

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

INFORMATICS AND PROGRAMMING

Научная статья

УДК 004.891.2, 004.827

doi: 10.17223/19988605/70/11

Модель знаний оболочки экспертных систем EESS

Алексей Михайлович Бабанов

*Национальный исследовательский Томский государственный университет, Томск, Россия
babanov@mail.tsu.ru*

Аннотация. Предлагается оригинальный подход к построению инструментов для тестирования знаний с использованием продукционных экспертных систем. Наличие механизма вывода и аппарата для работы с неточно представленной информацией обеспечивает недетерминированный характер работы вопросно-ответной системы. Возможность применения подхода в различных областях обеспечивает оболочка экспертных систем. Задачи тестирования знаний требуют развития традиционных представлений о продукционных экспертных системах. В статье формулируются требования к таким системам и определяется модель знаний расширенной оболочки EESS (Extended Expert System Shell). В основном идеи разработки родились в применении к проведению тестирования знаний студентов, но предлагаемые возможности будут полезны и в других областях, например в медицинской диагностике – традиционной предметной области экспертных систем.

Ключевые слова: оболочка экспертных систем; продукционная модель знаний; неопределенность информации; тестирование знаний.

Для цитирования: Бабанов А.М. Модель знаний оболочки экспертных систем EESS // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2025. № 70. С. 113–121. doi: 10.17223/19988605/70/11

Original article

doi: 10.17223/19988605/70/11

Knowledge model of the EESS expert system shell

Alexey M. Babanov

Tomsk State University, Tomsk, Russian Federation, babanov@mail.tsu.ru

Abstract. The article proposes an original approach to building tools for knowledge testing using production expert systems. The presence of an inference mechanism and an apparatus for working with inaccurately presented information ensures the non-deterministic nature of the question-answer system. The ability to apply the approach in various fields is provided by the expert system shell. Knowledge testing tasks require the extension of traditional ideas about production expert systems. The article formulates requirements for such systems and defines a knowledge model of EESS (Extended Expert System Shell). The development ideas were mainly born in application to testing students' knowledge, but the proposed capabilities will also be useful in other areas, for example, in medical diagnostics – a traditional subject area of expert systems.

Keywords: expert system shell; production knowledge model; information uncertainty; knowledge testing.

For citation: Babanov, A.V. (2024) Knowledge model of the EESS expert system shell. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 70. pp. 113–121. doi: 10.17223/19988605/70/11

Введение

Экспертные системы (ЭС), основанные на продукциях [1–3], позволяют представить знания экспертов в виде множества правил «если..., то...». С их помощью даже неподготовленный человек способен проанализировать типичную ситуацию и сделать профессиональный вывод. В ходе консультации ЭС определяет очередной вопрос, задает его человеку, принимает ответ, агрегирует его с предыдущими ответами и в ходе логического вывода определяет следующую текущую цель, которая приведет к очередному вопросу. На основании полученных ответов ЭС придет к определенному решению, выраженному в виде некоторого значения целевой переменной. Причем ЭС не бездумно задает всегда одни и те же вопросы в фиксированном порядке, а определяет разумную траекторию консультации в зависимости от полученных ею ответов.

К положительным характеристикам ЭС относятся:

- мощные механизмы прямого и обратного логического вывода;
- многочисленные настройки логического вывода, позволяющие тонко определить поведение системы;
- возможность использования аппарата для работы с нечеткими значениями и правилами, позволяющего оперировать неточными, размытыми знаниями.

Наряду с ЭС, раз и навсегда жестко фиксирующими набор правил в исполняемом коде, существуют так называемые оболочки экспертных систем (ОЭС) [4]. Они предлагают проектировщику ЭС некоторую модель представления знаний и язык, на котором опытный эксперт должен представить свой опыт анализа ситуаций предметной области. После определения ЭС пользователи могут проводить консультации. Таким образом, эти универсальные программные продукты позволяют использовать их в самых различных предметных областях.

В ранних ЭС пользователь, запустив консультацию, отвечал на вопросы в течение одного сеанса и в конце получал на экране результат – значение целевой переменной. Ответы и результат не сохранялись. В простейших случаях этой функциональности было достаточно.

Кроме однопользовательского сценария экспертизы с простыми ответами продукционную модель представления знаний и механизмы логического вывода можно использовать и в более сложных ситуациях, например при тестировании знаний. Для этого в первую очередь и создается расширенная оболочка экспертных систем EESS.

Целью данной разработки является обеспечение применения идей и моделей представления знаний ЭС для создания интеллектуальных систем тестирования знаний студентов. Использование экспертных систем в этом случае требует развития форм ответов пользователей, распределения консультации по нескольким сеансам и между несколькими пользователями, хранения данных экспертизы между сеансами и многого другого. О том, как при этом меняется модель представления знаний, и идет речь в статье.

1. Задачи расширенной оболочки экспертных систем

Опишем использование системы для тестирования знаний. Одна ЭС предназначена для тестирования по одному предмету (в примерах будет использоваться курс «Базы данных»). В течение периода обучения последовательно изучаются и тестируются разделы курса. Перед началом освоения курса преподаватель создает ЭС, включающую множество тестовых заданий различной сложности по разделам курса и правила определения окончательной оценки за тестирование. В начале курса он запускает глобальные консультации для каждого студента, и далее система по мере необходимости сама инициирует частные консультации со студентами по разделам курса. После подготовки к тестированию очередного раздела курса студент отвечает на тестовые задания консультации, а система оценивает его ответы и последовательно агрегирует эти оценки в итоговую оценку. Таким образом, глобальная консультация продолжается весь период обучения, а студенты выполняют множество мелких консультаций (тестирований) по окончании каждого раздела курса.

С момента появления ЭС и ОЭС предпринимались многочисленные попытки их совершенствования [5–7]. В каждом случае преследовались свои определенные цели. В нашем проекте развитие модели знаний оболочки ЭС идет по нескольким направлениям.

Построение базы знаний комплексной консультации – очень непростая интеллектуальная задача, часто неподвластная одному человеку. Таким образом, ЭС может представлять собой совокупность нескольких взаимосвязанных частей (экспертных подсистем), подготовленных различными узкими специалистами, но в целом решающих одну задачу. Это напоминает ситуацию, когда врач-терапевт направляет пациента на консультации к специалистам и впоследствии учитывает их мнения при постановке окончательного диагноза. Значит, система должна предусматривать соавторство экспертов и обеспечивать в ходе консультации обращение к другим консультациям. Этого же требует предлагаемая процедура тестирования. Экспертизы по разным разделам часто определяют различные педагоги.

В сложных случаях, когда одному человеку затруднительно отвечать на вопросы ЭС, требующие специальных знаний, можно устраивать что-то, напоминающее консилиум. При этом система при необходимости посылает вопрос нужному специалисту и продолжает консультацию, получив его ответ. Таким образом, консультация становится распределенной и многопользовательской. В нашем примере с тестированием начинает глобальную консультацию один пользователь (преподаватель), и далее подключаются другие пользователи (студенты).

Одно из применений расширенной оболочки – создание тестирующих систем, в которых традиционно используются самые разнообразные варианты тестовых заданий. Напрашивается расширение спектра возможных видов ответов. Помимо ввода одного четкого значения, неплохо было бы предусмотреть:

- ввод нечеткого значения (в случае сомнений пользователей);
- выбор из предложенных альтернатив (простой или множественный);
- ввод нескольких значений;
- упорядоченный ввод нескольких значений;
- упорядочивание предложенных альтернатив.

И, наконец, хорошо было бы иметь возможность откладывать консультацию, сохранять полученные ответы, промежуточные и окончательные результаты в долговременном хранилище для последующего возобновления консультации, анализа результатов и, возможно, изменения ответов и хода вывода. Это абсолютно необходимо для задачи тестирования знаний по всему курсу.

2. Общение с пользователем

Вся информация, касающаяся одной конкретной консультации (или экспертизы), представляется в виде значений переменных. Назовем их переменными экспертизы. Они могут быть четкими (иметь одно значение) или нечеткими. Каждое из значений нечеткой переменной имеет свой фактор уверенности – ФУ (число от 0 до 100, отражающее уверенность в том, что переменная имеет данное значение). Наряду с русскоязычной аббревиатурой ФУ часто используется англоязычная CF (Certainty Factor).

Кроме переменных экспертизы в модели знаний имеются системные переменные, которые предназначены для настройки ЭС на определенную стратегию поведения. Эти переменные будем называть переменными среды (их имена имеют префикс «Е.»). Некоторые из них, связанные с выводом и нечеткой информацией, будут представлены в дальнейшем. Остальные останутся за рамками статьи.

Если значения переменных среды устанавливаются разработчиком ЭС, хранятся в ней и преимущественно не изменяются в ходе консультации, то значения переменных экспертизы определяются либо ответами пользователя, либо механизмом вывода автоматически с использованием правил. Причем способ означивания система выбирает на основании значений переменных среды.

Проектируя работу с ответами пользователя, разработчики анализировали два варианта значений переменных экспертизы.

1-й вариант. Значение переменной – сам введенный объект (строка, номер альтернативы, множество, последовательность). Тогда надо расширять набор типов переменных, включая в него множества и последовательности. В этом случае сравнение с эталоном ответа должно осуществляться в посылке правил, что приведет к серьезному расширению грамматики и необходимости реализации сравнений множеств и последовательностей. К тому же этот вариант неудобен при массовом вводе тестов (а это неизбежно) и потребует большого количества правил, что необоснованно перегрузит ЭС.

2-й вариант. Реализовать специальные функции ввода ответов (строка, номер альтернативы, множество, последовательность) и в них осуществлять проверку с эталоном, представленным в параметрах функций, а возвращать агрегированное значение оценки выполнения задания (в истинностных значениях или в баллах). Тогда в посылках правил будет проще проверять любые ответы единообразно. При этом можно определять задания заранее и хранить их вместе с эталонами ответов в таблицах БД, а в определении переменной ссылаться на элементы заданий одного типа. Его можно выбирать и случайно.

Второй вариант проще и перспективней.

Покажем, например, как может выглядеть его реализация для ввода одиночного значения (функция GETVAL) простого (функция GETIND) и множественного (функция GETSET) выбора из предложенных альтернатив, а также выбора и упорядочивания (функция GETSEQ) предложенных альтернатив.

Заранее (в момент создания ЭС) для каждой функции вводятся таблицы с заданиями (табл. 1–4), что позволяет не указывать параметры тестового задания при обращении к функции.

Таблица 1

Таблица базы данных ЭС с заданиями, использующими функцию GETVAL

Переменная	Параметр QWE	Параметр ANSW
TEST1	Как называется комплекс, все элементы которого лежат в одной позиции?	Множество
TEST1	Другой вопрос	Другой ответ

Определение переменной TEST1 может выглядеть так. В первом варианте задание указывается в параметрах функции, во втором – ссылкой на первую строку соответствующей таблицы БД с заданным именем переменной. Для остальных примеров используется тот же принцип.

VAR: TEST1

GETVAL QWE: «Как называется комплекс, все элементы которого лежат в одной позиции?»

ANSW: «Множество».

Или обращаемся к ранее введенной таблице GETVAL (1).

Переменная принимает четкое значение TRUE или FALSE в зависимости от того, совпал ответ или нет.

Таблица 2

Таблица базы данных ЭС с заданиями, использующими функцию GETIND

Переменная	Параметр QWE	Параметр ALT	Параметр ANSW
TEST2	Что не является множеством?	1) отношение, 2) домен, 3) кортеж, 4) атрибут	3
TEST2	Другой вопрос		Другой ответ

VAR: TEST2

GETIND QWE: «Что не является множеством?» ALT: 1) отношение, 2) домен, 3) кортеж, 4) атрибут. ANSW: 3.

Или обращаемся к ранее введенной таблице GETIND (1).

Переменная принимает четкое значение TRUE или FALSE в зависимости от того, совпал ответ или нет.

Таблица 3

Таблица базы данных ЭС с заданиями, использующими функцию GETSET

Переменная	Параметр QWE	Параметр ALT	Параметр ANSW
TEST3	Укажите действия с данными, которые есть в любой модели данных	1) удаление, 2) установка текущей, 3) изменение, 4) включение, 5) выборка, 6) селекция, 7) поиск	{1, 3, 4, 5}
TEST3	Другой вопрос		Другой ответ

VAR: TEST3

GETSET QWE: «Укажите действия с данными, которые есть в любой модели данных» ALT: 1) удаление, 2) установка текущей, 3) изменение, 4) включение, 5) выборка, 6) селекция, 7) поиск. ANSW: {1, 3, 4, 5}.

Или обращаемся к ранее введенной таблице GETSET (1).

В этом случае для вычисления значения переменной возможны различные варианты манипуляции с количеством совпавших (С), количеством недостающих (Н) и количеством лишних (Л) элементов. Например, С, С-Н-Л и т.д.

Переменная принимает нечеткое значение TRUE с соответствующим ФУ. Если результат (С или С-Н-Л) совпал с мощностью эталона, значение = TRUE с ФУ 100. Если результат ≤ 0 , значение = TRUE с ФУ 0. Таким образом, ФУ для TRUE равен $\text{MAX}((\text{С или С-Н-Л})/|\text{ЭТАЛОН}|, 0) \times 100$.

Например, если ответ будет {1, 3, 4, 6, 7}, С = 3, Н = 1, Л = 2, С-Н-Л = 0. Значение = TRUE с ФУ 0. Если ответ будет {1, 2, 4, 6, 7}, С = 2, Н = 2, Л = 3, С-Н-Л = -3. Значение = TRUE с ФУ 0. Если ответ будет {1, 2, 3, 4, 5}, С = 4, Н = 0, Л = 1, С-Н-Л = 3. Значение = TRUE с ФУ 75.

Таблица 4

Таблица базы данных ЭС с заданиями, использующими функцию GETSEQ

Переменная	Параметр QWE	Параметр ALT	Параметр ANSW
TEST4	Укажите формы данных, соответствующие элементам восприятия мира человеком: связь, характеристика, тип связей, тип объектов, объект	1) отношение, 2) домен, 3) атрибут, 4) комплекс, 5) кортеж	<5, 3, 1, 1, 5>
TEST4	Другой вопрос		Другой ответ

VAR: TEST4

GETSEQ QWE: «Укажите формы данных, соответствующие элементам восприятия мира человеком: связь, характеристика, тип связей, тип объектов, объект» ALT: 1) отношение, 2) домен, 3) атрибут, 4) комплекс, 5) кортеж. ANSW: <5, 3, 1, 1, 5>.

Или обращаемся к ранее введенной таблице GETSEQ (1).

В этом случае для вычисления значения переменной возможны различные варианты манипуляции с количеством совпавших (С) и количеством стоящих на своих позициях (П) элементов. Например, С + П и т.д.

Переменная принимает нечеткое значение TRUE с соответствующим ФУ. Если результат (С + П) совпал с удвоенной мощностью эталона, значение = TRUE с ФУ 100. Если результат = 0, значение = TRUE с ФУ 0. Таким образом, ФУ для TRUE равен $(\text{С} + \text{П}) / (2 \times |\text{ЭТАЛОН}|) \times 100$.

Например, если ответ будет <1, 3, 4, 5, 5>, С = 4, П = 2, С + П = 6. Значение = TRUE с ФУ 60. Если ответ будет <1, 2, 4, 6, 7>, С = 1, П = 0, С + П = 1. Значение = TRUE с ФУ 10. Если ответ будет <5, 4, 3, 2, 1>, С = 3, П = 1, С + П = 4. Значение = TRUE с ФУ 40.

3. Правила вывода

Другой способ означивания переменных экспертизы применяется механизмом вывода. Каждая ЭС имеет целевую переменную экспертизы, и задачей системы является получение значения этой переменной. Это значение не вводится пользователем (иначе как бы система проявляла свой интеллект), а выводится на основании правила (или нескольких правил). Поэтому набор правил вывода составляет ядро знаний системы [8].

Основные элементы любого правила – посылка (IF) и заключение (THEN). Посылка представляет собой некоторое условие; если оно верно, то выполняются действия, указанные в заключении (в основном присвоения значений переменных экспертизы).

Условие посылки – это логическое выражение, состоящее из операций сравнения значений переменных экспертизы с точными константами и логических операций (NOT, AND, OR). Заключение составляет одна или несколько операций присвоения переменной экспертизы четких или нечетких значений. Операция присвоения “=” замещает значение, если оно существовало. К переменной также можно добавлять новые значения (с сохранением имеющихся). Для этого используется операция “+=”. Можно также удалять значения. Для этого применяется операция “-=”. Каждое добавление того же значения увеличивает его ФУ, каждое удаление значения уменьшает.

IF TEST3 >= 80

THEN MARK = {4 CF 90, 5 CF 100}

Здесь каждое значение нечеткой переменной имеет фактор уверенности (он указывается после ключевого слова CF).

В качестве источника значения в операции присвоения может быть указана консультация, значение целевой переменной которой и будет использовано в этом случае; например, THEN MARK = CONSULT(THEORY, “Иванов”). Здесь THEORY – имя экспертной системы, а “Иванов” – имя пользователя, который будет отвечать на вопросы во время консультации.

4. Работа с неточной информацией

Одним из обязательных требований к ЭС является наличие механизмов представления и обработки нечетких и недостоверных знаний [9–10]. Основными способами представления таких знаний являются:

- факторы уверенности, отражающие степень уверенности в некотором элементе знаний – значении переменной, посылке правила;
- нечеткие переменные – переменные, принимающие одновременно несколько значений, каждое со своим фактором уверенности.

Операции по обработке факторов уверенности, выполняемые в ходе сеанса работы с ЭС, сводятся к следующим:

- определение фактора уверенности посылки на основании значений переменных, используемых в ней;
- определение фактора уверенности значений переменных заключения после применения правила;
- определение обобщенного фактора уверенности значения переменной, полученного применением нескольких правил.

Способ вычисления фактора уверенности посылки заключается в поэтапном применении формул преобразования ФУ каждого из двух операндов бинарной логической операции в их общий ФУ. Для операции AND алгоритм преобразования ФУ задается переменной среды E.CFA (Certainty Factor of AND), а для операции OR – переменной E.CFO (Certainty Factor of OR). При использовании логической связки AND общий фактор уверенности не превышает минимального из факторов уверенности операндов (так как для истинности операции необходима истинность обоих условий). При использовании логической связки OR фактор уверенности посылки не может быть меньше, чем максимальный из факторов уверенности отдельных операндов (так как для истинности операции достаточно, чтобы истинным было хотя бы одно из условий).

Понятно, что для такого алгоритма подсчета ФУ посылки ее логическое выражение должно быть представлено в соответствующем виде, например в КНФ или в ДНФ. Этого можно потребовать от эксперта при создании ЭС, а можно строить их автоматически и сохранять в БД уже готовыми для вычислений ФУ.

Возможные значения этих переменных среды и соответствующие им формулы для вычисления фактора уверенности операции приведены в табл. 5, где CF₁, CF₂ – факторы уверенности операндов бинарной операции, CF_{оп} – фактор уверенности операции в целом).

Возможные значения переменных среды и формулы для вычисления фактора уверенности бинарной логической операции

Значения переменных	E.CFA	E.CFO
М	$CF_{оп} = \min(CF_1, CF_2)$	$CF_{оп} = \max(CF_1, CF_2)$
Р	$CF_{оп} = CF_1 \times CF_2 / 100$	$CF_{оп} = CF_1 + CF_2 - CF_1 \times CF_2 / 100$

Формулы, применяемые для вычисления факторов уверенности в экспертных системах, не имеют строгого математического обоснования. Формулы, соответствующие значениям “М”, близки к интуитивным правилам оценки уверенности, применяемым в рассуждениях человека и теории нечетких множеств. Формулы, соответствующие значениям “Р”, применяются в теории вероятностей.

Способ вычисления фактора уверенности значений переменных заключения, получаемых в результате применения правила, задается значением переменной среды E.CFSV (Certainty Factor of Single Value). Способ вычисления обобщенного фактора уверенности для совпадающих значений одной и той же переменной, полученных в нескольких правилах, задается значением переменной E.CFMV (Certainty Factor of Multiple Values).

Возможные значения этих переменных среды и соответствующие им формулы для вычисления факторов уверенности приведены в табл. 6. В таблице использованы следующие обозначения: $CF_{пос}$ – фактор уверенности посылки; $CF_{ук}$ – фактор уверенности значения, указанный в заключении; $CF_{закл}$ – фактор уверенности значения, полученный в результате применения правила; $CF_{закл1}$, $CF_{закл2}$ – факторы уверенности одного и того же значения некоторой переменной, полученного в заключениях разных правил; $CF_{об}$ – обобщенный фактор уверенности значения переменной, полученного в результате применения обоих правил.

Таблица 6

Возможные значения переменных среды и формулы для вычисления фактора уверенности значений переменных заключения

Значения	E.CFSV	E.CFMV
М	$CF_{закл} = \min(CF_{пос}, CF_{ук})$	$CF_{об} = \max(CF_{закл1}, CF_{закл2})$
Р	$CF_{закл} = CF_{пос} \times CF_{ук} / 100$	$CF_{об} = CF_{закл1} + CF_{закл2} - CF_{закл1} \times CF_{закл2} / 100$

Обобщенный фактор уверенности повышается по сравнению с факторами уверенности, полученными в отдельных правилах, так как значение переменной подтверждается несколькими правилами.

Покажем, как происходит работа ЭС в условиях неполной информации.

Пусть $TEST3 = \{TRUE \text{ CF } 75\}$, $TEST4 = \{TRUE \text{ CF } 60\}$, $E.CFA = \text{“М”}$, $E.CFO = \text{“М”}$, $E.CFSV = \text{“Р”}$, $E.CFMV = \text{“Р”}$.

В примере используется системная функция $CFV(<переменная>, <значение>)$. Она возвращает фактор уверенности указанного значения переменной; если такого значения нет, то возвращается фактор уверенности, равный нулю.

Предположим, что механизм вывода последовательно применяет правила R1 и R2.

Сначала рассматривается правило R1.

IF $TEST3 = TRUE \text{ AND } CFV(TEST3, TRUE) \geq 70$

THEN $MARK += \{3 \text{ CF } 60, 4 \text{ CF } 90, 5 \text{ CF } 100\}$.

Фактор уверенности условия $TEST3 = TRUE$ равен 75. Условие $CFV(TEST3, TRUE) \geq 70$ равно TRUE с ФУ 100. Фактор уверенности посылки в целом находится как минимальный из факторов уверенности отдельных условий (в соответствии с переменной среды E.CFA, так как в правиле используется связка AND): $\min(75, 100) = 75$.

Далее находятся факторы уверенности значений переменной заключения. Вычисление выполняется в соответствии со значением переменной среды E.CFSV. Для значения 3 фактор уверенности находится как $75 \times 60 / 100 = 45$ (здесь 75 – фактор уверенности посылки, 60 – фактор уверенности значения 3, указанный в заключении правила). Аналогично находятся факторы уверенности для значений 4 ($75 \times 90 / 100 = 68$) и 5 ($75 \times 100 / 100 = 75$). Таким образом, переменная MARK после рассмот-

рения правила R1 принимает следующие значения: {3 CF 45, 4 CF 68, 5 CF 75}. ФУ здесь представляет собой уверенность в соответствующем значении оценки за определенный раздел курса.

Далее обрабатывается правило R2.

IF TEST4 = TRUE AND CFV(TEST4, TRUE) >= 60

THEN MARK += {3 CF 70, 4 CF 90, 5 CF 80}.

Фактор уверенности условия TEST4 = TRUE равен 60. Условие CFV(TEST4, TRUE) >= 60 равно TRUE с ФУ 100. Фактор уверенности посылки в целом находится как минимальный из факторов уверенности отдельных условий (в соответствии с переменной среды E.CFA, так как в правиле используется связка AND): $\min(60, 100) = 60$.

Далее находятся факторы уверенности значений переменной заключения. Вычисление выполняется в соответствии со значением переменной среды E.CFSV. Для значения 3 фактор уверенности находится как $60 \times 70/100 = 42$. Аналогично находятся факторы уверенности для значений 4 ($60 \times 90/100 = 54$) и 5 ($60 \times 80/100 = 48$). Таким образом, переменная MARK в заключении правила R2 принимает следующие значения: {3 CF 42, 4 CF 54, 5 CF 48}.

Поскольку в заключении используется операция "+=", эти значения добавляются к значениям переменной MARK, полученным ранее (в правиле R1). К совпадающим значениям 3, 4 и 5 применяется алгоритм обобщения в соответствии со значением переменной E.CFMV. Для значения 3 фактор уверенности находится как $45 + 42 - 45 \times 42/100 = 68$. Аналогично находятся факторы уверенности для значений 4 ($68 + 54 - 68 \times 54/100 = 85$) и 5 ($75 + 48 - 75 \times 48/100 = 87$).

Таким образом, в результате выполнения правил R1 и R2 переменная MARK принимает следующие значения {3 CF 68, 4 CF 85, 5 CF 87}, и ЭС может сделать вывод, что по результатам первых двух тестовых заданий студент заслуживает оценки «отлично». Действительно, значение 5 имеет наивысший фактор уверенности.

Заключение

В статье представлены основные нетрадиционные элементы модели расширенной оболочки экспертных систем EESS, позволяющие применить технологию ЭС для тестирования знаний. К ним относятся:

- развитые способы получения ответов пользователей;
- использование в заключениях правил значений целевых переменных других ЭС;
- разнообразные возможности для учета неполноты информации.

Эти способности модели продиктованы требованиями, предъявляемыми со стороны экспертных систем для тестирования знаний студентов. Однако оболочка имеет универсальный характер и с успехом может использоваться во многих предметных областях.

Список источников

1. Russell S., Norvig P. Artificial Intelligence: A Modern Approach (Pearson Series in Artificial Intelligence). 4th ed. Pearson, 2020. 1136 p.
2. Vigo R., Zeigler D.E., Wimsatt J. Uncharted Aspects of Human Intelligence in Knowledge Based “Intelligent” Systems // *Philosophies*. 2022. V. 7 (46). P. 1–19. doi: 10.3390/philosophies7030046
3. Krivoulya G.F., Shkil A.S., Kucherenko D.Y. Analysis of production rules in expert systems of diagnosis // *Automatic Control and Computer Sciences*. 2013. Is. 47. P. 331–341.
4. Knowledge Engineering Shells / N.G. Bourbakis (ed.). Singapore ; River Edge, NJ : World Scientific Publishing Company, 1993. 536 p. (Advanced Series On Artificial Intelligence; v. 2).
5. Николайчук О.А., Павлов А.И., Юрин А.Ю. Компонентный подход: модуль производственной экспертной системы // *Программные продукты и системы*. 2010. № 3. С. 41–44.
6. Грищенко М.А., Николайчук О.А., Павлов А.И., Юрин А.Ю. Инструментальное средство создания производственных экспертных систем на основе MDA // *Образовательные ресурсы и технологии*. 2016. № 2 (14). С. 144–151.
7. Еремеев А.П. Проектирование экспертных систем средствами инструментальной системы GURU. М. : МЭИ, 1996. 52 с.
8. Nalepa G.J. Rules as a Knowledge Representation Paradigm // Nalepa G.J. Modeling with Rules Using Semantic Knowledge Engineering. Springer, 2018. P. 3–25. doi: 10.1007/978-3-319-66655-6

9. Walley P. Measures of uncertainty in expert systems // *Artificial Intelligence*. 1996. V. 83. P. 1–58.
10. Dubey S., Pandey R.K., Gautam S.S. Dealing with Uncertainty in Expert Systems // *International Journal of Soft Computing and Engineering (IJSCE)*. 2014. V. 4 (3). P. 105–111.

References

1. Russell, S. & Norvig, P. (2020) *Artificial Intelligence: A Modern Approach (Pearson Series in Artificial Intelligence)*. 4th ed. Pearson.
2. Vigo, R., Zeigler, D.E. & Wimsatt, J. (2022) Uncharted Aspects of Human Intelligence in Knowledge Based “Intelligent” Systems. *Philosophies*. 7(46). pp. 1–19. DOI: 10.3390/philosophies7030046
3. Krivoulya, G.F., Shkil, A.S. & Kucherenko, D.Y. (2013) Analysis of production rules in expert systems of diagnosis. *Automatic Control and Computer Sciences*. 47. pp. 331–341.
4. Bourbakis, N.G. (ed.) (1993) *Knowledge Engineering Shells*. Singapore; River Edge, NJ: World Scientific Publishing Company.
5. Nikolaychuk, O.A., Pavlov, A.I. & Yurin, A.Yu. (2010) Komponentnyy podkhod: modul' produktsionnoy ekspertnoy sistemy [Component approach: Module of production expert system]. *Programmnye produkty i sistemy*. 3. pp. 41–44.
6. Grishchenko, M.A., Nikolaychuk, O.A., Pavlov, A.I. & Yurin, A.Yu. (2016) Instrumental'noe sredstvo sozdaniya produktsionnykh ekspertnykh sistem na osnove MDA [Tool for creating production expert systems based on MDA]. *Obrazovatel'nye resursy i tekhnologii*. 2(14). pp. 144–151.
7. Eremeev, A.P. (1996) *Proektirovanie ekspertnykh sistem sredstvami instrumental'noy sistemy GURU* [Design of expert systems by means of the GURU instrumental system]. Moscow: MEI.
8. Nalepa, G.J. (2018) *Modeling with Rules Using Semantic Knowledge Engineering*. Springer. pp. 3–25. DOI: 10.1007/978-3-319-66655-6
9. Walley, P. (1996) Measures of uncertainty in expert systems. *Artificial Intelligence*. 83. pp. 1–58.
10. Dubey, S., Pandey, R.K. & Gautam, S.S. (2014) Dealing with Uncertainty in Expert Systems. *International Journal of Soft Computing and Engineering (IJSCE)* 4(3). pp. 105–111.

Информация об авторе:

Бабанов Алексей Михайлович – доцент, кандидат технических наук, доцент кафедры программной инженерии Института прикладной математики и компьютерных наук Национального исследовательского Томского государственного университета (Томск, Россия). E-mail: babanov2000@mail.ru

Автор заявляет об отсутствии конфликта интересов.

Information about the author:

Babanov Alexey M. (Associate Professor, Candidate of Technical Sciences, Associate Professor of the Department of Software Engineering at the Institute of Applied Mathematics and Computer Science of the National Research Tomsk State University, Tomsk, Russian Federation). E-mail: babanov2000@mail.ru

The author declares no conflicts of interests.

Поступила в редакцию 26.08.2024; принята к публикации 03.03.2025

Received 26.08.2024; accepted for publication 03.03.2025