

А.А. ВОЕВОДА, А.В. МАРКОВ, Д.О. РОМАННИКОВ  
**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ:  
ПРОЕКТИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ UML ДИАГРАММ  
И СЕТЕЙ ПЕТРИ НА ПРИМЕРЕ АСУ ТП ВОДОНАПОРНОЙ  
СТАНЦИИ**

---

*Воевода А.А., Марков А.В., Романников Д.О. Разработка программного обеспечения: проектирование с использованием UML диаграмм и сетей Петри на примере АСУ ТП водонапорной станции.*

**Аннотация.** Рассматриваются и анализируются некоторые из известных способов и методик разработки программного обеспечения (ПО) на этапе проектирования. Описывается методика, в которой используются UML диаграммы для моделирования свойств и динамики работы исследуемой системы, а также сети Петри для анализа динамических свойств спроектированных поведенческих диаграмм. Приводится усовершенствованная методика взаимного использования UML диаграмм и сетей Петри. Предлагаемая методика демонстрируется на примере системы АСУ ТП насосной станции, а именно, проектируется диаграмма прецедентов, диаграмм классов и диаграмм объектов, а также диаграмма деятельности, преобразование которой реализуется по формальным правилам. При анализе сети Петри, транслированной из диаграммы деятельности, были выявлены некорректные состояния, в которых оказывалась система при включении и выключении насосов оператором. После устранения данной ситуации, ошибок в динамике работы системы не осталось. В результате анализа получено представление о размере дерева достижимости системы, составляющее величину порядка  $10^6$ .

**Ключевые слова:** проектирование ПО, UML диаграммы, сети Петри, диаграмма классов, диаграмма объектов, диаграмма деятельности, пространство состояний, алгоритм, ПО для водонапорной станции.

*Voevoda A.A., Markov A.V., Romannikov D.O. Software development: software design using UML diagrams and Petri nets for example automated process control system of pumping station.*

**Abstract:** Methods and techniques of software design as one of the important stages of software development are described in the paper. The method of software design with using of UML with Petri nets for analyzing of dynamic properties of set UML diagrams is described. Authors offer improved method of using of integration of UML diagrams and Petri nets. The offered method was used for designing of software of automated process control system (APCS) of pumping station: designing of use case, class, object diagrams and sequences diagram that was transformed to Petri net with help of formal rules. Some incorrect states that occurred after pumps enabling/disabling by operator were identified by analysis of Petri net. Reachability tree of the system was gotten by analysis of the Petri net (the value of the tree is about  $10^6$  of nodes). Testing of offered system was showed on example of APCS of pump station.

**Keywords:** software design, UML diagrams, Petri nets, class diagram, object diagram, activity diagram, state space algorithm, software for pumping station.

---

**1. Введение.** Использование программного обеспечения (ПО) позволяет не только проще управлять сложными бизнес процессами,

но и помогает в повседневной жизни человека. Процесс создания ПО можно условно разбить на несколько этапов: понимание природы и сущности проблематики, которую решает программный продукт, сбор требований к системе, проектирование программного обеспечения и его реализация, тестирование, выпуск и сопровождение программного продукта. Одним из важнейших этапов является этап проектирования. При проектировании различных систем используют язык описаний и спецификаций SDL, ИБТ-сети, Workflow модели, графический язык UML<sup>1</sup>, математический аппарат сетей Петри<sup>2</sup>; а также различные алгоритмы: машина Тьюринга, нормальный алгоритм Маркова и др. Часто используется графический язык UML, который позволяет проектировать классы и объекты системы и рассматривать динамику работы их методов, а также моделировать статические свойства системы. Несомненным преимуществом UML является генерация полных диаграмм в коды нужного языка программирования, но существует недостаток – отсутствие чётких формальных правил построения и возможности анализа смоделированных диаграмм на выявление логических ошибок (зацикливаний, тупиковых состояний системы). В данной ситуации целесообразно применение математического аппарата сетей Петри, который разрабатывался для выявления ошибок в логике работы систем. Совместное использование UML диаграмм и сетей Петри рассматривается на задачах локальной автоматизации [1] и центре дистанционного управления и контроля работы таксофонов [2], а также на задаче перемещения манипулятора в пространстве с препятствиями [3]. Система, представленная в [1],

---

<sup>1</sup> UML – язык графического описания систем, использование которого позволяет увеличить скорость разработки программного продукта и уменьшить количество синтаксических, семантических и др. видов ошибок. Существует возможность автоматического преобразования UML диаграмм в программный код, а изменения в программном коде – обратно в UML (например, Rational Rhapsody).

<sup>2</sup> Сеть Петри – двудольный ориентированный граф, который состоит из вершин двух типов – позиций и переходов, соединенных между собой дугами, вершины одного типа не могут быть соединены непосредственно. В позициях могут размещаться метки (маркеры), способные перемещаться по сети. Сеть Петри представляют упорядоченным множеством  $N = (P, T, F, m_I)$ , где  $P = \{p_1, p_2, \dots, p_n\}$  – множество мест,  $T = \{t_1, t_2, \dots, t_m\}$  – множество переходов, таких что  $P \cap T = \emptyset$ ,  $F \subseteq P \times T \cup T \times P$  – отношение, а  $m_I: P \rightarrow N$  – начальная маркировка.

рассмотрена более детально, а часть, отвечающая за локальную автоматизику, детализирована.

**2. Методика совместного использования UML диаграмм и сетей Петри.** В работах [1-2, 4-8] приведены способы и методики взаимного использования UML диаграмм и сетей Петри, предлагаются правила и способы преобразования диаграмм в сети Петри. В работах [4-7] рассмотрена проблематика, а также предложены решения задачи преобразовании поведенческих диаграмм в сети Петри, которая основана на разработке правил преобразования отдельных элементов и групп элементов UML диаграмм в сети Петри. Предложенные правила преобразования [4-7] очень схожи с правилами, предложенными [2], но последняя упомянутая работа содержит более полный набор правил<sup>3</sup>. В работе [4] предложен способ для реализации автоматической трансляции UML диаграммы деятельности в сеть Петри, а в работе [9] предложен способ и детально описан алгоритм для реализации этой задачи. Отметим, что существует возможность реализовать методику проектирования в один этап [2], что способствует сокращению итераций моделирования диаграмм, изменить последовательность построения и проверки диаграмм, а также отказаться от использования диаграммы состояния [1, 2] и заменить выявление прецедентов, которые необходимо проверить более детально, на выявление соответствующих классов системы. В методиках [1, 2] используются диаграммы прецедентов – для сбора информации о системе, диаграммы классов и диаграммы объектов – для формирования сущностей, а проверка динамики работы системы реализуется при помощи диаграммы деятельности и/или диаграммы состояний.

При анализе методик и способов взаимного использования UML диаграмм и сетей Петри замечено, что присутствует лишняя детализация действий при проектировании диаграмм, а также возможность избежать построения диаграмм, наличие которых не критично. Изменив очерёдность построения диаграмм, представленных выше, и проверку их при помощи сетей Петри, сделав её более рациональной и эффективной, предлагается методика проектирования программного обеспечения (листинг 1).

---

<sup>3</sup> *Коротиков С.В.* Применение цветных иерархических сетей Петри для верификации UML-диаграмм на этапе анализа требований к системе дистанционного контроля и управления // Сборник трудов НГТУ. 2007. №1(47). С. 81–92.

```

analysis {
translation diagram to Petri net
analysis Petri net
if results =  $\rightarrow$  satisfactory, then
    correct Activity Diagram
else;
end if}

1: do Use Case Diagram
2: do Class Diagram
3: for needed class  $\in$  Class Diagram do Activity Diagram
4: analysis() of Activity Diagram
5: do Object Diagram
6: for needed object  $\in$  Object Diagram do Activity Diagram
7: analysis() of Object Diagram
8: do Activity Diagram for system
9: analysis() of Activity Diagram for system
10: generation code

```

Листинг 1. Методика взаимного использования UML диаграмм и сетей Петри

Разработанная методика (рисунок 1) состоит из семи этапов и включает в себя составление диаграммы прецедентов, которая позволяет выявить все сущности в виде актёров и вариантов использования; составление диаграммы классов с выделением свойств и методов класса; составление диаграммы объектов, участвующих в системе, для определения маркировки всех вершин у сети Петри.

Следующим этапом является построение диаграммы деятельности для описания динамических свойств системы. Полученная диаграмма транслируется в сеть Петри, опираясь на формальные правила преобразования. Используя автоматизированные программные пакеты для работы с сетями Петри, производится анализ, позволяющий выявить логические ошибки при моделировании диаграммы деятельности (нахождение неиспользуемых сценариев работы, выявление тупиковых веток алгоритмов, проверка достижимости состояний и т.п.) [10]. При получении положительных результатов по завершению проведения анализа и наличии полного набора диаграмм, смоделированная при помощи языка UML система готова для автоматической генерации кода. Если в системе присутствуют классы и объекты, которые нуждаются в детальной проработке логики их работы, то следует спроектировать диаграмму деятельности с последующим анализом отдельно. При необходимости можно воспользоваться пространством состояний или его упорядоченной версией – деревом достижимости, для наблюдения за динамикой работы по состояниям системы.

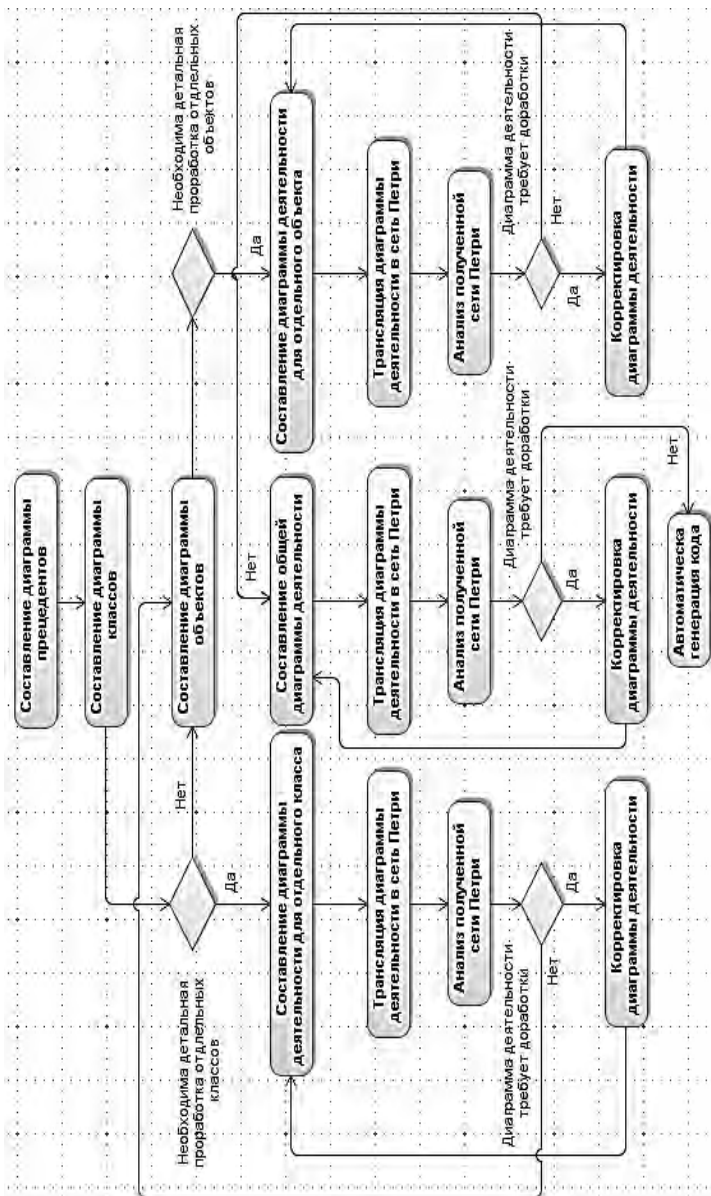


Рис. 1. Методика проектирования ПО при совместном использовании UML диаграмм и сетей Петри

**3. Проектирование АСУ ТП насосной станции по методике совместного использования UML диаграмм и сетей Петри.** Водонапорная станция представляет собой комплекс, состоящий из определенного набора насосов, задвижек и резервуаров для очистки и хранения воды (рисунок 2). Опишем её более детально.

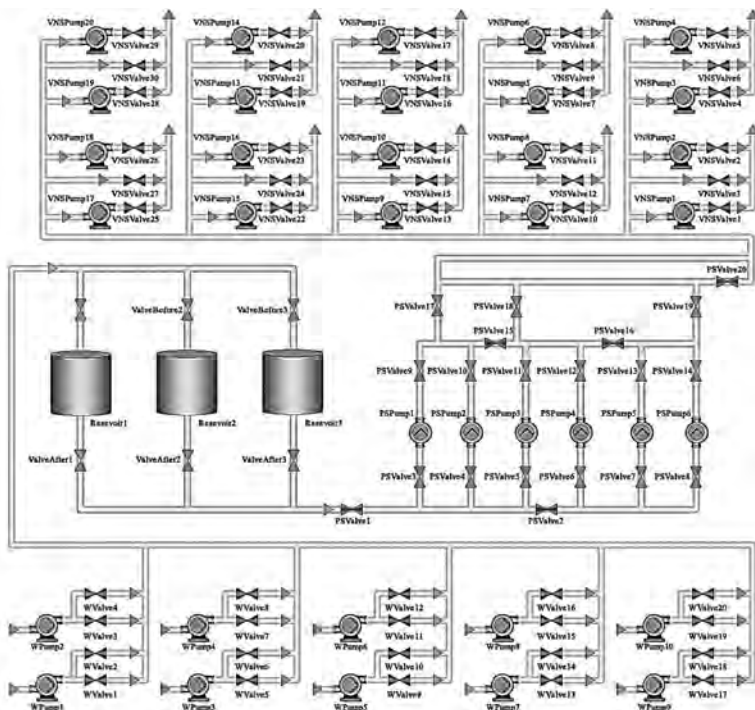


Рис. 2. Функциональная схема АСУ ТП водонапорной станции

На первых этапах происходит перекачка воды  $n$  количеством насосов ( $WPump\ n$ ) из скважин при открытых задвижках ( $WValve\ n$ ) до резервуаров ( $Reservoir\ n$ ). В резервуарах происходит очистка воды и дальнейший её перегон до насосной станции второго подъема, которая при помощи шести насосов ( $PSPump\ n$ ) перекачивает воду до локальных насосных станций. Локальные водонапорные станции состоят из двух насосов ( $VNSPump\ n$ ) и трех задвижек ( $VNSValve\ n$ ). Насосы поочередно включаются через выбранное заранее время, при поломке одного из насосов, включается другой. Если произошло отключение обоих насосов, то происходит открытие третьей задвижки и вода самотёком поступает до части потребителей. Данная система управляется оператором, который открывает задвижки, включает

насосы и следит за состоянием всех элементов (скважинные насосы, локальные насосы, задвижки на всех уровнях системы, резервуары с чистой водой и т.д.).

Необходимо поддерживать давление, равное пяти атмосферам, в трубопроводе локальной водонапорной станции, при колебании давления необходима его корректировка с помощью насосной станции, т.е. увеличением или уменьшением мощности его работы.

Функционирование технологического процесса происходит при поддержания и регулировании аналоговых величин. Для создания ПО на этапе проектирование выбирается несколько состояний дискретности для описания и моделирования величин, нуждающихся в регулировании: давление в трубопроводе, мощность работы двигателя насоса, а также состояния насосов, задвижек и резервуаров.

Во время проектирования диаграммы прецедентов системы, способствующей лучшему пониманию всей системы, определены три актёра: администратор, сервис-инженер, оператор. Администратор может добавлять, редактировать и удалять профили пользователей, а также просматривать полный список пользователей. Сервис-инженер имеет возможность просматривать свой профиль, а также задавать параметры для регулирования величин объекта управления. Оператор контролирует параметры, которые система должна поддерживать, включать/выключать насосы, открывать/закрывать задвижки, а также контролирует состояние каждого элемента системы и при возникновении аварийной ситуации, обязан зафиксировать тревогу и вызвать специалистов для её устранения.

Опираясь на методику (листинг 1), представленную выше, следующим шагом будет составление диаграммы классов. Данная диаграмма состоит из 12 классов, основными являются *PuVNS*, *ValveVNS*, родители которых – *Pu* и *Valve* со свойствами: значение давления и состояние, для *Pu* (насос), характерен параметр частоты (рисунок 3).

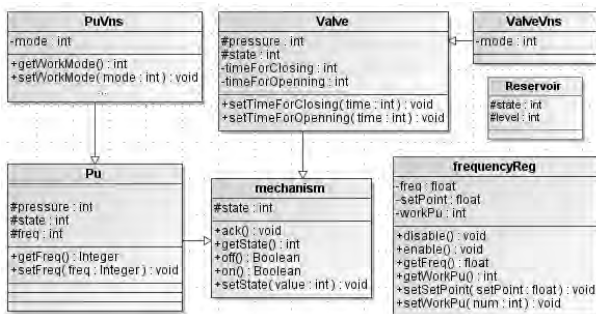


Рис. 3. Основная часть диаграммы классов АСУ ТП насосной станции

Методы этих классов нацелены на открытие и закрытие задвижек, а также на поддержания частоты работы двигателя насоса, соответственно. Остальные классы служат для снятия и/или поддержания параметров периферийных устройств (поддержание частоты работы двигателя – *frequencyReg*, снятие значений уровня наполнения резервуара – *Resrvoir*, взаимосвязи элементов при помощи класса *Periphery* и т.д.). Необходимость в детальной проработке отдельных классов отсутствует, поэтому после составления диаграммы классов следует этап построения диаграммы объектов.

Так как вся система содержит огромное количество элементов, а следовательно и объектов, приведём по одному экземпляру каждой группы схожих объектов. А именно объекты: *WPump1* – насос №1, находящийся на скважинах, *PSPump1* – насос №1, находящийся на водонапорной станции, *VNSPump1* – насос №1, находящийся на локальной водонапорной станции, *WValve1* – задвижка №1, находящаяся на скважинах и т.д. Свойства одного объекта в каждой группе объектов идентичны свойствам других объектов в группе. Следующим этапом после построения диаграммы объектов, является проектирование диаграммы деятельности. Спроектируем и детально рассмотрим диаграмму деятельности локальной водонапорной станции (рисунок 4).

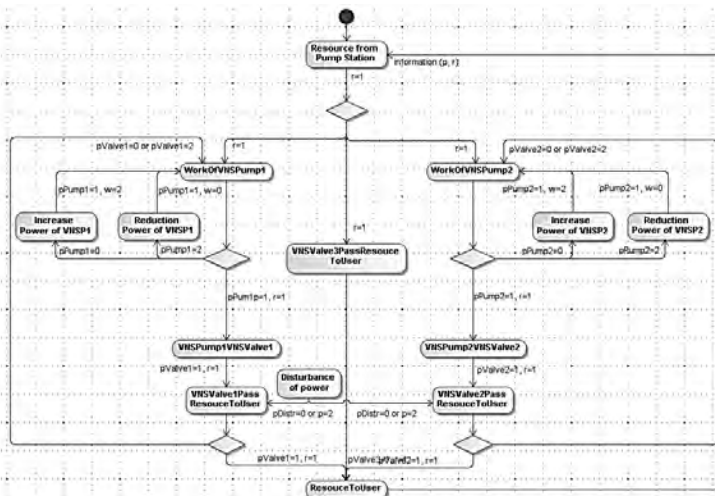


Рис. 4. Диаграмма деятельности АСУ ТП локальной насосной станции

Как говорилось ранее, логика функционирования локальной водонапорной станции заключается в работе одного из насосов в результате чередования смены их работы или попадания одного из насосов в аварийный режим. При необходимости (колебание давления)



или влияния каких-либо возмущающих факторов (*Disturbance of power*), происходит корректировка частоты работы двигателя насоса до нормализации давления в трубопроводе.

По завершению моделирования диаграммы деятельности переходим к трансляции полученной диаграммы в эквивалентную сеть Петри, с целью проверки корректности её построения. Преобразование диаграммы деятельности осуществляем, опираясь на правила [2]. Полученная сеть Петри (рисунки 5, 6) состоит из девяти основных мест:

$$P = \{FromPS, VNSPump1, VNSPump2, VNSValve1, VNSValve2, VNSValve3, RtoUser1, R1toUser1, R2toUser1, User1, Distrubance\},$$

16 переходов:

$$T = \{PS\ to\ VNSP1, PS\ to\ VNSP2, PS\ to\ VNSV3, VNSP1\ to\ VNSV1, VNSP2\ to\ VNS2, VNS1\ to\ U, VNS2\ to\ U, VNS3\ to\ U, Controller\ P1, Controller\ P2, P1R\ to\ User1, R\ to\ User1, P2\ R\ to\ User1, Power\ below\ of\ VNSP1, Power\ above\ of\ VNSP1, Power\ below\ of\ VNSP2, Power\ above\ of\ VNSP2\}$$

и имеет начальную маркировку  $m_i = (1, 1, 1, 1, 1, 1, 1, 1, 2)$ .

Построение сети Петри осуществляется при помощи правил, а также используются способы проектирования сетей Петри, представленные в [3]. После проведения анализа получены результаты, которые помогли скорректировать сеть и тем самым избежать ложных состояний, которые могли навредить работе системы. Например, при включении и выключении задвижек оператором, данные элементы получали текущие состояния до выбранного действия. Анализ показал, что состояния задвижек должны меняться. После корректировки ошибок работа системы начала выполняться корректно.

Анализ спроектированных сетей Петри осуществляется в программной среде CPN Tools (version 3.4.0) [11], позволяющей реализовать его в автоматическом режиме посредством исследования пространства состояний. Для понимания более общей картины пространства состояний отдельных элементов, так и всей системы, проведен анализ для различного числа локальных водонапорных станций (таблица 1). Получена зависимость достижения выбранного количества состояний во всем пространстве за промежуток времени  $t$ .

Таблица 1. Зависимость времени ( $t$ , сек) достижения выбранного количества состояний ( $m$ ) для  $n$  локальных водонапорных станций

$n$	$m = 10\ 000$	$m = 25\ 000$	$m = 50\ 000$	$m = 100\ 000$
1	22	178	908	6310
2	18	193	1156	6234
3	9	79	629	28965

Таким образом, можно предположить, что количество состояний всей системы, состоящей из десяти локальных водонапорных станций, превысит величину порядка  $10^6$ .

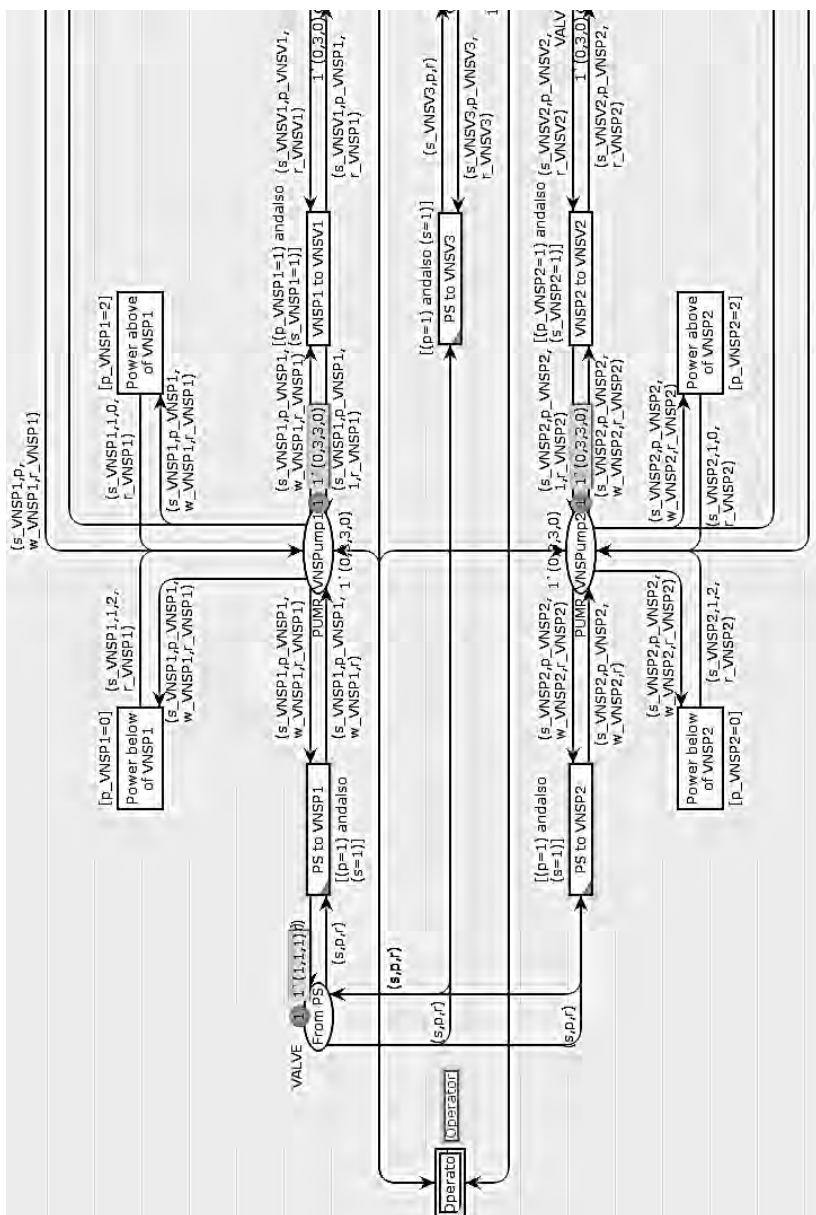


Рис. 5. Сеть Петри АСУ ТП локальной насосной станции (часть 1)

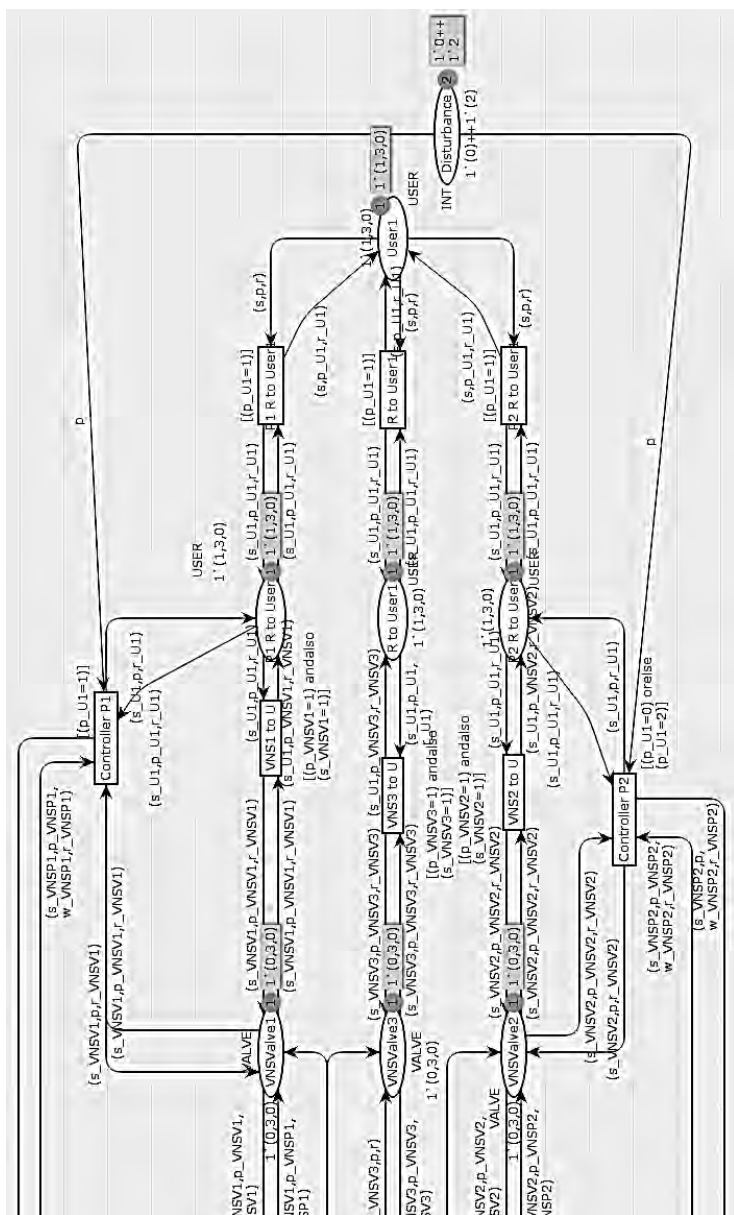


Рис. 6. Сеть Петри ACU TPI локальной насосной станции (часть 2)

**4. Заключение.** Одним из распространенных инструментов создания ПО является графический язык UML, который совместно с аппаратом сетей Петри, позволяет разработчикам проектировать системы за более короткий срок и с возможностью уменьшения соответствующих ошибок. В предлагаемой методике согласованного использования существующих подходов к автоматизации проектирования ПО разрабатываются диаграммы, отвечающие за статические свойства системы, а также поведенческие диаграммы, которые анализируются при помощи сетей Петри. Предварительно поведенческие диаграммы и, в частности, диаграмма деятельности, транслируются в сеть Петри согласно правилам преобразования [2].

Конструктивность данной методики продемонстрирована на примере системы АСУ ТП водонапорной станции, для которой смоделированы диаграмма прецедентов, диаграмма классов, диаграмма объектов и диаграмма деятельности. Диаграмма деятельности транслировалась в соответствующую ей сеть Петри для последующего анализа. Анализ сети Петри показал, что состояния задвижек должны меняться при включении и выключении их оператором, а не оставаться прежними, как было изначально. Других ошибок обнаружено не было, следовательно, можно приступить к генерации программного кода из UML диаграмм в автоматическом режиме.

Таким образом, продемонстрирована возможность применения предлагаемой методики на примере проектирования работы АСУ ТП водонапорной станции.

### Литература

1. Романников Д.О. Разработка программного обеспечения с применением UML диаграмм и сетей Петри для систем управления локальным оборудованием дисс. канд. техн. наук // Новосибирск: 2012. 195 с.
2. Коротиков С.В. Применение сетей Петри в разработке программного обеспечения центров дистанционного контроля и управления дисс. канд. техн. наук // Новосибирск: 2007. 216 с.
3. Марков А.В., Воевода А.А. Развитие системы “Перемещение манипулятора в пространстве с препятствиями” при помощи рекурсивных функций // Автоматика и программная инженерия. 2013. № 2. С. 35–41.
4. Zhu L.-Z., Kong F.-Sh. Automatic Conversion from UML to CPN for Software Performance Evaluation // Procedia Engineering. 2012. no. 29. pp. 2682–2686.
5. Thierry-Mieg Y., Hillah L.-M. UML behavioral consistency checking using instantiable Petri nets // In: ISSEng. 2008. no. 4. pp. 293–300.
6. Yang N., Yu H., Sun H., Qian Z. Modeling UML sequence diagrams using extended Petri nets // Telecommun Syst. 2012. Vol. 51. pp. 147–158.
7. Basile F., Chiacchio P., Grosso D.D. A two-stage modelling architecture for distributed control of real-time industrial systems: Application of UML and Petri Net // Computer Standards & Interfaces. 2009. vol. 31. pp. 528–538.
8. Романников Д.О., Марков А.В. Пример применения методики разработки ПО с использованием UML-диаграмм и сетей Петри // Научный вестник НГТУ. 2012. № 1(67). С. 175–181.
9. Марков А.В., Романников Д.О. Алгоритм автоматической трансляции диаграммы активности в сеть Петри // Доклады АН ВШ РФ. 2014. № 1 (22). С. 104–112.
10. Воевода А.А., Марков А.В. Инверсия простой ординарной сети Петри // Научный вестник НГТУ. 2013. № 4 (53). С. 215–218.
11. CPN Tools // URL: <http://cpntools.org/download>.

## References

1. Romannikov D.O. *Razrabotka programmnogo obespecheniya s primeneniem UML diagramm i setei Petri dlja sistem upravleniya lokal'nym oborudovaniem diss. kand. tehn. nauk* [Software development using UML diagrams and Petri nets for local control systems equipment Ph.D. thesis]. Novosibirsk: 2012. 195 p. (In Russ.).
2. Korotikov S.V. *Primenenie setei Petri v razrabotke programmnogo obespecheniya centrov distancionnogo kontrolja i upravleniya diss. kand. tehn. nauk* [Application of Petri nets in software development centers, remote monitoring and control Ph.D. thesis]. Novosibirsk: 2007. 216 p. (In Russ.).
3. Markov A.V., Voevoda A.A. [Development of the system "Moving the manipulator in space with obstacles" using recursive functions]. *Avtomatika i programmnaja inzhenerija – Automation and Software Engineering*. 2013. no. 2. pp. 35–41. (In Russ.).
4. Zhu L.-Z., Kong F.-Sh. Automatic Conversion from UML to CPN for Software Performance Evaluation. *Procedia Engineering*. 2012. no. 29. pp. 2682–2686.
5. Thierry-Mieg Y., Hillah L.-M. UML behavioral consistency checking using instantiable Petri nets. *Innovations Syst Softw Eng*. 2008. no. 4. pp. 293–300.
6. Yang N., Yu H., Sun H., Qian Z. Modeling UML sequence diagrams using extended Petri nets. *Telecommun Syst*. 2012. vol. 51. pp. 147–158.
7. Basile F., Chiacchio P., Grosso D.D. A two-stage modelling architecture for distributed control of real-time industrial systems: Application of UML and Petri Net. *Computer Standards & Interfaces*. 2009. vol. 31. pp. 528–538.
8. Romannikov D.O., Markov A.V. [Example application of the methodology of software development using UML diagrams and Petri nets]. *Nauchnyj vestnik NGTU – Science Bulletin of NSTU*. 2012. no. 1(67). pp. 175–181. (In Russ.).
9. Markov A.V., Romannikov D.O. [Algorithm for automatic translation of activity diagrams to Petri net]. *Doklady AN VSH RF – Reports AS GS RF*. 2014. no. 1 (22). pp. 104–112. (In Russ.).
10. Voevoda A.A., Markov A.V. [Inversion of ordinary Petri nets]. *Nauchnyj vestnik NGTU – Science Bulletin of NSTU*. 2013. no. 4 (53). pp. 215–218. (In Russ.).
11. CPN Tools. Available at: <http://cpntools.org/download>.

**Воевода Александр Александрович** — д-р техн. наук, профессор, кафедра автоматике, Новосибирский государственный технический университет. Область научных интересов: полиномиальный синтез, сети Петри, UML диаграммы. Число научных публикаций — более 200. [voevoda@ucit.ru](mailto:voevoda@ucit.ru); 630073, г. Новосибирск, пр. Карла Маркса 20.

**Voevoda Alexandr Aleksandrovich** — Ph.D., Dr. Sci., professor, department of Automation, Novosibirsk State Technical University. Research interests: polynomial synthesis, UML diagrams, Petri nets. The number of publication – more 200. [voevoda@ucit.ru](mailto:voevoda@ucit.ru); Karl Marx Avenue 20, Novosibirsk, 630073.

**Марков Александр Владимирович** — аспирант, кафедра автоматике, Новосибирский государственный технический университет. Область научных интересов: UML диаграммы, сети Петри, исследование пространства состояний. Число научных публикаций — более 20. [muviton3@gmail.com](mailto:muviton3@gmail.com); пр. Карла Маркса 20, г. Новосибирск, 630073; р.т. + 7 (952) 926-14-04.

**Markov Alexandr Vladimirovich** — Ph.D. student, department of Automation, Novosibirsk State Technical University. Research interests: UML diagrams, Petri nets, a study of state space. The number of publication — more 20. [muviton3@gmail.com](mailto:muviton3@gmail.com); Karl Marx Avenue 20, Novosibirsk, 630073; phone + 7 (952) 926-14-04.

**Романиков Дмитрий Олегович** — к-т техн. наук, кафедра автоматике, Новосибирский государственный технический университет. Область научных интересов: UML диаграммы, сети Петри, поиск программных ошибок. Число научных публикаций — более 25. [rom2006@gmail.com](mailto:rom2006@gmail.com); пр. Карла Маркса 20, г. Новосибирск, 630073; р.т. + 7 (961) 223-85-67.

**Romannikov Dmitry Olegovich** — Ph.D., NSTU, department of Automation. Research interests: UML diagrams, Petri nets, search software errors. The number of publication — more 25. [rom2006@gmail.com](mailto:rom2006@gmail.com); Karl Marx Avenue 20, Novosibirsk, 630073; phone + 7 (961) 223-85-67.

## РЕФЕРАТ

*Воевода А.А., Марков А.В., Романников Д.О.* **Разработка программного обеспечения: проектирование с использованием UML диаграмм и сетей Петри на примере АСУ ТП водонапорной станции.**

Рассматриваются и анализируются известные способы и методики разработки программного обеспечения (ПО) для этапа проектирования. Описывается методика, в которой используются UML диаграммы для моделирования свойств и сценариев функционирования системы, а также сети Петри для анализа динамических свойств спроектированных поведенческих диаграмм. Приводится усовершенствованная методика взаимного использования UML диаграмм и сетей Петри. Предлагаемая методика используется при проектировании ПО для АСУ ТП насосной станции, а именно, проектируется диаграмма прецедентов, диаграмм классов и диаграмм объектов, а также диаграмма деятельности, преобразование которой происходило по формальным правилам.

При анализе сети Петри, транслированной из диаграммы деятельности, были выявлены некорректные состояния, в которых оказывалась система при включении и выключении насосов оператором. После устранения данной ситуации, анализ дал положительный результат. Во время анализа получено представление о дереве достижимости системы, составляющее величину порядка  $10^6$ . Таким образом, работоспособность предлагаемой методики продемонстрирована на примере системы АСУ ТП насосной станции.

## SUMMARY

*Voevoda A., Markov A., Romannikov D.* **Software development: software design using UML diagrams and Petri nets for example automated process control system of pumping station.**

Methods and techniques of software design as one of the important stages of software development are described in the paper.

The method of software design with using of UML with Petri nets for analyzing of dynamic properties of set UML diagrams is described. Authors offer improved method of using of integration of UML diagrams and Petri nets. The offered method was used for designing of software of automated process control system (APCS) of pumping station: designing of use case, class, object diagrams and sequences diagram that was transformed to Petri net with help of formal rules.

Some incorrect states that occurred after pumps enabling/disabling by operator were identified by analysis of Petri net. Reachability tree of the system was gotten by analysis of the Petri net (the value of the tree is about  $10^6$  of nodes). Testing of offered system was showed on example of APCS of pump station.