

## **СОВОКУПНОЕ ИСПОЛЬЗОВАНИЕ СЕТЕЙ ПЕТРИ И UML-ДИАГРАММ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ\***

А.В. МАРКОВ, Д.О. РОМАННИКОВ

Рассматривается применение иерархических сетей Петри в методике разработки программного обеспечения с использованием диаграмм UML. Приводится построение диаграммы прецедентов по словесному описанию системы. Строится сеть Петри, отвечающая за поведенческие аспекты работы. Полученная сеть тестируется и преобразуется в диаграмму деятельности, после чего происходит построение диаграммы классов и объектов.

**Ключевые слова:** инженерия ПО, UML диаграммы, сети Петри, CPN Tools, пространство состояний.

### **ВВЕДЕНИЕ**

Данная статья является частью рассматриваемого примера использования методики разработки программного обеспечения с применением диаграмм UML и аппарата сетей Петри. Предложенная методика была сформулирована [1] на основе методик [2, 3] и примеров [4, 6, 8]. Синтезированную методику можно применять для систем, поведение которых нуждается в детальной проработке.

Примером выбрана система «умный светофор», построенная согласно методике [1] итерационно с пояснениями для каждого шага.

### **1. ОПИСАНИЕ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ «УМНЫЙ СВЕТОФОР»**

**Первый шаг** включает в себя сбор требований, предъявляемых к системе.

Разрабатываемая система «умный» светофор, улучшающая регулирование потока автотранспорта, включает в себя:

- светофоры T11, T12, T13, T14;
- IP-HUB HUB;
- видеокамеры Vc1, Vc2, Vc3, Vc4.

Работа системы будет осуществляться в двух режимах: автоматическом (переключение световых сигналов светофора происходит за фиксированный

период времени), ручного управления (переключение световых сигналов светофора осуществляется оператором-специалистом).

Система начинает свою работу с включения всех светофоров на перекрестке. Изначально светофоры будут функционировать в автоматическом режиме. Каждый светофор будет иметь шесть состояний (режимов) работы. После включения всех светофоров и подтверждения корректности их работы система может переключиться на ручное управление. При поломке светофора подаются звуковой и световой сигналы на пульте управления.

Переход на ручное управление будет осуществляться постепенно.

Вначале необходимо включить IP-HUB (устройство, предназначенное для передачи цифрового видеосигнала по сетям связи, поддерживающим протокол ТСР/ІР (обычно Internet) на единый центр мониторинга без потери качества). За подтверждением включения следует присвоение IP-адреса HUB'у.

Затем происходит включение четырех IP-видеокамер (стационарно установленная камера, имеющая встроенный IP-сервер, сетевой интерфейс, не нуждается в прямом подключении к аппаратным средствам оцифровки видеосигнала), каждая установлена в одном из направлений перекрестка. После подтверждения включения видеокамер ей присваивается внутренний IP-адрес в системе для дальнейшего мониторинга.

При корректном включении IP-камер и HUB система может быть переведена на ручное управление. Переход из автоматического режима к ручному будет производиться по завершению последнего режима работы светофора.

При незапланированном отключении, поломке, аварии видеокамеры или HUB в случае необходимости будет вызываться ремонтная бригада, а также осуществляться автономный переход системы в автоматический режим; при окончании смены оператора-специалиста перевод из ручного управления производится самим оператором.

## 2. СИНТЕЗ ДИАГРАММЫ ПРЕЦЕДЕНТОВ

Руководствуясь рассматриваемой методикой разработки на **втором этапе** происходит *синтез диаграммы прецедентов, устроенный на основе требований к системе в форме технического задания или словесного описания.*

Диаграмма прецедентов представлена на рис.1. На ней изображены два актера Operator и Administrator. Прецеденты onT11, onT12, onT13, onT14, offT11, offT12, offT13, offT14 демонстрируют возможность оператора включать, выключать светофоры, соответственно. Прецеденты onHUB, off HUB (включение/отключение HUB) расширены прецедентами onVc1, onVc2, onVc3, onVc4, offVc1, offVc2, offVc3, offVc4, означающими включение/отключение камер. askAlarm показывает, что оператор может квитировать аварию системы. Пре-

цедент `viewRegistrationInfo` означает возможность просмотра оператором информации о состоянии системы.

Ещё один актер – администратор системы (Administrator), который наследует все прецеденты администратора и может выполнять дополнительные. `addOperator` – добавление в систему новых операторов, `setNewSettings` – изменение параметров, влияющих на работу системы, `getReport` – просмотр отчета о работе системы.

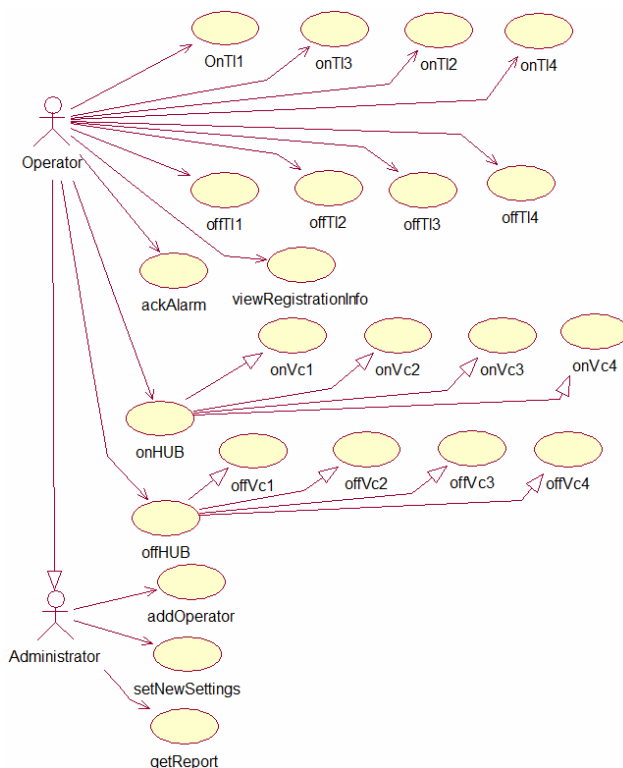


Рис. 1. Диаграмма прецедентов для системы «умный светофор»

### 3. ОПИСАНИЕ ДИНАМИКИ РАБОТЫ СИСТЕМЫ

**Третий шаг** в предложенной методике заключается в *составлении сети Петри*, которая показывает динамические свойства системы. Данная сеть

Петри применяется для моделирования последовательных (а иногда и параллельных) шагов вычислительного процесса.

Работа сети начинается с подтверждения возможности обмена информацией системы со светофором «tIAsk», после чего происходит включение светофора «tIOn», затем подтверждение включения светофоров программой «tIOnAckSys» (рис. 5). Работа самого светофора показана на рис. 6.

Как говорилось ранее, переход на ручное управление будет осуществляться после полного и корректного включения HUB'a «hubOn» (рис. 2), IP-камер «vcOn» (рис. 3), а также безотказной работы в автоматическом режиме «Switching on a hand control» (рис. 7).

Рассмотрим более детально один из сценариев работы при ручном управлении. Допустим, что произошёл сбой в работе одной из камер «Video camera(’s) breakage». Система подаст сигнал аварии «vcGetAlarm», будет вызвана ремонтная бригада «callOfARepairBrigade 1», а система будет переведена в автоматический режим до восстановления работы вышедшей из строя камеры.

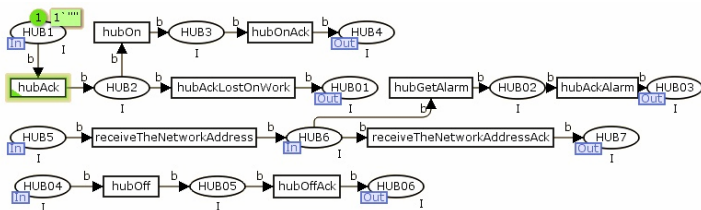


Рис. 2. Подстраница сети «HUB»

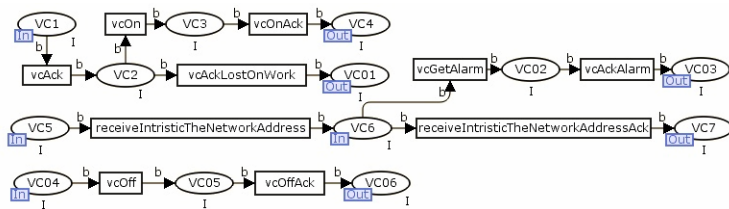


Рис. 3. Подстраница сети «Video\_camera»

Для более наглядного представления была создана иерархическая сеть, которая отражает работу оборудования (светофоров – подстраница сети «Traffic\_light» (рис. 4), HUB – подстраница сети «HUB», IP-камер – подстра-

ница сети «Video\_camera»), а также работу в автоматическом режиме (сеть «Automatic work») и при ручном управлении (сеть «Hand control»).

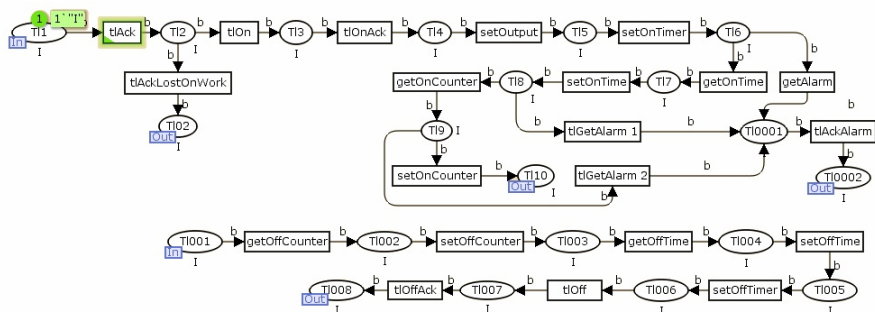


Рис. 4. Подстраница сети «Traffic\_light»

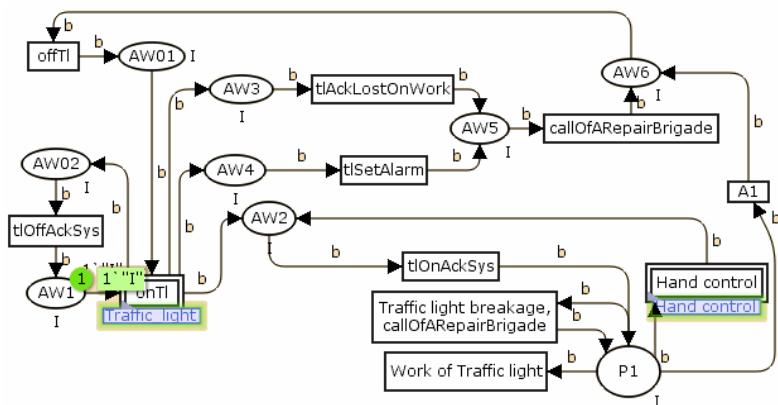


Рис. 5. Страница сети «Automatic work»

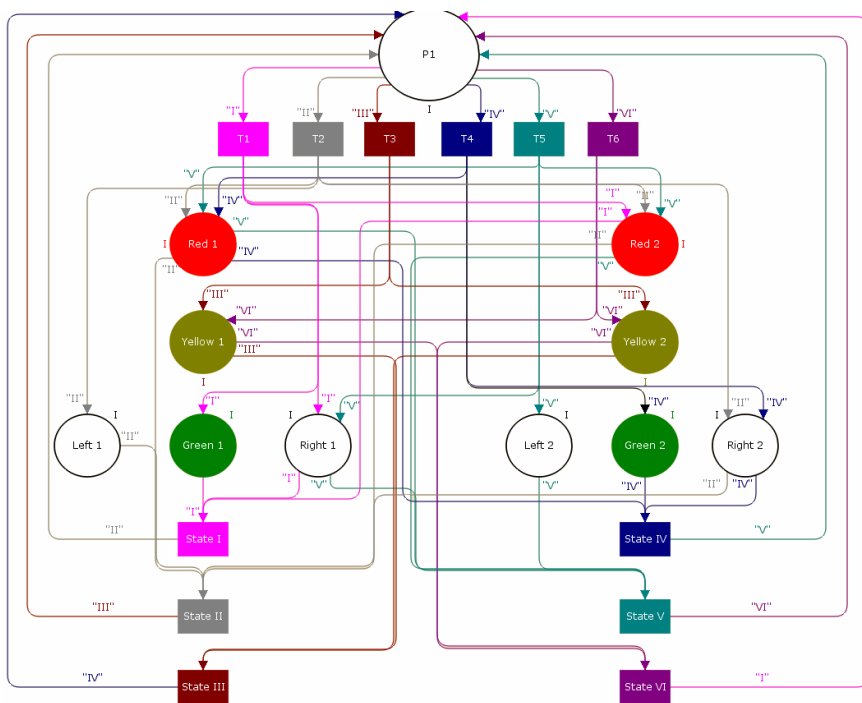


Рис. 6. Подстраница сети «Work of Traffic light»

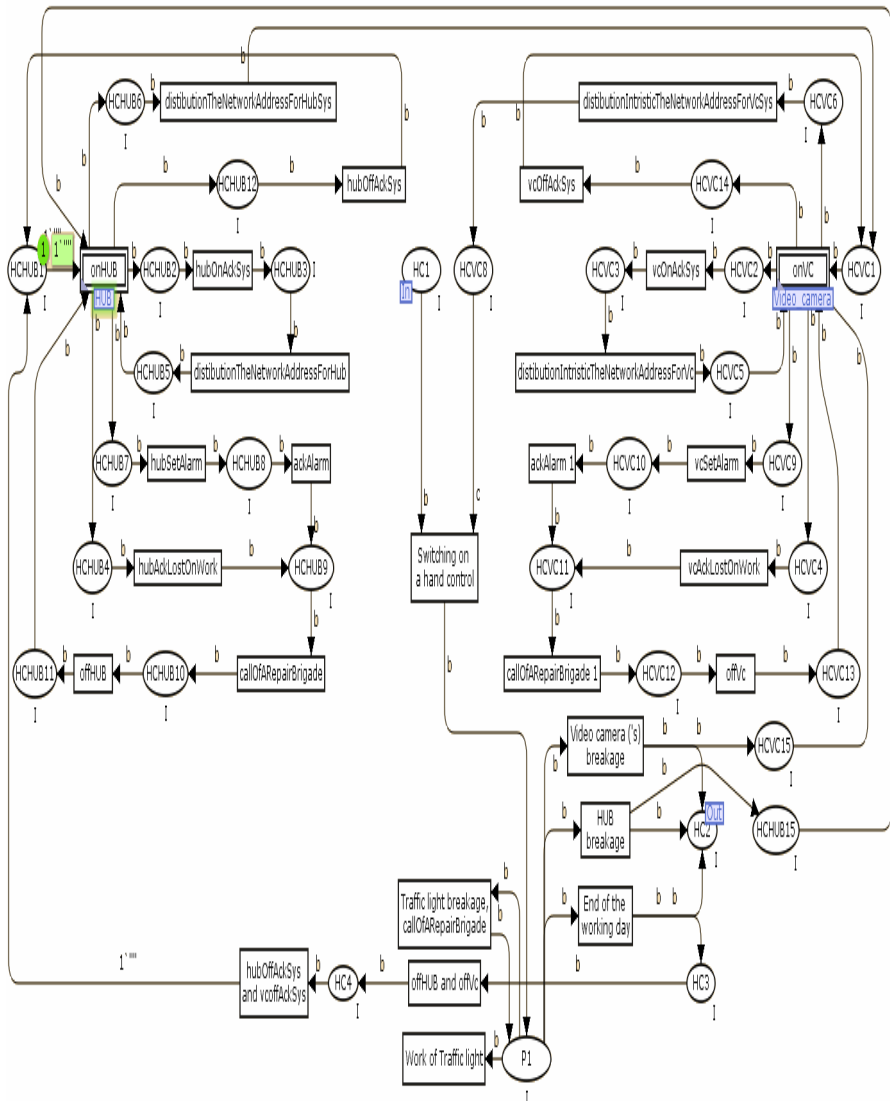


Рис. 7. Страница сети «Hand control»

**На четвёртом шаге** полученная сеть тестируется и производится определение свойств сети. При наличии дефектов в полученном отчете нужно установить источник ошибки и произвести корректировку сети.

Выбранная начальная маркировка сети показывает, что система начинает работу с включения светофоров, а переход на ручное управление возможен после последовательного включения HUB и видеокамер.

Тестирование сети осуществляется при помощи отчёта о пространстве состояний, сгенерированного программной средой CPN Tools. Также тестировать сеть можно с помощью других способов анализа: дерева достижимости и свободного языка сети [5].

#### Результаты моделирования сети при выбранной начальной маркировке

Результаты отчета	Выводы
State Space Nodes: 38562 Arcs: 89580 Secs: 172 Status: Full	Пространство состояний системы вычислено полностью за 172 секунд, содержит 38562 узлов, 89580 дуг.
Home Markings All Dead Markings None Dead Transition Instances None Live Transition Instances All	Все маркировки «домашние» – модель (алгоритм) обратима. В сети отсутствуют «мертвые» маркировки, следовательно нет в модели тупиковых состояний. Все переходы живые, мертвых переходов нет.
Fairness Properties	
Impartial Transition Instances None	Бесконечная последовательность срабатывания отсутствует
Fair Transition Instances Hand_control'State_I 1 Hand_control'State_II 1 Hand_control'State_III 1 Hand_control'State_IV 1 Hand_control'State_V 1 Hand_control'State_VI 1	Переходы, соответствующие правилам функционирования системы.
Just Transition Instances Hand_control'Traffic_light_breakage 1	Справедливость срабатывания перехода Hand_control'Traffic_light_breakage 1 обоснована

Основываясь на приведенном отчете (см. таблицу), можно сделать вывод: сеть Петри, отвечающая за динамические свойства, является корректно по-



строенной и может использоваться как основа при проектировании программного обеспечения для системы «умный светофор».

**Пятым шагом** сеть Петри, работа которой протестирована, а полученные свойства сети удовлетворяют требованиям, транслируется в диаграмму деятельности для отражения динамических свойств системы.

Трансляция осуществляется по формальным правилам, изложенным в [3, 9].

**На шестом и седьмом шаге** происходит построение диаграммы классов и диаграммы объектов, соответственно.

## ЗАКЛЮЧЕНИЕ

В данной статье был показан пример пошагового проектирования системы по методике написания программного обеспечения с использованием UML-диаграмм и сетей Петри. Наглядно разобраны часть этапов создания программного обеспечения от постановки задачи и словесного описания до анализа построенной сети Петри, отражающей логику работы проектируемой системы.

[1] *Марков А.В.* Автоматизация разработки программного обеспечения с использованием сетей Петри: магистерская диссертация, 2011.

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

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

[4] *Воевода А.А. Марков А.В.* О Тестирование UML-диаграмм с помощью аппарата сетей Петри на примере разработки ПО для игры «Змейка». Сборник научных трудов НГТУ. – 2010. – № 3. – С. 51–61.

[5] *Воевода А.А. Марков А.В.* О компактном представлении языков сетей Петри: сети с условиями и временные сети. Сборник научных трудов НГТУ. – 2010. – № 2. – С. 77–83.

[6] *Марков А.В.* Моделирование процесса поиска пути в лабиринте при помощи сетей Петри. Сборник научных трудов НГТУ. – 2010. – № 4. – С. 133–141.

[7] Романников Д.О. Марков А.В. Обзор работ посвященным разработке ПО с использованием UML и сетей Петри. Сборник научных трудов НГТУ. – 2011. – № 1.

[8] Воевода А.А. Романников Д.О. Использование UML и временных сетей Петри при разработке программного обеспечения. Сборник научных трудов НГТУ. – 2010. – № 3. – С. 61–70.

[9] Зимаев И.В. О возможности автоматической трансляции UML-диаграмм деятельности с сети Петри. Сборник научных трудов НГТУ. – 2009. – № 1. – С. 57–63.

**Марков Александр Владимирович** – магистрант кафедры автоматики Новосибирского государственного технического университета по направлению инженерия ПО. E-mail: muviton3@mail.ru

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

E-mail: rom2006@gmail.com

**A.A. Markov, D.O. Romannikov**

### **Cumulative usage of Petri nets and UML diagrams at software development**

Are considered application hierarchical Petri nets in a technique of software development with usage of UML diagrams. Creation of the use case diagrams under the verbal description of system is resulted. The Petri net which is responsible for behavioral aspects of operation is under construction. The received network is tested and will be transformed to activity diagram then there is a creation of the class diagram and object diagram.

**Key words:** software engineering, UML-diagrams, Petri nets, CPN Tools, state space.