

Методы реализации пользовательской модели ИУС в обучающих системах

В.К. Григорьев¹, О.А. Аксенов²

¹доцент, к.т.н.,

²аспирант,

^{1,2}кафедра математического обеспечения вычислительных систем,

Московский государственный университет радиотехники электроники и автоматики,

Москва, Россия

grigoriev@mirea.ru, aksenov@mirea.ru

АННОТАЦИЯ

В данной статье рассматриваются методические особенности создания тьюторных обучающих компонент (ТОС) информационно-управляющих систем. (ИУС) Проводится классификация методик реализаций ТОС и даются примеры их использования. Проводится анализ выбора рациональных методик для конкретных реализаций обучающих компонент.

Ключевые слова

тьюторная обучающая программа, информационно управляющая система, модель массового профессионального пользователя, инструментальная система

Введение

Процесс внедрения новых и модифицированных информационно-управляющих систем (ИУС) встречается с основной трудностью так называемого «разрыва знаний». Действительно в случае внедрения новых ИУС знания о ней имеет ограниченное число людей (чаще всего разработчики), которые чрезвычайно заняты самим процессом внедрения и отладки системы, а для эффективного использования системы эти знания требуются большому числу пользователей. В соответствии с работой [Григорьев, Кричевец 2003] пользователей системы можно разделить на три класса, среди которых обучение первых двух возможно с применением учебных центров, а обучение третьего класса массовых профессиональных пользователей (МПП) наиболее эффективно с использованием тьюторных обучающих систем (ТОС).

Важными особенностями разработки и создания ТОС являются адекватность реальным ИУС и возможность быстрого и эффективного создания ТОС с быстрой модификацией. Причем важной особенностью является тот момент, что разработка обучающей системы должна проходить параллельно с созданием ИУС или с этапом ее бета тестирования.

С этой точки зрения технология создания существенно опирается на инструменты создания. Одним из подходов создания инструментов, поддерживающих процесс разработки обучающих систем, представлен в статье [Аксенов 2003]. В настоящей статье будут рассмотрены основные методики создания ТОС и варианты инструментальной поддержки этих методик.

Методология

Методология создания тьюторных обучающих компонент ИУС основана на реализации в рамках обучающей программы модели ситуации, представляемой в виде задания, помощи и модели ИУС, на которой организуется основной диалог обучения. Принцип организации этого диалога основан на сценарном подходе к обучению, согласно которому взаимодействие обучаемого и тьюторной программы

задается набором инструкций сценария. Базовый интерфейс тьюторной компоненты ИУС, в рамках которого представлен данный диалог, описан в работе [Григорьев 1999] и включает такие основные области, как задание, помощь, элементы управления диалогом обучения и модели ИУС. Рассматриваемые в рамках данного подхода обучающие программы придерживаются именно этой модели интерфейса.

При создании обучающей программы составление заданий на обучение и информационное наполнение обучающей компоненты представляет из себя методическую часть создания тьюторной компоненты. Создание же программной модели ИУС, для которой создается обучающая компонента, может являться как задачей методистов, так и задачей программистов обучающей системы. При этом в зависимости от реализованных в обучающей системе подходов этап составления конечной обучающей системы может проходить как при участии, так и без участия программистов.

Какие же существуют методы реализации модели информационно-управляющей системы? Практическое исследование этих подходов выделило следующие основные варианты реализации этой модели:

- на снимках экранов (скриншотах), когда в качестве модели ПП используется картинка экрана, снятая с реальной программы;
- объектные, когда в качестве модели ПП используется некоторый код программного продукта, объекты интерфейса, программа, эмитирующая работу исходного ПП, перенесенные в обучающую систему;
- комбинированные, когда для представления объектов системы используются как снимки экранов, так и объекты системы. Подобный подход призван обеспечить большую функциональную приближенность элементов интерфейса, предоставляемых в диалоге обучения, по сравнению с системами, использующими исключительно снимки экрана. При этом комбинированные системы сохраняют простоту создания обучающих компонент, свойственную системам на снимках экранов.

Выбор того или иного метода диктуется следующими требованиями:

- функциональности обучающей системы;
- требуемой скорости ее создания;
- квалификации персонала, участвующего в создании обучающей компоненты;
- требованиями к объемам памяти обучающей системы;
- наличием исходных текстов, образов ресурсов программного продукта, для которого создается обучающая компонента и др.

Реализация

В течение семи лет в лаборатории «Электронные технологии в финансах» Московского института радиотехники электроники и автоматики ведется разработка тьюторных обучающих компонент ИУС. В рамках реализации этих компонент были исследованы и практически реализованы различные принципы создания таких систем. При этом можно сказать, что в каждой обучающей системе используются одинаковые тьюторные принципы обучения и подходы к проектированию интерфейса. Основным же отличием данных программных продуктов является методика реализации модели ИУС в рамках обучающей программы. На рисунке 1 представлена классификация данных тьюторных обучающих компонент.



Рисунок 1. Классификация реализованных программных моделей обучающих компонент ИУС.

Как видно из схемы, представленной на рисунке 1, в ходе исследования были апробированы 5 основных подходов к созданию модели информационно-управляющей системы. Данная классификация не претендует на полноту используемых методов реализации, однако, как показывает практика, введение новых подходов будет либо детализацией одного из выбранных подходов, либо комбинацией используемых методик. Рассмотрим подробнее каждую реализацию, с целью выявления положительных и отрицательных сторон каждого из используемых подходов.

Первой из реализованных в ходе данного исследования методик является создание запрограммированной модели ИУС. Пример реализации подобной компоненты показан на рисунке 2.

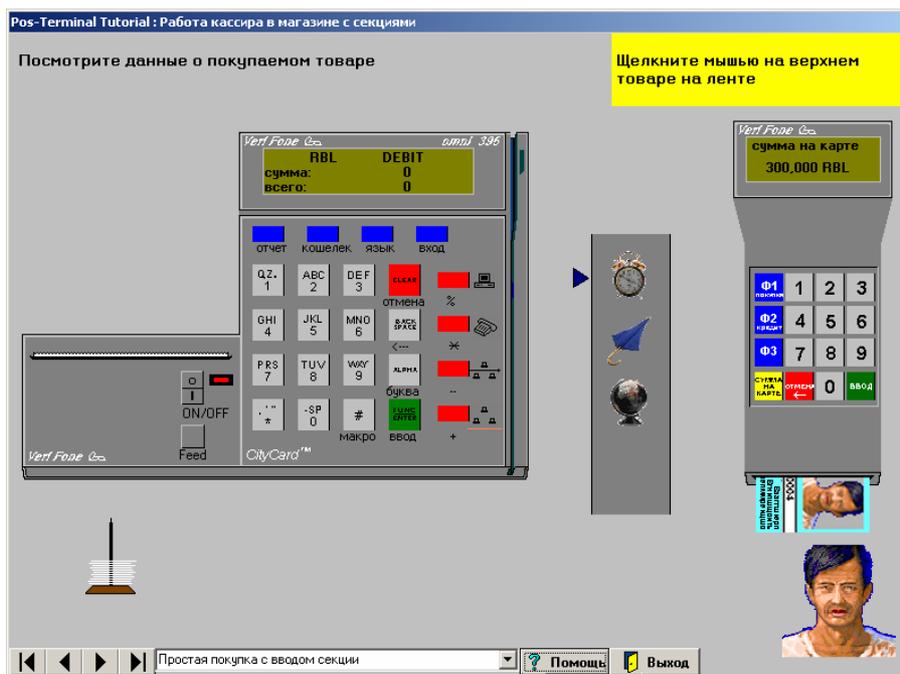


Рисунок 2. Тьюторная обучающая система «Работа кассира на POS терминале».

Этот подход является наиболее трудоемким в реализации, т.к. требует создание модели программного продукта, повторяющей его функциональность в таком объеме, насколько это требуется для организации диалога обучения. При этом часть функций по организации диалога ложиться на сценарий обучения, который управляет данной моделью.

Созданная таким образом модель по сути не является самой ИУС даже в приближенном виде, а является только программным компонентом, способным эмитировать действия реальной ИУС с возможностью выполнения таких базовых операций оболочки обучающей системы, как проверка правильности действия, организация отката действия (в случае неправильного выполнения и по запросу обучаемого), установка состояния системы в новое, согласно произведенным пользователем действиям. В данном случае, согласно требованиям ТОС «Работа кассира на POS терминале», были созданы две программные модели, а именно устройства кассового аппарата с устройством считывания смарт карт, и модели покупателя и ленты покупки. Данный компонент показывает, что реализация подобной модели может быть применима не только для программных продуктов, но и для физических устройств.

Использование подобного подхода требует обязательного наличия программистов при разработке обучающей системы, что требует таких долгих операций, как создание обучающей системы, тестирование ее программной реализации и последующее тестирование сценария данной системы [Григорьев, Каюшин 2000]. В случае разработки модели сложной ИУС этот подход может потребовать значительных затрат и самое главное, что создание обучающей системы может значительно опаздывать от момента выхода новой версии программного продукта. Однако реализованная таким образом система может достаточно точно симитировать работу программы, что является несомненным достоинством данного подхода.

Вторым из исследованных подходов является создание модели ИУС на основе снимков экранов. Этот подход является достаточно универсальным и простым в реализации, так как моделью системы является снимок ее экрана на текущем шаге. Однако это преимущество в простоте реализации является также и недостатком данного подхода, так как не дает возможности реализовать многие функциональные возможности объектов интерфейса, к примеру, мигающего курсора в поле редактирования и мн. др. Пример картинки данной обучающей программы приведен на рисунке 3.

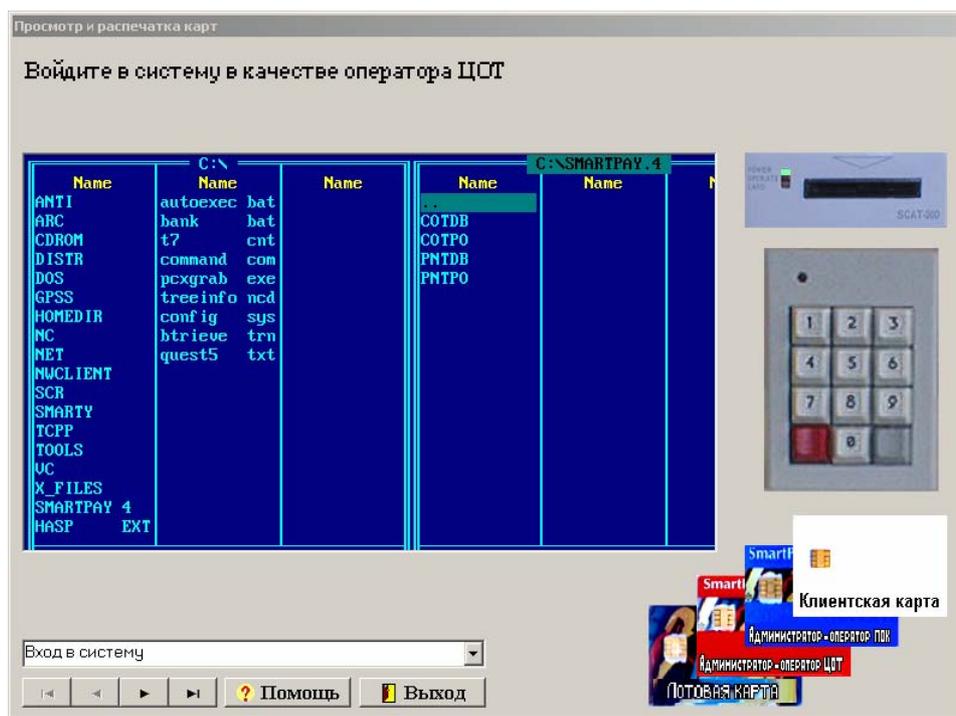


Рисунок 3. Экран тьюторной обучающей компоненты системы SmartPay.

Созданная с использованием такого подхода система [Громов 2001] не имела достаточного набора средств для автоматизации заполнения сценария, что являлось причиной достаточно долгой ее разработки, однако создание данной системы подтвердило возможность использования данного подхода для систем, работающих в текстовом режиме экрана. Следует заметить, что в данной обучающей компоненте все-таки присутствует одна запрограммированная модель, а именно модель устройства чтения карт, в задачи которой входит имитация данного устройства.

Исследование возможностей быстрой разработки обучающих компонент сделала необходимым разработку новых методик создания обучающих компонент, связанных с разработкой удобных инструментальных средств заполнения сценария [Аксенов 2001] и с разработкой методов автоматизации переноса ресурсов исходной ИУС в обучающую подсистему. Результатом данного исследования явились две программные модели обучающих компонент, реализованных для ИУС созданных в среде Borland Delphi в рамках лаборатории.

Методика встраивания обучающей компоненты, при которой модель ИУС строится из ее собственного кода, показала возможности такого способа создания обучающих компонент. Однако исследования в данной области показали, что созданная таким образом модель не во всех случаях способна удовлетворить таким требованиям к созданию обучающих компонент, как сохранение целостности данных реальной системы, обеспечение операций возврата на предшествующие шаги сценария и др. Сутью реализации такого подхода является перехват сообщений поступающих данному приложению на управляющие модули обучающей системы, которая либо запрещает, либо разрешает эти сообщения. Программирование данного подхода столкнулось с рядом трудностей, а именно с особенностями реализации некоторых интерфейсных элементов. Данная методика проверялась на проекте ИУС [Громов 2002], созданном в среде Borland Delphi, однако реализованные принципы могут применяться и к проектам, созданным с использованием других инструментальных средств. Таким образом, можно резюмировать, что использование подобной методики возможно, но сопряжено с рядом ограничений, что делает трудным ее применение для ряда систем (особенно использующих базу данных) и делает ее сильно зависимой от среды программирования, используемой для создания ИУС. Однако созданная таким образом модель системы является достаточно четким повторением оригинала, к примеру (рис. 4) это дало возможность создания диалога обучения графическому моделированию процесса распространения загрязнений в водоеме.

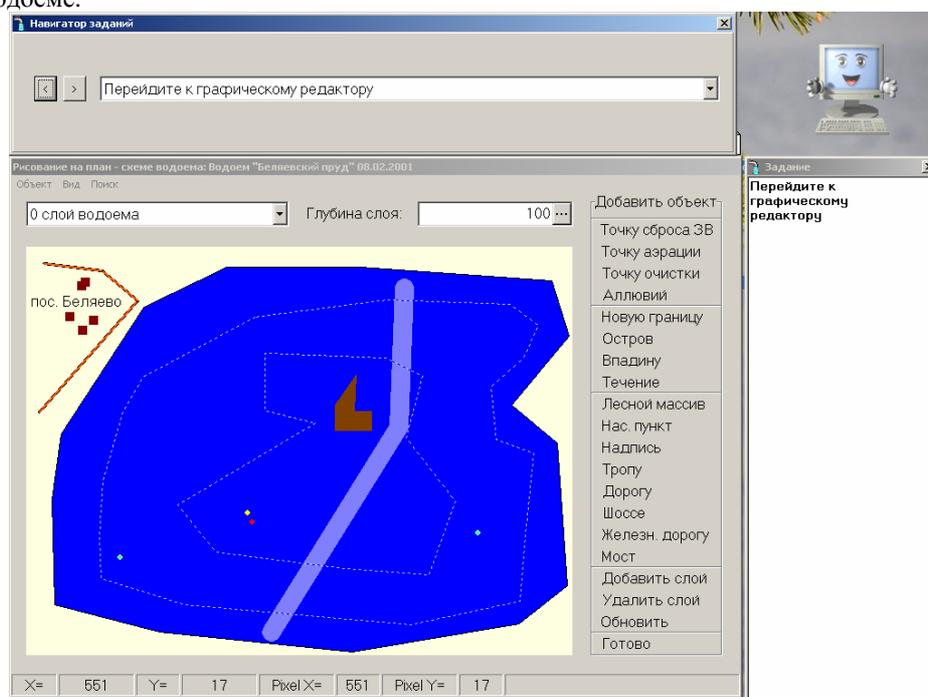


Рисунок 4. Обучающая компонента «Системы моделирования распространения загрязнений в водоеме».

Исследование методики автоматизации создания обучающих компонент привело к подходу, позволяющему переносить объекты интерфейса ИУС в обучающую систему. Этот подход был апробирован для ИУС «Система моделирования распространения загрязнений в речной сети» [Левит-Гуревич 2001] снимок экрана которой показан на рисунке 5, созданной с использованием среды Borland Delphi. Инструментальное средство данной обучающей системы позволяет переносить экранные формы Borland Delphi в состав обучающей системы. При этом функционирование объектов интерфейса задается на уровне языка сценария обучающей системы. Недостатки и достоинства данного подхода схожи с методом встраивания обучающей компоненты. Также следует отметить, что реализованная таким образом обучающая система, рассмотренная в [Аксенов 2002], является окном верхнего уровня по сравнению с моделируемой ИУС, что накладывает свои особенности, связанные с реализацией объектов интерфейса пользователя Windows. В данном случае задача создания модели ИУС ложится на прикладных программистов, которые создают функциональную модель системы на уровне языка сценария обучающей системы, что является достоинством данного подхода.

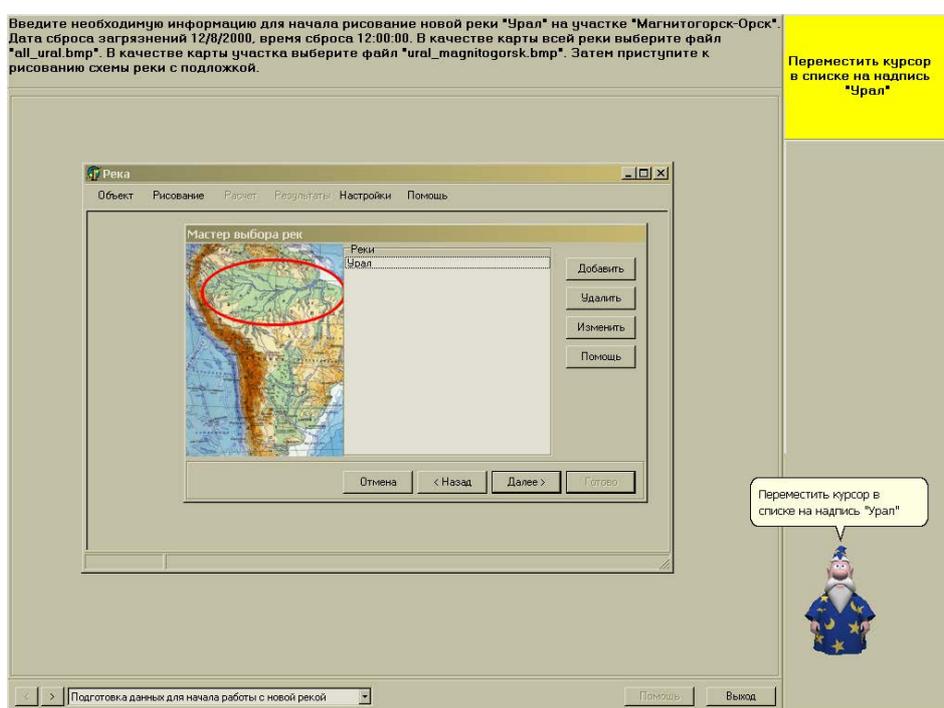


Рисунок 5. Обучающая компонента «Системы моделирования распространения загрязнений в речной сети».

Исследование методик переноса объектов интерфейса ИУС в обучающую систему показало, что технология создания обучающих компонент должна придерживаться следующих принципов: обучающая система должна быть проста в создании, а процесс ее создания должен быть как можно более автоматизированным и требовать минимального участия системных программистов.

На основании данных требований был спроектирован и создан двухмашинный инструментальный комплекс «Построитель тьюторов» [Григорьев 2003], в основе которого лежит как объектная модель управляющих элементов ИУС, так и модель, основанная на снимках экранов. При этом процесс создания модели ИУС, основанной на снимках экранов является автоматизированным и не требует от создателей обучающей компоненты навыков программирования. На рисунке 6 приведен пример экрана окна обучающей системы, в котором на снимке экрана присутствует объект «поля редактирования».

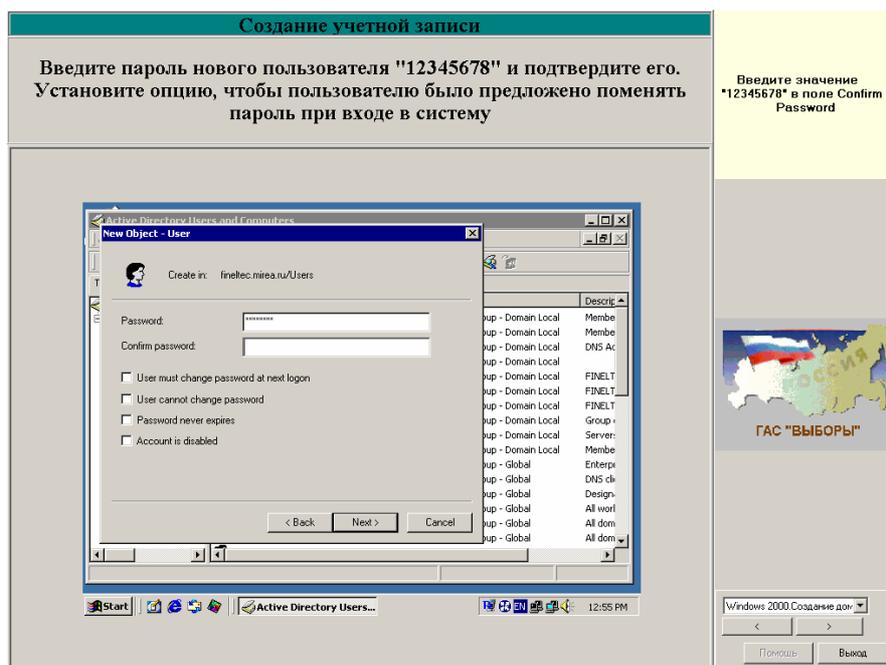


Рисунок 6. Пример экрана обучающей программы, созданной с использование комплекса «Построитель тьюторов».

Этот подход является достаточно универсальным и позволяет как быстрое создание обучающих компонент, так и применение средств программирования для решения нестандартных задач, возникающих при создании модели ИУС. Опыт использования данного комплекса [Аксенов 2003] показывает, что с помощью данного подхода возможно значительное ускорение процесса создания обучающих компонент.

В таблице 1 приведено сравнения подходов, испробованных при создании модели ИУС в рамках тьюторных обучающих компонент. В каждом подходе есть свои достоинства и недостатки, весомость которых зависит от конкретных требований к создаваемой обучающей системе.

Таблица 1. Сравнение достоинств и недостатков моделей интерфейса ИУС.

	Вид системы				
	Объектные			Снимки экранов	Комбинированные
Программа, имитирующая работу	Перенос объектов	Встраивание в исходный код			
Скорость создания обучающих компонент	Низкая	Высокая	Средняя	Высокая	Высокая
Необходимость привлечения квалифицированных разработчиков	Необходимо	Возможно	Необходимо	Не требуется	Возможно
Возможность создания системы без участия системных программистов	Нет	Да	Нет	Да	Да
Приближенность функциональных элементов интерфейса	Высокая	Высокая	Высокая	Низкая	Средняя
Объем сценария	Низкий	Низкий	Низкий	Высокий	Высокий/Средний

Заключение

Исследованная в рамках данной работы проблема создания моделей ИУС в рамках обучающих компонент является достаточно актуальной и востребованной задачей. Создаваемые с использованием данной технологии обучающие программы способны сопровождать внедрение новой ИУС, так и сопровождать курсы обучения. При этом важным аспектом является своевременный выход обучающей компоненты от момента появления новой версии ИУС.

Как показывает практика создания инструментальных средств разработки обучающих компонент, процесс создания модели ИУС проще формализуем в обучающих системах, использующих комбинированные подходы и подходы к созданию модели ИУС на основе снимков экранов. Однако использование снимков экранов не обеспечивает требуемой функциональности элементов управления ИУС, необходимых для создания моделей некоторых обучающих программ. Поэтому в настоящий момент наиболее эффективным и часто используемым является способ создания обучающих компонент, использующий комбинированные принципы, применяемые в ряде российских и зарубежных систем, который в частности реализован в программном комплексе «Построитель тьюторов».

Практическое использование реализованных программных моделей обучающих компонент ИУС подтверждает актуальность использования подобных принципов и применимость тех или иных методов реализации обучающих компонент. Использование же удобных инструментальных средств значительно ускоряет процесс разработки тьюторных обучающих компонент, что подтверждает опыт использования комплекса «Построитель тьюторов».

Литература

- [Григорьев, Кричевец 2003] Григорьев В.К., Кричевец А.К., «Корреляция типов пользователей и методов обучения в большой распределенной информационно-управляющей системе», «Educational Technology & Society» 6(4) 2003, стр. 194-203
- [Аксенов 2004] Аксенов О.А., Григорьев В.К., Инструментальный двухмашинный комплекс для создания обучающих компонент Информационно Управляющих Систем «Построитель тьюторов», №50200301100 от 26 декабря 2003 года
- [Аксенов 2003] Аксенов О.А., «Экспериментальное исследование эффективности инструмента для реализации обучающей компоненты ИУС», «Educational Technology & Society» 6(3) 2003, стр. 129-133;
- [Аксенов 2002] О.А. Аксенов, «Мультимедийные компоненты обучающих систем тьюторного типа», «51 Научно-техническая конференция», МИРЭА, Москва 2002 г.;
- [Громов 2002] Громов С.Ю., «Программная модель процесса распространения загрязняющих веществ в водоеме», «51 Научно-техническая конференция», МИРЭА, Москва 2001 г.
- [Левит-Гуревич 2001] Л.К. Левит-Гуревич, О.А. Аксенов, «Математическая и программная модели процесса распространения загрязняющих веществ в реке», «50 Научно-техническая конференция», МИРЭА, Москва 2001 г.
- [Громов 2001] С.Ю. Громов, «Обучающая программа по системе безналичных расчетов SmartPay», «50 Научно-техническая конференция», МИРЭА, Москва 2001 г.;
- [Аксенов 2001] Аксенов О.А., Григорьев В.К., «Инструментарий для создания тьютора системы CMS SmartCity 1.5», Современные информационные технологии в управлении и образовании, М: ФГУП НИИ «Восход», МИРЭА 2001 г., стр.203-207
- [Григорьев, Каюшин 2000] Григорьев В.К., Каюшин А.А., Бебчик А.А. Обучающие программы для кассиров. // ПЛАС loyalty – 2000. – № 2
- [Григорьев 1999] Григорьев В.К. Классифицирующие признаки тьюторных систем М.: Научно-техническая конференция МИРЭА, 1999