

## СИСТЕМА ISI ДЛЯ ВИЗУАЛИЗАЦИИ ПРОЦЕССА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

А.В. Жуков, В.Ф. Мацула (Калининград)

В настоящее время существует множество программных средств создания имитационных моделей. Однако проблемой некоторых из них является отсутствие возможности наглядной демонстрации хода имитации [1]. Такие пакеты, как Proof Animation [2], разработанная компанией Wolverine Software еще в 1989 г., позволяют устранить данную проблему. Одним из преимуществ Proof Animation является то, что она может взаимодействовать с разными средами моделирования через генерацию ими трассировочного файла. Однако она имеет устаревший интерфейс, устаревшую графику и несовместимость с современными ОС.

По приведенным причинам актуальной является задача разработки современной системы, обеспечивающей визуализацию процесса имитационного моделирования.

Разработанная система получила название ISI, что расшифровывается как *I See It* (англ. Я Вижу Это). Общая схема её работы представлена на рисунке 1.

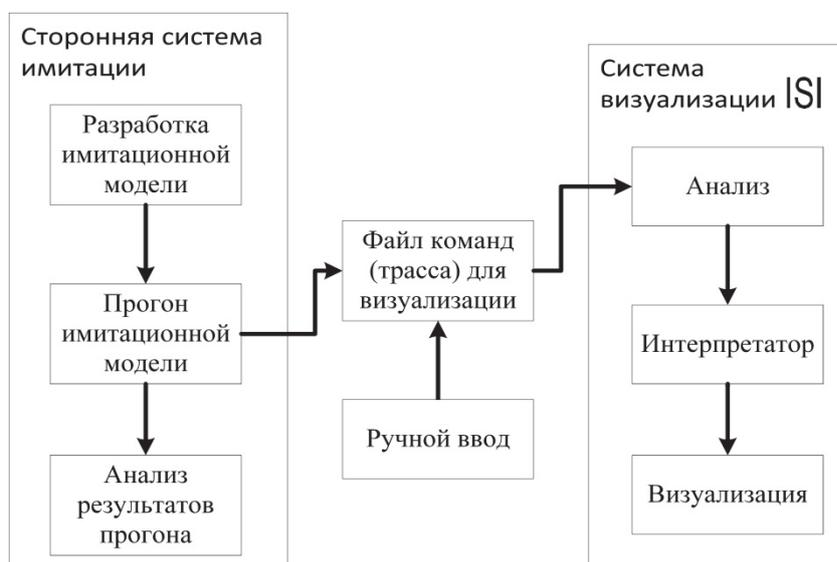


Рис. 1. Схема использования системы ISI

В процессе имитации, выполняемой в любой внешней по отношению к ISI системе моделирования, в требуемые моменты времени в текстовый файл (файл трассировки) должны записываться команды на специальном языке. Каждая команда указывает, когда, что и как должно быть визуальным образом отображено. После завершения моделирования содержимое файла трассировки подается на вход системы ISI, которая обрабатывает каждую команду, интерпретирует ее во внутренний код системы. Затем ISI формирует на экране монитора стилизованное анимационное изображение того, как проходил процесс моделирования. В любой момент можно остановить изображение и продолжить, когда это будет необходимо.

Для системы ISI разработан язык команд для описания визуализации в файле трассировки. В общем, каждая команда выглядит следующим образом (в форме Бэкуса-Наура) [3]:

**<command> ::= <Time><Object><Action>[<Arguments>]**, где:

*Time* – время начала выполнения команды в секундах. Обязательно для указания. Так же допускаются следующие варианты:

+*Time* – начало выполнения действия в момент, рассчитываемый как значения *Time* последнего явно определенного действия плюс *Time* текущего действия;

++*Time* – начало выполнения действия в момент, рассчитываемый как значение *Time* в предыдущем действии плюс *Time* текущего действия (в случаи первой строки, значение *Time* прибавляется к 0).

Значение *Time* должно быть вещественным положительным числом. В качестве разделителя используется точка. Все команды должны быть расположены в порядке возрастания времени.

*Object* – имя объекта над которым будет происходить действие (*Action*).

*Action* – действие, которое должен выполнить объект или выполнено над объектом. В системе ISI по умолчанию заложено только два действия: *Create* (создать объект) и *Destroy* (уничтожить объект). Все остальные действия подключаются посредством DLL, набор которых можно изменять в настройках системы. Стандартное DLL включает в свой состав такие функции, как:

- *Move* – прямолинейное движение;
- *Curemove* – движение по дуге;
- *Color* – изменение цвета/текстуры объекта;
- *Size* – изменение размера объекта, и другие.

*Arguments* – аргументы для действия. Содержание зависит от самого действия. В случаи прямолинейного движения это координаты в пространстве (X,Y,Z) и скорость или время (в этом случаи перед значением ставится английская буква ‘T’).

Ниже приведен фрагмент формального описания языка в форме Бэкуса-Наура [3].

```

<command> ::= <Time> <Object> <Action> [<Arguments>];
<Time> ::= [“+”]<number>;
<Object> ::= <word>;
<Action> ::= “create”|“move”|“curemove”|“destroy”|“color”|“size”;
<Arguments> ::= <klass>|(<x>”,”<y>”,”<z>”,”<speed>| (“T”<time>)|
<color>|<hight>;
<numeral> ::= “0”|“1”|“2”|...|“9”;
<number> ::= {/<numeral>/} [“.” {/<numeral>/}];
<letter> ::= “A”|“B”|“C”|...|“Z”|“a”|“b”|“c”|...|“z”;
<word> ::= <letter> {<letter>|<numeral>};
<klass> ::= “Cube”|“Sphere”|“Cone”|“HPlane”|“VPlane”|(<word>”.
“<format>);
<format> ::= “3ds”;
<x> ::= <number>;
<y> ::= <number>;
<z> ::= <number>;
<speed> ::= <number>;
<time> ::= <number>;
<color> ::= “White”|“Red”|“Green”|“Blue”|...|“Black”;
<hight> ::= <number>;

```

Каждый объект должен быть предварительно создан с помощью команды *Create*. В качестве аргумента используется или название стандартного типа объекта, например как *Cube* – куб, *Sphere* – шар, *Cone* – конус, *HPlane* – горизонтальная панель, *VPlane* – вертикальная панель, или некоторые другие, или полный путь к 3D модели (на данном этапе поддерживается модели в формате \*.3ds, \*.obj).

Помимо файла трассировки в системе *ISI* возможно создать и использовать файл объектов. В нем содержатся название типов объектов и та модель, которую они будут использовать (в качестве моделей могут выступать заранее созданные 3D модели, в этом случаи указывается путь к файлу, или стандартные примитивы: куб, шар, конус...). Это создано для того, чтобы пользователь мог изменить внешний вид 3D объекта в одном месте, не изменяя ее во всей трассе.

Команда в файле объектов выглядит следующим образом [3]:

**<ObjectModelCommand>::=<Type> <Model>**, где:

*Type* – имя типа объекта;

*Model* – путь к 3D модель.

В результате анализа программных средств и библиотек для работы с графикой в качестве графического движка был выбран *GLScene* — графический движок для создания приложений на языках программирования *Delphi*, *Free Pascal* и *C++*, использующий библиотеку *OpenGL* в качестве интерфейса программирования приложений. *GLScene* является свободным программным обеспечением и распространяется с лицензией *Mozilla Public License*. Последняя на данный момент версия – 1.5 от 2 января 2017 г. [4].

В качестве языка программирования был использован *C++*, как наиболее гибкий и оптимальный для данной задачи. В связи с выбором *GLScene* и *C++*, в качестве среды разработки была выбрана *Embarcadero RAD Studio 10.1 Berline*.

При разработке системы был использован объектно-ориентированный подход, и структура программы такова, что возможно создавать отдельные *DLL* библиотеки с действиями для визуализации (*Action*) и подключать их, что позволит с течением времени расширить функционал системы, не собирая её заново, а также используя сторонние решения других разработчиков. На рис. 2 представлен интерфейс системы *ISI*.

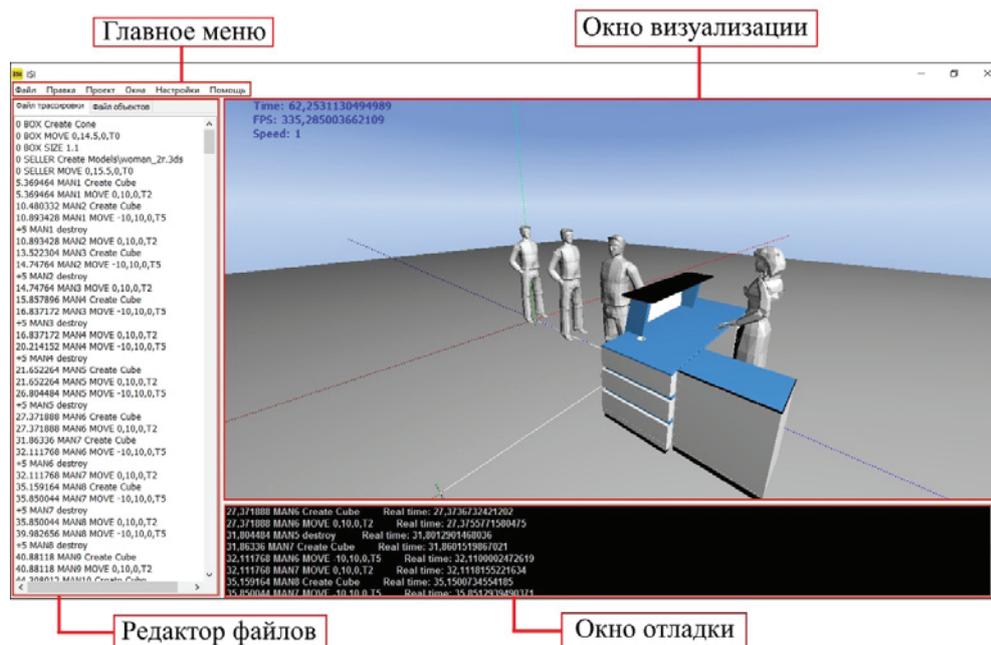
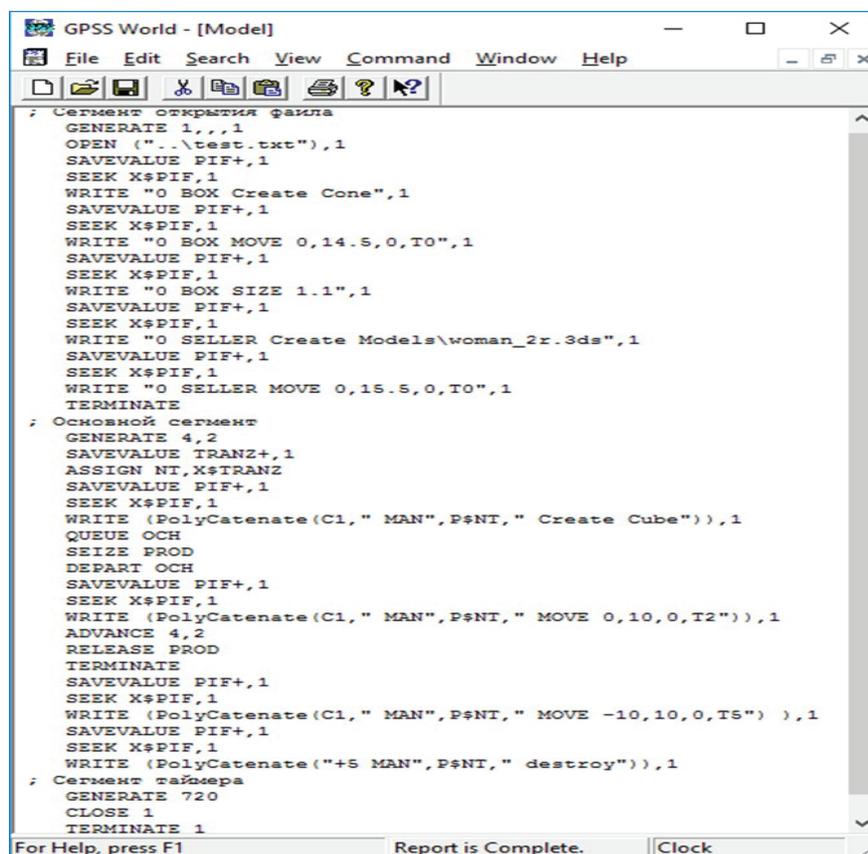


Рис. 2. Интерфейс системы *ISI*

Главное окно системы ISI разделено на три части: слева находится редактор файлов, в центре и справа – окно визуализации, снизу окно отладки, в котором отображается информация о прохождении визуализации и об ошибках. В окне визуализации отображено текущее время (Time), частота кадров (FPS) и скорость воспроизведения визуализации (Speed). Размер каждого из окон может быть изменен или окно можно скрыть.

Для перемещения камеры используются клавиши *WSAD* или стрелки как в большинстве привычных 3D редакторах. Движение осуществляется относительно направления взгляда камеры. Клавиши *UN* перемещают камеру вверх-вниз вдоль оси *OZ*. Зажав клавишу *Shift*, скорость перемещения будет увеличена в 2 раза. Клавишей *F3* запускается визуализация или запускается с начала, если уже была запущена. Клавишей *F2* устанавливается и снимается пауза. Клавиша *F4* – сброс визуализации в исходное состояние. Сочетание клавиш *Ctrl+G* и *Ctrl+H* соответственно замедляют и ускоряют воспроизведение визуализации.

Система ISI имеет возможность взаимодействовать со сторонними пакетами моделирования путем генерирования ими трассировочных файлов. Рассмотрим, как это происходит на примере системы GPSS World. На рис. 3 представлен фрагмент простой модели на языке GPSS World, имитирующей работу линии консультаций.



```

; Сегмент открытия файла
GENERATE 1,,1
OPEN ("..\test.txt"),1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE "O BOX Create Cone",1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE "O BOX MOVE 0,14.5,0,T0",1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE "O BOX SIZE 1.1",1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE "O SELLER Create Models\woman_2r.3ds",1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE "O SELLER MOVE 0,15.5,0,T0",1
TERMINATE

; Основной сегмент
GENERATE 4,2
SAVEVALUE TRANZ+,1
ASSIGN NT,X$TRANZ
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE (PolyCatenate(C1," MAN",P$NT," Create Cube")),1
QUEUE OCH
SEIZE PROD
DEPART OCH
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE (PolyCatenate(C1," MAN",P$NT," MOVE 0,10,0,T2")),1
ADVANCE 4,2
RELEASE PROD
TERMINATE
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE (PolyCatenate(C1," MAN",P$NT," MOVE -10,10,0,T5") ),1
SAVEVALUE PIF+,1
SEEK X$PIF,1
WRITE (PolyCatenate("+S MAN",P$NT," destroy")),1

; Сегмент таймера
GENERATE 720
CLOSE 1
TERMINATE 1

```

Рис. 3. Фрагмент текста модели в системе GPSS World

Первый сегмент модели описывает появление консультанта. Второй – возникновение и движение клиентов, обращающихся за консультацией. Для создания трассировочного файла используются команды *OPEN*, *SEEK* и *WRITE*.

Результатом работы программы на языке GPSS будет трассировочный файл объемом 708 строк. Ниже приведен отрывок из этого файла:

```
0 BOX Create Cone
0 BOX MOVE 0,13,0,T0
0 BOX SIZE 0.07
5.369464 MAN1 Create Cube
5.369464 MAN1 MOVE 0,10,0,T2
10.480332 MAN2 Create Cube
10.893428 MAN1 MOVE -10,10,0,T5
+5 MAN1 destroy
...
```

Рассмотрим назначение некоторых команд из этого фрагмента файла.

Команда 0 BOX Create Cone по наступлению 0-й секунды создает объект BOX. В качестве аргумента указывается тип объекта – Cone.

Команда 5.369464 MAN1 MOVE 0,10,0,T2 по наступлению 5.369464-й секунды переместит объект MAN1 из его текущего местоположения в точку (0,10,0) за две секунды.

Команда +5 MAN1 destroy по прошествии 5-ти секунд после последнего явно определённого времени (т.е. 10.893428) удалит объект MAN1. В дальнейшем это имя можно будет использовать повторно. Данная команда не имеет аргументов.

Полученный в результате работы GPSS World файл трассировки подается на вход системы ISI, а далее уже выполняются описанные ранее действия.

Также для большей наглядности был разработан файл объектов *Cube Models\man\_3.3ds* и *CONE Models\counter.3DS*. Заранее подготовленные 3D модели переопределяют стандартные типы объектов.

Фрагмент динамического 3D изображения модели линии консультаций можно увидеть на рис. 2 – за столом находится консультант (устройство), к которому подходят посетители (транзакты) и в очереди ожидают обслуживания.

Система ISI находится на стадии разработки, текущая версия 0.5.6373.675. В дальнейшем планируется увеличить скорость и надежность ее работы, расширить функционал.

### Литература

1. Simulation Software Survey [Электронный ресурс]. URL: <http://www.orms-today.org/surveys/Simulation/Simulation.html> (дата обращения 08.09.2017).
2. Proof Animation [Электронный ресурс]. URL: <http://www.wolverinesoftware.com/ProofProducts.htm> (дата обращения 08.09.2017).
3. Теория языков [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/26/26/lecture/801?page=7> (дата обращения 05.09.2017).
4. GLScene [Сайт]. URL: <https://ru.wikipedia.org/wiki/GLScene> (дата обращения 08.09.2017).