

Э. К. Шахов, А. И. Надеев, В. В. Акинин, Е. М. Голобокова

Разработка мультимедийных учебников: эффективность и «подводные камни»

Фирма Macromedia широко известна такими программными продуктами, как Macromedia Flash, Macromedia DreamWeaver. Однако в России практически неизвестен такой продукт фирмы, как Macromedia Authorware, пакет для создания мультимедийных обучающих средств. Ему посвящена монография [1], в которой даны азы использования компонентов пакета, однако многие важные тонкости разработки не нашли достаточного отражения. Попробуем восполнить этот пробел.

Как было отмечено, пакет Macromedia Authorware предназначен для создания мультимедийных обучающих средств, т. е. электронных учебников. Он предоставляет богатейшие возможности для организации процесса обучения одиночных учеников по так называемой кейс-технологии (когда заочно обучающемуся ученику высылаются кейсы со всеми учебными материалами), а также студентов очной формы обучения, дополняя лекционный материал. Для раскрытия всех возможностей пакета Macromedia Authorware рассмотрим абстрактный пример с созданием электронного учебника при помощи данного пакета.

При запуске Macromedia Authorware вам будет предложено воспользоваться одним из мастеров для создания шаблона будущего учебника. Однако встроенные мастера Authorware являются не самым удачным выбором для начала изучения возможностей Authorware из-за своей перегруженности. Поэтому лучше отказаться от использования мастеров, нажав кнопку «None». После этого будет создан пустой проект Authorware, представленный в виде Flowline (первоначально не содержащей элементов). Flowline является одним из ос-

новных понятий Authorware, она определяет всю идеологию данного пакета. Flowline представляет собой «нить», на которую «нанизываются» все графические, интерактивные, мультимедийные и прочие объекты, реализующие элементы учебника. При запуске проекта элементы просматриваются в естественном порядке: сверху-вниз, слева-направо, если явно не указан иной порядок прохождения.

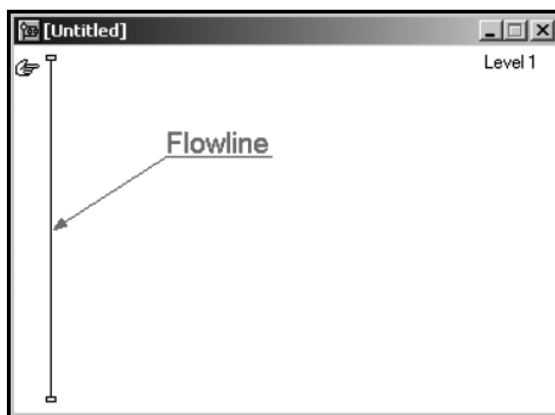


Рис. 1. Пустой проект

Рассмотрим подробнее процесс создания несложного учебника в среде Authorware. Заранее определимся: при создании учебника будем отталкиваться от структуры учебника реального. Так, обычный учебник начинается с обложки — с нее и начнем. Необходимо выбрать размеры экранной формы учебника (по умолчанию — 640 на 480 пикселей) или указать собственные размеры, а также заголовок для формы. При этом указанный заголовок в режиме разработки не отображается, но после компиляции проекта он появляется.

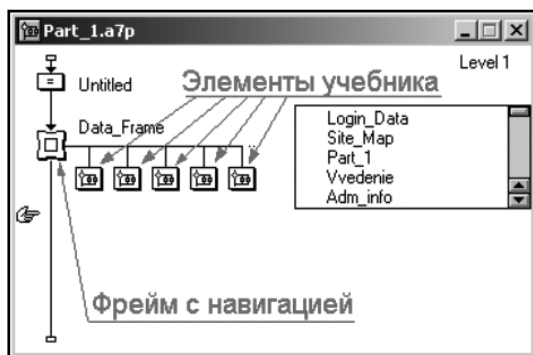


Рис. 2. Пример общей структуры учебника

Сразу необходимо заметить: если использовать элементы типа Framework, то все графические и интерактивные элементы, размещенные в нем, отображаются на всех страницах, принадлежащих фрейму. Этим свойством фрейма стоит воспользоваться для того, чтобы создать единое оформление всего учебника. Для этого достаточно сразу разместить на Flowline фрейм, а все «страницы» учебника присоединять уже к нему. Кроме единого оформления, это позволит, создав один раз систему навигации и управления процессом обучения, распространить ее на весь учебник — достаточно просто поместить ее внутри фрейма.

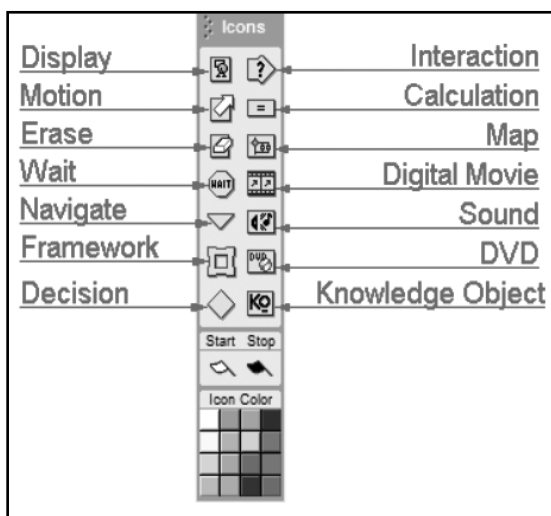


Рис. 3. Панель инструментов

Первая проблема, с которой мы можем столкнуться, — это осуществление навигации по урокам, как последовательной, так и выборочной (т. е. создание оглавления). Можно, конечно, представить оглавление учебника в виде последовательности страниц (элементов типа Display), связанных между собой гиперссылками, создав структуру наподобие интернет-сайта. Но в данном случае будет очень сложно следить за ссылочной целостностью оглавления. Кроме того, при частом изменении структуры учебника придется заново создавать все оглавление или его часть. Поэтому стоит поискать более легкий, наглядный и функциональный способ навигации.

Одним из наиболее эффективных способов создания оглавления является использование такого элемента WinControl, как TreeView, который включает все необходимые нам функции. Он позволяет создавать иерархическое меню с высокой степенью вложенности, что нам и требуется. Особенностью данного компонента является то, что он, как и другие элементы типа WinControl, создается уже на этапе работы приложения. Его создание, настройка и активизация производятся в иконке типа Calculation (контейнер с командами встроенного языка программирования).

Другой особенностью, которая часто отпугивает неопытных пользователей, является то, что для корректной работы средств навигации необходимо сразу продумать и создать внутреннюю структуру учебника в максимально полном объеме (кардинальная смена структуры учебника в дальнейшем будет довольно сильно затруднена). Кроме того, существует несколько особенностей работы с TreeView. Первая особенность: для программиста элемент TreeView представляет собой два списка, составленных по определенным правилам:

- 1) список названий глав;
- 2) список адресов начала глав.

Естественно, для работы оглавления необходимо, чтобы каждая глава имела собственный уникальный идентификатор (метку). В то же время такая организация меню в дальнейшем позволит легко создать средства последовательной навигации по учебнику.

Вторая особенность элемента TreeView: невозможность указать напрямую адрес перехода для самого 1-го пункта оглавления. Для этого придется пользоваться обходными путями.

Приведем описание этапов создания простейшего меню.

Этап 1. Описание глав меню

```
P1:= "Главный раздел 1"
P1_1:= " ^" Подраздел 1.1"
P1_2:= " ^" Подраздел 1.2"
P1_Full:=P1^Return^P1_1^Return^P1_2
P2:= "Главный раздел 2"
P2_1:= " ^" Подраздел 2.1"
P2_2:= " ^" Подраздел 2.2"
P2_Full:=P2^Return^P2_1^Return^P2_2
myItems:= P1_Full^Return^P2_Full
```

Этап 2. Описание идентификаторов разделов меню

```
A1_1:= IconID ("Part_1_1")
A1_2:= IconID ("Part_1_2")
A1_Full:= A1_1^Return^A1_2
A2:= ""
A2_1:= IconID ("Part_2_1")
A2_2:= IconID ("Part_2_2")
A2_Full:=A2^Return^A2_1^Return^A2_2
myItemsID:= A1_Full^Return^A2_Full
```

(В данном примере Part_1_1, Part_1_2 и т.д. — названия элементов, содержащих уроки для глав 1.1, 1.2 и т.д.).

При этом уровень вложенности того или иного пункта меню определяется количеством пробелов в начале строки.

Этап 3. Создание главного меню

```
tvw_01:= wcDisplayControl (20; 15; 760; 500;
"TreeView"; "tvw_chg_01")
wcSetPropertyValue (tvw_01; "Items"; myItems)
wcSetPropertyValue (tvw_01; "HideSelection";
"False")
```

Как показывает практика, не все гладко и при работе пользователя с таким оглавлением. Например, если перемещаться по оглавлению при помощи курсорных клавиш, то при наведении курсора на главу, не имеющую подразделов, программа тут же осуществляет переход на урок, не дождавшись команды перехода по ссылке (например, клавиши Enter). Поэтому для корректной работы необходимо либо блокировать опрос курсорных клавиш, либо перехватывать их нажатие.

Итак, мы успешно создали оглавление учебника. Пора переходить к его наполнению. Для этого в AuthorWare имеется большой набор различных изобразительных средств, вплоть до возможности импорта звуковых фрагментов и видеороликов в различных форматах (7-я версия, последняя на данный момент, позволяет даже работу с DVD). В общем виде каждая глава учебника может быть представлена в виде страниц (иконок типа Display), связанных между собой при помощи элементов Interaction (ожидание реакции пользователя), Wait (задержка на некоторое время) или «нанизанных» на звуковой или видеофрагмент. Также существует возможность вставки анимации в виде файлов формата GIF и Macromedia Flash, компонентов ActiveX. При этом AuthorWare позволяет синхронизировать процесс отображения информации на экране со звуковым сопровождением, поясняющим эту информацию.

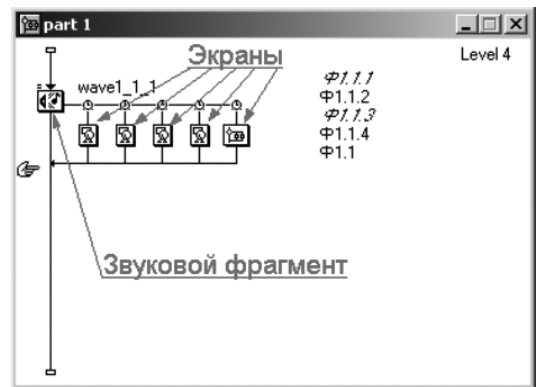


Рис. 4. Пример общей структуры учебника

Однако здесь тоже имеются свои подводные камни. Дело в том, что для синхронизации AuthorWare использует свой таймер, обладающий рядом недостатков, а использование системного таймера крайне затруднено. Поэтому желательно, чтобы между двумя последовательными событиями был временной интервал 50–100 миллисекунд, в противном случае может быть нарушен порядок отображения информации. Также желательно, чтобы все события (появление новой страницы на экране, ее стирание, движение каких-либо элементов) были синхронизированы со звуком не по времени, прошедшему с начала воспроизведения звукового фрагмента (опция Synchronize on seconds), а по времени воспроизведения собственно звукового фрагмента (Synchronize on position). Это позволит реализовать возможность постановки воспроизведения звукового фрагмента на «паузу» и затем безболезненно его возобновить (при расстановке событий по времени, прошедшему с начала воспроизведения, остановка звука никак не влияет на процесс воспроизведения анимации, и она продолжает идти «в немую»). Очень важно также располагать все события только на линии воспроизведения звука за исключением, возможно, последней страницы.

Группировка событий на линии воспроизведения звука (горизонтальная линия от иконки звука) в иконки-контейнеры типа Map Icon также не желательна, так как в этом случае при остановке звука последовательное выполнение команд (событий) остановится лишь тогда, когда указатель дойдет до конца группы, что также повлечет за собой несколько секунд «немой сцены».

Authorware позволяет создавать библиотеки ресурсов, что значительно сокращает физический объем учебника в случае, если в нем присутствует много идентичных элементов.

В связи с тем что информационное наполнение учебника, как правило, пред-

ставляет собой набор сменяющихся друг друга текстовых фрагментов и поясняющих изображений, сопровождаемых или не сопровождаемых звуковым оформлением (другие варианты потребуют либо гораздо больших аппаратных ресурсов для просмотра учебника, либо неоправданно больших затрат времени на его создание), рассмотрим подробнее работу с изображениями. Authorware декларирует возможность обработки графических файлов (изменение размеров изображения, отображение только части изображения). Однако с точки зрения быстродействия, а также качества результата лучше всю обработку изображений выполнять заранее в специализированных программах, а в Authorware вставлять изображения в режиме As is.

Следующий момент, на котором следует заострить внимание, — это возможность использования в учебнике стандартных функций Windows, хранящихся в файлах динамических библиотек Windows. Хотя в Authorware содержится много разных встроенных функций, иногда их бывает недостаточно, и приходится пользоваться функциями WinAPI.

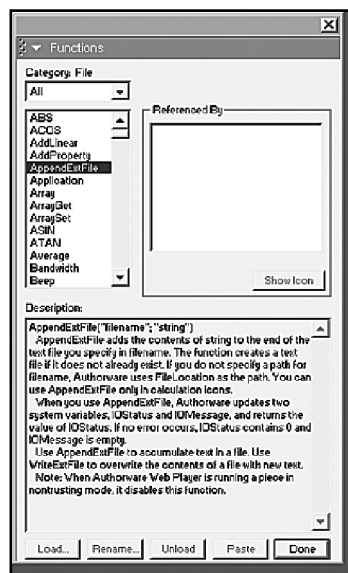


Рис. 5. Встроенные функции

Для этих целей в AuthorWare предусмотрен мастер импортирования функций из динамически подгружаемых библиотек (dll-файлов). Для импортирования функции следует включить окно Functions, затем выбрать категорию с именем нашего учебника. В нижней части окна станет активной кнопка Load. После ее нажатия будет предложено выбрать файл динамической библиотеки, в которой находится необходимая функция. В следующем окне Non-Authorware dll — имя функции, типы ее параметров и тип возвращаемого значения. К сожалению, типы параметров придется либо угадывать, либо смотреть в каких-либо справочных материалах — Authorware справок не дает. Также есть возможность здесь же заполнить поле описания функции или примечаний к ней (поле Description). После нажатия кнопки Load функция станет доступной во всем учебнике.

Например, чтобы сделать неактивной кнопку закрытия окна в заголовке, но сам заголовок оставить, можно выполнить довольно простой код, реализуемый при загрузке учебника:

DeleteMenu (GetSystemMenu (WindowHandle; 0); 61536; 0), где курсивом выделены импортированные функции.

AuthorWare также является эффективным средством создания тестов для проверки уровня усвоения изучаемого материала. Самый простой способ создания такого теста — использование специального мастера (Quiz Wizard), который вы без труда пройдете, шагая по этапам. В данном случае мастер сам создаст и настроит все необходимые ему структуры данных и обеспечит взаимодействие с базой данных, которая будет использоваться для записи результатов тестирования.

Возможны следующие варианты баз данных:

1. Text file — запись ведется в текстовые файлы. Для каждого пользователя создаются отдельные файлы, карта которых хранится в главном файле.

2. ODBC database — интерфейс ODBC, позволяющий работать с любым типом баз данных, для которых существуют источники ODBC под Windows. Исходя из назначения учебников, создаваемых при помощи Authorware, наиболее удобным будет использование либо баз данных MS Access (для одиночных пользователей и небольших классов), либо MS SQL Server или MySQL (для больших групп обучающихся).

AICC Compliant LMS — средства для работы с системами управления обучением (наподобие Lotus Learning Space). Честно говоря, присутствие такой возможности не является необходимостью: трудно представить, что кто-нибудь будет использовать Authorware совместно с системами управления обучением.

Мастер настроит и внешний вид, и функциональные возможности тестовой программы (присутствие или отсутствие окна регистрации пользователя, окна вывода результатов, максимальное количество попыток ввода пароля пользователя и т.д.).

Опять пойдем не совсем стандартным путем. Наша цель — создать систему проверки усвоенного материала, где будет не один огромный тест по результатам всего пройденного материала, а множество небольших тестов, раскиданных по всему учебнику (например, после каждой главы). При этом идентификация обучаемого должна проводиться один раз — при входе в учебник. Кроме контроля за успеваемостью, это позволит отслеживать, как продвигается ход обучения. Однако для этого потребуется самим разработать структуру базы данных, написать код, выполняющий подключение к базе данных при каждом входе в обучающий комплекс и сброс данных в базу, например, при помощи SQL-запросов.

Для создания тестов в AuthorWare 7.0 предоставлен набор из 7 видов вопросов. Опишем их кратко:

1. Drag-Drop Question — в вопросе нужно правильно расставить элементы на свои места при помощи «мыши».

2. Hot Object Question — ответом на вопрос считается выбор одного из представленных изображений.

3. Hot Spot Question — на изображении нужно «мышкой» выбрать ту область, которая, как вам кажется, является ответом на поставленный вопрос.

4. Multiple Choice Questions — из представленных текстовых ответов на вопрос правильными могут оказаться сразу несколько.

5. Single Choice Question — ответ только один.

6. Short Answer Question — ответ нужно ввести при помощи клавиатуры.

7. True-False Question — вы можете либо согласиться, либо не согласиться с утверждением, приведенным в вопросе.

Поскольку вопросов в нашем учебнике может быть много, при этом, как показывает практика, до 90% вопросов относятся к типу либо Multiple Choice Question, либо Single Choice, который при желании сводится к предыдущему типу, целесообразным будет как-либо автоматизировать процесс создания тестов. Для этого можно воспользоваться импортом XML-файлов с описанием вопросов и ответов теста в «Мастере вопросов».

В качестве примера рассмотрим программу XML Creator, написанную специально для решения подобной задачи. С ее помощью можно автоматически конвертировать текст из буфера обмена в структу-

ру XML-файла. В результате работы программы получается готовый XML-файл, который необходимо только импортировать в AuthorWare.

Программа проста в управлении. Например для вставки вопроса достаточно нажать одну клавишу (F4), и вопрос из текстового представления в буфере обмена преобразуется в самостоятельный XML-файл.

Структура XML-файлов формируется полностью программой и не нуждается в подробном рассмотрении. Достаточно сказать, что эти файлы содержат как данные, так и логику. В рассматриваемом примере описываются все варианты хитросплетений ответов на вопрос. В результате при прохождении этого вопроса вы можете получить либо положительную оценку, либо отрицательную, либо сообщение о том, что не все правильные варианты ответов были выбраны. Как мы убедились, данный тип вопросов наиболее сложный в описании при помощи XML-технологии по сравнению с остальными вопросами, где нужно определить лишь один правильный ответ, либо сравнить введенный с шаблоном, либо проверить, все ли объекты на своем месте.

Необходимость написания подобной программы появилась после того, как было обнаружено одно очень неудобное свойство мастеров Knowledge Objects. В их поля ввода, например, текстов вопроса и ответов нельзя вставлять данные из буфера обмена, что соответственно предполагает ручной перенос всех вопросов, а это огромная трата времени.

Структура получаемых XML-файлов на первый взгляд сложна. Однако разобраться не так уж и сложно. Файл содержит экземпляр класса вопроса со своими вложенными подклассами; у них, в свою очередь, имеются свойства, которые мы, по сути, и заполняем: текст вопроса, тексты ответов, подсказок к вопросу, баллы за каждый ответ. Достаточно легко в файле можно выделить сегменты вариантов ответов, оценок за ответы, текстов подсказок, логики и подсчета результатов. В данном случае, однако,

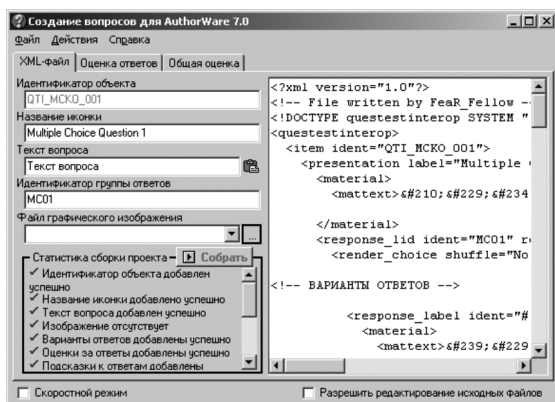


Рис. 6. Главное окно программы XML Creator

русские символы для заполнения полей не подходят, нужно использовать их коды (программа XML Creator это делает за пользователя), а спецсимволы лучше вообще исключить из текстов, иначе при импортировании файла в Authorware возникает исключительная ситуация. Об ошибке в XML-файле сообщается с точностью до номеров строки и символа в этой строке.

Очень важная особенность при импортировании вопросов состоит в том, что в Authorware 7.0 используется два типа мастеров вопросов множественного выбора: Multiple Choice Question 6.0 и Multiple Choice Question 6.5. Если тест создавался при помощи мастера проектов Quiz или Application, то в нем используются мастера вопросов версии 6.0, которые не имеют возможности импортирования XML-файлов. Поэтому следует внимательно относиться к тому, что вы используете в качестве «кирпичиков», из которых строится учебник.

Допустим, мы хотим проводить тестирование обучаемого после ознакомления с каждой главой учебника. Для этого потребуется один раз при входе авторизоваться, а при прохождении каждого теста — записывать в базу данных результат. Но обо всем по порядку.

Для входа в учебник в качестве базы будем использовать стандартный пример регистрации пользователя, предоставляемый разработчиками пакета AuthorWare 7.0. Внешний вид нашего окна регистрации можно настроить как угодно — например, так, как приведено на рис. 7.

Самая главная часть данной процедуры — открытие базы данных, посылка нескольких запросов, анализ полученных данных и закрытие базы. Все это происходит после ввода пользователем своих фамилии, имени и отчества. Можно ничего не менять в исходных кодах процедуры регистрации, однако, если будет необходимо, например, защитить базу данных паролем, придется добавить несколько строк для того, чтобы наш учебник мог работать с защищенной

базой. Мы будем работать с источником данных ODBC, так как это наиболее удобный в нашем случае вариант. Обмен данными будет осуществляться путем составления и выполнения SQL-запросов.

Рис. 7. Форма регистрации

Для правильного функционирования нашего учебника в плане ведения журнала прохождения тестов после регистрации необходимо сгруппировать все данные о текущем пользователе в одну переменную, которая в дальнейшем понадобится для записи результатов тестирования. Переменная, которую использует Knowledge Object Scoring для установления связи с базой данных и добавления результатов определенному пользователю, называется wzLoginData. Заполнить ее можно, например, так:

```
wzLoginData [#StorageType]:= #ODBC
//Тип базы данных
wzLoginData [#FirstName]:= uFirstName
//Фамилия
wzLoginData [#MiddleName]:= uMiddleName
//Имя
wzLoginData [#LastName]:= uLastName
//Отчество
wzLoginData [#UserID]:= LogonUserID
//Идентификатор
wzLoginData [#Password]:= LogonPassword[1]
//Пароль пользователя
wzLoginData [#LoginTime]:= SessionTime
//Время текущего сеанса
wzLoginData [#ODBCDataSource]:=
odbcDatasource
wzLoginData [#ODBCDesc]:=
"Student Records Database"
wzLoginData [#ODBCUID]:= oddbcUser
//Логин для доступа к базе данных
```

```
wzLoginData[#ODBCPWD]:= oddbcPassword// Пароль для доступа к БД
wzLoginData [#AccessFile]:= uRecordsFolder ^ "Users. mdb"//Путь к БД
wzLoginData [#LoginAttempts]:= logonAttempts//Количество попыток
//ввода пароля
wzLoginData [#FirstDate]:= uFirst Date//Дата первого входа
wzLoginData [#TotalTime]:= uTotalTime//Общее время обучения
wzLoginData [#GroupsComplete]:= sectionComplete//Пройдено тем
wzLoginData [#CurrentGroup]:= sectionCurrent//Текущая тема
wzLoginData [#CurrentQuestion]:= pageCurrent//Текущий вопрос
```

При создании более сложной структуры учебника база данных может находиться в заранее не определенной директории, так что следует учитывать и такие факторы. Путь к ней может храниться в реестре Windows или, например, в ini-файле. Для доступа к последним в AuthorWare 7.0 предусмотрено два Knowledge объекта — Read INI Value и Write INI Value. Разобраться в их настройке и работе несложно. Достаточно указать место положения ini-файла, секцию и имя параметра, а также имя переменной, которая хранит значение в учебнике.

Результаты тестирования пользователя: ИВАНОВ ИВАН	
Дата тестирования:	31.01.2005
Название теста:	1.2 Основные задачи
Всего вопросов:	1
Верных ответов:	1
Неверных ответов:	0
Результат:	100%

Рис. 8. Форма вывода результатов тестирования

Knowledge Object Scoring — удобный и достаточно простой мастер создания группы вопросов, результаты прохождения которых будут записаны текущему пользователю как отдельный тест. Опишем вкратце его настройку. Первый шаг — добавление в список тех вопросов, ответ на которые следует учитывать при подведении итогов тестирования. Далее следует ввести заголовки теста, который будет отображаться в базе данных и объединять вошедшие в данный тест вопросы. Можно определить и порог прохождения теста (по умолчанию

он задан 70%). Последний шаг — выбор варианта хранения результатов тестирования. В нашем случае мы используем базу данных ODBC. Внешний вид окна вывода результатов можно настроить на этапе просмотра созданного теста. Это может выглядеть, например, так, как показано на рис. 8.

Управление переходами между вопросами легко осуществляется с помощью иконки Framework. Для этого все вопросы и Knowledge Object Scoring следует разместить справа от Framework на Flowline слева направо в том порядке, в каком они должны представляться пользователю. При этом Scoring должен быть самым последним. Вся навигация в Framework в самом простом случае может сводиться к двум-трем кнопкам. Для этого в иконку Framework поместим другую иконку — Interaction. А вот уже на нее следует «нанизывать» те объекты,

которые вы намереваетесь использовать для навигации между вопросами. Это могут быть как иконки Calculation, так и Navigate. Удобство иконок Navigate состоит в том, что для быстрого перехода между вопросами, принадлежащими родительскому Framework, достаточно выбрать одно из предлагаемых направлений: Previous, Next, First, Last и Exit Framework/Return. Важным свойством объектов Response, которые создаются с каждой добавленной иконкой на одной горизонтальной линии с иконкой Interaction, является свойство Active if. Манипулируя им, можно добиться, например, того, чтобы кнопка «Преды-

дущий вопрос» не появлялась на первом вопросе, а кнопка «Следующий вопрос» была невидима до тех пор, пока пользователь не дал ответ на текущий вопрос. Сделать это очень легко, если знать основы встроенного в Authorware алгоритмического языка.

Рассмотрим пример условия активности кнопки «Следующий вопрос»:

```
CurrentPageNum@"Quest_FW"  
<PageCount@"Quest_FW"&RepCount <> 0,
```

где CurrentPageNum и PageCount — номер текущей страницы (вопроса) и количество страниц во фрейме Quest_FW соответственно, а RepCount определяет, был ли дан ответ на текущий вопрос.

Можно заметить, что в представленном выражении используется имя самого фрейма, по страницам которого происходят переходы. Удобен ли будет такой подход, если в учебнике будет несколько десятков, а то и сотен групп тестов? Нет, здесь следует поискать более изящные решения. Одно из возможных — записывать имя фрейма, в который мы входим, в переменную. Для этого следует вызвать окно Calculation для фрейма через его контекстное меню или нажатием сочетания клавиш Ctrl + =. Все, что будет написано в этом окне, будет выполняться непосредственно перед тем, как пользователь зайдет во фрейм, т.е. запустит тестирование. В окне же будет достаточно ввести одну строчку, чтобы записать имя иконки Framework в переменную:

```
CurrentQuestionFrame:= IconTitle (IconID)
```

Эта строка будет сохранять имя иконки, которая является «хозяином» данного выражения в переменную CurrentQuestionFrame. Условие навигации для вопросов теперь может выглядеть так:

```
CurrentPageNum@CurrentQuestionFrame  
<Page Count@CurrentQuestionFrame&RepCount <> 0
```

Об условиях активности кнопок навигации можно забыть и просто копировать

весь фрейм целиком, включая выражение, предшествующее его запуску.

Таким же способом можно решить и ряд других вопросов. Наиболее важным из них является управление воспроизведением звука в уроках. Это довольно актуальная проблема, поскольку почти каждый, кто хочет иметь полноценный учебник по той или иной тематике, задается вопросом: а можно ли будет в процессе прослушивания и изучения материала приостановить звук, чтобы сделать какие-то пометки в тетради или еще раз прочитать выведенное на экран определение? Причем в учебнике также должна быть возможность продолжения воспроизведения с того же места. Указанная задача решается описанным выше методом.

Для каждой иконки Sound можно задать предопределяющий код, в котором также присваивать определенной переменной имя иконки. Делается это для того, чтобы не перегружать учебник иконками навигации и управления воспроизведением. Так, например, для всех уроков и всех звуковых фрагментов можно делать всего один набор кнопок, размещенный в главном фрейме. Иначе придется для каждой новой иконки Sound делать свою кнопку «Пауза», а для каждого урока — свою навигацию для перехода между его частями. В нашем случае строка кода для иконки Sound выглядит следующим образом:

```
current_wave_name:= IconTitle (IconID),
```

а реакция на нажатие кнопки «Пауза» во всем учебнике представлена иконкой Calculation:

```
if my_playing = 1 then  
    MediaPause (IconID@current_wave_name;  
TRUE)  
my_playing:= 0  
else  
    MediaPause (IconID@current_wave_name;  
FALSE)  
my_playing:= 1  
end if,
```

где переменная `my_playing` является флагом, определяющим, в каком состоянии сейчас находится текущий звуковой фрагмент.

Внимание! Не забудьте сбрасывать флаг в положение, указывающее на то, что звук не в состоянии паузы, при выходе из урока в меню или при переходе в следующий урок. Иначе если вы вышли из урока с нажатой паузой, то в следующем уроке кнопка «Пауза» может сработать один раз «в холостую», т. е. после ее нажатия ничего не произойдет.

Мы затронули вопрос перехода из одного урока в другой, причем переход должен осуществляться строго на урок, который числится следующим в меню, даже если он принадлежит уже другой главе. В решении этой проблемы нам помогут переменные, которые мы заполняли при составлении меню, — в частности, переменная `myItemsID`. Разберем по шагам, что следует учесть при переходе между уроками.

1. Вычислить номер следующего пункта меню после текущего урока.

2. Найти идентификатор иконки, содержащей следующий урок.

3. Проверить, существует ли такая иконка, которая соответствует данной позиции в меню. В случае, если иконка не существует, идентификатор будет иметь значение 0.

4. Организовать цикл по проверке значения идентификатора до тех пор, пока он не примет ненулевое значение.

5. Выделить в меню новый найденный пункт. Это нужно для того, чтобы видеть после возврата в меню, какой урок был посещен последним.

6. Сбросить флаг `my_playing` (см. выше).

7. Перейти на новый найденный урок.

Все рассмотренные шаги представлены в листинге:

Переход между уроками

```
selectedItem:= selectedItem + 1
NextItem:=GetLine (myItemsID;
selectedItem)
repeat while NextItem = 0
selectedItem:= selectedItem + 1
NextItem:=GetLine (myItemsID;
selectedItem)
```

```
end repeat
wcSetPropertyValue (tvw_01; "Value";
selectedItem)
```

```
my_playing:= TRUE
```

```
GoTo (NextItem)
```

Важное замечание. При работе с Knowledge Objects необходимо помнить, что фактически каждый объект представляет собой не один, а сразу несколько различных элементов (иконок), связанных между собой. Дело в том, что в Authorware существует ограничение на максимально возможное количество элементов в учебнике — не более 32 768 штук. При самостоятельной разработке учебника достичь этого предела довольно сложно, однако активное использование Knowledge Objects может в корне переломить ситуацию.

В заключение можно добавить: несмотря на наличие в данном программном пакете некоторых нестыковок и шероховатостей, он остается одним из мощнейших специализированных средств создания мультимедийных учебников.

Более полное представление о возможностях Authorware можно получить, располагая конкретным экземпляром мультимедийного учебника. Авторы издали ряд таких учебников, и один из них — учебник для освоения языка программирования Турбо Паскаль. Одним из достоинств учебника является то, что для облегчения усвоения студентами наиболее сложных вопросов (например, рекурсий, структур динамических переменных и т. п.) соответствующий материал представлен с использованием интерактивных видеоклипов, в которых обучаемый может проинформировать пошаговую трассировку программы с динамическим отображением процессов, происходящих в структуре оперативной памяти.

Литература

Гультяев А.К. Macromedia Authorware 6.0. Разработка мультимедийных учебных курсов. СПб.: Учитель и ученик; Корона принт, 2002.