

Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Санкт-Петербургский государственный университет»

Кафедра системного программирования

Тарасова Полина Максимовна

Реализация сетей Петри в системе QReal

Курсовая работа

Научный руководитель:
ст. преп. Литвинов Ю.В.

Санкт-Петербург
2016

Введение

Сети Петри

Свойства сетей Петри

Методы анализа

Редакторы сетей Петри

PIPE v4.3 (PIPE2)

PNEditor

QReal

Реализация

Круговые порты

Маркеры

Заключение

Литература

Введение

Наиболее популярным инструментом для представления и анализа параллельных процессов являются сети Петри. Они позволяют моделировать различные свойства систем: процесс синхронизации, асинхронные события, распределение ресурсов, конкурентные операции, конфликты.

Сети Петри можно рассматривать, как графический и математический инструмент одновременно. Порой клиенту и разработчику сложно понять друг друга из-за неоднозначных текстовых описаний, сложной идеологии работы системы. Используя сети Петри можно получить наглядное графическое представление этих аспектов. Также сети Петри могут быть описаны математической моделью, отображающей поведение системы (например, системой алгебраических уравнений).

Сети Петри являются мощным средством для анализа поведенческих свойств, оценки производительности и моделирования дискретно-событийных систем.

На кафедре системного программирования СПбГУ существует проект QReal с открытыми исходным кодом. Это metaCASE-система — средство, которое позволяет создавать инструменты, необходимые для моделирования системы, и автоматизировать процесс разработки новых визуальных языков.

QReal позволяет создавать визуальные языки программирования и придавать элементам создаваемого языка разнообразные формы, для более удобного и простого описания элементов и взаимодействия с ними. Сети Петри легко представить в графическом виде. Поэтому возникла идея создания редактора, средств анализа и симуляции сетей Петри с помощью средств QReal. Данная курсовая работа является началом реализации описанной выше задачи.

Целью данной работы является разработка редактора сетей Петри с помощью средств быстрого создания предметно-ориентированных языков, предоставляемых платформой QReal. Для достижения цели достаточно было решить следующие задачи:

- изучить специализированную литературу по данной теме;
- создать редактора сетей Петри;
- внесение исправлений в платформу QReal, необходимых для корректной поддержки сетей Петри.

Обзор

Сети Петри

Сети Петри успешно применяются во многих сферах. Используются для моделирования и анализа протоколов связи, различного рода производственных систем, автоматизированных линий сборки, ресурсо-обменных систем и систем реального времени, в разработке программного обеспечения, а также для избежания тупиков (состояний системы, когда один или более процессов ожидают некоторого события, которое никогда не произойдет).

Система, моделируемая с помощью сетей Петри, может быть представлена как наглядной графической, так и математической моделью

Бывает полезным оценить эффективность моделируемой системы. Это позволяет оптимизировать как саму систему, так и подход к ее управлению. Использование сетей Петри в качестве математического инструмента как раз позволяет провести такую оценку.

Сети Петри могут быть рассмотрены как частный случай двудольного ориентированного графа, состоящего из трех типов объектов. Эти объекты: места, переходы и ориентированные дуги, связывающие места с переходами и переходы с местами.

Графически места изображаются кругами, а переходы — линиями или закрашенными прямоугольниками. Место p называется входным для перехода t , если существует ориентированная дуга, связывающая p с t . Место является выходным местом перехода, если существует ориентированная дуга, связывающая переход с местом. Для примера входное (выходное) место может представлять предшествующее (будущее) состояние, а переход — это событие. Входные места могут отражать доступность ресурсов, переходы — их использование (утилизацию), а выходные места — исчерпывание ресурсов (освобождение ресурсов).

С целью изучения динамического поведения моделируемой системы с точки зрения ее состояний и их изменений, каждое место может потенциально содержать пустое значение или положительное число токенов (маркеров), представленных точками. В любой данный момент времени распределение токенов на места называется маркированием сети Петри, определяющим текущее состояние моделируемой системы. Маркирование сети Петри с m местами представляется вектором M размерности $(m \times 1)$, элементы которого обозначаются как $M(p)$ и являются неотрицательными целыми числами, означающими количество токенов в соответствующих местах. Сети Петри, содержащие токены, называются маркированными сетями Петри.

Сеть Петри формально может быть определена следующим образом:

$PN = (P, T, I, O, M_0)$; где

1. $P = \{p_1, p_2, \dots, p_m\}$ — конечное множество мест,

2. $T = \{t_1, t_2, \dots, t_n\}$ — конечное множество переходов,
 $P \cup T \neq \emptyset$ и $P \cap T = \emptyset$,
3. $I : (P \times T) \rightarrow N$ — функция ввода, которая определяет ориентированные дуги из мест к переходам, где N — набор из неотрицательных целых чисел
4. $O : (P \times T) \rightarrow N$ — функция вывода, которая определяет ориентированные дуги из переходов к местам
5. $M_0 : P \rightarrow N$ — начальная маркировка

Если $I(p, t) = k$ ($O(p, t) = k$), тогда там существует k ориентированных параллельных дуг, соединяющих место p с переходом t (переход t с местом p). Если $I(p, t) = 0$ ($O(p, t) = 0$), то не существует ни одной ориентированной дуги, соединяющей p и t (t и p). Часто, в графическом представлении, параллельные дуги, соединяющие места с переходами (переходы с местами), представляются одной ориентированной дугой, маркированной количеством этих параллельных дуг, т.е. весом k .

Наблюдая за изменениями распределения токенов в сети, отражающими события или выполнения каких-либо операций, можно изучать динамические характеристики моделируемой системы. Существуют правила, по которым происходит передвижение токенов.

Правило включения

Переход t называется включенным, если каждое входное место p этого перехода содержит по крайней мере количество токенов, равное весу ориентированной дуги, соединяющей p и t .

Правило срабатывания

Срабатывание включенного перехода t удаляет из каждого входного места p количество токенов, равное весу ориентированной дуги, связывающей p и t . Это также помещает в каждом выходном месте p количество токенов, совпадающее с весом ориентированной дуги, соединяющей t и p .

Сеть Петри называется чистой или свободной от циклов, если нет места, которое являлось бы местом входа и местом выхода одновременно для одного и того же перехода. Сеть Петри, которая содержит петли, всегда может быть перекодирована в чистую сеть Петри.

Свойства сетей Петри

Сети Петри как математический инструмент обладают рядом свойств. Эти свойства, когда речь идет об интерпретации в контексте моделируемой системы, позволяют проектировщику системы распознать, есть ли определенные заявленные функциональные свойства у проектируемой системы. Можно выделить два типа свойств: поведенческие и структурные. Поведенческие свойства — это те, которые зависят от начального состояния или начальной маркировки сети Петри. Структурные свойства зависят только от топологии или структуры сети. Некоторые поведенческие свойства:

Достижимость

Маркировка M_i называется достижимой из маркировки M_0 , если существует последовательность переходов, срабатывание которых преобразует маркировку M_0 к M_i . Маркировка M_i называется немедленно достижимой из маркировки M_0 , если срабатывание включенного перехода в $M_0 \rightarrow M_i$. Множество всех доступных маркировок, достижимых из M_0 называются достижимым множеством и обозначаются как $R(M_0)$. Свойство достижимости позволяет понять, может ли система вообще перейти в нужное состояние.

Ограниченность и безопасность

Сеть Петри называется k -ограниченной, если количество токенов в любом месте p ($p \in P$), не превосходит k (k — неотрицательное целое число) для каждой маркировки M , доступной из начальной M_0 ($M \in R(M_0)$). Сеть Петри безопасна, если она 1-ограничена (т.е. в каждом месте не более одного токена). Это свойство помогает обнаружить потенциальные переполнения буферов в системе.

Консерватизм

Сеть Петри консервативна, если общее количество токенов в каждой достижимой маркировке сохраняется. С точки зрения структуры сети, это может произойти, только если количество входных дуг каждого перехода совпадает с количеством выходящих дуг. Однако, в реальной системе ресурсы часто объединяются/разделяются, поэтому полезнее анализировать взвешенную сумму токенов.

Живость

Сеть называется живой, если все маркировки M , которые доступны из начальной маркировки M_0 , в конечном счете могут запустить любой переход в сети с помощью некоторой последовательности срабатываний. Но это требование слишком строгое для представления реальной системы, поэтому были введены несколько разных уровней живости. Переход t в сети Петри называется:

- L_0 -живым (или мертвым) если нет срабатывающей последовательности в $L(M_0)$ в которой t может сработать
- L_1 -живым (потенциально срабатывающий) если t может сработать по крайней мере однажды в какой-то последовательности срабатываний в $L(M_0)$
- L_2 -живым если t может сработать по крайней мере k раз в некоторой срабатывающей последовательности $L(M_0)$, где k — любое возможное целое число
- L_3 -живым если t может срабатывать бесконечно часто в некоторой последовательности срабатываний в $L(M_0)$
- L_4 -живым (или живым) если t является L_1 -живым (потенциально срабатывающим) в каждой маркировке в $R(M_0)$

Согласно этой классификации, сеть Петри называется L_1 -живой, для маркировки M_0 , если каждый переход в сети L_1 -жив. Концепция живости позволяет анализировать систему на потенциальную возможность возникновения тупиковых ситуаций.

Обратимость и главное состояние

Важной проблемой в работе с реальными системами является способность таких систем к исправлению ошибок. Такие системы обязательно возвращаются из некорректного состояния к предшествующему корректному. Это требование тесно связано с обратимостью и главным состоянием сети Петри. Сеть Петри для начальной маркировки M_0 называется обратимой, если для каждой маркировки M в $R(M_0)$, M_0 доступна из M . Свойство главного состояния менее ограничительно и более практично, чем обратимость: состояние M_1 сети Петри называется главным состоянием, если для каждой маркировки M в $R(M_0)$, M_1 доступно из M . Это даёт возможность моделировать состояния инициализации, которые могут быть необратимы, но дальнейшая работа системы обладает свойством обратимости.

Методы анализа

Соответствие один-к-одному между моделью сети Петри и реальной системой позволяет более точно переносить аналитические результаты, полученные для модели, на оригинальную систему. Есть несколько методов анализа системы, тут будет рассмотрено два: алгебраический и пространства состояний.

Пространства состояний

Этот подход основан на перечислении всех возможных маркировок, доступных из изначальной маркировки M_0 . Начиная с начальной маркировки M_0 , можно построить дерево путём исполнения всех возможных переходов, включенных во всех возможных маркировках, доступных из M_0 . В дереве достижимости каждый узел отмечен маркировкой; дуги отмечены переходами. Корень дерева отмечен начальной маркировкой M_0 .

При построении дерева возможны ситуации, когда одна и та же маркировка встречается в дереве дважды на одном пути к корню (например, когда сеть обратима). В таком случае дерево получается бесконечным. И, чтобы такую сеть можно было анализировать, вводится символ ω , который обозначает бесконечное количество токенов в месте. С символом ω дерево получается всегда конечным, однако его использование снижает точность анализа.

Алгебраический

Альтернативный подход к представлению и анализу сетей Петри основан на матричных уравнениях. В этом подходе уравнения используются, чтобы представлять динамическое поведение сетей Петри. Основой этого подхода является матрица инцидентности, которая определяет все возможные соединения между местами и переходами в сети Петри. Матрица инцидентности чистой сети Петри — это целочисленная матрица A размера $n \times m$, где n — количество переходов, а m —

количество мест. Уравнение состояния для сети Петри представляет изменение в распределении токенов на местах (изменение маркировки) в результате срабатывания перехода.

Инварианты

Два понятия, связанные с матрицей инцидентности, особенно полезны в изучении свойств модели сети Петри. Это T-инвариант и P-инвариант.

Целочисленное решение x уравнения $(A^T)x = 0$ называется T-инвариантом. Вхождения, отличные от нуля, в T-инварианте представляют переходы, содержащие срабатывающую последовательность, преобразовывающую маркировку M_0 в M_0 , и количество раз, которое эти переходы появляются в этой последовательности, но это не определяет порядок срабатывания переходов.

Целочисленное решение y уравнения $Ay = 0$ называют P-инвариантом. Вхождения, отличные от нуля в P-инварианте, представляют веса, связанные с соответствующими местами так, что взвешенная сумма токенов на этих местах постоянна для всех маркировок, достижимых из начальной маркировки.

Редакторы сетей Петри

С помощью графического редактора очень удобно симулировать работу сетей Петри и анализировать их свойства. Существует множество графических редакторов сетей Петри. Ниже представлены два наиболее популярных.

PIPE v4.3 (PIPE2)

- + возможна интеграция с MATLAB
- + импорт и экспорт в XML
- + функциональный интерфейс
- + открытый исходный код
- + кросс-платформенность
(код написан на JAVA)
- + поддержка сетей Петри с ингибидорными дугами, а также цветных, временных, стохастических
- + поддержка инвариантов
- + проверка на достижимость

- перегруженность интерфейса и управления
- возможность редактирования только при помощи мыши
(пользователь вынужден производить десятки нажатий мыши)
- невозможность копирования уже созданной структуры
- допускает слишком свободное проектирование модели
(нельзя выбрать конкретный набор правил и начать моделирование только с их

помощью: система автоматически предлагает различную функциональность, которая не необходима для проектирования, а лишь запутывает пользователя)

- автоматические текстовые метки с именами элементов перекрывают токены (осложняет проектирование, так как важно количество ресурсов в системе, а не название ее элементов)
- ограниченность полотна (сверху и слева)
(это затрудняет навигацию по рабочему пространству и накладывает ограничение на рост сети в данных направлениях)

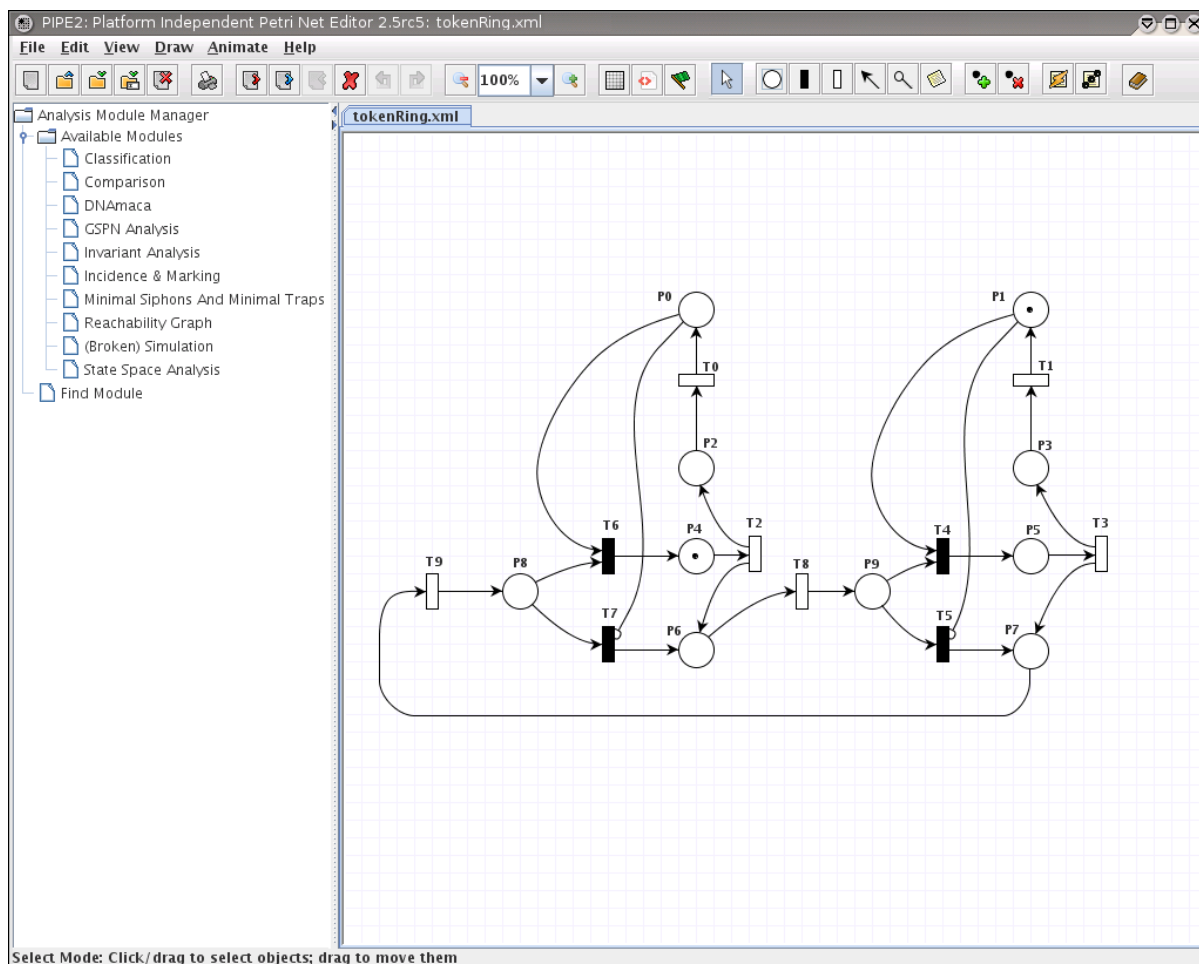


Рис.1 PIPE2

PNEditor

- + открытый исходный код
- + кросс-платформенность
- поддержка подсетей

(более того, выбранную подсеть можно сохранить в отдельный файл для последующего переиспользования, например, для замены другой подсети искомой)

- + поддержка ролей
(роль — подсеть из набора переходов, определяющая какие переходы могут быть осуществлены пользователем, использующим роль)
- + определение стохастических мест
- + поддержка иерархических сетей Петри
(т.е. поддержка переходов, в которые могут быть вложены сети Петри, срабатывание такого перехода — выполнение полного жизненного цикла вложенной сети)
- + проверка на ограниченность

- нет поддержки инвариантов
- импорт и экспорт в PNML
(The Petri Net Markup Language — специальный формат, основанный на языке разметки XML, используемый для представления сетей Петри)
можно рассматривать и как преимущество, в зависимости от требований
- нет проверки на достижимость
- JAVA версии 6 или позднее

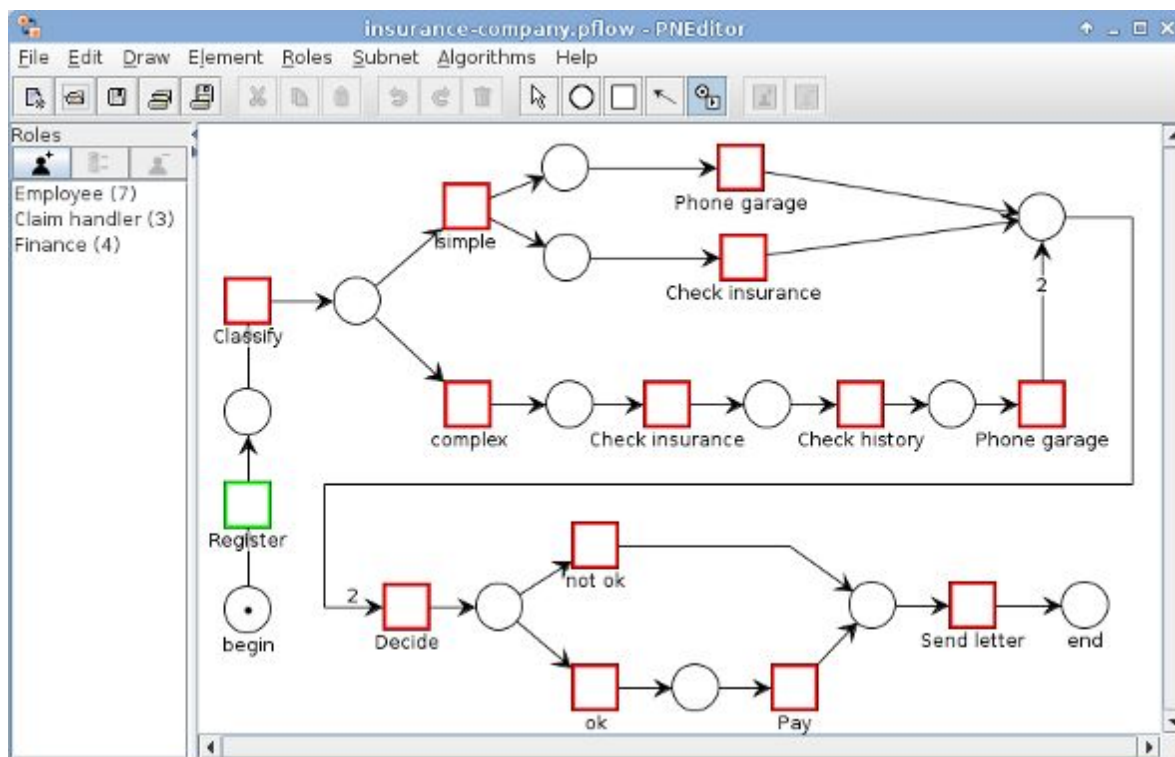


Рис.2 PNEditor

QReal

QReal — кросс-платформенная среда с открытым исходным кодом и с расширяемой архитектурой. Последнее позволяет добавлять новые инструменты для работы с графическими редакторами. В свою очередь, редакторы, полученные с помощью QReal, могут служить основой для создания других предметно-ориентированных визуальных языков.

Благодаря широкому выбору инструментов, создать примитивный редактор сетей Петри в системе QReal было возможно и до написания данной курсовой работы. Однако, в то время как приведенные выше аналоги имеют удобный интерфейс и наглядно представляют моделируемую систему, то с помощью редактора на базе QReal было бы невозможно создавать хоть какие-то сети Петри, помимо совсем тривиальных, и взаимодействовать с ними. Проще говоря, симулировать поведение полученной сети было бы невозможно, а саму сеть бесполезно анализировать.

Но в отличие от перечисленных решений, переиспользование которых невозможно или малоэффективно, QReal представляет больше возможностей для работы с редакторами, ведь возможна их интеграция с другими визуальными языками. Это делает среду очень перспективной для разработки.

В робототехнике сети Петри широко применяются при работе с разделяемыми ресурсами и для планирования и распределения задач между роботами в группе. На базе QReal, существует среда программирования роботов для школьников, TRIK-studio. Поэтому возможными перспективами развития данной работы может быть программирование роботов с помощью сетей Петри.

Реализация

Язык описания предметно-ориентированного языка называется метаязыком, а модели, которые описывают создаваемый визуальный язык, построенные с помощью метаязыка — метамоделями.

В QReal есть два генеративных способа получения редактора по метамодели: с промежуточным xml-представлением (QReal Xml Compiler (qrxс)) и без него (QReal Metamodel Compiler (qrmc)); и один интерпретативный (создание редактора “на лету”). Более подробно о скорости и результативности каждого из подходов рассказано в [1].

С помощью этих двух подходов были получены редакторы сетей Петри. Однако, в силу того, что интерпретатор менее используем в настоящий момент, было принято решение продолжать работу с редактором, полученным с помощью QRXC.

В рамках данной курсовой работы был выполнен первый этап в создании полноценного редактора сетей Петри — поддержано корректное графическое представление элементов сети.

Круговые порты

В QReal для соединения сущностей дугами необходимо наличие у сущностей специальной области, к которой присоединяются дуги — порта. На момент выполнения работы существовали лишь точечные порты (место входа/выхода дуги) и линейные (отрезок, к любой точке которого можно присоединить дугу). Но, поскольку место имеет форму круга, для того, чтобы корректно отображалась связь между местами и переходами, необходима была реализация нового вида портов — круговых, чтобы любую точку круга можно было связать дугой с другим элементом.

Круговые порты были поддержаны в метаредакторе QReal (визуальном редакторе синтаксиса визуального языка) и в промежуточном XML-представлении. Последнее QReal использует для генерации редактора, поэтому туда тоже понадобилось внести некоторые изменения: описание кругового порта — координаты центра, радиус. Использовать данную функциональность можно и при работе с другими визуальными языками, достаточно добавить перечисленные характеристики порта в XML-файл. Последнее QReal использует для генерации редактора, поэтому туда тоже понадобилось внести некоторые изменения: описание кругового порта — координаты центра, радиус. Также, в силу того, что круговые порты прежде не имели никакой реализации в системе, необходимо было внедрить новую функциональность и в ядро QReal. Понадобилось рассчитывать расстояние от точки вне порта до его ближайшей точки, чтобы определить место соединения дуги и порта. Это было реализовано следующим образом: с помощью линии, связывающей точку вне круга (начало или конец дуги) с его центром, высчитывался угол наклона, и с его помощью определялись координаты точки конца или начала дуги, лежащей на окружности.

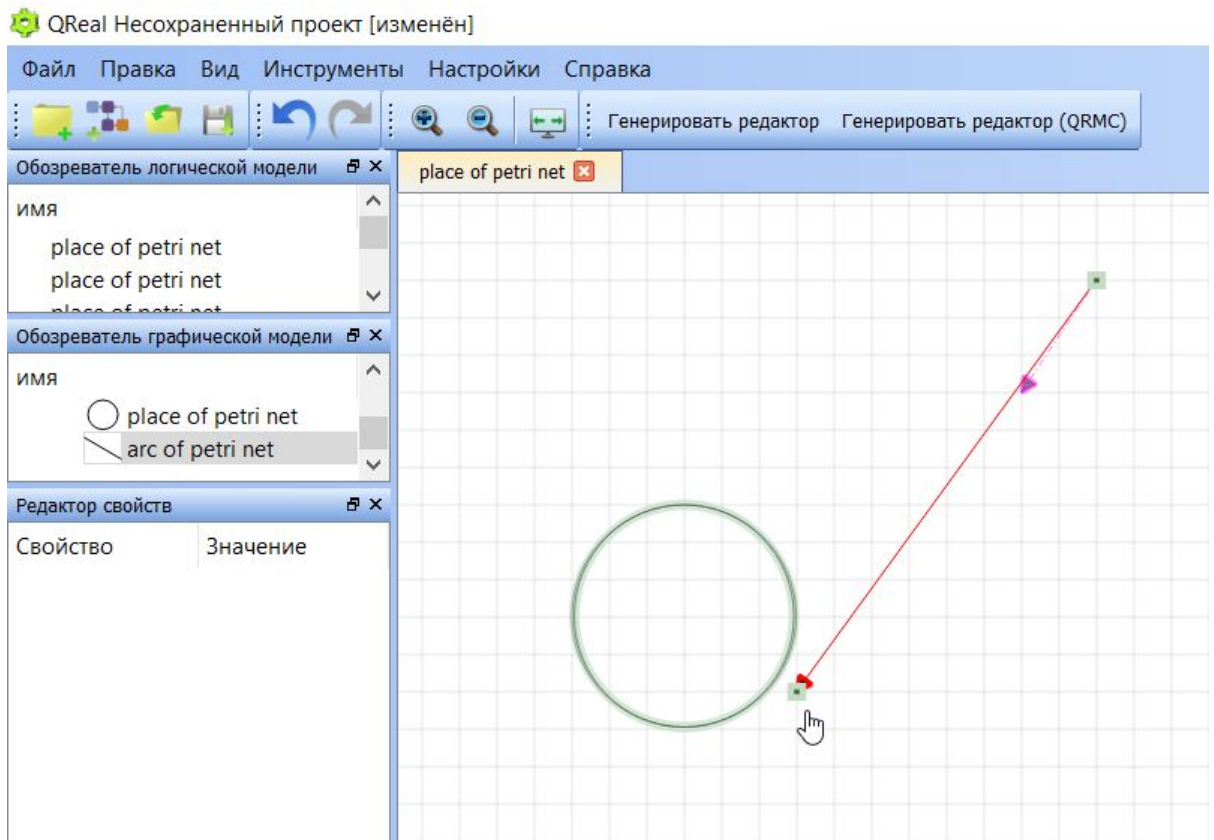


Рис.3 Круговой порт

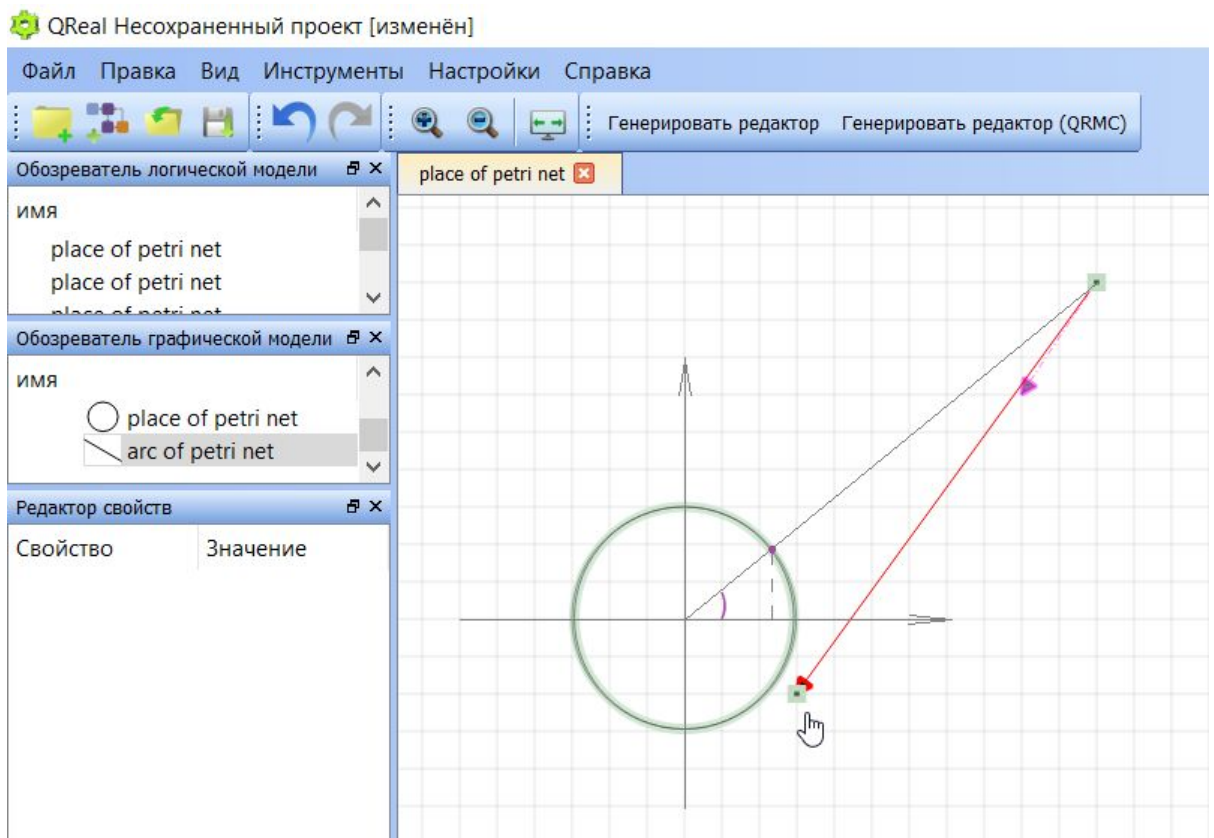


Рис. 3.а) Определение ближайшей точки на окружности к данной

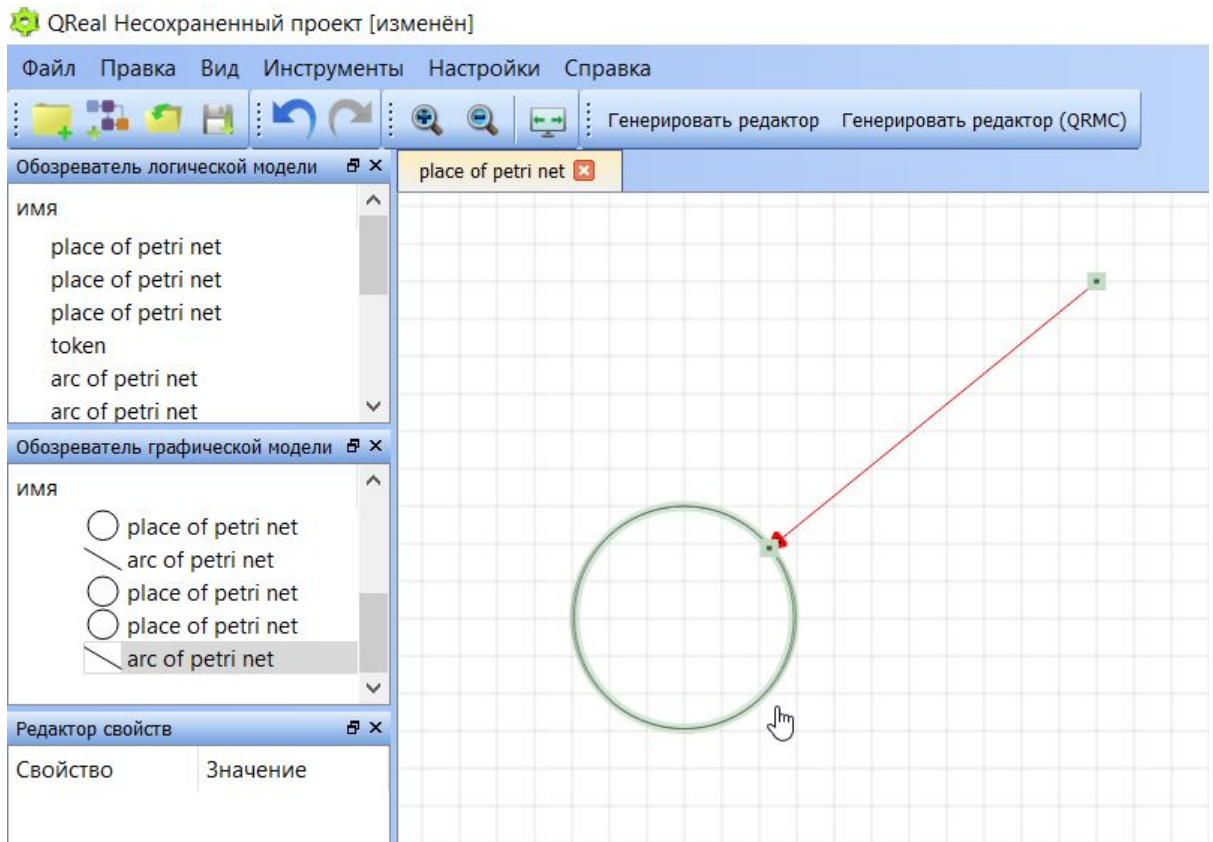


Рис. 3.б) Подцепление дуги к круговому порту

Маркеры

Очень важно знать точное количество ресурсов в сети. Для этого необходимо, чтобы количество токенов в местах можно было однозначно определить. Поэтому, требуется предусмотреть ситуацию, когда токенов в месте слишком много и они не могут отрисоваться без наложения друг на друга.

Идея корректного отображения маркеров заключается в том, чтобы при количестве маркеров меньше шести их изображения рисовались в месте, равномерно распределяясь в нём. При большем количестве маркеров отрисовывается только один, и рядом появляется число — их количество. Для реализации этого поведения используется такая возможность ядра QReal, как условные видимости объектов: каждому элементу формы фигуры можно сопоставить логическое выражение с использованием свойств сущности языка, и элемент формы фигуры будет показываться в редакторе только в том случае, если это выражение истинно.

Условная видимость объектов работала для изображений, но не была поддержана для текстовых меток.

Для этого пришлось внести следующие изменения в ядро системы: поддержать проверку условия видимости в классе, отвечающем за рисование текстовых меток.

Пример условия видимости для эллипса в .xml:

```
<ellipse fill-style="none" ...>  
    <showIf sign==" value="4" property="token"/>  
</ellipse>
```

Если значение свойства “token” равно четырем, то элемент будет отрисовываться.

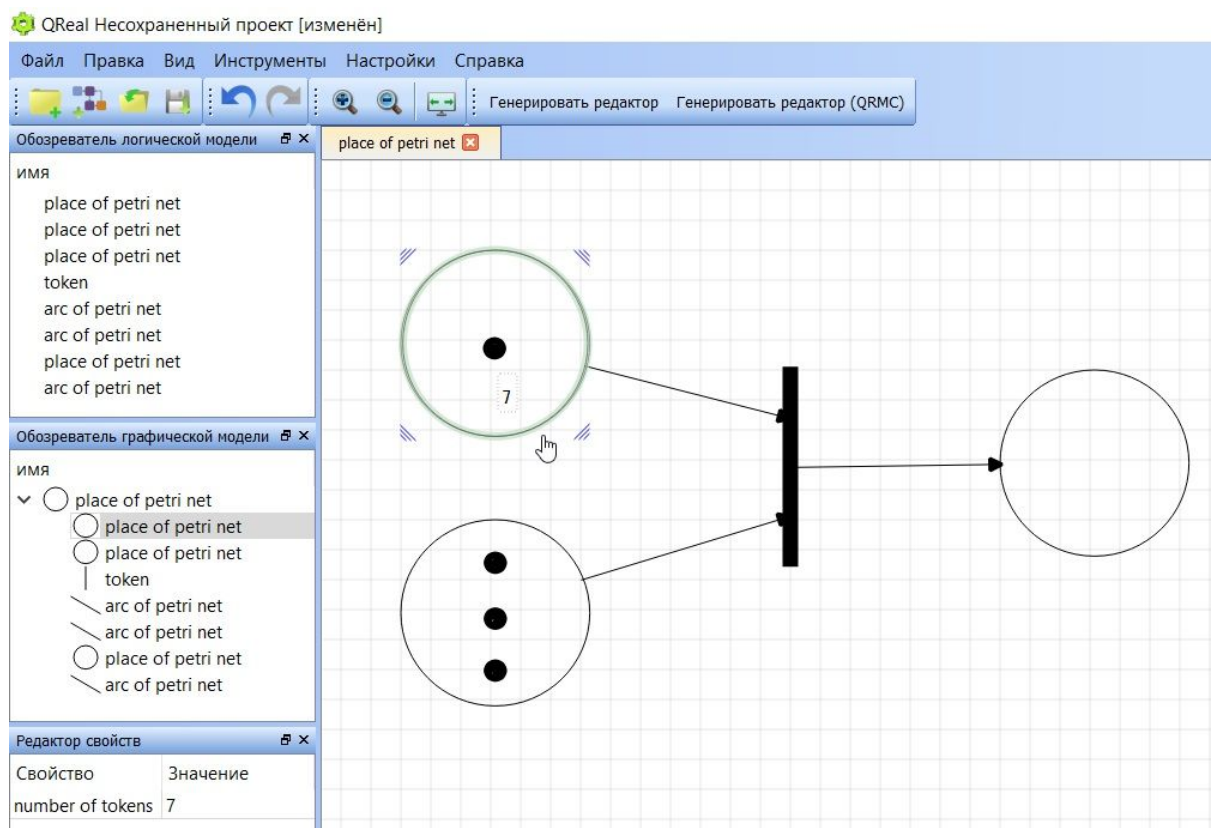


Рис.4 Маркеры

Заключение

В результате данной курсовой работы были решены следующие задачи:

- была изучена литература по сетям Петри, написан обзор свойств и методов анализа сетей Петри;
- выполнен обзор популярных редакторов сетей Петри;
- был создан прототип редактора сетей Петри генеративным способом;
- поддержан новый вид портов в системе QReal;
- осуществлена поддержка условий видимости для текстовых меток.

Литература

[1] Тарасова П.М., Храмышкина Ю.С., Литвинов Ю.В. Сравнение способов получения редакторов по метамодели и скорости их работы // Материалы научнопрактической конференции студентов, аспирантов и молодых учёных "Современные технологии в теории и практике программирования". СПб.: Издво Политехн. унта, 2015. С. 7980

[2] Терехов А.Н., Брыксин Т.А., Литвинов Ю. В., Смирнов К.К., Никандров Г.А., Иванов В.Ю., Такун Е.И. Архитектура среды визуального моделирования QReal. Системное программирование. Т. 4. СПб.: Издво СПбГУ. 2000. С. 165–169.

[3] Petri Nets and Industrial Applications: A Tutorial Richard Zurawski and MengChu Zhou IEEE. TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 41, NO. 6, DECEMBER 1994 561

[4] A Robotic Soccer Passing Task Using Petri Net Plans (Demo Paper) / F Palamara, V a Ziparo, L Iocchi [et al.] // Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers. 2008. pp. 1711–1712.

[5] Murata Tadao. Petri nets: Properties, analysis and applications // Proceedings of the IEEE. 1989. Т. 77, No 4. pp. 541–580.

[6] N.J. Dingle, W.J. Knottenbelt and T. Suto. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets // ACM SIGMETRICS Performance Evaluation Review (Special Issue on Tools for Computer Performance Modelling and Reliability Analysis), Vol. 36(4), March 2009, pp. 34-39.