

*Михаил Викторович КАДАНЦЕВ —  
научный сотрудник лаборатории  
интеллектуальных информационных систем*

*Валерий Алексеевич ШАПЦЕВ —  
директор лаборатории  
интеллектуальных информационных систем,  
доктор технических наук, профессор  
Тюменский государственный университет*

## **ФОРМИРОВАНИЕ КВАНТОВ ДЕЯТЕЛЬНОСТИ ПОСРЕДСТВОМ ЯЗЫКА НАД АЛФАВИТОМ КОМАНД, ИСПОЛНЯЕМЫХ КОМПЬЮТЕРОМ**

УДК 004.5

*АННОТАЦИЯ. В работе излагаются новые результаты описанного ранее и связанного с механизмом накопления «квантов» (последовательностей команд) исследования модели деятельности специалиста. Описывается язык команд и строится грамматика над этим языком. Предлагается метод выделения «квантов» в произвольной последовательности команд.*

*In this paper are set forth the new results of investigation which has been described before and connected with the mechanism of accumulation sequences of commands of specialist's activity model. The command language is described and on this basis the grammar is built. The method of accentuation of quantum in an arbitrary sequence of commands.*

**Введение.** В условиях интенсификации социально-экономической жизни все более актуально внимание к мощному резерву экономики информационного общества — неиспользуемому ресурсу производительности компьютеров: подавляющее большинство компьютеров (около 90%) используется на уровне 5-10% своих программно-технических возможностей. А если учитывать то, что служебные компьютеры используется в основном в рабочее время, можно говорить о существенно большем резерве. В результате экономика информатизации не использует более 80% своего потенциала. Часть неиспользуемой производительности компьютеров должна быть направлена на поддержку деятельности пользователя в его проблемной области, как профессиональной, так и бытовой. Решение проблемы повышения экономической эффективности использования компьютерных средств лежит через многодисциплинарные исследования и разработки в нескольких направлениях.

Настоящая работа посвящена проблеме выделения в последовательности команд «квантов» деятельности. Средством выделения квантов избраны конечные автоматы. В работе излагаются новые результаты исследования, описанного в [1] и связанные с механизмом накопления «квантов» модели деятельности специалиста, работа которого осуществляется с компьютером. Это накопление является мониторинговым компонентом разрабатываемой системы автоформализации предметной области пользователя. Уточняются понятия «действие» и «квант деятельности», введенные ранее в [1, 3].

**1. Обсуждение понятий «прецедент», «действие», «квант».** Перечисленные термины относятся к модели деятельности пользователя компьютера. Действие — это элемент, факт, событие в деятельности  $D$  специалиста. Определенная последовательность действий (возможно несколько последовательностей при смене условий  $C_m$ ) приводит к решению конкретной задачи

$R_j^{(i)}: \{a^{(ij)}_1, a^{(ij)}_2, a^{(ij)}_3, \dots, a^{(ij)}_{s(ij)} / C_0\} \rightarrow R_j^{(i)}$ , где  $a^{(ij)}_k$  — действие,  $i$  — номер вида деятельности,  $j$  — номер прецедента деятельности по решению  $j$ -ой задачи,  $s$  — количество элементов последовательности. Совокупность таких линейных последовательностей, приводящих к решению одно и той же задачи  $R_j^{(i)}$  при разных условиях  $C_m$ , названа нами прецедентом деятельности  $d_{ij}$  [1]. Начальное условие, при котором начинается прецедент, обозначено  $C_0$ .

Каждое действие, выполняемое посредством компьютера, может содержать «квант деятельности»  $\delta$  [1]. Понятием кванта обозначается одна команда или совокупность команд, выполняемых компьютером по инициативе пользователя. Так что действие представимо совокупностью квантов. Понятие кванта призвано автоматизировать процесс отображения в компьютере последовательности действий пользователя. Подробнее о квантах — в части 2 настоящей статьи.

Термин «прецедент» использован разработчиком UML И. Якобсоном [2] в следующем контексте: «Прецедент — спецификация (в нотации UML — В.Ш.) последовательности(ей) действий, которые может осуществлять система, подсистема или класс (объектов, искусственно созданных, термин ООП — В.Ш.), взаимодействуя с внешними акторами» (от англ. actor — М.К.) [2]. В общеупотребимом же смысле, «прецедент — случай или событие, имевшие место в прошлом и служащие примером или основанием для аналогичных действий (и условий — В.Ш.) в настоящем» [4]. В нашем исследовании этот термин применяется для обозначения совокупности условий и последовательности (последовательностей) действий (фактов), выполненных человеком и приведших к решению определенной задачи.

Иерархия рассматриваемых понятий отображена на рис. 1, 2. На них серым цветом выделены прецеденты, действия и команды, одинаковые (совпадающие, повторяющиеся) в различных видах деятельности, прецедентах и действиях соответственно.

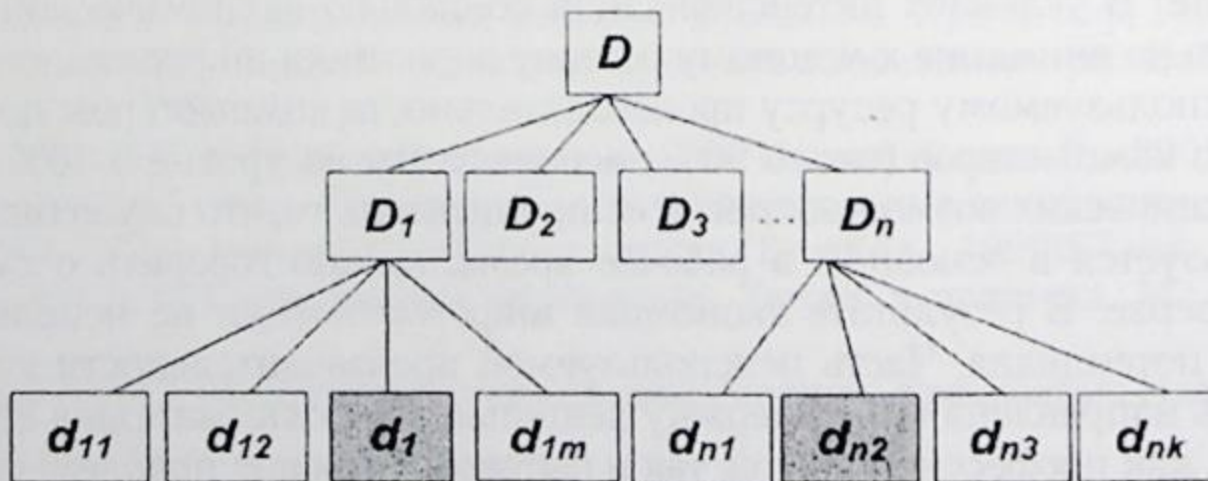


Рис. 1. Структура модели деятельности от корня до уровня прецедента

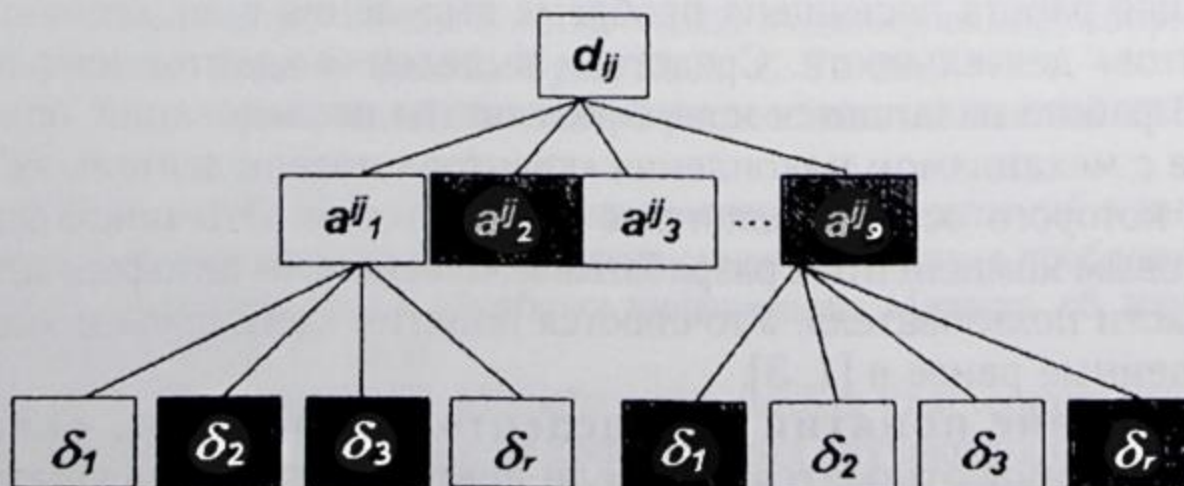


Рис. 2. Структура прецедента для уровня кванта деятельности

Заметим, что, как и у И. Якобсона, прецедент не содержит рекурсий, а только линейную последовательность действий и ее линейные ответвления при разных условиях.

**2. Кванты деятельности и команды, исполняемые компьютером.** Под командой здесь мы понимаем минимальное (элементарное) воздействие пользователя на ПО. Например, нажатие аппаратной или программной кнопки. Следует учесть, что в современных операционных системах и ПО простое нажатие кнопки (аппаратной или программной) порождает множество системных событий, приводящих к изменению состояния ПО, с которым в данный момент работает пользователь.

Практика фиксации команд, исполняемых компьютером, показала, что в разных действиях они повторяются [3]. С другой стороны, совокупность (последовательность) таких команд не позволила однозначно идентифицировать действия пользователя. Здесь уместно обратиться к понятию транзакции в теории БД как последовательности инструкций, направленных на решение определенной задачи и рассматриваемой как единое целое. Инструкции транзакции выполняют взаимоувязанные действия. Каждая инструкция решает часть общей задачи. Задача считается решенной только при выполнении всех инструкций [5]. По аналогии целесообразно определить и квант деятельности как последовательность команд компьютера, выполнение которых приводит к ожидаемому результату. Графически это отображено на рис. 3. Команды, выполняемые компьютером по инициативе пользователя, обозначены через  $\alpha_j$ .

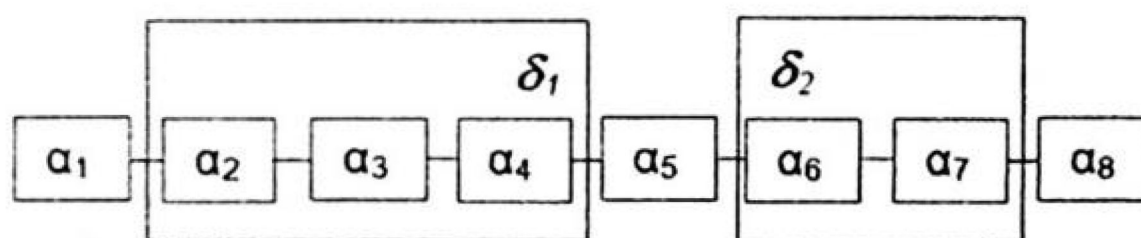


Рис. 3. Команды компьютера, объединенные в кванты деятельности

Здесь  $\alpha_1$ ,  $\alpha_5$ ,  $\alpha_8$  — вырожденные кванты, они состоят из одной команды.

Например, квант «форматирование текста» может состоять из команд: установка размера; установка цвета; установка шрифта. А вырожденный квант «удаление» состоит из одной команды — удаление выбранного объекта.

Каждая команда характеризуется моментом возникновения  $t$ , объектом воздействия  $o$  (в смысле ООП) и типом  $v$  воздействия на объект  $o$ . Причем воздействие на объект может быть оказано как методами самого объекта, так и методами других объектов. «В общем смысле объект — это сущность, обычно извлекаемая из словаря предметной области или решения. Каждый объект обладает идентичностью (его можно поименовать или иначе отличить от других объектов), состоянием (обычно с объектом бывают связаны некоторые данные) и поведением (с ним можно что-то делать или он сам может что-то делать с другими объектами)» [2]. Таким образом, минимальным набором, определяющим команду, будет совокупность: объект и операция, произведенная над объектом.

**3. Язык над алфавитом команд.** Язык команд и квантов — это искусственно создаваемая формальная система, которая должна быть описана аналитически. Конечный механизм задания языка — грамматика. С позиций теории формальных языков и грамматик рассматриваемые нами наименования команд компьютера составляют алфавит  $\Sigma$  (конечное непустое множество символов).

Языком над этим алфавитом является произвольное множество цепочек, специальным образом составленных из символов  $\Sigma$  [6].

Введем порождающую грамматику  $G[7]$ :  $G=(\Sigma, N, P, S)$ , где  $\Sigma$  — алфавит (терминальные символы);  $N$  — конечное множество вспомогательных (нетерминальных) символов, сформированных из терминальных;  $P$  — конечное множество правил вывода, называемых также продукциями;  $s$  ( $s \in N$ ) — начальный символ грамматики, один из множества нетерминальных символов, стартовый нетерминал.

Примем, что команды, задаваемые пользователем компьютеру,  $\Sigma = \{\alpha_j\}$ ,  $j=1, 2, \dots, m$ , — это терминальные символы некоторого языка. Нетерминальные символы из  $N$  изоморфны квантам деятельности  $\{\delta_k\}$ :  $|N|=|\{\delta_k\}|$ . Начальный символ (команда)  $s$  создаваемой нами формальной грамматики вводится для отображения начала очередной последовательности символов. Правило вывода в нашей грамматике определяет, к какому кванту деятельности относится конкретная последовательность терминальных символов (команд).

Кванты деятельности могут формироваться достаточно разнообразными последовательностями команд. Поэтому в соответствии с классификацией порождающих грамматик, введенной Н. Хомским [7], описываемый нами язык относится к контекстно-свободным (КС): может содержать рекурсии [6, 7]. Одним из методов распознавания квантов в последовательности символов, порожденных КС-грамматикой, являются автоматы со стековой памятью. Автомат со стековой памятью — это конечный автомат с запоминающим устройством типа стек.

Для выделения квантов деятельности в последовательности команд, фиксируемых компьютером, создадим конечный автомат  $A$  со стековой памятью [7]:  $A=(K, \Sigma, P, s, F)$ , где  $K$  — конечное множество возможных состояний автомата;  $P$  — функция перехода (таблица переходов) автомата, отображающая множество пар состояние-символ последовательности во множество символов;  $s$  — здесь начальное состояние  $s \in K$ ;  $F$  — интересующее нас множество состояний, связанных с квантами,  $F \subseteq K$ . В исходный момент автомат находится в состоянии  $s$ .

**4. Принцип распознавания кванта.** За основу берется алгоритм разбора, использующий LR таблицы. Однако язык, содержащий 50-100 терминалов и около 100 правил вывода, может иметь таблицу LR с несколькими сотнями состояний, а функция action может легко иметь 20 000 записей [8, 9].

Т.к. правил и терминалов может быть достаточно много и количество может возрастать, то надо оптимизировать алгоритм по затратам памяти и по скорости работы. Для этого вместо LR таблицы строится хэш таблица [11] по всем нетерминалам. А решение о свертке или переносе принимается в процессе чтения входной последовательности. Ключом хэш таблицы являются команды. В отличие от LR таблиц, перестроение хэш таблиц занимает меньше времени.

*Процедура выявления квантов деятельности.* Рассмотрим первый из трех автоматов КА-1. Он пропускает незначащие команды, читает первую команду и передает работу автомату КА-2, определяющему, какие кванты начинаются с текущей команды. Перемещаем команду в стек. После определения кванта, начинающегося с команды считанной КА-1, работа передается третьему автомату КА-3. КА-3 читает команды входной последовательности и помещает их в стек до тех пор, пока команды совпадут с соответствующими командами из выбранного на предыдущем шаге кванта. Если считанная команда не соответствует

команде в выбранном кванте, делаем свертку к предыдущему выбранному кванту. Сверткой называется снятие из стека определенного числа команд и помещение на их место кванта. Треугольниками на рис. 4-6 обозначены действия, предназначенные для смысловой обработки входного текста — семантические процедуры. Слово «процедура» употребляется в широком смысле, означая определенную последовательность действий. Для упрощения, понимая работы распознавателя, составим таблицу (табл. 1), в которой описаны состояния, символы, поступающие на вход, переходы между состояниями, возникновение ошибки в случае, если входной символ не равен ожидаемому, передача работы другому автомату, чтение символа из входной последовательности и номер семантической процедуры. Для сокращения занимаемого места «Да» и «Нет» заменено на «+» и «-», «Л» обозначает любой символ, «В» — команда «выбор объекта», «О» — символ конца текста, «Н» — незначащие команды.



Рис. 4. Синтаксическая диаграмма автомата КА-1 с семантическими процедурами, распознающими значимые команды

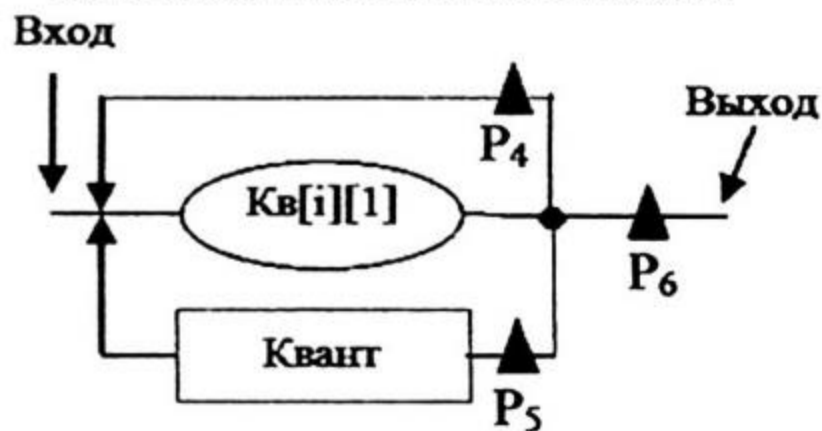


Рис. 5. Синтаксическая диаграмма автомата КА-2, с семантическими процедурами, фиксирующими кванты

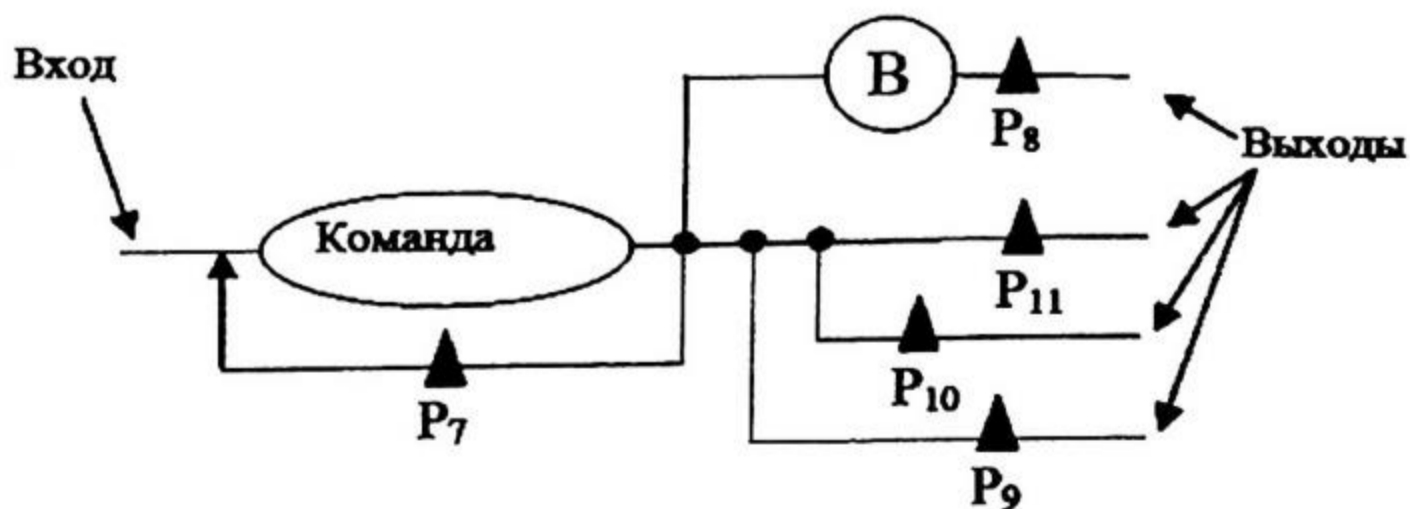


Рис. 6. Синтаксическая диаграмма автомата КА-3 с семантическими процедурами, распознающими кванты

Таблица 1

Таблица переходов распознавателя квантов

Состояние	Символ	Переход	Ошибка	Вызов	Читать	Процедура	Примечание
Анализ последовательности команд							
1	В	1	-	-	+	1	Пропуск незначимых команды
2	Л	8	-	+	-	2	Переход на кванты
3	Л	4	-	-	-		Начало цикла
4	В	4	-	-	+	1	Пропуск незначимых команды
5	Л	8	-	+	-	2	Переход на кванты
6	О, s	0	-	-	-	3	Выход
7	Л	3	-	-	-		Конец цикла
Фиксация последовательности квантов							
8	Л	9	-	-	+	4	Начало цикла. Поиск совпадения символа из последовательности с первым символом очередного кванта. Помещаем символ в стэк
9	Л	12	-	+	-	5	Переход на квант
10	О	0	-	-	-	6	Выход
11	Л	8	-	-	-		Конец цикл
Распознавание квантов							
12	Л	12	-	-	+	7	Считанный символ $\alpha$ совпадает с соответствующим символом выбранного кванта $\delta$ . Перемещаем $\alpha$ в стэк
13	В, Н	0	-	-	-	8	Последовательность не полностью соответствует выбранному кванту $\delta$ . Автомат заканчивает работу
14	Л	0	-	-	-	9	Считанный символ $\alpha$ не совпадает с соответствующим символом выбранного кванта $\delta$ . Автомат заканчивает работу
15	Л	0	-	-	-	10	Последовательность команд совпадает с последовательностью в выбранном кванте. Автомат делает свертку и заканчивает работу
16	О	0	-	-	-	11	Входная последовательность закончилась. Автомат заканчивает работу

Приведем пример правила, выделяющего квант деятельности  $\delta_1$  «ввод текста». Допустим, дана цепочка символов  $\alpha_1\alpha_1\alpha_2\alpha_2\alpha_2\alpha_3\alpha_2\alpha_1\alpha_3$ , где  $\alpha_1$  — выбор объекта, куда вводится текст;  $\alpha_2$  — ввод символа русского алфавита;  $\alpha_3$  — удаление предыдущего символа. Квант  $\delta_1$  может состоять из цепочки термина-

лов  $\alpha_2$  и  $\alpha_3$ . Причем первым может быть только  $\alpha_2$ . Записав это как правило вывода, получим  $\delta_1 \rightarrow \alpha_2 \delta_1 | \delta_1 \delta_1 | \alpha_2 | \alpha_3$ . Перед тем как начать распознавание, необходимо в начало строки подставить начальный символ  $s$ . Таким образом, строка примет вид  $s \alpha_1 \alpha_1 \alpha_2 \alpha_2 \alpha_2 \alpha_3 \alpha_2 \alpha_1 \alpha_3$ . Распознаватель закончит работу, как только на вход поступит  $s$ . Распознавание начинается с КА-1, в стек добавляется  $s$ , символы  $\alpha_1$  пропускаются. После считывания и помещения  $\alpha_2$  в стек работа передается КА-2. Выбираем первый квант, начинающийся с символа  $\alpha_2$ , это  $\delta_1$ . Запоминается индекс считанного символа. Работа передается КА-3. КА-3 считывает символы до тех пор, пока не закончится строка или не будет установлена принадлежность последовательности символов в стеке кванту. Символы  $\alpha_2$ ,  $\alpha_2$ ,  $\alpha_3$ ,  $\alpha_2$  входят в  $\delta_1$ , а  $\alpha_1$  нет. Т.к. все считанные символы входят в  $\delta_1$ , а значит квант установлен, то символы  $\alpha_2 \alpha_2 \alpha_2 \alpha_3 \alpha_2$  удаляются из стека и заменяются на  $\delta_1$ . Работа передается КА-1. КА-1 считывает символы, начиная с того, на котором закончил работу КА-3. Считывается символ  $\alpha_3$ , работа передается КА-2. Найти квант, начинающийся с  $\alpha_3$ , не удалось. Символ удаляется из стека. Работа передается КА-1. КА-1 считывает из входной последовательности начальный символ  $s$  и прекращает работу.

Рассмотренный пример показывает, что для точного распознавания необходимо иметь перечень (базу) корректных описаний квантов. Так, последовательность  $\alpha_1 \alpha_3$  — это квант «удаление объекта».

**Заключение.** С позиций теории формальных языков и грамматик, команды компьютера могут рассматриваться как некоторый алфавит. Над этим алфавитом создана грамматика, позволяющая посредством конечных автоматов выделять кванты деятельности в последовательности команд, исполняемых по инициативе пользователя компьютера.

Предложенный в работе алгоритм не может быть признан эффективным, т.к. требует полного перебора перечня описаний квантов. Кроме того, следует учесть, что команды в кванте могут меняться местами и/или не присутствовать. Так что для точного определения квантов необходимо в их перечне представлять все возможные перестановки символов, входящих в квант.

Для увеличения производительности целесообразно, в частности, накапливать статистику появления квантов определенной последовательности. Она позволит описать последовательность квантов Марковской цепью, что полезно при выборе очередного кванта для идентификации.

#### СПИСОК ЛИТЕРАТУРЫ

1. Каданцев М.В., Шапцев В.А. Автоформализация модели деятельности специалиста, работающего с компьютером // Математические методы в технике и технологиях. ММТТ-20. Т. 9 / Сб. трудов XX Междунар. науч. конф. Секция 7, 10 / Под ред. В.С. Балакирева. Ярославль: Изд-во Яросл. ГТУ, 2007. С. 134-137.
2. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. 2-е изд. / Пер. с англ. А.А. Слинкина. М.: ДМК Пресс, 2004. 432 с.
3. Каданцев М.В., Шапцев В.А. Мониторинг действий пользователя в среде MS Office // Информационные технологии и телекоммуникации в образовании, экономике и управлении регионом / Мат-лы межрег. н.-практ. конф. Тюмень. 1-30 ноября 2006 г. Тюмень: Изд-во ТюмГУ, 2007. С.37.
4. Словарь иностранных слов. 4-е изд. / Под. ред. И.В. Лехина и Ф.И. Петрова М.: ОГИЗ, 1947. 480 с.
5. Дж. Грофф, П. Вайберг. Энциклопедия SQL. 3-е изд. СПб.: Питер, 2004. 896 с.
6. Карпов Ю.Г. Теория автоматов СПб.: Питер, 2003. 208 с.

7. Свердлов С.З. Языки программирования и методы трансляции: Учебное пособие. СПб.: Питер, 2007. 638 с.
8. Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты. Пер. с англ. М.: ИД «Вильямс», 2003. 768 с.
9. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Том 1 Синтаксический анализ. М.: Мир, 1978. 611 с.
10. Крищенко В.А. Использование LR-таблиц для разбора ограниченного естественного языка // Электронный журнал «Исследовано в России» <http://zhurnal.apc.relam.ru/articles/958/2000/067.pdf>
11. Thomas H. Cormen. Introduction to algorithms. Massachusetts, MIT Press, 2001. 980 p.

*Александр Сергеевич ГИЛЬМАНОВ —  
аспирант кафедры программного обеспечения  
Тюменского государственного университета*

*Сергей Амирович ГИЛЬМАНОВ —  
профессор кафедры педагогики и психологии,  
Югорского государственного университета,  
доктор педагогических наук,  
(г. Ханты-Мансийск)*

УДК 004.057.5

## **МОДЕЛИРОВАНИЕ УЧЕБНЫХ ОБЪЕКТОВ ОБУЧАЮЩИХ СИСТЕМ, ИСПОЛЬЗУЮЩИХ ГИБРИДНЫЕ ТЕХНОЛОГИИ ДОСТАВКИ КОНТЕНТА**

*АННОТАЦИЯ. В данной статье рассмотрены подходы к информационному и математическому моделированию учебных объектов в электронных обучающих системах, использующих гибридные технологии доставки контента.*

*In this article we will show some basic approaches to mathematical and informational modeling of learning objects in e-learning systems that use hybrid technologies of content delivery.*

Электронные обучающие системы, использующие гибридные технологии доставки контента, являются относительно новым направлением в электронном образовании. Основным нововведением данной концепции является, во-первых, возможность использования для передачи образовательного контента не только стандартных путей доставки (персональные компьютеры, интернет, e-mail, локальные сети), но и любых других доступных каналов связи и устройств (мобильные телефоны, PDA-устройства, iPhone и всевозможные пути передачи данных — сотовые сети GSM/3G, wi-fi, Bluetooth, SMS и т. п.), а во-вторых, полная или частичная автоматизация подбора образовательного контента для передачи пользователю.

Цель данной статьи — обозначить один из возможных подходов к моделированию учебных объектов при применении гибридных технологий доставки образовательного контента, т. к. существующие методы не всегда способны обеспечить полноценное их использование.