

*Інна Стеценко, Катерина Лещенко*

## ІНТЕЛЕКТУАЛЬНИЙ КОМПОНЕНТ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ СТОХАСТИЧНИХ МЕРЕЖ ПЕТРІ

*Інна Стеценко, Катерина Лещенко*

## ИНТЕЛЛЕКТУАЛЬНЫЙ КОМПОНЕНТ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ СТОХАСТИЧЕСКИХ СЕТЕЙ ПЕТРИ

*Inna Stetsenko, Kateryna Leshchenko*

## STOCHASTIC PETRI NETS VISUAL PROGRAMMING INTELLECTUAL COMPONENT

*З розвитком інформаційних технологій зростає потреба в імітаційному моделюванні складних систем для відшукування оптимальних параметрів управління ними. У зв'язку з цим відбувається зростання вимог до засобів моделювання, зокрема постає задача забезпечення високої швидкості побудови та модифікації імітаційних моделей. Вирішувати цю задачу пропонується за допомогою технології Петрі-об'єктного моделювання, яка ґрунтується на формалізованому описі динаміки системи мережею Петрі.*

*Роботу присвячено розробленню інтелектуального компонента візуального програмування стохастичних мереж Петрі програмного забезпечення Петрі-об'єктного моделювання дискретно-подійних систем. Створено механізм збереження моделей, побудованих за допомогою маніпулювання графічними об'єктами, у вигляді програмного коду мовою Java, а також відновлення візуальних моделей з програмного коду з автоматичним розташуванням елементів в мережі Петрі так, щоб максимізувати легкість сприйняття моделей та швидкість їх побудови.*

**Ключові слова:** візуальне програмування, дискретно-подійна система, стохастична мережа Петрі, імітаційне моделювання, Петрі-об'єктна модель.

*Рис.: 3. Бібл.: 8.*

*С развитием информационных технологий возрастает потребность в имитационном моделировании сложных систем для нахождения оптимальных параметров управления ими. В связи с этим возрастают требования к средствам моделирования, в том числе возникает задача обеспечения высокой скорости построения и модификации имитационных моделей. Решать эту задачу предлагается при помощи технологии Петри-объектного моделирования, основанной на формализованном описании динамики системы сетью Петри.*

*Работа посвящена разработке интеллектуального компонента визуального программирования стохастических сетей Петри программного обеспечения Петри-объектного моделирования систем. Создан механизм сохранения моделей, построенных путем манипулирования графическими объектами, в виде программного кода на языке Java, а также восстановления визуальных моделей из программного кода с автоматическим размещением элементов сети Петри так, чтобы максимизировать легкость восприятия моделей и скорость их построения.*

**Ключевые слова:** визуальное программирование, дискретно-событийная система, стохастическая сеть Петри, имитационное моделирование, Петри-объектная модель.

*Рис.: 3. Библ.: 8.*

*The development of information technology has caused an increase in the need for simulation modeling of complex systems intended to determine the optimal management parameters. As a result, there is a tendency for the requirements for modeling tools to increase, including the emergence of the problem of providing rapid construction and modification of simulation models. The suggested way to solve this problem is to use the Petri-object modeling technology which is based on the formalized description of the system's dynamics by the means of Petri nets.*

*This work is dedicated to the development of a stochastic Petri nets visual programming intellectual component of the Petri-object modeling software. A mechanism has been created to save graphically created models in the form of program code in Java and restore visual models from the program code with automatic placement of Petri nets' elements in a way that maximizes the ease of model perception and the speed of model construction.*

**Key words:** visual programming, discrete event system, stochastic Petri net, simulation modeling, Petri-object model.

*Fig.: 3. Bibl.: 8.*

**Постановка проблеми.** З розвитком інформаційних технологій зростає потреба в імітаційних моделях складних систем, що використовуються для відшукування оптимальних параметрів управління та прийняття рішень. Підвищення складності систем, поведінку яких необхідно моделювати, пов'язане з тенденцією до зростання кількості їх структурних елементів, а також кількості взаємозв'язків між ними. У зв'язку з цим відбувається зростання вимог до засобів моделювання, зокрема постає задача забезпечення високої швидкості побудови імітаційних моделей дискретно-подійних систем, а також представлення моделей у вигляді, що максимально полегшив би їхнє сприйняття користувачем. Крім того, набуває важливості швидкість модифікації моделей у зв'язку зі структурними та функціональними змінами. Все це призводить до актуальності роз-

робки такого програмного забезпечення, яке дозволило б імітувати поведінку складних дискретно-подійних систем, швидко будувати та модифікувати їх моделі за допомогою візуального програмування та яке б забезпечувало зручність представлення імітаційних моделей користувачу.

У багатьох наукових роботах відзначається ефективність використання формалізму мереж Петрі для опису та дослідження паралельних процесів та процесів управління. Застосування мереж Петрі для моделювання комунікаційних систем розглядається в роботі [1]. Приклади моделювання та аналізу продуктивності складних стохастичних систем мережею Петрі наведені в роботі [2]. Використання мережі Петрі для моделювання бізнес-процесів описано в роботі [3].

Побудова моделей засобами візуального програмування означає автоматичну генерацію програмного коду, що відповідає за створення моделі, на основі маніпуляцій користувача з графічними об'єктами, які представляють структурні елементи імітаційної моделі. Візуальне представлення моделі системи прискорює процес конструювання складної моделі, зменшує кількість помилок при введенні моделі, значно спрощує процес її корегування або модифікації. Проте збереження графічних образів моделей для складних систем може призвести до значних витрат ресурсів комп'ютерної пам'яті. У той же час зберігання програмного коду є в десятки разів менш витратним. Тому створення програмного компонента, який би здійснював перетворення графічного образу моделі у програмний код і навпаки, вирішує проблему досягнення компромісу між зручністю використання моделі та витратами ресурсів на її зберігання.

**Аналіз існуючих засобів програмування мереж Петрі.** Існує багато програмних засобів імітаційного моделювання на основі мереж Петрі. Більшість із них містить графічні редактори, деякі орієнтовані на використання специфічної мови опису мережі Петрі. В основному програмні засоби забезпечують можливість роботи з певним класом мереж Петрі – з базовими мережами, часовими мережами Петрі, стохастичними або розфарбованими. Йдеться про такі засоби, як Artifex, Cosmos, CPN Tools, ExSpect, F-net, INA, JFem, Mercury. Компонентами таких засобів є графічні редактори, аніматори руху маркерів у мережі, симулятори та, в деяких випадках, підтримка математичного аналізу структурних властивостей мережі Петрі.

**Виділення не вирішених раніше частин загальної проблеми.** Основною перешкодою для широкого застосування мереж Петрі є надто велика кількість елементів для побудови моделей навіть простих систем. Як наслідок, розроблення та відлагодження таких моделей є трудомісткою задачею.

Об'єктно-орієнтована технологія має певні переваги для розбудови великих програмних систем. Тому є цілком логічною поява формалізмів, що поєднують об'єктно-орієнтовану технологію та мережі Петрі. Намагання доповнити мережу Петрі аспектами об'єктно-орієнтованого підходу відтворені у програмному засобі LOOPN [4]. Проте найбільш важливі властивості об'єктів залишились не втіленими, а саме створення екземпляру класу з заданою динамікою та успадкування динаміки об'єкта.

Технологія Петрі-об'єктного моделювання ґрунтується на концепції, що модель системи конструюється з елементів, динаміка яких задана стохастичною мережею Петрі. Об'єкти із заданою динамікою тиражуються у заданій кількості і з заданими параметрами, можуть успадковувати динаміку від інших об'єктів та створювати нові об'єкти, динаміка яких пов'язана з іншими об'єктами. Доведено, що динаміка Петрі-об'єктної моделі в цілому теж описується стохастичною мережею. Цей факт є важливим, оскільки гарантує коректність та обчислюваність моделі [5]. Ефективність та поліноміальна обчислювальна складність алгоритму імітації Петрі-об'єктної моделі доведені в роботі [6].

Використання технології Петрі-об'єктного моделювання для розробки та дослідження імітаційних моделей великих систем, таких як система управління транспортним рухом, розподілена обчислювальна система, виробнича система, система управління навчальним процесом у вищому навчальному закладі розглядається в роботі [7]. Проте практичне використання технології для широкого кола дослідників потребує розробки програмного комплексу методів і засобів, що підтримують усі етапи розроблення моделі з використанням візуальних засобів.

**Мета статті.** Головною метою цієї роботи є вирішення задачі розроблення інтелектуального компонента візуального програмування стохастичних мереж Петрі на базі існуючого програмного засобу Петрі-об'єктного моделювання дискретно-подійних систем [8]. Цей компонент повинен підвищити швидкість побудови моделей дискретно-подійних систем у вигляді мереж Петрі і зручність їх створення та модифікації. Для досягнення поставленої мети необхідно розробити компонент з такими функціональними можливостями:

- зберігання мережі Петрі, побудованої за допомогою маніпулювання графічними об'єктами (позиціями, переходами, вхідними та вихідними дугами), у вигляді програмного коду мовою Java, а саме у вигляді статичного методу спеціально призначеного для цього класу;

- відновлення візуальної моделі дискретно-подійної системи з програмного коду методу;

- автоматичне розташування елементів мережі, що забезпечує легкість сприйняття моделі користувачем (мінімізує кількість перетинів між дугами та випадків, коли будь-які елементи мережі частково перекривають один одного).

**Програмне забезпечення Петрі-об'єктного моделювання.** У цій роботі розглянуто стохастичні мережі Петрі з часовими затримками, багатоканальними переходами та інформаційними зв'язками. Саме з такими мережами Петрі працює програмний засіб, для якого розробляється компонент візуального програмування (далі – програмний засіб). Слід зазначити, що при рівності всіх часових затримок нулю матимемо просту мережу, що не є часовою; при детермінованих затримках (деякі з яких ненульові) матимемо детерміновану часову мережу; при рівності кількості каналів кожного переходу одиниці матимемо мережу без багатоканальних переходів. Тобто ми розглядаємо один із найбільш узагальнених типів мереж Петрі, і програмний засіб, очевидно, дозволяє працювати з будь-якими простішими мережами.

Часові затримки являють собою затримки при переміщенні маркерів у мережі, причому вони можуть бути пов'язані як із переходами, так і з позиціями. Часові затримки в найпростішому випадку визначаються детермінованою або випадковою величиною. На сьогоднішній день розроблена теорія мереж Петрі з часовими затримками, визначеними детермінованими величинами, а теорія мереж Петрі з часовими затримками, визначеними випадковими величинами, знаходиться у процесі розробки [5]. Останні є стохастичними мережами Петрі, що й розглядаються у цій роботі.

Інформаційні зв'язки між переходом і позицією дозволяють моделювати такі події, які відбуваються за певної умови та під час здійснення яких умова не зникає. Наприклад, подія «Приймає заборгованості» об'єкта Викладач здійснюється за умови наявності заборгованостей з дисциплін викладача, що відповідає наявності принаймні одного маркера в позиції «Кількість заборгованостей з дисциплін, які веде викладач» (рис. 1). Подія «Приймає заборгованості» об'єкта Викладач ініціює події «Здає екзамен» об'єктів Студент, які мають заборгованості, і тільки за результатом здачі екзамену Студентом здійснюється зменшення кількості заборгованостей. Якщо використати в цьому випадку звичайний зв'язок, то ще до здійснення події «Здає екзамен» хоч одного об'єкта Студент кількість заборгованостей буде зменшена на одиницю об'єктом Викладач [7].

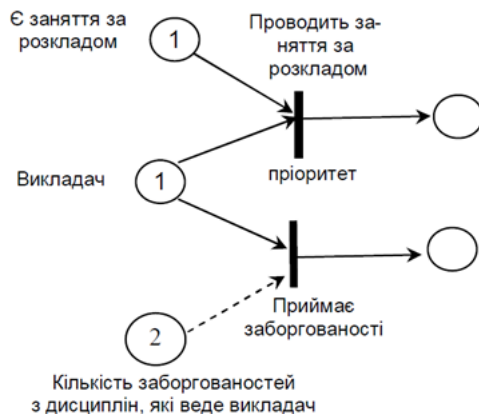


Рис. 1. Формалізація події «Приймає заборгованості» мережею Петрі

Мережі Петрі з багатоканальними переходами – це такі мережі, які допускають повторні запуски активних переходів, причому обмеження на кількість каналів можуть задаватися за допомогою маркування спеціальних позицій або вказуватися явно [7].

Компонент візуального програмування стохастичних мереж Петрі було розроблено на базі графічного модуля програмного засобу [8], що дозволяє будувати стохастичні мережі Петрі з багатоканальними переходами та інформаційними зв'язками шляхом маніпуляцій з графічними об'єктами (позиціями, переходами, вхідними та вихідними дугами), а також зберігати, відкривати, модифікувати збережені моделі, моделювати поведінку відповідних систем у часі, збирати статистичні дані щодо їхнього функціонування. Цей програмний засіб створено засобами мови програмування Java та інтегрованого середовища розробки Netbeans IDE.

До початку роботи над новим компонентом програмний засіб складався з п'яти пакетів:

1. PetriObj (плагін, що забезпечує користувача основними класами для реалізації Петрі-об'єктних моделей; це, наприклад, класи, які відповідають вхідним чи вихідним дугам мережі, позиціям, переходам, цілим мережам Петрі).

2. graphnet (бібліотека класів, яка розв'язує задачу надання графічним об'єктам, що є елементами мережі Петрі, функціональності, пов'язаної з їхнім існуванням у спільному візуальному просторі; містить класи, що відповідають графічним аналогам дуг, позицій, переходів тощо).

3. graphpresentation (бібліотека класів, що розв'язує задачу програмної розробки графічних об'єктів мережі Петрі та їх спільного взаємопов'язаного існування у візуальному просторі; ця бібліотека, зокрема, містить JFrame форми та клас для роботи з файлами під назвою FileUse, який дозволяє зберігати побудовані імітаційні моделі у форматі .pns та відкривати створені файли для продовження роботи з існуючими моделями).

4. graphreuse (бібліотека, що містить форму та інші класи, які забезпечують зручне редагування всіх параметрів мережі за допомогою декількох інтерактивних таблиць).

5. libnet (бібліотека для збереження методів побудови конкретних мереж Петрі).

**Проектування та розробка інтелектуального компонента візуального програмування мереж Петрі.** Проектування інтелектуального компонента візуального програмування мереж Петрі було розпочато з розділення його функціональності на дві частини: збереження мережі Петрі у вигляді методу в бібліотеці libnet та відновлення візуальної моделі з такого методу.

Для реалізації першої частини було додано два нових методи до описаного вище класу FileUse: saveNetAsMethod (генерація коду методу, що створює та повертає об'єкт типу PetriNet, який відповідає мережі Петрі у візуальному просторі програми, і виве-

## TECHNICAL SCIENCES AND TECHNOLOGIES

дення згенерованого коду на спеціальну панель) та `saveMethodInNetLibrary` (збереження коду з панелі у бібліотеці `libnet`). Крім того, було модифіковано клас `PetriNetsFrame` (створено елементи меню, що відповідають цим новим методам, і прив'язано обробку події їх натискання).

Для реалізації другої частини було проведено аналогічні зміни у класі `PetriNetsFrame` і додано нові методи до класу `FileUse`: `openMethod` (відновлення візуальної моделі системи з програмного коду обраного методу за допомогою регулярних виразів та виведення візуальної моделі у графічний простір), `convertMethodToPetriNet` (аналіз коду методу та генерація на його основі об'єкта класу `PetriNet`), `generateGraphNetBySimpleNet` (генерація об'єкта класу `GraphPetriNet` графічного аналогу мережі Петрі на базі об'єкта `PetriNet` із автоматичним розміщенням елементів мережі таким чином, щоб мінімізувати кількість перетинів між дугами та випадків, коли будь-які елементи мережі частково перекривають один одного). Крім того, було створено клас `MethodNameDialogPanel` – діалогову панель вибору методу для відкриття зі списку існуючих методів у бібліотеці `libnet`. Для реалізації методу `generateGraphNetBySimpleNet` було створено ще один новий клас – клас `VerticalSet`. Він є абстракцією, що використовується в алгоритмі візуалізації мережі, та являє собою набір елементів (позицій або переходів), які буде розміщено одним стовпцем, тобто один над одним. Кожен такий вертикальний набір може містити або лише позиції, або лише переходи.

Алгоритм, розроблений для візуалізації мережі Петрі у графічному просторі, полягає в наступному:

1. Побудувати список вертикальних наборів позицій та переходів:

1.1. Вилучити будь-який перехід зі списку наявних переходів та помістити його в перший вертикальний набір (тип набору: для переходів).

1.2. Доки є елементи в списку наявних позицій або в списку наявних переходів, повторювати:

1.2.1. Обнулити посилання на останній вертикальний набір та його індекс у списку вертикальних наборів.

1.2.2. Для кожного вертикального набору виконати:

1.2.2.1. Якщо він ще не позначений як «готовий», встановити на нього посилання на останній вертикальний набір, прирівняти індекс останнього набору до його позиції у списку вертикальних наборів (тобто вважати його останнім), вийти з циклу.

1.2.3. Якщо посилання на останній вертикальний набір порожнє, припинити роботу.

1.2.4. Якщо тип останнього набору – для позицій, виконати:

1.2.4.1. Побудувати список всіх переходів, що є вхідними до позицій останнього набору і є в списку наявних переходів.

1.2.4.2. Побудувати список всіх переходів, що є вихідними до позицій останнього набору, є в списку наявних переходів і не входять у список, побудований у п. 1.2.4.1.

1.2.4.3. Якщо список вхідних переходів не порожній та індекс останнього набору дорівнює нулю, додати на початок списку наборів новий порожній набір типу «для переходів», а також прирівняти індекс останнього вертикального набору до 1.

1.2.4.4. Якщо список вихідних переходів не порожній та індекс останнього набору на 1 менший за розмір списку наборів, додати в кінець списку наборів новий порожній набір типу «для переходів».

1.2.4.5. Для всіх вхідних переходів виконати:

1.2.4.5.1. Додати цей перехід до вертикального набору, що (за індексом) знаходиться перед останнім (на який збережено посилання, а не фізично останнім).

1.2.4.5.2. Позначити набір, до якого додали перехід, як «не готовий».

1.2.4.5.3. Вилучити перехід зі списку наявних переходів.

1.2.4.6. Для всіх вихідних переходів виконати:

1.2.4.6.1. Додати цей перехід до вертикального набору, що (за індексом) знаходиться після останнього.

1.2.4.6.2. Позначити набір, до якого додали перехід, як «не готовий».

1.2.4.6.3. Вилучити перехід зі списку наявних переходів.

1.2.5. Інакше виконати те ж саме, що у п. 1.2.4, тільки замінивши «переходи» на «позиції», а «позиції» на «переходи».

1.2.6. Позначити останній вертикальний набір як «готовий».

2. На основі побудованого списку згенерувати об'єкт класу GraphPetriNet:

2.1. Обнулити координати  $x$  та  $y$ .

2.2. Для кожного вертикального набору (починаючи з фізично першого у списку) виконати:

2.2.1. Збільшити  $x$  на 80 (пікселів).

2.2.2. Присвоїти значення  $y$  (якщо у вертикальному наборі парна кількість елементів, то  $y = -$  половина даної кількості, помножена на 80, - 40; інакше  $y = -$  половина даної кількості, помножена на 80, - 80).

2.2.3. Для кожного елементу вертикального набору виконати:

2.2.3.1. Збільшити  $y$  на 80.

2.2.3.2. Якщо тип набору – для позицій, виконати:

2.2.3.2.1. Створити на основі цієї позиції об'єкт класу GraphPetriPlace (графічний аналог позиції) з центром в точці  $(x, y)$ .

2.2.3.2.2. Додати цей об'єкт до списку графічних аналогів позицій.

2.2.3.3. Інакше виконати:

2.2.3.3.1. Створити на основі такого переходу об'єкт класу GraphPetriTransition (графічний аналог переходу) з центром в точці  $(x, y)$ .

2.2.3.3.2. Додати цей об'єкт до списку графічних аналогів переходів.

2.3. З'єднати графічні аналоги позицій та переходів вхідними та вихідними дугами.

2.4. Створити об'єкт класу GraphPetriNet, передавши у конструктор побудовані списки графічних аналогів позицій, переходів і дуг.

2.5. Перемістити даний об'єкт до центру видимої області візуального простору.

2.6. Повернути об'єкт з методу.

Фрагмент коду, який формує списки графічних аналогів позицій та переходів (п. 2.2), наступний:

```
for (VerticalSet set : sets) {
    ArrayList<PetriMainElement> elements = set.GetElements();
    int size = elements.size();
    x += 80;
    y = ((size % 2) == 0) ? (- (size / 2 * 80) - 40) : (- (size / 2 * 80) - 80);
    for (PetriMainElement elem : elements) {
        y += 80;
        if (set.IsForPlaces()) {
            PetriP place = (PetriP)elem;
            GraphPetriPlace grPlace = new GraphPetriPlace(place);
            grPlace.setNewCoordinates(new Point2D.Double(x, y));
            grPlaces.add(grPlace);
        } else {
            PetriT tran = (PetriT)elem;
            GraphPetriTransition grTran = new GraphPetriTransition(tran);
            grTran.setNewCoordinates(new Point2D.Double(x, y));
            grTransitions.add(grTran);
        }
    }
}
```

TECHNICAL SCIENCES AND TECHNOLOGIES

Загалом для реалізації нового компонента було модифіковано два існуючих класи бібліотеки graphpresentation (PetriNetsFrame і FileUse) та створено два нових класи (VerticalSet і MethodNameDialogPanel).

Результат генерування методу на основі візуальної моделі, побудованої за допомогою графічного інтерфейсу, показано на рис. 2.

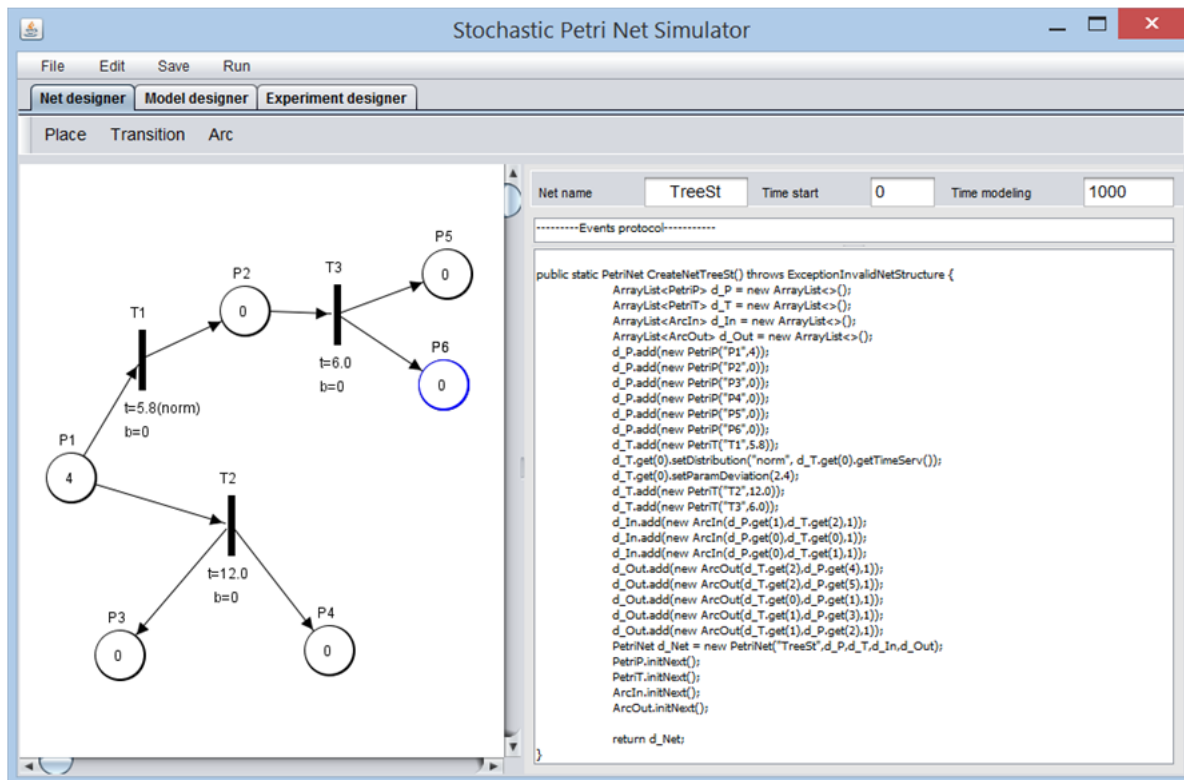


Рис. 2. Метод, згенерований на основі візуальної моделі

Результат відновлення візуальної моделі з методу демонструє рис. 3.

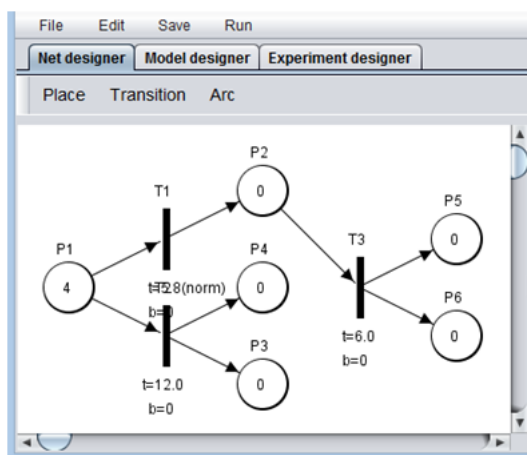


Рис. 3. Візуальна модель системи, згенерована на основі Java методу

**Висновки і пропозиції.** У результаті наукового дослідження розроблено інтелектуальний компонент візуального програмування стохастичних мереж Петрі, що являє собою вдосконалену версію однієї з бібліотек Java-класів програмного засобу Петрі-об'єктного моделювання дискретно-подійних систем. Цей компонент призначений для збереження моделей, побудованих за допомогою маніпулювання графічними об'єктами, у вигляді програмного коду мовою Java, а також відновлення візуальних моделей з програмного коду.

Розроблений компонент підвищує легкість сприйняття моделі користувачем за рахунок автоматичного розташування елементів мережі Петрі, що відновлюється з програмного коду, таким чином, щоб мінімізувати кількість перетинів між дугами та випадків, коли будь-які елементи мережі частково перекривають один одного. Крім того, компонент забезпечує вищу швидкість побудови та модифікації імітаційних моделей дискретно-подійних систем.

Інтелектуальний компонент планується використовувати під час розроблення іншого компонента програмного забезпечення Петрі-об'єктного моделювання, який дозволить би створювати візуальними засобами моделі систем вищого рівня та імітувати їхнє функціонування. Під моделями систем вищого рівня маються на увазі такі моделі, що складаються із великої кількості мереж Петрі. Також розроблений компонент може безпосередньо слугувати для швидкої побудови і модифікації простих стохастичних мереж Петрі, а також їхнього тиражування. Зручність у використанні компонента та відсутність необхідності знати мову Java роблять можливою роботу з імітаційними моделями користувачів, які не розуміються на розробці програмного забезпечення, а володіють лише базовими знаннями в галузі моделювання за допомогою мереж Петрі.

### Список використаних джерел

1. Zaitsev, D. A. Clans of Petri Nets: Verification of protocols and performance evaluation of networks / D. A. Zaitsev. – LAP LAMBERT Academic Publishing. – 2013. – 292 p.
2. Haas, P. J. Stochastic Petri Nets: Modelling, Stability, Simulation / P. J. Haas. – Springer Science & Business Media, 2006. – 510 p.
3. Aalst, W. Modeling Business Process – A Petri Net-Oriented Approach / W. Aalst, C. Stahl // The MIT Press. – 2011. – 400 p.
4. Lakos, C. Object Oriented Modeling with Object Petri Nets / C. Lakos // Concurrent Object-Oriented Programming and Petri Nets. – 2001. – P. 1-37.
5. Стеценко І. В. Теоретические основы Петри-объектного моделирования систем / И. В. Стеценко // Математичні машини і системи. – 2011. – № 4. – С. 136–148.
6. Stetsenko, I. V. Petri-Object Simulation: Software Package and Complexity / I. Stetsenko, V. Dorosh, A. Dyfuchyn // Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015). – Warsaw (Poland), 2015. – P. 381–385.
7. Стеценко І. В. Об'єктно-орієнтоване моделювання систем з використанням мереж Петрі / І. В. Стеценко // Вісник ЧДТУ. – 2011. – № 2. – С. 3–9.
8. Стеценко І. В. Проектування графічного модуля програмного забезпечення Петрі-об'єктного моделювання систем / І. В. Стеценко, О. В. Василевська // Вісник ЧДТУ. – 2013. – № 2. – С. 13–18.

### References

1. Zaitsev, D.A. (2013). *Clans of Petri Nets: Verification of protocols and performance evaluation of networks*. LAP LAMBERT Academic Publishing.
2. Haas, P.J. (2006). *Stochastic Petri Nets: Modelling, Stability, Simulation*. Springer Science & Business Media.
3. Aalst, W., Stahl, C. (2011). *Modeling Business Process – A Petri Net-Oriented Approach*. The MIT Press.
4. Lakos, C. (2001). Object Oriented Modeling with Object Petri Nets. *Concurrent Object-Oriented Programming and Petri Nets*, pp. 1–37.
5. Stetsenko, I.V. (2011). Teoreticheskie osnovy Petri-obektnogo modelirovaniia system [Theoretical foundations of the Petri-object simulation modeling]. *Matematychni mashyny i systemy – Mathematical Machines and Systems*, no. 4, pp. 136–148 (in Russian).
6. Stetsenko, I., Dorosh, V., Dyfuchyn, A. (2015). Petri-Object Simulation: Software Package and Complexity. *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015)*. Warsaw (Poland), pp. 381–385.



## TECHNICAL SCIENCES AND TECHNOLOGIES

7. Stetsenko, I.V. (2011). Obiektno-orientovane modeliuвання system z vykorystanniam merezh Petri [Object-oriented simulation modeling with the use of Petri nets]. *Visnyk ChDTU – Herald of Cherkassy state technological university*, no. 2, pp. 3–9 (in Ukrainian).

8. Stetsenko, I.V., Vasylevska, O.V. (2013). Proektuvannia hrafichnoho modulia prohramnoho zabezpechennia Petri-obiektnoho modeliuвання system [Graphic design of software module of Petri-object simulation systems]. *Visnyk ChDTU – Herald of Cherkassy state technological university*, no. 2, pp. 13-18 (in Ukrainian).

**Стеценко Інна Вячеславівна** – доктор технічних наук, доцент, професор кафедри автоматизованих систем обробки інформації і управління, Національний технічний університет України “КПІ ім. Ігоря Сікорського” (просп. Перемоги, 37, м. Київ, 03056, Україна).

**Стеценко Інна Вячеславівна** – доктор технических наук, доцент, профессор кафедры автоматизированных систем обработки информации и управления, Национальный технический университет Украины “КПИ им. Игоря Сикорского” (просп. Победы, 37, г. Киев, 03056, Украина).

**Stetsenko Inna** – Doctor of Technical Sciences, Associate Professor, Professor of the Department of Computer-Aided Management and Data Processing Systems, National Technical University of Ukraine “Igor Sikorsky KPI” (37 Peremohy Av., 03056 Kyiv, Ukraine).

**E-mail:** stiv.inna@gmail.com

**Scopus Author ID:** 55368781500

**Лещенко Катерина Сергіївна** – студентка, Національний технічний університет України “КПІ ім. Ігоря Сікорського” (просп. Перемоги, 37, м. Київ, 03056, Україна).

**Лещенко Екатерина Сергеевна** – студентка, Национальный технический университет Украины “КПИ им. Игоря Сикорского” (просп. Победы, 37, г. Киев, 03056, Украина).

**Leshchenko Kateryna** – student, National Technical University of Ukraine “Igor Sikorsky KPI” (37 Peremohy Av., 03056 Kyiv, Ukraine).

**E-mail:** katrean@inbox.ru