СБОРНИК **НАУЧНЫХ ТРУДОВ НГТУ. – 2010. – № 4(62) –** 117–126

УДК 62-50:519.216

ИСПОЛЬЗОВАНИЕ *UML*-ДИАГРАММ И ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ В МЕТОДЕ РАЗРАБОТКИ ПО Ч. 2*

А.А. ВОЕВОДА, Д.О. РОМАННИКОВ

Рассматривается применение временных сетей Петри в методе разработки ПО с использованием диаграмм *UML*. Приведен следующий набор диаграмм: структурная диаграмма, диаграмма объектов, диаграмма вариантов использования и диаграмма последовательности. Для диаграмм выполнено преобразование в сети Петри и выполнен анализ сети. Рассмотрен пример нахождения логических ошибок в диаграммах и их исправление.

Ключевые слова: *UML*-диаграммы, сети Петри, временные сети Петри.

ВВЕДЕНИЕ

Данная статья является продолжением статьи [1], в которой рассматривается пример использования методики построения ПО с использованием *UML*-диаграмм и временных сетей Петри. В первой части [1] был описан объект, на основании которого рассматривали применение методики разработки [2, 3, 4–8], построен необходимый набор *UML*-диаграмм.

Второй частью рассматриваемого примера является этап преобразования полученных *UML*-диаграмм во временные сети Петри и их анализ.

1. ПРЕОБРАЗОВАНИЕ *UML*-ДИАГРАММ В СЕТИ ПЕТРИ С ИСПОЛЬЗОВАНИЕМ ОТДЕЛЬНЫХ СЕТЕЙ ПЕТРИ ДЛЯ ОБЪЕКТОВ

Преобразование во временные сети Петри является шестым шагом рассматриваемой процедуры построения ПО. Построенные *UML*-диаграммы, полученные в последнем шаге статьи [1], необходимо преобразовать во временные сети Петри. Преобразование необходимо производить при помощи формальных правил. Правила преобразования подробно рассмотрены в п. [1].

Рассмотрим несколько вариантов преобразования: первый – преобразование, при котором структурные диаграммы объекта преобразовываются в *отдельную* сеть Петри. При таком подходе получается большое число одинаковых участков сети, но нет необходимости вводить дополнительные условия для переходов, так как участки сети разделены логически и переход меток от одного объекта к другому исключен.

Основываясь на данном подходе, получим временные сети Петри, представленные на рис. 1, 2, 3.

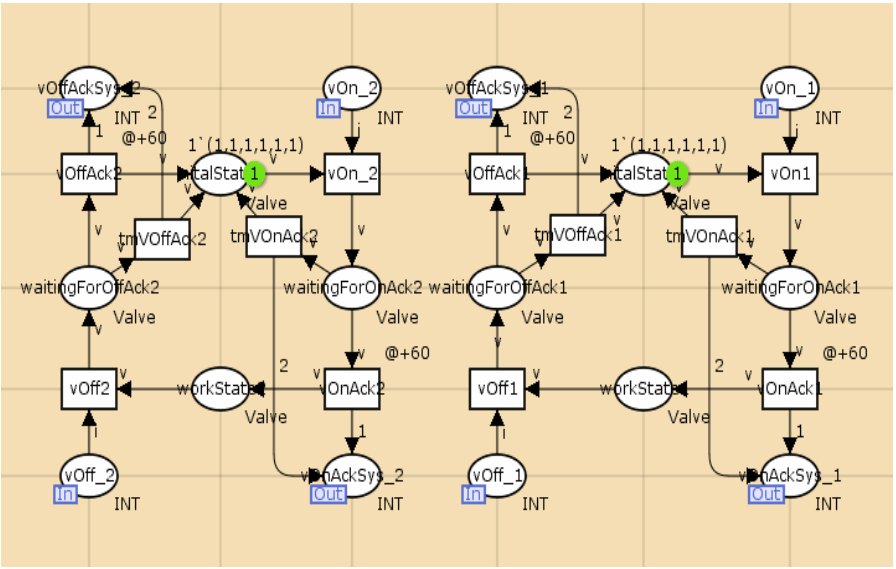

Рис. 1. Сеть Петри, соответствующая *UML*-диаграмме состояний объектов *Valve*

На данных рисунках приведены сети Петри, полученные формальным преобразованием из набора *UML*-диаграмм. На рис. 1 видно, что каждой задвижке системы соответствует свое отображение *UML*-диаграмм. На рис. 3 изображен участок сети Петри, разделенный на две логические части: правая для управления насосом и задвижкой номер один, левая – номер два. Сеть Петри, изображенная на рис. 2, нужна как связка между подсетями рис. 1 и рис. 3, таким образом, она носит только технический характер.

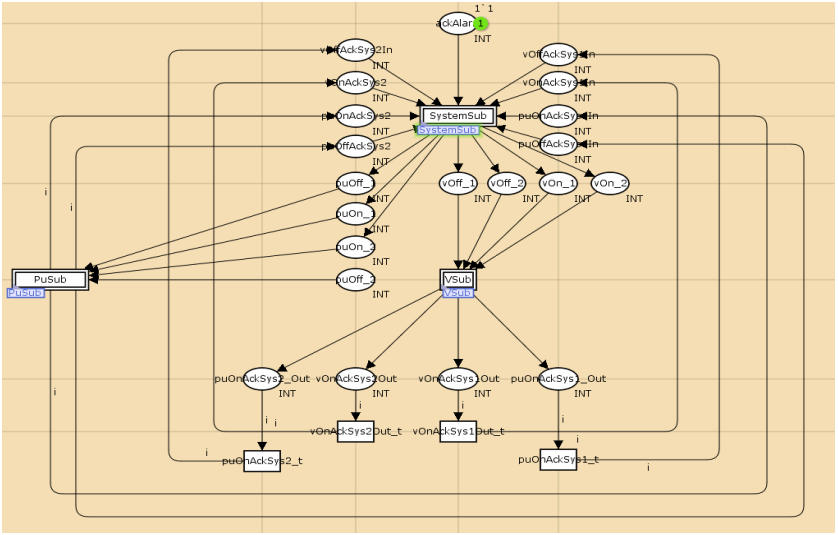

Рис. 2. Сеть Петри, связующая различные диаграммы *UML*

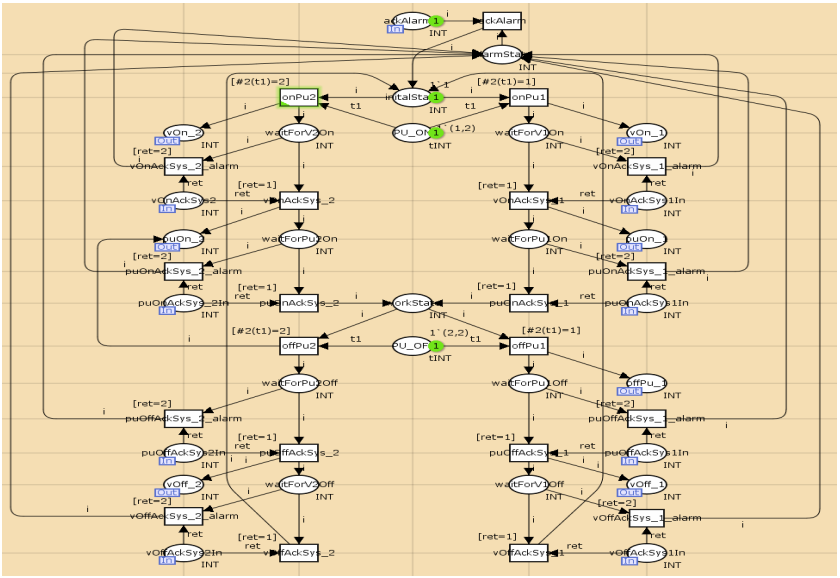

Рис. 3. Основная сеть Петри, соответствующая диаграмме класса SYSTEM

2. ПРЕОБРАЗОВАНИЕ *UML*-ДИАГРАММ В СЕТИ ПЕТРИ С ИСПОЛЬЗОВАНИЕМ ОДНОЙ СЕТИ ПЕТРИ ДЛЯ ОБЪЕКТОВ

Вторым вариантом преобразования является вариант, при котором разделение объектов *UML*-диаграмм происходит при помощи применения граничных условий переходов и дополнительных слоев метки. При использовании такого способа существенно уменьшается размер сети, в общем случае он уменьшается пропорционально количеству объектов системы. Недостатком такого способа является необходимость использования дополнительных условий для переходов сети.

Рассмотрим преобразование того же набора *UML*-диаграмм данным способом преобразования.

Основываясь на данном подходе, получим временные сети Петри, представленные на рис. 4, 5, 6, 7.

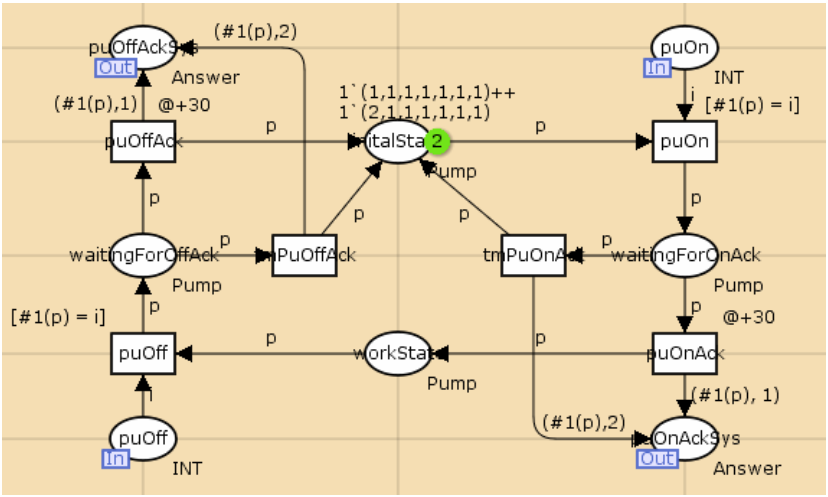

Рис. 4. Сеть Петри, соответствующая *UML*-диаграмме состояний объектов Pump

На данных рисунках приведены сети Петри, полученные формальным преобразованием из набора *UML*-диаграмм. На рис. 4, 5 видно, что для объектов используется один участок сети, а логическое разделение объектов достигается при помощи введения дополнительных условий, ограничивающих переходы.

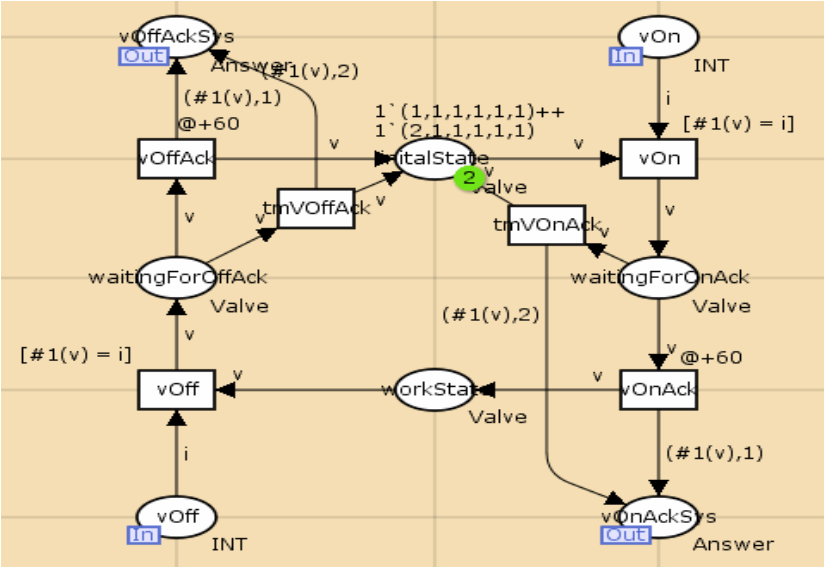

Рис. 5. Сеть Петри, соответствующая *UML*-диаграмме состояний объектов *Valve*

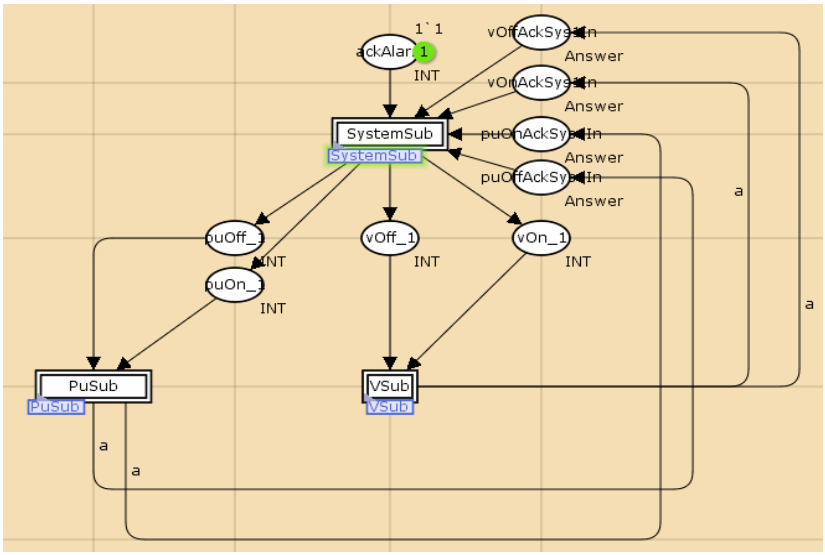

Рис. 6. Сеть Петри, связующая различные диаграммы *UML*

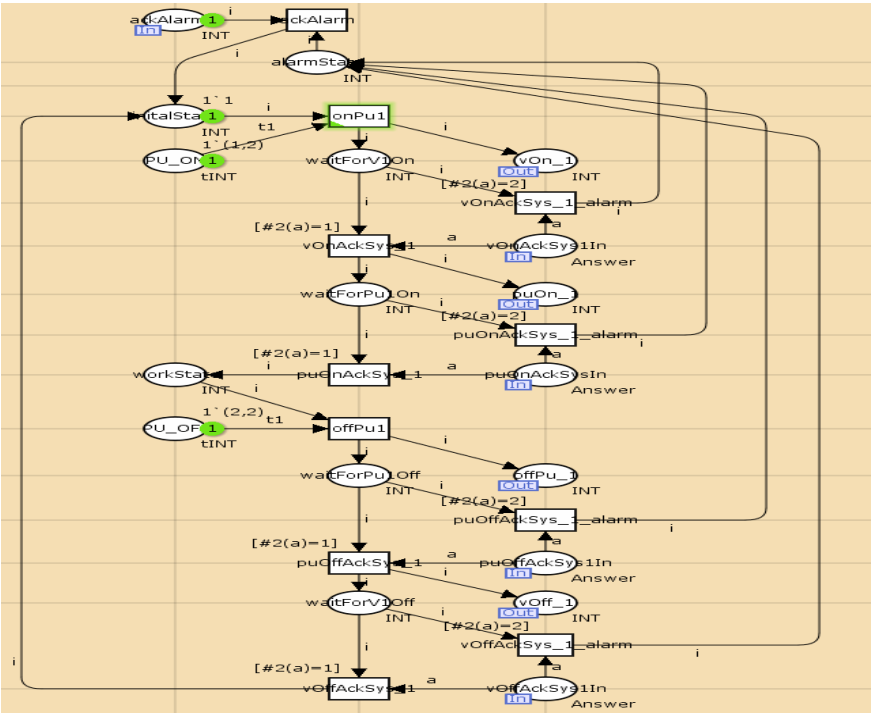

Рис. 7. Сеть Петри, соответствующая диаграмме класса SYSTEM

Сеть Петри, изображенная на рис. 7, выполняет идентичную функциональность, что и сеть Петри, изображенная на рис. 3. Сеть Петри на рис. 6 служит связкой сетей на рис. 7 и рис. 4, 5. Стоит отметить, что сети Петри, полученные в данном пункте значительнее компактнее, чем сети, из предыдущего пункта.

3. АНАЛИЗ СЕТЕЙ ПЕТРИ

Рассмотрим сравнительный анализ для каждого из приведенных выше способов преобразования.

Из результатов, приведенных в табл. 1 и 2, видно, что при одинаковой функциональности обоих подходов к преобразованию сложность сети Петри для первого варианта выше – большее количество узлов и дуг.

Таблица 1

Результаты моделирования для первого варианта преобразования

Результаты отчета	Выводы
State Space Nodes: 465 Arcs: 828 Secs: 0 Status: Full	Пространство состояний модели вычислено полностью и содержит 465 узел, 828 дуги и 0 секции
Home Markings Initial Marking is not a home marking Dead Markings 52 [98,97,95,83,81,...] Dead Transition Instances None Live Transition Instances None	Не все маркировки являются «домашними» – модель (алгоритм) не обратима. В сети присутствуют «мертвые» маркировки, что означает наличие в модели тупиковых состояний. В данном случае тупиковые состояния появляются из-за ограниченности команд оператора, т.е., когда закончатся метки, имитирующие команды от оператора, система зафиксирует свое состояние
Fairness Properties No infinite occurrence sequences	Нет бесконечных последовательностей срабатывания

Таблица 2

Результаты моделирования для второго варианта преобразования

Результаты отчета	Выводы
State Space Nodes: 335 Arcs: 634 Secs: 0 Status: Full	Пространство состояний модели вычислено полностью и содержит 335 узел, 634 дуг и 0 секции
Home Markings Initial Marking is not a home marking Dead Markings 21 [85,84,83,82,41,...] Dead Transition Instances None Live Transition Instances None	Не все маркировки являются «домашними» – модель (алгоритм) не обратима. В сети присутствуют «мертвые» маркировки, что означает наличие в модели тупиковых состояний. В данном случае тупиковые состояния появляются из-за ограниченности команд оператора, т.е., когда закончатся метки, имитирующие команды от оператора, система зафиксирует свое состояние
Fairness Properties No infinite occurrence sequences	Нет бесконечных последовательностей срабатывания

4. АНАЛИЗ СЕТЕЙ ПЕТРИ ДЛЯ СЕТИ ИЗ «ТРЕХ» ОБЪЕКТОВ

Рассмотрим схему, аналогичную рассматриваемой в первой части статьи [1], с включенным в нее параллельно дополнительным насосом и задвижкой.

Для первого варианта преобразования сетей Петри для каждой диаграммы необходимо добавить участки сети, необходимые для функционирования насоса, задвижки и логики включения. Для второго варианта включения необходимо лишь изменить количество меток для начальных мест насосов, задвижек и команд оператора. Рассмотрим количественные показатели результатов моделирования получившихся сетей.

Таблица 3

Результаты моделирования схемы из трех объектов для первого и второго вариантов преобразования

Результаты отчета	Выводы
State Space Nodes: 16767 Arcs: 48330 Secs: 67 Status: Full	**Первый вариант преобразования.** Пространство состояний модели вычислено полностью и содержит 16767 узел, 48330 дуг и 67 секции
State Space Nodes: 5191 Arcs: 14814 Secs: 8 Status: Full	**Второй вариант преобразования.** Пространство состояний модели вычислено полностью и содержит 5191 узел, 14814 дуг и 8 секции

ЗАКЛЮЧЕНИЕ

В данной статье рассмотрены два подхода к преобразованию *UML*-диаграмм в сети Петри. Как следует из результатов моделирования сетей, второй вариант, когда одинаковые объекты сети преобразовываются в один участок сети Петри, а логическое разделение производится при помощи меток и дополнительных условий переходов, приводит к увеличению количества объектов в данной структуре сети.

[1] *Воевода А.А. Романников Д.О.* Использование *UML* и временных сетей Петри при разработке программного обеспечения. Ч. 1 / Сб. науч. тр. НГТУ. – № 3(61). – 2010. – С. 61–70.

[2] *Романников Д.О.* Использование *UML*-диаграмм и сетей Петри в разработке программного обеспечения систем распределенной обработки информации, критических ко времени исполнения / Магистерская диссертация, НГТУ, 2010.

[3] *Коротиков С.В.* Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления / Дис. канд. техн. наук. – Новосибирск: издво-во НГТУ, 2007.

[4] *Воевода А.А. Романников Д.О. Зимаев И.В.* Применение *UML*-диаграмм и сетей Петри при разработке встраиваемого программного обеспечения. Научный вестник НГТУ. – 2009. – № 4(37). – С. 169–174.

[5] *Воевода А.А. Романников Д.О.* Моделирование сетей Петри в CPN Tools / Сб. науч. тр. НГТУ. – № 3(53). – 2008. – С. 49–54.

[6] *Воевода А.А. Романников Д.О.* О моделировании систем реального времени с использованием UML и сетей Петри / Сб. науч. тр. НГТУ. – № 1(55). – 2009. – С. 63–66.

[7] *Воевода А.А. Романников Д.О.* О проектировании программного обеспечения для микроконтроллеров с использованием *UML* / Сб. науч. тр. НГТУ. – № 4(58). – 2009.

[8] *Воевода А.А. Романников Д.О.* О комактном представлении языков раскрашенных сетей Петри / Сб. науч. тр. НГТУ. – № 3(53). – 2008. – С. 105–108.

Воевода Александр Александрович **–** профессор кафедры автоматик Новосибирского государственного технического университета.

E-mail: ucit@ucit.ru.

Романников Дмитрий Олегович – аспирант кафедры автоматики Новосибирского государственного технического университета.

E-mail: rom2006@gmail.com.

A.A. Voevoda, D.O. Romannikov

UML-diagramm and Timed Petri nets using in the approach of software design p.2

Article considers an approach of using *UML*-diagrams and Timed Petri nets for designing software. Set of *UML*-diagram (object diagram, statechart diagrams, and structure diagrams) is recommended to use for designing software systems. Rule of modeling of changing information between objects suggested. Considered an example of logic errors in the diagrams.

Key words: diagram, linear system, stability.