

АВТОМАТНЫЙ ПОДХОД В ПОСТРОЕНИИ КОМПЬЮТЕРНЫХ ОБУЧАЮЩИХ ДИАЛОГОВ

А. Д. Тазетдинов,

канд. техн. наук, директор центра информационных технологий
АНО ВПО «Международный банковский институт»

Рассматривается автоматный принцип построения вопросно-ответных структур компьютерных адаптивных обучающих диалогов, предлагается технология их создания и применения.

Введение

В наступившем XXI веке происходит смена образовательной парадигмы, появляются новые педагогические концепции и идеи. Содержание образования насыщается различными образовательными программами, стимулирующими развитие у обучающихся умений и способностей оперировать информацией, творчески решать поставленные задачи. Использование компьютерных средств обучения, телекоммуникационных сетей глобального масштаба изменяет традиционные способы использования информации. Ценностными ориентирами в образовании становятся личность ученика, его способность к самостоятельной деятельности по сбору, обработке и анализу информации, умение принимать решения и применять полученные знания в жизни.

Тенденциями современной образовательной системы становятся социальная направленность и активное использование обучения в ситуационно-смоделированных реальностях, где обучение происходит в социальной группе, члены которой в сотрудничестве создают малую культуру совместных артефактов, результатов деятельности, имеющих общие значения. Это не только ролевые и деловые игры, позволяющие приблизить обучение к реальной действительности, на первый план выходят такие понятия, как конструктивизм и конструкционизм. Смысл этих понятий в утверждении, что люди активно приобретают новое знание по мере взаимодействия с окружающим миром, и что обучение особенно эффективно, когда учащийся создает что-то для передачи опыта другим, выстраивая речевые сообщения, интернет-сообщения и более сложные продукты, например произведение изобразительного искусства или комплект программного обеспечения.

Подобные утверждения основаны на результатах многочисленных исследований в области когнитивной психологии [1]. Эти результаты указы-

вают на то, что восприятие визуально предъявленных слов (узнавание) происходит автоматически и практически независимо от внимания, тогда как восприятие семантически связной информации тесно связано с факторами внимания и активизируется только тогда, когда испытуемого просят обработать слово активно, например назвать его или молча классифицировать. Поэтому если необходимо, чтобы обучаемые имели знания в активной форме, требуется активный анализ и обязательное воспроизведение изученного материала.

В этих условиях особенно актуальное значение приобретает создание программного обеспечения, позволяющего в рамках автоматизированных обучающих систем (АОС) имитировать обучающий диалог на естественном языке, не накладывая никаких явных ограничений на терминологию и фразеологию ответа обучаемого. Тем не менее, совершенно очевидно, что дидактическая эффективность любых АОС будет определяться не совершенством технических решений, принятых в области информатики и вычислительной техники, а характером решений, принятых в сфере педагогической науки.

Технологии реализации обучающих диалогов

Естественный язык человека наряду с преимуществами в осуществлении коммуникативной функции обладает и недостатками, к числу которых относятся многозначность слов, сложность грамматических норм, громоздкость и необозримость его конструкций, ситуативность многих конструкций, контекстно-зависимое представление информации, небрежность употребления терминов и т. д.

В литературе термин «диалог» нередко трактуется весьма широко, и диалоговым иногда объявляется любое взаимодействие человека с компьютером. При столь расширенном толковании про-



■ Рис. 1. Технологии реализации диалогового режима в АОС

блема диалога размывается. Большинство исследователей и разработчиков адаптивных и интеллектуальных АОС полагают, что говорить о полноценном диалоге между учащимся и компьютером можно будет только после решения сложной проблемы кибернетики — понимания компьютером естественного языка. Во всех остальных случаях речь идет о мере ограничения либо языка, либо предметной области для реализации диалогового режима обучения (рис. 1).

Иногда эти ограничения весьма значительны и заметно затрудняют общение. Большинство АОС используют в качестве языка общения некоторое подмножество естественного языка, которое характеризуется строгой предметной направленностью, обедненной лексикой и упрощенным синтаксисом. По этой причине некоторые специалисты считают, что при общении с системой узкой ориентации целесообразно использовать не ограниченный естественный язык, а специализированные формальные языки, поскольку изучить директивы легче, чем запомнить ограничения, налагаемые на естественный язык, и следовать им в ходе диалога, следить за допустимостью формулировок сообщений. В специализированных системах в качестве средства общения иногда используются редуцированные проблемно-ориентированные языки — рабочие языки. Они имеют четкую предмет-

ную направленность, по сравнению с естественным ограничены в лексике, синтаксисе и сфере применения.

Адаптивные сценарии обучающих диалогов

В результате проведенного исследования технологий организации диалогового обучения с применением компьютера предлагается автоматный принцип построения вопросно-ответных структур обучающего диалога. Автоматы и автоматное программирование является сегодня не только перспективным, но и весьма актуальным направлением научных исследований. Большой вклад в развитие парадигмы автоматного программирования внесли А. А. Шалыто и его ученики [2–6], а также другие отечественные и зарубежные ученые, например С. И. Баранов [7]. Преимущество автоматной реализации сценариев диалогов перед другими технологиями состоит в том, что, с одной стороны, автоматные алгоритмы существенно проще таких технологий, как нейронные сети, алгоритмы data mining и другие методы поиска и анализа текстов, и это дает возможность преподавателям самостоятельно, без привлечения сторонних специалистов, с помощью специального инструментария создавать и управлять обучающими диалогами. С другой стороны, они обладают большей гибко-

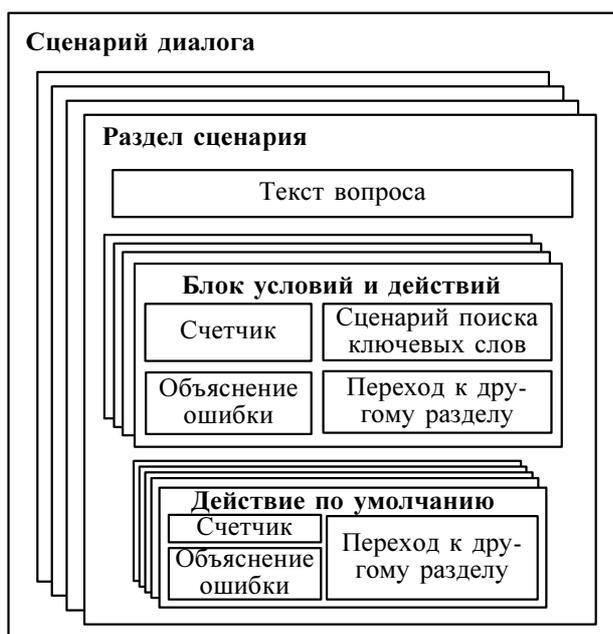
стью и интеллектуальностью по сравнению с простым поиском ключевых слов и даже деревьев решений с циклами [8].

Сценарии обучающих диалогов состояются из разделов, соответствующих отдельным состояниям автомата, объединенных переходами. Количество разделов не ограничено и определяется только логикой описываемого диалога. Раздел сценария является унифицированной структурой с фиксированным набором типовых блоков, количество которых может изменяться произвольным образом (рис. 2). Каждый раздел может содержать дополнительный вопрос, а также дополнительные блоки, осуществляющие многошаговый анализ ответа обучаемого, что позволяет создавать адаптивные обучающие диалоги произвольной длины и сложности. Переход из одного раздела в другой выполняется на основе параметров, заданных для текущего входа в раздел: блоками условий и действий для найденного выражения поиска или блоками действий по умолчанию. Такой подход в структурной организации сценариев позволяет применять универсальный алгоритм для обработки сценариев произвольных размеров и задавать контент сценария декларативно, посредством экранных форм пользовательского интерфейса.

Структура сценария также может быть задана в виде грамматики, интерпретируемой автоматным преобразователем. В качестве стандарта де-факто в таких случаях для удобства использования и переносимости принято использовать язык XML.

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE question [
<!ELEMENT ball (#PCDATA)>
```

```
<!ELEMENT tema (#PCDATA)>
<!ELEMENT captionq (#PCDATA)>
<!ELEMENT questiontext (#PCDATA)>
<!ELEMENT states (state)+ >
<!ELEMENT state (statnum, questiontext?, reparestr,
flod*, dodefault*) >
<!ELEMENT statenum (#PCDATA)>
<!ELEMENT questiontext (#PCDATA)>
<!ELEMENT reparestr (#PCDATA)>
<!ELEMENT flod (counter, flodexpr, rightreply,
strictsearch, listdescriptors, syscomment, ball,
move_to, show)>
<!ELEMENT flodexpr (#PCDATA)>
<!ELEMENT strictsearch (#PCDATA)>
<!ELEMENT listdescriptors (#PCDATA)>
<!ELEMENT dodefault (counter, syscomment, ball,
move_to, show)>
<!ELEMENT counter (#PCDATA)>
<!ELEMENT syscomment (#PCDATA)>
<!ELEMENT move_to (#PCDATA)>
<!ELEMENT show (#PCDATA)>
]>
```



■ Рис. 2. Структура сценария обучающего диалога

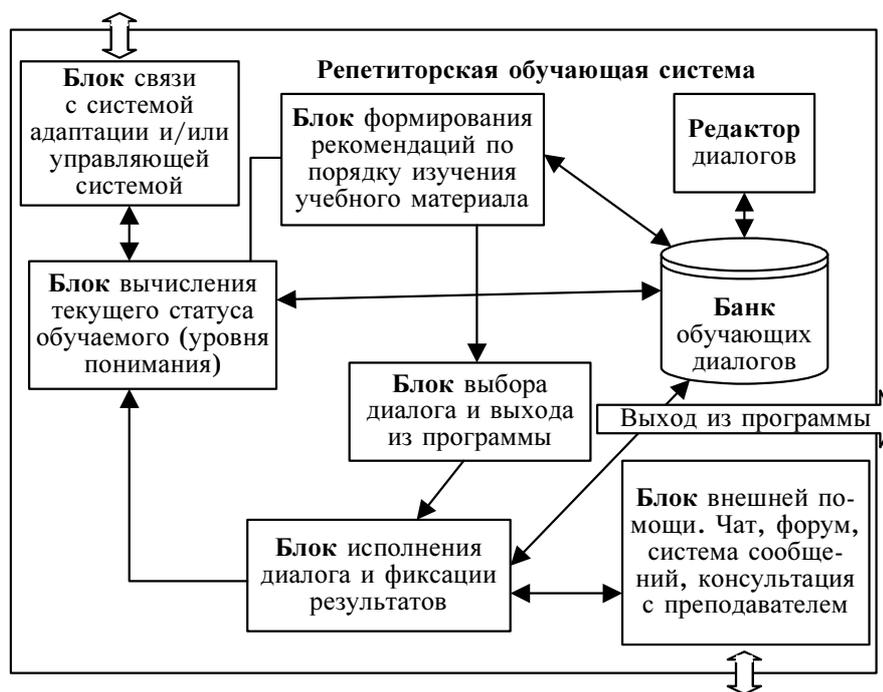
Блок условий и действий включает в себя алгоритм (сценарий) поиска определенных ключевых слов в ответе учащегося и связанных с ним набором реакций системы для текущего значения счетчика в случае совпадения выражения поиска с ответом обучаемого. Методика построения сложных сценариев (алгоритмов) поиска определенных ключевых слов основана на логико-семантическом методе анализа контекстно-зависимых высказываний, предложенным А. И. Стригуном [9].

Блок действий по умолчанию позволяет, для текущего значения счетчика, вывести соответствующий комментарий системы и перейти к требуемому разделу, согласно логики диалога. Нет необходимости описывать все возможные комбинации ответов, достаточно определить наиболее значимые, а остальные определить как неверные в действии по умолчанию. В любом случае, благодаря декларативному представлению контента сценария преподаватель всегда может самостоятельно изменить или расширить его содержимое. Таким образом, сценарий задает не жесткую линейную последовательность сцен, а гибкую схему (переходов между разделами), адаптированную к ответам учащихся.

Автомат с памятью

В отличие от многих АОС, использующих методы традиционного программирования, для компьютерных автоматизированных обучающих систем репетиторского типа (РАОС) автором был предложен автоматный подход реализации как самой системы и всех ее элементов, так и структуры сценариев многошаговых обучающих диалогов. Программная архитектура РАОС показана на рис. 3.

Отличие репетиторских систем от других АОС определяется, прежде всего, наличием диалоговой формы взаимодействия обучающегося с системой в реальном масштабе времени, где обратная связь осуществляется не только при контроле, но и в процессе усвоения знаний. Благодаря такому



■ Рис. 3. Программная архитектура репетиторской обучающей системы

подходу проверка работы системы может быть выполнена с помощью имитационной модели, построенной в любой из существующих сред моделирования. Для проверки теоретических положений и апробации результатов работы был создан программный модуль. В силу большого объема документации в данной статье рассматривается только его небольшая часть, а подробное описание приводится в работах [8–11]. Предлагается два варианта построения автомата: с помощью графа переходов и с помощью функций переходов и выходов. Первый более нагляден, а второй, на наш взгляд, удобен при создании программы проверки работы автомата в любой из программных сред, например в пакете MATLAB.

В качестве примера рассмотрим одну из частей блока исполнения обучающего диалога, подсистему анализа ответов обучаемого, которая реализует функции анализа ответов, интерактивные механизмы взаимодействия с пользователем в виде подробных описаний ошибок и переходов к дополнительным вопросам в случае неверных ответов. Логика работы анализатора ответов описывается схемой алгоритма, преобразуемой в автомат с помощью метода, предложенного Барановым [7] и развитого Шалыто и Туккелем [4]. В этом методе начало и конец схемы алгоритма соответствуют начальным состояниям автомата. Остальным состояниям для автомата Мура соответствует каждая операторная вершина. Каждый условный оператор соответствует определенному входному сигналу, а выходные сигналы и действия соответствуют действиям в операторных вершинах.

Проверка корректности построения и логики работы элементов системы выполнялась с помощью имитационного моделирования в пакете MATLAB-Simulink-Stateflow. Для повышения наглядности процесса моделирования были выполнены некоторые упрощения, не изменяющие логику работы автомата:

- входные и выходные переменные могут принимать значения $\{0, 1\}$;
- входное слово — это набор из всех входных переменных, который представляет собой плоскую проекцию одного из возможных вариантов событий, возникающих в разное время (распределенных по времени) работы системы; обозначается z_i , где i — порядковый номер набора;
- полной комбинацией возможных событий в системе является полный набор комбинаций входных переменных.

Формально конечный автомат A определяется как шестерка объектов: $A = \langle S, X, Y, s_0, \delta, \lambda \rangle$, где:
 S — конечное непустое множество (состояний);
 X — конечное непустое множество входных переменных (входной алфавит);
 Y — конечное непустое множество выходных переменных (выходной алфавит);
 $s_0 \in S$ — начальное состояние;
 $\delta: SX \rightarrow S$ — функция переходов;
 $\lambda: SX \rightarrow Y$ — функция выходов.

Интерпретатор сценариев реализует функции предъявления вопросов, анализа ответов, интерактивные механизмы взаимодействия с пользователем в виде подробных описаний ошибок и переходов к дополнительным вопросам в случае неверных

■ *Входные и выходные переменные интерпретатора сценариев*

Входные сигналы	Действия
$x1$ Ожидание идентификатора сценария или выбор раздела	$y0$ Ожидание идентификатора сценария, номера раздела или ответа обучающегося
$x2$ $i < n$	$y1$ Установить код ошибки $err = 0$, выбрать требуемый раздел, увеличить счетчик раздела
$x3$ Ответ совпал с маской правильного ответа	$y2$ Получение следующей маски правильного ответа
$x4$ $i < m$	$y3$ Сравнение маски с ответом
$x5$ Ответ совпал с маской неправильного ответа	$y4$ Получение следующей маски неправильного ответа
$x6$ Ошибка	$y5$ Получение параметров по умолчанию
$x7$ Ошибка	$y6$ Стандартный обработчик
	$y7$ Установить код ошибки $err = 1$
	$y8$ Вернуть объяснения ошибки и номер раздела для перехода

ответов. Входные и выходные переменные интерпретатора сценариев представлены в таблице.

Для работы интерпретатора определяются:

$$S = \{a0, a1, a2, a3, a4, a5, a6, a7\};$$

$$X = \{x1, x2, x3, x4, x5, x6, x7\};$$

$$Y = \{y0, y1, y2, y3, y4, y5, y6, y7, y8\};$$

$$s0 = a0.$$

В каждый момент времени $t = \{0, 1, 2, \dots, n\}$ автомат находится в некотором состоянии $a_i = a(t)$. Функция переходов определяет состояние $a(t)$ в некоторый момент t . Для описываемого автомата строятся функции переходов:

$$a0(t) = (a0(t-1) \wedge \bar{x}1(t-1)) \vee a7(t-1);$$

$$a1(t) = a0(t-1) \wedge x1(t-1);$$

$$a2(t) = a1(t-1) \vee (a2(t-1) \wedge x2(t-1) \wedge \bar{x}3(t-1));$$

$$a3(t) = (a2(t-1) \wedge \bar{x}2(t-1) \wedge \bar{x}3(t-1)) \vee (a3(t-1) \wedge x4(t-1) \wedge \bar{x}5(t-1));$$

$$a4(t) = a3(t-1) \wedge x4(t-1) \wedge \bar{x}5(t-1);$$

$$a5(t) = a4(t-1) \wedge x6(t-1);$$

$$a6(t) = a5(t-1) \wedge x7(t-1);$$

$$a7(t) = a6(t-1) \vee (a5(t-1) \wedge \bar{x}7(t-1)) \vee (a4(t-1) \wedge \bar{x}6(t-1)) \vee (a2(t-1) \wedge x3(t-1)).$$

Соответственно функции выходов:

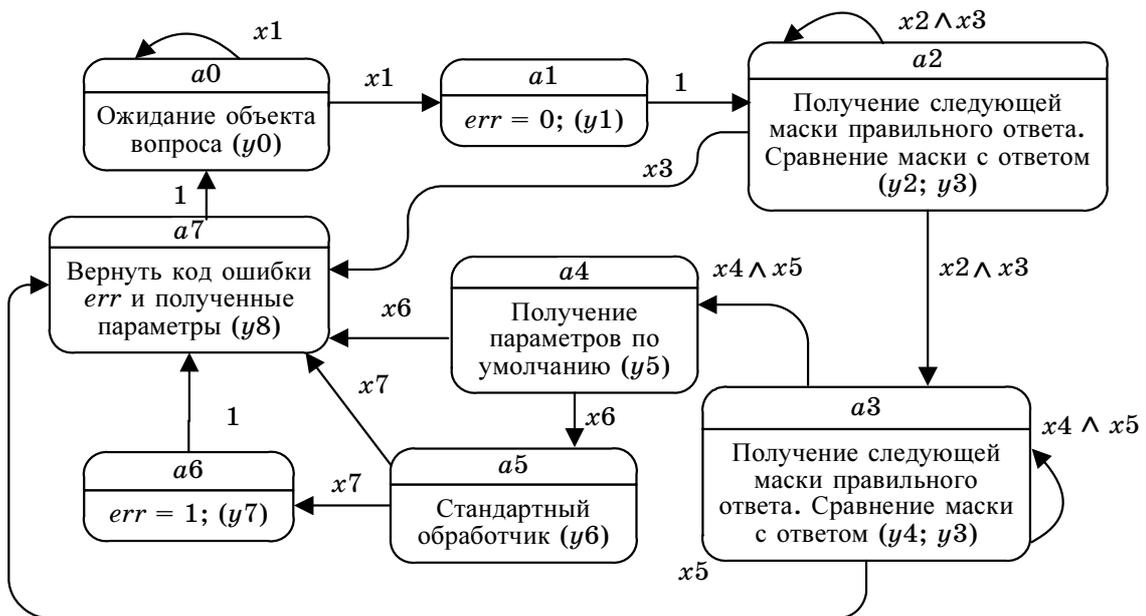
$$y0(t) = a0(t); y3(t) = a2(t) \vee a3(t); y6(t) = a5(t);$$

$$y1(t) = a1(t); y4(t) = a3(t); y7(t) = a6(t);$$

$$y2(t) = a1(t); y5(t) = a4(t); y8(t) = a7(t).$$

По этим выражениям составляется граф переходов (рис. 4) и разрабатывается управляющий блок, который обеспечивает работу анализируемого устройства.

Полный набор комбинаций входных переменных будет состоять из 127 входных слов $z0(0000000) - z127(1111111)$. Слову соответствует последовательность символов $x_1x_2x_3x_4x_5x_6x_7$, каждый из которых может принимать только значение из диапазона $\{0, 1\}$. Проверка корректности работы автомата на всех возможных комбинациях входных сигналов выполняется с помощью программы, созданной в пакете MATLAB и использующей уравнения функций переходов данного автомата. Отсутствие вершин, не задействованных в работе ни при каких входных переменных, определяется сравнением полученного множества с заявленным.



■ *Рис. 4. Граф переходов автомата*

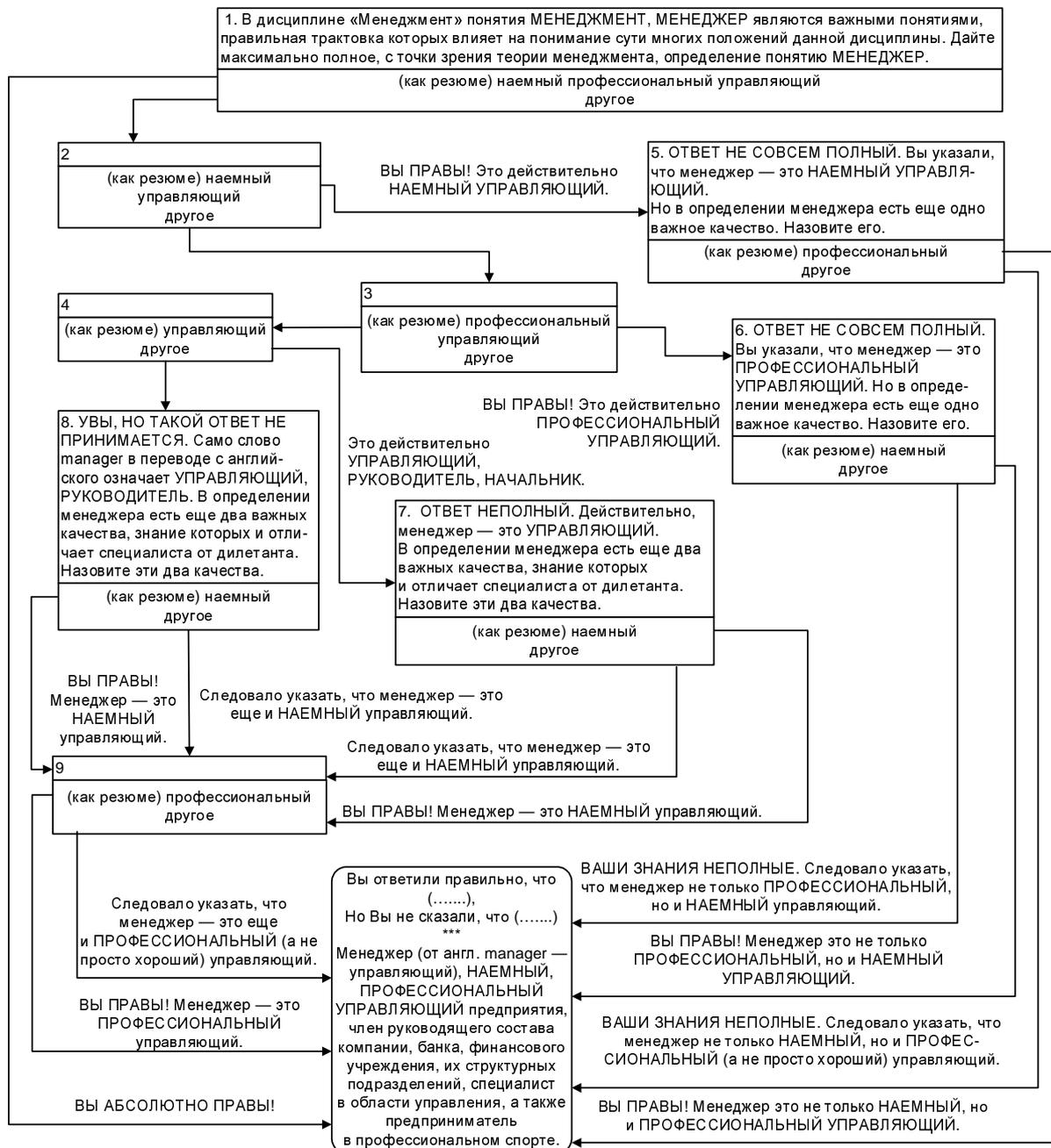
$S = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\};$
 $S' = \{a_0\} \cup \{a_0, a_1, a_2, a_3, a_4, a_7\} \cup \{a_0, a_1, a_2, a_3, a_4, a_5, a_7\} \cup \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\} \cup \{a_0, a_1, a_2, a_3, a_7\} \cup \{a_0, a_1, a_2, a_7\};$
 $S' = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\};$
 $S = \{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\} \Rightarrow S' \sim S.$

Вершин, не используемых ни при каких переменных, нет, следовательно, автомат работает корректно.

Для проверки теоретических положений и апробации результатов работы был создан модуль (см. рис. 3) к программному комплексу, разработа-

тываемому в рамках научно-исследовательской работы по теме «Разработка сетевой интегрированной информационно-обучающей системы для регионального ресурсного центра в сфере образования» по гранту мэрии Санкт-Петербурга. Как самостоятельное программное средство, модуль прошел регистрацию в отраслевом фонде алгоритмов и программ [10].

В данной статье приводится только фрагмент реализации с помощью автоматного подхода одного из блоков этого модуля с целью продемонстрировать эффективность автоматного подхода для



■ Рис. 5. Пример структуры обучающего диалога

создания обучающих систем диалогового типа. Целиком РАОС состоит из десяти блоков и включает в себя интерпретатор сценариев диалогов, анализатор ответов обучающихся, подсистему ввода вывода информации, подсистему связи с обучающей системой, а также редактор сценариев обучающих диалогов, реализуемых в виде автоматов с памятью. Результатом работы с системой является автоматизированный многошаговый обучающий диалог, позволяющий в рамках РАОС вести обучение на естественном языке, не накладывая никаких явных ограничений на терминологию и фразеологию ответа обучаемого. На рис. 5 представлен сокращенный пример такого обучающего диалога.

В настоящее время преподавателями четырех вузов по разработанной автором методике создано для 144 дисциплин около 11 тыс. обучающих диалогов, которые успешно используются в учебном процессе Международного банковского института, Санкт-Петербургского государственного политехнического университета, Санкт-Петербургского государственного университета экономики и финансов, Самарского Международного института рынка. Кроме того, к текущему моменту разработана реализация этого модуля для такой популярной среды обучения, как Moodle.

Заключение

В отличие от существующих методов организации обучающих диалогов в АОС, применение автоматного подхода на всех уровнях построения РАОС, включая адаптивные сценарии обучающих диалогов, позволяет отказаться от сложного семантического и синтаксического анализа ответа учащегося. Предлагаемая технология дает возможность существенно упростить программные алгоритмы, реализовать в обучающей системе относительную предметную независимость, использовать естественный язык для ввода ответов учащихся, а преподавателям самостоятельно, без привлечения программиста, создавать и модифицировать сценарии адаптивных диалогов посредством экранных форм пользовательского интер-

фейса. Практическое применение предложенных теоретических положений возможно:

- для построения самостоятельной адаптивной обучающей системы;
- в качестве структурной основы для построения отдельных модулей к различным обучающим системам.

Литература

1. Солсо Р. Когнитивная психология. 6-е изд. СПб.: Питер, 2006. 589 с.
2. Шопырин Д. Г., Шалыто А. А. Объектно-ориентированный подход к автоматному программированию // Информационно-управляющие системы. 2003. № 5. С. 29–39.
3. Степанов О. Г., Шалыто А. А., Шопырин Д. Г. Предметно-ориентированный язык автоматного программирования на базе динамического языка RUBY // Информационно-управляющие системы. 2007. № 4. С. 22–27.
4. Шалыто А. А., Туккель Н. И. Преобразование итеративных алгоритмов в автоматные // Программирование. 2002. № 5. С. 12–26.
5. Казаков М. А., Шалыто А. А. Автоматный подход к реализации анимации в визуализаторах алгоритмов // Компьютерные инструменты в образовании. 2005. № 3. С. 62–76.
6. Корнеев Г. А., Шалыто А. А. Автоматизированное построение визуализаторов алгоритмов дискретной математики // Компьютерные инструменты в образовании. 2006. № 5. С. 16–26.
7. Баранов С. И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
8. Тазетдинов А. Д. Интерактивные процессы в обучающих системах: методы управления. СПб.: Изд-во Политех. ун-та, 2007. 155 с.
9. Федоров Б. И., Джалишвили З. О. Логика компьютерного диалога. М.: Онега, 1994. 240 с.
10. Система тестирования с использованием свободного ответа: А. С. 4229 ОФАП / А. Д. Тазетдинов, А. И. Стригун. № ГР 50200500081 // Алгоритмы и программы: Сб. М.: ВНИИЦ, 2005.
11. Тазетдинов А. Д., Стригун А. И. Редактор компьютерных интеллектуальных тьюторов в ВУОКСе: Руководство пользователя / МБИ. СПб., 2005. 34 с.