключение преподаватель, как руководитель семинара, подводит итоги семинара. Он может (выборочно) проверить конспекты студентов и, если потребуется, внести в них исправления и дополнения.

Активность на практических занятиях оценивается по следующим критериям:

- ответы на вопросы, предлагаемые преподавателем;
- участие в дискуссиях;
- выполнение проектных и иных заданий;
- ассистирование преподавателю в проведении занятий.

Доклады и оппонирование докладов проверяют степень владения теоретическим материалом, а также корректность и строгость рассуждений.


7. ЛАБОРАТОРНЫЕ РАБОТЫ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)

Не предусмотрены учебным планом.

8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ

Курсовые и контрольные работы не предусмотрены учебным планом.

Реферат это одна из форм текущего контроля знаний и контроля самостоятельной работы. Реферат – это самостоятельная исследовательская работа, в которой автор раскрывает суть исследуемой проблемы; приводит различные точки зрения, а также


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

собственные взгляды не нее. Содержание реферата должно быть логичным; изложение материала должно носить проблемно-тематический характер.

Цель реферата как формы текущего контроля знаний и контроля самостоятельной работы - стимулировать раскрытие исследовательского потенциала учащегося, способность к творческому поиску, сотрудничеству, самораскрытию и проявлению возможностей.

Примерная тематика рефератов:

№ задания	Тема
1	Системы автоматизации проектных работ (САПР).
2	Экспертные системы, их применение для решения задач различных предметных областей.
3	Системы искусственного интеллекта, классификация, особенности.
4	Роль автоматизированных систем поддержки принятия решений в управлении экономическими объектами.
5	Области применения нейронных сетей, классы задач, решаемых благодаря их использованию.
6	Формализация и структурирование знаний при проектировании баз знаний. Модели знаний.
7	Автоматизированные информационные технологии и системы для интеллектуальной поддержки финансового управления и проведения финансового анализа состояния предприятия.
8	Назначение и области применения правовых информационно – поисковых справочных систем.
9	Электронные программы – словари.
10	Программы перевода текстов с одних языков на другие.
11	Инструментальные средства и языки программирования, применяемые для разработки систем искусственного интеллекта.
12	Общая характеристика классов задач, решаемых с помощью систем искусственного интеллекта.
13	Общая характеристика и основные компоненты автоматизированных систем поддержки принятия решений модельного типа.
14	Гипертекстовые поисковые Internet – системы.
15	Интеллектуальные обучающие программы.
16	Основные понятия теории предикатов, её использование для представления знаний.
17	Нечёткие множества, операции над ними. Использование нечётких выводов в экспертных системах.
18	Определение и методы построения когнитивных карт. Принятие решений с помощью когнитивных карт.
19	Применение автоматизированных систем поддержки принятия решений модельного типа в управлении предприятиями.
20	Применение систем искусственного интеллекта для статистического анализа данных и прогнозирования поведения объектов и систем.
21	Примеры использования генетических алгоритмов.
22	Информационные хранилища: принципы построения, основные компоненты.
23	CASE – технологии: назначение, примеры.
24	Классификация систем искусственного интеллекта.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

25	Контекстные системы поиска: назначение, примеры.
----	--

Формулировки приведенных выше тем являются примерными и могут быть изменены. Изменения согласуются с преподавателем, ведущим дисциплину. Кроме этого, обучающиеся могут предлагать собственные темы для исследования. Инициативные темы также согласуются с преподавателем.


В процессе изучения курса каждый должен подготовить реферат, который будет засчитан преподавателем, ведущим дисциплину.

Оценивая реферат, преподаватель обращает внимание на:

- соответствие содержания выбранной теме;
- отсутствие в тексте отступлений от темы;
- соблюдение структуры работы, четкость изложения и обоснованность выводов;
- умение работать с научной литературой - вычленять проблему из контекста;
- умение логически мыслить;
- культуру письменной речи;
- умение оформлять научный текст (правильное применение и оформление ссылок, составление библиографии и т.д.);
- умение правильно понять позицию авторов, работы которых использовались при написании реферата;
- способность верно, без искажения передать используемый авторский материал;
- соблюдение объема работы;
- соответствие установленным правилам оформления работы;
- аккуратность и правильность технического выполнения работы.

9. ПЕРЕЧЕНЬ ВОПРОСОВ К ДИФФЕРЕНЦИРОВАННОМУ ЗАЧЕТУ


1. Что такое знания, какова их природа?
2. В чем отличие процедурных знаний от декларативных?
3. Программный инструментарий разработки систем, основанных на знаниях.
4. Данные и знания.
5. Модели представления данных и знаний.
6. Языки символьной обработки и языки программирования для ИИ.
7. Формальные модели.
8. Модели вывода на знаниях.
9. Продукционные системы.
10. Язык расширенных сетей переходов ATNL – основные понятия и приемы программирования.
11. Онтологические модели представления знаний.
12. Введение в разработку систем, основанных на знаниях.
13. Введение в мультиагентные системы.
14. Технологии инженерии знаний.
15. Состояние и перспективы автоматизированного приобретения знаний.
16. Системы и средства представления онтологических знаний. Основные подходы.
17. Примеры задач машинного обучения: задачи классификации.
18. Примеры задач машинного обучения: задачи регрессии.
19. Примеры задач машинного обучения: задача ранжирования.
20. Решатель задач. Система обучения. База данных. База знаний.
21. Система объяснения. Система доверия. Блок обоснования.
22. Программы решения интеллектуальных задач. Игровые программы.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

23. Обработка Естественного Языка на ЭВМ. Основные области применения.
24. Интерфейс на естественном языке в интеллектуальных системах: основные требования к процессу понимания запросов. Общая схема анализа высказывания.
25. Лексическое значение слова и его описание средствами лингвистических информационных ресурсов. Фреймовое представление ситуации действительности и модель управления предикатного слова: сравнительный анализ.
26. Интерфейс на естественном языке: этап синтаксического анализа входного предложения.
27. Основные принципы построения правил и стратегий синтаксического анализа фраз естественного языка для задач компьютерной обработки текстов.
28. Эвристическое программирование. Методы поиска.
29. Представление знаний. Модели представления знаний. Их классификация.
30. Представление знаний правилами. Структура продукционной системы.
31. Процедурное представление знаний. Продукции. Деревья «И-ИЛИ». Деревья вывода.
32. Представление задач в пространстве состояний. Состояния и операторы. Представление операторов системой продукций.
33. Логические модели представления знаний. Формальная система. Интерпретация и свойства формальных систем.
34. Исчисление высказываний как формальная система. Исчисление предикатов как формальная система. Логические следствия.
35. Экспертные системы. Продукционные экспертные системы.
36. Структура экспертной системы. База знаний. Машина вывода.
37. Экспертные системы. Общая характеристика, структура и основные элементы экспертных систем.
38. Экспертные системы. Интеллектуальные информационные ЭС. Понятие о ИАД.
39. Экспертные системы. Классификация ЭС по назначению. Основные направления приложения ЭС. Классификация ЭС по методам построения.

10. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩИХСЯ

Самостоятельная работа студентов является особой формой организации учебного процесса, представляющая собой планируемую, познавательную, организационно и методически направляемую деятельность студентов, ориентированную на достижение конкретного результата, осуществляемую без прямой помощи преподавателя. Самостоятельная работа студентов является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, а также выполнение учебных заданий, подготовку к предстоящим занятиям и зачету. Она предусматривает, как правило, разработку рефератов, написание докладов, выполнение творческих, индивидуальных заданий в соответствии с учебной программой (тематическим планом изучения дисциплины). Тема для такого выступления может быть предложена преподавателем или избрана самим студентом, но материал выступления не должен дублировать лекционный материал. Реферативный материал служит дополнительной информацией для работы на практических занятиях. Основная цель данного вида работы состоит в обучении студентов методам самостоятельной работы с учебным материалом. Для полноты усвоения тем, вынесенных в практические занятия, требуется работа с первоисточниками. Курс предусматривает самостоятельную работу студентов со специальной литературой. Следует отметить, что самостоятельная работа студентов результативна лишь тогда, когда она выполняется систематически, планомерно

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


и целенаправленно. Задания для самостоятельной работы предусматривают использование необходимых терминов и понятий по проблематике курса. Они нацеливают на практическую работу по применению изучаемого материала, поиск библиографического материала и электронных источников информации, иллюстративных материалов. Задания по самостоятельной работе даются по темам, которые требуют дополнительной проработки. Общий объем самостоятельной работы студентов по дисциплине включает аудиторную и внеаудиторную самостоятельную работу студентов в течение семестра. Аудиторная самостоятельная работа осуществляется в форме выполнения тестовых заданий, кейс-задач, письменных проверочных работ по дисциплине. Аудиторная самостоятельная работа обеспечена базой тестовых материалов, кейс-задач по разделам дисциплины.

Формирование внутренней потребности к самообучению становится и требованием времени, и условием реализации личностного потенциала. Способность человека состояться на уровне, адекватном его претензиям на положение в обществе, всецело зависит от его индивидуальной вовлеченности в самостоятельный процесс освоения новых знаний. Поэтому одной из целей профессиональной подготовки специалиста является необходимость дать студенту прочные фундаментальные знания, на основе которых он смог бы обучаться самостоятельно в нужном ему направлении. Решение задач современного образования невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание их творческой активности и инициативы. Методологическую основу самостоятельной работы студентов составляет деятельностный подход, который состоит в том, что цели обучения ориентированы на формирование умений решать типовые и нетиповые задачи, т. е. на реальные ситуации, где студентам надо проявить знание конкретной дисциплины.

Основная задача организации СРС заключается в создании психолого-дидактических условий развития интеллектуальной инициативы и мышления на занятиях любой формы. Основным принципом организации СРС должен стать перевод всех студентов на индивидуальную работу с переходом от формального пассивного выполнения определенных заданий к познавательной активности с формированием собственного мнения при решении поставленных проблемных вопросов и задач. Таким образом, в результате самостоятельной работы студент должен научиться осмысленно и самостоятельно работать сначала с учебным материалом, затем с научной информацией, использовать основы самоорганизации и самовоспитания с тем, чтобы развивать в дальнейшем умение непрерывно повышать свою квалификацию.

Виды самостоятельной работы студентов, обеспечивающие реализацию цели и решение задач данной рабочей программы:

- подготовка к практическим занятиям;
- изучение тем дисциплины, выносимых для самостоятельного изучения;
- подготовка реферата;
- подготовка к тестированию;
- выполнение самостоятельных практических, работ на занятиях;
- работа со справочной, методической и научной литературой;
- разбор конкретных ситуаций, в том числе углубляющих теоретические знания;
- участие в дискуссиях;
- работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио- и видеозаписей);

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

- подготовка к проблемным семинарам, дискуссионным вопросам, «круглым столам» и др.;
- изучение отдельных тем (вопросов) учебных дисциплин, составление плана и тезисов ответа;
- ответы на контрольные вопросы;
- подготовка тезисов сообщений к выступлению на семинаре;
- экспресс-опросы по конкретным темам;
- подготовка к промежуточной аттестации.

Обучающиеся выполняют задания, самостоятельно обращаясь к учебной литературе. Проверка выполнения заданий осуществляется путем электронного тестирования и устного опроса на практических занятиях.


Материалы курса, выносимые студентам для самостоятельного изучения:

Форма обучения очная

Название разделов и тем	Вид самостоятельной работы	Объем в часах	Форма контроля
Тема 1. Введение в прикладные системы искусственного интеллекта.	<ul style="list-style-type: none"> – Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины; – Подготовка к тестированию; – Подготовка рефератов; – Подготовка к сдаче диф.зачета. 	30	проверка тестовых заданий, устный опрос, проверка кейсов
Тема 2. Системы, основанные на знаниях	<ul style="list-style-type: none"> – Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины; – Подготовка к тестированию; – Подготовка рефератов; – Подготовка к сдаче диф.зачета. 	32	проверка тестовых заданий, устный опрос, проверка кейсов, реферат
Тема 3. Прикладные системы искусственного интеллекта	<ul style="list-style-type: none"> – Проработка учебного материала с использованием ресурсов учебно-методического и информационного обеспечения дисциплины; – Подготовка к тестированию; – Подготовка рефератов; – Подготовка к сдаче диф.зачета. 	64	проверка тестовых заданий, устный опрос, проверка кейсов, реферат

Методические указания для обучающихся по освоению дисциплины

Для качественного усвоения обучающимися материала курса при выполнении ими индивидуальных заданий необходимо, чтобы все работы выполнялись студентами после проработки соответствующего материала. Основная задача по организации учебного процесса по данной дисциплине сводится к обеспечению равномерной активной работы обучающихся над курсом в течение всего учебного семестра. Обучающиеся должны регулярно прорабатывать курс пройденных семинаров, готовиться к занятиям. Для контроля качества усвоения учебного материала следует проводить опросы по изученной теме. Для

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

долговременного запоминания изученного материала следует увязывать вновь изучаемые вопросы с материалом предыдущих тем, добиваться преемственности знаний.

При выполнении заданий, вынесенных на самостоятельное изучение, необходимо наряду с библиотечным фондом пользоваться различными источниками знаний, размещенными в сети Интернет.

При изучении данного курса обучающимся предстоит выполнить следующие основные виды работ:

- Анализ теоретического материала;
- Проработка материала семинарских занятий;
- Выполнение практических заданий;
- Подготовка к семинару;
- Подготовка к тестированию.

Семинарские занятия желательно проводить с применением демонстрационного материала – презентации на ПК с проектором. С учетом современных возможностей, желательно обеспечивать слушателей раздаточным материалом на 1-2 занятия вперед. Материал этот должен носить иллюстративный характер (схемы, формулы, графики) и ни в коем случае не подменять конспекта, который слушатель должен составлять самостоятельно.

Текущий контроль

Для текущего контроля успеваемости (по отдельным разделам дисциплины) и промежуточной аттестации используется компьютерное тестирование, проверка реферата.

Методические указания для обучающихся по освоению отдельных дидактических единиц дисциплины

1. Планирование и организация времени, необходимого для самостоятельного изучения дисциплины.


Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

- Изучение конспекта семинара в тот же день, после занятия: 1-2 часа.
- Подготовка к семинарскому занятию: 1-3 часа.
- Изучение теоретического материала по учебнику и конспекту (включая дополнительные источники, в том числе, в электронной форме): 3-4 часа в неделю.
- Подготовка к тестированию по текущей теме: 1-3 часа.
- Всего в неделю: 6–9 часов.

2. Методические рекомендации по подготовке к практическим занятиям.

По данному курсу предусмотрены практические занятия. При подготовке следует изучить соответствующий теоретический материал по цифровой экономике, электронной коммерции, электронному бизнесу или электронным платежным системам. Теоретический материал курса становится более понятным, когда дополнительно к обучению на семинарах и изучению конспекта, изучаются и книги по современным информационным технологиям.

Подготовка к практическому занятию включает 2 этапа: 1й - организационный; 2й - закрепление и углубление теоретических знаний. На первом этапе обучающийся планирует свою самостоятельную работу, которая включает: - уяснение задания на самостоятельную работу; - подбор рекомендованной литературы; - составление плана работы, в котором определяются основные пункты предстоящей подготовки. Составление плана дисциплинирует и повышает организованность в работе. Второй этап включает непосредственную подготовку студента к занятию. Начинать надо с изучения рекомендованной литературы. Необходимо помнить, что на семинаре обычно рассматривается не весь материал, а только его часть. Остальная его часть восполняется в процессе самостоятельной работы. В связи с этим работа с рекомендованной литературой

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		


обязательна. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы обучающийся должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. Заканчивать подготовку следует составлением плана (конспекта) по изучаемому материалу (вопросу). Это позволяет составить концентрированное, сжатое представление по изучаемым вопросам. В процессе подготовки к занятиям рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь. При необходимости следует обращаться за консультацией к преподавателю. Идя на консультацию, необходимо хорошо продумать вопросы, которые требуют разъяснения. Записи имеют первостепенное значение для самостоятельной работы обучающихся. Они помогают понять построение изучаемого материала, выделить основные положения, проследить их логику и тем самым проникнуть в творческую лабораторию автора. Ведение записей способствует превращению чтения в активный процесс, мобилизует, наряду со зрительной, и моторную память. Следует помнить: у обучающегося, систематически ведущего записи, создается свой индивидуальный фонд подсобных материалов для быстрого повторения прочитанного, для мобилизации накопленных знаний. Особенно важны и полезны записи тогда, когда в них находят отражение мысли, возникшие при самостоятельной работе. Важно развивать у обучающихся умение сопоставлять источники, продумывать изучаемый материал. Большое значение имеет совершенствование навыков конспектирования у обучающихся. Преподаватель может рекомендовать студентам следующие основные формы записи: план (простой и развернутый), выписки, тезисы. Результаты конспектирования могут быть представлены в различных формах. План - это схема прочитанного материала, краткий (или подробный) перечень вопросов, отражающих структуру и последовательность материала. Подробно составленный план вполне заменяет конспект. Конспект - это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов:

- План-конспект - это развернутый детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении.
- Текстуальный конспект - это воспроизведение наиболее важных положений и фактов источника.
- Свободный конспект - это четко и кратко сформулированные (изложенные) основные положения в результате глубокого осмысливания материала. В нем могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом.
- Тематический конспект - составляется на основе изучения ряда источников и дает более или менее исчерпывающий ответ по какой-то схеме (вопросу).


3. Групповая консультация

Разъяснение является основным содержанием данной формы занятий, наиболее сложных вопросов изучаемого программного материала. Цель - максимальное приближение обучения к практическим интересам с учетом имеющейся информации и является результативным материалом закрепления знаний. Групповая консультация проводится в следующих случаях:

- когда необходимо подробно рассмотреть практические вопросы, которые были недостаточно освещены или совсем не освещены в процессе лекции;
- с целью оказания помощи в самостоятельной работе (написание рефератов, выполнение курсовых работ, сдача экзаменов (зачетов), подготовка конференций);

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

— если обучающиеся самостоятельно изучают нормативный, справочный материал, инструкции, положения.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

11. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) Список рекомендуемой литературы

Основная:

1. Основы цифровой экономики : учебник и практикум для вузов / М. Н. Конягина [и др.] ; ответственный редактор М. Н. Конягина. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-13476-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519464>.

2. Бессмертный, И. А. Интеллектуальные системы : учебник и практикум для вузов / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. — Москва : Издательство Юрайт, 2023. — 243 с. — (Высшее образование). — ISBN 978-5-534-01042-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511999>

Дополнительная

1. Болотова, Л. С. Системы поддержки принятия решений в 2 ч. Часть 1 : учебник и практикум для вузов / Л. С. Болотова ; ответственные редакторы В. Н. Волкова, Э. С. Болотов. — Москва : Издательство Юрайт, 2023. — 257 с. — (Высшее образование). — ISBN 978-5-9916-8250-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512250>

2. Болотова, Л. С. Системы поддержки принятия решений в 2 ч. Часть 2 : учебник и практикум для вузов / Л. С. Болотова ; ответственные редакторы В. Н. Волкова, Э. С. Болотов. — Москва : Издательство Юрайт, 2023. — 250 с. — (Высшее образование). — ISBN 978-5-9916-8251-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513142>

Учебно-методическая литература:


1. Сквоиков А. Г. Прикладные системы искусственного интеллекта : методические рекомендации для самостоятельной работы студентов направления 38.04.01 «Экономика». Профиль – искусственный интеллект в финансово-экономических системах / А. Г. Сквоиков ; Ульян. гос. ун-т, Ин-т экономики и бизнеса. - 2023. - 83 с. - Неопубликованный ресурс. - URL: <http://lib.ulsu.ru/MegaPro/Download/MObject/15143>. - Режим доступа: ЭБС УлГУ. - Текст : электронный. URL: http://lib.ulsu.ru/MegaPro/UserEntry?Action=Link_FindDoc&id=511372&idb=0

Согласовано:

С.А.В. Сидлисткарь / Голосова М.И. / М.И. 10.06.2023г.
 Должность сотрудника научной библиотеки ФИО подпись дата

б) Программное обеспечение

- Windows;
- Office;
- МойОфис Стандартный;
- Low-code платформа для реализации аналитических процессов Loginom;
- Среда построения экспертных систем CLIPS 6.3;
- Редактор онтологий Protege 5.6.2.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

в) Профессиональные базы данных, информационно-справочные системы

1. Электронно-библиотечные системы:

1.1. Цифровой образовательный ресурс IPRsmart : электронно-библиотечная система : сайт / ООО Компания «Ай Пи Ар Медиа». - Саратов, [2023]. – URL: <http://www.iprbookshop.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.2. Образовательная платформа ЮРАЙТ : образовательный ресурс, электронная библиотека : сайт / ООО Электронное издательство «ЮРАЙТ». – Москва, [2023]. - URL: <https://urait.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.3. База данных «Электронная библиотека технического ВУЗа (ЭБС «Консультант студента») : электронно-библиотечная система : сайт / ООО «Политехресурс». – Москва, [2023]. – URL: <https://www.studentlibrary.ru/cgi-bin/mb4x>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. ЭБС Лань : электронно-библиотечная система : сайт / ООО ЭБС «Лань». – Санкт-Петербург, [2023]. – URL: <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. ЭБС Znanium.com : электронно-библиотечная система : сайт / ООО «Знаниум». - Москва, [2023]. - URL: <http://znanium.com>. – Режим доступа : для зарегистрир. пользователей. - Текст : электронный.

2. КонсультантПлюс [Электронный ресурс]: справочная правовая система. / ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2023].

3. Базы данных периодических изданий:

3.1. eLIBRARY.RU: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2023]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

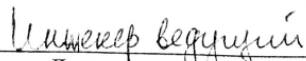
3.2. Электронная библиотека «Издательского дома «Гребенников» (Grebinnikon) : электронная библиотека / ООО ИД «Гребенников». – Москва, [2023]. – URL: <https://id2.action-media.ru/Personal/Products>. – Режим доступа : для авториз. пользователей. – Текст : электронный.

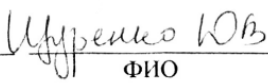
4. Федеральная государственная информационная система «Национальная электронная библиотека» : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2023]. – URL: <https://нэб.рф>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

5. Российское образование : федеральный портал / учредитель ФГАУ «ФИЦТО». – URL: <http://www.edu.ru>. – Текст : электронный.

6. Электронная библиотечная система УлГУ : модуль «Электронная библиотека» АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

СОГЛАСОВАНО:


Должность сотрудника УИТиТ



ФИО


подпись

01.06.2023
дата

12 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для проведения занятий лекционного типа, занятий практического/семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации представляют собой учебные аудитории,

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа дисциплины		

оснащенные оборудованием и техническими средствами обучения. Аудитории укомплектованы специализированной мебелью и техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.

13 СПЕЦИАЛЬНЫЕ УСЛОВИЯ ДЛЯ ОБУЧАЮЩИХСЯ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

– для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации;

– для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий, организация работы ППС с обучающимися с ОВЗ и инвалидами предусматривается в электронной информационно-образовательной среде с учетом их индивидуальных психофизических особенностей

Разработчик



доцент кафедры ЦЭ А.Г. Сквиков

07.06.23