OSTIS-2013
(Open Semantic Technologies for Intelligent Systems)

УДК 004.855

ТЕХНОЛОГИЯ КОМПОНЕНТНОГО (МОДУЛЬНОГО) ПРОЕКТИРОВАНИЯ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ СИСТЕМАТИЗАЦИИ И КОНТРОЛЯ ЗНАНИЙ

Касторнов А.Ф. [*], Касторнова В.А. [**]

[*] *Череповецкий государственный университет,*
г. Череповец, Россия
a_kastornov@mail.ru

[**] *Институт информатизации образования*
Российской академии образования, г. Москва, Россия
kastornova_vasya@mail.ru

В данной работе рассмотрены вопросы построения обучающей экспертной системы (ОЭС), призванной служить одним из педагогических инструментов систематизации и контроля знаний студентов. Предлагаемая авторами структура ОЭС основывается на модульном принципе ее построения и включает в себя модули, позволяющие создавать базу знаний в некоторой предметной области, наполнять ее содержанием, производить ее настройку (обучение), а затем использовать при проведении занятий.
Ключевые слова: обучающая экспертная система; модуль; база знаний; обучение экспертной системы.

Библиотека БГУИР

ВВЕДЕНИЕ

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

Экспертные системы (ЭС) 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, выполняющий функции эксперта при решении задач из некоторой предметной области. ЭС выдают советы, проводят анализ, дают консультации, ставят диагноз. Практическое применение ЭС на предприятиях способствует эффективности работы и повышению квалификации специалистов.

Главным достоинством экспертных систем является возможность накопления знаний и сохранение их длительное время. В отличие от человека к любой информации экспертные системы подходят объективно, что улучшает качество проводимой экспертизы. При решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

Экспертная система состоит из базы знаний (части системы, в которой содержатся факты), подсистемы вывода (множества правил, по которым осуществляется решение задачи), подсистемы объяснения, подсистемы приобретения знаний и диалогового процессора. При построении подсистем вывода используют методы решения задач искусственного интеллекта.

Основными отличиями ЭС от других программных продуктов являются использование не только данных, но и знаний, а также специального механизма вывода решений и новых знаний на основе имеющихся. Знания в ЭС представляются в такой 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, который не был предусмотрен. Более того, алгоритм обработки знаний заранее неизвестен и строится по ходу

решения задачи на основании эвристических правил. Решение задачи в ЭС сопровождается понятными пользователю объяснениями, качество получаемых решений обычно не хуже, а иногда и лучше достигаемого специалистами. В системах, основанных на знаниях, правила (или эвристики), по которым решаются проблемы в конкретной предметной области, хранятся в базе знаний. Проблемы 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 важная компонента экспертной системы, на которой основаны ее «интеллектуальные способности». В отличие от всех остальных компонент ЭС, база знаний - «переменная» часть системы, которая может пополняться и модифицироваться инженерами знаний и опыта использование ЭС, между консультациями (а в некоторых системах и в процессе консультации). Существует 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). Тем самым, 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 базе знаний служат для представления эвристических знаний (эвристик), 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 его поведение. База знаний изменяется в соответствии с поведением объекта. Примером этого обучения 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 незнании самого материала. На наш взгляд, сама тестовая форма проверки знаний далека от совершенства и не может качественно оценить знания, особенно при организации итогового контроля знаний по завершению изучения какой-либо крупной темы учебного курса. С помощью тестирования часто бывает трудно оценить знания учащихся с точки зрения выявления понимания ими и качественных характеристик, и взаимосвязей изучаемых в курсе базовых понятий.

В качестве примера можно привести изучение темы "Программное обеспечение ЭВМ" в курсе информатики. Известно, что здесь принято проводить классификацию ПО на три группы: базовое ПО, системы программирования и прикладное ПО. Аналогичная ситуация имеет место и при классификации типов данных в языке 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 особенно важно, не всегда удачен со стороны составителя теста. На наш взгляд, эта проблема легко решается, если в качестве инструмента проверки и систематизации знаний использовать экспертную систему учебного назначения.

Экспертные системы создаются для решения разного рода проблем. Основные типы их деятельности можно сгруппировать в следующие категории: интерпретация, прогноз, диагностика, наблюдение, проектирование, отладка, обучение, управление. Экспертные системы, выполняющие обучение, подвергают диагностике, "отладке" и исправлению (коррекции) поведение обучаемого. Обучающие системы создают модель того, что обучающийся знает и как он эти знания применяет к решению проблемы [Касторнова, 2011].

Разработанная нами экспертная система учебного назначения состоит 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 ВЗУ в ОЗУ, а также основной модуль *диалога с пользователем* [Касторнов, 2002]. Первые модули заполняются экспертом (преподавателем), а с последними двумя работают учащиеся. При инициализации системы эксперт определяет структуру базы знаний, где указываются число узлов, переменных и исходов, которые затем вводятся в эту базу. 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

Библиотека БГУИР

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 система обучается на конкретных примерах, до тех пор, пока она не перестанет ошибаться. Полученная база знаний записывается в виде системы файлов на диске. Готовая система потом может быть использована учащимися после ее загрузки с внешнего носителя. Система позволяет также после работы с ней производить ее модификацию путем удаления или добавления новых переменных и исходов с последующим ее "дообучением".

Рассмотрим более подробно работу ЭС на примере классификации программного обеспечения ЭВМ. На начальном этапе мы задаем системе исходную информацию 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 паки
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 всех двух узлов указывается наличие его характерных признаков.

За вводом примера наступает этап тренировки системы (модуль «тренировка»), в котором по соответствующему алгоритму [Нейлор, 1991] заполняется массив правил, позволяющих по набору некоторых 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 системы (модуль «обучение»). На этом этапе система, ориентируясь на сформированный ранее массив правил, ставит эксперту уточняющие вопросы относительно наличия тех или свойств выбранному исходу. Обучение заканчивается, как только система перестает ошибаться. Теперь результаты обучения заносятся в массив правил (модуль «запоминание»), при этом на внешнем носителе сохраняются и все ранее созданные массивы (в общей сложности семь файлов). Система готова к работе и ее можно использовать в качестве эксперта.

Работа пользователя (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 надо 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 получается за 2-3 хода, порой задается более десятка вопросов (все зависит, конечно, от общего числа введенных в БЗ переменных и искомых исходов). В нашем примере с ПО ЭВМ для идентификации исхода *ACCESS* 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*

Библиотека БГУИР

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 матема-ческие 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 не допустимо, так вызывает у студентов 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 нами 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

Библиотека ВГУИР

более узкому классу (*встроенный, пользовательский, специальный, статический, динамический*), а затем определяется окончательный результат.

Определение простого типа данных, соответствующего типам *REAL, INTEGER, CHAR* и *BOOLEAN* (узел 1) системе удается за 5 шагов путем постановки вопросов, соответствующим 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) → (0,1,1,1,0,0.0,0,1,0.0)
(10,7,1,6,3,4,9,2,11,13) → (0,1,1,1,1,0,0,0,0,0,0)
(10,7,2,11,6,1) → (0,0,0,0,1,1).

Представители специального типа *ССЫЛОЧНЫЙ* и *STRING* обладают совсем разными характеристиками, вот почему этот тип имеет совсем разный набор переменных для его определения. По этим же соображениям исход *СПЕЦИАЛЬНЫЙ* фигурирует одновременно в 1 и 2 узлах. Итак, для выхода на специальный тип относительно ссылочного типа необходимо ответить соответственно на следующие вопросы:

(1,3,5,6) → (0,0,0,1)
(1,4,3,6,2) → (0,1,0,0,1).

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

(1,3,2) → (0,1,1)
(1,4,3,2,6) → (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) → (0,1,0,1,0,0,0,0)
(10,7,1,6,3,4,13,5,8,2) → (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) → (0,0,0,0,0,1).

Исход *ДИНАМИЧЕСКИЕ ОБЪЕКТЫ* определяется еще быстрее – всего за два хода:
(1,4) → (1,0).

Это объясняется тем, что характерный признак «*Создается программой*» является присущим только этому типу данных. Как правило, такого рода переменные экспертная система всегда выносит на начало постановки вопросов (согласно заложенному в нее алгоритму [Нейлор, 1991]).

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) → (0,0,0,0,1,0,1,0,0)
(10,7,4,11,13,6,2,3,5,9) → (1,1,1,0,0,0,0,0,0,0)
(10,7,2,11,6,1,5,12,13,4) →(0,0,0,0,1,0,0,1,0,0)
(10,7,2,11,9,8,1,4,13) → (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) → (0,0,0,1,1)
(10,7,2,11,8,12) → (0,0,0,1,0,0)
(10,7,4,11) → (1,0,0,1).

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

1. *Работа с готовой и обученной базой знаний*

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

2. Обучение БЗ

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

3. 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".

4. Добавление к БЗ новых объектов

Эта форма работы полезна тем, что учащиеся, которык добавляя новый объект, должны увидеть, достаточно ли существующих в БЗ признаков для идентификации нового объекта путем переобучения системы, или же необходимо дополнить этот список новой (новыми) переменными.

5. Создание новой БЗ

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

Отметим также, что созданная нами оболочка ЭС написана на языке Паскаль, она является универсальной и в нее можно поместить любое наполнение из любой предметной области.

Библиотека БГУИР

Составитель (разработчик) этого наполнения должен решить, по существу, три проблемы:

1. Выбрать правильное (желательно минимальное) число узлов для для правильной идентификации (узнавания) объектов.
2. Подобрать для каждого узла соответствующие признаки (переменные) каждого его объекта (исхода).
3. Обучить путем многократной «прогонки» программы базу знаний ЭС, используя при этом либо удаление неиспользуемых признаков, либо добавления новых, если система допускает ошибки.

ЗАКЛЮЧЕНИЕ

В данной статье был рассмотрен пример построения обучающей экспертной системы на основе модульного проектирования, предусматривающего создания разноуровневых баз знаний. На двух рассмотренных случаях была показана технология выбора узлов, переменных и исходов, которые составляют компоненту «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 системы.

Библиографический список

[Нейлор, 1991] Нейлор К. Как построить свою экспертную систему: Пер. с англ. – М: Энергоатомиздат, 1991. – 286 с.

[Касторнов, 2002] Касторнов А.Ф. Экспертные системы как средства систематизации и контроля знаний. Ученые записки ИИО РАО, выпуск 7, Москва, 2002.- с. 98-114.

[Касторнова, 2011] Касторнова В.А. Современное состояние научных исследований и практико-ориентированных подходов к созданию и функционированию образовательного пространства. Монография. Череповец: ЧГУ, 2011. – 461 с.

TECHNOLOGY OF COMPONENT (MODULAR) DESIGN OF EXPERT SYSTEM FOR SYSTEMATIZATION AND CONTROL OF KNOWLEDGE

Kastornov A.F. *, Kastornova V.A. **

** Cherepovets State University, Cherepovets, Russia*

a_kastornov@mail.ru

***Institute of Informatization of Education Russian Academy of Education, Moscow, Russia*

kastornova_vasya@mail.ru

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

INTRODUCTION

The purpose of this work is practical realization of technology of modular design of expert system. In work the program creating the knowledge base which includes the modules, allowing to pack the various levels (knots) containing the list of questions (variables) which are raised by ES to the user correctly to identify this or that object (outcome) in some subject domain is described Pascal.

MAIN PART

The expert system consists of the knowledge base, containing the facts and rules, subsystems of a conclusion of an explanation, acquisition of knowledge and the dialogue processor. Rules serve in the knowledge base for submission of informal rules of the reasoning developed by the expert on the basis of experience of its activity. The expert systems which are carrying out training, subject to diagnostics, debugging and correction behavior of the trainee. Training systems create model of that the being trained knows and as it applies this knowledge to a solution. The expert system of educational appointment consists of modules: system initialization, input of examples, training, training, knowledge base storing on the external carrier (in the form of a set of massifs of the facts and rules), removal and addition of variables and outcomes, knowledge base loading from VZU in the RAM, and also the main module of dialogue with the user. Originally for BZ initial information on number of knots, number of variables and outcomes for each knot and the name of variables is set. After input of basic data there is a task of an example in which for each outcome of all knots existence of its characteristic signs is specified. Behind input of an example there comes a stage of training of system in which the array of rules is filled. After training it is trained systems. Results of training are brought in an array of rules. For training this knowledge base can be used in various forms: work with the ready and trained knowledge base, BZ training, BZ modification, addition to BZ of new objects, creation of new BZ.

CONCLUSION

In this article an example of creation of training expert system on the basis of the modular design providing creations of raznourovnevy knowledge bases was reviewed. On two considered cases the technology of a choice of knots, variables and outcomes which make to the facts component of the knowledge base was shown. The mechanism of creation was also presented "governed" a way of input to example BZ where the set of its characteristics (attributes), and the subsequent training of system is appropriated to each object (outcome).

Библиотека БГУИР