

ТРЕБОВАНИЯ К АРХИТЕКТУРЕ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ, ОБЕСПЕЧИВАЮЩЕЙ ВАРИАТИВНОСТЬ ТРАЕКТОРИЙ САМООБУЧЕНИЯ

Ваграменко Ярослав Андреевич,

*доктор технических наук, профессор, заместитель директора
по информационным образовательным ресурсам*

Федерального государственного научного учреждения

*«Институт информатизации образования» Российской академии образования,
ininformao@gmail.com*

Яламов Георгий Юрьевич,

*кандидат физико-математических наук, доцент, ведущий научный сотрудник
Федерального государственного научного учреждения*

*«Институт информатизации образования» Российской академии образования,
aio@tgori.ru*

Фанышев Роман Геннадьевич,

аспирант Федерального государственного научного учреждения

«Институт информатизации образования» Российской академии образования

Аннотация

В статье описывается интеллектуальная информационная система, обеспечивающая вариативность траекторий самообучения, которая рассматривается как специальный тип экспертной системы.

Ключевые слова:

интеллектуальная экспертная система; интеллектуальная информационная система; самостоятельное обучение; база знаний; база данных; алгоритм.

Доминирующей тенденцией современного образовательного процесса является повышение роли самостоятельного обучения для осуществления заочного образования в котором, главным образом, внимание уделяется современным прогрессирующим информационным технологиям. Поэтому существует необходимость в синтезе действующих в данный момент образовательных технологий для создания систем, способных давать экспертную оценку самостоятельной образовательной деятельности конкретного индивидуума, в том числе, поддерживать процесс выбора оптимальной образовательной траектории. Необходима экспертная система, интегрированная с внешними источниками образовательной информации.

В самостоятельной работе можно выделить компоненты, характерные для деятельности как таковой: мотивационные звенья, постановку конкретной задачи, выбор способов выполнения, исполнительское звено, контроль. В связи с этим можно выделить условия, обеспечивающие успешное выполнение самостоятельной работы:

- мотивированность учебного задания (для чего, чему способствует);
- четкая постановка познавательных задач;
- алгоритм, метод выполнения работы, знание студентом способов ее выполнения;
- четкое определение преподавателем форм отчетности, объема работы,

сроков ее представления;

- определение видов консультационной помощи (консультации – установочные, тематические, проблемные);
- критерии оценки, отчетности и т.д.;
- виды и формы контроля (практикум, контрольные работы, тесты т.п.).

Для более детального понимания проблемы построения подобной системы необходимо использовать классические методы и методологии разработки первоначальных информационно-программных решений. В данной статье рассмотрены требования к архитектуре такой интеллектуальной информационной системы, имеющей во многом характер экспертной системы, включая методы формирования ее архитектуры. Требования относятся к следующим типам самостоятельной работы студентов:

- выполнение системы заданий и указаний, предусматривающих самостоятельные исследования в рамках изучаемого курса;
- выбор темы рефератов и докладов, логично дополняющих и расширяющих область компетенции студента в рамках учебного курса;
- использование соответствующих своей, так называемой «модели обучаемого» (модель обучаемого — это абстрактное представление студента в виде совокупности сетевой, векторной, имитационной и фиксирующей моделей формирования информационного объекта), инструкции и методические указания к выполнению лабораторных работ, тренировочных упражнений, домашних заданий и т.д.;
- написание курсовых и дипломных проектов (данный тип самостоятельной работы способствует достижению основной цели образовательного процесса – получению навыков решения специализированного круга задач);
- организация работы со специальной, обязательной и дополнительной литературой;
- самооценивание и самотестирование знаний для отслеживания текущего уровня собственного соответствия образовательным нормам и стандартам.

Образовательная деятельность по формированию перечисленных выше типов самостоятельной работы студентов на сегодняшний день осуществляется инструментально-программными и прикладными программными средствами типа Microsoft Word, Macromedia Dreamweaver и т.д. Однако единого подхода к управлению информационным содержанием и определению полезности получаемых в итоге ресурсов нет.

Рассмотрим некоторые составные компоненты возможной архитектуры экспертной системы информационной поддержки самостоятельной работы студентов:

- модель обучаемого;
- модель обучения (совокупность основных спецификаций электронного образовательного процесса);
- модель объяснения (экспертной поддержки).

Простейшим вариантом модели обучаемого является векторная модель, которая каждому изучаемому понятию или умению ставит в соответствие некоторый элемент, принимающий значение «знает/не знает», в результате

уровень знаний студента (уровень его компетентности) в изучаемому курсу определяется векторным набором значений элементов. Преимуществом векторного подхода является простота использования и реализации, а недостатком является то, что в случае ее использования недостаточно формализован уровень связности между простейшими, так называемыми, образовательными единицами (тема, вопрос, проблема, задача, понятие, списочные структуры информационных ресурсов, оказывающих конечный эффект на познавательный процесс студента).

Более универсальным подходом является использование сетевой модели, представляющей собой многослойный математический граф, в узлах которого содержатся образовательные единицы, а дуги соединяют их логично между собой. Каждому узлу и дуге сопоставляется некоторая величина или набор величин, характеризующие степень владения обучаемым данным понятием или умением. Причем также допускается наследование величин, что формирует так называемый личный опыт работы студента с имеющимися образовательными единицами.

Таким образом, модель обучаемого, в простейшем случае, включает следующие компоненты:

- первичная учетная информация об обучаемом (может быть получена из существующей в вузе информационной системы) – Ф.И.О., номер группы, дата поступления, курс и т.д.;

- вторичная информация о личности обучаемого (формируется последовательно в процессе работы экспертной системы) – начальный уровень знаний, заключительный уровень знаний, алгоритмы и траектории обучения и выявления уровней знаний обучаемого, и т.д.

Модель обучаемого определяет архитектуру экспертной системы (рис. 1) [7].

Для реализации моделей, в рамках экспертной системы информационной поддержки может быть использован метод тестирования и процедуры ввода тестирующих вопросов в базу знаний, процедура формирования «идеальной» оценки, процедура подсчета неверных ответов и формирования итоговой, реальной оценки уровней знания студента. Ниже представлено поэтапное описание построения и функционирования экспертной системы информационной поддержки самостоятельной работы студентов.

1. Предварительный этап (разработка технического задания на проект, построение базы знаний, построение модели диалога, конфигурирование и др.).

2. Этап детального проектирования компонентов экспертной системы в (режим DesignTime для преподавателей-предметников).

- 2.1. Построение компонентов эталонной модели курса/дисциплины (выделение элементов курса/дисциплины, подготовка контрольных вопросов с коэффициентами сложности и т.д.).

- 2.2. Построение компонентов модели обучаемого (выбор алгоритма оценивания уровня знаний, компоновка набора тестов для выявления личностных характеристик и т.д.).

- 2.3. Построение компонентов модели обучения (конкретизация и построение обучающих воздействий).

3. Этап функционирования разработанной экспертной системы (режим RunTime для обучаемых).

- 3.1. Формирование моделей обучаемых (построение психологического портрета личности, выявление уровня знаний и умений путем проведения контрольных тестирований и т.д.).

3.2. Построение индивидуальных планов (стратегий) обучения для обучаемых.

3.3. Реализация текущего плана (совокупности обучающих воздействий) для конкретного обучаемого с последующим контролем знаний и умений [4].

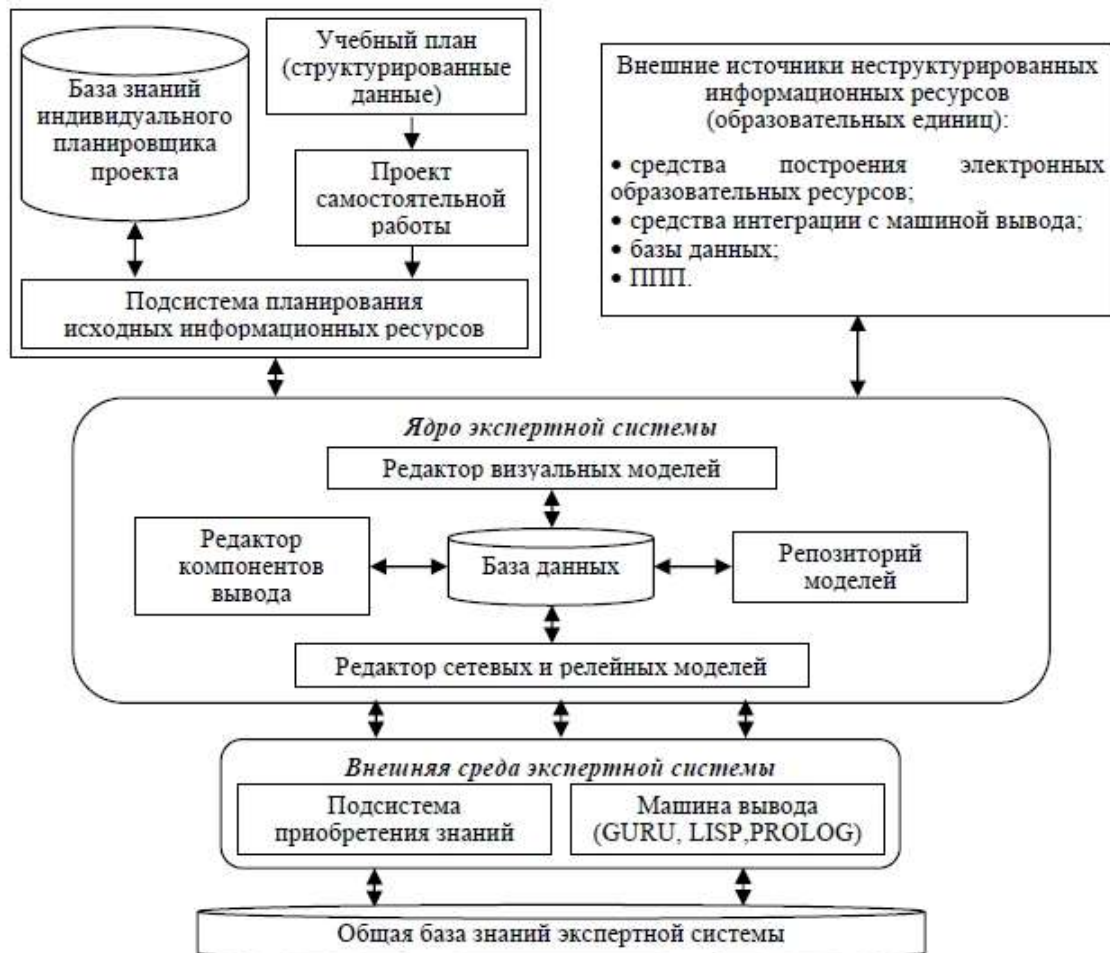


Рис. 1. Архитектура экспертной системы

Экспертная система для самообучения, как интеллектуальная информационная система, должна обеспечить реализацию следующих алгоритмов:

- 1) алгоритм подбора подходящего учебного плана в зависимости от результатов начального тестирования обучаемого (кроме его уровня знаний, могут быть выявлены и некоторые индивидуальные особенности);
- 2) алгоритм предъявления обучаемому учебного материала и алгоритм накопления результатов освоения;
- 3) алгоритм составления протокола обучения, хранящего в сжатой форме историю всех событий, и алгоритм составления на основании протокола обучения моделей обучаемого (определение типа обучаемого) и учебной ситуации;
- 4) алгоритм анализа ситуации – определение действий, которые целесообразно предпринять в данной учебной ситуации для данного типа обучаемого;
- 5) алгоритм подбора подходящей учебной стратегии;

б) алгоритм составления плана следующего этапа самообучения – наполнение выбранной стратегии учебным материалом.

В ходе обучения экспертная система должна протоколировать следующие события: успешно выполненные упражнения (фрагменты образовательного контента):

- 1) упражнения и отдельные части формализованной структуры, допущенные при выполнении упражнений ошибки с квалификацией класса ошибки;
- 2) обращения к справочной или иной дополнительной информации (храниться в рабочей области экспертной системы в виде репозитория фрагментов образовательного контента);
- 3) существенные превышения запланированного автором времени упражнения или, наоборот, выполнение упражнения значительно ранее запланированного времени;
- 4) предпринятые по инициативе обучаемого отклонения от учебной стратегии и другие вмешательства в ход обучения.

История событий накапливается во всех режимах самообучения, поэтому при переходе от режима свободной навигации к режиму обучения с экспертной системой учитываются предыдущие результаты (алгоритм верификации промежуточных результатов самообучения пользователя).

Алгоритмы обеспечения вариативности траекторий обучения и логический вывод экспертной системы

Для удобства задания автором алгоритмов обучения целесообразно выбрать продукционный тип базы знаний экспертной системы, в котором применяется кодирование динамики события E на протяжении всего обучения тройкой:

$$CE = (FE, NE, RE), \quad (1)$$

где: FE – тип кривой – описание динамики частоты события на интервале элементами конечного алфавита: часто, редко, сначала редко, потом чаще и т.д.; NE – длина кривой – отношение длины протокола к запланированному времени обучения TO ; RE – вес кривой – отношение числа произошедших событий к числу возможных или, как в случае пользования справочной информацией, к числу ожидаемых (в этом случае оно может быть больше 1).

Более подробный протокол обучения может быть получен, если хранить тройки CE_i (I_j) для нескольких наиболее важных интервалов I_j : учебная цель, урок, а также временных интервалов (сегодняшнее занятие, последние части т.п.) [2].

Ниже опишем некоторую формализацию процесса самообучения, лежащую в основе работы системы логических выводов экспертной системы, которая затем будет уточнена в виде автоматной схемы. Структурная схема модуля экспертной системы самообучения имеет вид, показанный на рисунке 2.

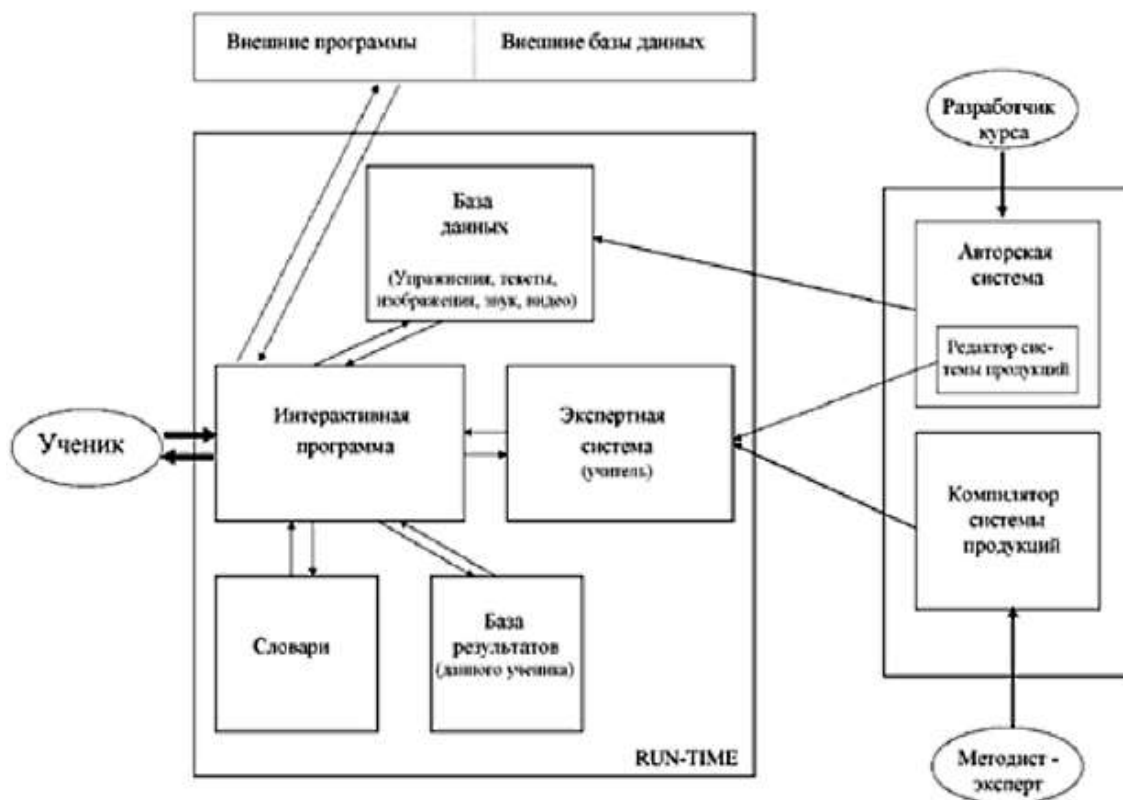


Рис. 2. Структурная схема модуля экспертной системы

Для реализации алгоритмов логического вывода фрагментов образовательного контента необходимо представить его в виде набора деревьев, имеющих перекрестные ссылки, что отражало бы не только иерархичность структуры обучающего материала, но и различного рода ссылки, создающие первичные, вторичные и другие структуры учебного материала, отражающие взаимосвязи различных учебных целей, задач, компетенций и управляющих воздействий.

В зависимости от типа модели обучаемого и его индивидуальных подходов к обучению (в общем виде подходы могут быть индуктивный, дедуктивный и гибридным) предлагается использовать три вектора обучения (быстрый, нормальный и медленный). На рисунке 3 изображены линейные стратегии обучения, соответствующие процессам освоения образовательной единицы в соответствии с эталонной моделью учебной программы дисциплины.

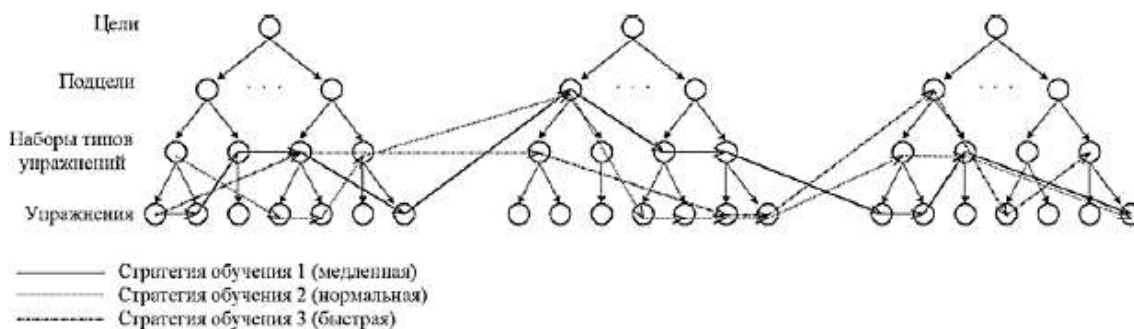


Рис. 3. Линейные стратегии обучения

В процессе формирования алгоритмов логического вывода экспертной

системы необходимо реализовывать возможности выбора стратегий обучения. В качестве анализа способностей к той или иной стратегии, система должна предложить вариант повторного упражнения того же типа. В случае допущения обучаемым ошибок локального характера, необходимо вернуть пользователя к ранее пройденному материалу. Используя подготовленную преподавателем (инженером базы знаний экспертной системы) систему оценочных шкал уровня освоения дисциплины, поэтапно проконсультировать пользователя по способам их ликвидации. В случае большого количества разнородных ошибок или изменения качества ошибок, система должна отследить траекторию прохождения узлов графа обучаемым, реализующего представление образовательного контента.

Управление логическим выводом посредством компонента «РЕШАТЕЛЬ»

Для реализации интерактивного обучения студента в режиме он-лайн необходимо выбрать соответствующий алгоритм и программную реализацию формирования правил логического вывода, хранящегося в рабочей области образовательного контента в момент консультации с экспертной системой. Правила экспертной системы — это продукции вида: «Если условие, то действие».

Для экспертной системы необходим специальный механизм логического вывода фактов и фрагментов образовательного контента (например, локальная верификация и оценка качества учебного процесса на основе сравнения выбранной стратегии прохождения учебного материала и эталонной). Реализовать алгоритмы логического вывода и обучения можно в виде достаточно большого набора кривых, допускающих естественную интерпретацию типа: «прогресс», «единичная ошибка».

Механизм логического вывода необходимо реализовать при помощи отдельного модуля «РЕШАТЕЛЯ» экспертной системы, который должен поддерживать следующие возможности:

- поддержка прямой и обратной (дедуктивной, индуктивной) стратегии вывода на основе «просмотра» узлов графа учебной дисциплины;
- поиск решения «В глубину» (просмотр фрагментов базы знаний формальных XML-структур электронно-образовательных ресурсов);
- поиск и разрешение конфликтов правил на основе интервьюирования преподавателя и верификации имеющихся правил;
- сохранение состояний и типов фрагментов и типов метазнаний (на основе модели Dublin Core) и правил (активное, неактивное) в рабочей памяти экспертной системы.

Процедуры «РЕШАТЕЛЯ» экспертной системы можно представить с помощью системы общепринятых процессов в виде:

$$I = \langle V, S, K, W \rangle, \quad (2)$$

где: V – процесс выбора, осуществляющий выбор из P и R подмножества активных продукций и подмножества активных данных; S – процесс сопоставления, определяющий множество пар: правило p_i данные $\{d_j\}$, где $p_i \in P_v$, $\{d_j\} \subset R_v$, причем каждое p_i применимо к элементам множества $\{d_j\}$; K — процесс разрешения конфликтов (или процесс планирования), определяющий, какой из идентификаторов будет выполняться; W — процесс, осуществляющий выполнение выбранного идентифицированного правила (то есть выполнение действий, указанных в правой части правила). Результатом

выполнения является модификация данных в R или операция ввода/вывода. Механизм вывода должен реализовываться не основе семантических и синтаксических методов выборки фрагментов данных из базы знаний. Данный подход позволяет без труда интегрировать подобные правила (метаправила) непосредственно в «РЕШАТЕЛЬ» так как они совершенно не зависят от рассматриваемой предметной области и способствуют декомпозиции сложных фрагментов метаинформации на подзадачи и методы их использования для генерирования исходного образовательного контента.

Ниже рассмотрим алгоритм реализации начальной выборки. Данный подход производится на основе имеющегося списка целей, что позволяет сократить предметную область выборки и логического вывода предварительной информации и способствует более детальному пониманию рекомендаций экспертной системы пользователю. В случае, если набор целей существует – используется режим обратного вывода (от более детальных целей), в случае, если список целей отсутствует – используется метод прямого ввода (например, текста запроса).

В процессе пробного тестирования данных алгоритмов было выявлено, что число конфликтов прямо пропорционально степени неточности искомой информации. Для разрешения данных противоречий и конфликтов правил в базе знаний экспертной системы используется нотация метаописания правил (метаправила). Ниже приведем их формальное описание:

$$\begin{aligned}
 & \text{если } P(r_1) > P(r_2) \text{ то } K(r_1) > K(r_2) \text{ иначе} \\
 & \text{если } N(r_1) < N(r_2) \text{ то } K(r_1) > K(r_2) \text{ иначе} \\
 & \text{если } D(r_1) > D(r_2) \text{ то } K(r_1) > K(r_2) \text{ иначе } K(r_1) \leq K(r_2) \\
 & r_1, r_2 \in P, N,
 \end{aligned} \tag{3}$$

где: N – количество неидентифицированных атрибутов в правилах;
 D – количество неидентифицированных атрибутов в действии правил;
 R – набор всех активных правил.

Таким образом, для начальной выборки необходимо выполнить расчет приоритетов правил, который выше для правил, имеющих меньшее число не идентифицированных атрибутов, большее число не идентифицированных атрибутов в действии и высокую вероятность появления. Формирование подобной выборки зависит от того, какая стратегия вывода используется в данном рабочем цикле.

Естественно, для корректного логического вывода необходимо применять алгоритмы сопоставления правил начальной выборки и метаправил в рабочей области активного цикла. Данный алгоритм осуществлен в виде рекурсивной функции, описанной ниже:

$$F(R, A, O, K), \tag{4}$$

где: R – набор активных. правил; A – список текущих целевых атрибутов (параметры запроса); O – список идентифицированных атрибутов; K – список идентифицированных правил. Ниже рассмотрим алгоритм работы рекурсивной функции сопоставления правил:

1. правила начальной выборки заново сопоставляются с набором атрибутов в рабочей памяти R в порядке приоритета;
2. в процессе выполнения одной копии функции F производится ее

повторный запуск для внесения вновь внесенных изменений в базу правил.

Таким образом, алгоритм разрешения конфликтов в правилах основан на процедуре расчета приоритета, который выше для тех, которые имеют большее количество идентифицированных экспертом (преподавателем) атрибутов правил.

Ниже приведем общий вид правил адаптивного логического вывода экспертной системы:

*Если {(событие1, тип кривой 1, интервал 1) и
(событие2, тип кривой 2, интервал 2) и...}
то {действие}*

Троек (событие; тип кривой; интервал) может быть от 1 до 10.

Каждому событию (в протоколе активности пользователя) в процессе обучения ставится в соответствие кривая определенного типа, заданная на некотором интервале.

Примеры событий (либо учебно-тематические заданий):

- частота пользования словарем;
- ошибки при выполнении упражнений;
- превышение временных рамок, отведенных на выполнение учебно-тематические задания; время обдумывания (ожидания, либо задержки) ответа на вопрос.

В процессе пользования объектами учебного назначения, пользователь за счет протоколирования накапливает собственный стиль и траекторию изучения их фрагментов. К примеру, изучая раздел или тему курса лекций, пользователь периодически обращается к глоссарию понятий, фактов и терминов, в данном случае целесообразно во всем интервале времени собирать статистику пользования данным типом контента для ее дальнейшего анализа и выработки решений экспертной системы. В данном случае речь идет об аналоговых величинах и отношениях. Некоторые виды кривых пользования глоссарием учебного объекта (рис. 4) (словаря понятий – фактов дисциплины), интерпретирующих динамику событий обучаемого, которые можно использовать в качестве эталонных моделей траекторий освоения учебного материала.



Рис.4. Примеры типов кривых ожидания событий

На кривых по вертикали отложена частота использования словаря, по горизонтали время (время урока, время изучения темы и т.д.).

Главной целью разработки алгоритмов обучения для экспертной системы была задача проектирования такой системы логических выводов,

которая моделировала бы всех участников образовательного процесса (учителя, ученика, учебный материал) и организовывала оптимальное их взаимодействие.

В проекте заложены алгоритмы формирования моделей обучаемого и преподавателя, введен организованный определенным образом учебный материал (формальная XML-структура) с элементами мультимедиа. На этой основе имитируется процесс реального обучения с учетом таких характерных его особенностей, как взаимная интеграция процессов верификаций моделей обучаемого, преподавателя и учебного курса, способности ученика, оптимальность стратегии дозировки знаний и упражнений учителем, скорость запоминания и забывания знаний учеником, продолжительность и устойчивость его активного состояния и т.п.

Самым важным моментом реализации алгоритма обучения является функция объяснения экспертной системы, которая должна быть основана на интеграции отдельно взятых графов (траектории обучения) и дерева решений.

Подсистемы логического вывода, алгоритмов обучения, основанных на объяснении, строятся либо по схеме фиксации деятельности пользователя, либо по схеме трассировки событий в рабочей области экспертной системы. В обоих этих случаях все множества событий, требующих объяснения, идентифицируются заранее и каждый из них сопоставляется декларативно или процедурно с фрагментами образовательного контента. Для реализации подобного компонента в данной статье были предложены процедуры логического вывода и алгоритм консультации пользователя с экспертной системой, основанный на трассировке графа решений в соответствии с выбранной траекторией обучения. Рассмотрены алгоритмы обучения, основанные на логическом выводе, важным моментом которого является способ управления данным процессом при помощи программного компонента «РЕШАТЕЛЯ», интегрированного в базу знаний экспертной системы.

В процессе работы экспертной системы информационной поддержки самообучения обучающий (учитель) и обучаемый (ученик) занимают централизованное место в силу того, что в основе их интерактивного взаимодействия строятся алгоритмы передачи знаний и обучение первого. При рассмотрении данного процесса более детально можно сделать вывод о том, что агрегирование процессов информационного обмена может быть обеспечено в случае их формализации в виде информационных моделей, которые могут быть интерпретированы как адаптивные автоматы [3]. Со стороны автомата-учителя на каждом шаге выбирается оптимальная с его точки зрения подача автомату-ученику обучающей информации на основе того, как усвоил на предыдущих шагах обучения такую информацию автомат-ученик.

База знаний и алгоритмы логического вывода экспертной системы в этом случае будут являться достаточно универсальными инструментами для генерации фрагментов образовательного контента в заданных предметных областях, кроме того база знаний является открытой, она легко пополняется информацией во всех своих основных частях. На сегодняшний день моделирование информации о предметных областях происходит при помощи узкоспециализированных языков представления знаний, то есть обучающие системы после наполнения их конкретным содержанием становятся экспертными системами по конкретным узкоспециализированным областям.

В соответствии с вышеизложенным в проблеме синтеза адаптивного «компьютерного учителя» необходимо решить следующие задачи:

- 1) синтез автомата-учителя;
- 2) синтез автомата-ученика;
- 3) разработка информационной системы, аналогичной учебнику с упражнениями;
- 4) выработка оптимальной стратегии взаимодействия компонент (1)-(3);
- 5) создание интерфейса с широкими сервисными услугами для пользователя.

Решение этих задач сопряжено с рассмотрением целого ряда вопросов, к их числу относятся следующие:

а) разработка динамических баз данных и знаний, состоящих из больших массивов синтаксической информации со сложной семантикой и нечеткими логическими связями, эти базы должны быть компактными по объему занимаемой памяти и в то же время позволять достаточно быстро получать необходимую информацию из них;

б) разработка признакового пространства описания состояний автомата-учителя и автомата-ученика с указанием функционально-метрических зависимостей между ними, позволяющих задавать функционирование этих автоматов;

в) разработка оптимальных стратегий взаимодействия автомата-учителя с автоматом-учеником как средствами собственно теории автоматов и нечеткой логики, так и процедурами типа распознавания образов и пр.

Теоретическое и математическое обоснование возможной архитектуры экспертной системы

Теоретическим фундаментом модели может быть автоматная модель гибридного вида.

Ниже рассмотрим более детально процесс формализации компоненты обучения (консультации с экспертной системой) с точки зрения системы взаимодействующих автоматов.

Процесс обучения может быть представлен в виде двух взаимодействующих автоматных моделей: модели управляемого A_0 (ученик) и модели управляющего A (обучающая программа-учитель). Выходом D автомата A и входом автомата A_0 является учебный материал: упражнения, пояснения, справочные материалы и т.п. Выходом B автомата A_0 и входом автомата A являются события, произошедшие в процессе обучения. Задача обучающей программы (управляющего автомата A) – управлять с помощью выхода D автоматом A_0 таким образом, чтобы за минимальное время привести его в заданное состояние (достичь заданного учебного результата).

Другим вариантом постановки задачи может быть достижение максимального результата за заданное время. Формально это может быть описано как минимизация числа ошибок в выходе B автомата A_0 на некотором отрезке времени T_{fin} , завершающем процесс обучения.

Эта задача управления может быть разбита на следующие подзадачи:

- 1) Предъявление ученику учебного материала, накопление результатов урока;
- 2) Составление протокола обучения, хранящего в сжатой форме историю всех событий выхода B автомата A_0 . Составление на основании протокола обучения моделей ученика (определение типа ученика) и учебной ситуации;
- 3) Анализ ситуации: определение действий, которые целесообразно

- предпринять в данной учебной ситуации для данного типа ученика;
- 4) Подбор подходящей учебной стратегии;
 - 5) Составление плана следующего урока наполнение выбранной стратегии учебным материалом.

В соответствии с этим автомат A может быть представлен в виде суперпозиции автоматов A_1 - A_5 , как это показано на рисунке 5. Автомат A_1 представляет собой преобразователь входов B_5 и B в выходы D и B_1 соответственно. В простейшем случае это может быть автомат с одним состоянием.

Множество состояний автомата A_2 – декартово произведение множеств Q_{21} , Q_{22} и Q_{23} , описывающих соответственно протокол обучения, модель ученика и модель учебной ситуации. Для описания (потенциально неограниченной) истории событий элементами конечного множества Q_{21} целесообразно кодирование динамики события E на протяжении всего обучения тройкой $CE = (FE, NE, RE)$, где FE – тип кривой, описывающий динамику частоты события на интервале элементами конечного алфавита; NE – длина, кривой, определяющая отношение длины протокола к запланированному вещественное число из интервала $[0, T_{max}/T_0]$, может быть с некоторой точностью описана элементом конечного множества); RE – вес кривой (как и длина кривой, вес может кодироваться элементами конечного множества). Таким образом, компонента q_{21} состояния автомата A_2 есть набор (CE, \dots, CE_k) для всех протоколируемых событий E_j . Более подробный протокол обучения может быть получен, если хранить тройки $CE_j(I_j)$ для нескольких наиболее важных интервалов I_j : учебная цель, урок, а также временных интервалов (сегодняшнее занятие, последний час и т.п.). По входу $B_1(t)$ и текущему состоянию $q_{21}(t)$ автомат A_2 вычисляет свое новое состояние $q_{21}(t+1)$, если необходимо, уточняет тип ученика q_{22} и учебной ситуации q_{23} .

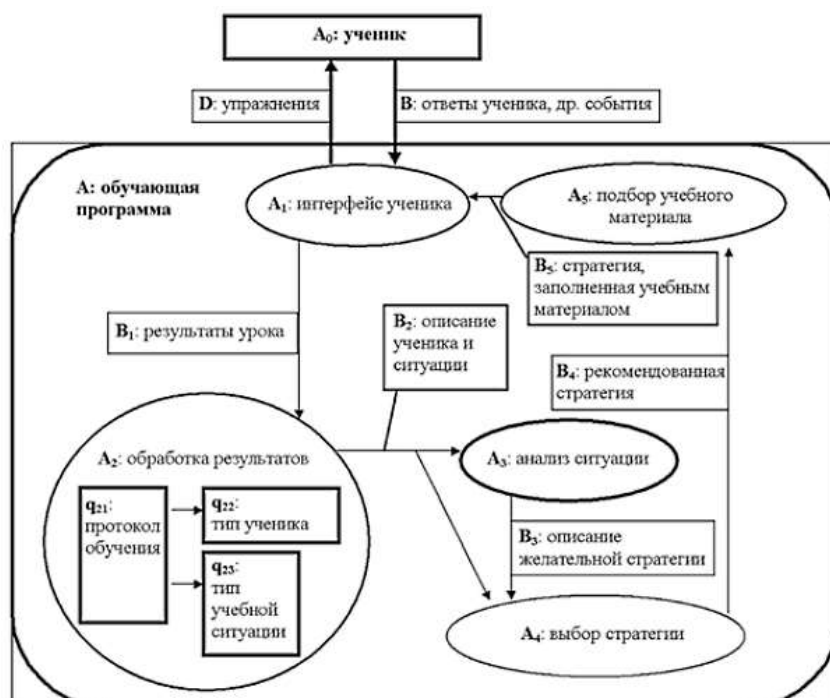


Рис 5. Общий вид взаимодействия автоматов в процессе обучения

Автомат A_3 занимает центральное место в схеме автомата A . Его задача принять решение о действиях, которые целесообразно предпринять в данной учебной ситуации (предложить дополнительные упражнения, повторение, возврат к началу темы и т. п.). Для решения этой задачи возможно применение системы продукций, в левой части которых находятся условия на значения входа B_2 , а в правой - значения выхода B_3 . Однако на практике построение такой системы продукций представляет собой существенную трудность для автора учебного курса, не обладающего, как правило, навыками такого рода. Поэтому более предпочтительным оказывается другое, менее трудоемкое для автора курса решение. Для этого принимается гипотеза о том, что автомат A_0 представляет собой вероятностный автомат из определенного класса M . Диаграмма Мура автомата A_0 приведена на рисунке ниже (рис. 6.) [3].

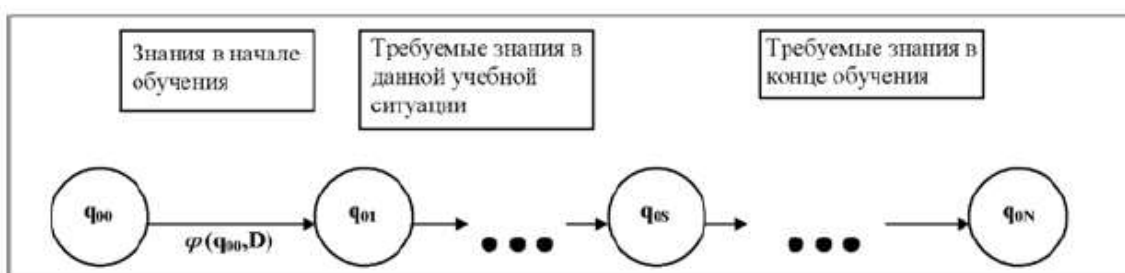


Рис. 6. Диаграмма Мура для решения задачи самообучения

В зависимости от своего состояния q_{21} автомат A_2 относит A_0 к некоторому подклассу M_i класса M ; этот подкласс M_i и представляет собой тип ученика. Характеристиками подкласса M_i являются длина цепочки состояний, функция перехода $\Psi(q, D)$, определяющая вероятность перехода в следующее состояние при значении D входа автомата A_0 , и функция выхода $\Psi(q, D, E)$, определяющая вероятность события E в состоянии q при значении входа D .

Зная значение M_i , автомат A_3 может спланировать последовательность действий, переводящих A_0 в состояние q_{0s} , желательное для данной учебной ситуации. Если предположить, что функция Ψ линейно зависит от сложности $L(D)$ упражнения D , $\varphi(q, D) = a_i \psi(q, D, E_i)$, то подкласс M_i будет задаваться несколькими параметрами: длиной N цепочки состояний и коэффициентами линейных функций φ и Ψ .

Таким образом, автору курса достаточно определить набор типов учеников и описать, какое состояние q_{0s} является желательным на каждом этапе обучения. После этого автомат A_2 может решать задачу отнесения A_0 к определенному подклассу M_i и определения его текущего состояния q_{0i} , а автомат A_3 может спланировать последовательность действий, переводящих A_0 в требуемое состояние q_{0s} .

Автомат A_4 представляет собой базу данных учебных стратегий, для которой вход (B_2, B_3) является запросом, а выход B_4 - результатом обработки этого запроса. Аналогично автомат A_5 может быть описан как база данных учебного материала, задача которой - подобрать материал, отвечающий запросу B_4 .

Чтобы избежать ситуации, когда A_4 и A_5 не могут найти данных, удовлетворяющих запросам, следует представлять B_3 и B_4 как упорядоченные наборы запросов (B_{30}, \dots, B_{35}) и (B_{40}, \dots, B_{4L}). Если запрос B_{30} не может быть удовлетворен, удовлетворяется запрос B_{31} и т.д. Более сложная модель может включать в себя обратную связь между автоматами A_4 и A_3 , а также A_5 и A_3 . Тогда, если запрос (B_2, B_3) или B_4 не может быть удовлетворен, автомат A_3 предлагает другой вариант действий B_3 [5].

Ниже приведем некоторое описание рекомендаций для выработки стратегий управляющим автоматом (экспертной системой) для тестовой компоненты обучения.

В простейшем случае тест понимается как набор вопросов, оцениваемых по шкале да/нет (справился с упражнением или нет). В зависимости от количества ошибок ученика предлагается следующий набор реакций обучающей системы.

1. Ошибок меньше одной $3N \pm 1$ - успешное прохождение теста, ученику предоставляется возможность быстрого просмотра ошибочных ответов, возможен (один) возврат в контрольную точку теста, разрешается движение дальше по обучающему материалу.
2. Ошибок больше одной $3N \pm \Delta$, но меньше пяти $12N \pm \Delta$ - среднее качество прохождения теста, необходим возврат на неправильные ответы (возможно с фиксированной скоростью просмотра), число возвратов в контрольную точку не более двух. При третьем возврате в контрольную точку ученик отсылается на учебный материал и только после его изучения повторяется тест.
3. Ошибок больше пяти $12N \pm \Delta$ - неудачное прохождение теста, предполагается обязательный возврат на неправильные ответы с экспертным временем на просмотр и числом возвратов в контрольную точку не более одного раза. При повторном возврате в контрольную точку ученик отсылается на учебный материал и только после его изучения происходит повторное выполнение теста.

Приведенная выше концепция и примеры формализаций тестовой составляющей экспертной системы самообучения являются инструментами математического моделирования процесса обучения, которые получают дальнейшее свое развитие в рамках формализованных алгоритмов и программного инструментария для компьютерного моделирования реальных процессов обучения.

Приведенные формальные фрагменты модели позволяет описывать и строить экспертные системы самообучения общего вида на основе автоматных моделей. В некоторых случаях будет удобно разработать свою модель для отдельных компонент учебного процесса, а затем встроить ее в уже функционирующую систему.

Представленная концепция архитектуры интеллектуальной информационной системы в виде экспертной системы, обеспечивающей вариативность траекторий самообучения, в общем случае может обеспечивать достижение эффективности самообучения, благодаря возможностям реализации самостоятельного выбора траекторий, оперирования информацией о результатах самообучения и привлечению информационных ресурсов, создаваемых как самим обучающимся, так и поступающим из внешних источников информацией.

Литература

1. Ваграменко Я.А., Фанышев Р.Г. Технология интеллектуального анализа текстовой информации в базах знаний образовательной экспертной системы // Педагогическая информатика. 2011. №1. С.57-62.
2. Ваграменко Я.А., Яламов Г.Ю., Фанышев Р.Г. Выбор источников информации, характера контента, оценки научного и социально значимой информации для поддержки самообразования // Педагогическая информатика. 2013. №2. С. 49-61.
3. Вашик К., Кудрявцев В.Б., Строгалов А.С. Проект «IDEA». Введение в новое поколение программного обеспечения типа ICBI для передачи знаний и навыков с помощью экспертной системы. Dortmund: Link&Link Software GmbH, 1995.
4. Гребнев А. Н. AtLeap – Java каркас с открытым исходным кодом для Web-приложений // Вестник ИжГТУ. 2006. №21. С. 64-68.
5. Кудрявцев В.Б., Алисейчик П.А., Вашик К. Моделирование процесса обучения // Фундаментальная и прикладная математика. 2009. т. 15. С. 111-169.
6. Построение экспертных систем: пер. с англ. / П. Хейс-Рот, Д Уотерман, Д. Ленат. М.: Мир, 1987. 430 с.
7. Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. М.: ООО Издательство «Научиздатлит», 2008. 482 с.
8. Фанышев Р.Г., Ваграменко Я.А. Моделирование алгоритмов обучения с применением средств логического вывода // Информатизация образования и науки. 2012. №24(16). С. 133-140.