комплексе, имеют вид анимационных или художественных головоломок, которые побуждают детей активно использовать аппарат мыслительной деятельности. В частности анализировать, сравнивать, синтезировать, обобщать те или иные процессы, которые отражают задания. При этом элементарными операциями для выполнения заданий являются: перенос объектов экрана на свои позиции, поворот этих объектов, изменение цвета объекта, выбор верного ответа или запись ответа в текстовое поле, выделение определенной части экрана.

Данная система создана специально для сайта дистанционного обучения. Она состоит из двух частей: среда создания заданий (в дальнейшем конструктор) и интерпретатор заданий. В конструкторе можно добавлять стандартные элементы, а также создавать собственные. Для каждого элемента задания существует свой образ, который указывает верную позицию при выполнении задания. Элементы настраиваются с помощью специальных панелей конструктора. В итоге конструктор выдает текстовую информацию, которая отражает настройку задания. Данный текст копируется в специальный текстовый файл и отправляется на сайт дистанционного обучения. В момент выбора определенного задания на странице сайта запускается интерпретатор, который считывает загруженный текстовый файл и в соответствии с настройками создает вид задания.

Разработанный программный продукт позволяет в короткое время создавать комплексы развивающих заданий, тестовых заданий. Авторами заданий могут быть не только учителя, но и дети. Таким образом, данная система позволяет развивать творчество учащихся. Для подготовки авторов заданий на сайте дистанционного обучения размещена библиотека видео уроков по созданию разнообразных заданий с использованием данного программного продукта. Глобальным курсом развития системы создания заданий является внедрение в нее базы данных, которая упростит и ускорит процесс создания заданий.

РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ

В. В. Романенко *(УО «ГГУ им. Ф. Скорины»)*
Научн. рук. ***С. Ф. Маслович,***
ассистент

Имитационная модель (ИМ) вычислительного процесса в МВС состоит из следующих элементов: задачи, поступающие в МВС для обработки и сама МВС. Задачи бывают пакетного и диалогового типов, причем диалог имеет больший приоритет.

Для исследования работы МВС необходимо описание характеристик РН: вектор параметров $\{X_{PH}\}$ и характеристики структуры этой РН $\{G_{PHi}\}$. Структура запросов каждого i-го типа задач (диалоговый и отложенный счет) является уникальной и определяется путем ранее проведенных экспериментов. Управляющими переменными моделирования являются: интенсивность поступления в узел МВС задач диалогового типа и отложенного счета $\lambda_{ДИАЛ}$ и $\lambda_{ОТЛ.\ СЧЕТ}$ соответственно, приоритеты π^1 определяющие важность характеристик задач отложенного счета при отправке их на обработку на устройства (CPU, RAM); количество CPU_i в МВС. Задаваемыми параметрами моделирования являются: скорость CPU_i (υ_{CPU}) задаваемую размером кванта времени, выделяемого j-ой задаче РН; скорость выполнения запросов на RAM (υ_{RAM}) также определяемую размером кванта времени, выделяемого для j-ой задачи РН.

При реализации ИМ МВС воспользуемся библиотекой Enterprise Library системы AnyLogic. Разрабатываемый класс Transact является расширением стандартного класса Entity. Entity является базовым классом для всех *заявок,* которые создаются, работают с ресурсами и принимают участие в процессе. Заявка – все, что может являться объектом, для которого задан какой-то процесс. Для генерации транзактов воспользуемся активными

объектами Source. Зададим для них функции генерации заявок диалогового пакетного типов $f(\lambda_{ДИАЛ})$ и $g(\lambda_{ОТЛ.\ СЧЕТ})$ соответственно. Соединим генераторы транзактов с очередью (объект класса Queue) при помощи *соединителей.* Соединитель – это графический элемент, соединяющий друг с другом интерфейсные элементы объектов – порты или переменные. Объект Queue моделирует очередь транзактов, ожидающих приема устройствами (CPU и RAM). Задаем для Queue дисциплины выбора из очереди. Процессоры и оперативную память будем моделировать используя объект Delay. Он задерживает заявки на заданный период времени (квант времени выделяемый на обработку транзакта). Объект класса Sink уничтожает поступившие заявки. Обычно используется в качестве конечной точки потока заявок.

ИНТЕГРАЛЬНЫЙ АЛГОРИТМ АНАЛИЗА КИНЕТИЧЕСКИХ КРИВЫХ

Д. С. Рыбалко (УО «ГГУ им. Ф. Скорины»)
Научн. рук. ***В. И. Кондратенко,***
ст. преподаватель

Задача разложения кинетических и сводящихся к ним кривых является актуальной и востребованной при анализе переходных процессов во временной и пространственной области. Особый интерес она представляет в связи с необходимостью осуществления обратного дискретного преобразования Лапласа (ОДПЛ) над массивами данных, где непосредственная операция ОДПЛ нереализуема ввиду необходимости получения комплексного продолжения дискретно заданной действительной функции. В настоящей работе рассмотрена реализация представления дискретно заданной функции совокупностью конечного числа экспоненциальных базисных функций, на основании интегрального алгоритма, сводящегося к решению системы линейных алгебраических уравнений.
Пусть

$$f(t) = \sum_{i=1}^{n} a_i e^{-p_i t}$$

Проинтегрируем $f(t)$ в пределах от нуля до бесконечности. Значение полученного интеграла определяется параметрами представления функции. Если мы произведем аналогичную операцию над $f(t)t^n$, то, как нетрудно показать,

$$F_n = \int_0^x t^n f(t)dt = \sum_{i=1}^{n} \frac{a_i}{p_i^{\,n}}$$

Повторяя данную операцию n раз, можно придти к системе линейных уравнений вида:

$$\begin{cases} F_1 = \sum_{i=1}^{n} \dfrac{a_i}{p_i} \\ F_2 = \sum_{i=1}^{n} \dfrac{a_2}{p_i^{\,2}} \\ \ldots \\ F_j = \sum_{i=1}^{n} \dfrac{a_i}{p_i^{\,j}} \\ \ldots \\ F_n = \sum_{i=1}^{n} \dfrac{a_i}{p_i^{\,n}} \end{cases}$$