

Технология и практика обучения

Олег Викторович Рогозин, кандидат технических наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана

ФОРМИРОВАНИЕ НАБОРОВ ТЕСТОВ В SCORM-СИСТЕМАХ НА ОСНОВЕ НЕЙРО-НЕЧЁТКИХ МОДЕЛЕЙ

В статье представлен метод эффективного подбора тестов и формирования фонда оценочных средств (ФОС) оценки уровня подготовки обучаемого при использовании дистанционных систем: определение формата и структуры учебных материалов; формирование методики формирования проверочных заданий и тестов; разработка системы тестирования с использованием оценки качественных показателей результата обучения на основе нейро-нечётких моделей. Создание системы дистанционного обучения включает использование модулей тестирования обучаемого. Представим, что в процессе обучения необходимо освоить учебный материал, сформировать компетенции, получить навыки и умения, приобрести новые знания.

Ключевые слова: модель дистанционного обучения, нечёткие знания, метаданные, нечёткий контроллер

ОБОБЩЁННАЯ МОДЕЛЬ ДИСТАНЦИОННОГО ПРОЕКТА ОБУЧЕНИЯ

Введём понятие дистанционного проекта обучения и сформируем набор показателей эффективности обучения на основе обобщённой модели проекта в виде: $L < L_n, G, T, R >$ (вопрос редактору. После знака «больше» нет данных), где L — проект обучения, L_n — начальное состояние инновационного проекта, G — цель, T — наборы тестов при

переходе от одного состояния проекта к другому, R — ресурсы. Уровень подготовки считается завершённым, когда не менее восьмидесяти пяти процентов тестов пройдены успешно, при этом оценка завершения тестов производится на основании качественных показателей и использовании математического аппарата нечёткой логики. Последовательность прохождения теста и, следовательно, степень освоения материала определяются управляющим графиком



обучения, который строится с использованием заголовков модулей изучаемой дисциплины.

ПРИМЕНЕНИЕ НЕЧЁТКОЙ ЛОГИКИ

Применение нечёткой логики позволяет с успехом решать задачи, исходные данные в которых являются слабо формализованными [3]. Кроме того, использование правил, составление которых проводится на языке, близком к естественному, существенно повышает степень приближения к требуемым результатам вывода.

В общем случае систему нечёткого вывода можно представить следующим образом (рис. 1). Основными её компонентами являются: *фаззификатор*, *машина нечёткого логического вывода* и *дефаззификатор*.

На вход фаззификатора поступает вектор X чётких переменных. Фаззификатор, используя данные о функциях принадлежности термов, преобразовывает его в вектор нечётких множеств \tilde{X} , соответствующий входному вектору. Машина нечёткого логического вывода по определённому алгоритму преобразует входной вектор нечётких множеств \tilde{X} в выходной вектор нечётких множеств \tilde{Y} , используя правила нечёткой базы знаний. Дефаззификатор, в итоге, преобразует выходной вектор \tilde{Y} в вектор чётких переменных Y , так же используя данные о функциях принадлежности термов.

Существуют различные системы представления нечётких знаний и основанные на них механизмы нечёткого вывода. Наиболее распространёнными являются системы *Мамдани* и *Сугэно*.

В системах типа Мамдани база знаний состоит из правил вида:

если $x_1 = 87: 89$ и x_2 средний, то y высокий.

В системах типа Сугэно база знаний состоит из правил вида:

если x_1 низкий и x_2 средний, то $y = a_0 + a_1x_1 + a_2x_2$.

Таким образом, основное отличие между системами Мамдани и Сугэно заключается в разных способах задания значений выходной переменной в правилах, образующих базу знаний. В системах типа Мамдани значения выходной переменной задаются нечёткими термами, в системах типа Сугэно — как линейная комбинация входных переменных.

Поскольку одной из основных причин использования нечёткой логики для механизма оценки является возможность задания требуемой зависимости на языке, близком к естественному, в данном проекте будет использоваться система логического вывода типа Мамдани.

МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В системах типа Мамдани нечёткие знания хранятся с использованием предикатных правил, которые определяют значения выходной переменной с использованием нечётких термов [1].

Предикатные правила состоят из множества *антецедентов* (условий правила) и *консеквентов* (следствий правила).

Будем использовать следующие разновидности правил:

1) **если x_1 есть [не] X_{1i} и x_2 есть [не] X_{2j} и ...,**

то y_1 есть [не] Y_{1m} , y_2 есть [не] Y_{2n} , ...;

2) **если x_1 есть [не] X_{1i} или x_2 есть [не] X_{2j} и ...,**

то y_1 есть [не] Y_{1m} , y_2 есть [не] Y_{2n} , ...

где:
 x_1, x_2, \dots — *входные переменные*,
 $X_1 = \{X_{11}, X_{12}, \dots\}$,

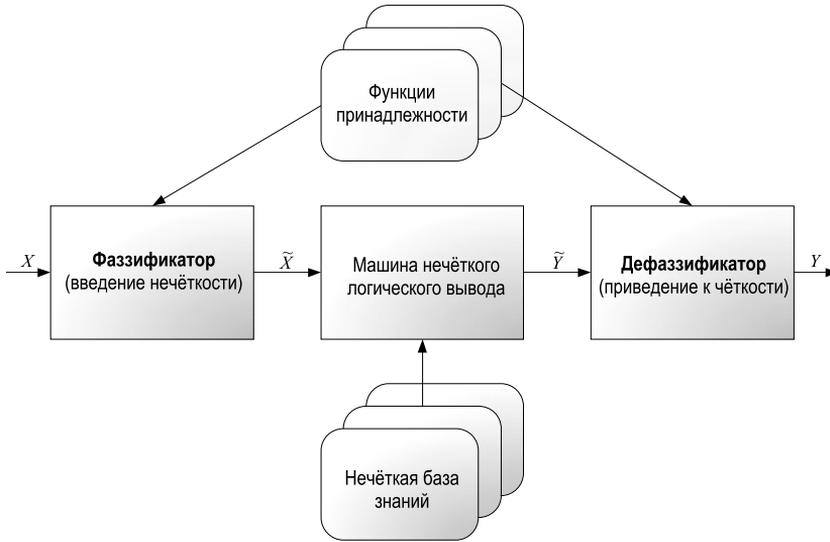


Рис. 1. Нечёткий логический вывод

$X_2 = \{X_{21}, X_{22}, \dots\}, \dots$ — соответствующие им множества термов;

y_1, y_2, \dots — выходные переменные,

$Y_1 = \{Y_{11}, Y_{12}, \dots\}, Y_2 = \{Y_{21}, Y_{22}, \dots\}, \dots$ — соответствующие им множества термов.

Таким образом, antecedentes правила представляют собой возможные условия реализации, представленные некоторыми термами входных лингвистических переменных.

Консеквенты правила представляют собой выводы, описанные некоторыми термами выходных лингвистических переменных.

Применение двух типов правил с различными функциями связывания antecedентов позволяет в общем случае сократить число правил базы знаний и сделать механизм их задания более гибким, облегчая формулирование требуемых зависимостей с использованием понятий естественного языка.

Также будем использовать возможность применения для каждого терма

в правиле квантификатора *не*. Использование квантификаторов позволяет избежать создания новых термов для описания очевидных зависимостей на основе существующих термов лингвистической переменной.

Также введём для каждого правила некую количественную характеристику $w \in [0, 1]$, которая будет представлять собой *вес* правила, определяющий степень участия этого правила в механизме вывода.

СТРУКТУРА ПРОЕКТА ОБУЧЕНИЯ

При выборе формата для хранения учебных курсов будем придерживаться международного стандарта IMS Content Packaging [8]. Данная спецификация описывает способы описания и упаковки учебных материалов, например отдельных курсов или их коллекции, в интероперабельные легко распределяемые «блоки» за счёт описания состава, структуры и расположения предоставляемых учебных материалов с указанием

типа содержимого. Стандарт рассчитан на разработчиков учебных материалов, продавцов обучающих систем, компьютерных платформ и провайдеров обучающих сервисов. Обучающие материалы описываются и упаковываются с использованием XML-формата, что позволяет добиться интероперабельности с любыми приложениями и инструментами, поддерживающими эту спецификацию.

В разрабатываемом проекте будем использовать понятие *учебный курс*, в качестве логической единицы представления учебного материала. В соответствии со спецификацией IMS Content Packaging учебный курс представляется в виде некоего независимого «пакета» (рис. 2), внутри которого содержатся непосредственно содержимое (ресурсы) и манифест (описание). В качестве ресурсов в «пакете» могут содержаться самые разнообразные виды файлов — начиная от аудио- и видеоформатов и заканчивая HTML-страницами или простыми текстовыми файлами. Внутреннее содержание курса будет рассмотрено позднее, данная же спецификация описывает только способ описания и упаковки обучающих материалов.

Манифест представляет собой описание структуры и содержимого курса в виде XML-файла, имеющего фиксированное имя *imsmanifest.xml* и определённый формат [8]. Манифест описывает:

- метаданные;
- информацию об организации учебных материалов;
- описание ресурсов и контрольных заданий;
- вложенные манифесты.

Метаданные учебного курса описываются в соответствии со спецификацией IEEE1484.12.1–2002 Standard for Learning Object Metadata [9]. Метадан-

ные описывают общие свойства и характеристики курса. Часто, при сравнительно простом использовании этой спецификации, в метаданные включаются описание курса, ключевые слова, направление обучения, уровень сложности и некую информацию об авторе. Более сложные реализации позволяют иерархически описывать целые группы свойств и характеристик. В этом случае спецификация определяет фиксированную иерархию характеристик, описываемых метаданными.

Информация об организации учебных материалов позволяет на основе одних и тех же ресурсов создать различные способы проведения учебного курса. Такая возможность достигается за счёт отдельного хранения ресурсов и информации об их использовании. Имея неупорядоченный набор ресурсов, описывающих какую-либо предметную область, можно создать несколько вариантов её изучения, например: краткую программу обучения, полную программу обучения и расширенную программу обучения. Краткая программа будет ссылаться только на ресурсы, содержащие основные понятия, полная может содержать основную часть ресурсов, а расширенная программа обучения будет дополнительно содержать в себе ссылки на ресурсы, которые наиболее полно описывают предметную область.

Секция ресурсов предназначена для детального описания включённых в состав курса ресурсов, каждый из которых представляет собой один или несколько файлов с обучающими материалами. При описании ресурса в манифесте указывается его тип, необходимый для правильного отображения содержимого, а также ссылка на основной файл,

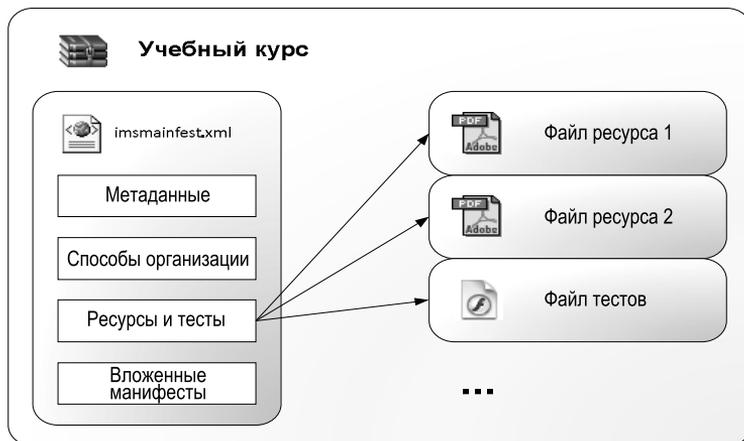


Рис. 2. Описание и упаковка учебных материалов

если ресурс содержит сразу несколько файлов. Например, если ресурс выполнен в виде HTML-страницы, то при его описании в манифесте будут перечислены все используемые файлы (сама HTML-страница, изображения, скрипты и т.д.), а также указана ссылка на файл HTML-страницы, поскольку именно его следует отображать при просмотре содержимого курса.

Вложенные манифесты позволяют описывать в рамках одного курса целую иерархическую структуру обучающих понятий. При описании более мелких обучающих понятий файлы их ресурсов помещаются в общий пакет наряду с основными, а их манифесты помещаются в основной манифест в качестве вложенных.

Таким образом, формируется определённый набор файлов, входящих в пакет: все файлы содержимого и файл манифеста. Файлы, представляющие содержимое, могут находиться в любых папках различного уровня вложенности, а файл манифеста *imsmanifest.xml* должен располагаться непосредственно в корневом каталоге курса.

Для удобства распространения и использования учебного курса спецификация IMS Content Packaging предлагает упаковывать все получившиеся файлы в единый PIF-файл (Package Interchange File). В качестве основного формата для PIF-файла рекомендуется использовать формат архива PKZip (*zip*).

В итоге все содержимое учебного курса представлено в виде одного файла, а использование алгоритмов сжатия позволяет экономить дисковое пространство, необходимое для его хранения. Для учебных материалов была выбрана структура, предлагаемая спецификацией IMS Content Packaging, содержимое учебных курсов в которой реализовано по концепции внешнего представления.

ПРИМЕНЕНИЕ АЛГОРИТМА НЕЧЁТКОГО ВЫВОДА ПРИ ТЕСТИРОВАНИИ

Механизм вывода позволяет системе получать новые знания на основе уже имеющихся. При использовании описанной выше предикатной модели представления знаний процесс логического вывода будет сводиться к опреде-

лению фактических значений выходных переменных на основании информации о фактических значениях входных переменных.

В общем случае любой логический вывод осуществляется за четыре этапа.

1) *Введение нечёткости*. Применение функций принадлежности к фактическим значениям входных переменных для определения степени истинности каждой предпосылки каждого правила.

2) *Логический вывод*. На основе вычисленных значений истинностей для предпосылок определяется степень истинности всего правила, которая применяется к переменным следствия. В результате этого для переменных следствия в каждом правиле будет сформировано своё нечёткое множество.

3) *Композиция*. Все нечёткие множества, сформированные для каждой переменной следствия во всех правилах, объединяются вместе, чтобы сформировать одно нечёткое множество для каждой выходной переменной.

4) *Приведение к чёткости*. Преобразование нечёткого множества каждой

выходной переменной к её фактическому значению.

Рассмотрим особенности работы алгоритма Мамдани на следующем примере (рис. 3).

Пусть база нечётких знаний определяется совокупностью двух правил:

если x есть A_1 и y есть B_1 , то z есть C_1 ;

если x есть A_2 или y есть B_2 , то z есть C_2 ,

где x и y — входные переменные, z — выходная переменная.

Для входных переменных заданы фактические значения x_0 и y_0 соответственно. Требуется определить фактическое значение z_0 лингвистической переменной z .

1. *Введение нечёткости*. Определяем степень истинности каждого условия каждого правила. Для этого к фактическому значению входной переменной применяется функция принадлежности соответствующего термина.

2. *Логический вывод*. Для определения степени истинности всего правила необходимо применить операцию, со-

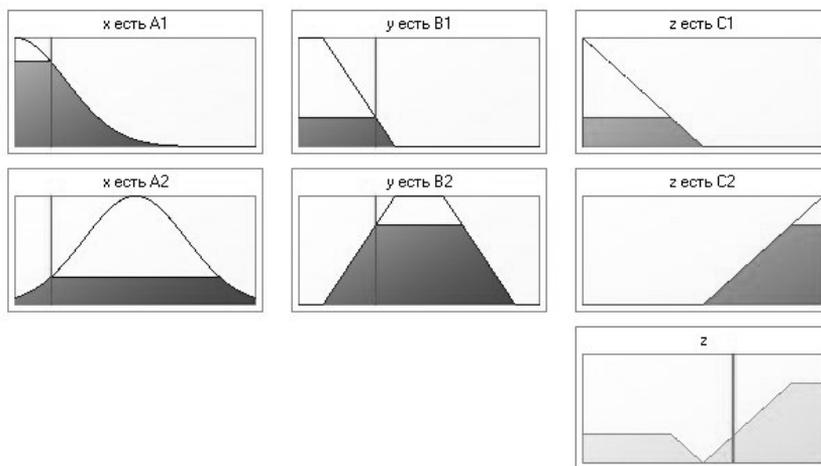


Рис. 3. Особенности работы алгоритма Мамдани



ответствующую логической операции, связывающей его антецеденты.

При использовании конъюнктивной формы записи правила (применении связки *и*), в качестве операции определения истинности, чаще всего используется функция минимума *min*. В случае же дизъюнктивной формы записи правила (применении связки *или*) — функция максимума *max*.

Полученная степень истинности используется для «отсечения» по высоте функций принадлежности всех термов следствия данного правила. Таким образом, для каждого термина следствия формируется своё нечёткое множество.

3. *Композиция*. Все нечёткие множества, назначенные разным терминам одной и той же переменной, объединяются с использованием операции композиции. Чаще всего в роли такой операции выступает функция максимума *max*. При этом результирующее множество конструируется как поточечный максимум по всем исходным множествам. В результате для каждой переменной следствия формируется отдельное нечёткое множество.

4. *Приведение к чёткости*. Для определения фактических значений каждой из переменных вывода в алгоритме Мамдани чаще всего используется *метод центра тяжести* — чёткое значение ω_0 выходной переменной ω определяется как центр тяжести для кривой $\mu(\omega)$.

МЕТОДИКИ ОЦЕНКИ УЧАЩИХСЯ

Для оценки обучающихся прежде всего необходимо выделить некие значащие факторы, которые и будут её определять. В ходе проведения дистанционного обучения есть возможность автоматического получения некоторых количествен-

ных характеристик, тем или иным образом характеризующих степень изучения материала. Например, в качестве таких характеристик могут выступать оценки *количества успешно пройденных тестов определённого курса*, а также *времени его изучения*. Будем называть такие оценки *фиксированными*.

Однако, во-первых, этих характеристик явно недостаточно для проведения адекватной комплексной оценки, а, во-вторых, само ограничение факторов оценки приводит к значительной потере гибкости и не даёт гарантии получения нужного результата в широком спектре областей применения дистанционного обучения. Под методикой оценки будем понимать совокупность правил, определяющих алгоритм вывода общей оценки на основе имеющихся фиксированных характеристик и дополнительных характеристик, описанных в рамках данной методики (рис. 4).

Рассмотрим более подробно понятие методики и процесс её использования для получения комплексной оценки.

При добавлении новой методики задаётся её собственный состав *категорий оценок*. После этого обучающийся по данной методике может получать численные оценки в рамках заданных категорий. В итоге по каждой категории становится возможным получить *общую оценку*, которая берётся как среднее:

$$S_i = \frac{\sum_{j=1}^{n_i} s_{ij}}{n_i}, \quad (1)$$

где

S_i — общая оценка по i -ой категории,

s_{ij} — j -ая оценка по i -ой категории,

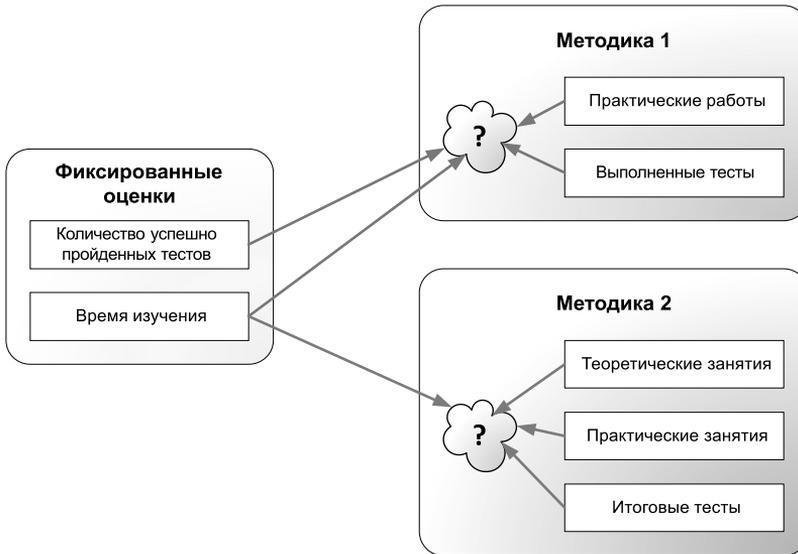


Рис. 4. Применение различных методик оценки обучающихся

n_i — количество оценок в i -й категории.

Наряду с общими оценками для каждого обучающегося автоматически формируются оценки из числа фиксированных.

Также в рамках методики оценки задаётся свой собственный нечёткий контроллер для логического вывода, который позволяет получить численное значение выходного параметра на основе значений входных параметров. В качестве входных параметров такой контроллер может использовать как общие оценки по категориям, так и любые (не обязательно все) имеющиеся оценки из набора фиксированных.

Учитывая, что в рамках данного проекта предусмотрена возможность создания и редактирования произвольных нечётких контроллеров, достигается необходимая гибкость для получения требуемой комплексной оценки.

РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ И ЕГО ПРИМЕНЕНИЕ

Как уже упоминалось выше, сайт дистанционного обучения представляет собой конечный инструмент для проведения учебного процесса. Поддерживая различные роли, он предоставляет необходимые функции всем членам процесса дистанционного обучения и позволяет проводить требуемую комплексную оценку. Таким образом, основными функциональными требованиями к сайту дистанционного обучения являются:

- организация процесса дистанционного обучения, хранение информации об участниках образовательного процесса;
- разграничение доступа и поддержка ролей для различных участников образовательного процесса;
- поддержка разработанного формата учебных курсов и возможность его отображения;
- поддержка файла описания системы нечёткого логического вывода и возмож-

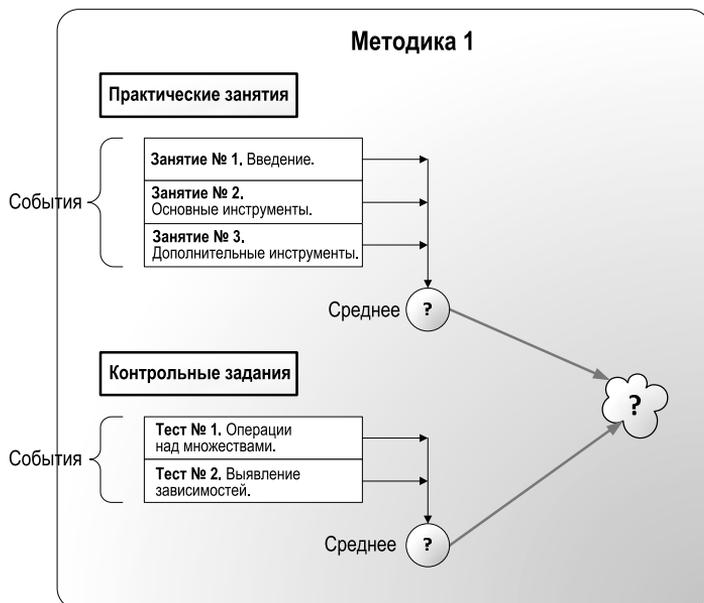


Рис. 5. Схема проведения комплексной оценки

ность проведения комплексной оценки на основе статистических данных.

ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Участники образовательного процесса представлены на сайте с помощью следующих ролей:

- студенты;
- преподаватели;
- администраторы.

Студенты — прямые потребители образовательного материала, являющиеся объектами оценки. Студенты могут заходить на сайт, изучать назначенные учебные курсы и просматривать текущие оценки своей деятельности. Управление осуществляется на уровне учебных групп, в которые объединяются студенты.

Каждому учебному курсу соответствует некая методика оценки. В рамках методики определяется состав ка-

тегорий, общие оценки, которые будут влиять на общую комплексную оценку по данной методике. Для этого в рамках каждого курса определяются набор событий по каждой категории, оценки по которым и определяют общую оценку по категории (рис. 5).

Группе студентов назначается определённый набор учебных курсов, каждый из которых контролируется каким-либо преподавателем. *Преподаватель* может заходить на сайт, просматривать статистику по учебным группам и студентам, обучение которых он контролирует, а также выставлять оценки в рамках событий по каждому учебному курсу.

Администраторы предназначены для заполнения практически всех структур данных сайта. Они редактируют составы студентов и учебных групп, определяют преподавателей, управляют размещением учебных курсов. Также администраторы выполняют настройку методик

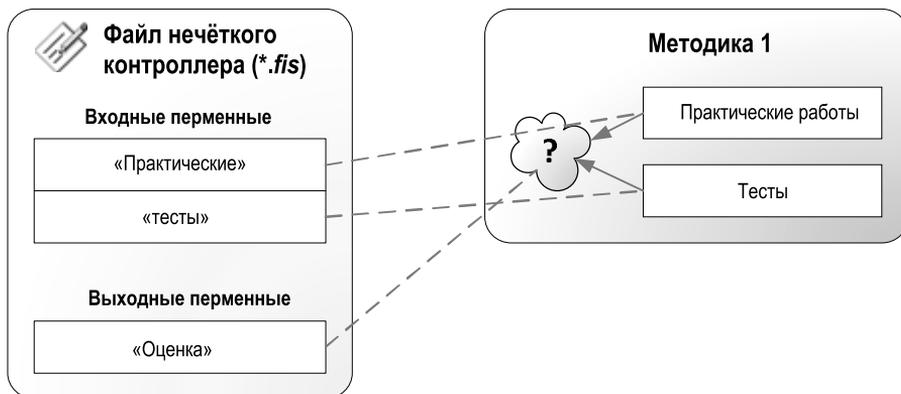


Рис. 6. Отображение переменных FIS-файла на категории в рамках методики оценки

оценки и выполняют подключение систем нечёткого логического вывода.

ПРИМЕНЕНИЕ НЕЧЁТКОГО ВЫВОДА

Как уже упоминалось выше, каждая методика оценки работает в соответствии с некоторым механизмом логического вывода. Для этого на сайт дистанционного обучения подключается нечёткий контроллер, разработанный в редакторе нечёткого вывода. Поскольку редактор нечёткого вывода позволяет создать произвольный нечёткий контроллер, а в рамках методики требуется выполнить оценку по заданным категориям, выполняется так называемое ото-

бражение переменных FIS-файла на категории (рис. 6).

Способ отображения задаётся при настройке методики оценки. Следует заметить, что отображение выполняется не только для заданных в рамках методики категорий, но и для так называемых фиксированных оценок, если их необходимо учитывать для комплексной оценки.

Таким образом, применение механизма отображения обеспечивает требуемую гибкость в проведении итоговой оценки, поскольку позволяет использовать для этих целей произвольный нечёткий контроллер.

ЛИТЕРАТУРА

1. Штовба С.Д. Введение в теорию нечётких множеств и нечёткую логику / С.Д. Штовба — М.: Мир, 2001. — 178 с.
2. Заде Л. Понятие лингвистической переменной и её применение к принятию приближённых решений / Л. Заде — М.: Мир, 1976. — 167 с.
3. Прикладные нечёткие системы: перевод с япон. / К. Асаи, Д. Ватада, С. Иваи и др.; под ред. Т. Тэрано, К. Асаи, М. Сугено. — М.: Мир, 1993.
4. Mamdani E.H. Applications of fuzzy algorithms for simple dynamic plant. Proc. IEE. 1974. — Vol. 121, № 12. — P. 1585–1588.
5. Норенков И.П. Концепция модульного учебника / И.П. Норенков // Информационные технологии. — 1996. — № 2.
6. IMS Content Packaging 1.1.4 Version 1.1.4 Final Specification (http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_bestv1p1p4.html).