

ФРЕЙМВОРК ДЛЯ ПОСТРОЕНИЯ И ИССЛЕДОВАНИЯ СЕТЕЙ ПЕТРИ И ИХ МОДИФИКАЦИЙ

А.Н. Ивутин, И.А. Страхов

Разработан фреймворк, представляющий стандартную функциональность сетей Петри. Данный фреймворк использован в разработке утилиты для моделирования сетей Петри-Маркова. Предложен вариант интерфейса программного обеспечения для моделирования сетей Петри-Маркова.

Ключевые слова: сети Петри-Маркова, моделирование, программное обеспечение, фреймворк, утилиты.

"Сети Петри" – собирательный термин, который включает в себя большое количество системных моделей, методов анализа, графического представления и условных обозначений, основанных на конкретных предположениях о мире обработки информации. Под сетью Петри понимается биграф вида

$$P = \{A, Z, I_A(Z), O_A(Z)\}, \quad (1)$$

где $A = \{a_1(a), \dots, a_j(a)\}$ – конечное множество позиций; $Z = \{z_1(z), \dots, z_j(z)\}$ – конечное множество переходов; $I_A(Z) = \{I_A(z_1(z)), \dots, I_A(z_j(z))\}$ – входная функция, отображающая переходы в конечные множества позиций $I_A(z_i(z)) = \{a_1[I_i(z)], \dots, a_j[I_i(z)]\}$; $O_A(Z) = \{O_A(z_1(z)), \dots, O_A(z_j(z))\}$ – выходная функция, отображающая переходы в конечные множества позиций $O_A(z_i(z)) = \{a_1[O_i(z)], \dots, a_j[O_i(z)]\}$. Множества позиций и переходов сетей Петри не пересекаются, т.е. $A \cap Z = \emptyset$ [1]. На рис. 1. приведен граф сети Петри.

Развитие теории сетей Петри привело к появлению многих разновидностей данных сетей. К ним относятся, так называемые, «раскрашенные» сети Петри. Созданные программные компоненты их использования получили широкое применение в различных прикладных программных комплексах (системах проектирования протоколов связи, программного обеспечения). Математический аппарат «раскрашенных» сетей Петри делает его применимым как при разработке систем, так и в процессе их верификации и внедрения.

Помимо «раскрашенных» сетей Петри широкое распространение получило другое расширение классической теории – временные сети Петри. Данный вид сетей наиболее адекватен системам, функционирующим в режиме реального времени.

Наиболее полно учет таких свойств, как стохастичность и параллельное развитие процессов, может быть осуществлен в моделях, ниже называемых сетями Петри-Маркова, в которых сочетаются аспекты, реле-

вантные случайным процессам в компонентах системы, и аспекты описывающие логику их взаимодействия. Сетью Петри-Маркова называется структурно-параметрическая модель, заданная парой

$$\theta = \{w, y\}, \quad (2)$$

где w – множество резидентных свойств (структурно-параметрические характеристики); y – множество вариационных свойств (характеристики состояния).

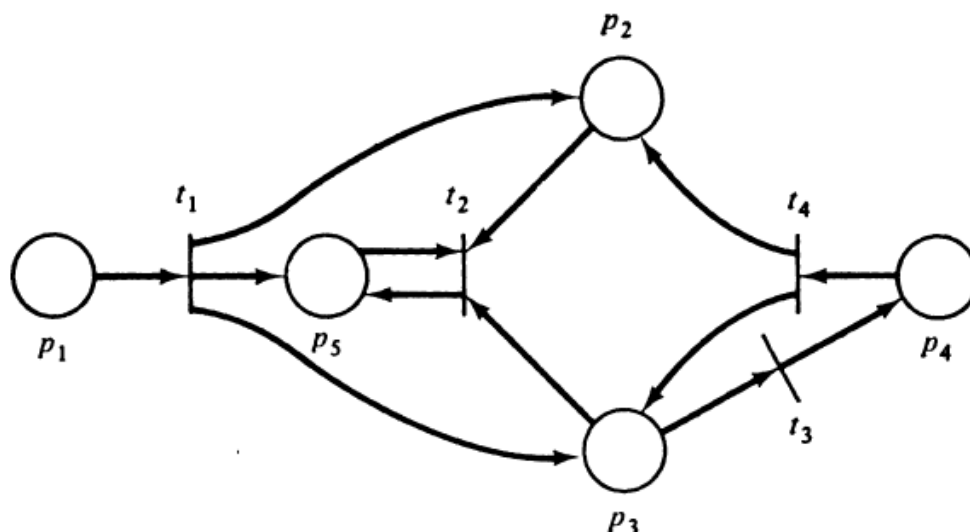


Рис. 1. Граф сети Петри

Резидентные свойства модели, в свою очередь, задаются парой

$$w = \{P, M\}, \quad (3)$$

где P – сеть Петри; M – случайный процесс.

Сеть Петри P определяет структуру сети, а случайный процесс накладывается на структуру и определяет временные и вероятностные характеристики [2].

Реализация инструмента для моделирования сетей Петри является трудоемким и дорогостоящим процессом. Большинство существующих проектов построения различных сетей Петри обладают схожими функциями и инструментами: графический редактор, средство моделирования и т. д., но они не позволяют создавать новые приложения. Основной целью создания фреймворка является обеспечение единого универсального интерфейса, позволяющего при минимальных затратах единообразным образом проводить построение и моделирование сетей Петри и их вариаций, а также предоставлять гибкие средства для построения приложений с различными визуальными интерфейсами. Данный подход позволит разработчикам сконцентрировать внимание на проектировании непосредственно алгоритма решения задачи, тогда как тонкости функционирования конкретного вида сети Петри будут решаться разрабатываемым фреймворком [3].

Однако существенной проблемой при создании программного продукта является невозможность разделить элементы графического интерфейса и логической части функционирования предметной области, для которой проектируется приложение.

Для того чтобы разработать библиотеку, необходимо выделить типовые формализмы, к которым относятся: вершины (позиций и переходы), а также ориентированные дуги, соединяющих их. Каждая из вершин обладает своей маркировкой – неотрицательным числом фишек, которые присваиваются вершине. Фишка – примитивное понятие, подобное позиции и переходу. В ходе выполнения сети, маркировка может меняться [4]. При этом нельзя соединять позиции с позициями и переходы с переходами.

Примем во внимание, что пользователями фреймворка могут быть не только опытные программисты. Поэтому его использование должно быть максимально простым: основная задача состоит в необходимости обеспечения такой функциональности, которая бы позволяла при минимальном вмешательстве со стороны разработчика создать готовое к конечному использованию приложение. Назовем такую структуру «чистой» сетью Петри.

На рис. 2 показана упрощенная архитектура приложения, созданного для конкретной разновидности сети Петри.

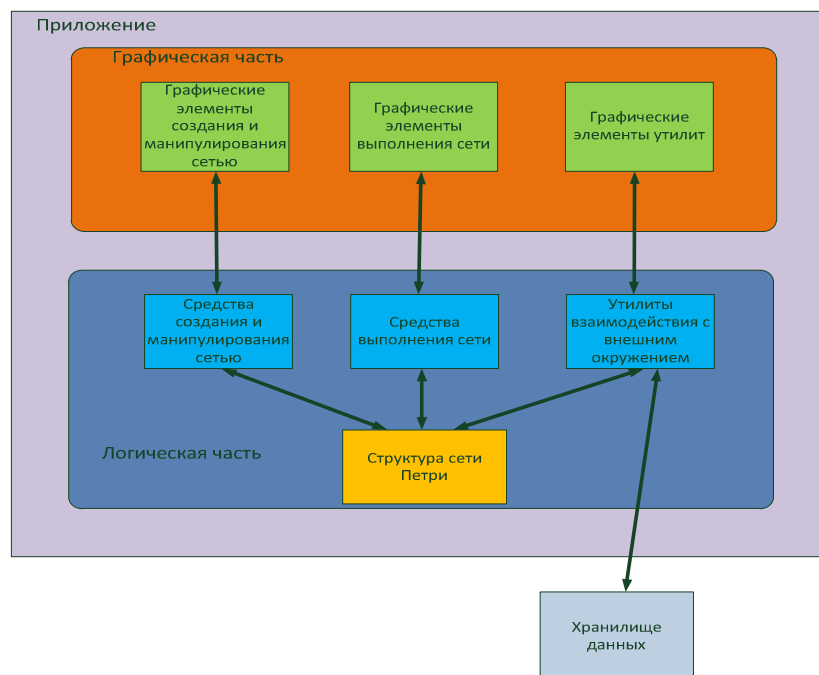


Рис. 2. Архитектура приложений

Фреймворк реализован на объектно-ориентированном языке программирования C# с платформой .Net Framework. Выбор языка разработки обусловлен следующими причинами:

- является кроссплатформенным языком программирования, кото-

- рый доступен практически на всех платформах;
- хорошо продумана семантика языка, что позволяет однозначно переносить объекты предметной области на соответствующие конструкции языка;
- имеет четкую концепцию для определения модулей и интерфейсов.

Структура «чистой» сети Петри состоит из нескольких классов, которые представляют граф сети вместе с разметкой. Основными классами являются PetriNetwork, PetriNode, PetriLink. Каждый класс предоставляет методы и свойства для доступа и изменения соответствующего элемента сети. Например, для объекта класса PetriNetwork, можно вызвать свойство Places, которое вернет список всех позиций в сетевой структуре. Кроме того, существуют методы и свойства, которые возвращают входные и выходные дуги, или же разметку элемента. У объекта класса PetriNode за это отвечает свойство Extension. Следует отметить, что все методы и свойства автоматически следят за целостностью данных. Например, нельзя задать отрицательную разметку для вершины. Также, при любых изменениях объекта, он (объект) сообщит о своих изменениях.

Покажем теперь один из вариантов графического интерфейса объектов сетевых структур. Платформа .Net включает в себя технологию Windows Presentation Foundation (WPF), позволяющую описывать графические элементы при помощи разметки XAML – eXtensible Application Markup Language (расширяемый язык разметки приложений). На рис. 3 приведен пример разметки для позиции.

```
<UserControl xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  mc:Ignorable="d"
  d:DesignHeight="35" d:DesignWidth="35" width="35" height="35" Background="Transparent">
  <Grid width="Auto" height="Auto" Background="Transparent">
    <Ellipse height="Auto" width="Auto" fill="pink" strokeThickness="5" stroke="black"/>
    <Label Name="countChipLabel" Content="" HorizontalAlignment="Center" VerticalAlignment="Center"/>
    <Ellipse Name="topSheet" height="Auto" width="Auto" fill="Transparent" strokeThickness="5" stroke="Transparent">
      <Ellipse.ContextMenu>
        <ContextMenu Name="contextMenu">
          <MenuItem Name="removeContextMenu" Header="удалить"/>
          <MenuItem Name="propertyContextMenu" Header="свойства" IsCheckable="False" />
        </ContextMenu>
      </Ellipse.ContextMenu>
      <Ellipse.ToolTip>
        <ToolTip>
          <TextBlock Name="toolTipTextBlock">hello</TextBlock>
        </ToolTip>
      </Ellipse.ToolTip>
    </Ellipse>
  </Grid>
</UserControl>
```

Рис. 3. XAML разметка позиции

Приведем пример реализации программного обеспечения для сетей Петри-Маркова. Для этого нам потребуется создать класс-наследник от класса PetriNetwork и перегрузить виртуальный метод [5] Modelling в соот-

ветствии с теорией сетей Петри-Маркова. Также, создадим классы-наследники от PetriNode и PetriLink и добавим в них свойства, характерные для выбранной разновидности сети. На третьем этапе спроектируем графический интерфейс приложения. В результате получили программу, показанную на рис. 4.

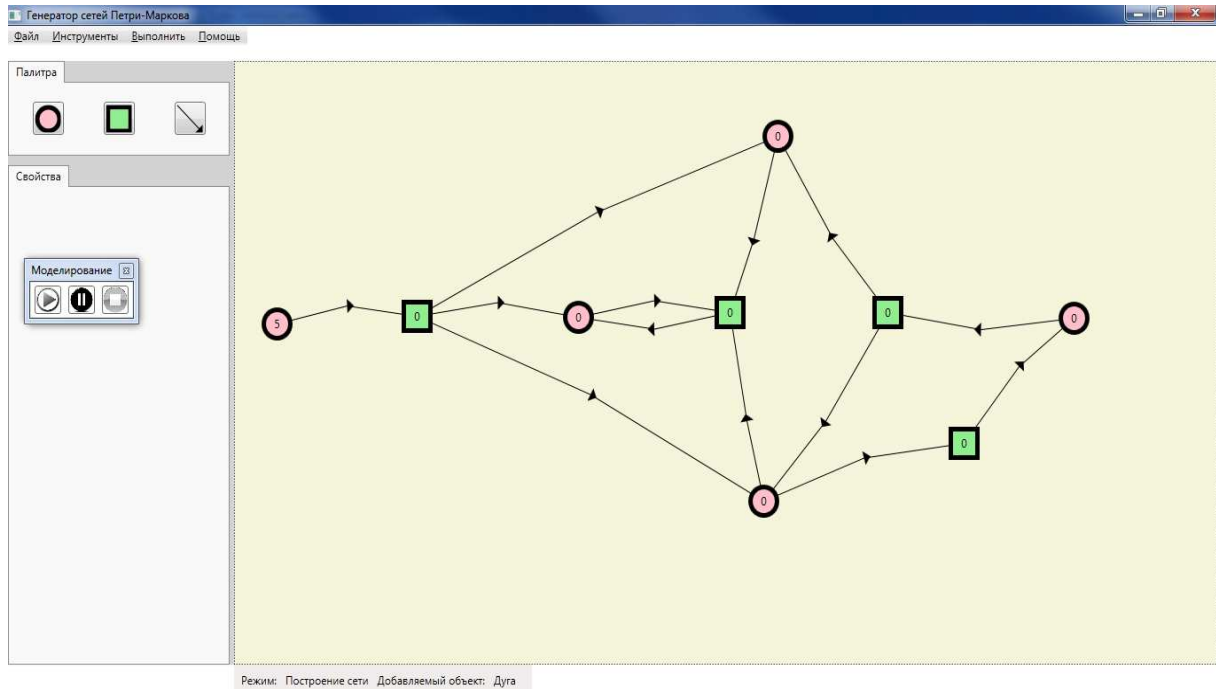


Рис. 4. Приложение для моделирования сетей Петри-Маркова

При работе программы, появляется необходимость сохранять и загружать сетевые структуры. Данный модуль также можно ввести в библиотеку. Благодаря платформе .Net, пользователям не придется переопределять поведение методов сохранения и загрузки. Сохранение будем производить в файлы с расширением *.xml на языке разметки xml соответственно.

Использование на практике разработанного фреймворка позволяет, как сократить время разработки приложений, так и уменьшить риск возникновения ошибки при создании программных продуктов. Одновременно с этим, реализация функционала «чистой» сети Петри дает возможность сократить затраты, связанные с обновлением приложений.

Список литературы

1. Котов В.Е. Сети Петри. М.: Наука. Главная редакция физико-математической литературы, 1984. 160 с.
2. Ларкин Е.В., Сабо Ю.И. Сети Петри-Маркова и отказоустойчивость авионики. Тула: Изд-во ТулГУ, 2004. 208 с.
3. Weber M., Kindler E., The Petri Net Kernel. Petri Net Technology for

Communication-Based Systems, 2003. P. 109-123.

4. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. М.: Мир, 1984. 264 с.

5. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0. М.: ООО «И.Д. Вильямс», 2011. 1392 с.

Ивутин Алексей Николаевич, канд. техн. наук, доц., alexey.ivutin@gmail.com, Россия, Тула, Тульский государственный университет,

Страхов Илья Андреевич, магистрант, iastrahov@gmail.com, Россия, Тула, Тульский государственный университет

PETRI NETS AND EXTENSIONS FRAMEWORK

A.N. Ivutin, I.A. Strahov

Developed a framework, which is a standard feature of Petri nets. This framework is used in the development of tools for the modeling of Petri-Markov nets. A version of the software interface of the simulation of Petri-Markov nets are proposed.

Key words: Petri-Markov net, simulating, software, framework, tools

Ivutin Alexey Nicolaevich, candidate of technical science, docent, alexey.ivutin@gmail.com, Russia, Tula, Tula State University,

Strahov Ilya Andreevich, undergraduate, iastrahov@gmail.com, Russia, Tula, Tula State University

УДК 004.658.6

ПОВЫШЕНИЕ НАДЕЖНОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ ПРИ ПОМОЩИ РЕПЛИКАЦИЙ

А.Н. Ивутин, И.С. Терехин

Описаны типы репликации (Синхронная репликация, Асинхронная репликация). Рассмотрены задачи, которые может решить метод репликации базы данных, а также недостатки использования метода. Обозначены проблемы, когда такой подход необходим.

Ключевые слова: репликация, база данных, производительность.

Ежегодно все больше сфер человеческой деятельности переходят на электронный вариант хранения данных. Для этого используются разнообразные системы управления баз данными (СУБД), такие как MSSQL, Oracle, MySQL и другие. Скорость работы современных реляционных