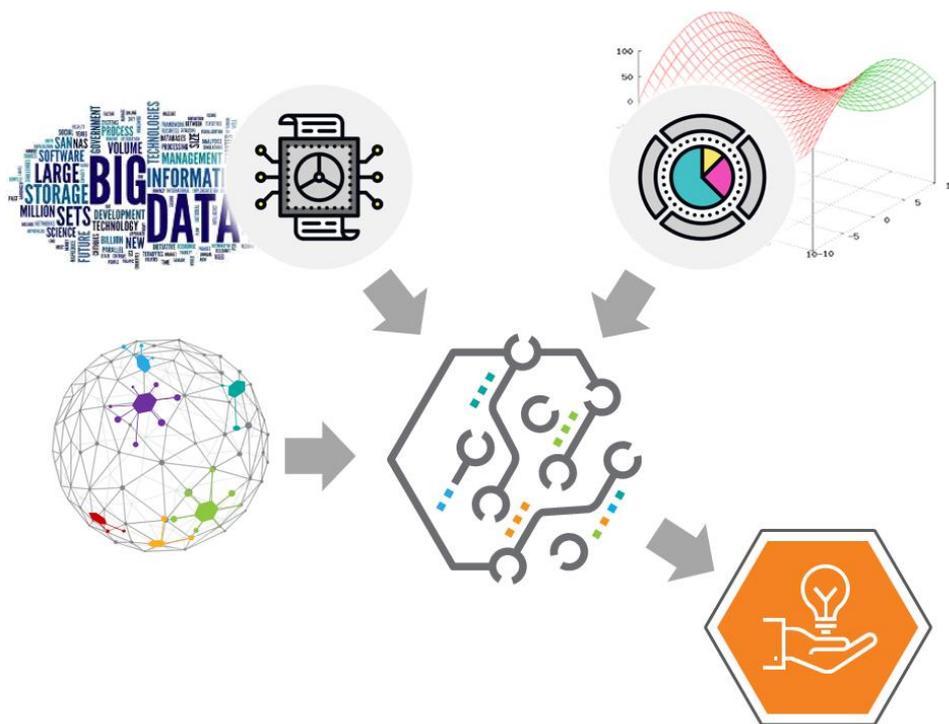


О.К. Головнин, А.С. Супрун

ТЕХНОЛОГИИ АДАПТИВНОГО ПЛАНИРОВАНИЯ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ



Санкт-Петербург
2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

О.К. Головнин, А.С. Супрун

**ТЕХНОЛОГИИ АДАПТИВНОГО ПЛАНИРОВАНИЯ
В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ
РЕШЕНИЙ**

Учебно-методическое пособие

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО

по направлению подготовки

09.04.01 Информатика и вычислительная техника

в качестве учебного пособия для реализации основных профессиональных
образовательных программ высшего образования магистратуры



Санкт-Петербург

2020

Головнин О.К., Супрун А.С. **Технологии адаптивного планирования в системах поддержки принятия решений.** – СПб: Университет ИТМО, 2020. – 88 с.

Рецензенты:

П.В. Ситников, к.т.н., заведующий корпоративной лабораторией комплексных цифровых решений (Университет ИТМО)

Пособие содержит теоретический и практический материал, касающийся технологий проектирования и реализации адаптивных экспертных систем поддержки принятия решений. Рассмотрен процесс создания таких систем, описаны основные подходы к их реализации. Особое внимание уделено адаптивному прецедентному подходу. Приведены задания для самостоятельной практической работы.

Ориентировано на студентов высших учебных заведений, обучающихся по направлению 09.04.01 Информатика и вычислительная техника и изучающих современные информационные технологии поддержки принятия решений.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2020

© О.К. Головнин, А.С. Супрун, 2020

Содержание

Введение.....	4
Раздел I. Адаптивные технологии поддержки принятия решений.....	6
Глава 1. Поддержка принятия решений на основе рассуждения по прецедентам.....	6
1.1 Методология рассуждения по прецедентам.....	6
1.2 Планирование по прецедентам.....	7
Вопросы и задания.....	11
Глава 2. Интеллектуальный анализ данных в системах поддержки принятия решений.....	17
2.1 Методы и алгоритмы нечеткой логики.....	17
2.2 Двухнаправленная интеграция генетических алгоритмов и нечеткой логики.....	19
2.3 Искусственные нейронные сети.....	21
2.4 Семантические сети.....	22
Вопросы и задания.....	25
Раздел II. Реализация систем поддержки принятия решений.....	31
Глава 3. Информационные средства поддержки принятия управленческих решений.....	31
3.1 Виды информационных систем.....	31
3.2 Процесс создания систем.....	35
Вопросы и задания.....	38
Глава 4. Продукционные системы.....	42
4.1 Построение дерева принятия решений и системы правил.....	42
4.2 Выбор алгоритма логического вывода.....	46
4.3 Проектирование структур данных.....	49
4.4 Программная реализация системы.....	51
4.5 Тестирование системы.....	59
Вопросы и задания.....	63
Глава 5. Адаптивные решения на основе прецедентов.....	68
5.1 Реализация модели CBR.....	68
5.2 Реализация алгоритма CBR-цикла.....	70
5.3 Извлечение прецедентов.....	72
5.4 Анализ алгоритмов оценивания исследуемых объектов.....	75
5.5 Автоматизация тестирования решений.....	76
Вопросы и задания.....	79
Заключение.....	84
Список рекомендуемой литературы.....	87

Введение

Эффективное применение методов и моделей поддержки принятия решений, особенно в случаях, при которых в процессе логического вывода происходит изменение факторов внешней среды или рабочего набора базы знаний, является трудной научной задачей [4, 7]. Применение адаптивных интеллектуальных информационных технологий в процессе принятия решений становится необходимостью, связанной с увеличением количества информации, поступающей в органы управления и непосредственно к руководителям, усложнением решаемых проблем, необходимостью взаимного учета гетерогенных взаимосвязанных факторов [8, 11]. В настоящем пособии рассматриваются методологии и технологии построения экспертных систем, основанных на прецедентах, которые представляются паттернами, что определяет новизну адаптивного подхода [3]. Такой подход обеспечивает возможность использования и адаптации опыта, накопленного системой поддержки принятия решений, без привлечения эксперта, что позволяет повысить точность поиска решения текущей проблемы и сократить время поиска за счет гибких подходов в работе с данными.

Пособие состоит из двух разделов: в первом рассматриваются технологии прецедентного поиска решений и способы обработки первичных данных, на основе которых строятся решения; второй раздел посвящен практическим аспектам реализации экспертных систем и применению прецедентных технологий, в нем приведены примеры построения систем информационной поддержки принятия решений с использованием адаптивных технологий.

Каждая глава в разделах завершается вопросами и заданиями, предназначенными для приобретения соответствующих навыков и контроля полученных знаний с помощью тестирования. В конце пособия приводится список рекомендуемой литературы.

Материал пособия необходимо изучать последовательно и систематически. Теоретический материал должен быть глубоко усвоен. Следует избегать механического запоминания отдельных принципов решения различных задач. Необходимо уметь применить усвоенные материалы как общую схему к решению практических задач. Решению практических задач должно быть уделено особое внимание, поскольку оно является наилучшим средством глубокого изучения особенностей адаптивных систем поддержки принятия решений.

Пособие предназначено для формирования профессиональных знаний и практических навыков применения информационных технологий адаптивной поддержки принятия решений. Пособие ориентировано на студентов высших учебных заведений, изучающих современные интеллектуальные информационные технологии, в том числе технологии

построения экспертных систем поддержки принятия решений в динамических средах.

В результате освоения материалов пособия обучающийся приобретает следующие профессиональные компетенции, а также умения и навыки:

- знание основных подходов адаптивного планирования в задаче информационной поддержки принятия решений;
- умение ставить и решать типовые задачи в контексте экспертных систем и систем поддержки принятия решений;
- владение навыками разработки интеллектуальных систем поддержки принятия решений и экспертных систем с адаптивными элементами в базе знаний;
- знание основных алгоритмов, используемых при логическом выводе по базе знаний;
- умение разрабатывать алгоритмы, использующие технологию на основе прецедентов для логического вывода;
- владение навыками применения информационных технологий в процессе анализа неструктурированных данных, используемых в системах поддержки принятия решений.

Распределение трудозатрат студентов в аудитории и в процессе самостоятельной работы формируется в соответствии с программой изучаемой дисциплины.

Раздел I. Адаптивные технологии поддержки принятия решений

Глава 1. Поддержка принятия решений на основе рассуждения по прецедентам

1.1 Методология рассуждения по прецедентам

Ключевым понятием методологии Case-based reasoning (CBR) поддержки принятий решений по прецедентам является прецедент (паттерн) [12]. Паттерн определяется как единичная запись предыдущего опыта в формализованных терминах в определённом контексте. Конкретная информация, содержащаяся в паттерне, зависит от предметной области и целей использования. В случае использования CBR для решения задач поддержки принятия решений в паттерне должны содержаться условие задачи и способ решения. Множество всех прецедентов, накопленных в процессе работы CBR, формирует информационное хранилище данных, называемое *базой знаний*.

CBR представляет собой цикл (рисунок 1): решение проблемы, выделение решения в виде паттерна и его сохранение, решение новой проблемы.

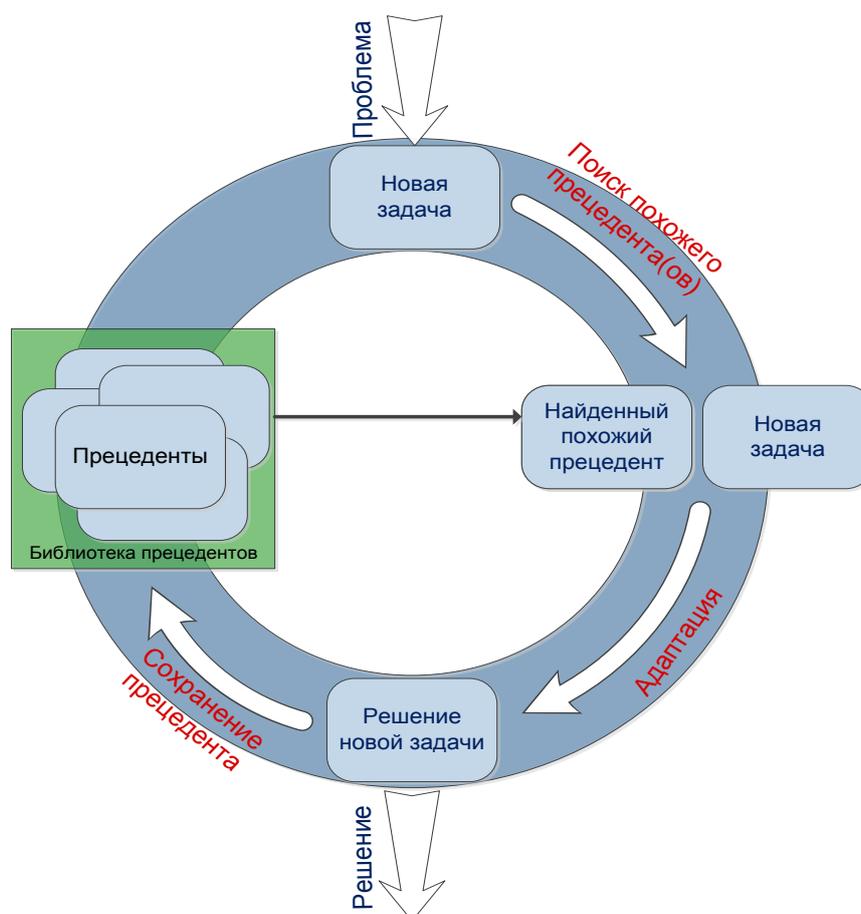


Рисунок 1 – Цикл рассуждения по прецедентам

СВР-цикл может быть описан следующими тремя процессами:

1. *Поиск похожего прецедента* – поиск паттерна с наиболее схожим условием задачи.

2. *Адаптация* – получение решения для новой задачи (с новым условием) на основе найденного паттерна. Процесс адаптации включает проверку полученного нового решения на корректность к ошибкам и его коррекцию.

3. *Сохранение паттерна* – сохранение некоторой части полученного опыта, которая может оказаться полезной для решения новых задач (пополнение или корректировка базы знаний).

Таким образом, решение каждой новой задачи в СВР сводится к решению задач поиска схожих паттернов, получения из них путем адаптации решения для новой задачи с новым условием и сохранения нового случая решения задачи в базе знаний.

Для входящих в СВР процессов, как и для самого паттерна (как хранилища информации об условии и решении), существует несколько вариантов реализации. На уровне процессов универсальные решения отсутствуют, так как реализации процессов основаны на знаниях о предметной области, что делает их специфичными в каждой конкретной прикладной задаче. В индивидуальном порядке решаются следующие вопросы:

- состав параметров в паттерне;
- возможность обобщения паттернов;
- архитектура базы знаний;
- функция проверки паттернов на сходство;
- функция поиска паттернов для новых задач;
- способ проведения адаптации;
- сохранение нового решения в базу знаний или модификация существующего.

1.2 Планирование по прецедентам

Case-based planning (СВР) – планирование на базе знаний паттернов – повторное использование знаний о предыдущем опыте для решения новых задач [6, 13]. Данный подход имеет множество преимуществ. Эффективность планирования возрастает при применении предыдущего опыта. Наличие приблизительного решения уменьшает количество усилий при поиске решения задачи по сравнению с построением нового решения без опоры на уже существующие решения. Возможность сохранения неправильных или неверных решений позволяет избежать потенциальных проблем при решении подобных задач. Использование паттерновой информации способно обеспечить высокое качество решений в силу того, что паттерн описывает действительно произошедшее на практике, а не возможное лишь в теории событие. Использование методов планирования

становится невозможным в тех случаях, когда описание предметной области неполно или ее устройство недостаточно понятно.

Структура прецедента

Паттерн, описывающий прецедент, для решения задачи планирования включает в себя:

- условие задачи (начальное состояние входных параметров и цель);
- способ решения задачи или описание решения;
- описание проблем или комментарии лица, принимающего решения, относительно решения задачи.

Существуют различные способы выражения постановки задачи: от множества фактов и множества атомарных подцелей до создания сложной структуры, опирающейся на модель предметной области. Решение задачи может храниться в виде непосредственно решения или способа достижения этого финального решения – последовательности принятых во время планирования решений, таких как выбор действия.

База знаний хранит паттерны в обобщенном виде и в виде конкретных примеров. Как описание условия задачи, так и описание решения может быть обобщено. Паттерны в базе знаний хранятся как отдельные единицы опыта и разделяются на части. В первом случае каждый эпизод планирования приводит к появлению отдельного целостного описания для этого эпизода. Во втором случае будут сохранены решения подзадач исходной задачи, в результате чего в библиотеку будет записано сразу множество решений для различных задач, что увеличивает вероятность их повторного использования. Решение исходной задачи целиком при этом может и не сохраняться. Если решение все же сохранено, его план будет представлен в виде множества ссылок на получившиеся решения подзадач.

Поиск прецедента

Процесс поиска прецедента, похожего на новую задачу, может быть ориентирован с учетом различных критериев:

- найти прецедент с наиболее надежным результирующим планом;
- найти прецедент с планом, имеющим минимальное время исполнения;
- найти прецедент с минимальным временем получения нового решения (минимум модификаций);
- найти прецедент, отражающий наиболее современный (поздний) опыт;
- и так далее.

Как правило, задача поиска прецедента делится на три подзадачи:

- выбор свойств для сопоставления;
- сопоставление;

- выбор решения.

В первой подзадаче необходимо выбрать свойства, учитываемые при поиске лучшего прецедента. В системах планирования, основанных на прецедентах, для этого обычно используются цель, начальное состояние и, иногда, описание сложностей, способных возникнуть во время решения задачи.

Сопоставление заключается в поиске одного или нескольких планов, которые частично или полностью совпадают по выбранным свойствам с текущей проблемой. Сопоставление проводится несколькими различными способами, наиболее распространенными из которых являются полное сопоставление и использование мер подобия.

При полном сопоставлении проводится поиск таких прецедентов, у которых соответствующие свойства точно совпадают с выбранными свойствами новой задачи. При отсутствии нужного решения выполняется обобщение выбранных свойств новой задачи и повторная попытка найти прецедент.

Мера подобия – функция, вычисляющая степень сходства заданного прецедента и новой задачи. Простейшая мера подобия может быть определена как количество общих подцелей. Более гибкие меры подобия вычисляют взвешенную сумму общих подцелей и степень сходства начальных состояний. Балансировка важности и полезности подцелей осуществляется весами. Веса могут как быть одинаково определены для всех прецедентов сразу в виде констант, так и являться отдельным набором значений для каждого отдельного прецедента в более сложных предметных областях. При использовании весов необходимо принять во внимание следующие аспекты.

Важность подцелей. Важные подцели должны оказывать большее влияние при сопоставлении прецедента и новой проблемы.

Трудность достижения подцелей. Легко достижимые подцели не должны иметь высокий приоритет на фазе поиска прецедента.

Независимость подцелей. При зависимости подцелей количество подцелей, которые удовлетворяются прецедентом изначально (до адаптации), не будет отражать сложность адаптации прецедента. Прецедент может оказаться неудовлетворительным даже при неудовлетворении одной подцели.

Один из способов настройки весов проводит постепенное обучение важности подцелей, основанное на частоте успешного или неуспешного повторного применения прецедента.

Так как использование меры подобия автоматически ранжирует найденные решения, каждый найденный прецедент имеет степень подобия новой задаче в качестве полезного побочного эффекта.

При нахождении нескольких подходящих прецедентов в результате решения подзадачи сопоставления планировщик должен выбрать один или

несколько из них. При использовании мер подобия эта подзадача решается автоматически. Но возможно применение и прочих методов, основанных, например, на степени обобщенности прецедента (чем более конкретен прецедент, тем меньше усилий потребуется на адаптацию). Также возможно применение интерактивного способа.

Адаптация

Задача адаптации состоит:

- в подмене цели и начальных условий выбранного прецедента целью и начальными условиями новой задачи;
- в обеспечении корректности плана после подмены.

После подмены цели и начальных условий может возникнуть необходимость в изменении плана (готового или полученного по трассе вывода). С исчезновением целей некоторые шаги плана могут оказаться ненужными. Возможно появление новых целей, а также новых подцелей, так как ряд шагов становится неприменим из-за изменения начальных условий. Корректировка плана может быть проведена либо пользователем, либо автоматически.

Автоматическая адаптация не всегда целесообразна. Иногда адаптация вообще не требуется, даже при некорректности плана. Например, когда найденные решения не будут применяться системой, а будут выведены пользователю в качестве подсказки или рекомендации как примерный план действий. Даже если система не применяет найденные решения, в некоторых предметных областях получение неверных решений недопустимо, и необходим контроль со стороны человека. В этом случае предложенный системой план действий может быть скорректирован пользователем.

Существует несколько подходов автоматической адаптации.

Простейшим методом адаптации является наложение ограничений. При применении данного подхода переменные связываются с конкретными объектами новой задачи без изменения структуры плана. Такой метод применим только для планов, полученных из прецедентов, со структурой, аналогичной новой задаче, но оперирующей другими объектами. Рекомендуется использовать этот метод вместе с каким-либо другим методом, выполняющим трансформацию структуры плана.

Эвристические методы адаптации используют наборы эвристических правил преобразования плана. Иногда эти преобразования описываются набором продукционных правил, трансформирующих план в зависимости от контекста. Эвристический метод обычно используется при необходимости изменения структуры плана.

Рекурсивная адаптация использует планирование по прецедентам для каждой из неудовлетворенных подцелей. Найденное решение некоторой подцели может породить множество открытых подцелей,

которые достигаются тем же методом. Из-за необходимости выполнять объединение текущего решения задачи и способа разрешения очередной подцели, в общем виде метод является достаточно сложным.

Генеративная адаптация нацелена на достижение еще не достигнутых целей методом, аналогичным порождению плана. Этот метод рекомендуется использовать совместно с другими видами адаптации для достижения наиболее удовлетворительных результатов. Когда другой метод не может справиться с созданием требуемой части плана, для этой цели применяется какой-либо алгоритм классического планирования. В качестве примеров систем, использующих этот вид адаптации, можно привести: SPA, DerSNLP, PRODIGY/NoLimit.

Адаптация по паттернам применяет методологию CBR к случаям адаптации. Система сохраняет адаптации в базе знаний и пользуется этим опытом для осуществления новых адаптаций. Для заданной новой проблемы адаптации ведется поиск паттернов адаптации с наиболее похожими целями и выполняется попытка применения найденных методов адаптации в текущей задаче. Примером системы с адаптацией по прецедентам является DIAL.

Сохранение опыта

По итогу выполнения фаз поиска и адаптации создается план для решения текущей задачи. Сохранение текущего опыта замыкает цикл рассуждений по паттернам и пополняет базу знаний. Обучение осуществляется посредством наблюдения за ответной реакцией при исполнении предложенного решения. Модель может адаптироваться как на положительных, так и на отрицательных примерах.

Сохранение знаний является необязательным шагом. База знаний может быть составлена заранее и содержать решения наиболее распространенных в данной предметной области задач. Отказ от использования адаптации может оказаться целесообразным в статичных задачах, когда требуется повысить эффективность принимаемых решений. Решения по базе знаний, составленной экспертами в предметной области, будут более эффективными по сравнению с решениями, сформированными автоматически. Поэтому получаемая таким образом база знаний, с одной стороны, должна быть более компактна, а с другой – содержать паттерны, наиболее пригодные для повторного использования.

Вопросы и задания

Тест для самоконтроля

1. Что является ключевым понятием методологии Case-based reasoning (CBR)?

Варианты ответа:

- A. Форма;
 - B. Распределение;
 - C. Паттерн;
 - D. Патент.
2. Сколькими процессами может быть описан CBR-цикл?
- Варианты ответа:
- A. Одним процессом;
 - B. Двумя процессами;
 - C. Тремя процессами;
 - D. Четырьмя процессами.
3. Как по-другому называется информационное хранилище данных?
- Варианты ответа:
- A. База данных;
 - B. База информации;
 - C. База знаний;
 - D. Банк данных.
4. Паттерн, описывающий прецедент, для решения задачи планирования, не включает в себя:
- Варианты ответа:
- A. Условие задачи (начальное состояние входных параметров и цель);
 - B. Способ решения задачи или описание решения;
 - C. Сохранение нового решения в базу знаний или модификация существующего;
 - D. Описание проблем или комментарии лица, принимающего решения, относительно решения задачи.
5. Какой подзадачи для поиска прецедента не существует?
- Варианты ответа:
- A. Выбор свойств для сопоставления;
 - B. Сопоставление;
 - C. Выбор паттерна;
 - D. Выбор решения.
6. Какие аспекты необходимо принять во внимание при использовании весов?
- Варианты ответа:
- A. Важность подцелей, независимость подцелей, легкость достижения подцелей;
 - B. Важность подцелей, зависимость целей, трудность достижения целей;
 - C. Важность подцелей, трудность достижения подцелей, независимость подцелей;

- D. Важность подцелей, зависимость подцелей, легкость достижения подцелей.
7. При сохранении опыта база знаний должна...
- Варианты ответа:
- A. Быть расширенной;
 - B. Содержать паттерны для повторного использования;
 - C. Быть компактной и содержать паттерны для повторного использования;
 - D. Быть расширенной и содержать паттерны для повторного использования.
8. Какой подход автоматической адаптации не существует?
- Варианты ответа:
- A. Наложение ограничений;
 - B. Эвристические методы;
 - C. Прямые методы;
 - D. Генеративная адаптация.
9. Паттерн – это:
- Варианты ответа:
- A. Распространённый подход к решению класса часто встречающихся проблем, являющийся неэффективным, рискованным или непродуктивным
 - B. Единичная запись предыдущего опыта в формализованных терминах в определённом контексте
 - C. Шаблоны ошибок, которые совершаются при решении различных задач
 - D. Неповторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования, в рамках некоторого часто возникающего контекста
10. Какой процесс не входит в СВР-цикл?
- Варианты ответа:
- A. Поиск похожего прецедента;
 - B. Адаптация;
 - C. Внедрение паттерна;
 - D. Сохранение паттерна.
11. Вид хранения паттерна в базе знаний:
- Варианты ответа:
- A. Только в обобщенном виде;
 - B. Только в виде конкретных задач;
 - C. В обобщенном виде и в виде конкретных задач;
 - D. Не в обобщенном виде и не в виде конкретных задач.
12. В случае сохранения паттерна в базе знаний как отдельной единицы опыта происходит...
- Варианты ответа:

- A. Сохранение множество решений для различных задач;
- B. Сохранение решения подзадач исходной задачи, в результате чего в библиотеку будет записано сразу;
- C. Сохранение эпизода планирования, которое приводит к появлению отдельного целостного описания для этого эпизода;
- D. Сохранение всего процесса планирования.

13. Установите соответствие между подзадачами поиска прецедента и его описанием

1	Выбор свойств для сопоставления	A	Выбор альтернативы из множества вариантов достижения конкретной цели
2	Сопоставление	B	Не является подзадачей поиска прецедента
3	Выбор паттерна	C	Поиск одного или нескольких планов, которые частично или полностью совпадают по выбранным свойствам с текущей проблемой.
4	Выбор решения.	D	В системах планирования, основанных на прецедентах, для этого обычно используются цель, начальное состояние и, иногда, описание сложностей, способных возникнуть во время решения задачи

14. Мера подобия – это...

Варианты ответа:

- A. Функция, вычисляющая степень различия заданного прецедента и новой задачи;
- B. Процедура, вычисляющая степень различия заданного прецедента и новой задачи;
- C. Функция, вычисляющая степень сходства заданного прецедента и новой задачи;
- D. Процедура, вычисляющая степень сходства заданного прецедента и новой задачи.

15. Задача адаптации не состоит из...

Варианты ответа:

- A. Подмены цели выбранного прецедента целью новой задачи;
- B. Подмены начальных условий выбранного прецедента начальными условиями новой задачи;
- C. Обеспечения отклонения плана после подмены;
- D. Обеспечения корректности плана после подмены.

16. Установите соответствие между подходами автоматической адаптации и ее описанием:

1	Наложение ограничений	А	Применение данного подхода нацелено на достижение еще недостигнутых целей методом, аналогичным порождению плана
2	Эвристический метод адаптации	В	При применении данного подхода используют планирование по прецедентам для каждой из неудовлетворенных подцелей. Найденное решение некоторой подцели может породить множество открытых подцелей, которые достигаются тем же методом
3	Рекурсивная адаптация	С	При применении данного подхода используют наборы определенных правил преобразования плана, которые описываются набором продукционных правил, трансформирующих план в зависимости от контекста
4	Генеративная адаптация	Д	При применении данного подхода переменные связываются с конкретными объектами новой задачи без изменения структуры плана

17. Наложение ограничений применимо...

Варианты ответа:

- А. Для планов, полученных из прецедентов, со структурой, отличной от новой задачи, но оперирующей другими объектами;
- В. Для планов, полученных из прецедентов, со структурой, аналогичной новой задаче, но оперирующей этими же объектами;
- С. Для планов, полученных из прецедентов, со структурой, аналогичной новой задаче, но оперирующей другими объектами;
- Д. Для планов, полученных из прецедентов, со структурой, отличной от новой задачи и оперирующей этими же объектами.

18. Эвристический метод адаптации используется...

Варианты ответа:

- А. При необходимости сохранения структуры плана;

- В. При необходимости объединения текущего решения задачи и способа разрешения очередной подцели;
- С. При необходимости изменения структуры плана;
- Д. При необходимости разделения текущего решения задачи и способа разрешения очередной подцели.
19. Рекурсивная адаптация используется...
- Варианты ответа:
- А. При необходимости изменения структуры плана;
- В. При необходимости сохранения структуры плана;
- С. При необходимости объединения текущего решения задачи и способа разрешения очередной подцели;
- Д. При необходимости разделения текущего решения задачи и способа разрешения очередной подцели.
20. Модель может адаптироваться...
- Варианты ответа:
- А. Только на положительных примерах;
- В. Только на отрицательных примерах;
- С. Как на положительных, так и на отрицательных примерах;
- Д. Ни на положительных, ни на отрицательных примерах.

Ответы на вопросы теста

1. С
2. С
3. С
4. С
5. С
6. С
7. С
8. С
9. С
10. С
11. С
12. С
13. 1-Д; 2-С; 3-В; 4- А;
14. С
15. С
16. 1-Д; 2-С; 3-В; 4- А;
17. С
18. С
19. С
20. С

Глава 2. Интеллектуальный анализ данных в системах поддержки принятия решений

2.1 Методы и алгоритмы нечеткой логики

Одним из важных способов анализа задач исследования данных является их перенос на формальный язык и анализ полученной модели [9].

Неопределенности при недостаточном количестве информации делятся следующим образом:

- неполнота;
- недостоверность;
- неизвестность.

Недостоверность делится на физическую (проявляется под влиянием внешней среды) и лингвистическую (источником служит словесное обобщение).

Существует два типа физической неопределенности:

- случайность – это погрешность, возникающая под действием внешней среды;
- неточность – погрешность измерительных приборов.

Лингвистическая неопределенность делится на следующие группы:

- неоднозначность толкования высказывания (существует семантическая и синтаксическая неоднозначность);
- неопределенность толкования слова – возникает, если несколько разных задач управления относятся к одному слову.

Для разрешения неопределенностей физического типа применяют методы из классической теории множеств или теории вероятностей. Но разрешение физических неопределенностей не решает всех проблем, так как всё больше систем должны разрешать задачи лингвистической неопределенности, связанные со словесными обобщениями.

Главными особенностями нечеткой логики являются:

- правила получения решения при помощи условных высказываний описываются при помощи логических выводов;
- нечеткая логика работает с большим числом частных правил. При этом для всех отдельных областей применяются свои правила, что позволяет упростить процессы обработки данных;
- полученные при помощи условных высказываний правила помогают разрешить задачи классификации в реальном времени, т.е. происходит постепенное уточнение данных оператором. Такой способ позволяет улучшить качество классификатора уже в процессе его использования.

Между нечеткой логикой и мышлением человека существуют общие черты, исходя из этого методы управления, разработанные с использованием нечеткой логики, принято считать эвристическими. Эвристический способ получения решения задач построен не на

математических моделях и алгоритмах, а при помощи логического рассуждения.

Эвристические алгоритмы чаще всего используются в самоорганизующихся системах, которые решают проблемы нечеткости данных, основанные на мышлении человека. Но с течением времени самоорганизующиеся системы всё больше способны к самообучению по ранее полученной информации.

Реализация процесса возможна по следующему алгоритму (рисунок 2).

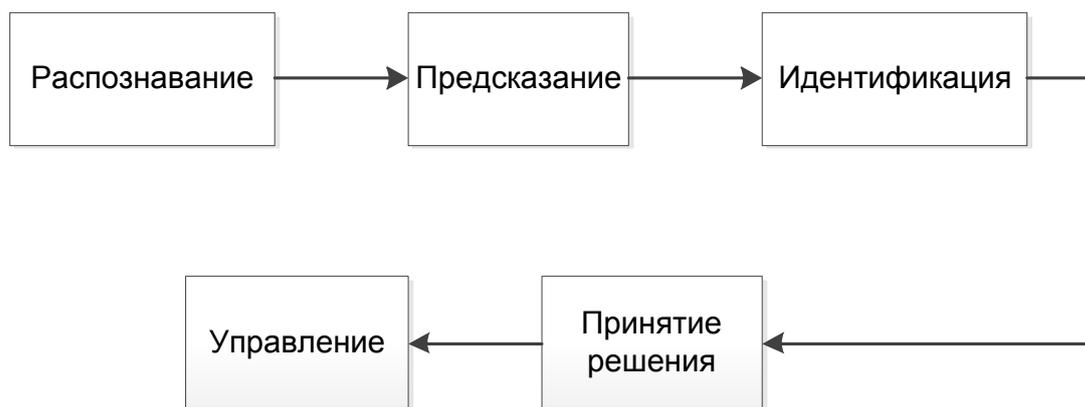


Рисунок 2 – Реализация процесса управления

Расчет факторов уверенности

Символьные, числовые и логические выражения могут иметь фактор уверенности, т.е. быть нечеткими. Тогда результат вычисления выражения – это несколько значений с разными факторами уверенности. Для того чтобы указать, как объединять факторы уверенности при вычислении фактора уверенности логического выражения или для присваиваний значений нечетких переменных, используется следующая логика:

- М – минимум;
- Р – произведение;
- А – среднее от М и Р;
- В – баланс между М и Р.

Определение фактора уверенности переменной во время консультации может осуществляться следующим образом:

- РА – вероятностная сумма средних;
- РВ – вероятностная сумма балансов;
- РР – вероятностная сумма произведений;
- РМ – вероятностная сумма минимумов;
- ММ – максимум минимумов;
- МР – максимум произведений;
- МА – максимум средних;

- МВ – максимум балансов;
- АА – среднее произведений;
- АМ – среднее минимумов;
- АА – среднее средних;
- АВ – среднее балансов;
- ВР – баланс произведений;
- ВМ – баланс минимумов;
- ВА – баланс средних;
- ВВ – баланс балансов.

2.2 Двухнаправленная интеграция генетических алгоритмов и нечеткой логики

Применение нечеткой логики позволяет принимать решения в условиях неопределенности. Нечеткая логика обеспечивает базис для представления различных форм знаний в неопределенных средах, позволяет моделировать взаимодействие переменных информационной системы. Генетические алгоритмы дают возможность обучения, глобального и локального поиска.

Объединение нечеткой логики с генетическими алгоритмами помогает привести различные свойства генетического алгоритма к текущим критериям пространства поиска (рисунок 3).

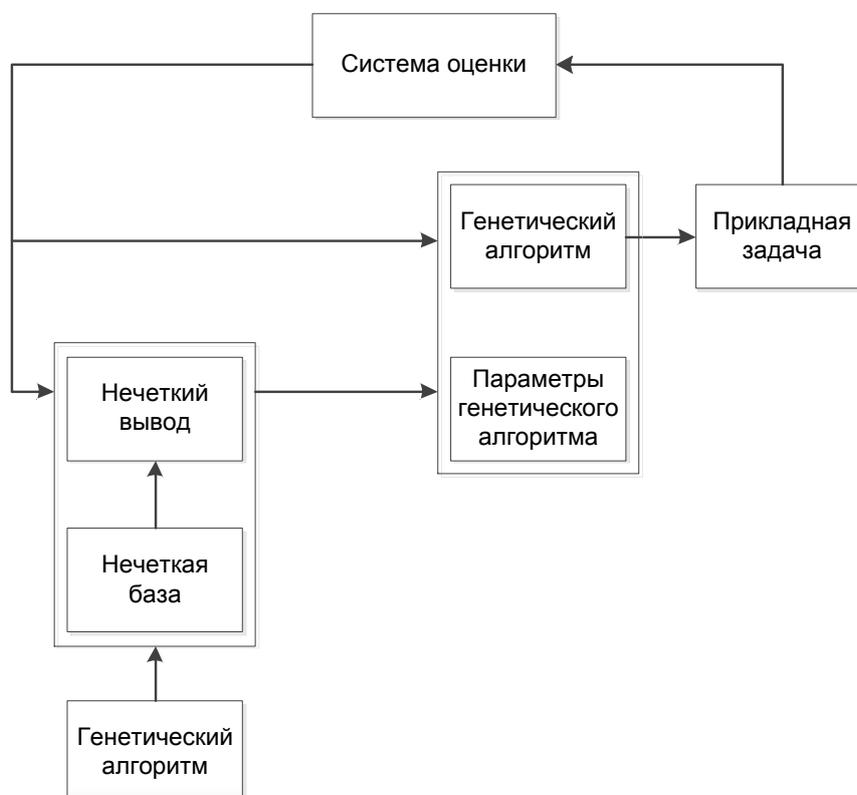


Рисунок 3 – Генетический алгоритм с функцией обратной связи

Воспользовавшись генетическим алгоритмом для оптимизации задач, использующих нечеткую логику, необходимо настроить базу нечетких знаний, которые содержат нечеткие лингвистические правила и термины (рисунок 4).

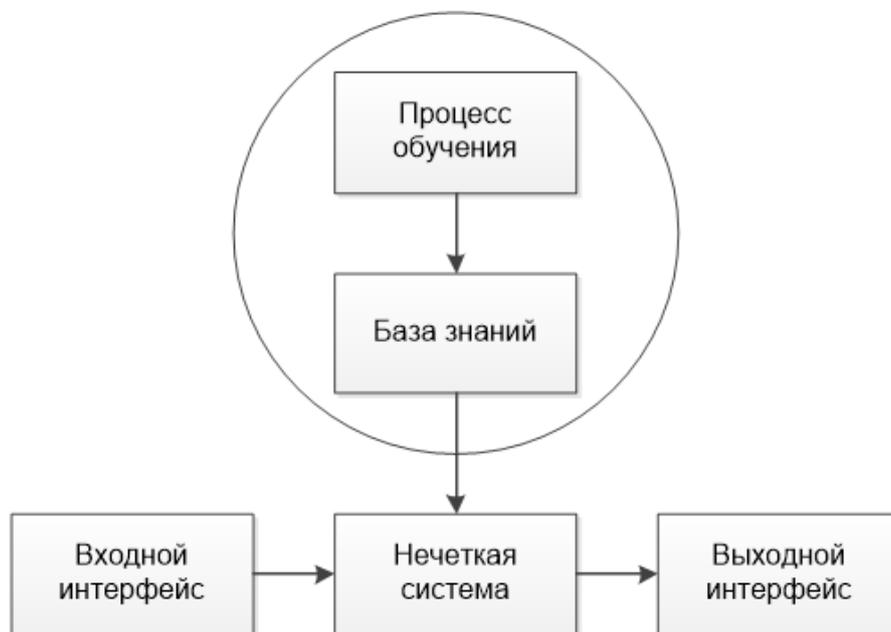


Рисунок 4 – Использование генетического алгоритма для оптимизации задач

Для реализации такого алгоритма необходимо определить следующие параметры: входы и выходы систем, лингвистические термы и нечеткие правила.

Входы и выходы системы – входные данные используют разные показатели эффективности популяции и отличные показания таких параметров, как максимальная, средняя и минимальная степень пригодности, значение разброса.

Поддерживаются следующие алгоритмы:

1. *Генотипический разброс*, вычисляется следующим образом:

$$ED = \frac{(d_{cp} - d_{min})}{(d_{max} - d_{min})},$$

где d_{cp} , d_{min} , d_{max} – среднее, минимальное и максимальные отклонения индивидуальной эффективности P_i от наилучшего значения в виде R размером N :

$$d_{cp} = N * i * \frac{1}{N},$$

$$d = \sum(C_{best} * Ps),$$

$$d_{max} = \max \{d(C_{best}, Ps), Ps \in R\},$$

$$d_{min} = \min \{d(C_{best}, Ps), Ps \in R\}.$$

Разброс значений ED от 0 до 1. При небольших значениях ED все значения располагаются вблизи наилучшего, из этого следует сходимость популяции.

2. *Фенотипический разброс*, который измеряет отношение между самой лучшей и средней степенью пригодности:

$$PDM = \frac{f_{best}}{f_{cp}}.$$

Текущие параметры управления могут также рассматриваться как входы.

Лингвистические термы – любой вход и выход имеет связь с набором лингвистических терм. Нечеткие правила – после указания входов и выходов, определения лингвистических средств и функций, определяются нечеткие правила, которые должны описывать связь между ними.

2.3 Искусственные нейронные сети

Искусственная нейронная сеть – система, которая имеет возможность изменять собственную структуру под действием внешних факторов. Основной составляющей частью нейронной сети является формальный нейрон, где:

- x_1, x_2, \dots, x_n – вектор входных сигналов,
- w_1, w_2, \dots, w_n – значения весов входных сигналов нейрона,
- x^* – выходной сигнал нейрона.

Формальный нейрон содержит следующие элементы: весовые коэффициенты (1), сумматор (2) и преобразователь (3) (рисунок 5).

Весовые коэффициенты задают силу связи между нейронами. Сумматор исполняет сложение сигналов, поступающих на вход, первоначально умножая их на соответствующие весовые коэффициенты. Преобразователь задает функцию одного аргумента – выхода сумматора, такая функция называется функцией активации.

Формальные нейроны объединяются таким образом, что выходные сигналы одних нейронов являются входными для других. Связанные между собой формальные нейроны образуют нейронную сеть.

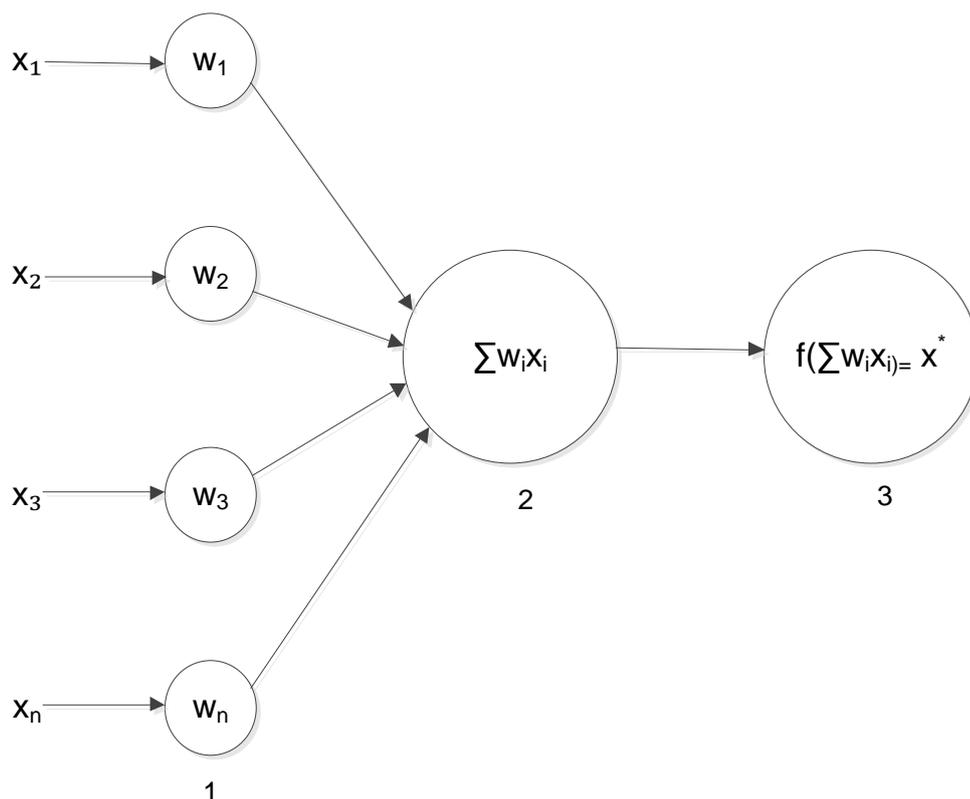


Рисунок 5 – Формальный нейрон

Существует три типа нейрона:

- входной нейрон предназначен для принятия входного сигнала;
- выходной нейрон предоставляет результирующий выходной сигнал;
- скрытый нейрон обрабатывает сигнал и передает его на следующий слой.

По типам межнейронных связей делят нейронные сети на следующие виды:

- рекуррентные нейронные сети – нейронные сети с обратной связью, в которых сигналы передаются вне зависимости от расположения нейронов;
- сети прямого распространения – сигнал способен распространяться только от входного нейрона к выходному.

Для обучения нейронных сетей существует два способа обучения:

- обучение с учителем;
- обучение без учителя.

2.4 Семантические сети

Семантические сети применяются для писания метода представления знаний [4]. Этот метод принципиально основывается на сетевой структуре. Под сетевой структурой мы понимаем совокупность элементов / вершин / концептов / понятий, связанных между собой ссылками / связями /

отношениями. Графически сеть представляется графом произвольного вида, вершинам которого соответствуют понятия, а дугам – связи или отношения. Первое использование этого аппарата – моделирование памяти человека в психологии.

Семантическая сеть может использоваться для анализа выражений естественного языка. Язык является уникальным средством описания знаний, однако он имеет недостатки, такие как избыточность, неоднозначность.

В основе сетевых моделей лежат общие принципы моделирования. Самым общим понятием любой модели является понятие сущности. Под сущностью понимается любой объект физической или абстрактной природы, относительно которого ведется моделирование: процесс, явление, физический объект, абстрактный объект. Модельная сущность называется М-сущностью (десигнат/метка).

Семантическая сеть является абстрактной моделью, т.е. набором М-сущностей некоторой предметной области. Главным недостатком сетевых моделей является их большая размерность. Учет всех фактов и отношений предметной области требует построения сложных сетей, в которых количество вершин может достигать нескольких десятков тысяч. В такой сети становится сложной проблема поиска решения и нужной информации. Поэтому аппарат семантических сетей сильно связан с развитием средств вычислительной техники. Любую семантическую сеть можно представить в машинном виде.

Простейший вид семантической сети – дискретная семантическая сеть. Для ее формального определения нужно задать граф:

$$ДСС = \langle G, A, P \rangle,$$

где G – граф семантической сети,

A – алгоритм поиска на сети,

P – правило интерпретации сети (соответствие вершин и дуг сущностям).

В сети могут быть определены правила иерархии и установлена иерархия понятий. Иерархия понятий возникает, когда выделяется объект более высокого уровня.

Для сетей могут задаваться законы распространения возбуждений, которые входят в алгоритмы поиска. Возбуждение описывает степень активности вершины или дуги в процессе поиска решения.

Семантические группы

Семантические группы – некоторое множество экземпляров, обладающих определенной мерой сходства по признаку и поведению:

$$CG = \langle W, I, D, T \rangle,$$

где W – имя семантической группы,

I – идентификатор семантической группы,

D – проектная мощность семантической группы (максимальное количество экземпляров),

T – фактическое заполнение семантической группы экземплярами.

Имя семантической группы образуется как результат выполнения операции обобщения. Простые семантические группы содержат в качестве экземпляров только имена экземпляров.

Экземпляры семантической группы являются частным понятием семантической группы.

В результате применения операции обобщения семантические группы могут выстраиваться в иерархии, поэтому связи могут быть двух видов: P -связи – вертикальные межуровневые и Γ -связи – горизонтальные на одном уровне между экземплярами.

Экземпляры семантических групп не состоят из других семантических групп. Метод семантических групп позволяет под ограниченное число запросов получить несколько вариантов семантических сетей. Результатом консультации с такой сетью является нахождение одной или нескольких вершин (экземпляров семантических групп), удовлетворяющих условиям запроса.

Семантическая сеть называется релевантной запросу, если она может отвечать на запросы этого вида. Под информационным запросом понимают некоторое утверждение, содержащее требования по предметной области и условия, которые должны быть выполнены.

Выделение семантических групп производится экспертом на стадии анализа задач. Обычно используется методика шкал Осгуда, позволяющая по набору заранее определенных признаков относить экземпляры к той или иной группе. Каждая шкала должна быть дискретной и непрерывной.

Язык, который задает отношения и описывает сеть на этом уровне, назовем языком семантических групп. Как и всякий язык, он содержит словарь, в котором есть правая и левая часть: в правой части расположены имена семантических групп, чаще всего корневые основы, а в левой части содержатся соответствующие идентификаторы.

Словарь позволяет перейти от ограниченного естественного языка к языку семантических групп, т.е. присвоить запросу определенную кодировку.

Язык формирования ответа позволяет представить результат в виде активных выходных вершин в требуемой для пользователя форме с заголовком и дополнительными комментариями.

Схема логического вывода

Общая схема логического вывода методом семантических групп выглядит следующим образом.

По условиям запроса часть групп считаются входными. В начале консультации все вершины семантической сети являются неактивными – им приписываются 0 или *False*. Все связи считаются неактивными. Результат формируется следующим образом. Активность обычных вершин строится по дизъюнктивному признаку, т.е. достаточно активности одного входа, чтобы вершина была активной и все связи из данной вершины были активными. Чтобы сделать селекцию или отбор, надо ввести конъюнктивные вершины. Конъюнктивные вершины работают по принципу конъюнкции. Чтобы вершина была активной и выход был активным, необходима активность всех входов одновременно. Результирующие вершины называются выходными.

Каждая конъюнктивная группа является отдельной семантической группой. Семантическая сеть в методе семантических групп содержит все возможные ответы на параметрический запрос данного типа.

Представление связей в семантической группе производится следующим образом: каждая связь задается записью:

$$\langle J \rangle \langle n \rangle \langle p \rangle \langle I \rangle \langle m \rangle,$$

где J и I – идентификаторы семантической групп,

n и m – номера экземпляров,

p – вид связи (или коэффициент силы связи).

Существуют два основных способа реализации метода семантических групп:

1. Объектно-ориентированный способ. Объявляется класс вершин, класс связей. В качестве свойств приписываются флаги, указывающие на активность.

2. Массивы. Экземпляры и связи хранятся в виде массивов. Метод требует предварительного расчета размеров массивов.

Вопросы и задания

Тест для самоконтроля

1. На какие типы не делится лингвистическая неопределенность?

Варианты ответа:

А. Неоднозначность толкования, ошибка толкования, неопределенность толкования, неточность толкования;

В. Неоднозначность толкования, неопределенность толкования;

С. Ошибка толкования, неоднозначность толкования;

D. Неточность толкования, неопределенность толкования, неоднозначность толкования.

2. Что не является главной особенностью нечеткой логики?

Варианты ответа:

A. Правила получения решения при помощи условных высказываний описываются при помощи логических выводов;

B. Нечеткая логика работает с большим числом частных правил. При этом для всех отдельных областей применяются свои правила, что позволяет упростить процессы обработки данных;

C. Правила, полученные при помощи условных высказываний, помогают разрешить задачи классификации в реальном времени;

D. Нечеткая логика не имеет общих черт с мышлением человека, исходя из этого методы управления, разработанные с использованием нечеткой логики, принято считать математическими.

3. В каких системах чаще всего используются эвристические алгоритмы?

Варианты ответа:

A. Самоорганизующихся;

B. Статических;

C. Квазистатических;

D. Во всех редко.

4. Какова возможная реализация алгоритма процесса управления?

Варианты ответа:

A. Распознавание → Предсказание → Идентификация → Принятие решения → Управление;

B. Распознавание → Идентификация → Принятие решения → Управление;

C. Предсказание → Распознавание → Идентификация → Принятие решения → Управление;

D. Идентификация → Принятие решения → Управление.

5. Как происходит работа формального нейрона?

Варианты ответа:

A. Весовые коэффициенты задают силу связи между нейронами. Сумматор исполняет сложение сигналов, поступающих на вход, и делит полученный сигнал на сумму весовых коэффициентов. Преобразователь задает функцию одного аргумента – выхода сумматора, такая функция называется функцией активации;

B. Весовые коэффициенты задают коэффициенты целевой функции. Сумматор исполняет сложение сигналов, поступающих на вход, первоначально умножая их на

соответствующие весовые коэффициенты. Преобразователь задает функцию одного аргумента – выхода сумматора, такая функция называется функцией активации;

C. Весовые коэффициенты задают силу связи между нейронами. Сумматор исполняет сложение сигналов, поступающих на вход, первоначально умножая их на соответствующие весовые коэффициенты. Преобразователь задает функцию одного аргумента – выхода сумматора, такая функция называется функцией активации;

D. Весовые коэффициенты задают силу связи между нейронами. Сумматор исполняет сложение сигналов, поступающих на вход, и соответствующих весовых коэффициентов. Преобразователь задает функцию многих аргументов – выходов сумматора.

6. Какой элемент не является составляющим формального нейрона?

Варианты ответа:

- A. Весовой коэффициент;
- B. Сумматор;
- C. Преобразователь;
- D. Дешифратор.

7. Какого типа нейронов не существует?

Варианты ответа:

- A. Входной нейрон;
- B. Выходной нейрон;
- C. Промежуточный нейрон;
- D. Скрытый нейрон.

8. Рекуррентные нейронные сети – это...

Варианты ответа:

- A. Нейронные сети с обратной связью, в которых сигналы передаются вне зависимости от расположения нейронов;
- B. Нейронные сети, в которых сигнал способен распространяться только от входного нейрона к выходному;
- C. Нейронные сети, в которых сигнал способен распространяться только от выходного нейрона к входному;
- D. Нейронные сети без обратной связи, в которых сигналы передаются вне зависимости от расположения нейронов.

9. Сколько существует способов обучения нейронной сети?

Варианты ответа:

- A. 1;
- B. 2;
- C. 4;
- D. 3.

10. Как можно представить нейронную сеть графически?

Варианты ответа:

- А. графом произвольного вида;
- В. двоичным деревом;
- С. ориентированным графом;
- Д. деревом произвольного вида.

11. Каков главный недостаток сетевых моделей?

Варианты ответа:

- А. Большая размерность;
- В. Плохая пропускная способность;
- С. Плохая масштабируемость;
- Д. Сложность составления.

12. Любую ли семантическую модель можно представить в машинном виде?

Варианты ответа:

- А. Нет, она должна отвечать определенным требованиям;
- В. Нет, семантические модели не представимы в машинном виде;
- С. Да;
- Д. Неизвестно.

13. Как формально определить дискретную семантическую сеть?

Варианты ответа:

- А. Задать граф: $DCC = \langle G, A, P \rangle$;
- В. Задать граф: $CG = \langle W, I, D, T \rangle$;
- С. Задать записью: $\langle J \rangle \langle n \rangle \langle p \rangle \langle I \rangle \langle m \rangle$;
- Д. Дать словесное описание.

14. В результате применения операции обобщения семантические группы могут выстраиваться в иерархии. Какие виды связи при этом используются?

Варианты ответа:

- А. Только П-связи – вертикальные межуровневые;
- В. Только Г-связи – горизонтальные на одном уровне между экземплярами;
- С. П-связи – вертикальные межуровневые и Г-связи – горизонтальные на одном уровне между экземплярами;
- Д. Никаких связей.

15. Метод семантических групп позволяет...

Варианты ответа:

- А. Под ограниченное число запросов получить несколько вариантов семантических сетей;
- В. Под ограниченное число запросов получить единственный вариант семантической сети;
- С. Под неограниченное число запросов получить единственный вариант семантической сети;

D. Под неограниченное число запросов получить несколько вариантов семантических сетей.

16. Какие требования методика шкал Осгуда предъявляет для используемых шкал?

Варианты ответа:

- A. Каждая шкала должна быть дискретной и непрерывной;
- B. Каждая шкала должна быть дискретной и конечной;
- C. Каждая шкала должна быть непрерывной;
- D. Каждая шкала должна быть дискретной.

17. Какой записью задаются связи в семантической группе?

Варианты ответа:

- A. $\langle J \rangle \langle n \rangle \langle p \rangle \langle I \rangle \langle m \rangle$;
- B. $\langle J \rangle \langle I \rangle \langle m \rangle$;
- C. $\langle h \rangle \langle e \rangle \langle I \rangle \langle p \rangle$;
- D. У них нет определенного представления записью.

18. Какие значения имеют все вершины семантической сети в начале консультации в логическом выводе методом семантических групп?

Варианты ответа:

- A. 0 или *False*;
- B. 1 или *True*;
- C. Бесконечность;
- D. Любое.

19. По какому признаку строится активность обычных вершин в логическом выводе методом семантических групп?

Варианты ответа:

- A. Смешанному;
- B. Дизъюнктивному;
- C. Конъюнктивному;
- D. Инверсивному.

20. Каковы основные способы реализации метода семантических групп?

Варианты ответа:

- A. Объектно-ориентированный способ, массивы;
- B. Массивы, деревья;
- C. Объектно-ориентированный способ, массивы, списки;
- D. Деревья, графы.

Ответы на вопросы теста

- 1. B
- 2. D
- 3. A
- 4. A
- 5. C

6. D
7. A
8. A
9. B
10. A
11. A
12. C
13. A
14. C
15. A
16. A
17. A
18. A
19. B
20. A

Раздел II. Реализация систем поддержки принятия решений

Глава 3. Информационные средства поддержки принятия управленческих решений

3.1 Виды информационных систем

Поддержка принятия решений с помощью информационных технологий, включая анализ и выработку альтернатив, осуществляется следующими методами [2]:

1. Информационный поиск – процесс поиска документальной неструктурированной информации. Процесс информационного поиска содержит в себе последовательность операций, которые направлены на сбор, обработку и предоставление необходимой информации лицам, заинтересованным в результате поиска. В общем случае поиск информации состоит из четырех этапов:

- определение (уточнение) информационной потребности и формулировка информационного запроса;
- установление совокупности возможных держателей информационных массивов (источников);
- извлечение информации из выявленных информационных массивов;
- ознакомление с полученной информацией и оценка результатов выполненного поиска.

Различают следующие виды информационного поиска:

- полнотекстовый – по всему содержимому документа;
- по метаданным – по неким атрибутам документа, поддерживаемым системой;
- по изображению – по содержанию изображения.

Информационный поиск осуществляется следующими методами:

- адресный (процесс поиска документов по чисто формальным признакам, указанным в запросе);
- семантический (процесс поиска документов по их содержанию);
- документальный (процесс поиска в хранилище информационно-поисковой системы первичных документов или в базе данных вторичных документов, соответствующих запросу пользователя);
- фактографический (процесс поиска фактов, соответствующих информационному запросу).

2. Интеллектуальный анализ данных, под которым понимается выявление скрытых взаимосвязей или закономерностей между переменными в больших массивах, представляющих собой набор необработанных данных. Такой анализ предусматривает решение задач классификации, моделирования и прогнозирования, кластеризации, сокращения описания, ассоциации, анализа отклонений, визуализации и

др.

3. Извлечение (поиск) знаний в базах данных – это процесс, проводимый с целью обнаружения в базах данных полезных знаний. Эти знания представляются в виде правил, закономерностей, связей между элементами данных, прогнозов и др. В процессе извлечения в базах данных основным инструментом поиска знаний являются аналитические технологии data mining, которые позволяют реализовать задачи классификации, кластеризации, прогнозирования, предсказания и т.д.

4. Процесс извлечения знаний в базах данных включает в себя выполнение последовательности операций, необходимых для поддержки аналитического процесса. К ним относятся: консолидация данных; подготовка анализируемых выборок данных (в том числе обучающих); очистка данных от факторов, мешающих их корректному анализу; трансформация – оптимизация данных для решения определенной задачи; анализ данных; интерпретация и визуализация результатов анализа, их применение в бизнес-приложениях.

5. Рассуждение на основе прецедентов является одним из методов поддержки принятия решений с помощью информационных технологий. В данном подходе поддержка принятия решений происходит при помощи прецедентов, представляющих собой случаи, происходившие ранее и являющиеся оправданием или примером для последующих аналогичных случаев. Вывод на основе прецедентов позволяет решить новую задачу, используя или адаптируя решение задачи, которая уже известна. Такие методы рассуждений предусматривают использование цикла рассуждения на основе прецедентов.

6. Имитационное моделирование – это метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно переиспользовать во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов.

7. Генетический алгоритм, который представляет собой эвристический алгоритм поиска. Данный алгоритм применяется для решения задач комбинирования и вариации искомых параметров с использованием механизмов, которые напоминают биологическую эволюцию, а также задач оптимизации и моделирования путем последовательного подбора. Генетический алгоритм является разновидностью эволюционных вычислений. Его отличительная особенность состоит в акценте на использование оператора "скрещивания", который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе. Генетические алгоритмы служат главным образом для поиска решений в очень больших, сложных пространствах поиска.

8. Искусственные нейронные сети представляют собой

математические модели и их аппаратные или программные реализации, построенные по принципу функционирования и организации биологических нейронных сетей – сетей нервных клеток живого организма. Первой моделью мозга, моделирующей процессы, протекающих в мозге при мышлении, был перцептрон. Впоследствии эти модели стали использовать в практических целях, в основном для решения задач прогнозирования. С точки зрения машинного обучения нейронная сеть представляет собой частный случай методов распознавания образов, дискриминантного анализа, методов кластеризации и т.п. С математической точки зрения обучение нейронных сетей – это многопараметрическая задача нелинейной оптимизации. С точки зрения кибернетики нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники. С точки зрения развития вычислительной техники и программирования нейронная сеть – способ решения проблемы эффективного параллелизма, а с точки зрения искусственного интеллекта она является основой философского течения коннективизма и главным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.

9. Искусственный интеллект – это наука и разработка интеллектуальных машин и систем, особенно интеллектуальных компьютерных программ, направленных на понимание человеческого интеллекта. При этом используемые методы не обязаны быть биологически правдоподобны. Ученые исследовали только некоторые механизмы интеллекта, поэтому под интеллектом в пределах данной науки имеют в виду вычислительную часть способности достигнуть целей. Если в основе работы системы лежат методы искусственного интеллекта, то говорят об интеллектуальной системе.

В зависимости от уровня признаков интеллектуальности интеллектуальные системы делятся на следующие виды (рисунок б):

1. Система с интеллектуальным интерфейсом представляют собой систему, обеспечивающую поиск неявной информации в тексте для произвольных запросов на ограниченном естественном языке или в базе данных.
2. Экспертная система – это система, позволяющая решать слабоформализуемые задачи на основе накапливаемого в базе знаний опыта работы экспертов в проблемной области.
3. Самообучающаяся система – это система, автоматически формирующая единицы знаний на основе примеров реальной практики.
4. Адаптивная система – система, которая автоматически изменяет данные алгоритма своего функционирования и иногда свою

структуру для сохранения или достижения необходимого состояния при изменении внешних условий.

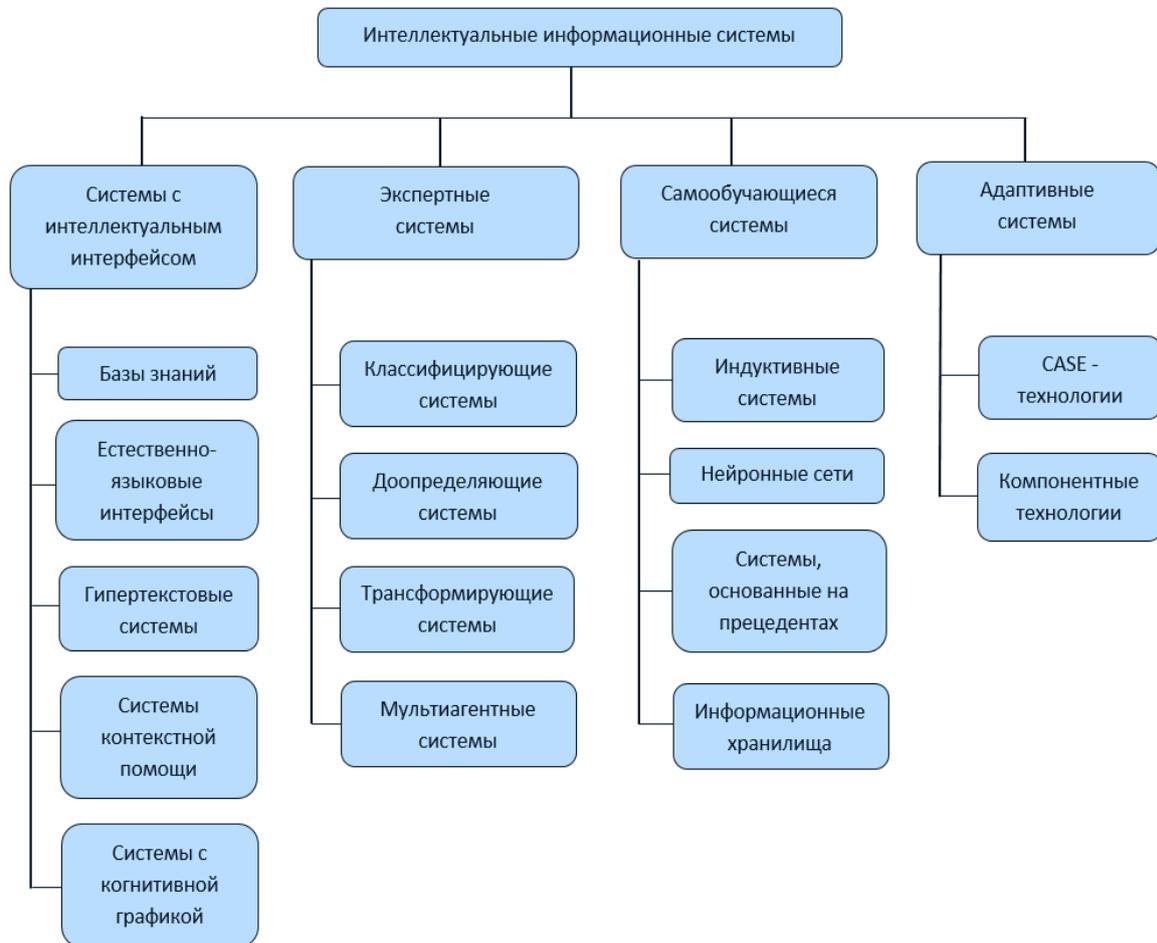


Рисунок 6 – Классификация интеллектуальных информационных систем

Рассмотрим классификацию экспертных систем.

По решаемой задаче экспертные системы можно представить как:

- интерпретация данных;
- диагностирование;
- мониторинг;
- проектирование;
- прогнозирование;
- сводное планирование;
- оптимизация;
- обучение;
- управление;
- ремонт;
- отладка.

По связи с реальным временем:

- статические – решающие задачи в условиях не изменяющихся во времени исходных данных и знаний;
- квазидинамические – интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени;
- динамические – решающие задачи в условиях изменяющихся во времени исходных данных и знаний.

3.2 Процесс создания систем

Система (в общем случае) – это совокупность элементов, связанных друг с другом и образующих определённую целостность, единство.

Информационная система – это система, предназначенная для хранения, поиска и обработки информации.

Экспертная система – компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации.

База знаний – это база данных, содержащая правила вывода и информацию о опыте и знаниях в некоторой предметной области.

В информационной сфере экспертные системы и базы знаний рассматриваются совместно, как модели поведения экспертов в определенной области знаний, с использованием процедур логического вывода и принятия решений, а база знаний – как совокупность фактов и правил логического вывода в выбранной предметной области.

При разработке экспертной системы принято делить ее на три основных модуля, как показано на рисунке 7.

- база знаний;
- машина логического вывода;
- интерфейс с пользователем.

Экспертные системы в информационных технологиях рассматривают вместе с базами знаний как модели поведения экспертов в определенной предметной области с использованием процедур принятия решения и логического вывода, а базы знаний – как набор фактов и правил логического вывода в определенной предметной области [5].

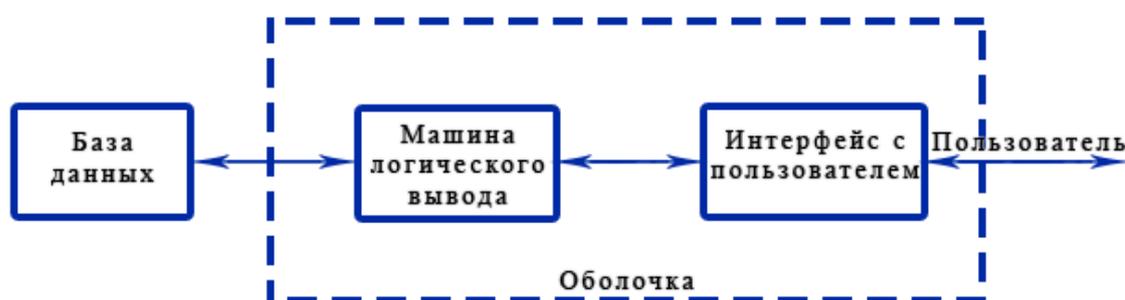


Рисунок 7 – Структура экспертной системы

Такие же задачи решает программное обеспечение, называемое «Мастер». Оно используется для упрощения интерактивного общения в

прикладных и системных программах. Особенностью таких программ является отсутствие базы знаний – все действия запрограммированы.

В состав экспертной системы входят следующие элементы [5]:

- интерфейс пользователя;
- пользователь;
- интеллектуальный редактор базы знаний;
- эксперт;
- инженер по знаниям;
- оперативная память;
- база знаний;
- решатель (механизм вывода);
- подсистема объяснений.

База знаний хранит правила анализа информации по проблеме, заданной пользователем. Система обрабатывает эту информацию и выводит рекомендации по разрешению проблемы.

База знаний состоит из двух составляющих:

- факты – статические сведения о предметной области;
- правила – набор инструкций, который позволяет выводить новые факты, исходя из уже известных.

Обычно факты описывают постоянные явления для данной предметной области. Система получает от пользователя во время работы и сохраняет в оперативную память данные, зависящие от условия конкретной задачи.

Рассмотрим методы представления знаний (рисунок 8):

- объектные методы представления знаний в большей степени ориентированы на представление структуры фактуального знания, а правила - операционного.
- логическая модель реализует и объекты, и правила с помощью предикатов первого порядка, является строго формализованной моделью с универсальным дедуктивным и монотонным методом логического вывода;
- продукционная модель позволяет осуществлять эвристические методы вывода на правилах и может обрабатывать неопределенности в виде условных вероятностей или коэффициентов уверенности, а также выполнять монотонный или немонотонный вывод;
- семантическая сеть отображает отношения объектов в виде графов, где узел представляет объект, а дуги определяют отношения между объектами;
- фреймовая модель как частный случай семантической сети использует для реализации операционного знания присоединенные процедуры;

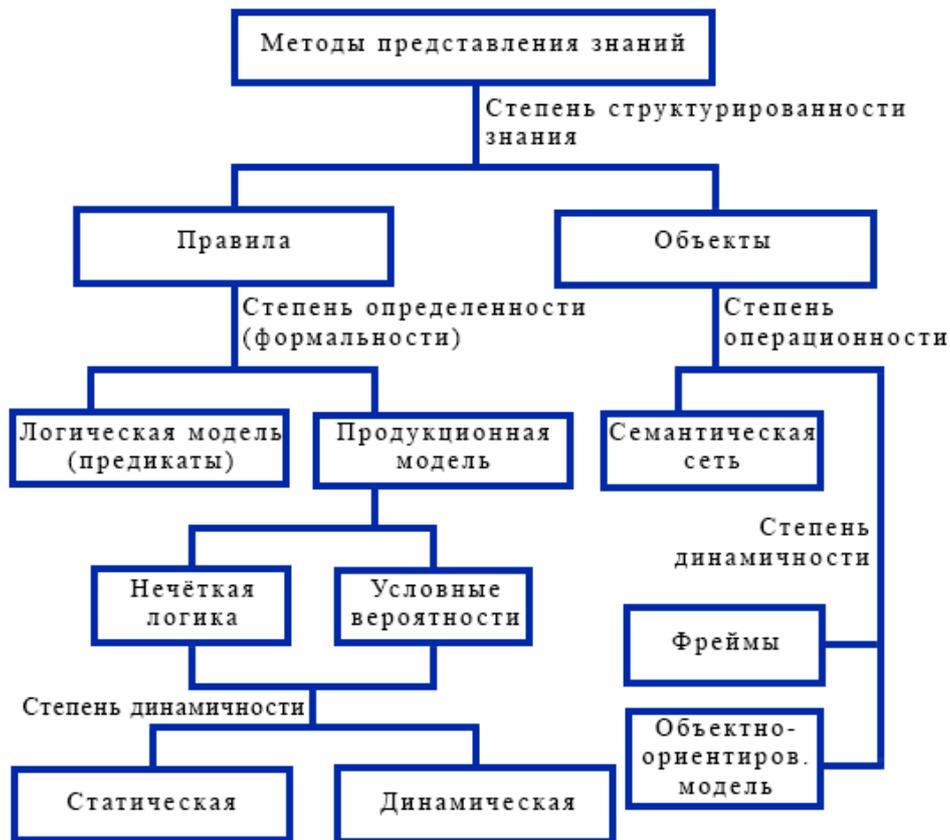


Рисунок 8 – Методы представления знаний

- объектно-ориентированная модель как развитие фреймовой модели, реализуя обмен сообщениями между объектами, в большей степени ориентирована на решение динамических задач и отражение поведенческой модели.

Этапы разработки экспертных систем [5]:

- этап идентификации проблем – определяются задачи, которые подлежат решению, выявляются цели разработки, определяются эксперты и типы пользователей;
- этап извлечения знаний – проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач;
- этап структурирования знаний – выбирается система и определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями;
- этап формализации – осуществляется наполнение экспертом базы знаний. В связи с тем, что основой системы являются знания, данный этап является наиболее важным и наиболее трудоемким

- этапом разработки системы. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном системе. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач;
- реализация системы – создается один или несколько прототипов системы, решающие требуемые задачи;
 - этап тестирования – производится оценка выбранного способа представления знаний в системе в целом.

Вопросы и задания

Тест для самоконтроля

1. Сколько шагов в этапе поиска информации?
Варианты ответа:
 - A. 3;
 - B. 4;
 - C. 2;
 - D. 5.
2. Какими методами осуществляется поддержка принятия решений с помощью информационных технологий?
Варианты ответа:
 - A. Машинное обучение;
 - B. Консолидация данных;
 - C. Имитационное моделирование;
 - D. Вывод информации.
3. Какие существуют виды информационного поиска?
Варианты ответа:
 - A. Полнотекстовый;
 - B. По изображению;
 - C. Адресный;
 - D. Семантический.
4. К какому типу алгоритмов поиска относится генетический алгоритм?
Варианты ответа:
 - A. Эволюционный;
 - B. Естественный;
 - C. Эвристический;
 - D. Стабилизирующий.
5. Для чего, в основном, служат генетические алгоритмы поиска?
Варианты ответа:
 - A. Для поиска решений в ограниченных пространствах поиска;

- В. Для поиска решений в очень больших пространствах поиска;
- С. Для поиска по изображению;
- Д. Для обучения.

6. Что было первой моделью мозга, моделирующей процессы, протекающие в нём?

Варианты ответа:

- А. Дендрит;
- В. Нейрон;
- С. Синапс;
- Д. Перцептрон.

7. Обучение нейронных сетей с математической точки зрения – это...

Варианты ответа:

- А. Передача входных данных в виде искаженного эталона;
- В. Математическая модель графа;
- С. Многопараметрическая задача нелинейной оптимизации;
- Д. Обнаружение внутренней взаимосвязи.

8. Что понимается под интеллектуальным анализом данных?

Варианты ответа:

- А. Поиск новых связей;
- В. Выявление скрытых взаимосвязей или закономерностей между переменными;
- С. Выявление связей или закономерностей между данными;
- Д. Поиск закономерностей между записями.

9. Что производится при обнаружении в базах данных полезных знаний?

Варианты ответа:

- А. Сортировка данных;
- В. Просмотр данных;
- С. Анализ данных;
- Д. Извлечение данных.

10. В виде чего предоставляются данные?

Варианты ответа:

- А. Правил;
- В. Прогнозов;
- С. Закономерностей;
- Д. Списков.

11. Какие существуют виды интеллектуальных систем?

Варианты ответа:

- А. Распорядительная система;
- В. Экспертная система;
- С. Адаптивная система;
- Д. Прогнозируемая система.

12. Какая система обеспечивает поиск неявной информации в тексте для произвольных запросов на ограниченном языке или в базе данных?

Варианты ответа:

- A. Самообучающаяся система;
- B. Экспертная система;
- C. Система с интеллектуальным интерфейсом;
- D. Экспертная система.

13. Какие существуют виды самообучающихся систем?

Варианты ответа:

- A. Базы знаний;
- B. Доопределяющие системы;
- C. Индуктивные системы;
- D. Информационные хранилища.

14. Как по решаемой задаче можно представить экспертную систему?

Варианты ответа:

- A. Мониторинг;
- B. Учёт;
- C. Отчёт;
- D. Ремонт.

15. Как характеризуются квазидинамические системы?

Варианты ответа:

- A. Решающие задачи в условиях не изменяющихся во времени исходных данных и знаний;
- B. Интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени;
- C. Интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом данных;
- D. Решающие задачи в условиях изменяющихся во времени исходных данных и знаний.

16. С помощью чего логическая модель реализует правила?

Варианты ответа:

- A. Предикатов первого порядка;
- B. Синапсов первого порядка;
- C. Логической структуры;
- D. Физической модели.

17. Что из перечисленного входит в состав экспертной системы?

Варианты ответа:

- A. Инженер по знаниям;
- B. Механизм ввода;
- C. Машина логического вывода;
- D. Администратор.

18. В виде чего семантическая сеть отображает отношения объектов?

Варианты ответа:

- A. Текст;
- B. Изображение;
- C. Граф;
- D. Набор единиц и нулей.

19. На каком этапе создания экспертной системы выполняется содержательный анализ проблемной области?

Варианты ответа:

- A. Этап формализации;
- B. Этап идентификации проблем;
- C. Этап решения;
- D. Этап извлечения знаний.

20. На каком этапе создания экспертной системы осуществляется наполнение экспертом базы знаний?

Варианты ответа:

- A. Этап формализации;
- B. Этап реализации системы;
- C. Этап извлечения знаний;
- D. Этап тестирования.

Ответы на вопросы теста

- 1. B
- 2. C, B
- 3. A, B
- 4. C
- 5. B
- 6. D
- 7. C
- 8. B
- 9. D
- 10. A, B, C
- 11. B, C
- 12. C
- 13. C, D
- 14. A, D
- 15. B
- 16. A
- 17. A, C
- 18. C
- 19. D
- 20. A

Глава 4. Продукционные системы

4.1 Построение дерева принятия решений и системы правил

Начальным этапом в процессе проектирования экспертной системы продукционного типа является разработка дерева принятия решения [10]. Дерево принятия решения представляет собой граф особого вида, характеризующий последовательность действий и проверок при решении задачи. Проверки условий помещаются в овал, действия обозначаются прямоугольником. Дерево принятия решения отражает всю логику принятия решения в заданной предметной области. После построения дерева принятия решения можно определить систему правил.

Пример дерева принятия решения для экспертной системы в области контроля знаний по информатике представлен на рисунке 9. Дерево описывает структуру самого процесса (в два этапа, с контролем времени) и логику принятия решения (выставление баллов, сравнение с порогом, введенным экспертом).

На основе дерева принятия решения разрабатывается система правил.

Список правил для дерева, приведенного на рисунке 9, может быть сформирован следующим образом:

```
RULE: R1
  if (gotov1 == true) and (etap == 1) and (t1 <
time1) and (k1 < kolvol) and (otvet == true) then (k1
= k1 + 1)
```

```
RULE: R2
  if (gotov1 == true) and (etap == 1) and (t1 <
time1) and (k1 < kolvol) and (otvet == false) then
(k1 = k1 + 1) and (oshibki = oshibki + 1)
```

```
RULE: R3
  if (gotov1 == true) and (etap == 1) and (t1 <
time1) and (k1 == kolvol) and (oshibki == 0) then
(mark1 = 5) and (etap = 2)
```

```
RULE: R4
  if (gotov1 == true) and (etap == 1) and (t1 <
time1) and (k1 == kolvol) and (oshibki == 1) then
(mark1 = 4) and (etap = 2)
```

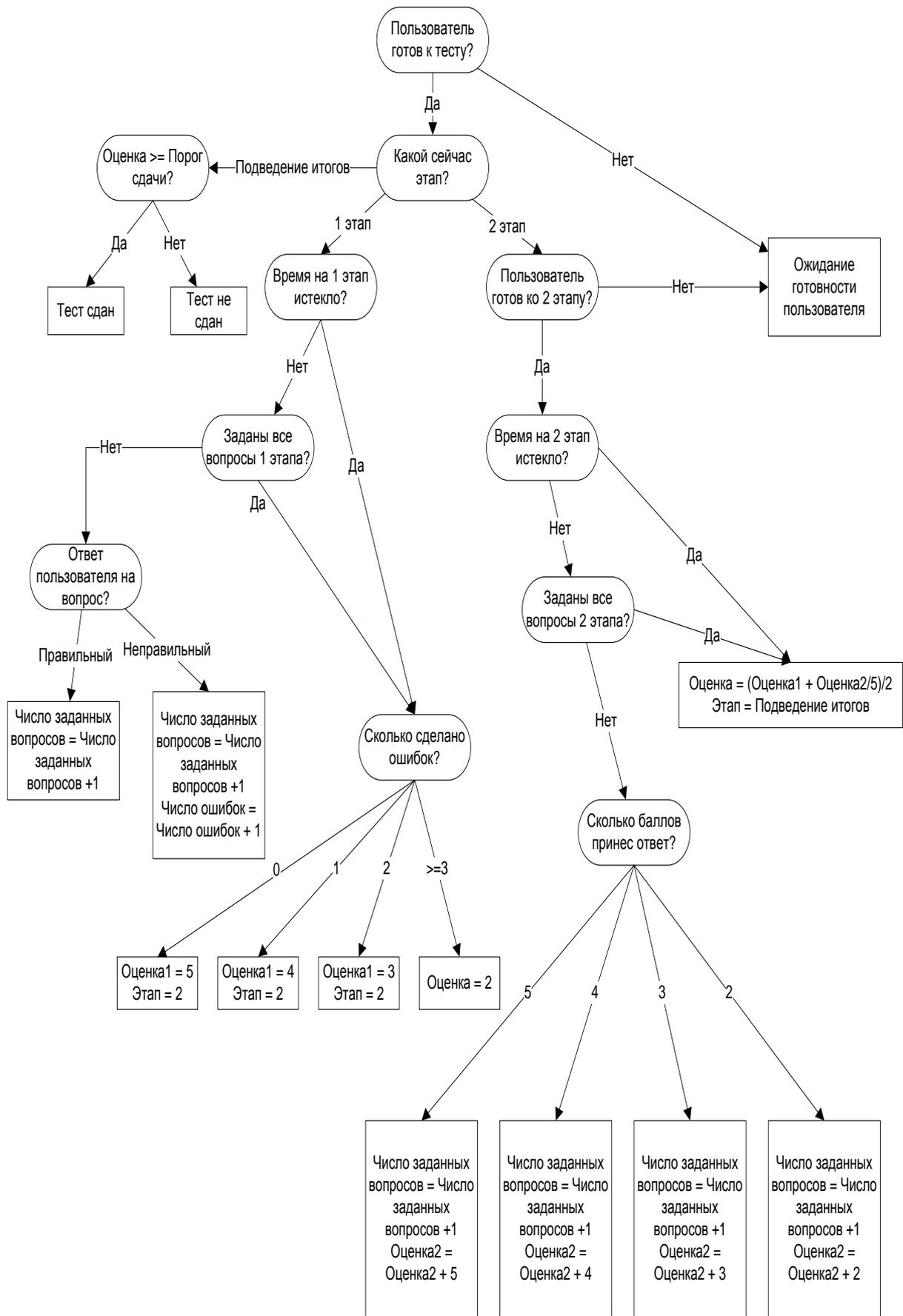


Рисунок 9 – Дерево принятия решений

RULE: R5
if (gotov1 == true) and (etap == 1) and (t1 < time1) and (k1 == kolvo1) and (oshibki == 2) then (mark1 = 3) and (etap = 2)

RULE: R6
if (gotov1 == true) and (etap == 1) and (t1 < time1) and (k1 == kolvo1) and (oshibki > 2) then (mark = 2)

RULE: R7
if (gotov1 == true) and (etap == 1) and (t1 >= time1) and (k1 == kolvo1) and (oshibki == 0) then (mark1 = 5) and (etap = 2)

RULE: R8
if (gotov1 == true) and (etap == 1) and (t1 >= time1) and (k1 < kolvo1) and (oshibki == 1) then (mark1 = 4) and (etap = 2)

RULE: R9
if (gotov1 == true) and (etap == 1) and (t1 >= time1) and (k1 < kolvo1) and (oshibki == 2) then (mark1 = 3) and (etap = 2)

RULE: R10
if (gotov1 == true) and (etap == 1) and (t1 >= time1) and (k1 < kolvo1) and (oshibki > 2) then (mark = 2)

RULE: R11
if (gotov2 == true) and (etap == 2) and (t2 < time2) and (k2 < kolvo2) and (balli == 5) then (k2 = k2 + 1) and (mark2 = mark2 + 5)

RULE: R12
if (gotov2 == true) and (etap == 2) and (t2 < time2) and (k2 < kolvo2) and (balli == 4) then (k2 = k2 + 1) and (mark2 = mark2 + 4)

RULE: R13
if (gotov2 == true) and (etap == 2) and (t2 < time2) and (k2 < kolvo2) and (balli == 3) then (k2 = k2 + 1) and (mark2 = mark2 + 3)

```
RULE: R14
  if (gotov2 == true) and (etap == 2) and (t2 <
time2) and (k2 < kolvo2) and (balli == 2) then (k2 =
k2 + 1) and (mark2 = mark2 + 2)
```

```
RULE: R15
  if (gotov2 == true) and (etap == 2) and (t2 <
time2) and (k2 == kolvo2) then (mark = ((mark1 +
(mark2 / kolvo2)) / 2))
```

```
RULE: R16
  if (gotov2 == true) and (etap == 2) and (t2 >=
time2) and (k2 < kolvo2) then (mark = ((mark1 +
(mark2 / k2)) / 2))
```

```
RULE: R17
  if (gotov1 == false) then (gotov1 = unknown)
```

```
RULE: R18
  if (gotov2 == false) then (gotov2 = unknown)
```

```
RULE: R19
  if (mark >= porog) then (result = true)
```

```
RULE: R20
  if (mark < porog) then (result = false)
```

Список правил приведен в условном виде:

```
RULE: <имя правила>
  IF <конъюнкция условий> THEN <конъюнкция
действий>
```

В приведенных правилах используются переменные, перечисленные в таблице 1, из них:

- переменные porog, time1 и time2 задаются экспертом.
- переменные otvet, balli, gotov1, gotov2 получают значения в результате действий пользователя, проходящего контроль знаний.
- остальные переменные получают значения в результате выполнения правил.
- целевой переменной является result.

Таблица 1 – Список переменных набора правил

№	Имя переменной	Значение по умолчанию	Описание переменной
1	gotov1	unknown	Готовность пользователя к тесту
2	gotov2	unknown	Готовность пользователя ко второму этапу
3	etap	1	Текущий этап
4	time1	2	Время на первый этап
5	time2	5	Время на второй этап
6	t1	unknown	Текущее время первого этапа
7	t2	unknown	Текущее время второго этапа
8	mark1	unknown	Оценка за первый этап
9	mark2	unknown	Оценка за второй этап
10	mark	unknown	Оценка за тест
11	kolvo1	5	Количество вопросов первого этапа
12	kolvo2	5	Количество вопросов второго этапа
13	k1	0	Текущий вопрос первого этапа
14	k2	0	Текущий вопрос второго этапа
15	otvet	unknown	Ответ пользователя на вопрос первого этапа
16	oshibki	0	Число ошибок, сделанных пользователем на первом этапе
17	balli	unknown	Баллы за вопрос второго этапа
18	porog	3	Порог сдачи теста, заданный экспертом
19	result	unknown	Результат сдачи теста

4.2 Выбор алгоритма логического вывода

Алгоритм – четко определенная последовательность действий, которая приводит к решению поставленной задачи за конечное число шагов. В качестве алгоритма логического вывода в данном примере использован алгоритм прямой цепочки рассуждений. Идея прямой цепочки рассуждений заключается в последовательном применении правил по порядку, с возвратом после просмотра всех правил снова в начало, до тех пор, пока не будет получено значение целевой переменной.

Существует много вариаций алгоритмов прямой цепочки рассуждения, однако их суть состоит в следующем: в результате последовательного тестирования часть правил инициализируется, что ведет к изменению состояния системы. Процесс логического вывода

продолжается до тех пор, пока целевой переменной не присваивается значение, или до тех пор, пока ее значение не перестанет меняться. Однако, если такое произойдет, значит – правила несовместны.

Исходными данными для реализации алгоритма являются правила. Дополнительно к ним ведутся три таблицы и одна запись:

1. Текущая очередь вывода – таблица, в которой хранятся имена переменных с неизвестными значениями.
2. Таблица текущих значений переменных – таблица, в которой хранятся имена всех переменных с соответствующими значениями.
3. Таблица структуры набора правил:

Имя переменной	№ правила	№ условия	Следствие	Текущее значение

4. Запись текущего состояния системы:

Имя переменной	№ правила	№ условия	Текущее значение

Алгоритм прямой цепочки рассуждения (рисунок 10) может быть описан следующей последовательностью шагов:

Шаг 1. Создать таблицы и запись текущего состояния система. Заполнить таблицу переменных, занести в очередь целевую переменную. Организовать два цикла: по правилам и внутри каждого правила.

Шаг 2. Рассматривается первый конъюнктивный член первого правила. Если значения переменных этого члена известны из таблицы текущих значений переменных, то проверить его на истинность. Если неизвестны, то проверить, нельзя ли запросить у пользователя. Если нельзя, то занести переменную в очередь и перейти к следующему правилу. Если можно запросить, то переменная запрашивается и обновляются все таблицы. Если конъюнктивный член истинен, то перейти к следующему конъюнктивному члену, иначе – перейти к следующему правилу.

Шаг 3. Если конъюнкция в условии правила истинна, то выполняется правая часть правила. Если при активизации правила достигается значение целевой переменной, то процесс завершается. Если какие-либо переменные приняли значения, то они удаляются из очереди. В конце логического вывода очередь должна быть пуста.

Схема алгоритма прямой цепочки рассуждений представлена на рисунке 10.

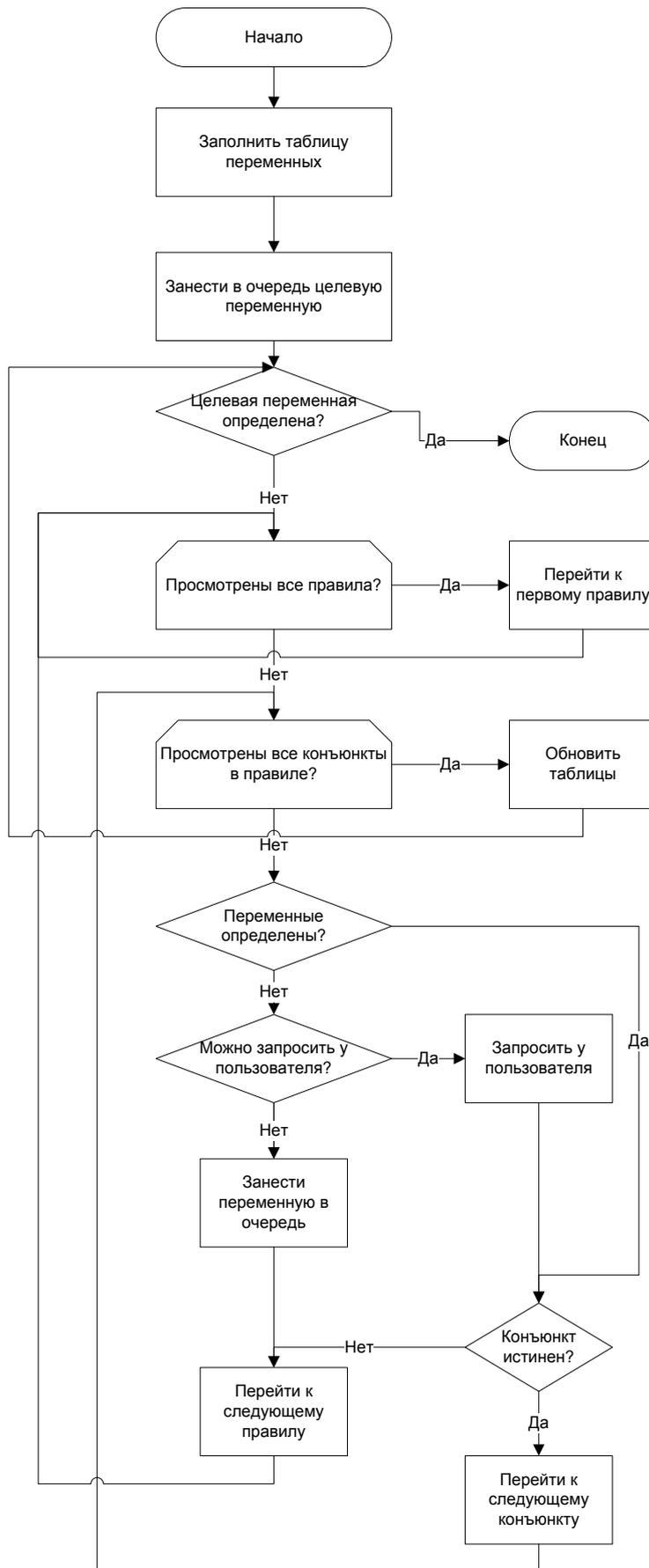


Рисунок 10 – Схема алгоритма прямой цепочки рассуждения

4.3 Проектирование структур данных

На этом этапе создания продукционной экспертной системы для поддержки принятия решений необходимо выполнить проектирование структур данных с использованием объектно-ориентированного подхода, при котором в отдельные классы выделяются множества объектов, обладающих общими атрибутами, операциями, отношениями и семантикой.

Выделенные классы для рассматриваемого примера представлены на рисунках 11–13.

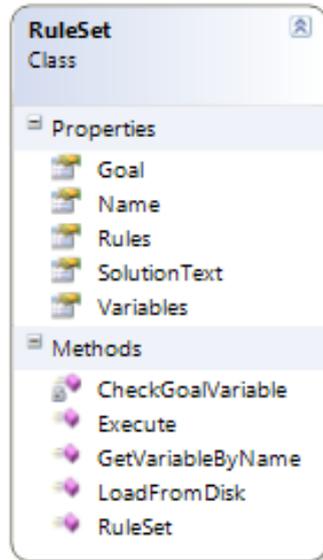


Рисунок 11 – Структура класса RuleSet

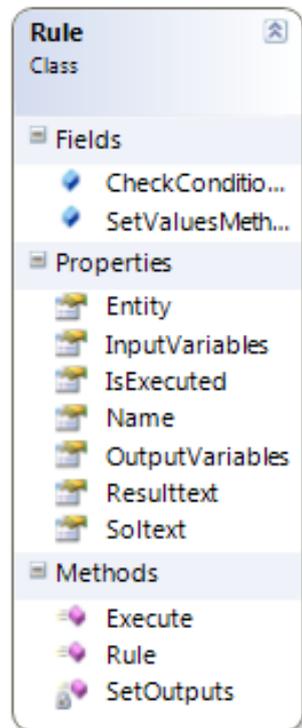


Рисунок 12 – Структура класса Rule

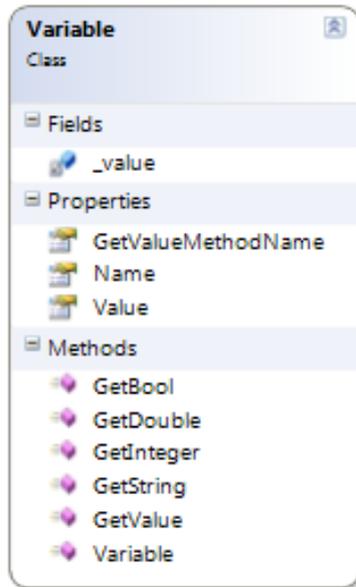


Рисунок 13 – Структура класса Variable

Класс RuleSet представляет собой контейнер набора правил. Он имеет свойства:

- Goal – целевая переменная;
- Name – имя набора правил;
- Rules – список правил;
- SolutionText – объяснение решения;
- Variables – таблица переменных.

Также в нем описаны необходимые методы:

- CheckGoalVariable – проверка установки значения целевой переменной;
- Execute – выполнение набора правил;
- GetVariableByName – получение переменной из таблицы по ее имени;
- LoadFromDisk – загрузка набора правил с диска.

Класс Rule представляет собой одно правило. В нем содержатся свойства:

- Entity – сущность правила, посылка и следствие;
- InputVariables – переменные посылки;
- IsExecuted – флаг выполнения правила;
- Name – имя правила;
- OutputVariables – переменные следствия;
- Resulttext – результаты применения правила;
- Soltext – объяснение решения для правила.

Методы класса Rule:

- Execute – выполнение правила;
- SetOutputs – обновление таблиц, если правило выполнено.

Класс `Variable` представляет собой переменную. Значение переменной хранится в свойстве класса `Object`, таким образом, переменная может быть любого типа, будь то строковый, числовой, булевский и любой другой. Свойства класса `Variable`:

- `GetValueMethodName` – метод, с помощью которого можно спросить пользователя;
- `Name` – имя переменной;
- `Value` – значение переменной.

Методы класса `Variable`:

- `GetBool` – приводит переменную к булевскому типу;
- `GetDouble` – приводит переменную к числовому типу;
- `GetInteger` – приводит переменную к целочисленному типу;
- `GetString` – приводит переменную к строковому типу;
- `GetValue` – метод получения значения переменной.

4.4 Программная реализация системы

Для реализации системы используется язык программирования высокого уровня `C#`. `C#` является современным объектно-ориентированным языком, созданным специально для программирования под платформу `.NET`. Вместе с данной платформой он предоставляет программисту такие удобства, как богатая библиотека готовых классов, автоматическая «сборка мусора» – удаление неиспользуемых объектов из памяти, легкое манипулирование данными с использованием `Language Integrated Query` – `SQL`-подобных конструкций языка, механизм управления исключительными ситуациями и много другое.

Для разработки системы используется среда `Microsoft Visual Studio`, которая предоставляет богатые возможности по ускоренному созданию пользовательского интерфейса, обладает всем необходимым набором функций, таких как подсветка синтаксиса, компиляция, отладка с возможностью пошагового выполнения. Также в ней реализована встроенная поддержка `Unit`-тестирования и коллективной работы над проектом, организовано удобное управление исходными файлами. `Visual Studio` позволяет наиболее полно использовать возможности выбранного языка программирования `C#` и платформы `.NET`.

Программная система имеет два основных режима:

1. Режим эксперта. В режиме эксперта можно:

- добавлять/редактировать/удалять вопросы;
- добавлять/редактировать/удалять пользователей системы;
- проверять успеваемость и просматривать протоколы;
- осуществлять настройку времени тестирования и порога прохождения теста;
- просматривать правила.

2. Режим консультации. В режиме консультации:

- система задает 5 простых вопросов;
- система задает 5 дополнительных вопросов;
- ведется контроль времени ответов;
- ведется учет книг для подготовки;
- ведется протокол;
- система отображает результаты теста.

При запуске программы на экране появляется форма авторизации, изображенная на рисунке 14. Основное назначение этой формы – обеспечить разграничение полномочий пользователя и эксперта.

Рисунок 14 – Форма авторизации

После ввода логина и пароля система автоматически определяет в каком режиме продолжить работу – в режиме пользователя или в режиме эксперта.

На рисунке 15 приведена главная экранная форма эксперта. В форме имеется меню, отвечающее за всю функциональность режима эксперта.

Текст вопроса	Книга для подготовки	Страницы в книге	Тип вопроса
Сколькими битами можно закодир...	Роберт Диксон - Информация в...	23	YesNo
Чему равен 1 байт?	Николаус Вирт - Программиров...	275	YesNo
Массовое производство персонал...	Р.Н. Петров - История ПК	12, 13	Extended
За основную единицу измерения к...	Н.И. Лисицин - Информация и ...	18	Extended
В детской игре "Угадай число" пер...	Дейкстра - Алгоритмы и структ...	56	Extended
Как записывается десятичное чис...	Луи Брю - Аналитическая инфо...	46	Extended
Производительность работы комп...	Пособие по информатике для 1...	8, 9	Extended
Что изменяет операция присваива...	Фаронов - Турбо Паскаль	54,55	YesNo
Что такое HTML?	Каряка А.Н. - Настольная книга...	4	YesNo
Модель объекта должна быть адек...	Коварцев - Курс лекций по Мат...	18	YesNo
Класс является экземпляром объек...	Лезина И.В. - Курс лекции по Java	5	YesNo
Какие из перечисленных объектов с...	Курс лекций по С++	18	Extended

Рисунок 15 – Форма эксперта

Меню «Файл» позволяет сохранить изменения в базе знаний и выйти из режима эксперта.

В центре формы находится таблица с вопросами, которые автоматически загружаются из базы знаний. Для добавления/изменения/удаления вопроса используются одноименные кнопки. Кроме того, функциональность по работе с вопросами дублируется в главном меню «База знаний». Добавление и редактирование вопросов осуществляется с помощью формы, изображенной на рисунке 16. Редактирование вариантов ответа осуществляется с помощью формы, изображенной на рисунке 17.

Редктирование вопроса

Тип вопроса

С вариантами ответа "Да"/"Нет"

С несколькими вариантами ответа

Книга для подготовки

Название Р.Н. Петров - История ПК

Страницы 12, 13

Вопрос

Массовое производство персональных компьютеров началось в ...

Варианты ответа

Вариант ответа	Оценка
в 40е годы.	2
в 50е годы.	2
в 80е годы.	4
в 90е годы.	5

Добавить

Изменить

Удалить

Сохранить

Отмена

Рисунок 16 – Форма редактирования вопроса

Редктирование варианта ответа

Вариант ответа

в 40е годы.

Баллы за ответ/Правильность

2

Сохранить

Отмена

Рисунок 17 – Форма редактирования варианта ответа

Меню «Экспертная система» позволяет просмотреть список правил системы, настроить параметры прохождения теста, управлять пользователями системы и просматривать протоколы сдачи тестов. Форма

настроек тестирования представлена на рисунке 18. Эксперт может редактировать время проведения этапов тестов и порог успешной сдачи теста.

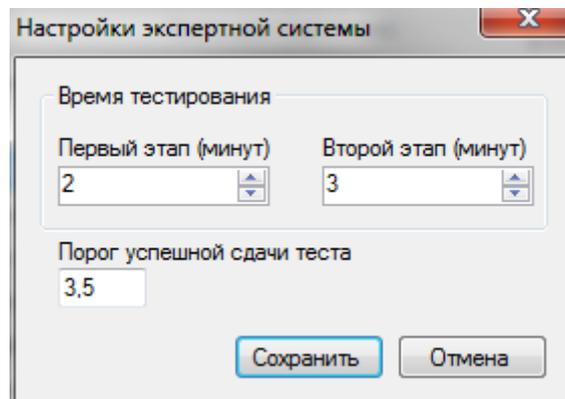


Рисунок 18 – Форма настройки тестирования

На рисунке 19 изображена форма просмотра правил, согласно которым функционирует экспертная система. На рисунке 20 изображена форма работы с пользователями системы. В ней можно просматривать регистрационную информацию пользователей, создать нового пользователя (рисунок 21), удалить пользователя.

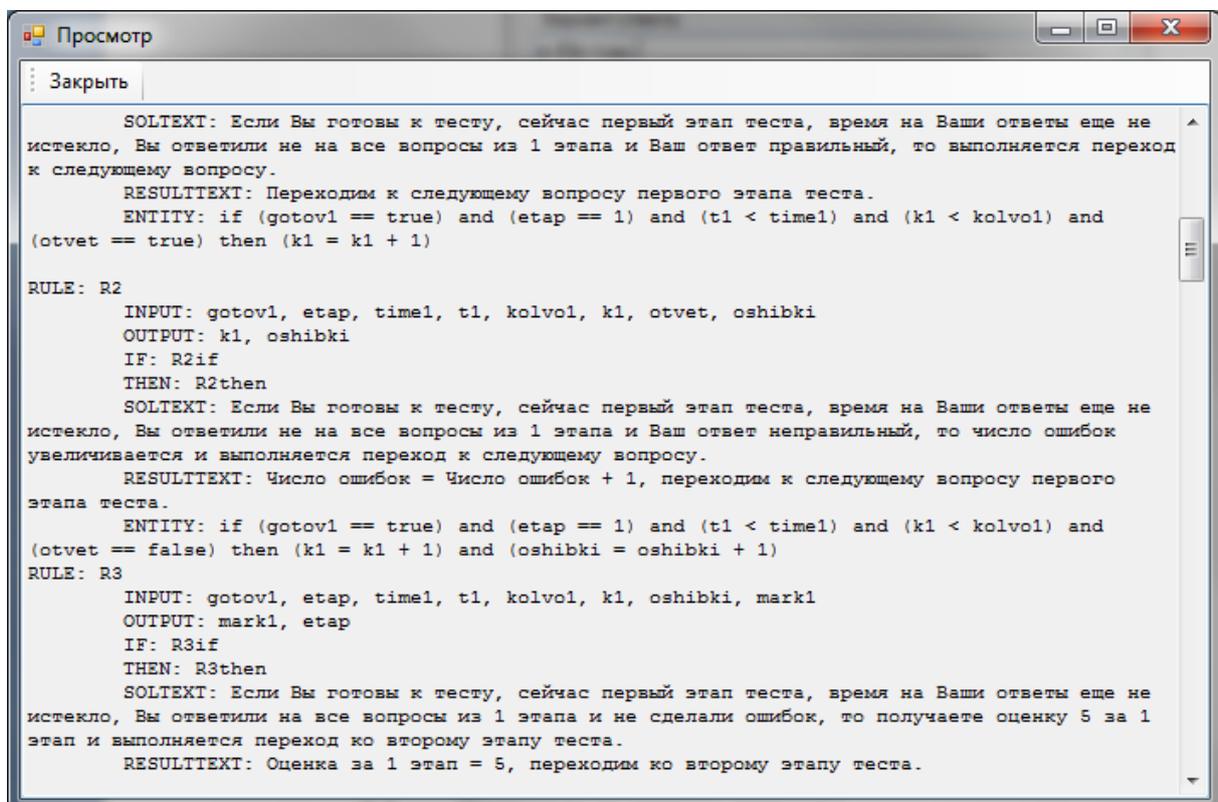


Рисунок 19 – Форма просмотра правил

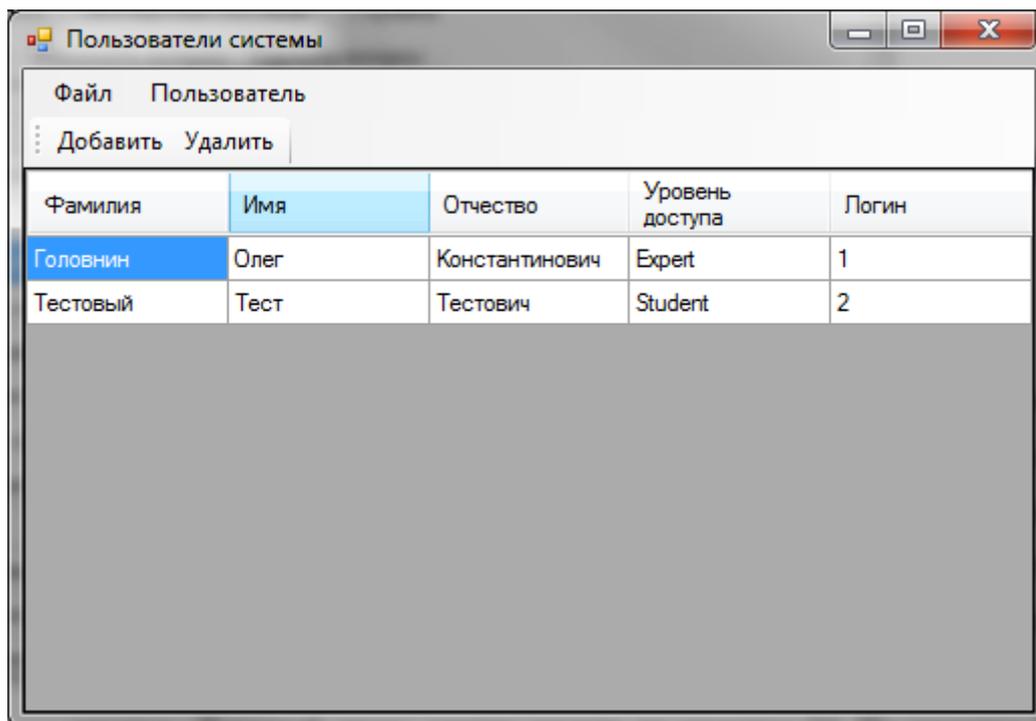


Рисунок 20 – Форма работы с пользователями системы

Рисунок 21 – Форма добавления пользователя системы

На рисунке 22 изображена форма просмотра результатов сдачи теста. В ней отображаются все пользователи, сдававшие тест. По каждому пользователю ведется протокол, который можно просмотреть. Форма просмотра протокола изображена на рисунке 23.

В режиме консультации весь экран занимает главная форма (изображенная на рисунке 24), основной целью которой является помеха в подглядывании в другие открытые окна при сдаче теста. Сразу при запуске формы происходит запуск консультации.

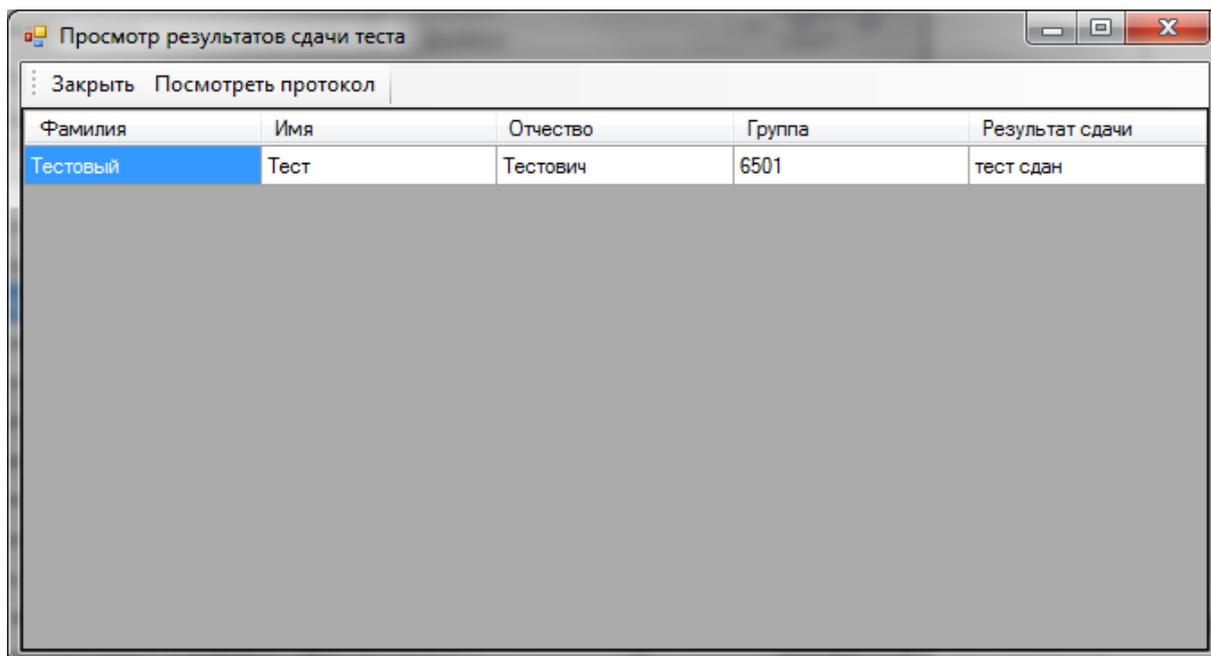


Рисунок 22 – Форма просмотра результатов сдачи теста

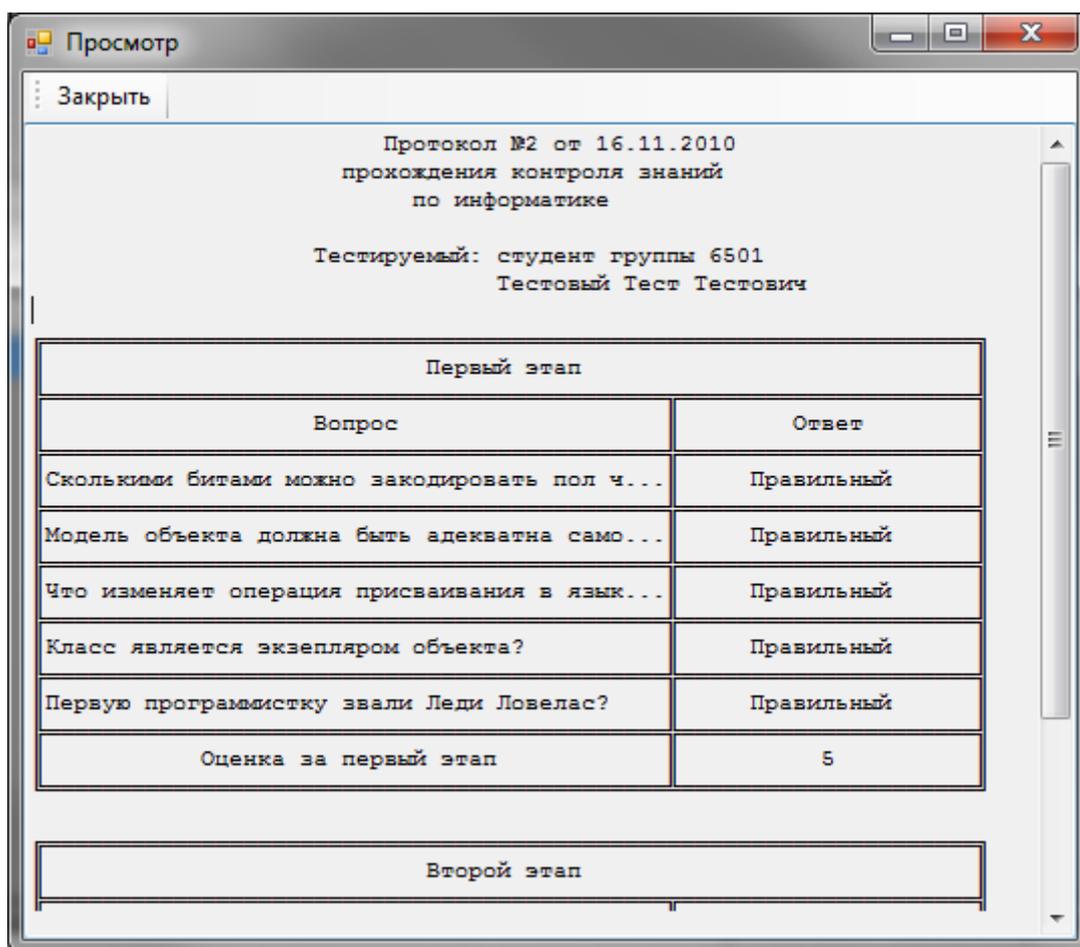


Рисунок 23 – Форма просмотра протокола

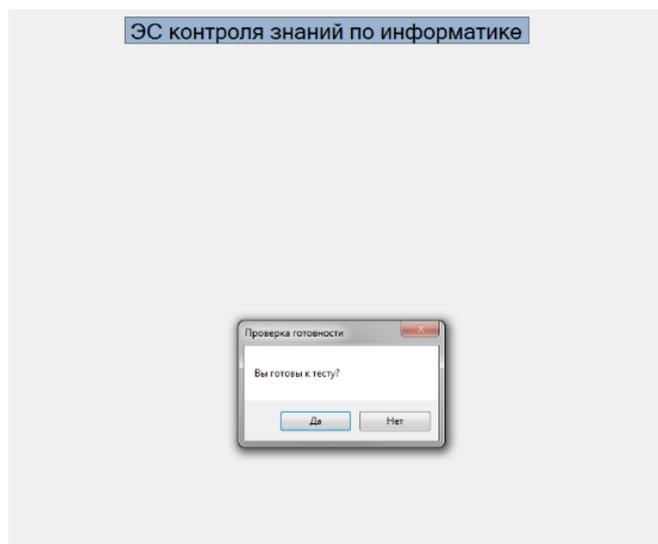


Рисунок 24 – Главная форма режима консультации

На рисунке 25 изображен пример формы с просьбой ответить на основной вопрос теста. На рисунке 26 изображен такой же пример для дополнительного вопроса. В обеих формах отображается краткая справка по ответу на вопрос. Во всех формах есть ответ «Не знаю», позволяющий пользователю продолжить отвечать на вопросы, если он не знает ответа на поставленный вопрос. После окончания консультации пользователь видит форму, представленную на рисунке 27. В ней отображаются набранные пользователем баллы, результат теста. В форме имеются кнопки, с помощью которых пользователь может посмотреть объяснение решения, литературу (рисунок 28), протокол и выйти из приложения.

Рисунок 25 – Форма с простым вопросом

Справочная информация по первому этапу

Сейчас проводится второй этап теста. В нем Вам предлагается ответить на вопросы, которые имеют несколько вариантов ответа. Правильными, в той или иной степени, могут быть несколько вариантов ответа. В зависимости от выбранных Вами вариантов, начисляются баллы за ответ. В поле "Вопрос" отображен вопрос, на который Вам необходимо дать ответ. В поле "Варианты ответа" отображены возможные варианты ответа на этот вопрос. Для ответа на вопрос, вы должны указать выбранные вами варианты с помощью мыши и нажать кнопку "Ответить". В случае, если вы не знаете ответа на вопрос, можно воспользоваться кнопкой "Не знаю".

Вопрос

В детской игре "Угадай число" первый участник загадал целое число в промежутке от 1 до 8. Второй участник задает вопросы: "Загаданное число больше числа _?" Какое количество вопросов при правильной стратегии (интервал чисел в каждом вопросе делится пополам) гарантирует угадывание?

Варианты ответа

2
 1
 4
 3
 8

Не знаю

Ответить

Рисунок 26 – Форма с дополнительным вопросом

Количество набранных баллов за тест: **4,7**
 Результат теста: **тест сдан**

Показать объяснение

Показать книги

Показать протокол

Выход

Рисунок 27 – Форма с результатами теста

Просмотр

Закреть

Книга: Курс лекций по C++, страницы: 18
 Книга: Еленев Д.В. - Курс лекций по сетям ЭВМ, страницы: 15-113
 Книга: Н.И. Лисицин - Информация и метрология, страницы: 18

Рисунок 28 – Форма с рекомендуемой литературой

4.5 Тестирование системы

Для демонстрации работы программы приведем тестовый пример. Войдем в систему под логином «2», который соответствует студенту «Тестовый Тест Тестович» из группы «6501». Сразу по нажатию кнопки «ОК» осуществляется вход в режим консультации и запускается на исполнение набор правил:

Система задает вопрос: «Вы готовы к тесту?»

Ответ: «Да».

Начинается первый этап теста:

1. Вопрос: «Что такое HTML?»

Ответ: «Язык разметки гипертекста»

2. Вопрос: «Что изменяет операция присваивания в языке Pascal?»

Ответ: «Значение переменной»

3. Вопрос: «Модель объекта должна быть адекватна самому объекту?»

Ответ: «Да»

4. Вопрос: «Первую программистку звали Леди Ловелас?»

Ответ: «Не знаю»

5. Вопрос: «Алгоритм прямой цепочки рассуждения использует ...?»

Ответ: «Стек»

Первый этап заканчивается.

Система задает вопрос: «Вы готовы ко второй части теста?»

Ответ: «Да»

Начинается второй этап теста:

1. Вопрос: «Производительность работы компьютера (быстрота выполнения операций) зависит от...»

Ответ: «частоты процессора», «объема оперативной памяти»

2. Вопрос: «Что из этого не относится к протоколам обмена информацией?»

Ответ: «HTML», «UML»

3. Вопрос: «В детской игре "Угадай число" первый участник загадал целое число в промежутке от 1 до 8. Второй участник задает вопросы: "Загаданное число больше числа _?" Какое количество вопросов при правильной стратегии (интервал чисел в каждом вопросе делится пополам) гарантирует угадывание?»

Ответ: «3»

4. Вопрос: «Какие из перечисленных свойств относятся к ООП?»

Ответ: «Не знаю»

5. Вопрос: «За основную единицу измерения количества информации принят...»

Ответ: «1 бит», «1 байт»

В результате прохождения теста система выдает сообщение о том, что тест сдан и набрано 3,5 балла.

По запросу система выводит источники, которые рекомендуется использовать для подготовки:

Книга: Тацкие станки и машины Тьюринга, страницы: 13

Книга: Курс лекций по основам ЭС, страницы: 15

Книга: Курс лекций по сетям ЭВМ, страницы: 15-113

Книга: Курс лекций по С++, страницы: 18

Книга: Информация и метрология, страницы: 18

Обеспечивается возможность посмотреть объяснение результатов консультации:

Консультация с системой: ЭС контроля знаний по информатике

Начало консультации: 20.01.2020 10:04:54

R1: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы из 1 этапа и Ваш ответ

правильный, то выполняется переход к следующему вопросу.

Результат: Переходим к следующему вопросу первого этапа теста.

R1: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы из 1 этапа и Ваш ответ правильный, то выполняется переход к следующему вопросу.

Результат: Переходим к следующему вопросу первого этапа теста.

R1: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы из 1 этапа и Ваш ответ правильный, то выполняется переход к следующему вопросу.

Результат: Переходим к следующему вопросу первого этапа теста.

R2: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы из 1 этапа и Ваш ответ неправильный, то число ошибок увеличивается и выполняется переход к следующему вопросу.

Результат: Число ошибок = Число ошибок + 1, переходим к следующему вопросу первого этапа теста.

R2: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы из 1 этапа и Ваш ответ неправильный, то число ошибок увеличивается и выполняется переход к следующему вопросу.

Результат: Число ошибок = Число ошибок + 1, переходим к следующему вопросу первого этапа теста.

R5: Если Вы готовы к тесту, сейчас первый этап теста, время на Ваши ответы еще не истекло, Вы ответили на все вопросы из 1 этапа и сделали две ошибки, то получаете оценку 3 за 1 этап и выполняется переход ко второму этапу теста.

Результат: Оценка за 1 этап = 4, переходим ко второму этапу теста.

R11: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы и Ваш ответ заслуживает 5 баллов, то Вам прибавляется 5 баллов и выполняется переход к следующему вопросу.

Результат: Баллы за 2 этап = Баллы за 2 этап + 5, переходим к следующему вопросу второго этапа теста.

R12: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы и Ваш ответ заслуживает 4 балла, то Вам прибавляется 4 балла и выполняется переход к следующему вопросу.

Результат: Баллы за 2 этап = Баллы за 2 этап + 4, переходим к следующему вопросу второго этапа теста.

R11: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы и Ваш ответ заслуживает 5 баллов, то Вам прибавляется 5 баллов и выполняется переход к следующему вопросу.

Результат: Баллы за 2 этап = Баллы за 2 этап + 5, переходим к следующему вопросу второго этапа теста.

R14: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы и Ваш ответ заслуживает 2 балла, то Вам прибавляется 2 балла и выполняется переход к следующему вопросу.

Результат: Баллы за 2 этап = Баллы за 2 этап + 2, переходим к следующему вопросу второго этапа теста.

R12: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили не на все вопросы и Ваш ответ заслуживает 4 балла, то Вам прибавляется 4 балла и выполняется переход к следующему вопросу.

Результат: Баллы за 2 этап = Баллы за 2 этап + 4, переходим к следующему вопросу второго этапа теста.

R15: Если сейчас 2 этап теста, вы готовы ко 2 этапу, время на Ваши ответы еще не истекло, Вы ответили на все вопросы, то вычисляется оценка за тест.

Результат: Оценка за тест = (Оценка за 1 этап + Баллы за 2 этап/Число вопросов 2 этапа) / 2

R19: Если Ваша оценка больше порога, то вы сдали тест.

Результат: Результат теста = Вы успешно сдали.

Конец консультации: 20.01.2020 10:08:37

Всего активировано правил: 13

Вопросы и задания

Разработка экспертной системы

Разработать продукционную экспертную систему для адаптивной поддержки принятия решений в следующей предметной области:

1. Назначение лекарственных препаратов.
2. Подбор программы тренировок.
3. Подбор схемы питания.
4. Диагностика заболеваний.
5. Диагностика поломок автомобиля.
6. Диагностика поломок компьютера.
7. Диагностика сети электроснабжения.
8. Формирование портфеля акций.
9. Управление светофорной сигнализацией.
10. Оценка текущего состояния процесса функционирования

Реализовать подход на основе четкой или нечеткой логики. Реализовать прямую или обратную цепочку рассуждения в качестве механизма логического вывода.

Тест для самоконтроля

1. Дерево принятия решения отражает всю логику принятия решения?

Варианты ответа:

- A. Да;
- B. Нет;

- C. Не всю;
- D. Да, но только в заданной предметной области.

2. Дерево описывает структуру самого процесса?

Варианты ответа:

- A. Без этапов, сразу;
- B. В два этапа, с контролем времени;
- C. В два этапа, без контроля времени;
- D. Дерево не описывает структуру процесса.

3. Какая из переменных задается экспертом?

Варианты ответа:

- A. time1;
- B. otvet;
- C. gotov2;
- D. result.

4. Какая переменная отвечает за результат сдачи теста?

Варианты ответа:

- A. oshibki;
- B. otvet;
- C. time2;
- D. result.

5. Последовательное применение правил по порядку, с возвратом после просмотра всех правил снова в начало, до тех пор, пока не будет получено значение целевой переменной – это...

Варианты ответа:

- A. Цепочка рассуждений с ветвлением;
- B. Прямая цепочка рассуждений;
- C. Цикл;
- D. Ни одно и перечисленных.

6. Первый этап алгоритма прямой цепочки рассуждений заключается в...

Варианты ответа:

- A. Обновлении таблицы;
- B. Просмотре всех правил;
- C. Заполнении таблицы переменными;
- D. Занесении переменной в очередь.

7. Подход, при котором в отдельные классы выделяются множества объектов, обладающих общими атрибутами, операциями, отношениями и семантикой, называется...

Варианты ответа:

- A. Процедурным;
- B. Структурным;
- C. Объектно-ориентированным;
- D. Аспектно-ориентированным.

8. Класс RuleSet представляет собой контейнер?

Варианты ответа:

- A. Набора правил;
- B. Атрибутов;
- C. Целевых функций;
- D. Стандартных типов.

9. Выполнение набора правил осуществляется командой...

Варианты ответа:

- A. Run;
- B. Execute;
- C. LoadFromDisk;
- D. CheckGoalVariable.

10. Обновление таблиц, если правило выполнено, осуществляется командой...

Варианты ответа:

- A. Execute;
- B. Update;
- C. SetOutputs;
- D. LoadFromDisk.

11. Язык C# создан специально для программирования под платформу...

Варианты ответа:

- A. Java Runtime Environment;
- B. Nero;
- C. Abbyu;
- D. .NET.

12. В Microsoft Visual Studio в режиме эксперта можно...

Варианты ответа:

- A. Вести протокол;
- B. Проверять успеваемость и просматривать протоколы;
- C. Задавать вопросы;
- D. Отображать результаты теста.

13. Процесс логического вывода продолжается до тех пор, пока...

Варианты ответа:

- A. Целевой переменной не присваивается значение;
- B. Пользователь не нажмет на кнопку;
- C. Закончится цикл;
- D. Не выключится компьютер.

14. Существует много вариаций алгоритмов прямой цепочки рассуждения, однако их суть состоит в следующем: в результате последовательного тестирования часть правил инициализируется, что ведет к...

Варианты ответа:

- A. Перезагрузке системы;
 - B. Удалению системы;
 - C. Изменению состояния системы;
 - D. Переименованию системы.
15. Исходными данными для реализации алгоритма являются...
- Варианты ответа:
- A. Переменные;
 - B. Команды пользователя;
 - C. Файл с именем «исходные данные»;
 - D. Правила.
16. Если конъюнкция в условии правила истинна, то...
- Варианты ответа:
- A. Выполняется левая часть правила;
 - B. Выполняется правая часть правила;
 - C. Ничего не выполняется;
 - D. Переходим к следующему шагу.
17. Если при активизации правила достигается значение целевой переменной, то...
- Варианты ответа:
- A. Она обнуляется, и программа продолжает работу;
 - B. Переменные удаляются из очереди;
 - C. Процесс завершается;
 - D. Запускается новый процесс.
18. Если какие-либо переменные приняли значения, то...
- Варианты ответа:
- A. Они удаляются из очереди;
 - B. Процесс приостанавливается;
 - C. Они суммируются;
 - D. Появляется сообщение.
19. В конце логического вывода очередь должна быть...
- Варианты ответа:
- A. Пуста;
 - B. Заполнена нулями;
 - C. Заполнена значениями;
 - D. Переопределена.
20. Удаление неиспользуемых объектов из памяти называется...
- Варианты ответа:
- A. Обновлением;
 - B. Форматированием;
 - C. Сборкой мусора;
 - D. Чисткой.

Ответы на вопросы теста

1. D
2. B
3. A
4. D
5. B
6. C
7. C
8. A
9. B
10. C
11. D
12. B
13. A
14. C
15. D
16. B
17. C
18. A
19. A
20. C

Глава 5. Адаптивные решения на основе прецедентов

5.1 Реализация модели CBR

Модель поддержки принятия решений на основе технологии CBR поиска описания прецедентов (паттернов) решения аналогичных задач в базах знаний обеспечивает интерпретацию и представление данных, знаний, информации, понимания в процессе разработки управленческих решений. Также данная модель позволяет выявить, хранить и анализировать прецеденты в исследуемой предметной области. Прецедент представляет собой паттерн, характеризующий определенные свойства задачи поиска решения.

Задача принятия решений по управлению сложной системой является многокритериальной. В общем виде в системе поддержки принятия решения данная задача T (Task) представляется следующим совокупностью характеристик:

$$T = \langle S, A, C, M, P, R \rangle,$$

где S (*Situation*) – текущая ситуация принятия решений;

A (*Alternatives*) – множество альтернативных прецедентов, на основе которых осуществляется выбор;

C (*Criteria*) – множество критериев для оценки предлагаемых прецедентов;

M (*Model*) – модель расчёта вектор критериев для каждого из предлагаемых прецедентов;

P (*Preferences*) – предпочтения для каждого из критериев;

R (*Rule*) – правило, предназначенное для выбора окончательного варианта.

В применяемом подходе прецедент представляет собой паттерн, который является структурой, отражающей некоторые свойства. Данные свойства характерны для задач поиска решения. Паттерны имеют слотовую структуру и классифицируются по приложениям, степени обобщения описываемых решений и группам задач. Паттерны находятся между собой в отношениях обобщения, ассоциации и зависимости.

Паттерн P (*Pattern*) в системе поддержки принятия решений описывается тетрадой

$$P = \langle N, G, S, R \rangle,$$

где N (*Name*) – уникальное название паттерна;

G (*Goal*) – условия и задача применения паттерна;

S (*Solution*) – решение поступившей задачи;

R (*Result*) – результаты применения данного паттерна.

В системе поддержки принятия решений может быть использована совокупность рассуждений по прецедентам и качественных рассуждений, основанных на онтологической модели представления знаний. На рисунке 29 представлен процесс принятия решений на основе интеграции различных типов рассуждений, в котором выделены следующие элементы:

- структура паттерна (1);
- формализованные знания о предметной области (2);
- сохраненные паттерны (3);
- новый паттерн (4);
- результаты качественного моделирования (5).

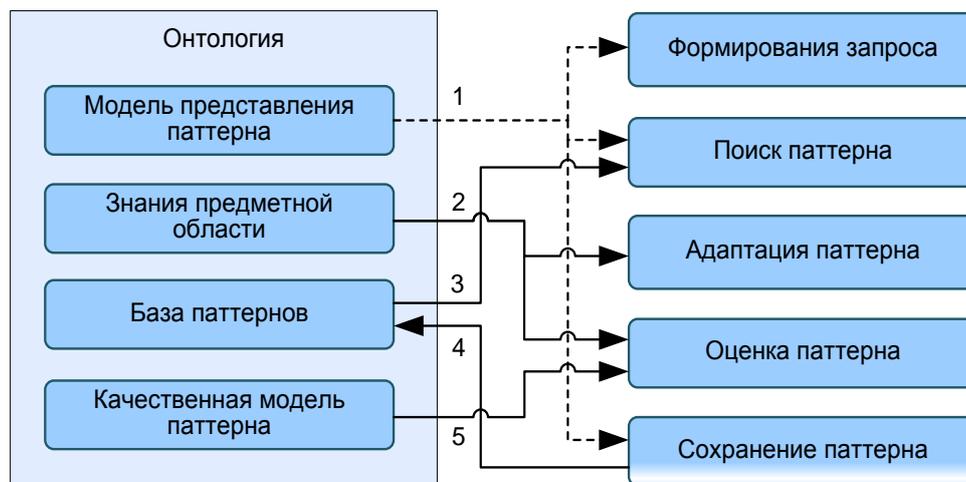


Рисунок 29 – Интеграция рассуждений по прецедентам

Метод рассуждений по паттернам использует как общие знания предметной области, аналогичные ситуации и решения, так и специфические данные и выводы из конкретных ситуаций (паттернов). Решение по задаче принимается при помощи нахождения аналогичной ситуации в прошлом и использования принятого решения.

После загрузки паттернов выполняется выбор паттерна, наиболее соответствующего для текущей задачи поиска решения. Производится сравнение признаков в текущей ситуации и в выбранных паттернах. Вводится метрика (расстояние) на пространстве всех признаков, в котором определяется точка, соответствующая текущему случаю. В рамках данной метрики находится ближайшая к ней точка из точек, которые представляют паттерны. Каждому признаку назначается вес, определяющий его относительную ценность.

Полностью степень близости D паттерна по всем признакам вычисляется как:

$$D = \frac{\sum_j w_j * sim(x_{ij}, x_{kj})}{\sum_j w_j},$$

где w_j – вес j -го признака;

sim – функция подобия (метрика);

x_{ij} и x_{ik} – значения признака x_j для текущего случая и существующего паттерна, соответственно.

Таким образом, CBR-подход обеспечивает возможность использования опыта, накопленного системой, без привлечения эксперта. Подход позволяет повысить точность поиска решения текущей проблемы и сократить время данного поиска. Прецедентный подход позволяет исключить повторное получение ошибочного решения за счет использования решений для аналогичных задач.

5.2 Реализация алгоритма CBR-цикла

При реализации CBR-подхода, цикл которого представлен на рисунке 30, используются схожие ситуации и решения, а также специфические данные и выводы из реальных ситуаций. После нахождения аналогичной ситуации в прошлом принимается решение по исследуемой задаче на основе уже принятого решения. Поэтому применение данной технологии позволяет без привлечения эксперта использовать опыт, который ранее был накоплен системой.

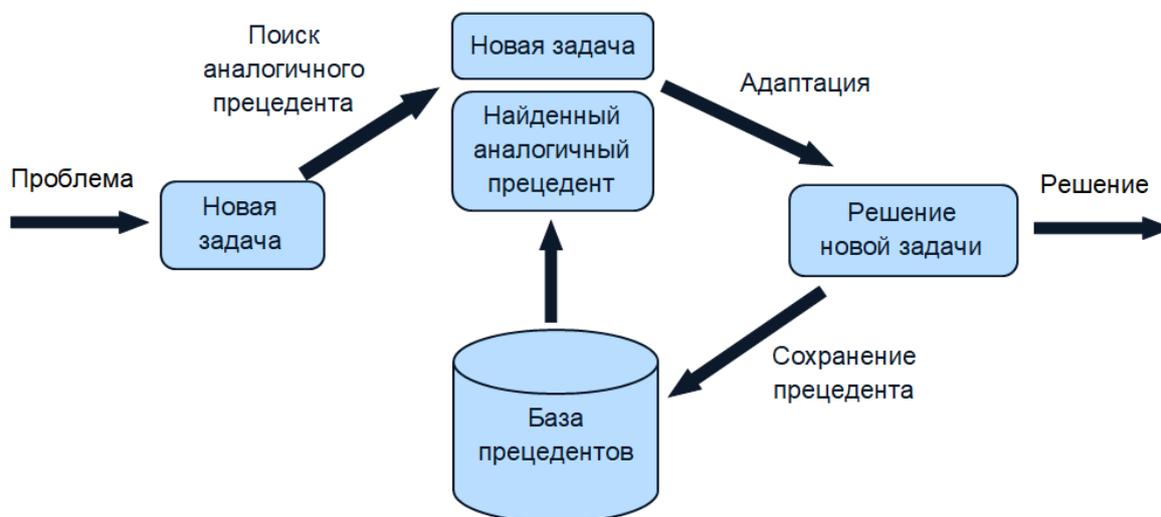


Рисунок 30 – Схема реализации CBR-цикла

CBR-цикл (прецедентный подход) состоит из следующих этапов:

1. Восстановление (извлечение). Данная стадия начинается с того, что поступившая проблемы представляется в виде нового прецедента пользователю. В процессе извлечения один или несколько прецедентов

выбираются из базы прецедентов. Для решения новой проблемы выбираются только прецеденты, полезные для нее. Прецеденты отбираются, основываясь на подобии (близости) новой проблемы к проблемам, описание которых содержат прецеденты из базы прецедентов. Предполагается, что близкие проблемы имеют подобные решения.

На качество работы автоматизированной СВР-системы довольно сильно влияет точная формализация подобия. Оценка подобия является знанием, которое сильно зависит от предметной области. При использовании сложных структур для представления прецедентов вычисление оценки подобия может оказаться очень сложным, в том числе и в вычислительном отношении.

2. Повторное использование. Для решения новой проблемы используются данные, которые содержатся в извлеченных прецедентах. Возможность повторного использования в ряде случаев достигается путем адаптации (преобразования) восстановленных решений. В настоящий момент существует несколько методов адаптации для СВР систем. Основным отличием различных методов адаптации заключается в применении одного из следующих подходов [1]:

- трансформационная адаптация основывается на совокупности операторов или правил адаптации, описывающих, каким образом на изменения в решении влияют отличия в описании извлеченной проблемы. Происходит анализ отличий между извлеченным и новым прецедентом. Далее происходит отбор применимых преобразований и выполнение предложенных изменения решения. Основы данного подхода были описаны в работах по трансформационной аналогии;
- методы порождающей адаптации для своего эффективного применения требуют наличия решателя, способного на основании обобщенных знаний решить проблему без использования прецедентов. При этом используются не решения, но последовательность шагов (порядок) решения проблемы, описанные в прецедентах. Эта последовательность используется для управления (направления) решателем.

Стадия повторного использования может быть в вычислительном отношении очень затратной, особенно в случае использования сложных структур для представления прецедентов, в которых решение прецедента требует детального описания. Затраты на адаптацию зависят от степени подобия (близости) между восстановленным (извлеченным) и новым прецедентами.

3. Переработка (пересмотр, проверка). Данная стадия очень слабо отражена в литературе, в большинстве случаев это связано с тем, что задача пересмотра (проверки) решения обычно не является задачей СВР системы. Решение, определенное СВР системой, проверяется и по

возможности исправляется человеком - экспертом предметной области.

4. Сохранение. На стадии сохранения происходит обучение CBR системы, суть которого заключается в добавлении пересмотренного прецедента в базу прецедентов. Таким образом, опыт решения новой проблемы становится доступным для повторного использования в будущем. Однако данный подход имеет очевидные недостатки, наиболее важным из которых является непрерывный рост базы прецедентов, что в дальнейшем приводит к понижению эффективности извлечения. Во избежание возникновения данной проблемы были предложены стратегии выборочного добавления прецедентов в базу прецедентов и выборочного «забывания» уже сохраненных прецедентов.

5.3 Извлечение прецедентов

Одним из главных этапов CBR-цикла является извлечение прецедента из базы прецедентов. Для успешной реализации рассуждений на основе прецедентов необходимо обеспечить корректное извлечение прецедентов из базы правил. Выбор метода извлечения прецедентов напрямую связан со способом представления прецедентов.

Для извлечения прецедентов из базы правил системы могут применяться различные методы, наиболее популярными из которых являются следующие [1]:

- метод ближайшего соседа и его модификации (например метод k ближайших соседей);
- метод поиска на деревьях решений;
- метод извлечения на основе знаний;
- метод извлечения с учетом применимости прецедентов и др.

Извлечение прецедентов методом поиска на деревьях решений

Этот метод предполагает нахождение требуемых прецедентов путем разрешения вершин дерева решений. Каждая вершина дерева указывает, по какой ее ветви следует осуществлять дальнейший поиск решения. Выбор ветви осуществляется на основе информации о текущей проблемной ситуации.

Метод деревьев решений реализует принцип так называемого «рекурсивного деления». Эта стратегия также называется «Разделяй и властвуй». В узлах, начиная с корневого, выбирается признак, значение которого используется для разбиения всех данных на 2 класса. Процесс продолжается до тех пор, пока не выполнится критерий остановки. Это возможно в следующих ситуациях:

- все (или почти все) данные данного узла принадлежат одному и тому же классу;
- не осталось признаков, на основе которых можно построить новое разбиение;

- дерево превысило заранее заданный «лимит роста» (если таковой был заранее установлен).

Следовательно, требуется добраться до концевой вершины, соответствующей одному или нескольким прецедентам. В случае, когда концевая вершина является связанной с некоторым подмножеством прецедентов, можно использовать метод ближайшего соседа для выбора наиболее подходящего из извлеченных прецедентов. Обычно рекомендуется использовать данный подход для больших баз прецедентов. В связи с тем, что основная часть работы по извлечению прецедентов выполняется заранее на этапе построения дерева решений, этот подход значительно сокращает время, требуемое для поиска решения.

Метод извлечения прецедентов на основе знаний

Метод извлечения прецедентов на основе знаний, в отличие от предыдущих методов, позволяет учесть знания экспертов по конкретной предметной области (коэффициенты важности параметров, выявленные зависимости и т.д.) при извлечении прецедентов. Метод реализует подход, основанный на индексации прецедентов специальным образом (семантической индексации). При определении прецедентов учитываются важности параметров прецедентов, заданные экспертом или лицом, принимающим решения, и другая информация, позволяющая учесть знания о конкретной предметной области. За счет этого значительно сокращается время поиска решения, что является существенным достоинством данного метода [1].

Процесс выполнения индексации усложняется с ростом числа прецедентов в базе правил и необходимостью проводить индексацию динамически. Для реализации метода требуется предусмотреть в структуре прецедентов и базе правил возможности представления и хранения семантической информации, а также дополнительные затраты на поддержание базы правил для учета знаний о конкретной предметной области. Метод может успешно применяться совместно с другими методами извлечения прецедентов, особенно когда база правил имеет большие размеры и предметная область является открытой и динамической.

Извлечение прецедентов методом ближайшего соседа

Метод ближайшего соседа (соседей) – наиболее популярный и часто используемый метод для извлечения прецедентов, который также широко применяется для решения задач классификации, регрессии, распознавания образов и т.д. В основе метода лежит определенный способ измерения степени схожести (близости) прецедента и текущей проблемной ситуации. Говоря формально, необходимо ввести метрику на пространстве параметров (признаков, свойств) для описания прецедентов и текущей

ситуации, а затем, определяя на основе выбранной метрики расстояние между точками, соответствующими прецедентам, и точкой, соответствующей текущей ситуации, выбрать точку (прецедент), ближайшую к текущей ситуации.

Особенности метода ближайшего соседа:

- простота использования полученных результатов;
- решения не уникальны для конкретной ситуации, возможно их использование для других случаев;
- целью поиска является не гарантированно верное решение, а лучшее из возможных;
- позволяет провести более глубокий анализ данных, чем дерево решений;
- данный метод не создает каких-либо моделей или правил, обобщающих предыдущий опыт.

В большинстве случаев для представления прецедентов достаточно параметрического представления, т.е. представления прецедента в виде набора параметров с конкретными значениями и решения:

$$CASE_S(x_1, x_2, \dots, x_n, R),$$

где $x_1 \dots x_n$ – параметры ситуации, описывающей данный прецедент ($x_1 \in X_1, x_2 \in X_2, \dots, x_n \in X_n$),

R – диагноз и рекомендации,

n – количество параметров прецедента,

X_1, \dots, X_n – области допустимых значений соответствующих параметров прецедента, являющиеся подмножествами простых типов данных (чисел, строк).

Данное представление является простым параметрическим представлением прецедента и является геометрической моделью n -мерного пространства, где каждый прецедент является точкой этого пространства, а каждый параметр рассматривается как одна из координат в пространстве.

Поиск решения на основе прецедентов заключается в определении степени сходства текущей ситуации с прецедентами из базы правил. Степень сходства зависит от близости текущей ситуации и прецедента и определяется с помощью алгоритма определения ближайшего соседа. В алгоритме определения ближайшего соседа используется простое по координатное сопоставление текущей ситуации с прецедентом. Различные модификации метода ближайшего соседа широко применяются для решения задач классификации, кластеризации, регрессии и распознавания образов. Обычно решение выбирается на основе нескольких ближайших точек (соседей), а не одной (метод ближайших соседей).

5.4 Анализ алгоритмов оценивания исследуемых объектов

Алгоритм должен обладать следующими свойствами:

- однозначность, заключающаяся в совпадении получаемых результатов при применении алгоритма к одним и тем данным;
- дискретность, означающая последовательное выполнение простых или определённых ранее шагов, на которые разбит алгоритм;
- массовость, состоящая в возможности применения алгоритма к ряду однотипных задач;
- результативность, предполагающая получение результата после выполнения конечного числа шагов алгоритма.

Одним из главных этапов СВР-цикла является извлечение прецедента из базы прецедентов. Для успешной реализации рассуждений на основе прецедентов необходимо обеспечить корректное извлечение прецедентов из базы правил. Выбор метода извлечения прецедентов напрямую связан со способом представления прецедентов.

При росте базы прецедентов обычно наблюдается уменьшение эффективности извлечения, так как приходится анализировать все большее количество прецедентов. Для более эффективного извлечения предлагается применять индексирование прецедентов.

Эффективность метода во многом зависит от выбора метрики. Основные типы метрик, которые могут быть использованы в задачах поиска близких прецедентов:

1. Евклидова метрика (евклидово расстояние):

$$d_{CT} = \sqrt{\sum_{i=1}^n (x_i^C - x_i^T)^2};$$

2. Квадрат евклидова расстояния:

$$d_{CT} = \sum_{i=1}^n (x_i^C - x_i^T)^2;$$

3. Манхэттенская метрика:

$$d_{CT} = \sum_{i=1}^n |x_i^C - x_i^T|;$$

4. Расстояние Чебышева:

$$d_{CT} = \max_i |x_i^C - x_i^T|;$$

5. Мера близости Журавлева:

$$d_{CT} = \sum_{i=1}^n I_i^{CT},$$

$$\text{где } I_i^{CT} = \begin{cases} 1, & \text{если } |x_i^C - x_i^T| < \varepsilon \\ 0, & \text{в противном случае} \end{cases};$$

6. Мера сходства по Хэммингу:

$$S(C, T) = \frac{n_{CT}}{n},$$

где C – прецедент;

T – текущая проблемная ситуация в n -мерном пространстве.

5.5 Автоматизация тестирования решений

Автоматизированное тестирование предназначено для проверки работы функций программного обеспечения, таких как авторизация и аутентификация пользователей, запуск, выполнение, разбор и получение результатов, производимых в автоматическом режиме.

Внедрение автоматизированного тестирования увеличивает количество найденных ошибок в программном обеспечении и повышает скорость проведения. Для того, чтобы автоматизировать процесс тестирования, необходима документация о программном обеспечении, в которой описываются функциональные возможности и их применение. Перед введением автоматизированного тестирования нужно оценить целесообразность такого подхода и определить, какие части программного обеспечения необходимо подвергать проверке. После определения целесообразности автоматизированного тестирования составляется план действий, в котором описаны этапы создания тестов и выбраны способы их реализации.

В адаптивных системах поддержки принятия решений целесообразно вводить автоматизированное тестирование. Такой подход к процессу тестирования был выбран из-за ряда условий, принадлежащих системе:

- наличие однообразных действий, таких как проверка заполнения полей формы, создания большого количества пользователей системы;

- использование одинаковых функций несколько раз, с высоким риском проявления ошибок;
- большой объем записываемой информации в базу данных;
- валидация сообщений;
- наличие функций поиска для большого количества данных.

Среди автоматического тестирования можно выделить следующие подходы:

- модульное тестирование (unit test) – проверка отдельных компонентов системы;
- GUI тесты – тестирование с помощью имитации работы пользователя в системе, при помощи графического пользовательского интерфейса, т. е. проверка работы кнопок, заполнения полей форм, валидация.

Модульные тесты

В рамках наблюдения за работой системы создаются искусственные ситуации, которые называются тестами. Тесты создаются на основе требований к программному обеспечению. Созданные тесты образуют группы и применяются на программном обеспечении. В итоге имеется набор результатов, в которых указывается, насколько правильно работает та или иная часть системы (рисунок 31). На основе результатов тестирования можно определить ошибки в программном обеспечении.

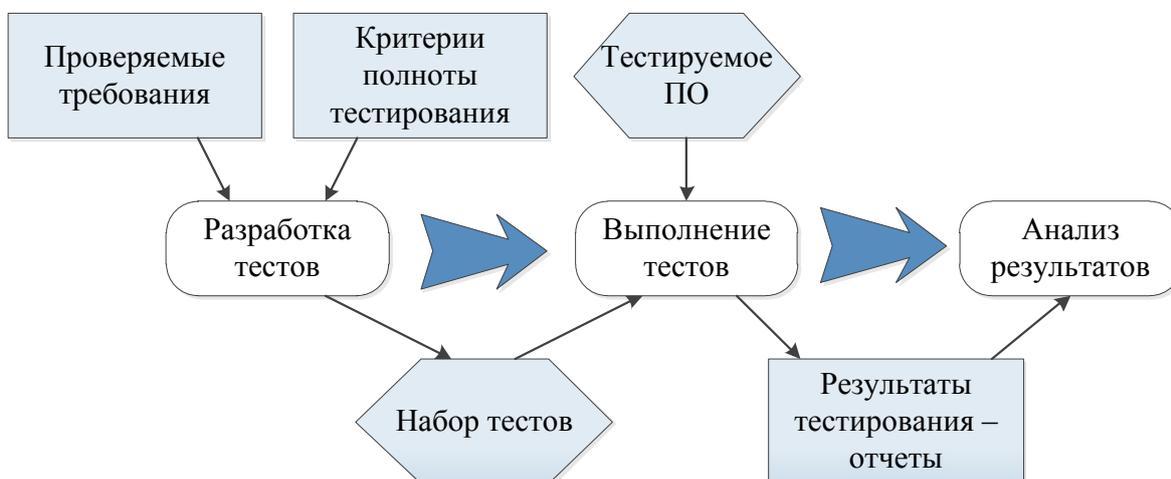


Рисунок 31 – Процесс тестирования

Поиск ошибок в системе посредством GUI тестов требует моделирования работы графического интерфейса, такое поведение программного обеспечения можно проанализировать и посредством unit тестов.

Перед разработкой модульных тестов необходимо разработать набор тест-кейсов, позволяющих понять порядок действий для выполнения задач системы.

Тест-кейсом называется наименьший элемент тестирования, который включает описание работы, характеристик и условий, предназначенных для проверки функциональности отдельных частей системы.

Тест-кейс включает следующие параметры:

- уникальный идентификатор тест-кейса;
- наименование;
- предусловия – набор шагов, выполняющихся до начала теста;
- шаги – список действий, после выполнения которого выдается соответствующий результат;
- соответствующий результат – результат, ожидаемый при выполнении тест-кейса;
- статус – сравнение соответствующего результата с фактическим; возможно три вида статуса: положительный, отрицательный и невыполненный;
- история изменений – изменения, вносимые в тест-кейс для достижения фактического результата тестируемого компонента системы.

Тест-кейс должен удовлетворять следующим требованиям:

- требования к программному обеспечению необходимо проверить и затем разбить на отдельные части;
- тест-кейс должен обеспечивать высокую вероятность обнаружения ошибки;
- формулировка шагов в тест-кейсе должна быть четко определена;
- все шаги теста должны быть связаны между собой;
- недопустимы связи между отдельными тест-кейсами;
- соответствующий результат записывается в тест-кейс до его заполнения;
- тест-кейсы должны охватывать не только правильное выполнение функций системы, но моделировать возможные ошибки пользователей при работе с программой;
- тест-кейс должен проверять на возможность наличия лишних функциональных возможностей системы, не прописанных в документации.

Интеграционное тестирование

После тестирования отдельных компонентов необходимо протестировать их взаимодействие. Для этого используем интеграционный подход к тестированию программного обеспечения.

Интеграционное тестирование – это тестирование, с помощью которого по заданным характеристикам проверяется интеграция модулей, связь между модулями и интеграция отдельных подсистем в одну общую систему (рисунок 32).

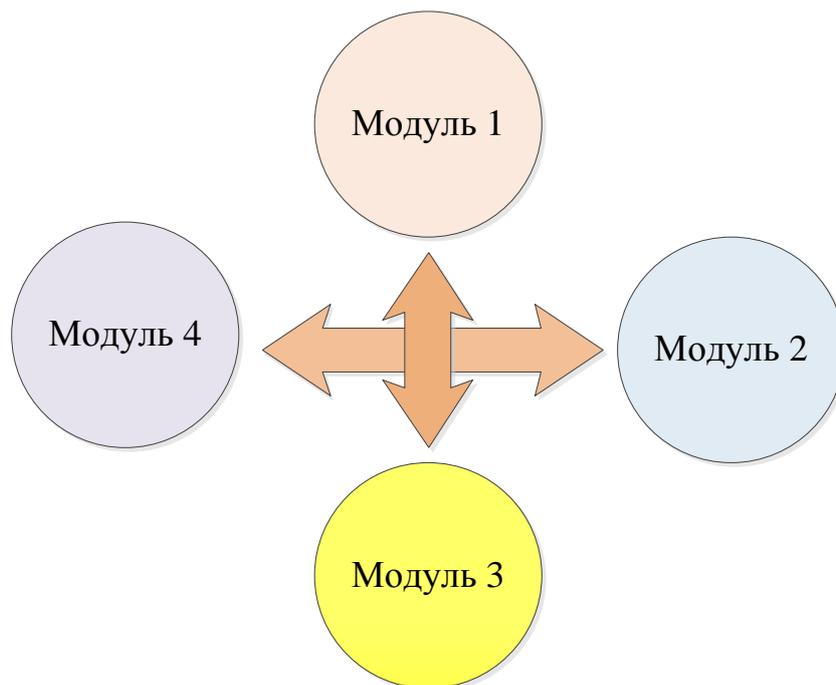


Рисунок 32 – Объединение модулей при интеграционном тестировании

Существует три способа интеграционного тестирования.

- Снизу вверх (Bottom Up Integration). Все низкоуровневые модули объединяются в один, и происходит проверка их взаимодействия. Затем собирается следующий уровень модулей. Такой способ требует, чтобы все модули уровня были готовы одновременно, также необходима большая затрата времени.
- Сверху вниз (Top Down Integration). Объединение модулей происходит сверху вниз. Такой подход также требует наличия полноценного модульного уровня.
- Большой взрыв («Big Bang» Integration). Все разработанные модули объединяются в законченную систему, и затем осуществляется интеграционное тестирование. Такой способ не требует наличия всех модулей в системе, тем самым можно увеличить количество тестовых проходов. Но такой подход требует хорошей базы модульного тестирования.

Вопросы и задания

Тест для самоконтроля

1. Какие из перечисленных характеристик относятся к совокупности характеристик задачи принятия решений?

Варианты ответа:

- A. Alternatives;
- B. Model;
- C. Solution;

- D. Result;
 - E. Preferences;
2. Паттерн в системе поддержки принятия решений описывается тетрадой...
- Варианты ответа:
- A. Goal, Name, Solution, Result;
 - B. Goal, Criteria, Model, Solution;
 - C. Rule, Model, Name, Goal;
 - D. Preferences, Goal, Name, Solution.
3. В каких отношениях между собой находятся паттерны?
- Варианты ответа:
- A. Обобщение, ассоциация, зависимость;
 - B. Обобщение, композиция, агрегация;
 - C. Агрегация, реализация, зависимость;
 - D. Реализация, бинарная ассоциация, обобщение.
4. CBR-подход обеспечивает возможность использования опыта, накопленного системой?
- Варианты ответа:
- A. Только с привлечением эксперта;
 - B. Без привлечения эксперта;
 - C. Как с экспертом, так и без него.
5. Выберите правильную последовательность этапов CBR-цикла...
- Варианты ответа:
- A. Сохранение, повторное использование, переработка, восстановление;
 - B. Восстановление, повторное использование, переработка, сохранение;
 - C. Восстановление, переработка, сохранение, повторное использование;
 - D. Сохранение, переработка, повторное использование, восстановление.
6. Как описывается характеристика задачи принятия решения Criteria?
- Варианты ответа:
- A. Множество критериев для оценки предлагаемых прецедентов;
 - B. Множество критериев для оценки векторов расчёта критериев предлагаемых прецедентов;
 - C. Множество критериев для оценки альтернативных прецедентов;
 - D. Данная характеристика не входит в совокупность характеристик задачи принятия решения.

7. Возможность повторного использования решений в ряде случаев достигается путём...

Варианты ответа:

- A. Агрегации восстановленных решений;
- B. Адаптации восстановленных решений;
- C. Композиции восстановленных решений;
- D. Обобщения восстановленных решений.

8. Какие недостатки имеет СВР-подход?

Варианты ответа:

- A. Невозможность повторного использования опыта;
- B. Перетирание предыдущих прецедентов пересмотренным;
- C. Рост базы прецедентов;
- D. Для решения новой проблемы могут использоваться неактуальные для неё прецеденты.

9. Стадия переработки (пересмотра) решения обычно является задачей...

Варианты ответа:

- A. Самой СВР системы;
- B. Сторонних систем;
- C. Эксперта предметной области;
- D. Либо самой системы, либо эксперта.

10. Какой из этапов СВР-цикла является наиболее вычислительно затратным при использовании сложных структур прецедентов?

Варианты ответа:

- A. Восстановление;
- B. Повторное использование;
- C. Переработка;
- D. Сохранение.

11. Процесс выполнения индексации с ростом числа прецедентов в базе правил...

Варианты ответа:

- A. Усложняется;
- B. Упрощается;
- C. Не изменяется;
- D. Может как усложняться, так и упрощаться.

12. Какой из методов извлечения прецедентов широко применяется для решения задач классификации, регрессии и распознавания образов?

Варианты ответа:

- A. Метод ближайшего соседа;
- B. Метод поиска на деревьях решений;
- C. Метод извлечения на основе знаний;
- D. Метод извлечения с учетом применимости прецедентов.

13. Какой из методов извлечения прецедентов рекомендуется использовать для больших баз прецедентов?

Варианты ответа:

- A. Метод ближайшего соседа;
- B. Метод поиска на деревьях решений;
- C. Метод извлечения на основе знаний;
- D. Метод извлечения с учетом применимости прецедентов.

14. Какой из методов извлечения прецедентов учитывает коэффициенты важности параметров и выявленные зависимости при извлечении прецедентов?

Варианты ответа:

- A. Метод ближайшего соседа;
- B. Метод поиска на деревьях решений;
- C. Метод извлечения на основе знаний;
- D. Метод извлечения с учетом применимости прецедентов.

15. Какие варианты из перечисленных являются критерием остановки рекурсивного деления в методе деревьев решений?

Варианты ответа:

- A. Все (или почти все) данные данного узла принадлежат одному и тому же классу;
- B. Дерево превысило заранее заданный «лимит роста»;
- C. Не осталось признаков, на основе которых можно построить новое разбиение;
- D. Значение признака разбивает данные на 3 класса.

16. Что из нижеперечисленного относится к особенностям метода ближайшего соседа?

Варианты ответа:

- A. Решения уникальны для конкретной ситуации;
- B. Решений не уникальны для конкретной ситуации;
- C. Метод не создает каких-либо моделей или правил, обобщающих предыдущий опыт;
- D. Метод позволяет создавать модели или правила, обобщающие предыдущий опыт.

17. Какие из перечисленных свойств должны быть присущи алгоритму определения ближайшего соседа?

Варианты ответа:

- A. Однозначность;
- B. Массовость;
- C. Уникальность;
- D. Цикличность;
- E. Результативность.

18. Какой подход помогает увеличить эффективность извлечения прецедентов?

Варианты ответа:

- A. Улучшение аппаратных характеристик базы правил;
- B. Индексирование прецедентов;
- C. Выбор оптимального метода извлечения;
- D. Подробное описание имеющихся прецедентов.

19. Какие из перечисленных метрик наиболее часто используются в задачах поиска близких прецедентов?

Варианты ответа:

- A. Мера близости Журавлева;
- B. Расстояние Левенштейна;
- C. Расстояние Чебышева;
- D. Метрика Васерштейна.

20. Какие из перечисленных метрик не используются в задачах поиска близких прецедентов?

Варианты ответа:

- A. Евклидова метрика;
- B. Мера близости Журавлева;
- C. Расстояние Левенштейна;
- D. Мера Хэмминга.

Ответы на вопросы теста

1. A, B, E
2. A
3. A
4. B
5. B
6. A
7. B
8. C
9. C
10. B
11. A
12. A
13. B
14. C
15. A, B, C
16. B, C
17. A, B
18. B
19. A, C
20. C

Заключение

В пособии рассмотрены теоретические вопросы и практические аспекты разработки адаптивных систем поддержки принятия решений и экспертных систем, основанных на знаниях. Описана специфика разработки подобных систем, рассмотрены технологии, закладываемые в их основу: модели представления знаний, методы искусственного интеллекта, алгоритмы логического вывода. Раскрыты особенности проектирования систем, использующих прецеденты (паттерны) для поддержки принятия решений. Описаны средства получения первичных данных, необходимых для функционирования систем поддержки принятия решений. По каждой главе приведены вопросы и задания, направленные на приобретение знаний и формирование навыков по рассмотренной проблематике.

В настоящем пособии не рассматриваются общие вопросы теории принятия решений и методологические основы построения экспертных систем.

Для дальнейшего изучения тем, связанных с технологиями поддержки принятия решений, экспертными системами, искусственным интеллектом, приведен список рекомендуемой литературы.

Список рекомендуемой литературы

1. Андрейчиков, А.В. Интеллектуальные информационные системы [Текст] : учебник / А.В. Андрейчиков, О.Н. Андрейчикова. – М. : Финансы и статистика, 2004. – 422 с.
2. Бураков, П.В. Экспертные системы принятия решения [Текст] : учебное пособие / П.В. Бураков, С.С. Гусева. – СПб. : СПбГУ ИТМО, 2001. – 104 с.
3. Гаскаров, Д.В. Интеллектуальные информационные системы [Текст] : учебник / Д.В. Гаскаров. – М. : Высш. шк., 2003. – 430 с.
4. Джарратано, Дж. Экспертные системы: принципы разработки и программирования [Текст] / Дж. Джарратано, Г. Райли. – М. : Вильямс, 2007. – 1152 с.
5. Катулев, А.Н. Математические методы в системах поддержки принятия решений [Текст] / А.Н. Катулев, Н.А. Северцев. – М. : Высшая школа, 2005. – 312 с.
6. Кравченко, Т.К. Системы поддержки принятия решений [Электронный ресурс] : учебник и практикум / Т.К. Кравченко, Д.В. Исаев. – М. : Издательство Юрайт, 2018. – 292 с. – URL : <http://www.biblionline.ru/book/B2FF1983-705C-49F2-BE27-1362F66D576E>.
7. Левин, Р. Практическое введение в технологию искусственного интеллекта и экспертных систем с иллюстрациями на бейсике [Текст] / Р. Левин, Д. Дранг, Б. Эделсон. – М. : Финансы и статистика, 1990. – 239 с.
8. Люгер, Дж.Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем [Текст] / Дж.Ф. Люгер. – М. : Вильямс, 2005. – 864 с.
9. Муромцев, Д.И. Введение в технологию экспертных систем [Текст] : учебное пособие / Д.И. Муромцев. – СПб. : СПбГУ ИТМО, 2005. – 110 с.
10. Паттерны проектирования сложноорганизованных систем [Текст] / Т.И. Михеева, С.В. Михеев, О.К. Головнин, О.Н. Сапрыкин. – Самара : Интелтранс, 2015. – 216 с.
11. Рассел, С. Искусственный интеллект. Современный подход [Текст] / С. Рассел, П. Норвиг. – М. : Вильямс, 2018. – 1408 с.
12. Ручкин, В.Н. Универсальный искусственный интеллект и экспертные системы [Текст] : учебное пособие / В.Н. Ручкин, В.А. Фулин. – СПб. : БХВ-Петербург, 2009. – 238 с.
13. Статические и динамические экспертные системы [Текст] : учебное пособие / Э.В. Попов, И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. – Москва : Финансы и статистика, 1996. – 318 с.
14. Таунсенд, К. Проектирование и программная реализация экспертных систем на персональных ЭВМ [Текст] / К. Таунсенд, Д. Фохт. – М. : Финансы и статистика, 1990. – 320 с.
15. Халин, В.Г. Системы поддержки принятия решений [Электронный ресурс] : учебник и практикум / В.Г. Халин. – М. : Издательство Юрайт,

2018. – 494 с. – URL : www.biblio-online.ru/book/C65198DA-46BA-4EC4-BOED-FFEEACE35A61.

16. Частиков, А.П. Разработка экспертных систем. Среда CLIPS [Текст] : учебное пособие / А.П. Частиков, Т.А. Гаврилова, Д.Л. Белов. – СПб. : БХВ-Петербург, 2003. – 606 с.

Список используемой литературы

1. Башлыков, А.А. Применение методов теории прецедентов в системах поддержки принятия решений при управлении трубопроводными системами [Текст] / А.А. Башлыков // Автоматизация, телемеханизация и связь в нефтяной промышленности. – 2016. – № 1. – С. 23–32.
2. Головнин, О.К. Информационные технологии, вычислительная техника и компьютерные сети в транспортных системах [Текст]: учеб. пособие / О.К. Головнин, А.А. Столбова, С.Э. Хоружников. – Самара: НПЦ ИТС, 2019. – 88 с.
3. Головнин, О.К. Прецедентная система поддержки принятия решений по делам об административных правонарушениях [Текст] / О.К. Головнин, Е.А. Романова // Программные продукты и системы. – 2018. – № 1. – Т. 31. – С. 44–50.
4. Дерябкин, В.П. Информационные системы в технике и технологиях [Текст]: учеб. пособие / В.П. Дерябкин, А.С. Овсянников, В.П. Павлов. – Ч. 3. – Самара: Самар. гос. архитектур.-строит. ун-т, 2004. – 180 с.
5. Дошина, А.Д. Экспертная система. Классификация. Обзор существующих экспертных систем [Текст] / А.Д. Дошина // Молодой ученый. – 2016. – № 21(125). – С. 756-758.
6. Кочкин Г.А. Проблемы рассуждений по прецедентам, детализации, интеграции оценки схожести прецедентов [Электронный ресурс] / Г.А. Кочкин, В.Р. Кочкина, И.А. Голубкин // Инженерный вестник Дона. – 2013. – № 4. – URL: <http://www.ivdon.ru/ru/magazine/archive/n4y2013/2215>.
7. Муромцев, Д.И. Введение в технологию экспертных систем [Текст]: учеб. пособие / Д.И. Муромцев. – СПб.: СПбГУ ИТМО, 2005. – 110 с.
8. Норенков, И.П. Поддержка принятия решений на основе паттернов проектирования [Электронный ресурс] / И.П. Норенков, М.Ю. Уваров // Наука и образование. – 2011. – № 9. – URL: <http://technomag.edu.ru/doc/228646.html>.
9. Рассел, С. Искусственный интеллект. Современный подход [Текст] / С. Рассел, П. Норвиг. – М.: Вильямс, 2018. – 1408 с.
10. Таунсенд, К. Проектирование и программная реализация экспертных систем на персональных ЭВМ [Текст] / К. Таунсенд, Д. Фохт. – М.: Финансы и статистика, 1990. – 320 с.
11. Трофимов, И.В. Планирование на базе рассуждений по прецедентам [Электронный ресурс]. – URL: <http://ai-center.botik.ru/planning/index.php?ptl=materials/071cbp.htm>.
12. Aamodt, A. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches [Текст] / А. Аамодт // Artificial Intelligence Communications. – IOS Press, 1994. – P. 55–70.
13. Hammond, K.J. Case-based Planning: A Framework for planning from experience [Текст] / К.К. Хаммонд // Cognitive Science. – Vol. 14 (3). – 1990. – P. 385–443.

**Головнин Олег Константинович,
Супрун Антон Сергеевич**

**ТЕХНОЛОГИИ АДАПТИВНОГО ПЛАНИРОВАНИЯ
В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ
РЕШЕНИЙ**

Учебно-методическое пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49