Міністерство освіти і науки, МОЛОДІ ТА СПОРТУ України

Одеський Національний Університет ім. І.І.Мечникова

Кафедра математичного забезпечення комп'ютерних систем

Программная реализация Высокоуровневых Сетей Петри.

Допускається до захисту

« » \_\_\_\_\_\_\_\_\_\_ 2012 р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / /

*Курсова робота*

студента IV курсу ФІТ

Поступальського Олега Сергійовича

*Науковий керівник:*

доцент, кандидат фіз.-мат. наук

Пенко В.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc293409902)

[1 Основные сведения о сетях Петри 4](#_Toc293409903)

[1.1 Определение простых сетей Петри 4](#_Toc293409904)

[1.1.1 Графическое представление 4](#_Toc293409905)

[1.1.2 Аналитическое представление. 6](#_Toc293409906)

[1.2 Назначение сетей Петри. 7](#_Toc293409907)

[1.3 Способы проектирования и анализа систем. 7](#_Toc293409908)

[1.4 Анализ существующих решений 9](#_Toc293409909)

[2 Описание программной реализации 12](#_Toc293409910)

[2.1 Библиотека MagicLibrary. 12](#_Toc293409911)

[2.1.1 Пространство имён MVC: 12](#_Toc293409912)

[2.1.2 Пространство имён MathUtils: 13](#_Toc293409913)

[2.1.3 Пространство имён Graphic: 16](#_Toc293409914)

[2.2 Программа GraphEditor. 17](#_Toc293409915)

[2.3 Программа Petri Net Editor. 19](#_Toc293409916)

[2.4 Перспективы 23](#_Toc293409917)

[Заключение 24](#_Toc293409918)

[Литература 25](#_Toc293409919)

# Введение

Всегда существуют какие-либо задачи, которые требуют решения, например, работа, предлагаемая заказчиком. К решению задачи можно подойти совершенно различными способами. Самый простой из них – это по постановке задачи сразу искать её решение. Этот способ хоть и самый простой, но имеет самую большую вероятность неудачи, ведь в ходе решения задачи можно столкнуться с большим количеством неучтённых факторов, проблем и подзадач.

Начать решение любой задачи рекомендуется с правильной формулировки задачи. Многие довольно часто недооценивают работу, связанную с формулировкой задачи. Однако многие специалисты полагают, что «хорошо поставить задачу - значит на половину ее решить». Хотя в большинстве случаев заказчику кажется, что он уже сформулировал свою проблему, системный аналитик знает, что предлагаемая клиентом постановка задачи является моделью его реальной проблемной ситуации и неизбежно имеет целевой характер, оставаясь приблизительной и упрощенной. Поэтому необходимо проверить эту модель на адекватность, что приводит к развитию и уточнению первоначальной модели. Очень часто первоначальная формулировка изложена в терминах не тех языков, которые необходимы для построения модели.

Первоначально моделью называли некое вспомогательное средство, объект, который в определенных ситуациях заменял другой объект. Например, манекен в определенном смысле заменяет человека, являясь моделью человеческой фигуры.

Моделью называется объект-заместитель, который в определенных условиях может заменять объект-оригинал, воспроизводя интересующие нас свойства и характеристики оригинала, причем имеет существенные преимущества:

* дешевизну;
* наглядность;
* легкость оперирования и т.п.

В наше время модели часто используются для представления и анализа сложных динамических систем, которые состоят из большого количества элементов, связанных различными типами связей между собой. Чем больше такая система, тем сложнее становится её модель.

Перед аналитиком также предстаёт задачи представления модели, т.е. в каких терминах представить элементы реальной системы в модели. Довольно мощным инструментом для решения этой задачи являются Сети Петри, которые в терминах графов представляют различные состояния системы и протекающие в них процессы, которые в свою очередь изменяют состояние системы.

Сама модель Сети Петри была предложена доктором Петри в его диссертации в 1962 и с тех пор многие аналитики создали различные модификации этих самых сетей. Некоторые из них позволяют упростить модель, а некоторые добавляют в модель больше динамичности, что очень важно при моделировании параллельных динамических систем. Самые распространённые модификации Сетей Петри:

* Временные Сети Петри;
* Иерархические Сети Петри;
* Цветные Сети Петри;

Но всё же для моделирования больших систем выбора одной модели не достаточно, ведь анализ модели сводится к рассмотрению всевозможных вариантов протекания работы системы. Для решения такой задачи напрашивается использование мощности вычислительных систем, что позволяет автоматизировать процесс анализа модели, тем более в наше время мощных ЭВМ это очень актуально.

Существует множество программ для построения моделей в виде Сетей Петри. Одной из самых популярных является пакет CPN Tools, который позволяет построить и проанализировать Высокоуровневые Сети Петри.

Но CPN Tools требует некоторых знаний языков программирования, которые редко являются качествами аналитиков. В связи с этой проблемой я поставил перед собой задачу упростить процесс создания и выполнения Цветных Сетей Петри.

В связи со сказанным выше постановка моей задачи выглядит следующим образом:

**Построение программного пакета для работы с Цветными Сетями Петри.**

Для достижения поставленной цели мне было необходимо выполнить следующие подзадачи:

* Изучение теоретических сведений о Сетях Петри, различных модификациях и Цветных Сетях Петри в частности;
* Решение ряда задач, используя CPN Tools;
* Проектирования архитектуры библиотеки классов для работы с Цветными Сетями Петри;
* Проектирование программного пакета для редактирования и симуляции Цветных Сетей Петри;
* Тестирование пакета на ряде задач;

# Основная часть работы

## Теоретические сведения

Основной предметной областью моей работы являются Сети Петри.

### Основные сведения о сетях Петри

#### Определение простых сетей Петри

Сеть Петри - это инструмент для описания и исследования динамических систем.

Одно из основных достоинств сетей Петри заключается в том, что они могут быть представлены как в графической форме, что обеспечивает их наглядность, так и в аналитической.

##### Графическое представление

При графической интерпретации сеть Петри является графом особого вида, состоящим из вершин двух типов - позиций (place) и переходов (transition), соединенных ориентированными дугами, причем каждая дуга может связывать лишь разнотипные вершины (позицию с переходом или переход с позицией). Вершины-позиции обозначаются кружками, вершины - переходы - прямоугольниками (или черточками).

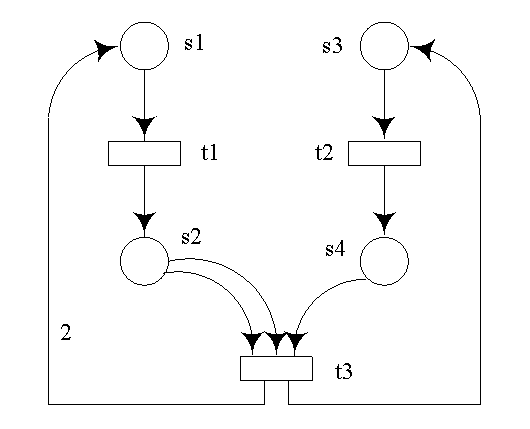


Рис. 1.1 - Пример графа сети Петри

Описывая сеть на основе исследуемой системы, переходы можно сопоставить событиям, происходящим в системе, а позиции - условиям их возникновения. Переход (событие) характеризуется определенным числом входных и выходных позиций, соответствующих предусловию и постусловию данного события. Совокупность переходов, позиций и дуг позволяет описать статическую (неподвижную) систему. Для описания динамики вводится еще один объект - так называемый маркер (token), или метка позиции, которая соответствует выполнению того или иного условия (обозначается точкой внутри позиции). Расположение маркеров в позициях называется разметкой (состоянием) сети. Переход считается активным, если в каждой его входной позиции есть маркеры не меньшего количества, чем количество (вес) дуг выходящих из позиции в переход, что равносильно выполнению всех необходимых условий для наступления события. Наступление события в терминах сетей Петри представляется срабатыванием перехода, при этом маркеры из входных позиций изымаются в количестве равном кратности соответствующих дуг и добавляются в каждую выходную позицию в зависимости от весов выходящих дуг. Текущее состояние исследуемой системы определяется распределением маркеров по позициям сети (маркировкой), а динамика поведения системы отображается перемещением маркеров по позициям сети.

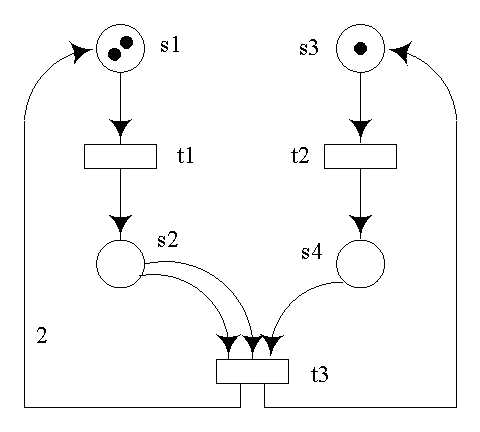


Рис. 1.2 - Пример маркированной сети Петри

##### Аналитическое представление.

Сеть Петри - набор , где – двудольный граф, с множеством вершин и весовых дуг, а – функция, называемая начальной пометкой. Каждой позиции ставится в соответствие некоторое число (разметка позиции).

Формально работа сети Петри описывается множеством последовательностей срабатываний переходов и множеством разметок позиций.

#### Назначение сетей Петри.

Сети Петри используются для анализа и моделирования систем для исправления и модификации систем. С их помощью можно выделить основные недочёты и ошибки системы.

Развитие теории сетей Петри проводилось по двум направлениям:

* Формальная теория сетей Петри. Её задачей является разработка основных средств, методов и понятий, необходимых для применения сетей Петри
* Прикладная теория сетей Петри связана главным образом с применением сетей Петри к моделированию систем и их анализу.

В настоящее время теория сетей Петри содержит большое количество моделей, методов и средств анализа, имеющих обширное количество приложений практически во всех отраслях вычислительной техники и даже вне неё. В соответствии с требованиями прикладных областей были разработаны различные модификации сетей Петри, направленные на учет временных, вероятностных характеристик, использование данных, построение иерархических моделей и т. д.

#### Способы проектирования и анализа систем.

На сегодняшний момент сети Петри приобрели широкую популярность в мире и благодаря своим возможностям заинтересовали многих людей в изучении и применении их на практике. Например, в сфере информационных технологий сети Петри применяются для моделирования аппаратного и программного обеспечения ЭВМ. Аппаратное обеспечение обычно рассматривают на нескольких уровнях:

* на одном уровне ЭВМ состоят из простых устройств памяти и вентилей;
* на более высоком уровне в качестве основных компонент системы выбираются функциональные блоки ЭВМ и регистры;
* на еще более высоком уровне целые вычислительные системы могут быть компонентами сети ЭВМ.

Сильным свойством сетей Петри является их способность моделировать каждый из этих уровней.

Конечно, информационной средой область применения сетей Петри не ограничивается. Фактически, в этом смысле никаких границ не существует. Можно заниматься моделированием GPS (Global Position System) в технике, военных операций Пентагона или даже процессом слюноотделения собаки Павлова, что свидетельствует о широте использования, универсальности этого способа моделирования.

Существует несколько подходов к проектированию и анализу систем:

1. Сети Петри рассматриваются как вспомогательный инструмент анализа. В начале для построения системы используются общепринятые методы проектирования. Затем построенная система моделируется с помощью сети Петри, после чего модель анализируется. Любые трудности, встречающиеся при анализе, указывают на изъяны в проекте. Для их исправления необходимо модифицировать проект. Модифицированный проект затем снова моделируется и анализируется. Этот цикл повторяется до тех пор, пока проводимый анализ не приведет к успеху. Такой подход иллюстрируется на рис. 1.3. Заметим, что его можно использовать и для анализа уже существующих действующих в настоящее время систем.

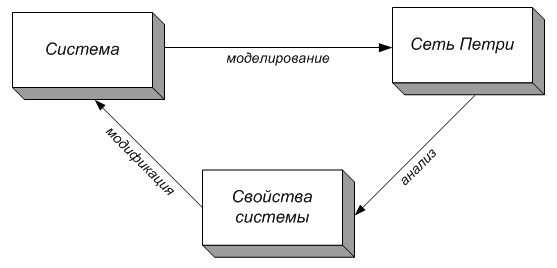


Рис. 1.3. - Использование сетей Петри для моделирования и анализа систем.

1. Другой подход заключается в том, что весь процесс проектирования и определения характеристик проводится в терминах сетей Петри. Методы анализа применяются только для создания проекта сети Петри, свободного от ошибок. Здесь задача заключается в преобразовании представления сети Петри в реальную рабочую систему.

Общей задачей курсового проекта является автоматизация создания сетей Петри, обеспечение с помощью ресурсов компьютера их сохранения, исполнения в реальном времени, редактирования на любом этапе проектирования.

### Анализ существующих решений

Проблема моделирования сетями Петри в данное время очень актуальна. В связи с этим создается большое количество программных продуктов (ПП), затрагивающих различные аспекты использования сетей Петри. Соотношение коммерческих и свободно распространяемых программ на компьютерном рынке в этой области приблизительно равное, их количество довольно велико и оно со временем только растет.

Качественных бесплатных программ по сетям Петри в русской части интернета найти практически не возможно. Серьезно этой проблемой в основном занимаются отдельные группы людей, представляющие российские ВУЗы, причем чаще всего их разработки носят коммерческий характер. А вот в Европе и Америке можно найти немало целых каталогов программ, доступных для скачивания. Прежде чем приступить к написанию своей программы я проанализировал некоторые отечественные и зарубежные ПП (описания программ взяты из интернета):

* **The Petri Net Kernel (PNK) version 2.0** (рис. 1.4) – предоставляет инфраструктуру для создания утилит сетей Петри. Разработка Petri Net Kernel Team, Humboldt-University Berlin, Germany. Написана на Java, 1999-2000 г.
  + *преимущества*: оригинальный редактор, удобная база для серьезных разработок
  + *недостатки*: отсутствие конкретной пользы для обыкновенных пользователей

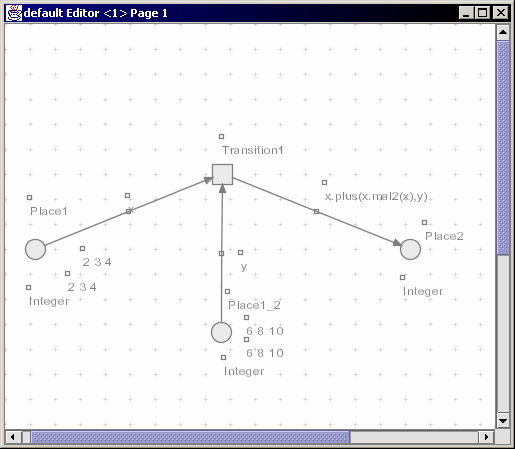


Рис. 1.4. The Petri Net Kernel.

* **HPSim** – разработка by Henryk Anschuetz. Написана на Visual C++, поддерживает классические сети Петри, цветные сети, моделирование с временным параметром.
  + *преимущества*: качественная разработка в своей области; хорошо продуманный, мощный редактор, анимация выполнения сети
* **PIPE 2.5** (рис. 1.5) – версия 2.5 разработана Pere Bonet. Написана на Java, поддерживает классические, временные и приоритетные сети Петри (В 3 версии доступны цветные сети, к сожалению, не было возможности использовать её).
  + *преимущества:* мощный, удобный редактор, анимированное выполнение сети, возможность расчета различных свойств сети
  + *недостатки:* нет достаточного описания для не знающего пользователя об особенностях некоторых не классических компонентов

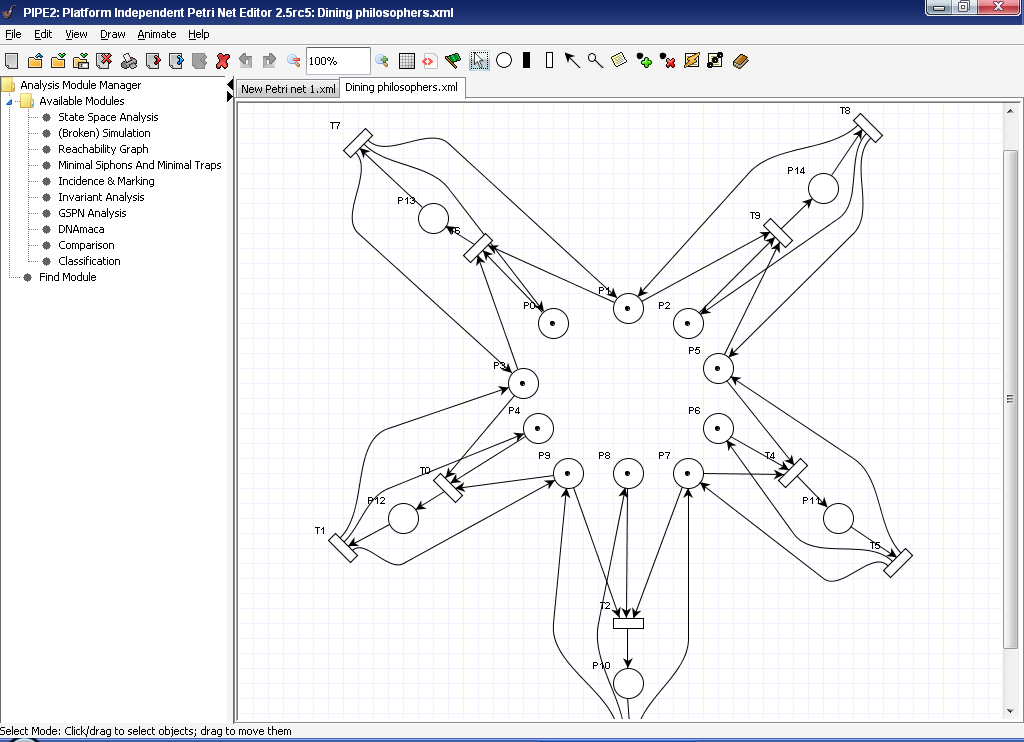


Рис 1.5 Platform Independent Petri Net Editor 2.5

* **CPN Tools** (рис. 1.6) – программа, разработанная Michael Westergaard and H.M.W. (Eric) Verbeek.

Программа является бесплатной и кроссплатформенной, она позволяет создавать/редактировать/симулировать Цветные, Иерархические и Временные Сети Петри в графическом режиме.

Для реализации Цветных Сетей CPN Tools использует язык программирования Standard ML, который вполне подходит для работы с типами, предусмотренными для работы с элементами Цветной Сети.

Редактор CPN Tools имеет интересный интерфейс – инструменты разделены по категориям, каждую из которых можно расположить в отдельном окошке, либо скомпоновать несколько в одном используя вкладки. Также можно поступать с окном редактирования сети, т.е. объединять с другими открытыми сетями.

Контекстное меню (рис 1.7.), которое появляется при зажатии правой кнопки мыши на нужно объекте (элементе сети, окне, инструменте и т.д.), выполнено в круговой форме, что поначалу немного удивило, но потом оказалось довольно удобным.

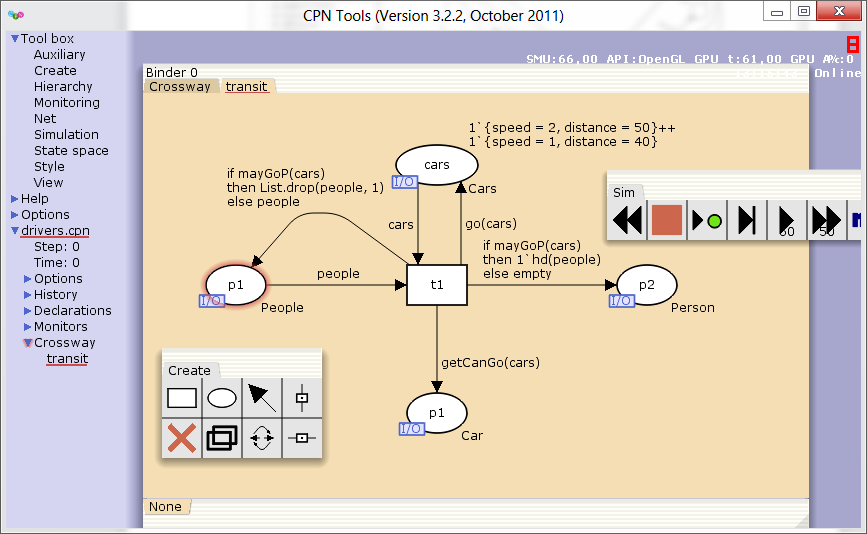


Рис. 1.6 CPN Tools v. 3.2.2

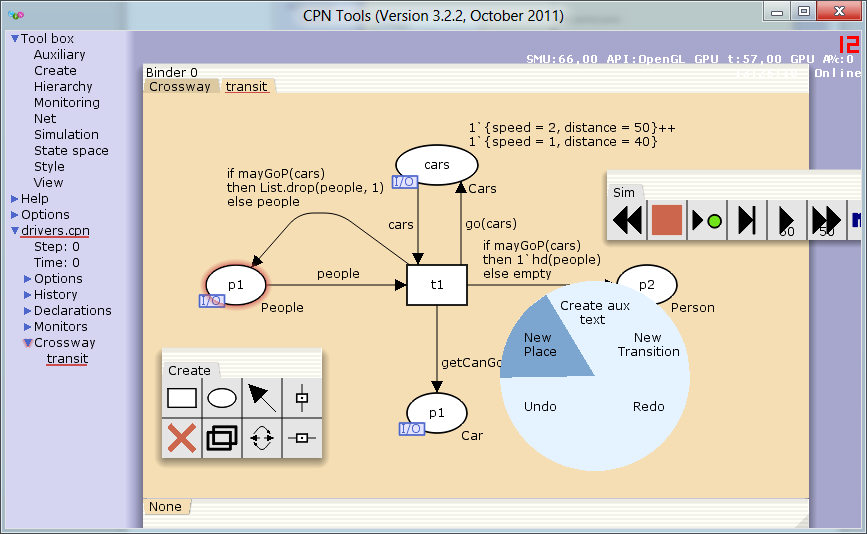


Рис 1.7 Контекстное меню в CPN Tools v. 3.2.2

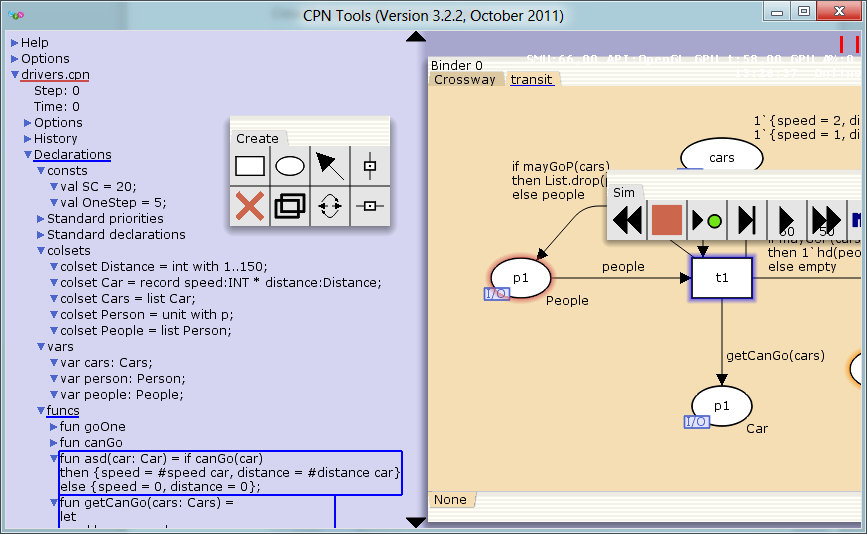


Рис 1.8 Раздел объявления переменных, цветов и функций в CPN Tools v. 3.2.2

Анализ этих продуктов позволил определить требуемый набор функций и особенности визуального интерфейса, которые сделают приложение удобным и переспективным для использования в своей области.

## Проектирование усовершенствованной системы

Программный пакет проектировался как 2 отдельных модуля:

1. Библиотека классов
2. Клиентское приложение

Архитектура программы проектировалась таким образом, чтобы можно было использовать программные классы для работы с Цветными Сетями Петри, не зависимо от клиентского приложения.

Клиентское же приложение проектировалось как графический редактор и конструктор для формального представления графа сети, использующее библиотеку классов.

### Проектирование архитектуры библиотеки

Библиотека классов проектировалась как набор классов, сгруппированный по типу решаемых задач. Например, набор классов для работы с арифметическими выражениями выделяется в отдельную группу Functions, а классы для работы с графами - в группу Graphs. Таким образом использование библиотеки становится удобнее, используя только ту группу классов, которая необходима для текущей задачи.

Некоторые группы классов используют другие группы классов. Это обосновывается тем, что некоторые задачи включают в себя подзадачи, которые уже реализованы в библиотеке.

### Проектирование клиентского приложения

Проектировка клиентского приложения включает проектирование графического интерфейса и набора форм для реализации необходимых функций для работы с графами и Сетями Петри.

Клиентское приложение включает 2 взаимозаменяемых модуля:

* Графический редактор сети
* Конструктор (формальный редактор) сети

#### Графический редактор сети

Эта часть клиентского приложения позволяет редактировать граф в графическом режиме, используя панель инструментов, содержащая необходимый набор инструментов для редактирования графа и сети. Редактирование производится с помощью мыши. Редактор позволяет перемещать, удалять, добавлять, редактировать элементы сети. Каждый элемент сети содержит свой уникальный редактор, т.е. для редактирования каждого элемента пользователю отображается отдельная форма, содержащая поля, зависящие от конкретно выбранного элемента.

Также графический редактор является основным окном, поэтому все меню для работы с файлами сетей содержатся в нём, а также вкладки для каждого открытого файла.

#### Конструктор сети

Конструктор сети предназначен для конструирования Сети Петри, используя элементы управления формы, которые выводят перечень вершин, матрицу инцидентности и кнопки для редактирования отдельных вершин и описания сети. Т.е. конструирование сети сводится к формальному заданию сети.

Для редактирования дуг используется матрица инцидентности, т.е. для добавления или редактирования дуги необходимо вписать нужное значение в ячейку таблицы на пересечении необходимых вершин.

Для редактирования отдельных элементов в конструкторе находится выпадающий список со всеми элементами сети и кнопка для вызова редактора элемента.

## Описание реализации

Программа реализована согласно выводам из проектирования пакета, т.е. соответствует описанным выше идеям.

Рассмотрим реализацию отдельных частей.

### Классы для работы с Цветными Сетями Петри.

Сеть Петри в своём формальном описании является двудольным ориентированным графом. Так как в моей библиотеке уже были графы, я использовал их для реализации работы с Цветными Сетями Петри.

Отдельные элементы графа Петри наследуются от простых элементов графов, что позволило использовать обычный редактор графа, немного расширив его функционал. Так как мною уже был реализован простой графов, я использовал его за основу своего редактора.

Для наглядности я привёл вырожденное дерево классов:

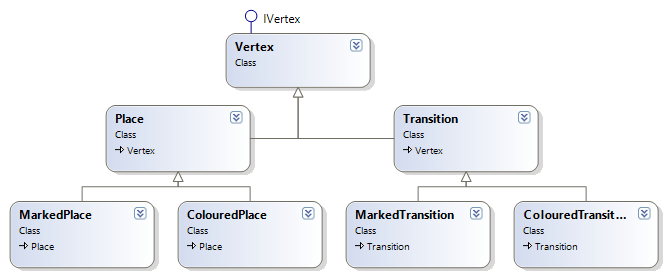


Рис. 3.1. Дерево классов вершин

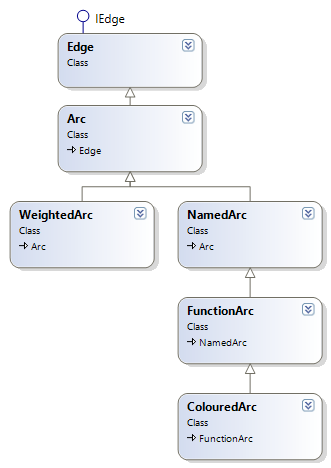


Рис. 3.2. Дерево классов дуг

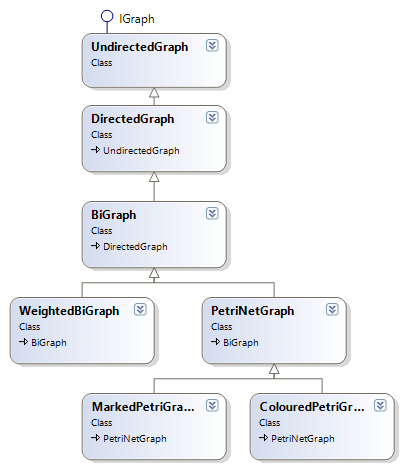


Рис. 3.3. Дерево классов графов

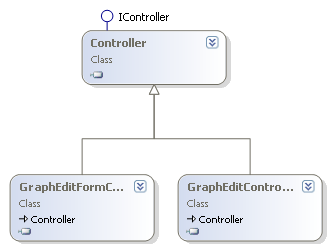
Из приведённых деревьев видна иерархия элементов графа. Например, на рис. 3.1. изображено дерево, листья которого – это классы для представления элементов Цветной Сети Петри, в частности – позиции и переходы. Соответственно на рис. 3.2. изображены дуги, в низу иерархии которых находятся дуги Цветной Сети, которые наследуются от функциональных дуг, т.е. они тоже содержат значение функции. На рис. 3.3. изображена иерархия графов, в низу которой изображены классы Сетей Петри: Классической и Цветной. Они в свою очередь находятся под двудольным графом, что указывает на аналогичное поведение графа Петри и двудольного графа, т.е. дугами могут быть соединены только разнотипные вершины.

### Классы для работы с арифметическими выражениями

## 2.2 Программа GraphEditor.

Приложение, которое представляет собой редактор стандартных видов графов. Оно построено с использованием архитектуры MVC (рис 2.5) и библиотеки графов.

Для работы с графами был разработан класс GraphView, который представляет собой стандартный Вид для использования на любом компоненте WinForms – Control. Также специально для него был разработан контроллер GraphEditController, в котором и осуществляются все манипуляции с графом: добавление, удаление, сохранение и т.д. Использование этого класса состоит в том, чтобы передавать в его конструктор необходимый отображаемый компонент WinForms (например, саму Form) и с помощью внешнего контроллера изменять необходимые свойства Вида, такие как action – действие, которое сейчас будет производиться (добавление вершины, добавление ребра, перемещение и т.п.).



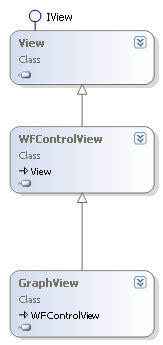


Рис 2.5 – Диаграмма классов видов и контроллеров в GraphEditor

Пример использования класса GraphView – форма GraphEditForm (рис 2.6), на которой есть элемент TabControl (его страницы и используются для редактирования графа), и меню, которое используется для управления действиями пользователя.

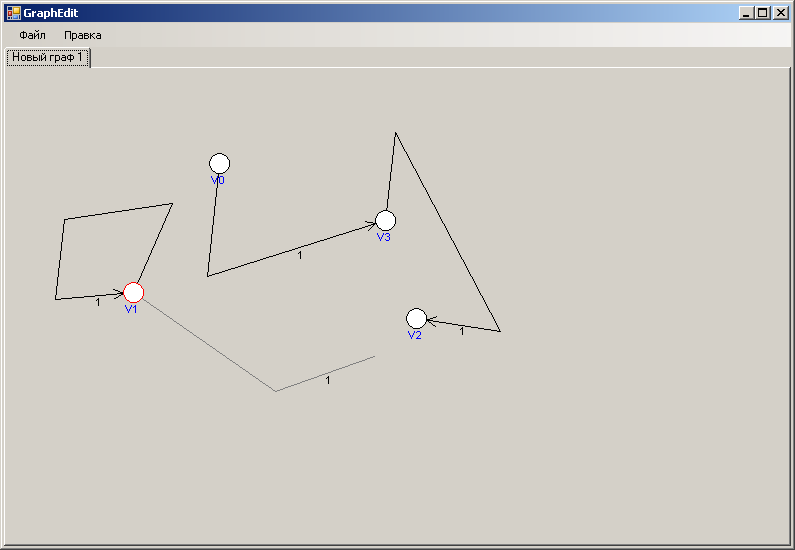


Рис 2.6 Пример программы для редактирования графа

Также в приложении GraphEditor (специально под WinForms) были разработаны «Обёртки» для всех элементов графа и самого графа (рис 2.7). Эти «Обёртки» используются для хранения графических характеристик объектов, таких как: координаты, размеры, метод рисования и т.п. Это позволяет сохранить архитектуру MVC, т.е. не связывать компоненты Модели и Вида (одно из условий MVC – это то, что Вид может знать о Модели, но не наоброт).

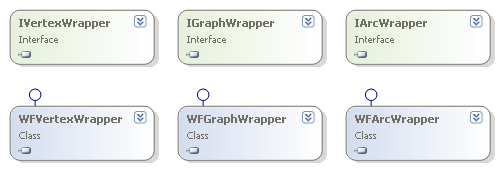


Рис 2.7 – Диаграмма классов – обёрток для графа и его элементов в GraphEditor.

## 2.3 Программа Petri Net Editor.

Приложение Petri Net Editor – это приложение для просмотра и редактирования графов Петри, а также для запуска сетей на их основе. Так для редактирования графа существует отдельная форма, добавив к которой возможность «запуска сети» мы получаем полноценное приложение для моделирования сетей Петри. Такое разделение позволило мне во всех аспектах разобраться в связи самой сети и графом её представления.

Граф Петри – это двудольный граф, что видно на рисунке 2.8. Два подмножества вершин мало того, что не могут быть соединяться дугами сами с собой, так еще и имеют разный тип, и это видно на рисунке 2.9. Реализация решения данной проблемы показывает гибкость реализации графа.

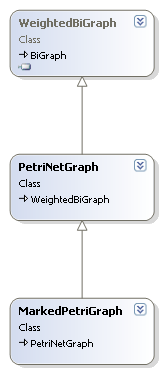


Рис 2.8 – Диаграмма классов графов сетей Петри в Petri Net Editor.

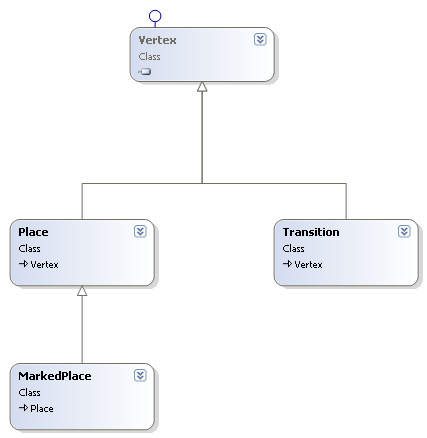


Рис 2.9 – Диаграмма классов, представляющих вершины в графе Петри.

Данное приложение – производное от GraphEditor, ведь их схожесть несомненна (рис 2.6 и 2.7). Основные действия с графами были описаны в ранее указанном приложении, что, посредством добавления некоторых особенностей графа Петри, позволило создать полноценное приложение для моделирования сетей Петри.

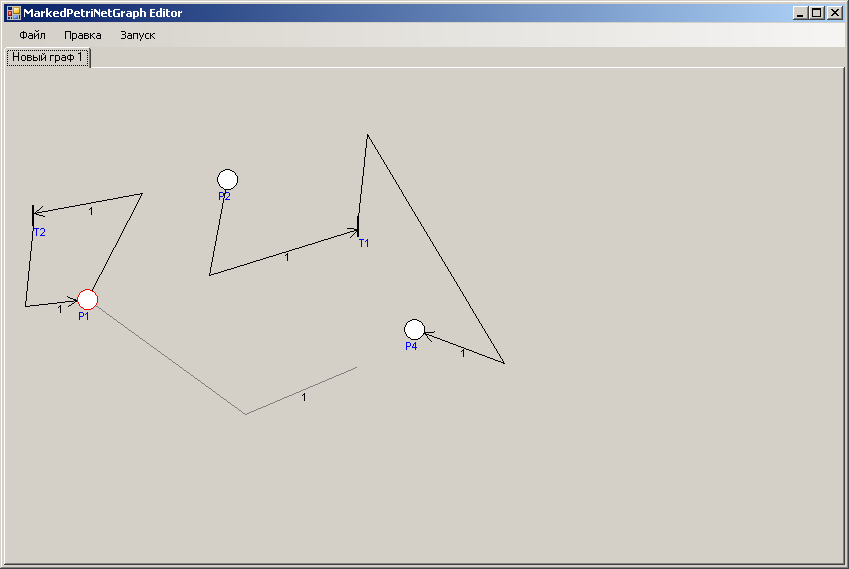


Рис 2.10 – Пример программы для редактирования сети Петри (для сравнения с рисунком 2.6).

То, что в программе было создано две формы (PetriGraphEditorView [рис 2.11] и MarkedPetriGraphEditorView [рис 2.12]), которые в одинаковой степени работают, является показателем того, что приложение довольно гибко и на его основе можно построить сеть Петри любой сложности.

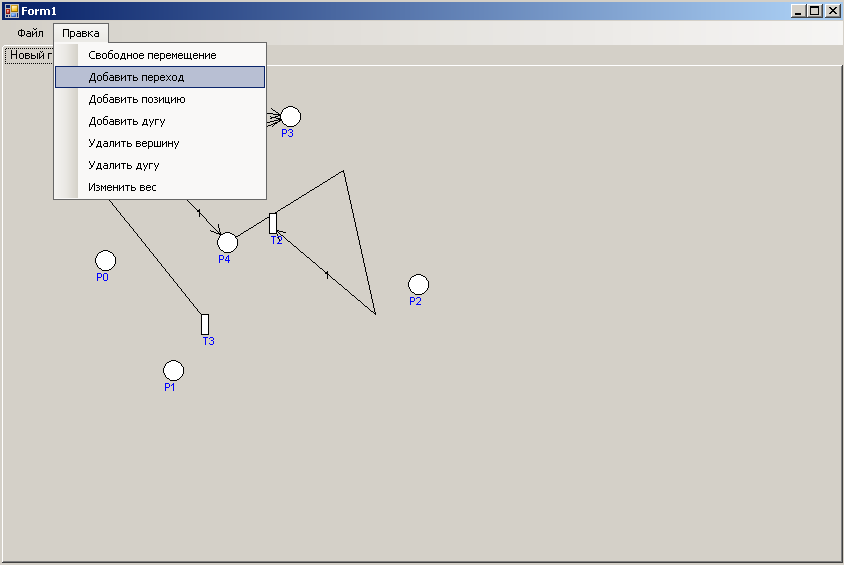


Рис 2.11 – Пример работы программы для редактирования графа Петри.

Для «Запуска сети» пользователю даны 3 инструмента:

* Выбор запускаемого перехода, при этом естественно проверяется, ли вообще возможен запуск выбранного перехода. Для удобства все доступные переходы выделяются красным цветом;
* Запуск перехода, выбранного случайным образом, при этом из всех доступных переходов случайным образом выбирается один;
* Запуск сети, определённый количеством итераций, то есть количество запусков переходов, выбранных случайным образом, после которого выполнение останавливается. При этом можно выбрать задержку в мили секундах между каждым переходом, это позволяет пользователю следить за выполнением.

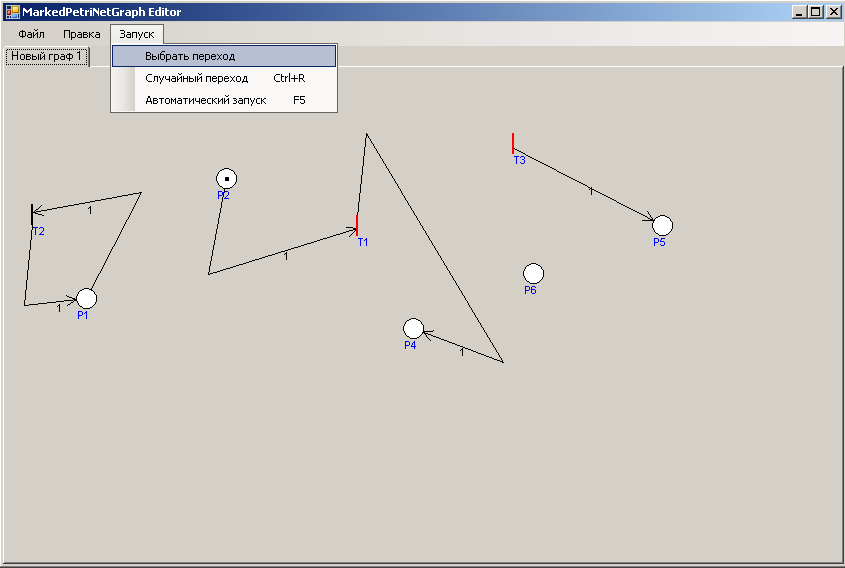


Рис 2.12 – Пример работы программы для редактирования сети Петри с доступными переходами.

## 2.4 Перспективы

Реализация программных средств, созданных при выполнении данной работы, предоставляет возможность в дальнейшем производить выполнение сети Петри параллельно, исходя из теории сетей Петри. Т.к. моей задачей было построение сети Петри на плоскости и тестовый запуск сети, для демонстрации её выполнения пользователю, не было реализовано параллельное выполнение сети.

# Заключение

Мною были изучены основные положения теории сетей Петри, рассмотрены различные модификации классической модели и их особенности и преимущества.

Была спроектирована библиотека классов с учетом основных требований расширяемости и гибкости, после чего была реализована сама библиотека.

Разработанная программа позволяет реализовать построение сети Петри на плоскости и позволяет осуществлять её выполнение. На основе библиотеки классов можно разрабатывать более сложные программы.

Программа представляет собой редактор сети Петри, в котором можно редактировать сети, а также сохранять текущие и загружать уже существующие сети Петри.

# Литература

1. Питерсон Дж. Теория сетей Петри и моделирование систем. - М.: Мир, 1984.
2. Котов В. Е. Сети Петри. - М.: Наука, 1984.