

УДК 519.6

Редуцирование пространства состояний сети Петри для объектов из одного класса*

А.А. ВОЕВОДА, Д.О. РОМАННИКОВ

Рассматривается преобразование UML-диаграммы состояния, описывающей поведение класса, в сеть Петри. Проанализированы существующие правила преобразования диаграмм UML в сети Петри. Предложен ряд изменений в правила преобразования, позволяющие производить формальное однозначное преобразование диаграмм UML в сети Петри так, что при помощи результирующей сети Петри появляется возможность моделировать поведение нескольких объектов одного класса одновременно.

Ключевые слова: UML-диаграммы, сети Петри, CPN, программное обеспечение.

ВВЕДЕНИЕ

Разработка программных систем требует значительных затрат ресурсов на тестирование конечного программного продукта. Однако из-за высокой степени сложности программного обеспечения (ПО) и большого числа различных вариантов использования не редко ПО содержит ошибки. Часть таких ошибок невозможно найти в условиях среды тестирования. В таких случаях для гарантии создания качественного ПО применяют различные формальные методики разработки, например [1-7]. Одним из ключевых этапов подобных методик является преобразование полученного набора UML-диаграмм в сети Петри. Данное преобразование должно быть осуществлено формально и по возможности однозначно на основании набора правил преобразования. В данной статье приведен анализ существующего набора правил преобразования. Выделены его недостатки, главным из которых является отсутствие возможности моделирования поведения нескольких объектов одного класса. Предложен скорректированный набор правил преобразования, позволяющий производить формальное однозначное преобразование произвольно построенных UML-диаграмм состояния. Преобразование проиллюстрировано на примере диаграмм состояния, но его также можно применять к другим UML-диаграммам.

1. АНАЛИЗ НАБОРА ПРАВИЛ ПРЕОБРАЗОВАНИЯ

Набор основных правил преобразования UML-диаграмм в сети Петри приведен в ряде работ [3, 5], а полный набор правил преобразования предложен в [1].

Преобразование диаграммы состояния основано на следующих правилах:

1) состояние отображается в место; 2) переход диаграммы состояний преобразовывается в переход сети Петри; 3) события отображаются в метки, которые будут храниться во входном месте и использоваться как условия срабатывания переходов, а действия в метки, которые будут генерироваться и сохраняться в месте, распределяющем события; 4) тип события состоит из трех компонент: событие, параметр и признак события – внутреннее или внешнее; 5) для каждой сети Петри, полученной при преобразовании UML-диаграмм состояния, создается место «ED» – место-накопитель меток-событий, из которого при помощи условий переходов

* Получена 5 сентября 2011 г.

«mIn» и «mOut» метки направляются в место входных событий класса «IP» или в место-накопитель внешних событий – «InternalLinking» являющегося местом слияния для всех сетей Петри классов; б) дополнительный фильтр событий может быть установлен добавлением цвета (параметра) в цвет (тип) событий.

Кроме вышеуказанных правил могут быть использованы дополнительные правила в зависимости от специфики решаемых задач, например, таких как построение ПО для центров дистанционного контроля и управления [1]. Данные правила не приведены в этой статье в силу своей специфики.

Одним из недостатков данного набора правил преобразования является то, что полученная в результате сеть Петри не позволяет моделировать поведение нескольких объектов одного класса одновременно. Другим значительным недостатком является то, что нет возможности получить значение исполняемого метода, а также моделировать вызов статических методов.

2. МОДИФИЦИРОВАННЫЙ НАБОР ПРАВИЛ ПРЕОБРАЗОВАНИЯ

Для решения вышеприведенных проблем преобразования UML-диаграмм в сети Петри необходимо внести изменения в набор правил. Правила 1) и 2) являются базовыми правилами преобразования и должны быть оставлены неизменными. Правила 3) – 6) необходимы для формирования дополнительных мест для моделирования вызова событий, моделирующих вызов методов. Концепция, которую реализуют правила 3) – 6) является достаточно удачной для моделирования событий. К сожалению, ее нельзя применять к моделированию вызова методов, так как параметры, которые передаются в метод, нельзя передать через места *IP*, *ED*, *InternalLinking*, потому что данные места имеют определенный тип данных, через который нельзя передать произвольные параметры метода¹. Следовательно, необходимо сформировать новые принципы построения сети Петри из UML-диаграммы.

Предлагается к правилам 1) и 2) добавить следующие правила: 3) для вызова методов используются дополнительные места, соединенные с переходом, обозначающим срабатывание метода; 4) возврат программы на место вызова метода и возвращение значения выполнения метода осуществляется при помощи дополнительного места, соединенного с переходом, в котором заканчивается выполнение метода; 5) возврат программы на место вызова метода и возвращение значения выполнения метода осуществляется при помощи дополнительного места, соединенного с переходом, в котором заканчивается выполнение метода; 6) к переходам сети необходимо добавить защитные условия, которые будут определять, какая метка (метка представляет собой объект) может совершить переход (т. е. для какого объекта будет вызван метод). Защитное условие будет работать на основе уникального идентификатора объекта, к которому может относиться как имя класса, так и обычный целочисленный номер. Идентификатор объекта должен храниться в метке, которая моделирует поведение объекта. Для этого необходимо расширить тип метки и ввести дополнительное значение для хранения идентификатора.

3. ПРИМЕР ПРЕОБРАЗОВАНИЯ UML-ДИАГРАММЫ СОСТОЯНИЯ В СЕТЬ ПЕТРИ

Применение правил преобразования проиллюстрируем на примере разработки ПО для автоматизированной системы поддержания постоянного давления в водонапорной станции [2, 6, 7]. Рассмотрим диаграмму состояний на рис. 1, которая представляет собой фрагмент системы. Данная диаграмма состояний отображает поведение заслонки. Изначально заслонка находится в положении *closed*. При срабатывании метода *on* состояние меняется на *opening* до того

¹ Если бы сети Петри имели возможность передавать произвольный тип параметра в метке, то правила 3) – 6) можно было бы использовать (при некоторой модификации) в преобразовании UML-диаграмм в сети Петри.

момента, как поступит сигнал подтверждения открытия задвижки *ValveOpenAck*. При его поступлении заслонка переходит в состояние *open*. Если сигнал не поступил в течение 50 секунд, то объект переходит в состояние *error*. Аналогичное поведение происходит при срабатывании метода *off*. Из состояния *error* в состояние *closed* задвижка переходит при вызове метода квитирования ошибки *ack*.

При применении немодифицированных правил преобразования, данная диаграмма состояния преобразовывается в сеть Петри, приведенную на рис. 2. Из функционирования данной сети видно, что если будет создано несколько объектов – несколько раз передано событие *new*, а затем вызван метод *on*, то возникнет неопределенная ситуация, в которой не понятно для какого объекта был вызван метод. Так же, исходя из структуры сети, вызов метода объекта происходит асинхронно, когда при реальном исполнении программного кода вызов метода является синхронным. Данная особенность приводит к тому, что в данной структуре нет возможности получить значение, возвращаемое методом.

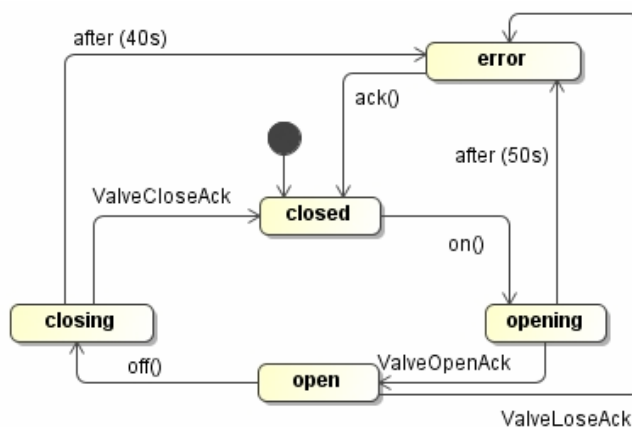


Рис. 1. Пример диаграммы состояния

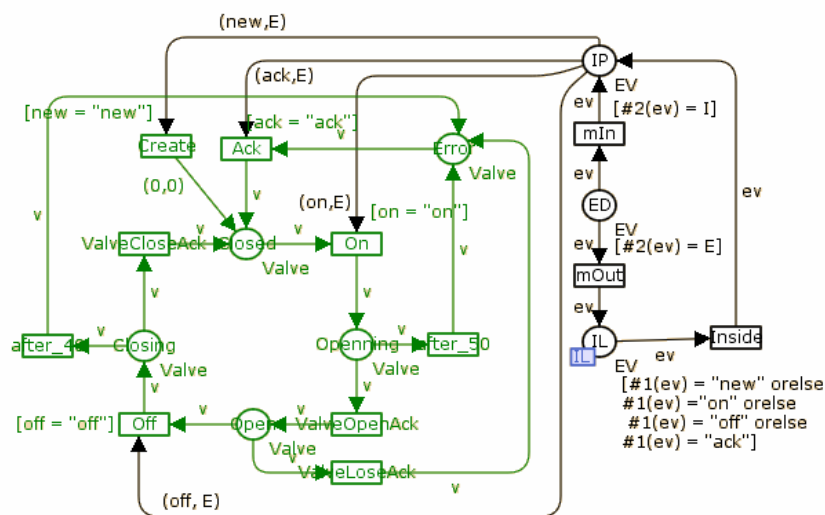


Рис. 2. Сеть Петри, полученная путем преобразования по не модифицированным правилам формального преобразования

Проанализируем работу сети Петри, полученную в результате преобразования той же диаграммы состояний, но при помощи модифицированных правил преобразования. Полученная в результате преобразования сеть Петри представлена на рис. 3. В начальном состоянии

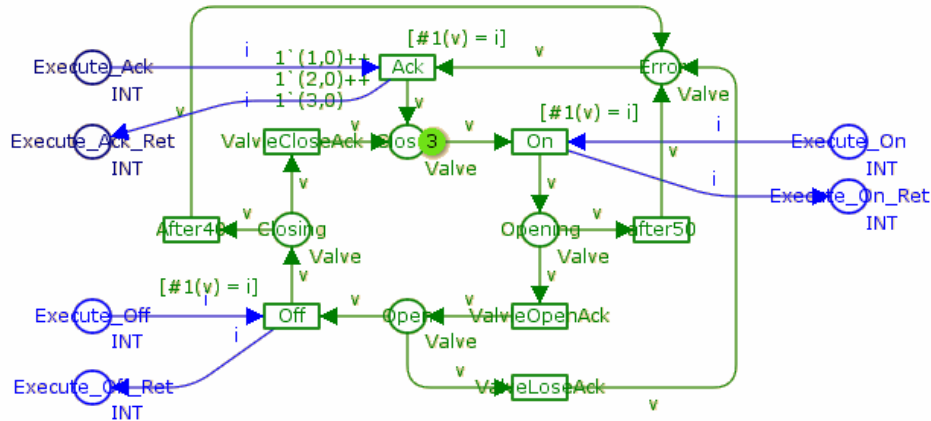


Рис. 3. Сеть Петри, полученная путем преобразования по модифицированным правилам формального преобразования

метки находятся в месте *Closed*. Каждая метка моделирует поведение объекта. Она расширена идентификатором объекта, в данном случае целочисленным числом. Для передачи меток используется переменная v . Места *Execute_off*, *Execute_On*, *Execute_Ack* являются местами, моделирующими вызовы методов объекта. Места *Execute_off_Ret*, *Execute_On_Ret*, *Execute_Ack_Ret* являются местами, моделирующими возвращение результата исполнения метода. Тип входных и выходных меток определяется входными/выходными данными метода. Так же должен быть передан идентификатор объекта, в данном случае целочисленное значение. Другими словами, если метод не имеет параметров, то входное место будет иметь тип *INT*, и если метод имеет значение *void*, то выходное место также будет иметь тип *INT*. При наличии параметров в методе они передаются в метод после идентификатора.

Переходы *On*, *Off* моделируют вызов метода и дополнены защитным условием, определяющим объект, для которого будет вызван данный метод, $-\ [#1(v) = i]$, где выражение $\#1(v)$ – показывает извлечение первого элемента из переменной v . Выражение $= i$ – показывает сравнение элемента с переменной i .

ЗАКЛЮЧЕНИЕ

В данной статье рассмотрены модифицированные правила преобразования UML-диаграммы состояния в сеть Петри. Предложенные правила преобразования позволяют получать из UML-диаграммы сеть Петри, которая позволяет моделировать поведение нескольких объектов одного класса одновременно. Так же благодаря новым правилам преобразования структура сети Петри позволяет получать значение метода, а также моделировать вызов статических методов. Предложенные правила преобразования позволяют отказаться от необходимости придерживаться определенной структуры при построении диаграммы состояний UML и получить однозначное преобразование диаграммы состояния в сеть Петри при произвольной структуре сети. Рассмотрен пример преобразования диаграммы состояния в сеть Петри, основываясь на исходном и модифицированном наборах правил преобразования.

СПИСОК ЛИТЕРАТУРЫ

- [1] **Коротиков С.В.** Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления: дис. канд. техн. наук. – Новосибирск: НГТУ, 2007.
- [2] **Воевода А.А.** О преобразовании UML-диаграмм в сети Петри / А.А. Воевода, Д.О. Романников // XIV Международная научно-техническая конференция – Информационно-вычислительные технологии и их приложения. – Пенза, декабрь 2010. – С. 60–63.
- [3] **Baresi L.** On formalizing UML with high-level Petri net / L. Baresi, M. Pezze // Concurrent object-oriented programming and Petri nets: advances in Petri nets / G.A. Agha, F. De Cindio, G. Rozenberg (Eds.). – Berlin; Heidelberg: Springer, 2003. – P. 276–304. – (LNCS, Vol. 2001).
- [4] **Bernardi S.** From UML Sequence Diagrams and Statecharts to analyzable Petri Net models / S. Bernardi, S. Donatelli, J. Merseguer // Proceeding WOSP '02 Proceedings of the 3rd international workshop on Software and performance. – 2002. – С. 35–45.
- [5] **Kwan-Hee H.** Integration of UML and Petri Net for the Process Modeling and Analysis in Workflow Applications / H. Kwan-Hee, Y. Seock-Kyu, K. Bohyun // Proceedings of the 13th WSEAS International Conference on COMPUTERS. - 2009. – С. 255–262.
- [6] **Romannikov D.O.** Using Timed Petri Nets in approach of software design with UML diagrams / D.O. Romannikov, A.A. Voevoda // Proceedings of DST-RFBR Sponsored Indo-Russian Joint Workshop on Computational Intelligence and Modern Heuristics in Automation and Robotics, 20–22 September 2010, Surat, India.
- [7] **Воевода А.А.** Использование UML-диаграмм и временных сетей Петри в методе разработки ПО ч. 2 / А.А. Воевода, Д.О. Романников // Сборник научных трудов НГТУ. – 2010. – 4 (62). – С. 117–126.

Воевода Александр Александрович, доктор технических наук, профессор кафедры автоматизации Новосибирского государственного технического университета. Основное направление научных исследований – управление многоканальными объектами, исследование свойств UML-диаграмм и сетей Петри. Имеет более 200 публикаций. E-mail: ucit@ucit.ru

Романников Дмитрий Олегович, аспирант кафедры автоматизации Новосибирского государственного технического университета. Тема диссертационного исследования – автоматизация написания программного обеспечения для технических систем. Имеет 12 публикации. E-mail: rom2006@gmail.com

A.A. Voevoda, D.O. Romannikov

Reducing the state space of Petri nets for objects of one class

Transformation of the UML state diagram to Petri net was discussed. New set of the rules of transformation was suggested. New rules allow performing simulation of several objects of one class onetime. Also new rules allow to get result of the method execution and execution of static methods.

Key words: UML, Petri net, CPN, software.