

УДК 519.876.5:004.94

І.В. Стеценко

## Дослідження дискретно-подійних систем з використанням технології Петрі-об'єктного моделювання

Предложена технология использования программного обеспечения Петри-объектного моделирования систем для исследования дискретно-событийных систем. Технология основана на формализованном описании динамики функционирования элементов системы стохастической сетью Петри с многоканальными переходами. Программное обеспечение представлено *java*-библиотекой классов.

The technology using the software of the Petri-object modeling systems for the study of the event-discrete systems. The technology is based on a formalized description of the dynamics of the elements of the system stochastic Petri nets with the multi-channel transitions. The software is presented with a *java*-class library.

Запропоновано технологію використання програмного забезпечення з Петрі-об'єктного моделювання систем для дослідження дискретно-подійних систем. Технологія ґрунтується на формалізованому описі динаміки функціонування елементів системи стохастичною мережею Петрі з багатоканальними переходами. Програмне забезпечення подано *java*-бібліотекою класів.

**Вступ.** Сучасні технічні, технологічні та інформаційні системи мають високий рівень складності, а саме велику кількість взаємопов'язаних елементів, різноманітні процеси функціонування елементів та підсистем. Методи дискретно-подійного моделювання найбільш пристосовані для опису таких систем. В залежності від опису моделі розрізняють аналітичне та імітаційне моделювання. Аналітичні моделі надають можливість здійснювати дослідження властивостей систем в залежності від параметрів системи в узагальненому вигляді. Імітаційні моделі, на відміну від аналітичних, здатні враховувати специфічні деталі функціонування елементів системи. Кожний метод має свої переваги моделювання, тому важливо мати таку формалізацію системи, яка б дозволяла застосовувати обидва підходи.

Розроблена технологія Петрі-об'єктного моделювання орієнтована на використання стохастичної мережі Петрі з багатоканальними та конфліктними переходами, з інформаційними зв'язками для опису динаміки системи. Математичні рівняння станів такої мережі Петрі розроблені в роботі [1]. Алгоритм імітації Петрі-об'

ектної моделі, побудований у відповідності до математичного опису, підтверджує його коректність, дозволяє оцінити складність алгоритму та запобігти помилкам при його програмуванні [2]. Отже, технологія Петрі-об'єктного моделювання надає можливість поєднати переваги аналітичного та імітаційного моделювання систем, а також розширити клас систем, що можуть бути формалізовані стохастичною мережею Петрі. Використання технології з візуальними засобами представлення моделі дозволяє перейти від програмування алгоритму до програмування моделі.

### Постановка задачі та аналіз літературних джерел

Формалізацію системи з використанням специфікації *Discrete Event System Specification* [3] орієнтовано на деталізований опис усіх можливих станів елементів системи та визначення правил переходів з одного стану в інший. Інтеграція елементів відбувається завдяки опису множин вхідних та вихідних подій. Формалізація такого типу є узагальненням систем масового обслуговування і використовується у більшості програмних засобів з імітаційного моделювання (*Arena*, *ExtendSim*, *Plant Simulation*) у вигляді колекцій блоків, налаштованих на виконання тих або інших функцій – очікування, обробка, комплекта-

**Ключові слова:** імітаційне моделювання, стохастична мережа Петрі, об'єктно-орієнтований підхід, візуальне програмування.

ція, транспортування та ін. [4]. Проте програмування елементів управління в таких системах є надскладним, оскільки алгоритми, що задають функціонування блоків, є закритими.

Стохастичні мережі Петрі мають широкі можливості для представлення систем, оскільки можуть бути використані як для опису паралельних процесів, так і для опису процесів управління [5]. Але через велику кількість елементів мережі Петрі, використовуваних навіть для простих систем, застосування їх для складних систем потребує зусиль, не виправданих отриманими результатами моделювання. Широко відомий програмний засіб *CPNTools* [6] орієнтований на формалізацію системи розфарбованою мережею Петрі. Перевагою саме такої модифікації мережі Петрі є зменшення кількості елементів, необхідних для представлення складної системи. Проте використання розфарбованої мережі Петрі значно ускладнює алгоритм імітації та унеможливує аналітичне дослідження властивостей системи.

Поєднання об'єктно-орієнтованого підходу та мереж Петрі описано в роботі [7], де елементам мережі Петрі надано рис об'єктно-орієнтованого підходу, наприклад, маркери можуть бути об'єктами, фрагменти мережі Петрі – переходами. Такий підхід не виправдав очікувань, оскільки не дозволяє використовувати усі переваги об'єктно-орієнтованого підходу – створення примірників класів із заданими властивостями та успадкування класів.

Основною проблемою в сучасних дослідженнях дискретно-подійних систем є відсутність уніфікованого засобу формалізації, досить зручного як для програмування алгоритму імітації, так і для розробки візуальних засобів програмування моделі для складної і простої системи. Через відсутність зручних засобів формалізації на побудову алгоритмів імітації витрачаються зусилля, не виправдані результатами моделювання. Оскільки моделі дискретно-подійних систем є складовими сучасних інформаційних систем управління та систем прийняття рішень, то розробка нових технологій моделювання, що розширюють клас модельованих систем, є актуальною.

Метою даного дослідження є розробка компонентів візуального моделювання дискретно-подійних систем, які збільшують швидкість конструювання моделі та її налагодження. Поставлено задачу удосконалення програмного забезпечення з Петрі-об'єктного моделювання – нового формалізму дискретно-подійних систем, теоретичні засади якого розроблені в [8].

### **Технологія візуального моделювання динаміки об'єктів**

У технології Петрі-об'єктного моделювання елементам об'єктно-орієнтованого підходу надаються риси мережі Петрі, а саме, поле класу містить опис динаміки елементів цього класу у вигляді стохастичної мережі Петрі.

Процес розробки моделі відбувається у такий спосіб. Виділяються елементи моделі та розробляються структурні зв'язки між ними. Проектування динаміки елементів моделі здійснюється на основі стохастичної мережі Петрі з багатоканальними і конфліктними переходами та інформаційними зв'язками [9]. На першому кроці процес функціонування об'єкта поділяється на події та визначаються умови, за яких вони відбуваються. Події представляються переходами, а умови – позиціями мережі Петрі. Маркери у позиціях характеризують виконання умови. Наявність кількох маркерів у позиціях означає багатократне виконання умови. Для кожного переходу задається часова затримка, за якої здійснюється подія. Часова затримка може бути задана детермінованою або стохастичною величиною. Для миттєвої події задається детерміноване нульове значення. Інформаційні зв'язки дозволяють перевірити наявність умови без змінювання її при виконанні події, що розширює можливості мережі Петрі для представлення систем з елементами управління. Зауважимо, що при розробці динаміки об'єкта відтворюється його функціонування з урахуванням інформації, яка надходить від інших об'єктів.

Алгоритм імітації стохастичної мережі Петрі з багатоканальними та конфліктними переходами і інформаційними зв'язками реалізовано з просуванням часу до найближчої події та просуванням моделі в часі за подійним під-

ходом [4]. Програмне забезпечення розроблено у вигляді *Java*-бібліотек *PetriObj* та *PetriObjPaint* [10]. Перша бібліотека містить класи для розробки Петрі-об'єктної моделі, друга – для її графічного представлення. Клас *Petri Nets Frame* бібліотеки *PetriObjPaint* призначений для візуальної розробки стохастичної мережі Петрі та збереження її у кількох форматах: у графічному вигляді для перегляду та подальшої модифікації; у вигляді об'єкта класу *PetriNet* для безпосереднього використання при розробці Петрі-об'єктів; у вигляді методу класу *Net Library* для створення *Petri Net*-об'єкта з заданими параметрами.

Технологія розробки динаміки об'єкта передбачає такі кроки (рис. 1):

- створення мережі Петрі з використанням модуля візуальної розробки;
- тестування та корегування розробленої мережі Петрі з використанням модуля імітаційного моделювання;
- генерування програмного коду для створення мережі Петрі та його корегування з метою визначення параметрів;
- збереження мережі Петрі.

Зауважимо, що збереження мережі Петрі як методу для її створення, надає потрібну гнучкість для створення елементів моделі з заданими параметрами та, водночас, забезпечує найменші

витрати обчислювальних ресурсів для збереження динаміки об'єктів. Наприклад, для збереження мереж Петрі 10 об'єктів у графічному вигляді потрібно в середньому 40 Кб, а для збереження цієї ж мережі як методу потрібно тільки 2 Кб. Звісно, що для великих систем цей ефект є суттєвим.

### Конструювання моделі з об'єктів

Розробка моделі системи виконується у такій послідовності:

- проектування об'єктно-орієнтованої структури моделі;
- розробка множини елементів моделі;
- проектування та розробка зв'язків між елементами моделі.

Динаміка системи забезпечується динамікою її елементів. Передбачувані зв'язки між елементами підтверджують, що динаміка всієї системи представляється мережею Петрі, що доведено в роботі [8]. На діаграмі зв'язків показано спільні позиції Петрі-об'єктів та зв'язки, вздовж яких передаються маркери з переходу одного Петрі-об'єкта в інші. На рис. 2 подано приклад проектування зв'язків моделі «Технологічна автомат-машина» [11]. Петрі-об'єкти класу *Resource* поєднуються за допомогою спільних позицій *P1* і *P2*. У програмі таким зв'язком можуть бути поєднані будь-скільки Петрі-об'єктів. Поєднання Петрі-об'єкта *Resource* з Петрі-об'єктом *Generator* здійснюється за допомогою спільної позиції *P1*. Спільна для всіх Петрі-об'єктів позиція *Permission* здійснює передачу інформації про стан одного Петрі-об'єкта іншому і виконує функцію управління. У даному випадку ця позиція слугує для одночасного включення/виключення усіх елементів машини при поломці одного з них.

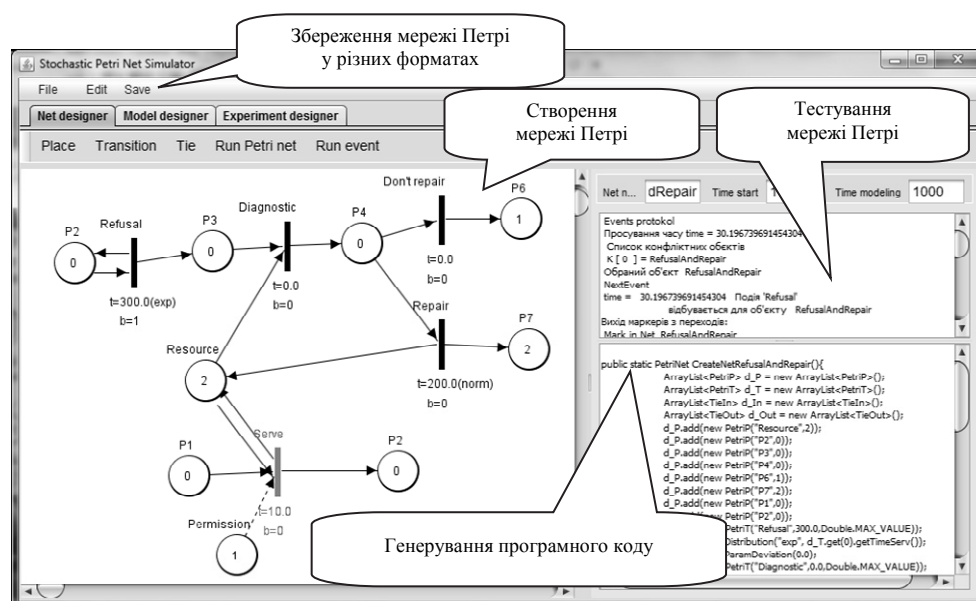


Рис. 1

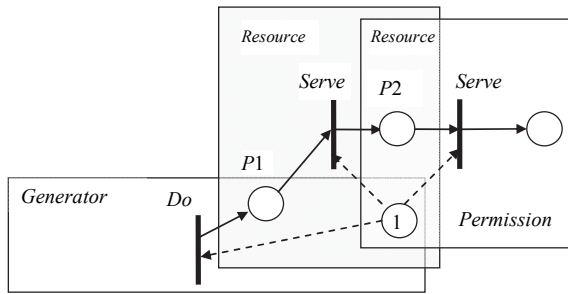


Рис. 2

Програмна реалізація моделі складається з таких кроків:

- розробки Петрі-об'єктів;
- розробки зв'язків Петрі-об'єктів;
- формування списку Петрі-об'єктів;
- запуску моделі на заданому часовому проміжку.

Петрі-об'єкти створюються як примірники класу *PetriSim* бібліотеки *PetriObj*. Класи Петрі-об'єктів можуть створюватись як класи-нащадки класу *PetriSim*. Усі розроблені Петрі-об'єкти додаються у список Петрі-об'єктів. Петрі-об'єктна модель створюється конструктором класу *PetriObjModel* з параметром, в якому передається список Петрі-об'єктів (рис. 3). Запуск моделі здійснюється методом *voidGo* (*doubletime*) класу *PetriObjModel*.

### Методи дослідження моделі

Аналітичне дослідження базової мережі Петрі дозволяє дослідити загальні властивості си-

стеми, такі, як *K*-обмеженість та активність [12]. Матричні рівняння станів стохастичної мережі [1] дозволяють використовувати для дослідження загальних властивостей мережі Петрі методи, які використовуються для базової мережі.

$$\mathbf{M}(t_n) - \mathbf{M}(t_0) + \mathbf{a}^+ \cdot (\mathbf{v}(t_n) - \mathbf{v}(t_0)) = (\mathbf{a}^+ - \mathbf{a}^-) \cdot \gamma(t_n), \quad (1)$$

де  $t_0, t_n$  – початковий та кінцевий моменти часу,  $\mathbf{M}(t)$  – вектор маркування в момент часу  $t$ ,  $\mathbf{v}(t)$  – вектор кількості активних каналів переходів,  $\gamma(t_n)$  – вектор кількості входів у переходи за період часу  $[t_0, t]$ ,  $\mathbf{a}^-$  – матриця входів,  $\mathbf{a}^+$  – матриця виходів мережі Петрі.

Оскільки використання цих методів передбачає матричний опис мережі Петрі, то при великій кількості елементів мережі таке дослідження надто складне. У зв'язку з цим перехід від дослідження властивостей системи до дослідження властивостей елементів моделі є важливим. Розбиття мережі Петрі на функціональні підмережі запропоновано у [13]. При використанні Петрі-об'єктного підходу, навпаки, мережа Петрі, яка описує динаміку системи, складається з мереж Петрі, які описують динаміку її елементів. Оскільки кількість переходів та позицій мережі Петрі, що описують окремих Петрі-об'єкт є незначною, то прове-

діння дослідження властивостей динаміки Петрі-об'єкта є нескладною задачею. Властивість обмеженості системи, наприклад, є наслідком обмеженості всіх її елементів. Дослідження інших властивостей системи є нерозв'язною задачею.

Імітаційне моделювання дискретно-подійної моделі виконується з метою оцінювання вихідних значень параметрів при заданих значеннях вхідних, визначення впливу факторів на вихідні величини, при оптимізації параметрів за

```
private static PetriObjModel constructModel(double timeModeling, double timeGen, double t1, double t2, double t3, double t4) {
    ArrayList<PetriSim> list = new ArrayList<>();

    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("FirstOperation", 1, 1, 36.0, t1, 900.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("SecondOperation", 1, 1, 36.0, t2, 750.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("ThirdOperation", 1, 1, 36.0, t3, 800.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("FourthOperation", 1, 1, 36.0, t4, 600.0), timeModeling));

    list.add(new PetriSim(NetLibrary.CreateNetGenerator(timeGen), timeModeling));
    list.get(0).getNet().getListP()[6] = list.get(list.size() - 1).getNet().getListP()[4]; //ніч"єднємо генератор

    list.get(1).getNet().getListP()[6] = list.get(0).getNet().getListP()[1];
    list.get(2).getNet().getListP()[6] = list.get(1).getNet().getListP()[7]; //є"єднємо
    list.get(3).getNet().getListP()[6] = list.get(2).getNet().getListP()[7]; //є"єднємо

    list.get(1).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //єднїє позиція Permission
    list.get(2).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //єднїє позиція Permission
    list.get(3).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //єднїє позиція Permission
    list.get(4).getNet().getListP()[2] = list.get(0).getNet().getListP()[8]; //єднїє позиція Permission для генератора

    PetriObjModel model = new PetriObjModel(list);

    return model;
}
```

Рис. 3

результатами експерименту. Програмне забезпечення містить модулі для представлення графічних залежностей з використанням бібліотеки *jFreeChart* [14] та для проведення оптимізації з використанням еволюційних методів.

**Висновки.** Технологія Петрі-об'єктного моделювання, на відміну від інших технологій моделювання дискретно-подійних систем, поєднує в собі переваги аналітичного та імітаційного методів, надає можливість візуального програмування динаміки складних систем, забезпечує коректність алгоритму імітації та його швидкодію для моделей з великою кількістю елементів. Модель дискретно-подійної системи формується на основі використання стохастичної мережі Петрі для опису динаміки структурних елементів системи та об'єктно-орієнтованого підходу для опису структури системи. Поєднання нового формалізму з відповідним програмним забезпеченням створює технологію Петрі-об'єктного моделювання систем, що дозволяє: складати формальний опис динаміки дискретно-подійної системи на основі динаміки її елементів; створювати класи типових елементів зі схожою динамікою; розробляти структуру системи на основі множини конструктивних елементів та взаємозв'язках між ними; використовувати уніфікований опис динаміки як елементів, так і об'єктів управління; здійснювати перехід до більш високого рівня програмування моделі: від програмування елементів моделі до конструкції моделі, від програмування подій до конструювання динаміки системи з використанням мереж Петрі.

Удосконалення програмного забезпечення компонентом візуального програмування збільшує швидкість конструювання моделі та її налагодження, зменшує кількість обчислювальних ресурсів, необхідних для зберігання.

Реалізація технології Петрі-об'єктного моделювання виконана у вигляді *java*-бібліотек.

1. *Stetsenko I.V.* State equations of stochastic timed Petri Nets with Informational relations // *Cybernetics and Systems Analysis*. – 2012. – **48**, N 5. – P. 784–797.
2. *Стеценко И.В.* Алгоритм имитации Петри-объектной модели // *Математичні машини і системи*. – 2012. – № 1. – С. 154–165.
3. *Zeigler B., Praehofer H., Tag Gon Kim.* Theory of Modeling and Simulation. – New York: Academic Press, 2000. – 510 p.
4. *Law A.M.* Simulation Modeling and Analysis. – New York: McGraw-Hill Int., 2014. – 804 p.
5. *Bause F., Kritzinger P.S.* Stochastic Petri Nets: An Introduction to the Theory. – Cape Town: Vieweg+Teubner, 2002. – 218 p.
6. *Aalst W., Stahl C.* Modeling Business Process – A Petri Net-Oriented Approach. – Cambridge: The MIT-Press, 2011. – 400 p.
7. *Lakos C.* Object Oriented Modeling with Object Petri Nets / Concurrent Object-Oriented Programming and Petri Nets, 2001. – P. 1–37.
8. *Стеценко И.В.* Теоретические основы Петри-объектного моделирования систем // *Математичні машини і системи*. – 2011. – № 4. – С. 136–148.
9. *Стеценко И.В.* Моделирование систем: Навч. посіб. – Черкаси: Маклаут, 2011. – 502 с.
10. *Стеценко И.В., Василевська О.В.* Проектирование графического модуля программного обеспечения Петри-объектного моделирования систем // *Вісн. Черкаського держ. технол. ун-ту*. – 2013. – № 2 – С. 13–18.
11. *Стеценко И.В., Титарчук А.А.* Моделирование надежности технологической машины // Там же. – 2014. – № 1. – С. 85–90.
12. *Murata T.* Petri Nets: Properties, Analysis and Applications // *Proc. of IEEE*. – 1989. – **77**, N 4. – P. 541–580.
13. *Зайцев Д.А.* Инварианты функциональных подсетей // *Тр. Одес. нац. акад. связи им. А.С. Попова*. – 2003. – № 4. – С. 57–63.
14. *JFree* (This site is owned and operated by David Gilbert and Thomas Morqner). – <http://www.jfree.org/jfreechart>

Поступила 15.09.2014  
Тел. для справок: +38 068 817-0742 (Черновцы)  
E-mail: [stiv66@yandex.ua](mailto:stiv66@yandex.ua)  
© И.В. Стеценко, 2014

## Исследование дискретно-событийных систем с использованием технологии Петри-объектного моделирования

**Введение.** Современные технические, технологические и информационные системы характеризуются высоким уровнем сложности, а именно множеством взаимосвязанных элементов, разнородностью процессов функционирования элементов и подсистем. Методы дискретно-событийного моделирования наиболее приспособлены для описания таких систем. В зависимости от описания модели различают аналитическое и имитационное моделирование. Аналитические модели дают возможность осуществлять исследование свойств систем в зависимости от параметров системы в наиболее общем виде. Имитационные модели, в отличие от аналитических, способны учитывать специфические детали функционирования элементов системы. Каждый метод моделирования имеет свои преимущества, поэтому необходима такая формализация системы, которая позволяла бы применять оба подхода.

Разработанная технология Петри-объектного моделирования ориентирована на использование стохастической сети Петри с многоканальными и конфликтными переходами, с информационными связями для описания динамики системы. Математические уравнения состояний такой сети Петри разработаны в [1]. Алгоритм имитации Петри-объектной модели, построенный в соответствии с математическим описанием, позволяет утверждать его корректность, дает возможность оценить сложность алгоритма и предотвратить ошибки при программировании алгоритма имитации [2]. Таким образом, технология Петри-объектного моделирования позволяет объединить преимущества аналитического и имитационного моделирования систем, а также расширить класс систем, формализуемых стохастической сетью Петри. Использование технологии вместе с визуальными средствами представления модели позволяет перейти от программирования алгоритма к программированию модели.

### Постановка задачи и анализ литературных источников

Формализация системы с использованием спецификации *Discrete Event System Specification* [3] ориентирована на детализированное описание всех возможных состояний элементов системы и определение правил переходов из одного состояния в другое. Интеграция элементов осуществляется благодаря описанию множеств входных и выходных событий. Формализация такого типа – это обобщение систем массового обслуживания и используется в большинстве программных средств по имитационному моделированию (*Arena, ExtendSim, PlantSimulation*) в виде коллекций блоков, настроенных на использование тех или иных функций – ожидания, обработки, комплектации, транспортировки и др. [4]. Но программирование эле-

ментов управления в таких системах очень сложно, поскольку алгоритмы, задающие функционирование блоков, закрыты.

Стохастические сети Петри имеют широкие возможности для представления систем, поскольку могут быть использованы как для описания параллельных процессов, так и для описания процессов управления [5]. Но из-за большого количества элементов сети Петри, используемых даже для простых систем, использование их для сложных систем требует усилий, которые не оправдываются полученными результатами моделирования. Широко известное программное средство *CPN Tools* [6] ориентировано на формализацию системы, раскрашенной сетью Петри. Преимущество такой модификации сети Петри – уменьшение количества элементов, необходимых для представления сложной системы. Однако использование раскрашенной сети Петри значительно усложняет алгоритм имитации и исключает аналитическое исследование свойств системы.

Объединение объектно-ориентированного подхода и сетей Петри описано в работе [7], где элементам сети Петри придаются черты объектно-ориентированного подхода, например, маркеры могут быть объектами, а фрагменты сети Петри – переходами. Такой подход не оправдал ожиданий, поскольку не позволяет использовать все преимущества объектно-ориентированного подхода – создание экземпляров классов с заданными свойствами и наследование классов.

Основная проблема в современных исследованиях дискретно-событийных систем – это отсутствие унифицированного средства формализации, достаточно удобного как для программирования алгоритма имитации, так и для разработки визуальных средств программирования модели для сложной и для простой систем. Из-за отсутствия удобных средств формализации на построение алгоритмов имитации тратятся усилия, которые не оправдываются полученными результатами моделирования. Поскольку модели дискретно-событийных систем есть составными частями современных информационных управляющих систем и систем принятия решений, то разработка новых технологий моделирования, расширяющих класс таких систем, – актуальная задача.

Цель данного исследования – разработка компонентов визуального моделирования дискретно-событийных систем, увеличивающих скорость конструирования модели и ее отладку. Поставлена задача усовершенствования программного обеспечения Петри-объектного моделирования – нового формализма дискретно-событийных систем, теоретические основы которого разработаны в [8].

## Технология визуального моделирования динамики объектов

В технологии Петри-объектного моделирования элементам объектно-ориентированного подхода придаются черты сети Петри, а именно, поле класса содержит описание динамики элементов этого класса в виде стохастической сети Петри.

Процесс разработки модели происходит так. Выделяются элементы модели, разрабатываются структурные связи между ними. Проектирование динамики элементов модели осуществляется на основе стохастической сети Петри с многоканальными, конфликтными переходами и информационными связями [9]. На первом шаге процесс функционирования объекта делится на события и определяются условия, при которых они осуществляются. События представляются переходами, а условия – позициями сети Петри. Маркеры в позициях характеризуют выполнение условий. Наличие нескольких маркеров в позициях означает многократное выполнение условий. Для каждого перехода задается временная задержка, при которой осуществляется событие. Временная задержка может быть задана детерминированной или стохастической величиной. Для мгновенного события задается детерминированное нулевое значение. Информационные связи позволяют проверить наличие условия без изменения его при осуществлении события, что расширяет возможности сети Петри для представления систем с элементами управления. Заметим, что при разработке динамики объекта воспроизводится функционирование объекта с учетом информации, поступающей от других объектов.

Алгоритм имитации стохастической сети Петри с многоканальными и конфликтными переходами, с информационными связями реализован с продвижением времени до ближайшего события и продвижением модели во времени событийным способом [4]. Программное обеспечение разработано в виде *Java*-библиотек *PetriObj* и *PetriObjPaint* [10]. Первая библиотека содержит классы для разработки Петри-объектной модели, другая – для графического представления. Класс *PetriNetsFrame* библиотеки *PetriObjPaint* предназначен для визуальной разработки стохастической сети Петри и сохранения ее в нескольких форматах: в графическом виде для просмотра и дальнейшей модификации; в виде объекта класса *PetriNet* для непосредственного использования при разработке Петри-объектов; в виде метода клас-

са *NetLibrary* для создания *PetriNet*-объекта с заданными параметрами.

Технология разработки динамики объекта предусматривает такие этапы (рис. 1):

- создание сети Петри с использованием модуля визуальной разработки;
- тестирование и корректирование разработанной сети Петри с использованием модуля имитационного моделирования;
- генерирование программного кода для создания сети Петри и его корректирование с целью определения параметров сети Петри;
- сохранение сети Петри.

Заметим, что сохранение сети Петри в виде метода для ее создания дает необходимую гибкость для создания элементов модели с заданными параметрами и, одновременно, обеспечивает наименьшие затраты вычислительных ресурсов для сохранения динамики объектов. Например, для сохранения сетей Петри 10 объектов, которые сохраняются в графическом виде, нужно в среднем 40 Кб, а для сохранения этой же сети как метода нужно только 2 Кб. Конечно, для больших систем этот эффект существен.

### Конструирование модели из объектов

Разработка модели системы осуществляется так:

- проектирование объектно-ориентированной структуры модели;
- разработка множества элементов модели;
- проектирование и разработка связей между элементами модели.

Динамика системы обеспечивается динамикой ее элементов. Предусмотренные связи между элементами подтверждают, что динамика всей системы представляется

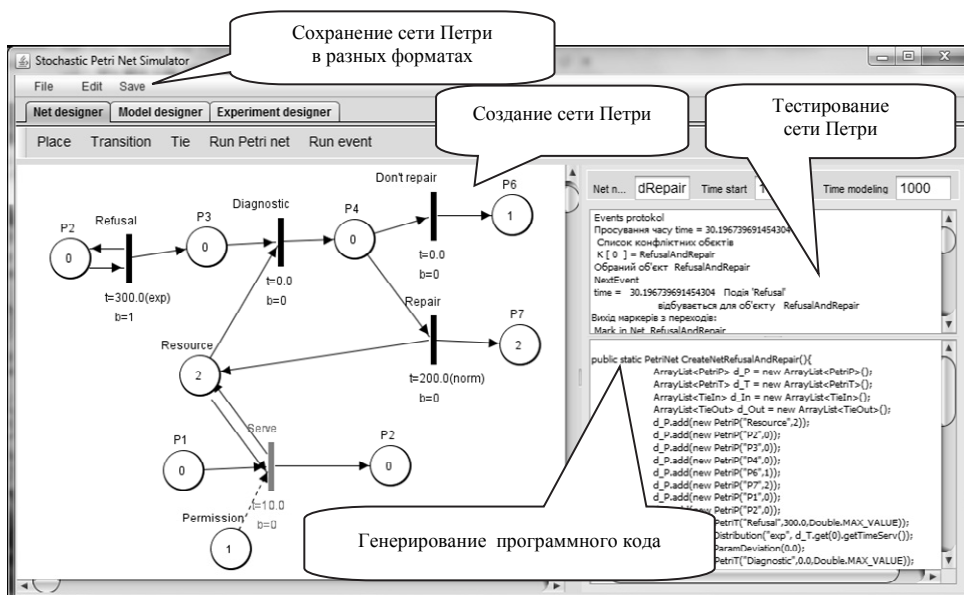


Рис. 1

сеть Петри, что доказано в работе [8]. На диаграмме связей показаны общие позиции Петри-объектов и связи, вдоль которых передаются маркеры из перехода одного Петри-объекта в другие. На рис. 2 приведен пример проектирования связей модели «Технологическая автомат-машина» [11]. Петри-объекты класса *Resource* объединяются общими позициями *P1* и *P2*. В программе такой связи могут быть объединены сколько угодно Петри-объектов. Объединение Петри-объекта *Resource* с Петри-объектом *Generator* осуществляется с помощью общей позиции *P1*. Общая для всех Петри-объектов позиция *Permission* осуществляет передачу информации о состоянии одного Петри-объекта другому и выполняет функцию управления. В данном случае эта позиция служит для одновременного включения/выключения всех элементов машины при поломке одного из них.

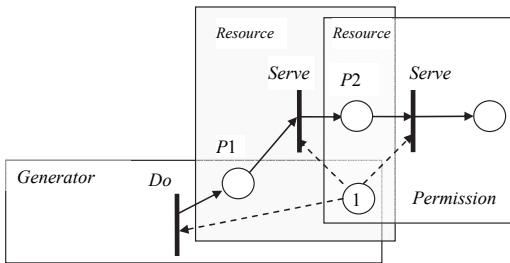


Рис. 2

Программная реализация модели состоит из таких этапов:

- разработки Петри-объектов;
- разработки связей Петри-объектов;
- формирования списка Петри-объектов;
- запуска модели на заданном временном интервале.

Петри-объекты создаются как экземпляры класса *PetriSim* библиотеки *PetriObj*. Классы Петри-объектов могут наследоваться как классы-наследники класса *PetriSim*. Все разработанные Петри-объекты добавляются в список Петри-объектов. Петри-объектная модель создается конструктором класса *PetriObjModel* с параметром, в котором передается список Петри-объектов (рис. 3). Запуск модели осуществляется методом *voidGo (doubletime)* классу *PetriObjModel*.

```
private static PetriObjModel constructModel(double timeModeling, double timeGen, double t1, double t2, double t3, double t4) {
    ArrayList<PetriSim> list = new ArrayList<>();

    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("FirstOperation ", 1, 1, 36.0, t1, 900.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("SecondOperation ", 1, 1, 36.0, t2, 750.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("ThirdOperation ", 1, 1, 36.0, t3, 800.0), timeModeling));
    list.add(new PetriSim(NetLibrary.CreateNetOperateAndRepair("FourthOperation ", 1, 1, 36.0, t4, 600.0), timeModeling));

    list.add(new PetriSim(NetLibrary.CreateNetGenerator(timeGen, timeModeling));
    list.get(0).getNet().getListP()[6] = list.get(list.size() - 1).getNet().getListP()[4]; //подключаем генератор

    list.get(1).getNet().getListP()[6] = list.get(0).getNet().getListP()[7]; //з"єдн"єм"є
    list.get(2).getNet().getListP()[6] = list.get(1).getNet().getListP()[7]; //з"єдн"єм"є
    list.get(3).getNet().getListP()[6] = list.get(2).getNet().getListP()[7]; //з"єдн"єм"є

    list.get(1).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //сп"єльна позиц"є Permission
    list.get(2).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //сп"єльна позиц"є Permission
    list.get(3).getNet().getListP()[8] = list.get(0).getNet().getListP()[8]; //сп"єльна позиц"є Permission
    list.get(4).getNet().getListP()[2] = list.get(0).getNet().getListP()[8]; //сп"єльна позиц"є Permission для генератора

    PetriObjModel model = new PetriObjModel(list);

    return model;
}
```

Рис. 3

## Методы исследования модели

Аналитическое исследование базовой сети Петри позволяет исследовать такие общие свойства системы, как *K*-ограниченность и активность [12]. Матричные уравнения состояний стохастической сети, полученные в работе [1], позволяют использовать для исследования общих свойств стохастической сети Петри методы, используемые для базовой сети Петри.

$$\mathbf{M}(t_n) - \mathbf{M}(t_0) + \mathbf{a}^+ \cdot (\mathbf{v}(t_n) - \mathbf{v}(t_0)) = (\mathbf{a}^+ - \mathbf{a}^-) \cdot \gamma(t_n), \quad (1)$$

где  $t_0, t_n$  – начальный и конечный моменты времени,  $\mathbf{M}(t)$  – вектор маркировки в момент времени  $t$ ,  $\mathbf{v}(t)$  – вектор количества активных каналов переходов,  $\gamma(t_n)$  – вектор количества входов в переходы за интервал времени  $[t_0, t]$ ,  $\mathbf{a}^-$  – матрица входов,  $\mathbf{a}^+$  – матрица выходов сети Петри.

Поскольку использование этих методов предусматривает матричное описание сети Петри, то при большом количестве элементов сети такое исследование становится слишком сложным. В связи с этим переход от исследования свойств системы к исследованию свойств элементов модели будет существенным. В работе [13] предлагается развитие сети Петри на функциональные подсети. При использовании Петри-объектного подхода, наоборот, сеть Петри, описывающая динамику системы, составляется из сетей Петри, описывающих динамику ее элементов. Поскольку количество переходов и позиций сети Петри, описывающих отдельный Петри-объект незначительно, то исследование свойств динамики Петри-объекта – несложная задача. Свойство ограниченности системы, например, следует из ограниченности всех ее



элементов. Исследование других свойств системы остается нерешенной задачей.

Имитационное моделирование дискретно-событийной модели осуществляется с целью оценивания исходных значений параметров при заданных значениях входных, определения влияния факторов на исходные величины, оптимизации параметров по результатам экспериментирования. Программное обеспечение содержит модули для представления графических зависимостей с использованием библиотеки *jFreeChart* [14] и для проведения оптимизации с использованием эволюционных методов.

**Заключение.** Технология Петри-объектного моделирования, в отличие от других технологий моделирования дискретно-событийных систем, объединяет в себе преимущества аналитического и имитационного методов моделирования, дает возможность визуального программирования динамики сложных систем, обеспечивает корректность алгоритма имитационного быстрого действия для моделей с большим количеством элементов. Модель дискретно-событийной системы составляется на основе использования стохастической сети Петри для описания динамики структурных элементов системы и объектно-ориентированного подхода для описания

структуры системы. Объединение нового формализма с соответствующим программным обеспечением создает технологию Петри-объектного моделирования систем, позволяющую: составлять формальное описание динамики дискретно-событийной системы на основе динамики ее элементов; создавать классы типовых элементов со схожей динамикой; разрабатывать структуру системы на основе множества конструктивных элементов и взаимосвязей между ними; использовать унифицированное описание динамики как элементов управления, так и объектов управления; осуществлять переход к более высокому уровню программирования модели: от программирования элементов модели к программированию конструкции модели, от программирования событий к конструированию динамики системы с использованием сети Петри.

Усовершенствование программного обеспечения компонентом визуального программирования увеличивает скорость конструирования модели и ее отладки, уменьшает количество вычислительных ресурсов, необходимых для ее сохранения.

Реализация технологии Петри-объектного моделирования выполнена в виде *java*-библиотек.



**Внимание !**

**Оформление подписки для желающих  
опубликовать статьи в нашем журнале обязательно.  
В розничную продажу журнал не поступает.  
Подписной индекс 71008**