УДК 004.8

ПОСТРОЕНИЕ НЕЧЕТКОЙ ГИБРИДНОЙ СИСТЕМЫ ОБУЧЕНИЯ И КОНТРОЛЯ

И. Ф. Астахова, А. И. Шашкин, И. В. Сухотерина

Воронежский государственный университет

Поступила в редакцию 22.01.2015 г.

Аннотация. Рассматривается построение нечеткой гибридной системы обучения и контроля. Для классификации учащихся используется сеть Кохонена, для предсказания развития учебного курса предлагаются методы нечеткой логики.
Ключевые слова: информационная система, нейронная сеть, нечеткая логика, учебный курс.
Annotation. Distant learning system is worked out. Kochonen neuron net is considered for classification. The fuzzy logic methods are considered for predication of development course.
Keywords: the information system, neuron system, fuzzy logic, learning course.

ВВЕДЕНИЕ

Проблемы проектирования и разработки систем обучения и контроля связаны с существующими противоречиями: а) между необходимостью качественного и эффективного обучения персонала, обеспечивающего использование автоматизированных информационно-управляющих систем, и отсутствием по данным системам современных учебных средств, ориентированных на использование образовательных критериев; б) повышения эффективности технологического процесса разработки программного обеспечения и отсутствием единого математического подхода к совершенствованию методики разработки этих курсов [1–2].

Вопросами автоматизации учебного процесса занимаются многие ученые, практически, во всех вузах России: в Московском государственном университете им. М. В. Ломоносова (Сухомлин В. А.), в Московском государственном гуманитарном университете им. М. А. Шолохова (Монахов В. М.), в Самарском государственном аэрокосмическом университете им. акад. С. П. Королева (Зеленко Л. С.), в Марийском государственном университете (Токтарова В. И.), в Северном федеральном университет им. М. В. Ломоносова (Чиркова Л. Н.) и др [1]. В Воронежском государственном университете этим вопросам также всегда уделялось достаточно внимания (см., например, [3–4]).

Целью настоящей работы является создание системы обучения и контроля, включающей обучающую (для работы обучаемого) и администрирующую (для работы преподавателя) части. Разрабатываемая система должна быть универсальной, не должна зависеть от конкретного курса, т. е. иметь возможность «закладывать» в нее любой курс по любому учебному предмету. Преподаватель должен иметь возможность подключать учебные курсы, редактировать их и, при необходимости, удалять, а также иметь доступ к результатам обучения. Обучаемые должны иметь возможность выбирать учебный курс, изучать теоретические положения данного курса, проходить тестирование и получать навыки

© Астахова И. Ф., Шашкин А. И., Сухотерина И. В., 2015

решения задач. В результате работы автоматизированного курса обучаемый должен получать: знания по выбранной предметной области, умения применять различные методы и алгоритмы, навыки решения задач.

Для разработки системы необходимо создать модель системы учебного курса, провести классификацию студентов по имеющимся у них знаниям; выработать предложения по совершенствованию учебного курса. Система должна не только решать поставленные перед ней задачи, но и быть удобной в использовании и отвечать высокому уровню современных интерактивных приложений.

1. НЕЧЕТКАЯ ГИБРИДНАЯ СИСТЕМА ОБУЧЕНИЯ И КОНТРОЛЯ

Данная работа посвящена созданию нечеткой гибридной системы обучения и контроля. Гибридными системами называются такие, в которых используется две или более различных компьютерных технологии [5]. Употребление нескольких компьютерных технологий при небольших затратах времени позволяет провести классификацию знаний студентов, оценить их знания и количественно и качественно и выдать рекомендации по совершенствованию преподавания указанного курса. Использование нечеткости при прогнозировании развития курса позволяет применять недостоверную или неполную информацию о знаниях студентов, полученную в течение нескольких лет по разным разделам определенного курса.

Схема нечеткой гибридной системы обучения и контроля представлена на рис.1., она представлена с подсистемами в [6].

Из представленной схемы видно, что в этой системе объединены разделы, которые присутствуют по отдельности в разных подсистемах.

2. ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ

При создании системы было принято решение об обязательном учете разного уровня знаний студентов. Для определения уровня

Рис.1. Схема нечеткой гибридной системы обучения и контроля

знаний студентов и разделения их на соответствующие группы была применена нейрокомпьютерная технология, а именно, задача классификации решалась с использованием сети Кохонена [7].

В целом, задача классификации заключается в разбиении объектов на классы, причем основой разбиения служит вектор параметров объекта. Сами классы часто бывают неизвестными заранее, а формируются динамически при тестировании. Один из простых подходов к классификации состоит в том, чтобы предположить существование определенного числа классов и произвольным образом выбрать координаты прототипов. Затем каждый вектор из набора данных связывается с ближайшим к нему прототипом, и новыми прототипами становятся центроиды всех векторов, связанных с исходным прототипом.

На этих принципах основано функционирование сети Кохонена, обычно используемой для решения задач классификации. Данная сеть обучается без учителя на основе самоорганизации. По мере обучения векторы весов нейронов становятся прототипами классов – групп векторов обучающей выборки. На этапе решения информационных задач сеть относит новый предъявленный образ к одному из сформированных классов.

Как известно [7], сеть Кохонена состоит из одного слоя нейронов. Число входов каждого нейрона n равно размерности вектора параметров объекта. Количество нейронов m совпадает с требуемым числом классов, на которые нужно разбить объекты. Меняя число нейронов, можно динамически изменять число классов.

Обучение сети начинается с задания небольших случайных значений элементам весовой матрицы w. В дальнейшем происходит процесс самоорганизации, состоящий в модификации весов при предъявлении на вход векторов обучающей выборки. Каждый столбец весовой матрицы представляет собой параметры соответствующего нейрона-классификатора. Для каждого j-го нейрона ($j=\overline{1,m}$) определяется расстояние от него до входного вектора X:

$$d_j = \sum_{i=1}^{n} (x_i - w_{ij})^2.$$

Далее выбирается нейрон с номером k, $1 \le k \le m$, для которого это расстояние минимально (то есть сеть относит входной вектор к классу с номером k). На текущем шаге N-м обучения будут модифицироваться только веса нейронов из окрестности нейрона k:

$$w_{ij}^{N+1} = w_{ij}^{N} + \alpha_N (x_i - w_{ij}^{N}),$$

где α_N – темп обучения, выражаемый в виде числа.

Первоначально в окрестности любого из нейронов находятся все нейроны сети, но с каждым шагом эта окрестность сужается. В конце этапа обучения участвуют в обучении только веса нейрона с номером k. Темп обучения α_N с течением времени также уменьшается (часто полагают [8] $\alpha_0 = 0.9$, $\alpha_{N+1} = \alpha_N - 0.001$). Образы обучающей выборки предъявляются последовательно, и каждый раз происходит подстройка весов.

Алгоритм обучения сети Кохонена состоит из следующих шагов.

1. Инициализация сети.

Присвоение весовым коэффициентам сети $w_{ij}, i=\overline{1,n}$, $j=\overline{1,m}$ малых случайных значений. Задаются значения: α_0 – начальный темп обучения и D_0 – максимальное расстояние между весовыми векторами (столбцами матрицы W).

2. Предъявление сети нового входного сигнала X.

3. Вычисление расстояния от входа X до всех нейронов сети:

$$d_j = \sum_{i=1}^{n} (x_i - w_{ij}^{N})^2, \; j=\overline{1,m}.$$

4. Выбор нейрона k, $1 \le k \le m$ с наименьшим расстоянием d_k от входа до нейронов сети.

5. Настройка весов k-го нейрона и всех нейронов, находящихся от него на расстоянии, не превосходящем D_N:

$$w_{ij}^{N+1} = w_{ij}^{N} + \alpha_N (x_i - w_{ij}^{N}).$$

6. Уменьшение значений α_N, D_N.

Шаги 2–6 повторяются до тех пор, пока веса не перестанут меняться (или пока суммарное изменение всех весов станет очень малым).

После обучения сети классификация знаний студентов выполняется посредством подачи испытуемого вектора на вход сети и вычисления расстояния от него до каждого нейрона с последующим выбором нейрона с наименьшим расстоянием в качестве индикатора правильности классификации.

Для данной системы на вход реализованной сети Кохонена подается маска ответов на пять классификационных вопросов. Сеть распределяет ответы на три класса: 0 – недостаточный для дальнейшего прохождения основного тестирования; 1 – минимально достаточный для дальнейшего прохождения основного тестирования; 2 – достаточный для дальнейшего прохождения основного тестирования.

Задача классификации заключается в разбиении объектов на классы, причем основой разбиения служит вектор параметров объекта. Сами классы часто бывают неизвестными заранее, а формируются динамически. Один из самых простых подходов к классификации состоит в том, чтобы предположить существование определенного числа классов и произвольным образом выбрать координаты прототипов. Затем каждый вектор из набора данных связывается с ближайшим к нему прототипом, и новыми прототипами становятся центроиды всех векторов, связанных с исходным прототипом.

3. АЛГОРИТМ ПРОГНОЗИРОВАНИЯ РАЗВИТИЯ КУРСА С ПОМОЩЬЮ НЕЧЕТКОЙ ЛОГИКИ

Прогнозирование представляет собой научно-обоснованное оценивание будущих состояний интересующего нас объекта. Прогнозы могут быть как количественными, так и качественными, в данном случае будет даваться количественный прогноз.

Нечеткий подход позволяет опираться на любую априорную информацию и давать *нечеткое решение*, соответствующее требованиям по уровню точности.

В настоящей работе был использован алгоритм прогнозирования, состоящий из следующих шагов [9].

1. Определение универсального множества U, которое представляет собой интервал между наименьшей и наибольшей вариациями коэффициентов правильных ответов в группе: $U = [V_{\min} - D, V_{\max} + D]$, где $V_{\min}$ – наименьшая вариация временного ряда; $V_{\max}$ – наибольшая вариация временного ряда (под термином «вариация» для текущего года имеется в виду разница между коэффициентами правильных ответов в группе в текущем и предыдущем годах).

2. Деление универсального множества U на несколько интервалов равной длины, включающих различные значения вариаций $U = \{u_i\}$ и определение середин интервалов u_i^{cp}.

3. Введение лингвистической переменной и определение соответствующих лингвистических значений $A = A_i$, $i = 1,...m$, т.е. определение множества нечетких множеств $F(t)$. Нечеткие множества A_i в универсальном множестве U определяются с помощью формулы:

$$\mu_{A_i}(u_i) = \frac{1}{1 + \left(C\left(V - u_i^{cp}\right)\right)^2},$$

где C – постоянное число, которое подбирается таким образом, чтобы обеспечить преобразование четких количественных чисел в нечеткие, т. е. их вхождение в интервал [0; 1]:

$$A = \left\{\left(\mu_A(u_1), u_1\right), ..., \left(\mu_A(u_n), u_n\right)\right\},$$

$$u_i \in U, \mu_A(u_i) \in [0,1].$$

4. Фаззификация исходных данных, т. е. преобразование четких количественных значений в нечеткие. Эта операция позволяет в значениях функций принадлежности отразить соответствующие количественным значениям качественные представления об ответах в данной группе:

$$A_t = \left\{\left(\mu_{A_t}(u_1)/u_1\right), ..., \left(\mu_{A_t}(u_n)/u_n\right)\right\},$$

$$\mu_{A_t}(u_i) = \frac{1}{1 + \left(C\left(V_t - u_i^{cp}\right)\right)^2},$$

где $t = 1, 2, \ldots, K$; $i = 1, 2, \ldots, n$; K – количество лет, используемых в прогнозе; V_t – вариация года t (разница количества правильных ответов в группе в году t и году $t-1$); $C = \text{const}$.

5. Выбор параметра $1 < w < L$, соответствующего отрезку времени, предшествовавшему текущему году, где L – общее число лет, учитываемых в расчетах.

6. Вычисление матрицы нечетких отношений:

$$R(t) = O^W(t) \cap K(t),$$

где $O^W(t)$ – операционная матрица, содержащая в качестве строк величины A_t;

$t \in [K - W, K]$, $K(t)$ – матрица-критерий, размера $1 \times m$ для прогнозируемого года t.

7. Прогнозирование количества тестируемых студентов:

$$F(t) = [\max(R_{11}, ..., R_{W1}), ..., \max(R_{1m}, ..., R_{Wm})].$$

8. Деффазификация полученного результата, т. е. переход от нечетких значений к четким (количественным). Для этого ожидаемая вариация на текущий год вычисляется по формуле:

$$V(t) = \frac{\sum_{i=1}^{m} \mu_t(u_i)\, u_i^{cp}}{\sum_{i=1}^{m} \mu_t(u_i)}.$$

В качестве лингвистической переменной была использована переменная «вариация количества правильных ответов в группе», которая имела следующие значения: {(отрицательная значительная вариация), (отрицательная малая вариация), (нет изменений),

Таблица 1

Сводная информация об ответах студентов по годам

№	Группа вопросов	2009 г.	2010 г.	2011 г.	2012 г.	2013 г.
1	Механизм удаленного доступа к БД	6	13	15	18	22
2	Модели данных и технология проектирования БД	11	17	23	24	26
3	Теоретические аспекты построения БД	16	21	27	30	35
4	Реляционные БД	30	27	29	32	33
5	Реляционная алгебра и реляционное исчисление	22	18	30	34	34
6	Связи и транзакции	13	18	24	27	29
7	Основы языка SQL	35	33	35	40	42
8	Основные операторы языка SQL	37	36	38	36	38
9	Простейшие запросы на языке SQL	24	30	33	42	44
10	Сложные запросы на языке SQL	15	17	25	37	42
	Всего сдавало	38	40	42	44	46
Сводная таблица о правильных ответах студентов к их общему числу по годам в процентах						
1	Механизм удаленного доступа к БД	15,79	32,50	35,71	40,91	47,83
2	Модели данных и технология проектирования БД	28,95	42,50	54,76	54,55	56,52
3	Теоретические аспекты построения БД	42,11	52,50	64,29	68,18	76,09
4	Реляционные БД	78,95	67,50	69,05	72,73	71,74
5	Реляционная алгебра и реляционное исчисление	57,89	45,00	71,43	77,27	73,91
6	Связи и транзакции	34,21	45,00	57,14	61,36	63,04
7	Основы языка SQL	92,11	82,50	83,33	90,91	91,30
8	Основные операторы языка SQL	97,37	90,00	90,48	81,82	82,61
9	Простейшие запросы на языке SQL	63,16	75,00	78,57	95,45	95,65
10	Сложные запросы на языке SQL	39,47	42,50	59,52	84,09	91,30

(малая положительная вариация), (значительная положительная вариация)}.

С использованием предложенной системы была получена таблица (см. табл. 1) результатов прогнозирования количества правильных ответов по курсу «Базы данных и экспертные системы» у студентов 4 курса факультета прикладной математики, информатики и механики.

Как следует из таблицы, в 2013 г. наименьшее число правильных ответов будет дано на вопросы из групп «Механизм удаленного доступа к БД», «Модели данных и технология проектирования БД» и «Сложные запросы на языке SQL». Ошибка прогнозирования составила ≈ 20 %.

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НЕЧЕТКОЙ ГИБРИДНОЙ СИСТЕМЫ

В качестве средства реализации поставленных задач была выбрана связка программных средств PHP+MySQL+HTML [10]. Эта связка была выбрана по следующим причинам. PHP – язык программирования, позволяющий организовать удобный доступ к базам данных. Взаимодействие PHP+HTML дает неплохие возможности представления информации пользователю. Благодаря тому, что PHP является серверным языком сценариев, разработанные алгоритмы выполняются на сервере, выдавая пользователям конкретные результаты. Это позволяет получить доступ к разработанному программному обеспечению из любой компьютерной сети учебного заведения.

Интерфейс пользователя системы представлен на рис. 2 в виде совокупности взаимосвязанных страниц.

Как видно из рис. 2, интерфейс и соответствующее программное обеспечение позволяет выполнять все задачи обучения и контроля по учебному курсу.

Приложение имеет объектно-ориентированную структуру. Главным её узлом является сценарий **index.php**. Фактически, пользователь, запрашивая любую информацию, всегда вызывает только этот сценарий, который, в свою очередь, выполняет координирующую и управляющую функцию. Сценарий **index.php**, получая запрос от пользователя в виде набора параметров, определяет необходимый

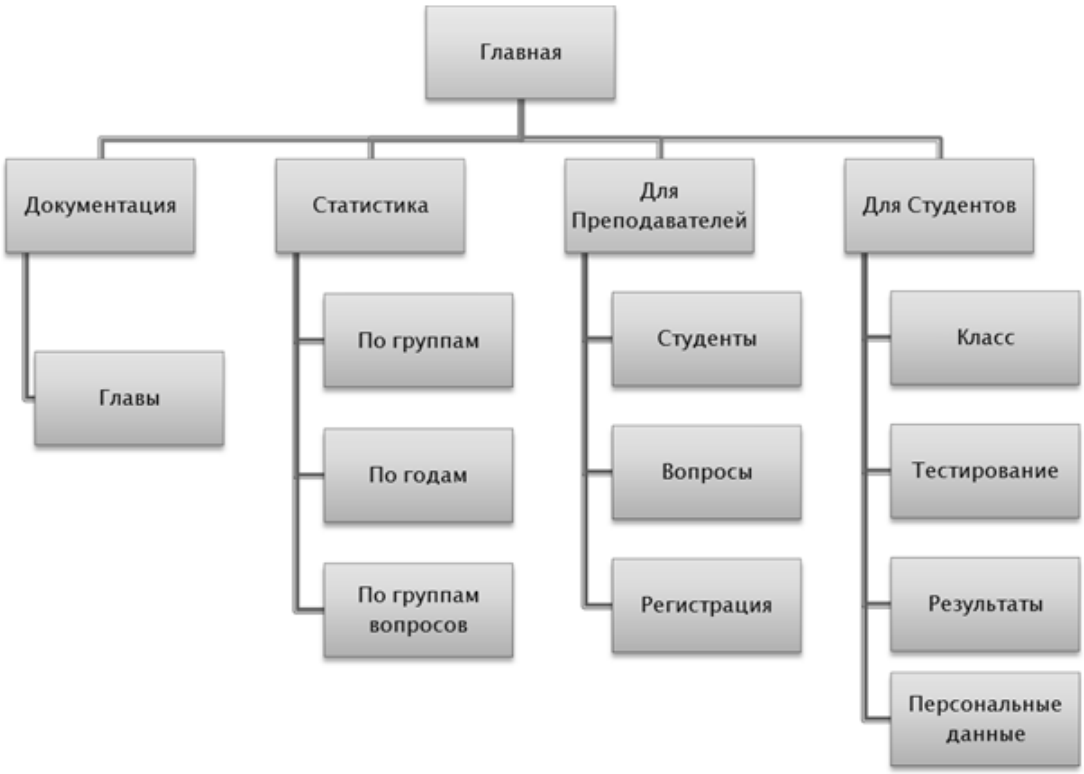

Рис. 2. Интерфейс пользователя системы обучения и контроля

модуль для выполнения и вызывает его. Все модули являются потомками чисто абстрактного класса и наследуются либо от него, либо друг от друга, имея, таким образом, единую структуру.

Для связи модулей с базой данных используется библиотека **db.class.php.** Она включает в себя необходимые для пользователя функции манипулирования данными в удобном для него виде.

Получив необходимую информацию из базы и обработав ее соответствующим образом, модуль передает управления пакету **Smarty**. **Smarty** – это компилирующий обработчик шаблонов для PHP и на PHP. Говоря более точно, он предоставляет собой один из инструментов, который позволяет добиться отделения прикладной логики и данных от представления. Это очень удобно в ситуациях, когда программист и верстальщик шаблона – различные люди. Одно из предназначений **Smarty** – отделение логики приложения от представления. Конечно же, шаблоны могут содержать в себе и логику, но лишь при условии, что эта логика необходима для правильного представления данных. Такие задачи, как подключение других шаблонов, чередующаяся окраска строчек в таблице, приведение букв к верхнему регистру, циклический проход по массиву для его отображения и т. д. – всё это примеры логики представления.

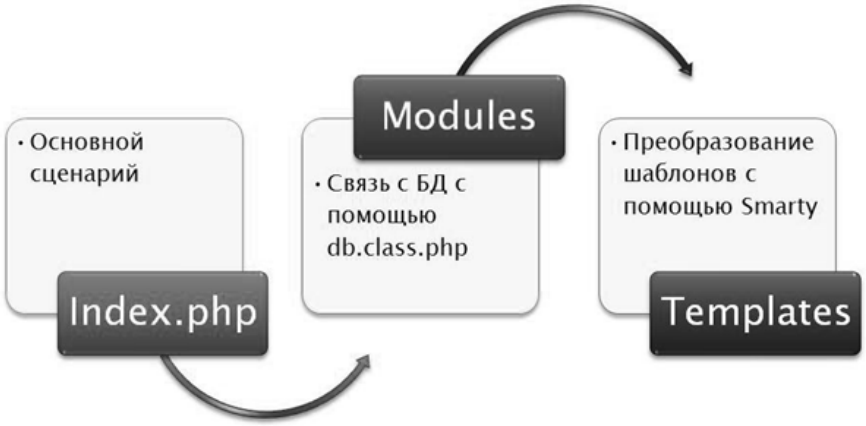

Рис.3. Взаимодействие функциональных блоков внутри системы

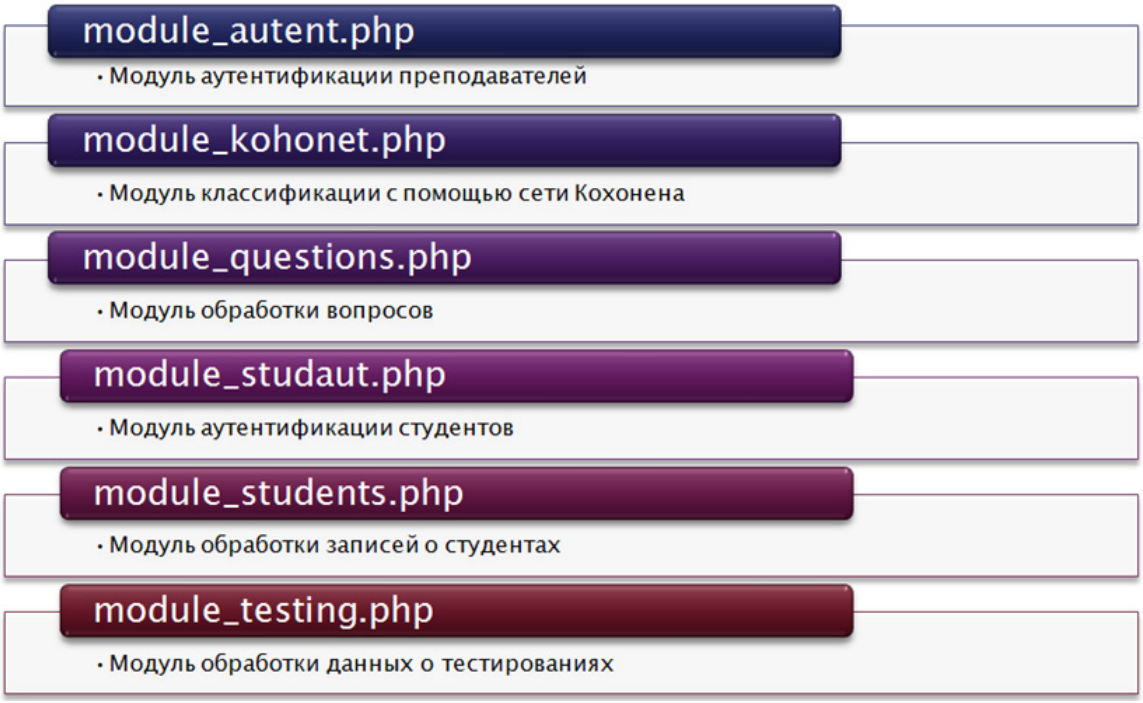

Рис. 4. Основные модули проекта

Примерная последовательность выполнения сценариев и представления конечной информации пользователю отражена на рис. 3.

Как видно из рис. 4, основные модули системы обеспечивают использование любых тестов по учебным дисциплинам. Каждый модуль представляет собой класс, полностью реализующий поставленную узкоспециализированную задачу. Таким образом достигается разграничение прав доступа и распределение задач между отдельными частями системы.

В целом, можно сказать, что созданная в настоящей работе нечеткая гибридная система решает все поставленные перед ней задачи и позволяет провести эффективное обучение и контроль знаний учащихся по любому учебному курсу.

ЗАКЛЮЧЕНИЕ

В настоящей работе создана нечеткая гибридная система, используемая для обучения студентов и контроля их знаний. Сформирована структура системы, определено взаимодействие её функциональных блоков. Разработан алгоритм прогнозирования развития учебного курса с помощью нечеткой логики. Создан удобный интерфейс как для преподавателя, так и для студента.

Особенностью системы является использование нескольких компьютерных технологий для ее разработки, а именно, нейронных сетей и нечеткой логики, что дает возможность провести классификацию студентов по имеющемуся у них уровню знаний и количественно, и качественно. Аппарат нечеткой логики позволяет при небольшом количестве информации дать рекомендацию преподавателю о направлениях совершенствования учебного курса.

Система прошла практическую апробацию на курсе «Базы данных и экспертные системы» на факультете ПММ в ВГУ в течение последних 10 лет. Однако она может быть использована для проведения любого учебного курса и, одновременно, для осуществления контроля знаний учащихся.

СПИСОК ЛИТЕРАТУРЫ

1. Современные информационные технологии и ИТ-образование / Сб. избранных трудов IX международ. науч.-практ. конф. / под ред. В. А. Сухомлина. Москва : ИНТУИТ.РУ, 2014. – 957 с.

2. *Власенко А. А.* Разработка алгоритма построения учебного плана в рамках адаптивной системы дистанционного образования / А. А. Власенко, А. И. Шашкин // Вестник Воронежского государственного университета. Серия Системный анализ и информационные технологии. – 2014. – № 1. – С. 107–114.

3. *Запрягаев С. А.* Компьютерные технологии в учебном процессе Воронежского госуниверситета / С. А. Запрягаев, С. Д. Кургалин, А. П. Толстобров // Материалы семинаров научно-образовательного центра «Волновые процессы в неоднородных и нелинейных средах» / Отв. ред. А. С. Сидоркин. – Воронеж, 2003. – С. 195–211.

4. *Запрягаев С. А.* Развитие информационных образовательных технологий в Воронежском государственном университете / С. А. Запрягаев, С. Д. Кургалин, А. П. Толстобров // Технологии информационного общества – Интернет и современное общество : труды V Всероссийской объединенной конференции, г. Санкт-Петербург, 25–29 ноября 2002 г. – Санкт-Петербург, 2002. – С. 179–181.

5. *Батыршин И. З.* Нечеткие гибридные системы. Теория и практика / под ред. Н. Г. Ярушкиной. – Москва : ФИЗМАТЛИТ. – 2007. – 208 с.

6. *Астахова И. Ф.* Модель для обучающей контролирующей системы / И. Ф. Астахова, И. В. Сухотерина // Вестник Воронежского государственного университета. Серия Системный анализ и информационные технологии. – 2014. – № 1. – С. 95–101.

7. *Горбань А. Н.* Нейронные сети на персональном компьютере / А. Н. Горбань, Д. А. Россиев. – Новосибирск : Наука, 1996. – 276 с.

8. *Майоров А. Н.* Теория и практика создания тестов для системы образования. – Москва : Интеллект центр, 2001. – 296 с.

9. *Борисов А. Н.* Модели принятия решений на основе лингвистической переменной / А. Н. Борисов, А. В. Алексеев, О. А. Крумберг и др. – Рига : Зинатне, 1982. – 256 с.

10. Профессиональное PHP программирование / Дж. Кастаньетто, Х. Рафат, С. Шуман и др. – Санкт-Петербург : Символ-Плюс, 2001. – 912 с.

Астахова И. Ф. – д.т.н., профессор кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики Воронежского государственного университета.
Тел.: (473)2-208-698, E-mail: astachova@list.ru

Astachova I. F. – Doctor Technical Science. Professor of mathematical software, Voronezh State University.
Tel.: (473)2-208-698, E-mail: astachova@list.ru

Шашкин А. И. – доктор физико-математических наук, профессор, заведующий кафедрой математического и прикладного анализа, Воронежский государственный университет.
E-mail: dean@amm.vsu.ru.

Shashkin A. I. – Doctor of Physical and Mathematical Sciences, Head of Department of Mathematics and Applied Analysis, Voronezh State University.
E-mail: dean@amm.vsu.ru.

Сухотерина И. В. – аспирантка кафедры математического и прикладного анализа факультета прикладной математики, информатики и механики Воронежского государственного университета.
Тел.: 8 (920) 460-87-55
E-mail: SuhoterinaIV@gmail.com

Suhoterina I. V. – Graduate student of the Voronezh State University.
Tel.: 8 (920) 460-87-55
E-mail: SuhoterinaIV@gmail.com