

А. Д. ТАЗЕТДИНОВ

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ОБУЧАЮЩИХ ДИАЛОГОВ В КОМПЬЮТЕРНЫХ СИСТЕМАХ РЕПЕТИТОРСКОГО ТИПА

Обсуждаются вопросы использования конечных автоматов с памятью для построения многошаговых обучающих диалогов в системах репетиторского типа, что позволяет существенно упростить программные алгоритмы, имитировать обучающий диалог на естественном языке, не накладывая явных ограничений на терминологию и фразеологию ответа обучаемого, и реализовать в обучающей системе гибкое индивидуализированное обучение. Вывод соответствующих реплик или подсказок в каждом из состояний автомата, в зависимости от текущего значения счетчика посещений этого состояния, способствует существенному повышению интеллектуальности диалога.

Ключевые слова: репетиторские обучающие системы, диалоговые системы, автоматизированные обучающие системы.

Потребностью повышения качества образования обусловлено создание различных автоматизированных обучающих систем (АОС), в которых применяются прогрессивные методы обучения с использованием новейших информационных технологий. Среди этих систем особо можно выделить компьютерные автоматизированные обучающие системы репетиторского типа. 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 знаний, но и в процессе их усвоения. Тем не менее существует ряд проблем как технического, так и педагогического характера, решение которых могло бы способствовать активному развитию и использованию репетиторских АОС в учебном процессе. К техническим проблемам следует отнести проблемы анализа текста ответов обучающихся и технологии создания структур вопрос — ответ в диалоговом модуле. Как полагают большинство исследователей и разработчиков адаптивных и интеллектуальных АОС, 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 об ограничении либо семантической составляющей языка, либо предметной области для реализации диалогового режима обучения [1]. В этих условиях актуальное значение приобретает создание программного обеспечения, позволяющего в рамках репетиторской АОС имитировать обучающий диалог на естественном языке, не накладывая явных ограничений на терминологию и фразеологию ответа обучаемого.

В работе [2] на основе анализа более чем 100 часов учебных взаимодействий обучающихся и преподавателей выделены такие особенности учебных диалогов, как их многошаговая структура и инициированные преподавателем серии диалогов. Кроме позитивного влияния этих факторов на результаты обучения, их легче воплотить в обучающей системе, чем сложные методы и

стратегии, предлагаемые другими исследователями и создателями интеллектуальных обучающих систем. Обычно диалог при индивидуальном обучении растянут, т.е. состоит из нескольких реплик, с помощью которых преподаватель и учащийся совместно ищут ответ на вопрос или решают проблему. Процесс объяснений, уточнений и мысленных моделей представления материала крайне необходим для обучения и обычно более эффективен, чем простое предоставление информации учащемуся.

В настоящей статье по результатам проведенного анализа технологий диалогового обучения в АОС излагаются соображения по использованию в репетиторских АОС автоматов с памятью для построения структур вопрос — ответ в ходе обучающего диалога и реализации сценариев диалогов по этому принципу. Наличие памяти в каждом из состояний автомата (шагов или сцен диалога) позволяет осуществлять дифференцированный анализ ответов обучаемого (в зависимости от текущего счетчика посещений этого состояния) и вывод соответствующих реплик или подсказок, тем самым повышая интеллектуальность диалога.

Существуют два уровня формирования логики диалога. Первый (или внешний) уровень — это собственно сам сценарий и его разделы, соответствующие состояниям автомата (рис. 1), условия переходов из одного состояния в другое, а также комментарии и подсказки системы. Количество разделов не ограничено и определяется только логикой описываемого диалога. В каждом разделе могут содержаться дополнительный вопрос, а также дополнительные блоки, осуществляющие многошаговый анализ ответа обучаемого, что позволяет создавать диалоги произвольной длины и сложности. Переход из одного раздела в другой выполняется на основании совпадения ответа обучающегося с шаблоном возможного варианта ответа, заданного в блоке, содержащем условия и действия сценария для данного раздела, или в блоке, содержащем действия по умолчанию. Структура сценария также может быть задана в виде грамматики, интерпретируемой автоматным преобразователем. В качестве стандарта в таких случаях принято использовать язык XML.

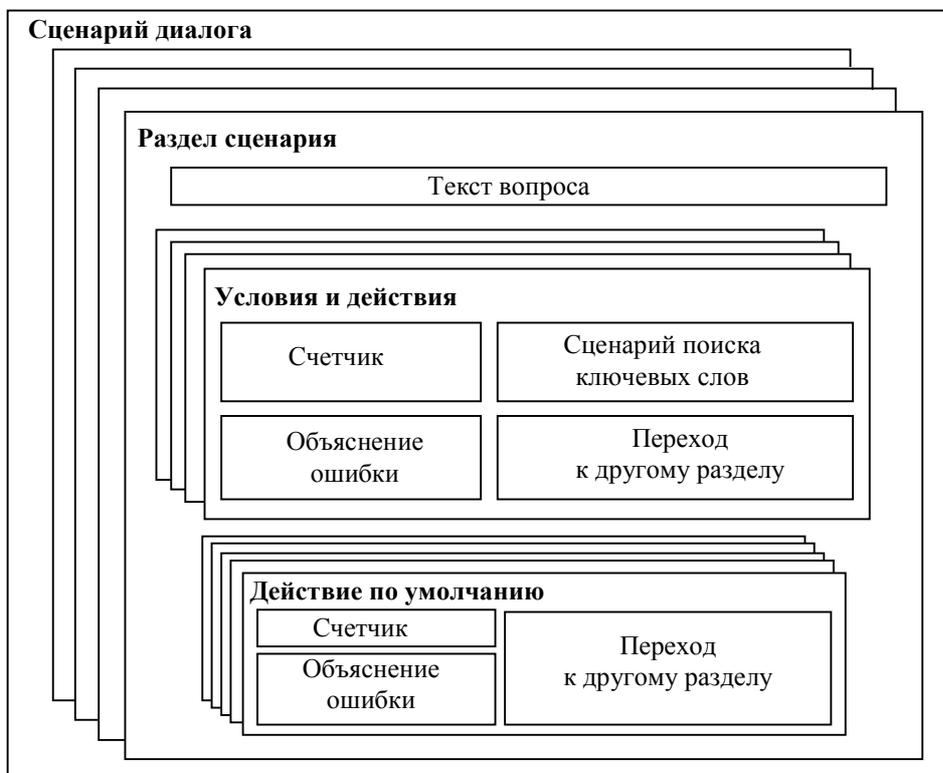


Рис. 1

Второй (или внутренний) уровень формирования логики диалога — это сценарий (алгоритм или выражение) поиска ключевых слов, определяющий, какие слова и в какой

последовательности должны быть найдены: должны ли они находиться в какой-либо связи между собой, должны ли они вычеркиваться из строки поиска и/или восстанавливаться после нахождения, и какая степень точности (распознавания) должна быть соблюдена. В качестве механизма описания алгоритма поиска ключевых слов предлагается метод, разработанный А. И. Стригуном [3, 4] для синтаксического анализа контекстно-зависимых высказываний на основе разбора функций логических дескрипторов. Механизм префиксной записи операторов идеально подходит для стековой или рекурсивной обработки выражения, а само выражение является алгоритмом выполнения проверки текста ответа учащегося. Преимуществом языка, лежащего в основе данного метода, являются простота, гибкость и возможность составления собственных алгоритмов для каждого требующего проверки выражения. В расширенной форме представления грамматик Бэкуса — Наура этот язык описывается следующим образом:

```

флод ::= (оператор { флод } конец_оператора | оператор {
дескриптор } конец_оператора)
спец_символ ::= '``' расширен_поиск
расширен_поиск ::= (?,* )
дескриптор ::= { дескриптор } | (спец_символ, расширен_поиск,
конец_оператора, буква, цифра, '_')
оператор ::= ('&', '@', '^', '|', '\', '/', '[', ']', '{',
'}', '~', '~+', '~-')
конец_оператора ::= '.'

```

Обработку сценариев обучающих диалогов предлагается выполнять с помощью интерпретатора, построенного также по принципу автомата с памятью. Интерпретатор сценариев выполняет такие функции, как предъявление вопросов и анализ ответов, кроме того, он реализует интерактивные механизмы взаимодействия с пользователем в виде подробных описаний ошибок и переходов к дополнительным вопросам в случае неверных ответов. Логика работы интерпретатора описывается алгоритмом, преобразуемым в автомат Мура (рис. 2, здесь a_i — состояние автомата) с помощью метода, предложенного в работах [5—8]. Набор входных переменных и действий автомата представлен в таблице.

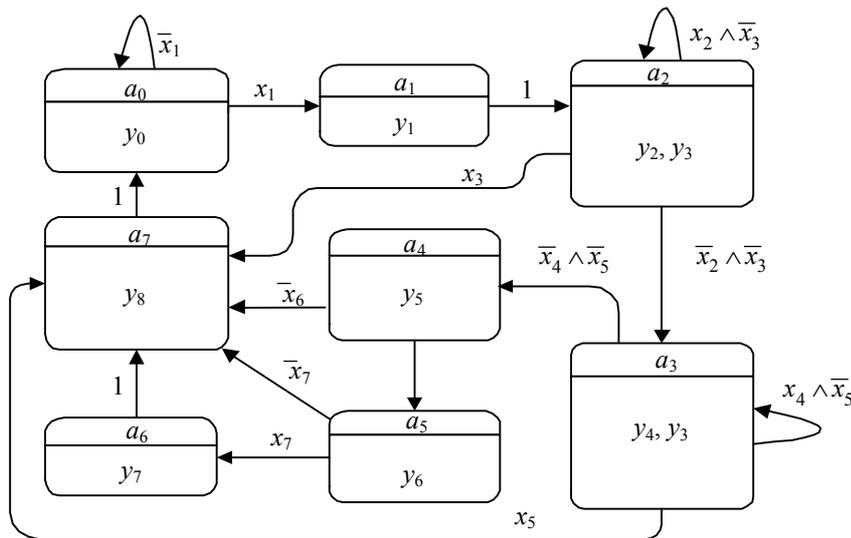


Рис. 2

Изложенный подход к структурной организации сценариев позволяет применять универсальный алгоритм для обработки сценариев произвольных размеров и задавать контент сценария декларативно, посредством экранных форм пользовательского интерфейса.

| Входная переменная | Действие |
|---|---|
| x_1 — ожидание сценария диалога или выбор раздела диалога | y_0 — ожидание сценария, номера раздела или ответа обучающегося |
| x_2 — проверка ответа обучающегося по всем доступным шаблонам правильного ответа раздела сценария | y_1 — установление кода ошибки: $err = 0$; выбор требуемого раздела, увеличение счетчика раздела |
| x_3 — совпадение ответа с шаблоном правильного ответа | y_2 — получение следующего шаблона правильного ответа |
| x_4 — проверка ответа обучающегося по всем доступным шаблонам неправильного ответа раздела сценария | y_3 — сравнение шаблона с ответом |
| x_5 — совпадение ответа с шаблоном неправильного ответа | y_4 — получение следующего шаблона неправильного ответа |
| x_6 — ошибка | y_5 — получение параметров по умолчанию |
| x_7 — ошибка | y_6 — выполнение действий по умолчанию |
| — | y_7 — установление кода ошибки: $err = 1$ |
| — | y_8 — возврат кода ошибки или номера следующего раздела сценария |

Применение рассмотренного подхода при построении обучающих диалогов дает возможность существенно упростить программные алгоритмы, реализовать в обучающей системе относительную предметную независимость и использовать естественный язык для ввода ответов учащихся. Вывод соответствующих реплик или подсказок в каждом из состояний автомата, в зависимости от текущего значения счетчика посещений этого состояния, способствует существенному повышению интеллектуальности диалога. Таким образом, использование автоматов с памятью позволяет реализовать в репетиторских АОС гибкое индивидуализированное обучение. Предложенные теоретические положения могут найти практическое применение при построении самостоятельной адаптивной обучающей системы и в качестве структурной основы для создания отдельных модулей к различным обучающим системам.

СПИСОК ЛИТЕРАТУРЫ

1. Тазетдинов А. Д. Интерактивные процессы в обучающих системах: методы управления. СПб.: Изд-во Политехн. ун-та, 2007.
2. Graesser A. C., Person N. K., Magliano J. P. Collaborative dialog patterns in naturalistic one-on-one tutoring // *Appl. Cognitive Psychology*. 1995. N 9. P. 359—387.
3. Федоров Б. И., Джалиашивили З. О. Логика компьютерного диалога. М.: Онега, 1994.
4. А.с. 4229 ОФАП. Система тестирования с использованием свободного ответа / А. Д. Тазетдинов, А. И. Стригун. № ГР 50200500081 // Сб. Алгоритмы и программы. М.: ВНИИЦ, 2005.
5. Баранов С. И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
6. Шалыто А. А., Туккель Н. И. Преобразование итеративных алгоритмов в автоматные // *Программирование*. 2002. № 5. С. 12—26.
7. Шопырин Д. Г., Шалыто А. А. Объектно-ориентированный подход к автоматному программированию // *Информационно-управляющие системы*. 2003. № 5. С. 29—39.
8. Степанов О. Г., Шалыто А. А., Шопырин Д. Г. Предметно-ориентированный язык автоматного программирования на базе динамического языка RUBY // Там же. 2007. № 4. С. 22—27.

Сведения об авторе

Андрей Дамирович Тазетдинов — канд. техн. наук; Международный банковский институт, Санкт-Петербург; директор центра информационных технологий;
E-mail: 191_nkr@bk.ru

Рекомендована кафедрой
технологии программирования
СПбГУ ИТМО

Поступила в редакцию
19.05.08 г.