

в информационных биосистемах позволяет, с одной стороны, прогнозировать терапевтические эффекты различных воздействий, а с другой — достоверно моделировать процессы переноса электрических импульсов в механизмах биоуправления.

Библиографический список

1. Владимирев В.В., Рощупкин Д.И., Потапенко А.Я., Деев А.И. Биофизика. М.: Медицина. 1983. 272 с.
2. Волькенштейн М.В. Биофизика. М.: Наука, 1981. 576 с.
3. Медицинская биофизика. /Под ред. В.О.Самойлова. Л.: Воениздат. 1986. 480 с.
4. Стивенс Ч. Моэг. М.: Мир, 1984. С. 31-57.
5. *Hodgkin A., Huxley A. // J. Physiol, 1952, 117, p. 500-544.*

УДК 378:371.315.7

А.Д.Кочелев, Л.В.Туманова

ВОПРОСЫ МЕТОДИЧЕСКОЙ И СТРУКТУРНОЙ ОРГАНИЗАЦИИ АВТОМАТИЗИРОВАННЫХ УЧЕБНЫХ КУРСОВ

(г. Москва)

Одним из важнейших принципов, лежащих в основе создания автоматизированных курсов (АУК), является обеспечение максимальной активности и свободы обучаемого в процессе ведения диалога. Опыт применения автоматизированных обучающих систем (АОС) показывает, что обучение в диалоге, при котором свобода обучаемого сведена к минимуму (обучаемый не имеет возможности по своей инициативе вернуться к пройденному материалу, запросить у ЭВМ помощь (подсказку, наводящие соображения), консультацию, задать ЭВМ встречный вопрос и т.д.), малоэффективно.

Важным частным средством повышения активности обучаемого является реализация в учебном курсе сочетания свободного (управляемого обучаемым) и программно-управляемого обучения, что дает возможность обучаемому, с одной стороны, выбрать собственную стратегию изучения

курса, а с другой – следовать оптимальной (с точки зрения автора) стратегии. Реализация данного принципа предопределяет структуру учебного курса (типа "меню") и требования к возможностям обучаемо-го управлять последовательностью прохождения курса: выбрать по собственному усмотрению тему из списка – "меню", "листать" кадры темы как в прямом, так и в обратном порядке, возвращаться из любого места курса к списку тем, в произвольном месте заканчивать работу с курсом и т.д.

Другой важный принцип создания АУКа касается его структурной организации.

Автоматизированный учебный курс, рассмотренный как программный продукт, обладает следующими характерными чертами:

имеет, как правило, большой объем и сложную структуру ;

подвергается корректировке и модификациям не только на этапе отладок, но и в процессе эксплуатации (расширение курса, изменение последовательности изложения, замена задач и т.д.). Эти черты делают особо актуальным применение при написании и отладке обучающей программы принципов структурного подхода: модульность, программирование и отладка курса "сверху вниз", прозрачность и наглядность его структуры и др., обеспечивающих быструю и параллельную отладку отдельных модулей и получение легко модифицируемой программы.

На уровне сегментов (крупных программных блоков, реализующих обычно самостоятельные темы обучающего курса) модульность реализуется благодаря возможности их фактического написания независимо друг от друга, как отдельных мини-курсов, что обеспечивается сервисными средствами пакета прикладных программ и автоматизированных обучающих систем (ППП АОС) вуз. Эта возможность, однако, позволяет добиться модульности курса и параллельной отладки его частей лишь на самом общем, сегментном уровне структуризации, опираясь на естественную информационную независимость (или слабую зависимость) отдельных сегментов (тем) курса. Между тем, наиболее важным является реализация модульного принципа в рамках сегмента. Минимальной программной единицей, которая может быть оформлена в виде самостоятельного модуля, является проблема – фрагмент программы, обеспечивающий единичное взаимодействие в процессе обучения между обучающим и системой типа: вопрос обучаемому – ответ обучаемого – реплика системы на ответ обучаемого.

Организация модульной структуры сегмента предполагает наличие специальных языковых средств. В базовом языке обучающихся курсов (ЯОК) ПШП АОС вуз эти средства были слабо развиты. Однако в новой версии ЯОК/2 языка (см. /1/) они существенно обогатились. Во-первых, были введены два новых оператора GL <метка> - вызов процедуры <метка> и RV - возврат из процедуры. Оператор GL выполняет переход на метку начала процедуры и запоминает текущее состояние проблемы. Оператор RV обеспечивает восстановление того текущего состояния проблемы, которое соответствует последнему выполняемому оператору вызова и осуществляет возврат к соответствующей точке данной или следующей проблемы. Точка возврата определяется аргументом оператора. Отсутствие этих операторов в базовом ЯОК приводило к необходимости моделировать процедуру (вводить дополнительные метки - точки возврата, запоминать каждый раз точки выхода из проблемы и т.д.), что часто элиминировало преимущества организации процедуры.

Во-вторых, в ЯОК были включены средства определения так называемых команд обучаемого, реализующих встречный вызов. Дело в том, что в процессе обучения у обучаемого часто возникает необходимость осуществить стручный вызов - обратиться к системе за справочной информацией, необходимой для ответа на поставленный вопрос, получить доступ к вычислительным ресурсам и т.п. В базовом ЯОК возможность встречного вызова необходимо программировать в каждой проблеме, где такой вызов разрешен. При этом возникают те же трудности, что и при моделировании процедур. Команда обучаемого в простейшем случае имеет вид \langle <метка> . Если обучаемый вместо ответа на вопрос введет такую команду, она не будет анализироваться операторами анализа ответа. Вместо этого вызывается и выполняется процедура с меткой \langle <метка> , а затем оператор RV процедуры возвращает обучаемого к ответу на вопрос.

Рассмотренные средства представляют довольно широкие возможности для организации сегмента курса с модульной структурой. Ниже мы рассмотрим два типа таких структур, реализующих обучение в режиме "меню", при котором обучаемый сам может выбирать последовательность изучения курса.

Первый тип модульной структуры, реализующий режим "меню", можно проиллюстрировать на примере организации обучающего курса НО-

ВОСТИ (автор Д.В.Васильев), содержащего описание новых возможностей ППП АОС вуз, не отраженных в текущей документации системы. Курс НОВОСТИ состоит из оглавления и ряда процедур встречного вызова, соответствующих параграфам оглавления. Метки процедур имеют вид @N, где N - номер параграфа в оглавлении. Обучаемый, ознакомившись с оглавлением курса, может с помощью команды встречного вызова @N получить на экране информацию заинтересовавшего его параграфа, после изучения которой он вновь возвращается (оператором RN соответствующей процедуры) к оглавлению и имеет возможность осуществить новый вызов.

Главным достоинством рассмотренной структуры является простота и легкость модификации (для включения/исключения процедуры достаточно добавить/удалить текст процедуры и соответствующий параграф в оглавлении). В то же время такую структуру нельзя признать универсальной. Наиболее удобно использовать ее при организации справочного курса (каким и является курс НОВОСТИ), в котором соседние параграфы, вообще говоря, содержательно не связаны друг с другом. В тех же случаях, когда последовательность изучения курса важна, такую структуру использовать нецелесообразно, поскольку в ней реализована лишь первая часть принципа сочетания свободно выбираемой и программно управляемой последовательности изучения материала, а именно: каждый очередной параграф для изучения выбирает сам обучаемый. В тех же случаях, когда обучаемый не знаком с курсом и не может самостоятельно выбирать оптимальную последовательность его прохождения, необходимо, чтобы эта последовательность организовывалась программно.

Оптимальной для реализации указанного принципа в режиме "меню" является модульная структура, в которой четко разделены управляющая часть, осуществляющая в соответствующем порядке вызов процедур (параграфов) курса, и процедурная часть - набор вызываемых процедур, обеспечивающих изучение отдельных параграфов курса.

Проиллюстрируем этот тип структуры на примере программы PRPLUK - АУКа по алгоритмическому языку ПЛ-I.

Программа PRPLUK написана в соответствии с принципами структурного подхода. Она имеет модульную структуру и содержит процедуры трех типов:

рабочие процедуры, каждая из которых обеспечивает изучение в режиме диалога самостоятельного курса;

головная процедура, обеспечивающая выдачу на экран "меню" — оглавление разделов курса и осуществляющая в соответствующем порядке (стандартном, заданном программно или индивидуальном, задаваемым обучаемым) вызов рабочих процедур;

служебные процедуры ОТВЕТ и ПОМОЩЬ, первая из которых обеспечивает вывод в фиксированное окно на экране текущей информации о других возможных (кроме ответа на вопрос системы) ответах обучаемого (вернуться на : шаг назад, продвигнуться на шаг вперед, обратиться к помощи и вернуться к оглавлению курса), а вторая реализует оказание обучаемому помощи (по его запросу) — выдачу на экран видеотерминала определений, рассматривавшихся ранее (но забытых обучаемым) понятий и терминалов.

Рабочая процедура состоит из последовательности и вопросных проблем, отражающих содержание соответствующего раздела курса. Каждая проблема реализует элементарное взаимодействие обучаемого с системой. Информационная проблема выводит на экран очередную порцию теоретического материала; работа заканчивается, когда обучаемый вводит команду смены кадра. Вопросная проблема высвечивает вопрос по изучаемой теме; работа обычно заканчивается, когда обучаемый введет правильный ответ. В этом случае она выводит на экран комментарий к ответу, после чего управление передается следующей проблеме рабочей процедуры или возвращается в головную процедуру, если закончилась последняя проблема очередной рабочей процедуры.

Отметим некоторые специфические черты программной организации рабочей процедуры. После каждого оператора, реализующего выдачу вопросного кадра, следует оператор вызова служебной процедуры ОТВЕТ. Это обеспечивает появление в окне экрана информации о других возможных ответах обучаемого лишь в те моменты диалога, когда эта информация действительно может пригодиться обучаемому. При анализе каждого ответа обучаемого на поставленный системой вопрос проверяются также и ответы, обозначающие разрешенные встречные запросы обучаемого (о помощи, возврате назад и продвижении вперед), что обеспечивает постоянную готовность процедуры обслужить запрос обучаемого. При организации вопросных проблем используется датчик случайных расстановок, позволяющий случайным образом выбирать и выводить на экран конкретное вопросное задание из пронумерованного набора однотипных вопросных заданий. Этот прием позволяет из-

Ожидать высвечивания одних и тех же вопросов при возврате к данной рабочей процедуре.

Каждая информационная проблема организована как самостоятельная процедура. Это дает возможность выводить на экран данный информационный кадр и из других точек программы, обращаясь к информационной проблеме как к процедуре. Эта возможность использована при организации процедуры ПОМОЩЬ. Таким образом удастся избежать дублирования информации, что не только экономит память, но и облегчает модификацию информационных кадров.

Служебная процедура ПОМОЩЬ высвечивает на экране "меню" - список терминов и понятий, определения которых обучаемый может получить, и приглашение ввести нужный термин или слово НЕТ. После ввода термина система высвечивает на экране его определение (если, конечно, термин введен правильно, в противном случае система выдает на экран сообщение об ошибке и вновь высвечивает "меню" и приглашение). Изучив определение термина, обучаемый вводит соответствующую команду (любой символ), после чего перед ним вновь высвечивается "меню" и приглашение вновь ввести термин. Если обучаемый не имеет более потребности использовать процедуру ПОМОЩЬ, он должен ввести слово НЕТ, после чего работа процедуры заканчивается и управление передается в точку вызова процедуры ПОМОЩЬ. Программно высвечивание на экране запрошенного обучаемым термина реализуется посредством вызова соответствующей процедуры (информационной проблемы) из рабочей процедуры.

Служебная процедура ОТВЕТ представляет собой одну информационную проблему, которая высвечивает в фиксированном окне экрана информацию о дополнительных возможных ответах - встречных вопросах обучаемого.

По окончании выполнения рабочей процедуры (изучен последний раздел курса), а также по специальному встречному запросу обучаемого, перед ним вновь высвечивается "меню" - оглавление разделов курса и приглашение выбрать (для изучения) очередной раздел курса (ввести его номер) или закончить работу с программой *PRPLUK* (ввести команду *SOF*, отключающую видеотерминал обучаемого от данного курса).

Реализация АУК в виде программы с модульной структурой обладает рядом достоинств. Во-первых, обеспечивается возможность более быстрого написания программы, поскольку разные модули могут писать-

ся одновременно разными программистами. Во-вторых, резко облегчается модификация и, в частности, наращивание программы, что для обучающего курса весьма актуально. И наконец, в третьих, структура программы становится весьма прозрачной и наглядной: сначала идет головная процедура, отражающая в укрупненном виде (в терминах вызываемых рабочих процедур) логику работы программы, а затем уже сами рабочие процедуры.

Подчеркнем необходимость строгого функционального различения команд встречного вызова обучаемого $\langle \text{метка} \rangle$ и команд безусловного перехода типа M *метка*. Команды первого типа предполагают обязательный возврат в точку вызова. При их использовании начинает накапливаться глубина вызова процедур, поэтому длина начатого ими самостоятельного диалога теоретически не должна превышать 7 "шагов" (вызов процедур). Команды безусловного перехода допускают любое количество "шагов" самостоятельного диалога, но не обеспечивают автоматического возврата назад. Такой возврат (если он необходим) должен осуществить сам обучаемый другой безусловной командой M *метка*.

К недостаткам (вполне допустимым) предложенной структуры следует отнести необходимость использования в курсе удвоенного числа меток. Для обеспечения такой возможности следует при регистрации курса в явном виде заказывать 1 блок меток на 15 блоков, а не на 25, как это принято по умолчанию. Кроме того, для работы операторов CL и AN при регистрации курса необходимо задать как минимум два блока вспомогательной памяти.

Библиографический список

1. Васильев А.В. Развитие языка ЯОС АОС ВУЗ // Исследование и применение АОС в учебном процессе. М., 1985.

О.И.Косолапов, М.А.Чистякова

О НЕКОТОРЫХ ВАЖНЕЙШИХ ПРИНЦИПАХ СОЗДАНИЯ
АВТОМАТИЗИРОВАННЫХ УЧЕБНЫХ КУРСОВ И ИХ РЕАЛИЗАЦИИ

(Г. Москва)

Наличие широкого спектра вычислительной техники и ее относительная доступность для вузов делают актуальной проблему рационального использования ЭВМ в учебном процессе. Из публикаций и выставок известно множество самых различных систем, пакетов прикладных программ, автоматизированных курсов и т.д., так или иначе использующих возможности ЭВМ в организации учебного процесса и автоматизирующих различные аспекты задачи обучения.

Общепринятой методологии автоматизации учебного процесса к настоящему времени не сложилось, поэтому представляется целесообразным отметить некоторые принципы, способные внести конкретный вклад в разработку автоматизированных систем обучения. Прежде всего необходимо учитывать, что современные требования к специалисту, готовящемуся в вузе, формулируются в терминах "умений".

Следует учитывать, что воспитание важнейших интеллектуальных и профессиональных умений требует для каждого конкретного научного направления большой методологической проработки и в настоящее время не завершено. Однако можно сформулировать общие условия, выполнение которых необходимо обеспечить во всех случаях. Видимо, для воспитания и выработки требуемых умений необходимо организовать некоторую деятельность обучаемого в определенной среде. При этом, чем ближе учебная среда к среде профессиональной деятельности будущих специалистов, тем больше возможность приобретения в процессе обучения адекватных полноценных профессиональных умений.

Отсюда следует вывод, имеющий общий методологический характер. При использовании ВТ в учебном процессе огромные возможности современных вычислительных систем целесообразно употребить в первую очередь для моделирования среды обучения, адекватной профессиональной творческой среде, в которой готовящийся специалист будет работать.

Анализ лучших систем, использующих ЭВМ в учебном процессе (в ЛЭТИ, ЛГУ, МГУ, КуАИ и других вузах), позволяет убедиться в целесообразности упомянутого выше принципа.