

ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ ОБРАЗОВАНИЕМ

Касторнова Василина Анатольевна,

*Федеральное государственное бюджетное научное учреждение
«Институт управления образованием Российской академии образования»,
ведущий научный сотрудник, кандидат педагогических наук, доцент,
kastornova_vasya@mail.ru*

Андреев Алексей Евгеньевич,

*Федеральное государственное бюджетное научное учреждение
«Институт управления образованием Российской академии образования»,
старший научный сотрудник, кандидат технических наук, danvas3@rambler.ru*

Касторнов Анатолий Федосеевич,

*Череповецкий государственный университет, г. Череповец,
профессор кафедры математики и информатики, кандидат педагогических
наук, профессор, a_kastornov@mail.ru*

Яламов Георгий Юрьевич,

*Федеральное государственное бюджетное научное учреждение
«Институт управления образованием Российской академии образования»,
ведущий научный сотрудник, кандидат физико-математических наук,
geo@portalsga.ru*

ПРИМЕНЕНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ ТЕХНОЛОГИИ МОДУЛЬНОГО ПРОЕКТИРОВАНИЯ ЭКСПЕРТНЫХ СИСТЕМ КОНТРОЛЯ ЗНАНИЙ В СФЕРЕ ОБРАЗОВАНИЯ³

Аннотация

В статье рассмотрены вопросы построения обучающей экспертной системы (ОЭС), призванной служить одним из педагогических инструментов систематизации и контроля знаний обучаемых. Предлагаемая авторами структура ОЭС основывается на модульном принципе построения и включает в себя модули, позволяющие создавать базу знаний (БЗ) в некоторой предметной области, наполнять ее содержанием, производить настройку (обучение) ОЭС, а затем использовать при проведении занятий.

Ключевые слова:

обучающая экспертная система; модуль; база знаний; обучение экспертной системы.

³ Статья подготовлена в рамках выполнения государственного задания по теме «Интеллектуализация информационных систем и технологических процессов в сфере образования».

Целью данной статьи является практическая реализация технологии модульного проектирования экспертной системы (ЭС). В статье описана Паскаль-программа, создающая базу знаний и включающая в себя модули, позволяющие компоновать различные уровни (узлы), содержащие перечень вопросов (переменных), которые ставит ЭС обучаемому, чтобы правильно идентифицировать тот или иной объект (исход) в некоторой предметной области.

Структура ЭС

Экспертные системы возникли как значительный практический результат в применении и развитии методов искусственного интеллекта (ИИ) – совокупности научных дисциплин, изучающих методы решения задач интеллектуального (творческого) характера с использованием ЭВМ. ЭС – это набор программ, выполняющих функции эксперта при решении задач из некоторой предметной области. ЭС систематизируют информацию, проводят анализ, дают консультации, ставят диагноз. Практическое применение ЭС в сфере образования способствует повышению эффективности работы и квалификации специалистов.

Главным достоинством ЭС является возможность накопления и модификации знаний, сохранение их для последующего использования. В отличие от человека к любой информации экспертные системы подходят с точки зрения формальной логики, что улучшает качество проводимой экспертизы. При решении задач, требующих обработки большого объема знаний, возможность осуществления некорректного вывода при таком подходе чрезвычайно мала.

Экспертная система состоит из базы знаний (части системы, в которой содержатся факты), подсистемы вывода (множества правил, по которым осуществляется решение задачи), подсистемы приобретения знаний, подсистемы объяснения, модуля обучаемого и диалогового процессора. Качество ЭС определяется качеством и размером базы знаний, для построения которой требуется провести опрос специалистов, являющихся экспертами в конкретной предметной области, а затем организовать и систематизировать полученные знания для последующего использования. При построении подсистем вывода используют методы решения задач ИИ. За добавление в БЗ знаний новых правил и модификацию имеющихся отвечает подсистема приобретения знаний, в задачу которой входит приведение правил к виду, позволяющему подсистеме вывода применять эти правила в процессе работы, а также проверять вводимые или модифицируемые правила на непротиворечивость с уже имеющимися правилами. Подсистема объяснения позволяет получить ответы на такие вопросы: «Как было получено такое решение?» (ответ на этот вопрос представляет собой трассировку всего процесса вывода с указанием использованных фрагментов БЗ); «Почему было принято такое решение?» (ответом на этот вопрос является ссылка на вывод, непосредственно предшествовавший полученному решению). Простейшим вариантом модуля обучаемого является векторная модель, которая каждому изучаемому понятию или умению ставит в соответствие некоторый элемент, принимающий значение

«знает/не знает», в результате, уровень знаний обучаемого (уровень его компетентности) в изучаемом курсе определяется набором значений элементов этого вектора. Диалоговый процессор представляет собой планировщик с интерфейсом на естественном языке, преобразующий описание задачи в программу решения на основе информации БЗ.

Принципы функционирования ЭС

Основными отличиями ЭС от других программных продуктов являются использование не только данных, но и знаний, а также специальный механизм вывода решений и новых знаний на основе уже имеющихся. Знания в ЭС представляются в форме, которая может быть легко обработана на ЭВМ. В ЭС известен алгоритм обработки знаний, а не алгоритм решения задачи. Поэтому, применение алгоритма обработки знаний может привести к получению такого результата при решении конкретной задачи, возможность которого не была изначально очевидна. Более того, алгоритм обработки знаний заранее неизвестен и строится в процессе решения задачи на основании эвристических правил. Решение задачи в ЭС сопровождается понятными пользователю объяснениями, качество получаемых решений обычно не хуже, а иногда и лучше предлагаемого специалистами в предметной области. В системах, основанных на знаниях, правила (или эвристики), по которым решаются задачи в конкретной предметной области, хранятся в БЗ. Задачи ставятся перед системой в виде совокупности фактов, описывающих некоторую ситуацию, и система с помощью базы знаний пытается вывести заключение из этих фактов.

База знаний – наиболее важный компонент экспертной системы, на котором основаны ее «интеллектуальные способности». В отличие от всех остальных компонентов ЭС, база знаний – «переменная» часть системы, которая может пополняться и модифицироваться при использовании ЭС. Обычно это происходит между консультациями, но в некоторых системах может осуществляться и в процессе консультации. Существуют несколько способов представления знаний в ЭС, однако общим для всех них является то, что знания представлены в символьной форме (элементарными компонентами представления знаний являются тексты, списки и другие символьные структуры). Таким образом, в ЭС реализуется принцип символьной природы рассуждений, который заключается в том, что процесс рассуждения представляется как последовательность символьных преобразований.

Наиболее распространенный способ представления знаний – в виде конкретных фактов и правил, по которым из имеющихся фактов могут быть выведены новые. Правила в БЗ служат для представления эвристических знаний, т.е. неформальных правил рассуждения, вырабатываемых экспертом на основе опыта его деятельности.

Системы, основанные на знаниях, могут входить составной частью в компьютерные системы обучения. Система получает информацию о деятельности некоторого объекта (например, студента) и анализирует ее. База знаний изменяется в соответствии с действиями объекта. Примером такого действия может служить компьютерная игра, сложность которой увеличивается по мере возрастания степени квалификации играющего.

В последние годы в учебном процессе все большее распространение находят компьютерные технологии проверки знаний, которые принимают, в основном, форму компьютерного тестирования. Данная технология имеет свои достоинства и недостатки, главным из которых является возможность «угадывания» правильного ответа при слабом владении материалом. Таким образом, тестовая форма проверки знаний далека от совершенства и не может качественно оценить знания, особенно при организации итогового контроля знаний по завершении изучения какой-либо крупной темы учебного курса. С помощью тестирования часто бывает трудно оценить знания обучаемых с точки зрения выявления понимания ими как качественных характеристик, так и взаимосвязей изучаемых в курсе базовых понятий.

Этапы создания модели ЭС

В качестве примера можно привести изучение темы «Программное обеспечение ЭВМ» (ПО ЭВМ) в курсе информатики. Известно, что здесь принято проводить разделение ПО на три группы: базовое ПО, системы программирования и прикладное ПО. Аналогичная ситуация имеет место и при классификации типов данных в языке программирования Паскаль. Здесь важно уметь определять классификацию типов по таким характеристикам как: простые, специальные и структурные типы, а в рамках этих классов отличать встроенные, пользовательские, статические и динамические типы данных. С помощью теста проверить эти знания довольно проблематично, так как по своей сути в вопросе теста нужно указать несколько вариантов ответов, один из которых правильный, а все остальные – либо заведомо неправильные, либо правдоподобные. Подбор этих вариантов не всегда прост, и что особенно важно, не всегда удачен со стороны составителя теста. Однако, эта проблема легко решается, если в качестве инструмента проверки и систематизации знаний использовать экспертную систему учебного назначения.

Экспертные системы создаются для решения разного рода задач. Основные типы этих задач можно сгруппировать в следующие категории: интерпретация, прогноз, диагностика, наблюдение, проектирование, отладка, обучение, управление. ЭС, используемые при обучении, подвергаются диагностике, «отладке» и «исправлению» (коррекции) действия обучаемого. Обучающие системы создают модель того, что обучаемый знает и как он эти знания применяет к решению задачи [2]. Дальнейшее совершенствование ЭС, на современном этапе развития технологий, приводит к более сложным системам с элементами искусственного интеллекта, называемым системами нечеткой логики.

Пример реализации ЭС

В качестве примера приведена экспертная система учебного назначения, состоящая из следующих модулей: инициализация системы, ввод примеров, тренировка, обучение, запоминание базы знаний на внешнем носителе (в виде набора массивов фактов и правил), удаление и добавление переменных и исходов, загрузка базы знаний из внешнего запоминающего устройства в оперативное запоминающее устройство (ОЗУ), а также основной модуль диалога с пользователем [1]. Первые модули заполняются экспертом (преподавателем), а с последними двумя работают обучаемые.

При инициализации ЭС эксперт определяет структуру базы знаний, где указывает число узлов, переменных и исходов. При этом, набор переменных для каждого узла задает характерные признаки (атрибуты), присущие объектам (исходам) этого узла. Например, в базе знаний «Программное обеспечение ЭВМ» среди переменных первого узла указываются такие признаки: создает файлы, управляет работой ЭВМ, создает тексты, решает математические задачи и пр., которые определяют указанную выше классификацию ПО ЭВМ. Во втором узле в число переменных входят исходы первого узла и дополнительные признаки, которые позволяют отличить MS DOS от Windows NT, Word от Access, Pascal от Assembler. На следующем этапе ЭС обучается на конкретных примерах до тех пор, пока не перестанет ошибаться. Полученная в результате обучения ЭС база знаний записывается в виде системы файлов на внешнем носителе. Готовая ЭС впоследствии может быть использована обучаемыми после ее загрузки с внешнего носителя. ЭС позволяет также после работы с ней производить ее модификацию путем удаления и/или добавления новых переменных и исходов с последующим ее «дообучением».

Рассмотрим более подробно работу ЭС на примере классификации программного обеспечения ЭВМ. На начальном этапе экспертной системе задается исходная информация о числе узлов (их 2), числе переменных и исходов для каждого узла, а также названия переменных и исходов (модуль «инициализация системы»):

УЗЕЛ 1

Переменная 1 – Создает папки

Переменная 2 – Создает директории

Переменная 3 – Загружается автоматически

Переменная 4 – Однозадачный режим

Переменная 5 – Многозадачный режим

Переменная 6 – Графический интерфейс

Переменная 7 – Текстовый интерфейс

Переменная 8 – Создает файлы

Переменная 9 – Создает исполняемые файлы

Переменная 10 – Создает программы

Переменная 11 – Нуждается в трансляции

Переменная 12 – Создает тексты

Переменная 13 – Создает БД

Переменная 14 – Создает ЭТ

Переменная 15 – Создает графику

Переменная 16 – Осуществляет символьные преобразования

Переменная 17 – Управляет работой ЭВМ

Переменная 18 – Создает БЗ

Переменная 19 – Создает анимации

Переменная 20 – Производит импорт-экспорт

Исход 1 – Базовое ПО

Исход 2 – Система программирования

Исход 3 – Прикладное ПО

УЗЕЛ 2

- Переменная 1 – Базовое ПО
- Переменная 2 – Система программирования
- Переменная 3 – Прикладное ПО
- Переменная 4 – Процедурный
- Переменная 5 – Операторный
- Переменная 6 – Низкий уровень
- Переменная 7 – Визуальный
- Переменная 8 – Поддерживает работу сети
- Переменная 9 – Осуществляет вычисления
- Переменная 10 – Строит диаграммы
- Переменная 11 – Строит БД
- Переменная 12 – Строит ЭТ
- Переменная 13 – Решает математические задачи
- Переменная 14 – Допускает надстройки
- Переменная 15 – Создает текст
- Исход 1 – MS DOS
- Исход 2 – WINDOWS 3.1, 3.11
- Исход 3 – WINDOWS 98 и выше
- Исход 4 – ASSEMBLER
- Исход 5 – BASIC
- Исход 6 – PASCAL
- Исход 7 – PROLOG
- Исход 8 – VISUA_BASIC
- Исход 9 – WORD
- Исход 10 – PAINT
- Исход 11 – EXCEL
- Исход 12 – ACCESS
- Исход 13 – MATHCAD
- Исход 14 – POWER_POINT

После ввода исходных данных идет задание примера (модуль «ввод примера»), в котором для каждого исхода обоих узлов указывается наличие его характерных признаков.

За вводом примера следует этап тренировки ЭС (модуль «тренировка»), в котором по соответствующему алгоритму [3] заполняется массив правил, позволяющих по набору нескольких значений переменных (не обязательно всех) определить соответствующий им исход. Однако, этап тренировки, как правило, формирует довольно «сырой» массив правил, который не всегда обеспечивает поиск адекватного переменным (атрибутам) исхода. Поэтому, рекомендуется после тренировки провести обучение ЭС (модуль «обучение»). На этом этапе ЭС, ориентируясь на сформированный ранее массив правил, ставит эксперту уточняющие вопросы относительно наличия тех или иных свойств выбранному исходу. Обучение заканчивается, как только ЭС перестает ошибаться. После этого результаты обучения

заносятся в массив правил (модуль «запоминание») и на внешнем носителе сохраняются все ранее созданные массивы (в общей сложности семь файлов). ЭС готова к работе и ее можно использовать в качестве эксперта.

Работа обучаемого (студента) начинается с загрузки ЭС с внешнего носителя в ОЗУ (модуль «загрузка БЗ») и последующего диалога с ней. При диалоге на экране монитора возникают вопросы, на которые необходимо ответить в форме ввода с клавиатуры значений 0 или 1 в зависимости от наличия или отсутствия указанных свойств у искомого объекта (исхода). Число задаваемых ЭС вопросов и порядок их следования целиком зависит от уровня ее обученности экспертом (преподавателем). Иногда ответ формируется за 2-3 шага, порой задается более десятка вопросов (все зависит от общего числа введенных в БЗ переменных и искомым исходам). В приведенном примере с ПО ЭВМ для идентификации исхода ACCESS экспертная система в 1 узле задает вопросы: создает тексты, создает файлы, создает исполняемые файлы и принимает решение что это – Прикладное ПО, а потом во 2 узле спрашивает: создает графику, проводит вычисления, создает БД.

Ниже следует протокол работы ЭС по определению исхода WINDOWS 98 и выше:

УЗЕЛ 1

- (7) – Текстовый интерфейс есть <0/1>? 0
 - (8) – Создает файлы есть <0/1>? 0
 - (10) – Создает программы есть <0/1>? 0
 - (15) – Создает графику есть <0/1>? 0
 - (20) – Производит импорт-экспорт есть <0/1>? 0
 - (1) – Создает паки есть <0/1> ?1
 - (2) – Создает директории есть <0/1> ?0
 - (3) – Загружается автоматически есть <0/1>? 1
 - (4) – Однозадачный режим есть <0/1>? 0
 - (5) – Многозадачный режим есть <0/1> ?1
 - (9) – Создает исполняемые файлы есть <0/1>? 0
 - (13) – Создает БД есть <0/1>? 0
- Предлагается Базовое ПО в качестве исхода.

УЗЕЛ 2

- (11) – Строит БД есть <0/1>? 0
 - (4) – Процедурный есть <0/1>? 0
 - (6) – Низкий уровень есть <0/1>? 0
 - (13) – Решает математические задачи есть <0/1>? 0
 - (5) – Операторный есть <0/1>? 0
 - (7) – Визуальный есть <0/1>? 1
 - (14) – Допускает надстройки <0/1>? 0
 - (8) – Поддерживает работу сети есть <0/1>? 1
 - (10) – Строит диаграммы есть <0/1>? 0
- Предлагается WINDOWS 98 и выше в качестве возможного исхода.

Анализируя список переменных и исходов БЗ для классификации ПО ЭВМ, можно заметить, что он обладает некоторой повторяемостью: часть переменных первого и второго узлов совпадают (создает БД, Создает ЭТ, создает текст и пр.). Кроме того, приходится иногда для надежности работы ЭС исходы 1 узла делать переменными для второго. Такое явление в экспертных системах допускается, но в обучающих ЭС это недопустимо, так как вызывает у обучаемых некоторое недоумение – зачем дважды спрашивать одно и то же. Однако, в данной БЗ такое повторение просто необходимо в силу многочисленности исходов 2 узла. Очевидно, что признак «Создает БД» необходим, чтобы отличить прикладное ПО от базового и системы программирования в 1 узле, и ACCESS от EXCEL – во втором. Одним из путей избавления от этого «недостатка» является увеличение числа узлов, что и показано в следующем примере. Первоначально была построена двухузловая БЗ «Типизация данных», однако добиться ее безошибочной работы даже после многочисленных сеансов обучения не удалось. Переход на три узла решил все проблемы.

Итак, рассмотрим теперь работу ЭС на примере классификации типов данных языка Паскаль. Здесь уже задаются 3 узла, каждый из которых определяется своим набором переменных:

УЗЕЛ 1

- Переменная 1 – Описывается TYPE-VAR
- Переменная 2 – Содержит другие величины
- Переменная 3 – Вводится одним READ
- Переменная 4 – Выводится одним WRITE
- Переменная 5 – Имеет индексацию своих элементов
- Переменная 6 – Содержит адрес ячейки ОЗУ
- Переменная 7 – Описывается VAR
- Исход 1 – ПРОСТОЙ ТИП
- Исход 2 – СПЕЦИАЛЬНЫЙ ТИП
- Исход 3 – СТРУКТУРНЫЙ ТИП

УЗЕЛ 2

- Переменная 1 – Создается программой
- Переменная 2 – Образует динамические объекты
- Переменная 3 – Создается пользователем
- Переменная 4 – Занимает стандартный объем памяти
- Переменная 5 – Содержит текст
- Переменная 6 – Аналог линейного массива
- Исход 1 – ВСТРОЕННЫЙ ТИП
- Исход 2 – ПОЛЬЗОВАТЕЛЬСКИЙ ТИП
- Исход 3 – СПЕЦИАЛЬНЫЙ ТИП
- Исход 4 – СТАТИЧЕСКИЙ ОБЪЕКТ
- Исход 5 – ДИНАМИЧЕСКИЙ ОБЪЕКТ

УЗЕЛ 3

- Переменная 1 – Ординальный
- Переменная 2 – Числовой
- Переменная 3 – Содержит только два значения
- Переменная 4 – Внутренняя индексация
- Переменная 5 – Внешняя индексация
- Переменная 6 – Фиксированный набор значений
- Переменная 7 – Ограниченный набор значений
- Переменная 8 – Имеет одного наследника
- Переменная 9 – Имеет двух наследников
- Переменная 10 – Упорядоченность элементов
- Переменная 11 – Удаляется из памяти программой
- Переменная 12 – Объединяет разные типы
- Переменная 13 – Располагается в ВЗУ
- Исход 1 – READ
- Исход 2 – INTEGER
- Исход 3 – CHAR
- Исход 4 – BOOLEAN
- Исход 5 – ИНТЕРВАЛЬНЫЙ
- Исход 6 – ПЕРЕЧИСЛИМЫЙ
- Исход 7 – ССЫЛОЧНЫЙ
- Исход 8 – STRING
- Исход 9 – ARRAY
- Исход 10 – SET
- Исход 11 – RECORD
- Исход 12 – FILE
- Исход 13 – ЛИНЕЙНЫЙ СПИСОК
- Исход 14 – СБАЛАНСИРОВАННОЕ ДЕРЕВО
- Исход 15 – ДЕРЕВО ПОИСКА

В базе знаний «Типизация данных» идентификация 15 типов данных осуществляется в три этапа: сначала определяется принадлежность одной из групп типов (простая, специальная, структурная), затем уточняется принадлежность более узкому классу (встроенный, пользовательский, специальный, статический, динамический), определяется окончательный результат.

Определение простого типа данных, соответствующего типам REAL, INTEGER, CHAR и BOOLEAN (узел 1) системе удается за 5 шагов путем постановки вопросов, соответствующих номерам переменной (1,4,3,6,2), при ответах на них (0,1,0,0,0).

Для ИНТЕРВАЛЬНОГО и ПЕРЕЧИСЛИМОГО типов – это удается за 6 шагов при чередовании переменных с номерами: (1,3,5,6,7,2) и их значениях (0,0,0,0,1,0).

ВСТРОЕННЫЙ тип (узел 2) определяется за 5 шагов: (1,4,3,6,2) → (0,1,0,0,0), а ПОЛЬЗОВАТЕЛЬСКИЙ тип (узел 2) – за 3 шага: (1,4,3) → (0,0,1).

В 3 узле типы INTEGER, BOOLEAN, ПЕРЕЧИСЛИМЫЙ ТИП узнаются соответственно по следующей системе вопросов и ответов:

$$(10,7,1,6,3,11,5,4,2,13,12) \rightarrow (0,1,1,1,0,0,0,0,1,0,0)$$

$$(10,7,1,6,3,4,9,2,11,13) \rightarrow (0,1,1,1,1,0,0,0,0,0,0)$$

$$(10,7,2,11,6,1) \rightarrow (0,0,0,0,1,1).$$

Представители специального типа ССЫЛОЧНЫЙ и STRING обладают совсем разными характеристиками, поэтому типы имеют несовпадающие наборы переменных для их определения. По этим же причинам исход СПЕЦИАЛЬНЫЙ фигурирует одновременно в 1 и 2 узлах. Итак, для выхода на специальный тип из ссылочного типа необходимо ответить на следующие вопросы:

$$(1,3,5,6) \rightarrow (0,0,0,1)$$

$$(1,4,3,6,2) \rightarrow (0,1,0,0,1).$$

СПЕЦИАЛЬНЫЙ тип для двух узлов в случае типа STRING определяется так:

$$(1,3,2) \rightarrow (0,1,1)$$

$$(1,4,3,2,6) \rightarrow (0,0,0,0,1).$$

Окончательно в узле 3 для идентификации типов ССЫЛОЧНЫЙ ТИП и STRING система ставит вопросы в следующем порядке:

$$(10,7,1,6,4,5,2,13) \rightarrow (0,1,0,1,0,0,0,0)$$

$$(10,7,1,6,3,4,13,5,8,2) \rightarrow (0,1,0,0,0,1,0,0,0,0).$$

СТРУКТУРНЫЙ тип определяется системой по одному вопросу «Описывается TYPE-VAR». Исход СТАТИЧЕСКИЕ ОБЪЕКТЫ для всех ее четырех типов ARRAY, SET, RECORD, FILE определяется вопросами:

$$(1,4,3,2,6,5) \rightarrow (0,0,0,0,0,1).$$

Исход ДИНАМИЧЕСКИЕ ОБЪЕКТЫ определяется еще быстрее – всего за два шага: $(1,4) \rightarrow (1,0)$.

Это объясняется тем, что характерный признак «Создается программой» присущ только этому типу данных. Как правило, такого рода переменные ЭС всегда выносит в начало постановки вопросов (согласно заложенному в нее алгоритму [3]).

Ниже следуют вопросы и ответы на них, которые определяют поиск статических объектов ARRAY, SET, RECORD, FILE в узле 3:

$$(10,7,2,11,6,1,5,13,12) \rightarrow (0,0,0,0,1,0,1,0,0)$$

$$(10,7,4,11,13,6,2,3,5,9) \rightarrow (1,1,1,0,0,0,0,0,0)$$

$$(10,7,2,11,6,1,5,12,13,4) \rightarrow (0,0,0,0,1,0,0,1,0,0)$$

$$(10,7,2,11,9,8,1,4,13) \rightarrow (0,0,1,1,0,0,0,1,1).$$

ЛИНЕЙНЫЙ СПИСОК, СБАЛАНСИРОВАННОЕ ДЕРЕВО и ДЕРЕВО ПОИСКА находятся, соответственно, с помощью следующей последовательности вопросов и ответов:

$$(10,7,2,11,8) \rightarrow (0,0,0,1,1)$$

$$(10,7,2,11,8,12) \rightarrow (0,0,0,1,0,0)$$

$$(10,7,4,11) \rightarrow (1,0,0,1).$$

Для целей обучения рассматриваемая база знаний может быть использована в различных формах.

Работа с готовой БЗ

Эта форма работы используется для контроля знаний, когда обучаемому дается задание добиться получения на выходе ЭС всех конечных объектов (от MS DOS до PowerPoint, от REAL до ДЕРЕВА ПОИСКА). При этом преподаватель может просмотреть протокол ответов обучаемых на вопросы ЭС.

Обучение ЭС

Здесь преподаватель вводит в БЗ только переменные и исходы, а обучаемый должен обучить ЭС до получения правильных ответов. Можно также обучить ЭС и предложить обучаемым добиться получения от нее безошибочных ответов путем ввода достаточного числа обучающих примеров.

Модификация БЗ

На этом этапе работы с готовой БЗ предлагается изучить ее «поведение», выявить лишние (неактивные) переменные, удалить их из базы и провести последующее «переобучение».

Добавление к БЗ новых объектов

Эта форма работы полезна тем, что обучаемые сами принимают решение о том, достаточно ли существующих в БЗ признаков для идентификации нового объекта путем переобучения ЭС, или же необходимо дополнить этот список новой (новыми) переменными.

Создание новой БЗ

В этом случае обучаемый должен предварительно продумать список всех переменных и исходов, а затем ввести их в ЭС и произвести ее обучение.

Отметим также, что созданная оболочка ЭС написана на языке Паскаль, она является универсальной и в нее можно поместить любое наполнение из любой предметной области. Составитель (разработчик) этого наполнения должен решить, по существу, три проблемы:

Выбрать правильное (желательно минимальное) число узлов для корректной идентификации (узнавания) объектов.

Подобрать для каждого узла соответствующие признаки (переменные) каждого его объекта (исхода).

Обучить путем многократной «прогонки» программы базу знаний ЭС, используя при этом либо удаление неиспользуемых признаков, либо добавление новых, если ЭС допускает ошибки.

На двух рассмотренных примерах построения обучающей экспертной системы на основе модульного проектирования, предусматривающего создание разноуровневых баз знаний, была показана технология выбора узлов, переменных и исходов, которые составляют компонент «факты» базы знаний. Был также представлен механизм создания «правил» путем ввода в БЗ примера, где каждому объекту (исходу) присваивается набор его характеристик (атрибутов), и последующего обучения экспертной системы.

Литература

1. Касторнов А.Ф. Экспертные системы как средства систематизации и контроля знаний // Ученые записки ИИО РАО. 2002. Вып 7. С. 98-114.
2. Касторнова В.А. Современное состояние научных исследований и практико-ориентированных подходов к организации и функционированию образовательного пространства. Череповец: ЧГУ, 2011. 461 с.
3. Нейлор К. Как построить свою экспертную систему: пер. с англ. М.: Энергоатомиздат, 1991. 286. с.

Kastornova Vasilina Anatol'evna,

The Federal State Budgetary Scientific Institution

*«Institute of Management of Education of The Russian Academy of Education»,
the Leading scientific researcher, Candidate of Pedagogics, Associate professor,
kastornova_vasya@mail.ru*

Andreev Aleksej Evgen'evich,

The Federal State Budgetary Scientific Institution

*«Institute of Management of Education of The Russian Academy of Education»,
the Senior research associate, Candidate of Technics, danvas3@rambler.ru*

Kastornov Anatolij Fedoseevich,

*Cherepovets State University, Professor of Department of Mathematics and
Informatics, Candidate of Pedagogics, Professor, a_kastornov@mail.ru*

Yalamov Georgij Yur'evich,

The Federal State Budgetary Scientific Institution

*«Institute of Management of Education of The Russian Academy of Education»,
the Leading scientific researcher, Candidate of Physics and Mathematics,
geo@portalsga.ru*

**APPLICATION AND PROSPECTS OF DEVELOPMENT
OF TECHNOLOGY OF MODULAR DESIGN
OF EXPERT MONITORING SYSTEMS OF KNOWLEDGE
IN EDUCATION**

Annotation

In article the questions of creation of the training expert system (TES) urged to serve one of pedagogical instruments of systematization and control of knowledge of trainees are considered. The structure of TES offered by authors is based on the modular principle of construction and includes the modules allowing to create the knowledge base in some subject domain, to fill her with contents, to make control (training) of TES, and then to use when carrying out occupations.

Keywords:

training expert system; module; knowledge base; training of expert system.