

## СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВОЗМОЖНОСТЕЙ СОВРЕМЕННЫХ СРЕДСТВ ОБРАБОТКИ АУДИОВИЗУАЛЬНОЙ ИНФОРМАЦИИ ДЛЯ СОЗДАНИЯ ПЕДАГОГИЧЕСКИХ ПРИЛОЖЕНИЙ

Технологии мультимедиа без сомнения сегодня представляют собой одно из передовых достижений в сфере применения ПЭВМ в обучении. Однако создание полноценных программ, использующих средства мультимедиа, требует достаточно высокого уровня знаний и, как правило, под силу лишь подготовленному программисту. Большинство же учителей информатики, не говоря уже об учителях-предметниках, такими знаниями не обладают. И это составляет одну из основных причин той плачевной ситуации с образовательным программным обеспечением, которая наблюдается в настоящее время: учитель, способный дать обучающим программам полноценное с точки зрения методики информационное наполнение, не может довести эти идеи до стадии пригодного к эксплуатации программного продукта; для профессиональных же программистов, напротив, не затруднительно создать высококлассную «оболочку», но ее наполнение очень редко оказывается удачным для использования в школах.

Возможный вариант решения данной проблемы - дать возможность учителям, в том числе предметникам, самостоятельно разрабатывать необходимые учебные программы, обладая лишь навыками пользователя ПК. С этой целью создан ряд программных пакетов, базирующихся на идеологии «программирование без программирования» (идеологии ПБП).

К системам, построенным на базе идеологии «программирование без программирования» (идеологии ПБП) будем относить программные пакеты, предоставляющие пользователю возможность создавать более или менее полноценные интерактивные приложения без написания собственно программного кода на каком бы

то ни было языке программирования. Как правило, подобные системы представляют собой рабочую среду, позволяющую сформировать набор рабочих окон (фреймов), содержащих произвольный фон, ряд управляющих объектов (типа кнопок), а также объектов, воспроизведение которых представляет собой один из видов действий. Управляющим объектам (а также в ряде случаев отдельным окнам и комплекту окон в целом) могут быть присвоены типовые действия из предоставляемого в отдельном меню набора, среди которых - переход к другому окну, воспроизведение звука, анимации, видео и т. п.

Идеология ПБП включает в себя такие идеологии как: работа с типовыми классами объектов, отслеживание событий и элементы визуального программирования. Ключевым же моментом ПБП является характер интерактивного конструирования создаваемого приложения из готовых объектов и модулей действий, как из своего рода «строительных кубиков» в детском конструкторе, когда разрабатываемый сценарий пользователь может даже вовсе не уметь программировать ни на одном из языков программирования и даже не знать, какие вычислительные процессы скрыты за каждым объектом или действием. Вместе с тем системам ПБП, в отличие от полноценных языков программирования, свойственна ограниченность предоставляемых пользователю возможностей (то есть ограниченный набор объектов, манипуляций над ними и запускаемых при этом действий), целиком зависящая от воли разработчика данного пакета ПБП.

Данный материал базируется на результатах сравнительного анализа четырех программных пакетов, относимых к

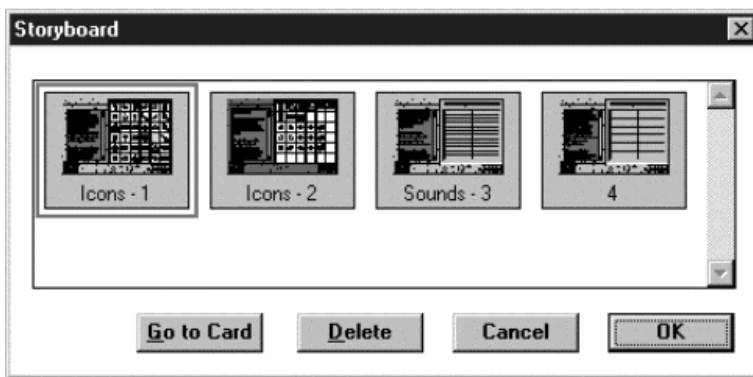


Рисунок 1. Набор карточек, входящих в состав одного сценария (окно расширения StoryboardExtra).

системам ПБП, с точки зрения требований, накладываемых образовательной сферой применения разрабатываемых с их помощью сценариев: HyperStudio (разработчик - фирма Roger Wagner, Великобритания), MultiVision (разработчик - фирма IST), HyperMetod (разработчик - фирма Prog.Systems AI Lab, Санкт-Петербург) и HM-Card (Institute of Information Processing and Computer Supported New Media (ИИСМ), Грац, Австрия).

### ТИПОВАЯ ИДЕОЛОГИЯ ПБП

Типовой пакет ПБП представляет собой интерактивную систему (мультимедиа-конструктор), предназначенную для разработки программ (сценариев) в основном демонстрационного характера с использованием технологий мультимедиа, включая возможность воспроизводить на экране текст (в том числе содержащий гиперссылки), графические изображения, звук, анимации и видеофрагменты. Такой сценарий, как правило, представляет собой файл-документ, для работы с которым (просмотра) необходимо наличие полной версии пакета, в котором он был разработан, либо его усеченной версии - «плеер»

ра» (иногда возможна также генерация автономно исполняемого программного модуля).

Основным элементом сценария является карточка (сцена), представляющая собой единичное окно (или кадр экранного изображения) с собственным фоном и объектами интерфейса (текст, графика, анимация, видеофрагменты, кнопки) (рисунок 1). Разрабатываемое педагогическое приложение может

включать в себя несколько отдельных сценариев, один из которых запускается на исполнение первым, а остальные вызываются из него.

Объектам (как правило, кнопкам и «активным зонам»; в ряде случаев также другим объектам) могут быть присвоены действия, срабатывающие по заданной манипуляции пользователя с данным объектом (щелчок мышью на объекте или наведение на него курсора мыши и т. д.) либо запускаемые автоматически по истечении определенного времени с момента открытия данной карточки. Типовыми действиями являются: переход с одной карточки на другую или переход к другому сценарию, запуск другого приложения Windows,

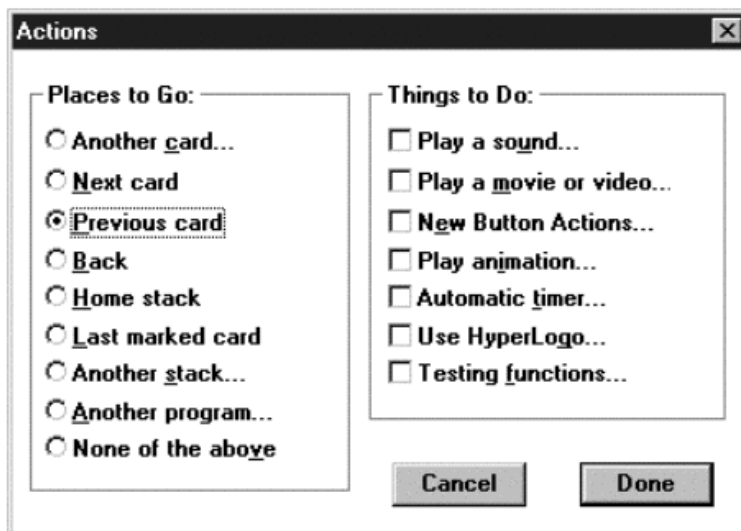


Рисунок 2. Действия, присваиваемые объектам.

воспроизведение звука, анимации или видеофрагмента, а также другие управляющие действия (рисунок 2). Набор возможных действий и объектов, как правило, жестко задан, однако в ряде пакетов ПБП предусматривается встроенный язык программирования (аналогичный какому-либо языку высокого уровня или языку макрокоманд), позволяющий программировать произвольные желаемые действия, а иногда и создавать новые объекты. (Правда, при работе с подобным языком программирования преимущества идеологии ПБП в значительной мере утрачиваются.)

Обычно каждому объекту может быть присвоено несколько разнотипных действий. Объект может использоваться для перехода на другую карточку, сопровождаемого иным действием (например, звуковым эффектом), либо для запуска каких-либо реакций программы на манипуляции пользователя в пределах данной карточки (скажем, запуск анимации с параллельным звуковым сопровождением). В некоторых пакетах почти тот же набор действий (кроме переходов) может быть присвоен карточке или сценарию в целом, их выполнение запускается при открытии или закрытии пользователем данной карточки (сценария), либо при щелчке мышью на фоне карточки вне каких-либо размещенных на ней объектов.

## ПРЕИМУЩЕСТВА И НЕДОСТАТКИ СУЩЕСТВУЮЩИХ ПАКЕТОВ ПБП

### Пакет HyperStudio

В пакете HyperStudio (рисунок 3) основным преимуществом является удачная идея сочетания как интерактивной среды конструктора мультимедиа-приложений, так и подключаемых извне дополнительных модулей действий и встроенного языка программирования. В отличие от него, пакет MultiVision при столь же удобной интерактивной среде не содержит в себе языка программирования, пакет же HyperMetod при наличии мощного аппарата скриптов обладает слишком слабой интерактивной средой конструктора программ.

Основные недостатки HyperStudio - слишком слабый набор действий, требуемых для создания тестирующих приложений и неудачный выбор в качестве встроенного языка программирования языка типа Лого, преимущественно ориентированного на работу с машинной графикой, а не на реализацию вычислительных и интерфейсных алгоритмов. Кроме того, неудобством является хранение готового сценария в виде неавтономного «документа», для просмотра которого требуется сам пакет или его «усеченная» версия - «Плейер».

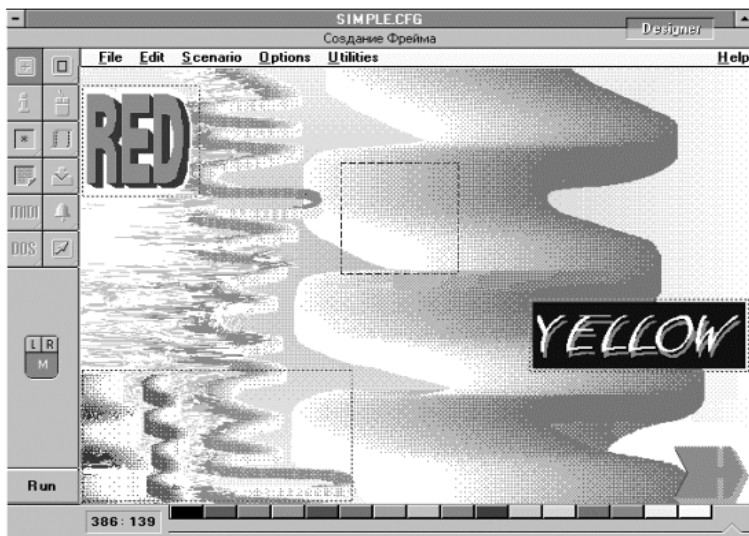


Рисунок 3. Общий вид экрана HyperStudio с извлеченными из меню окнами инструментов и палитры.

В целом HyperStudio представляет собой скорее систему для создания мультимедийных демонстрационных роликов типа «компьютерных справочников» с «однонаправленным» (от компьютера к пользователю) потоком информации, но зато доступную даже при минимальном уровне подготовки пользователя.

#### **Пакет MultiVision**

Основными преимуществами пакета MultiVision (рисунок 4) являются: многомодульная структура пакета (позволяющая «развязать» между собой различные по характеру подготовительные работы), возможность создания готового сценария в виде автономного исполняемого модуля как под DOS, так и под Windows (в том числе с генерацией полноценного поддискетного дистрибутива), идеология «слота», позволяющая присваивать одному и тому же объекту несколько наборов действий (правда, в Hyper Method эта возможность реализована еще более широко), и великолепный механизм реализации анимаций (модуль Deformer и возможность синхронизации нескольких анимаций и звукового сопровождения). В новой версии существенным дополнением стало добавление поддержки тестирования, хотя бы в простейшем варианте «вопрос - выбор правильного ответа из набора представленных».



**Рисунок 4.** Рабочий экран пакета MultiVision.

Недостатками Multi Vision являются: всегда постоянный размер окон, отсутствие поддержки воспроизведения видеофрагментов и допустимость присвоения действий только активным зонам.

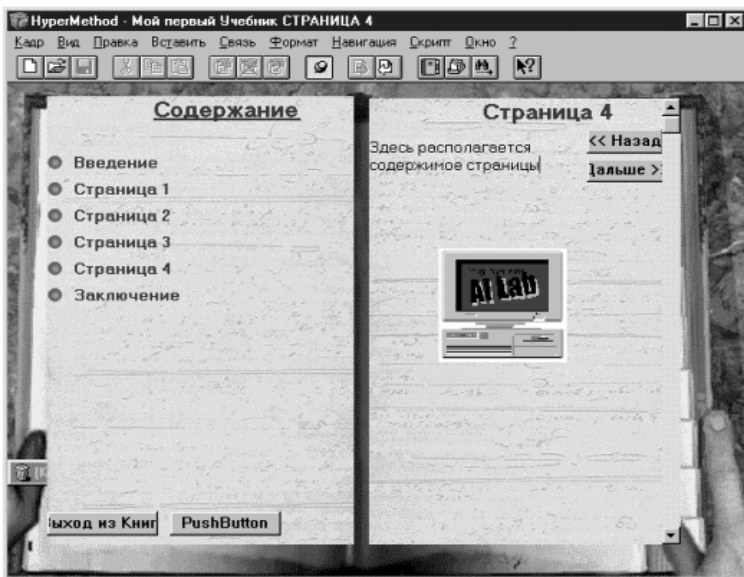
В целом пакет MultiVision представляет собой простое для освоения средство для создания презентационных роликов и простейших «электронных учебников» с опорой на анимацию (вместо видео) и реализацией экспресс-опроса после каждой пройденной темы. Более широкие возможности могут быть достигнуты путем разработки дополнительных модулей на языках программирования типа Бейсика, Паскаля или Си, подключаемых к сценарию на MultiVision в качестве внешних исполняемых программ.

#### **Пакет HyperMethod**

Основным преимуществом пакета HyperMethod (рисунок 5) является практически неограниченный (благодаря использованию механизма OLE) набор возможностей и мощнейший Бейсик-подобный скриптовый язык. Кроме того, весьма полезна возможность задания произвольного размера, положения и фона (в том числе строки заголовка) любого окна.

Главный недостаток пакета HyperMethod - слабая реализация именно интерактивной среды конструктора программ, что делает этот пакет недоступным большинству непрофессиональных пользователей.

В целом пакет HyperMethod представляет собой достаточно мощное средство для разработки практически любых мультимедиа-приложений, в том числе и для сферы образования, но предназначенное почти исключительно для профессиональных программистов и немногим менее сложное, чем язык Visual Basic.



**Рисунок 5.** Рабочее окно пакета HyperMethod («Монтажный стол»).

**Пакет НМ-Card**

Главное, решающее преимущество пакета НМ-Card - это, без сомнения, реализованный в нем механизм создания интерактивного диалога (рисунок 6). Полезными также являются механизм переменных MailBox, возможность мерцания для выделения значимых фрагментов текста или изображения и такие графические объекты как Вектор и особенно Кривая (сплайн).

Недостатков же у данного пакета, к сожалению, немало.

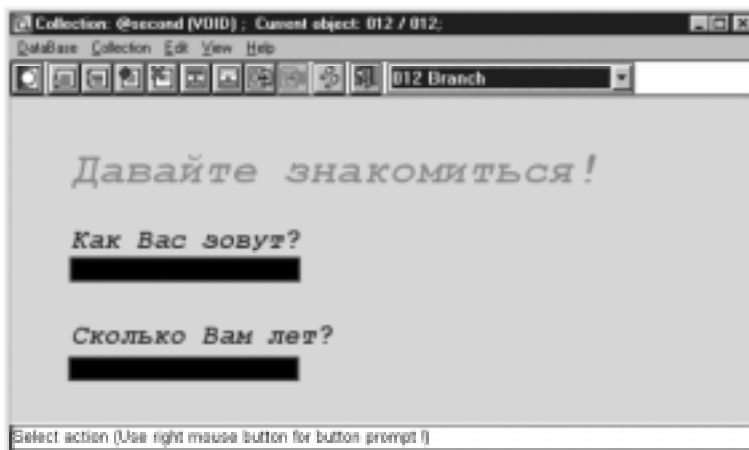
Во-первых, для неподготовленного пользователя слишком сложны как идеология «контейнеров», так и необходимость работы с линкером для реализации даже самого простого сценария, состоящего из линейной последовательности нескольких коллекций. Гораздо удобнее была бы концепция отдельных простых сценариев - комплектов атомарных коллекций, вызываемых друг из друга, как это предусмотрено в HyperStudio и других

аналогичных системах, или, по крайней мере, предоставить возможность создания хотя бы простейших сценариев без использования линкера.

Во-вторых, крайне неудобно то, что создаваемая для каждого сценария база данных (ее файлы) всегда записывается на диск под одним и тем же именем. Таким образом, разные сценарии можно хранить на диске только в разных подкаталогах.

Третий недостаток состоит в смешении понятий объектов и действий в одном общем понятии

«объект НМ-Card». Гораздо рациональнее было бы сохранить прежний принцип присваивания объектам действий, в том числе скриптов. Кроме того, в пакете НМ-Card отсутствует «штатная» возможность работы с диском (даже в таком минимальном варианте, как, например, в HyperStudio), хотя эта возможность могла бы быть полезной не только для хранения данных (MailBox) на диске для использования при следующих запусках сценария, но и для протоколирования ответов учащихся на вопросы тестирующего сценария. Нет здесь и стандартного механизма



**Рисунок 6.** Рабочее окно пакета НМ-Card (редактор коллекций).

воспроизведения видеозаписей, чтобы их можно было разместить непосредственно в нужном месте карточки и с требуемым размером изображения, а также обеспечить управление воспроизведением непосредственно из сценария.

И, наконец, эффекты мерцания и шторки логичнее было бы также присваивать объекту или группе так же, как действия.

Таким образом, в целом пакет NM-Card в наибольшей степени из четырех рассмотренных пакетов подходит для реализации интерактивных сценариев типа тестирующих программ, не требующих богатого графического оформления. Однако ему свойственен ряд существенных недостатков, самый главный из которых -

чрезмерная сложность работы с отдельными редактором и линкером, из-за чего рекомендовать его можно лишь для в достаточной степени подготовленных пользователей.

В целом же приходится, к сожалению, сделать малоутешительный вывод, что ни один из существующих в настоящее время пакетов ПБП пока что не способен в полной мере отвечать требованиям, предъявляемым к системе, предназначенной для создания учителями-предметниками педагогических приложений. Разработка подобной системы является «задачей номер один» в решении проблемы компьютерной поддержки преподавания учебных предметов.

*Усенков Дмитрий Юрьевич,  
старший научный сотрудник  
Института информатизации  
образования РАО.*

**НАШИ АВТОРЫ**