

важных ролей отводится функциональным элементам обучения (ФЭО) – конкретным средствам решения методических и педагогических задач. Многообразие приемов разъяснения изучаемого материала порождает многообразие технологий разработки ФЭО. При разработке обучающих систем, содержащих большое количество сложных формул и рисунков, встраиваемых в HTML-страницы, мы столкнулись с необходимостью разделения этих объектов на сегменты, к которым впоследствии можно было бы обеспечить локальный доступ с помощью стандартной технологии WEB-страниц. При этом в не активизированном (пассивном) состоянии соответствующий объект должен иметь сплошную не фрагментированную структуру, восприниматься как единый объект, обеспечивающий последовательное непрерывное изложение материала. Активизация фрагментов объекта может быть вызвана необходимостью разъяснения их смысла для данного конкретного ученика. Существенно то, что заранее трудно предсказать, насколько обширная «консультация» требуется для понимания в каждом конкретном случае, т.е. в основу форматирования ФЭО следует закладывать предположение о наличии нескольких уровней базовых знаний учащегося. Соответствующий подкласс класса ФЭО представлен нами стереотипом `fragment`, и в данной статье мы рассмотрим лишь возможную *технологию* разработки таких элементов.

Разделение рисунка на активизируемые сегменты.

В качестве инструмента реализации данной операции может быть выбран редактор Adobe Image Ready, устанавливаемый, как правило, вместе редактором Adobe Photoshop 8.0. Для помещения фрагментированного объекта в окно редактора Adobe Image Ready достаточно скопировать его в буфер обмена из доку-

мента, в котором он был создан для помещения на HTML-страницу электронного учебника. Создавать новый документ Adobe Image Ready предпочтительно после помещения объекта в буфер обмена, т.к. в этом случае редактор по умолчанию предлагает по команде `new` из меню `file` открыть окно, размеры которого соответствуют размерам объекта

Разбивка элемента на сегменты осуществляется с помощью инструмента «Кусочек» (`Slice`). После сегментирования файл следует сохранить, выбирая в меню `File > Save Optimized As...` В результате получается HTML-страница, содержащая таблицу, в ячейках которой находятся выделенные части рисунка. Все изображения хранятся в папке “image”. Заходим в эту папку. Найдя необходимую картинку, например, файл – `formula_05.gif`, а в HTML-коде строку «`formula_05.gif`», можно связать с этой частью рисунка гиперссылку, всплывающую подсказку или событие, происходящее при наведении на эту область курсора мыши.

Всплывающая подсказка

В теге рисунка (``) ставим еще один параметр: `alt= «наш комментарий»`. Например, ``

Гиперссылка

Окружаем тег рисунка тегом ссылки (`<a>` и ``) например

```
<a href = "коментрий.htm">img src="1.gif" alt="рисунок"</a>
```

Событие

Создаем в нашем документе форму с текстовым полем в любом месте, например, ниже формулы. Для этого в исходном коде после тега закрытия таблицы (`</table>`) и тега закрытия абзаца (`</p>`) вставим такой код:

```
<p>
<form name=forma>
<input name="comment" type="text" value=""
size="300">
</form>
</p>
```

Новый абзац
Открытие формы, определение ее названия
Создание текстового поля определение его названия, длины и начальный текст
Закрытие формы
Конец абзаца

В тег рисунка добавим такой фрагмент:

```
onmouseover="javascript:document.forma.coment.value='Ускорение точки а';"
```

Это фрагмент кода javascript, который при наведении курсора мыши на рисунок вписывает в текстовое поле определенную строку, обращаясь к ней как объекту.

Вместе получится:

```

```

Можно также создать событие для покидания курсором площади рисунка, чтобы очистить текстовое поле.

Разумеется, предложенную технологию создания фрагментированных ФЭО имеет смысл использовать только тогда, когда исходные объекты не составляют-

ся изначально из фрагментов, каждый из которых может быть обработан в редакторе HTML-страниц.

СПЕЦИФИКАЦИЯ ФЭО В UML-МОДЕЛИ ОБУЧАЮЩЕЙ СИСТЕМЫ

Рыков В.Т.*, Рыкова Е.В.**

*Кубанский государственный университет,

**Кубанский государственный технологический университет,

Краснодар

Понимание обучения как целенаправленного педагогического процесса организации и стимулирования активной познавательной деятельности учащегося заставляет весьма осторожно относиться к заявлениям о разработке автоматизированных обучающих систем, так как предлагаемые в печати до настоящего време-

ни средства воздействия на процесс компьютерного обучения выглядят весьма ограниченными. Высказываемые заявления об адаптивности таких систем по отношению к индивидуальным возможностям учащегося чаще всего декларативны или реализуются в виде многократных проходов по одному и тому же кругу обучения. Да и сами эти проходы назвать обучением можно лишь приближенно.

Разумеется, это не означает, что такие системы разрабатывать не надо, что они бесполезны. Однако в современных условиях, на наш взгляд, на первое место выдвигается не проблема аппаратных средств разработки обучающих систем (на решение этой проблемы направлены усилия программистов всего мира, не зависимо от того, для чего они разрабатывают свои программные продукты), а проблема разработки модели обучения на уровне технологического процесса. Это, прежде всего, разработка конкретных средств организации процесса компьютерного обучения, резонансного воздействия на органы чувств обучаемого, стимулирования его познавательной активности. Такие средства организации процесса обучения мы назвали «функциональными элементами обучения» – ФЭО [1]. Спецификация ФЭО является необходимым условием развития технологии разработки действительно обучающих систем.

Проблему детальной спецификации изделия нельзя решать в отрыве от методов предварительного моделирования этого изделия и задач, порождающих необходимость его изготовления. В качестве основного средства моделирования обучающих систем и реализуемых в них процессов можно рассматривать унифицированный язык объектно-ориентированного моделирования – UML [2], для которого в последнее время разработано большое количество интегрированных сред (например, MS Visio 2003), подключающих к нему при необходимости и известные языки программирования.

В работе [3] нами уже рассматривались некоторые расширения UML, основанные на терминологии, введенной в работе [4] и ряде последующих работ. В данной статье мы хотели подробнее остановиться на функциональном элементе обучения как основном тактическом средстве реализации самого процесса.

1. Стереотип FMT (Functional member of training) – функциональный элемент обучения. При введении нового стереотипа – понятия, не отраженного в терминологии UML, этот стереотип вступает в определенные отношения со стереотипами стандарта UML. С точки зрения задач и обязанностей каждого ФЭО, осуществляющего реализацию конкретных методических приемов разъяснения учебного материала, стереотип FMT следует отнести к стереотипу metaclass из списка стереотипов UML, так как элементами FMT как класса являются также классы ФЭО, решающие различные методические задачи, и объединяемые общими атрибутами. Определяя стереотип FMT, приходится учитывать то, что, являясь, прежде всего, представителем базового стереотипа class, он может рассматриваться и как представитель других базовых стереотипов. Например, в иерархии конкретной модели обучения он обязательно является компонентом этой модели, т.е. может быть отнесен к базовому

стереотипу Component со всеми причитающимися ему атрибутами, операциями и обязанностями. Как средство общения с пользователем FMT проявляет принадлежность к базовому классу Interface. По отношению к некоторому пакету, структурирующему конкретный процесс обучения, FMT может рассматриваться как атрибут этого пакета. При отнесении к каждому такому базовому атрибуту изменяется список атрибутов и самого FMT. Рассмотрим пока только «инвариантные» атрибуты стереотипа FMT, вытекающие из общей структуры обучающей системы и являющиеся принадлежностью любого подкласса метакласса FMT.

2. Базовые атрибуты FMT.

Атрибут **activation** – стереотип, обозначающий способ активации (приведения в рабочее состояние) функционального элемента обучения. Значения этого атрибута можно определить по-разному: просто описывая способ активизации словами – строковыми переменными, или присваивая способам активации какие-либо численные значения. Начальное значение этого атрибута может соответствовать как активному состоянию (введем стереотип Active), так и в не активизированном – пассивном – состоянии (напрашивается соответствующий стереотип Passive). Используемые в настоящее время методы активизации функциональных элементов обучения связаны, как правило, с манипуляцией мышью. В силу этого атрибут activation может принимать следующие значения: on Mouse – активизируется при прохождении над ним курсора мыши, on click left Mouse – активизируется щелчком по левой клавише мыши при наведении курсора на активизируемый объект, on click Right Mouse – активизируется щелчком по правой клавише мыши. Так как последнее действие используется, как правило, для вызова контекстного меню, то и по отношению к рассматриваемому атрибуту его лучше использовать в том случае, когда последующее развитие событий зависит от некоторого выбора, предоставляемого раскрывающимся меню. Возможен также сложный набор способов активации, например, стереотип Before, означает некоторую предварительную активацию ФЭО – изменение цвета, размера, выпуклости или других видимых характеристик разъясняемого символа, указывающее на то, что к нему может быть применено следующее действие (интуитивно – это щелчок левой клавишей), приводящее в рабочее состояние механизм разъяснения или что-либо подобное. В этом случае со стереотипом Before должен быть связан стереотип After start, наличие которого означает последующую активацию невидимого ФЭО.

Атрибут **Circle of attention** – круг внимания. Определяет положение ФЭО на экране. Для реализации этого стереотипа экран следует разделить на круги, соответствующие распределению внимания по плоскости экрана монитора. Форма экрана соответствует разделению экрана на квадраты, которые в зависимости от размера экрана можно идентифицировать, например, с помощью буквенных и числовых значений. Экран от 14" до 20"-21" (наиболее распространенные размеры мониторов) имеет смысл разделить на 9 прямоугольников, снабдив их буквенно-цифровыми идентификаторами, например, a,b,c – по горизонтали

и 1, 2, 3 – по вертикали. Тогда первый круг внимания будет состоять из одного прямоугольника (b2), второй – из прямоугольников (a2), (b1), (c2) и (b3). Третий

круг внимания образуют прямоугольники (a1), (c1), (c3) и (a3) (рис. 1).

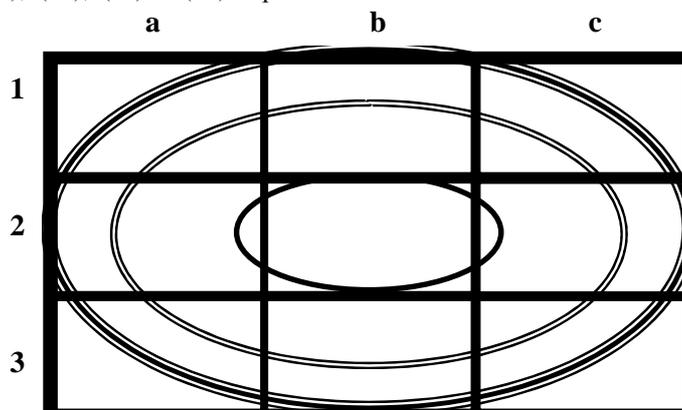


Рисунок 1. Круги внимания

Эти буквенно-цифровые идентификаторы и представляют собой значения атрибута круг внимания, указывающего на степень важности активного ФЭО на данном этапе его использования. Если ФЭО в рассматриваемый на статической диаграмме момент на экране отсутствует, то этот атрибут принимает нулевое значение. В процессе развития процесса управления обучением значение круга внимания может, а, как правило, и должно изменяться, порождая тем самым одну из операций над функциональными элементами обучения.

Атрибуты *Altitude* – высота и *Width* – ширина. В зависимости от назначения ФЭО, решаемых с его помощью задач, эти атрибуты могут варьироваться в достаточно широких пределах, но чаще всего имеет смысл использовать две крайние ситуации. Если задача обучения требует полного переключения внимания с полным исключением исходного объекта из кругов внимания, тогда целесообразно совместить окно ФЭО со всей экранной плоскостью. Если же понимание изучаемого материала требует поддерживать связь с «материнским» окном, то рассматриваемые атрибуты должны принимать значения 1/3 высоты и ширины экрана монитора, занимая один из прямоугольников в круге внимания.

Перечисленные атрибуты должны одновременно рассматриваться с точки зрения их отношения к еще одному важному стереотипу **through Operating** – сквозное действие. Как атрибут ФЭО сквозное действие определяет принадлежность к последовательности действий, направленных на решение сверхзадачи – *Super problem* – методической задачи, определяемой моделью специалиста и отношением рассматриваемой задачи ко всему базовому набору знаний, умений и навыков, определяющих профессиональную подготовку специалиста. Помимо такого глобального значения сверхзадачу можно рассматривать и в локальном смысле – это основные сведения, которые должны быть закреплены в памяти учащегося, для освоения им других учебных курсов. Значения, которые должно принимать сквозное действие как атрибут ФЭО в самом примитивном его понимании (принадлежит или не принадлежит сквозному действию) можно ограничить двумя: 1 и 0 или же булевыми true

и false. В случае принадлежности сквозному действию ФЭО наделяется дополнительными атрибутами, определяемыми уже сквозным действием как операцией – последовательностью действий. Стереотип **through Operating**, отнесенный к базовому стереотипу **Operating** – операция, порождает множество стереотипов, часть из которых являются атрибутами ФЭО.

Атрибут **Line** – стереотип, тесно связанный с атрибутом «круг внимания», определяющий расположение функционального элемента относительно осевой линии long-кванта. Естественно присваивать этому атрибуту значения 0, 1 ... Нулевое значение атрибута присваивается элементу, расположенному на осевой линии long-кванта, 1 – на линии первого cross-кванта и т.д.

Атрибут **Behavior** – поведение. В настоящее время нами рассматриваются три возможных значения этого стереотипа: **static**, **stationary** и **dynamic**. Стереотип **static** определяет статические элементы обучения, не изменяющие своего вида в процессе обучения. Стереотип **stationary** в нашем понимании определяет функциональные элементы обучения, изменяющиеся только вследствие определенных действий, таких, как **on Mouse**. прекращение действия возвращает ФЭО в исходное статическое состояние. Стереотип **dynamic** отражает внутреннюю динамику описываемых событий, выстраивая их в определенной временной последовательности. Таким образом, стереотипы **static** и **stationary** предназначаются лишь для того, чтобы активизировать внимание учащегося по отношению к данному ФЭО, стереотип же **dynamic** раскрывает движение, содержащееся в излагаемом учебном материале. Начальное значение атрибута **Behavior** динамического элемента может быть **stationary**, так как активизация его динамических свойств связана с потребностью учащегося знакомства с этими свойствами. Атрибут **Behavior** следует рассматривать как наиболее динамичный атрибут из всех перечисленных. Практика разработки обучающих систем, подчиненная более точным методическим и педагогическим задачам, чем это делалось до сих пор, должна постоянно вносить изменения в содержание этого стереотипа.

Visibility – видимость атрибутов. Все рассмотренные атрибуты, кроме атрибута **activation** могут иметь видимость **private**, т.е. распознаваться только предназначенной для их обработки программой. Атрибут, определяющий активацию ФЭО, должен быть доступен, прежде всего, пользователю, находящемуся в заранее определенной области обучающей системы, поэтому значение этого стереотипа должно быть **public** – общедоступный. Естественно, что этому атрибуту соответствует некоторый видимый на экране монитора указатель на активацию. Причем этот указатель должен явно восприниматься учащимся как активатор ФЭО. В противном случае учащийся может пройти мимо возможности активации, даже, если нуждается в дополнительных пояснениях. Этот недостаток имеет место, например, в обучающих программах по физике фирмы 1С. Имеющиеся в общей инструкции указания не могут заменить видимость активатора в самом контексте. Кроме того, как известно, инструкции, введения и прочие обобщающие вспомогательные элементы учащимися, как правило, вообще пропускаются.

Рассмотренные стереотипы ни в коем случае не исчерпывают все возможные атрибуты, определяющие подклассы класса FMT. Каждый конкретный класс FMT, реализуемый конкретными средствами (часто он является результатом использования нескольких программных продуктов) обрастает дополнительными атрибутами, наследуя лишь основные атрибуты метакласса FMT.

Индивидуализация (или адаптивность) автоматизированной обучающей системы должна, на наш взгляд, определяться детальной разработкой большого набора функциональных элементов обучения, осуществляющих коррекцию воображения и базовых знаний учащегося. Сами же функциональные элементы обучения строятся исходя из законов восприятия и методических и педагогических задач процесса обучения.

СПИСОК ЛИТЕРАТУРЫ

1. Рыкова Е.В., Рыков В.Т. Функциональный элемент обучения как дидактическая единица ИОС. //Успехи современного естествознания № 3, 2005, с. 96
2. Буч. Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. – М.: ДМК Пресс, 2005. – 432 с.
3. Рыков В.Т., Рыкова Е.В. Иерархия базовых стереотипов UML-модели обучающей системы. //Электронная конференция «Компьютерное моделирование в науке и технике», октябрь 2005, – www.congressinform.ru/articles/start.php
4. Рыкова Е.В., Рыков В.Т. Компьютерные обучающие системы и информационные потоки //Успехи современного естествознания № 3, 2004, с. 87

ДИНАМИКА КРИПТОГРАФИЧЕСКИХ ПРЕОБРАЗОВАНИЙ И РАСШИРЕНИЕ КЛАССА ФУНКЦИОНАЛЬНЫХ ЭЛЕМЕНТОВ ОБУЧЕНИЯ

Рыков В.Т., Шевченко С.В.

*Кубанский государственный университет,
Краснодар*

Динамические функциональные элементы обучения [1] получили большое распространение в коммерческих обучающих программах, иллюстрируя, прежде всего, динамику физических процессов в виде графиков соответствующих величин. Построение графика под наблюдением, а иногда и управлением учащегося одновременно с имитацией реального физического процесса развивает воображение и позволяет «прочувствовать» процесс. Эта динамика очевидная, так сказать, «открытая». Однако часто динамического представления требуют и внешне статичные объекты, отражающие внутреннюю динамику математических преобразований.

К таким объектам относятся алгоритмы различных криптографических преобразований. Будучи представлены статическими изображениями последовательности действий над числами, они значительным числом студентов (превышающим, как правило, 50 %) и воспринимаются как не изменяющиеся формы. Знакомые очертания самих форм создают ощущение сиюминутного понимания, скрывая от самих студентов и преподавателя реальное непонимание сущности алгоритма. Это непонимание открывается преподавателю, как правило, только на этапе проверки умения применить полученные знания на практике. Между тем большинство нетривиальных криптоалгоритмов осуществляющих постоянное движение чисел по непрерывно изменяющимся траекториям.

В том случае, если траектории движения чисел не изменяются, имеет смысл использовать стационарные функциональные элементы обучения, отображающие постоянные траектории, содержание элементов которых начинает изменяться в соответствии с заданным алгоритмом (например, скремблер) по команде учащегося, реализация которой заложена в кнопках «старт» и «стоп». С помощью аниматора Flash MX 7.0 алгоритм движения чисел реализуется в виде последовательных перемещений и превращений чисел в соответствии с исходным текстовым файлом.

Более сложная динамика алгоритма Хаффмана, реализуемая в виде дерева с изменяющимся строением ветвей, оказывается трудной для восприятия без соответствующих пояснений и многократных повторений. Это заставляет расширить число базовых атрибутов функционального элемента обучения, путем введения стереотипов Sound broadcasting – звуковое сопровождение и speaker – диктор, уточняющий, что звуковое сопровождение должно носить характер дикторских пояснений.

Усложняется и режиссура данного функционального элемента. На наш взгляд имеет смысл разъясняемые диктором действия демонстрировать дважды: до дикторского текста и сразу после него. В спецификации соответствующих элементов можно добавить стереотип Cycle – цикл, указывающий на наличие