

К ВОПРОСУ РАЗРАБОТКИ СРЕДСТВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

**А.Ю. Лабинский, кандидат технических наук, доцент.
Санкт-Петербургский университет ГПС МЧС России**

Рассмотрены особенности создания компьютерных систем имитационного моделирования. Рассмотрены возможности разработанной компьютерной системы имитационного моделирования, созданной на алгоритмическом языке C# и позволяющей создавать имитационные модели в виде консольных программ на языке C#.

Ключевые слова: алгоритмический язык C#, имитационная модель, компьютерная программа

THE PROBLEM OF DEVELOPMENT THE MEANS OF SIMULATION MODELING

A.Yu. Labinskiy. Saint-Petersburg university of State fire service of EMERCOM of Russia

This article presents the problem of development the means of simulation modeling. The centre of attention use descriptions of development the computing system of simulation modeling on algorithmic language C#.

Keywords: algorithmic language C#, simulation model, computing program

«Имитационная модель – это формальное описание процесса функционирования исследуемой системы, взаимодействия её компонент во времени, учитывающее наиболее существенные причинно-следственные связи, присущие системе и обеспечивающие проведение компьютерных экспериментов [1]».

Процесс (алгоритм) функционирования системы моделируют в виде программы, реализуемой на ЭВМ. Для объекта, состоящего из множества элементов, разрабатываются алгоритмы работы как системы в целом, так и элементов системы.

В процессе имитационного моделирования используется модельное представление времени, которое дискретно и по которому организуется синхронизация событий в модели.

«Создание имитационной модели происходит в несколько этапов [1]:

1. Постановка задачи исследования объекта.
2. Построение концептуальной модели, которая включает описание параметров объекта-оригинала, описание показателей качества объекта, состав и способы представления исходных данных и результатов моделирования.
3. Построение математической модели, включая разработку алгоритма и моделирующей программы для ЭВМ. Компьютерная имитационная модель (программа для ЭВМ) может быть создана с помощью языков программирования (Basic, C++, C#, Fortran, Pascal и т.д.), языков моделирования (GPSS, Simula, Lisp и т.д.) и универсальных математических пакетов (MatLab, Mathcad, Maple).
4. Проверка работы и корректировка программы, включающая структурные и локальные изменения имитационной компьютерной модели.
5. Исследование модели, которое включает в себя исследование свойств имитационной модели и исследование с помощью имитационной модели объекта-оригинала.
6. Анализ результатов имитационного моделирования.

«Имитационное моделирование имеет следующие достоинства [2]:

– возможность детального описания функционирования компонентов системы;

- возможность исследования динамики системы, включая все возможные ситуации работы системы (в том числе аварийные режимы);
- возможность более эффективно выполнять исследование, проектирование и оптимизацию системы (процесса), так как работа с имитационной компьютерной моделью позволяет учитывать влияние множества различных параметров».

Компьютерная система имитационного моделирования

Компьютерная система имитационного моделирования (СИМ или Simulation Modeling System – SMS) содержит программу-оболочку пользовательского сервиса (ПОПС), обеспечивающую сервис при работе с шаблонами имитационных моделей, текстовыми и графическими файлами, содержащими исходные данные, результаты моделирования и справочную информацию.

Возможности и особенности разработанной СИМ следующие:

- возможность использования для разработки имитационных моделей базовых шаблонов, описывающих системы массового обслуживания [3] и системы автоматического управления (САУ) [4], использующих нечеткую логику [5, 6], нечеткую кластеризацию [6], нейронные сети [6, 7], генетические алгоритмы [6, 7] и методы решения дифференциальных уравнений [8];
- возможность непосредственной работы с шаблоном на популярном языке программирования C# в процессе разработки компьютерной имитационной модели;
- в процессе компиляции создается консольная программа, не требующая тщательной разработки графического интерфейса, что позволяет сосредоточиться на логике компьютерной имитационной модели;
- возможность изменения атрибутов объектов и глобальных переменных компьютерной имитационной модели;
- возможность использования математических выражений и функций;
- возможность получения случайных чисел с заданным законом распределения;
- возможность построения диаграмм и графиков, включая трехмерные графики;
- возможность пополнения набора базовых шаблонов имитационных моделей путем разработки шаблонов на языке C# в данной СИМ или в интегрированных средах разработки программного обеспечения, например, в Microsoft Visual Studio.

На рис. 1 представлен интерфейс ПОПС разработанной компьютерной СИМ в режиме загрузки базовых шаблонов на языке C#.

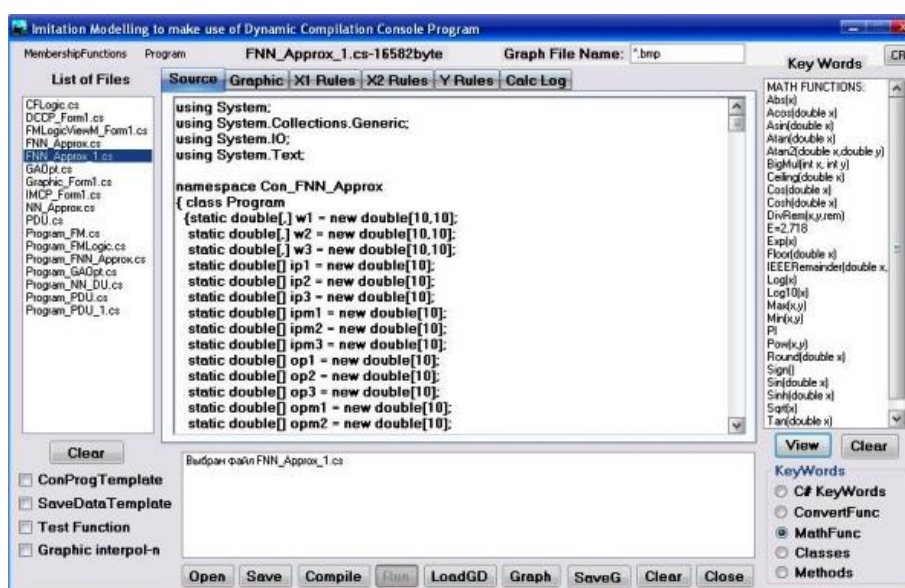


Рис. 1. Интерфейс ПОПС с открытой вкладкой «Source»

В левое окно программы выводится список файлов – шаблонов консольных приложений. Содержание выделенного файла из списка левого окна вставляется в центральное окно. В центральном окне осуществляется просмотр и редактирование выбранного шаблона. В правом окне производится просмотр по выбору списка ключевых слов языка C#, списка встроенных функций преобразования типов, списка встроенных математических функций, списка классов или списка методов классов. Выделенный элемент списка правого окна помещается в буфер обмена и может быть вставлен в нужное место шаблона в центральном окне. В нижнее окно выводится информация процесса компиляции файла, включая информацию об ошибках компиляции. С помощью меню «Templates» можно загрузить базовые шаблоны консольных приложений.

На рис. 2 представлена блок-схема событий интерфейса ПОПС модуля редактирования шаблона на языке C#, его компиляции и запуска исполняемого файла.

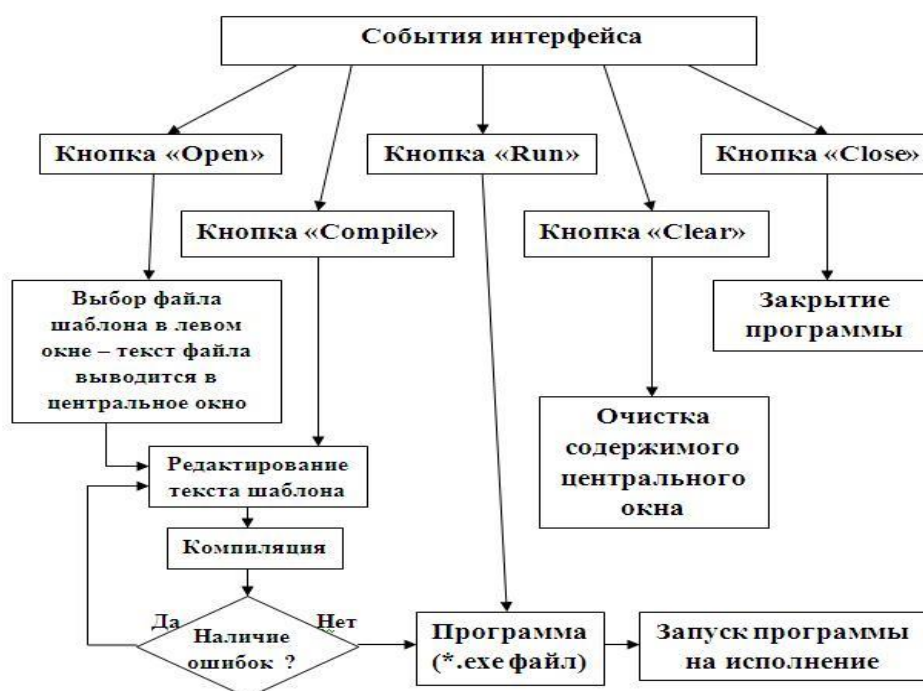


Рис. 2. Блок-схема событий интерфейса ПОПС

ПОПС создана с использованием языка программирования C# в среде Visual Studio фирмы Microsoft. ПОПС имеет модульную структуру и состоит из следующих модулей:

- модуль редактирования шаблона на языке C#, его компиляции и запуска исполняемого файла (консольной программы);
- модуль подготовки исходных данных для шаблона имитационной модели системы массового обслуживания (СМО);
- модуль подготовки исходных данных для шаблона имитационной модели сети массового обслуживания (СеМО);
- модуль подготовки исходных данных для системы нечеткого вывода;
- модуль загрузки базовых шаблонов на языке C# по системам и сетям массового обслуживания, системе автоматического управления, нечеткой логике, нечеткой классификации, нейронным сетям, генетическим алгоритмам и численным методам решения дифференциальных уравнений;
- модуль вывода исходных данных и результатов расчетов в текстовом и графическом (графики и диаграммы) виде и сохранения их на диске, соответственно, в виде файлов *.TXT, *.RTF и *.BMP;
- модуль просмотра справочной информации.

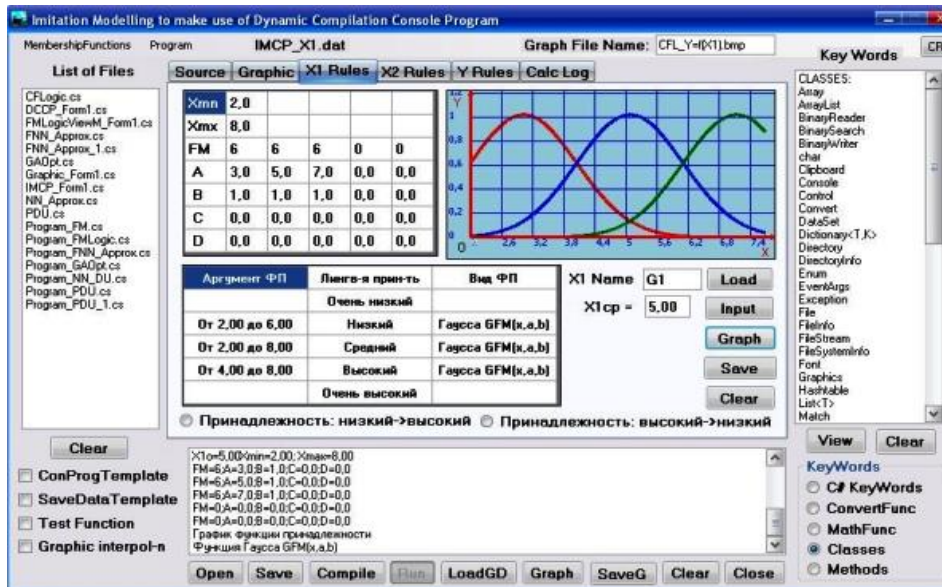


Рис. 5. Интерфейс ПОПС с открытой вкладкой «X1 Rules»

На рис. 6 представлен интерфейс ПОПС в режиме вывода результатов в графическом виде.

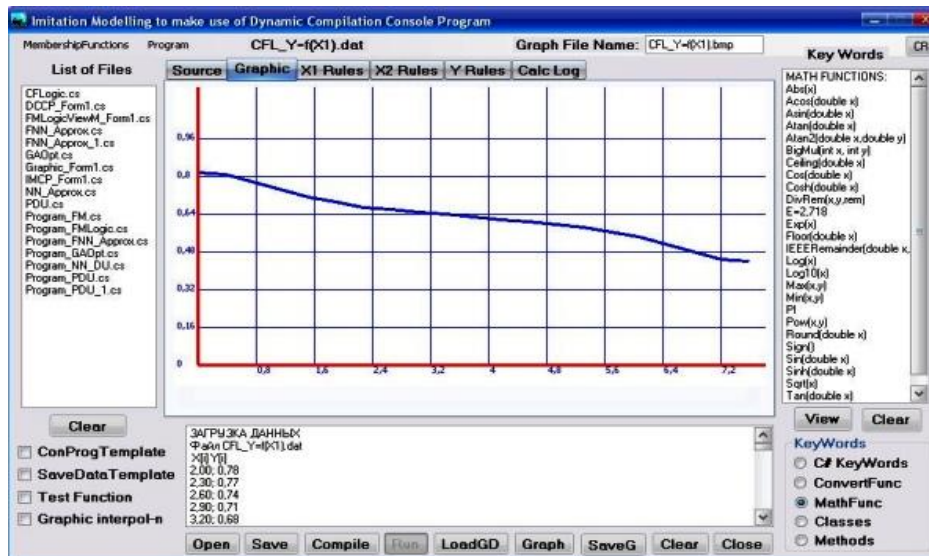


Рис. 6. Интерфейс ПОПС с открытой вкладкой «Graphic»

Модуль ПОПС подготовки исходных данных для СМО позволяет производить выбор типов законов поступления и обслуживания запросов из 10 типов законов распределения (Барра, Вейбулла, Лапласа, Парето, Релея, Эрланга и др.) на вкладке «Law Distrib». Далее задаются исходные данные СМО, которые сохраняются в файле на диске (вкладка «SysQ»).

Модуль ПОПС подготовки данных для СеМО позволяет производить моделирование сети массового обслуживания, в узлах которой расположены СМО с экспоненциальной плотностью распределения событий (пуассоновский поток). Моделируются замкнутые и незамкнутые СеМО, в узлах которых расположены одноканальные и многоканальные СМО с отказами и с ожиданием при ограниченной и неограниченной длине очереди ожидания. Переход заявок от узла к узлу имеет случайный характер и задается матрицей переходов СеМО. Модель оценивает вероятности состояний СеМО, среднее число заявок в каждом узле сети и вероятности незанятости СМО как узлов сети (рис. 4).

На рис. 7 представлен интерфейс ПОПС в режиме вывода текстовой и графической (блок-схема) информации в процессе загрузки базового шаблона по нейронным сетям.

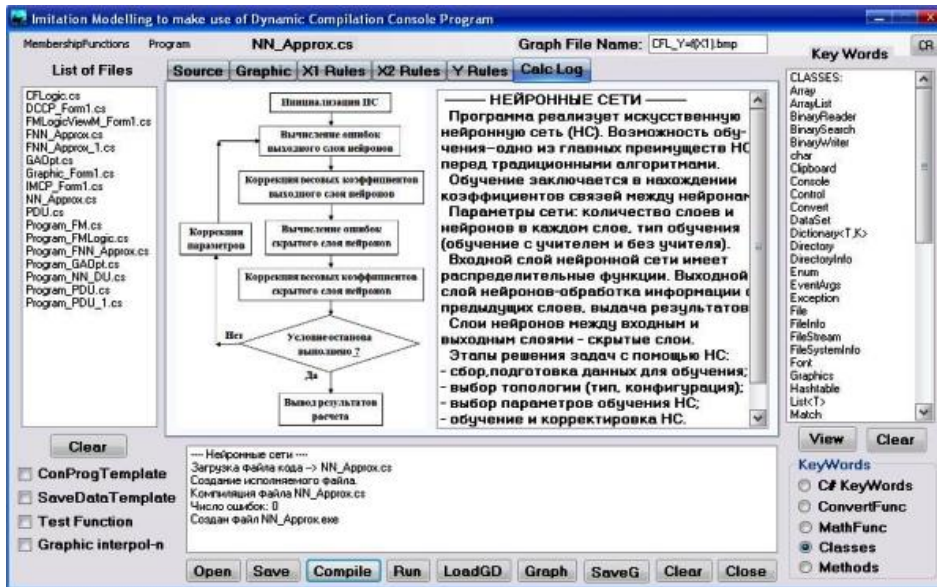


Рис. 7. Интерфейс ПОПС с открытой вкладкой «Calc Log»

Модуль ПОПС подготовки исходных данных для системы нечеткого вывода позволяет производить выбор типа функции принадлежности из представленных в графическом виде 9 типовых функций (меню «MembershipFunction»). Далее, исходя из особенностей моделируемого объекта, производится задание параметров функций принадлежности и создание базы продукционных правил с указанием диапазонов изменения аргумента и вида функций принадлежности, а также лингвистической принадлежности переменных (рис. 5).

Базовый шаблон компьютерной модели САУ описывает систему охлаждения и автоматическое управляющее устройство (система управления), которое управляет расходом охлаждающей жидкости. Модель реализует три режима управления: четкое управление, нечеткое управление и нейро-нечеткое управление. На рис. 8 представлен интерфейс ПОПС в режиме вывода графика изменения расхода охлаждающей жидкости при нейро-нечетком управлении. Исходные данные: значение начального уровня жидкости в баке $H_0=1,5$ м и ступенчатое изменение требуемого уровня жидкости H_z от 0,8 м до 1,2 м.

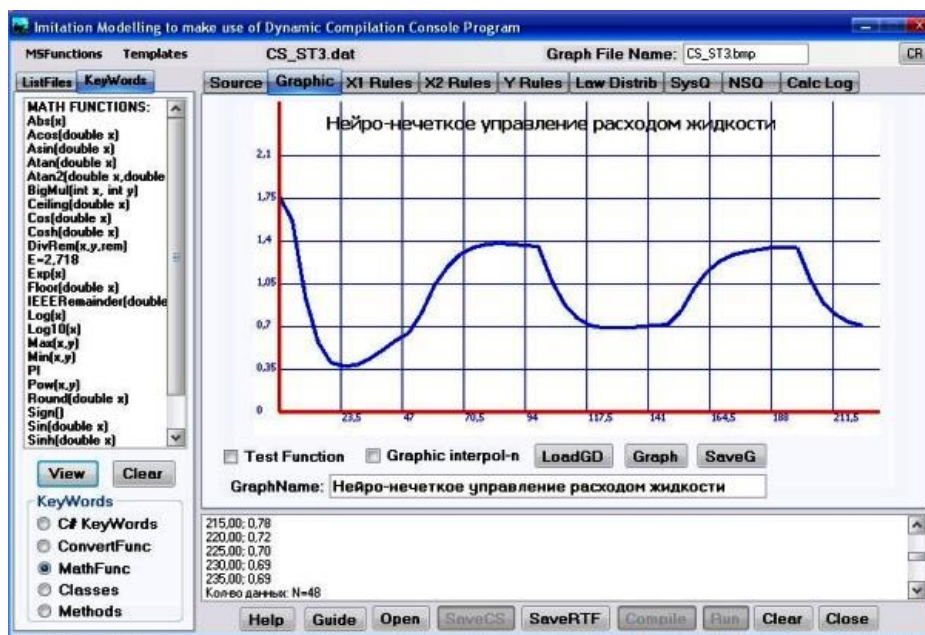


Рис. 8. Интерфейс ПОПС с открытой вкладкой «Graphic»

Базовый шаблон компьютерной модели нечеткой кластеризации описывает систему автоматической классификации множества объектов, которые задаются векторами признаков в пространстве признаков. В результате определяются кластеры и классифицируются объекты. Объекты принадлежат всем кластерам с определенной функцией принадлежности. Степень принадлежности объекта кластеру определяется расстоянием от объекта до центра кластера. На рис. 9 представлен интерфейс ПОПС с открытой вкладкой «FCGraph» с окном графического представления 16 объектов, объединенных в три кластера трех цветов.

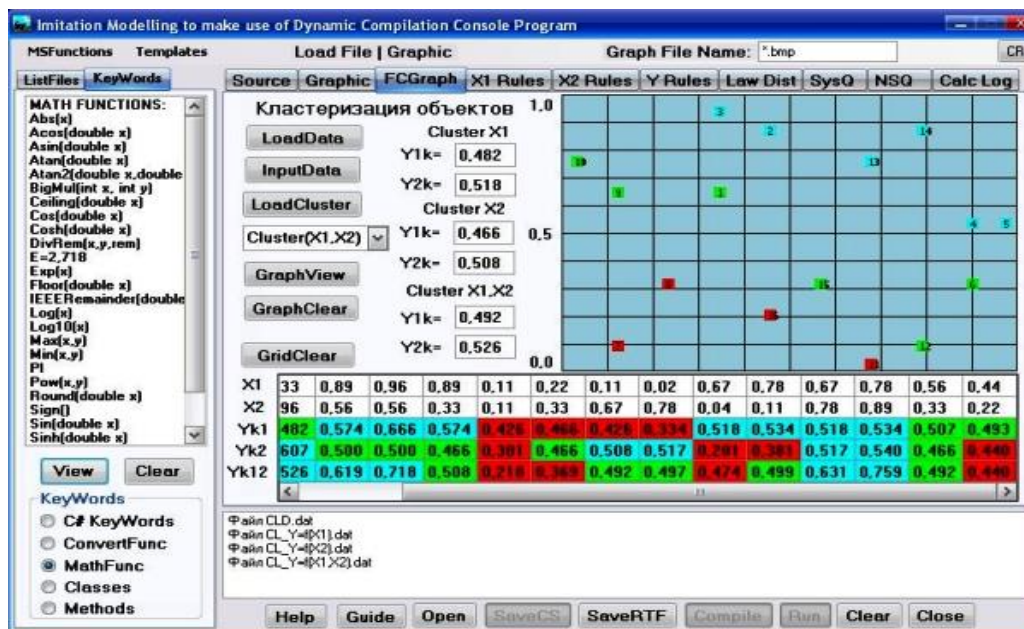


Рис. 9. Интерфейс ПОПС с открытой вкладкой «FCGraph»

В состав системы компьютерного моделирования SMS входит программа Graph3D построения графиков функций двух переменных $Y=f(X1, X2)$ в трехмерном виде. Пример построения трехмерного графика представлен на рис. 10.

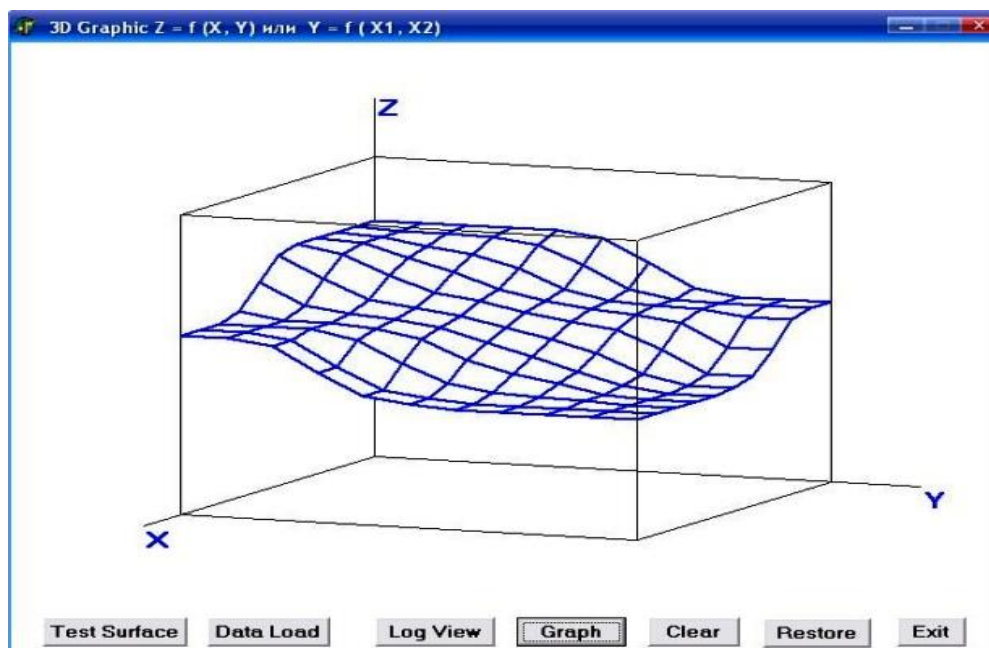


Рис. 10. Окно программы Graph3D построения трехмерного графика

Создание имитационной модели с использованием ПОПС, разработанной СИМ, происходит в несколько этапов:

- подготовка исходных данных (данные для модели с нечеткой логикой подготавливаются с помощью вкладок «X1 Rules», «X2 Rules», «Y Rules» с последующей записью данных в файлы IMCP_X1.dat, IMCP_X2.dat, IMCP_Y.dat на диске, данные для модели СМО подготавливаются с помощью вкладок «Law Distrib» и «Sys Q» с последующей записью данных в файл DSMO.dat на диске, данные для модели СеМО подготавливаются с помощью вкладки «NSQ» с последующей записью данных в файл DNSQ.dat на диске, для остальных моделей данные подготавливаются путем корректировки соответствующих шаблонов);

- ввод и корректировка базовых шаблонов (меню «Templates» или кнопка «Open», вкладка «Source», центральное окно);

- компиляция шаблона и запуск исполняемого файла консольной программы (кнопки «Compile» и «Run»), в случае наличия ошибок (нижнее окно) необходима коррекция текста шаблона (вкладка «Source», центральное окно);

- просмотр результатов расчета (нижнее окно, вкладка «Calc Log», просмотр графической информации – кнопки «LoadGD» и «Graph», вкладки «Graphic», «FCGraph», кнопки «Engage» и «Service», вкладка «Sys Queue»);

- сохранение текста отредактированного шаблона и результатов на диске – кнопки «Save» и «SaveG»;

- сохранение текста на языке C#, описывающего имитационную модель в виде консольной программы, в файле *.RTF – кнопка «SaveRTF».

Вывод: существует достаточно большое число компьютерных СИМ, позволяющих создавать имитационные модели с помощью различных языков программирования, языков моделирования, универсальных математических пакетов и специализированных пакетов программ. Разработанная компьютерная СИМ создает имитационные модели в виде консольных программ на языке программирования C#.

Литература

1. Советов Б.Я., Яковлев С.А. Моделирование систем: учеб. пособие для вузов. М.: Высш. шк., 2011.

2. Лоу А.М., Кельтон В.Д. Имитационное моделирование. СПб.: Питер, 2014.

3. Лабинский А.Ю. Моделирование системы массового обслуживания с использованием нейронной сети // Науч.-аналит. журн. «Вестник С.-Петербур. ун-та ГПС МЧС России». 2019. № 2. С. 52–56.

4. Лабинский А.Ю., Нефедьев С.А., Бардулин Е.Н. Использование нечеткой логики и нейронных сетей в системах автоматического управления // Науч.-аналит. журн. «Вестник С.-Петербур. ун-та ГПС МЧС России». 2019. № 1. С. 44–50.

5. Лабинский А.Ю. Моделирование системы нечеткого вывода // Природные и техногенные риски (физико-математические и прикладные аспекты). 2016. № 2 (18). С. 5–10.

6. Лабинский А.Ю. Перспективные направления компьютерного моделирования сложных процессов и систем: монография. СПб.: С.-Петербур. ун-т ГПС МЧС России, 2017.

7. Лабинский А.Ю., Подружкина Т.А. Особенности использования генетических алгоритмов и нейронных сетей // Природные и техногенные риски (физико-математические и прикладные аспекты). 2015. № 4 (16). С. 56–61.

8. Лабинский А.Ю. Моделирование случайного процесса с использованием принципа квазирегулярности // Природные и техногенные риски (физико-математические и прикладные аспекты). 2019. № 1 (29). С. 41–45.

References

1. Sovetov B.Ya., Yakovlev S.A. Modelirovanie sistem: ucheb. posobie dlya vuzov. M.: Vyssh. shk., 2011.
2. Lou A.M., Kel'ton V.D. Imitacionnoe modelirovanie. SPb.: Piter, 2014.
3. Labinskij A.Yu. Modelirovanie sistemy massovogo obsluzhivaniya s ispol'zovaniem nejronnoj seti // Nauch.-analit. zhurn. «Vestnik S.-Peterb. un-ta GPS MCHS Rossii». 2019. № 2. S. 52–56.
4. Labinskij A.Yu., Nefed'ev S.A., Bardulin E.N. Ispol'zovanie nechetkoj logiki i nejronnyh setej v sistemah avtomaticheskogo upravleniya // Nauch.-analit. zhurn. «Vestnik S.-Peterb. un-ta GPS MCHS Rossii». 2019. № 1. S. 44–50.
5. Labinskij A.Yu. Modelirovanie sistemy nechetkogo vyvoda // Prirodnye i tekhnogennye riski (fiziko-matematicheskie i prikladnye aspekty). 2016. № 2 (18). S. 5–10.
6. Labinskij A.Yu. Perspektivnye napravleniya komp'yuternogo modelirovaniya slozhnyh processov i sistem: monografiya. SPb.: S.-Peterb. un-t GPS MCHS Rossii, 2017.
7. Labinskij A.Yu., Podrzhkina T.A. Osobennosti ispol'zovaniya geneticheskikh algoritmov i nejronnyh setej // Prirodnye i tekhnogennye riski (fiziko-matematicheskie i prikladnye aspekty). 2015. № 4 (16). S. 56–61.
8. Labinskij A.Yu. Modelirovanie sluchajnogo processa s ispol'zovaniem principa kvaziregulyarnosti // Prirodnye i tekhnogennye riski (fiziko-matematicheskie i prikladnye aspekty). 2019. № 1 (29). S. 41–45.