

DOI: 10.20535/kpi-sn.2019.5-6.174596

УДК 519.179.2:004.942

О.О. Супруненко*

Черкаський національний університет ім. Богдана Хмельницького, Черкаси, Україна

*corresponding author: ra-oks@i.ua

КОМБІНОВАНИЙ ПІДХІД ДО ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ДИНАМІКИ ПРОГРАМНИХ СИСТЕМ НА ОСНОВІ ІНТЕРПРЕТАЦІЙ МЕРЕЖ ПЕТРІ

Проблематика. Стаття стосується питання побудови комбінованого підходу до імітаційного моделювання динаміки програмних систем на основі WF-мереж та оціночних інтерпретацій мереж Петрі. Представлено дослідження інтерпретацій мереж Петрі та формальних мов, якими вони описуються, для їх адаптації до моделювання динаміки програмних систем.

Мета дослідження. Обґрунтування застосування комбінованого підходу на основі WF-мереж та оціночних мереж Петрі для моделювання програмних систем; визначення особливостей інструментарію при описі програмних систем мережами Петрі; доповнення патернів блочного моделювання конструкціями, що дають можливість описати задачі синхронізації, контролю, циклічного функціонування та їх комбінацій.

Методика реалізації. Представлено доведення можливості використання мереж Петрі, які описуються мовою L-типу та мовою G-типу в єдиній моделі. Наведено та проаналізовано структурні варіанти використання конструкцій мереж Петрі з їх параметрами для забезпечення живості та безконфліктного функціонування сформованої моделі. Представлено модель інтерфейсу користувача розподіленої системи, що забезпечує спільне редагування документа. На прикладі узагальненої моделі складання та поширення розкладу продемонстровано використання патернів при аналізі мережного представлення системи.

Результати дослідження. Сформовано правила побудови моделей програмних систем, що зберігають структурну безконфліктність і дають можливість аналізувати параметри елементів моделі при імітаційному моделюванні.

Висновки. У статті дано підґрунтя для розробки комбінованого підходу до моделювання програмних систем, яке базується на сполученні WF-мереж та оціночних мереж Петрі, що дає можливість враховувати особливості опису програмних систем при формуванні динамічних моделей.

Ключові слова: моделювання програмної системи; мережа Петрі; WF-мережа; формальні мови мереж Петрі; задачі синхронізації.

Вступ

Моделювання програмної системи на етапі її проектування дає можливість уточнити структуру системи, з'ясувати необхідні умови її передбачуваної поведінки, а також здешевити створення програмного продукту за рахунок раннього виявлення неузгодженостей зі специфікацією та помилок у реалізації функціональних особливостей системи. Оскільки програмні системи належать до недетермінованих динамічних систем, їх моделювання важливо проводити з використанням засобів динамічного моделювання, які дають змогу описати структуру системи, відобразити та проаналізувати множину процесів, серед яких часто виявляються асинхронні паралельні процеси, оцінити ефективність функціонування та досяжність визначених станів системи. Цим вимогам цілком відповідає апарат мереж Петрі, який має переваги порівняно з іншими засобами динамічного моделювання.

Так, графові засоби динамічного моделювання, до яких належать орієнтовані графи з метризованими відношеннями [1], функціональні мережі [2], динамічні орієнтовані графи [3], π -числення, алгебри процесів [4] дають змогу описувати систему в певний момент часу та моделювати її поведінку під впливом зовнішніх факторів. Ці графові засоби моделювання дають можливість описати стани, в які переходить система під впливом певних факторів, характеристики цих станів і умови зміни станів, однак вони мають досить багато обмежень і складні у формуванні структури й аналізі моделі.

Для моделювання задач планування та управління застосовують орієнтовані зважені графи з метризованими відношеннями, які дають змогу отримати однозначний математичний опис моделі та призначені для візуального аналізу [5] модельованої системи. У задачах системної динаміки використовуються елементи E-мереж, які уможливають проведення часткового автоматизованого аналізу побудованих

моделей, але накладають досить жорсткі обмеження на опис елементів досліджуваної системи.

Функціональні мережі можуть використовуватися для опису детермінованих і стохастичних складових модельованих систем. Їх різновидами є мережі GERT і стохастичні інтерпретації мереж Петрі [2]. Мережі GERT дають можливість моделювати циклічні, багаторазово повторювані та паралельні події, вони порівняно з деревами подій чи деревами відмов мають більші логічні можливості. При формуванні мережі GERT використовуються дев'ять різних символів і три типи дуг [6], тому ці мережі досить складно використовувати й аналізувати.

При використанні динамічних орієнтованих графів є можливість проводити вербально-кількісний опис модельованої системи, а також здійснювати імітаційне моделювання системи. При детальному аналізі елементів динамічних орієнтованих графів [3] можна провести паралель з інтерпретаціями мереж Петрі, зокрема з оціночними мережами Петрі [7]. Розбіжності спостерігаються тільки в наявності петель у динамічних орграфах, які в мережах Петрі (PN) не можуть реалізовуватися безпосередньо. Їх реалізація у PN можлива через інший тип вершин [7], тобто якщо петля починається від вершини переходу, то вона має пройти через додаткову вершину місця і тільки після цього повертається до вершини переходу, яка є її початком. Така конструкція дає змогу фіксувати поточний стан системи, що важливо для її подальшого аналізу.

Також значний інтерес на сьогодні прикутий до алгебри процесів, нотація якої дає можливість описувати процеси для стохастичних моделей із паралельними структурами, а також для побудови моделей, інформація про які є неповною [7]. В алгебрі процесів застосовується технологія "блочного моделювання", яка дає змогу на верхньому рівні моделі формувати структуру та визначати характер взаємодії компонентів, а на нижньому рівні описувати алгоритми обробки робочого навантаження окремими компонентами моделі [7], що підвищує адекватність опису моделі. Але для візуального аналізу систем при їх проектуванні такий інструмент виявився ускладненим, оскільки дії (описуються вершинами переходів) моделюються явним чином, а стани – неявним [9]. Тому в нашій роботі використовується інструментарій моделювання, який дає можливість у явному вигляді описувати і проводити аналіз як переходів, які моделюють дії в системі, так і

станів, у яких перебуває система між певними переходами.

Основою цього інструментарію є інтерпретації класичних мереж Петрі, для яких виявлено комплекс властивостей, що дають змогу проводити аналіз функціонування моделі та відлагоджувати її структуру відповідно до вимог, зокрема за властивостями живості, досяжності та збереженості [10]. Для відображення функцій керування та контролю створено мережі Петрі з пустими переходами, які дають можливість зосередити увагу дослідника на зовнішніх функціях модельованої системи, а разом із тим проаналізувати і внутрішню динаміку, що особливо важливо при аналізі роботи з контрольними точками та ресурсами [11]. Також на основі послідовностей $\sigma_{1k} = t_1, t_2, \dots, t_k$ автори роботи [12] пропонують методи синтезу мереж Петрі за визначеними вимогами. Натомість ці інтерпретації мають низку обмежень, які не дають змоги використовувати структуроподібний опис програмних систем у моделі, або властивості, такі, наприклад, як безконфліктність, доведені для обмеженого класу інтерпретації [13].

Постановка задачі

Метою роботи є побудова комбінованого підходу до імітаційного моделювання динаміки програмних систем на основі WF-мереж та оціночних інтерпретацій мереж Петрі. Для досягнення мети розв'язуються такі *задачі*: 1) обґрунтовується можливість застосування комбінованого інструментарію на основі WF-мереж та оціночних мереж Петрі для моделювання програмних систем, 2) визначаються основні відмінності між структурними особливостями потоків робіт і описом процесів у програмних системах, 3) доповнюється блочний інструментарій, побудований на основі WF-мереж, для опису програмних систем.

Формування комбінованого підходу до імітаційного моделювання програмних систем на основі інтерпретацій мереж Петрі

Теорія мереж Петрі ґрунтується на теорії автоматів [14]. Особливістю мереж Петрі є наявність двох видів несумісних вершин: вершин переходів t_i та вершин місць p_ϵ , з'єднаних між собою орієнтованими дугами за певними правилами S . У мережах Петрі переходам t_i відповідають однойменні автомати A_i , а вузлам p_ϵ –

умови активізації чи постактивізації (p_f) автоматів A_i [15].

Класична мережа Петрі (PN) є спеціалізованим біхроматичним (дводольним) несумісним мультиграфом

$$PN = (P, T, K, S), \quad (1)$$

який складається з двох типів вершин – вершин переходів $T = \{t_i\}$ та вершин місць $P = \{p_\varepsilon\}$, з'єднаних між собою спрямованими дугами $K = \{k_q\}$ за функціональними правилами S .

Під терміном “несумісний” розуміють неможливість з'єднання вершин з однойменних множин, тобто можуть бути з'єднані елементи множини вершин місць з елементами множини вершин переходів чи навпаки $K \subseteq P \times T \cup T \times P$.

Набір функціональних правил класичної мережі Петрі [14]

$$S = \{I_Q, I_R, w, b, \mu, L\} \quad (2)$$

вміщує функції інцидентності

$$I_Q : K_Q \rightarrow Q \cup P \times T \text{ та } I_R : K_R \rightarrow R \cup T \times P,$$

які визначають правила з'єднання вершин мережі Петрі між собою, вагову функцію $w : K \rightarrow N$, яка описує кратність дуг мережі, функцію позначень $b : T \cup P \rightarrow L$, якою розмічають елементи мережі для здійснення аналізу, розміткову функцію $\mu : P \rightarrow N \cup \{0\}$, яка ставить у відповідність кожному місцю $p_\varepsilon \in P$ ціле число міток N чи 0, абетку сигналів керування $L' = X \cup Y$, яка визначає вхідні $X = \{x_1, x_2, \dots, x_s\}$ та вихідні $Y = \{y_1, y_2, \dots, y_v\}$ множини сигналів керування.

На сьогодні розроблені численні інтерпретації та модифікації мереж Петрі, що обумовлено адаптацією класичних мереж Петрі до прикладних задач і прагненням позбавитись частини критичних властивостей [7]. Основними інтерпретаціями мереж Петрі є безпечні, оціночні, числові (макрочислові), кольорові та часові мережі Петрі. Сучасні інтерпретації та модифікації мереж Петрі включають ієрархічні та вкладені мережі [16], які дають змогу забезпечити композиційний опис паралельних та конкуруючих процесів у складних системах і мають розвинений інструментарій для їх дослідження. Мережі Петрі з інгібіторними дугами подібні до семантики структур подій [17].

Розроблені чотири класи формальних мов мереж Петрі, що важливо для моделювання й

аналізу побудованих моделей. Клас мов мереж Петрі визначається [14] як клас формальних мов, який породжується мережами Петрі. Він має абетку Σ – кінцеву множину символів, з елементів якої утворюється рядок – будь-яка послідовність символів з абетки Σ . Існує рядок нульової довжини – пустий рядок λ , він не має жодного символа. Множину всіх рядків із символів абетки Σ позначають Σ^* . Множина всіх непустих рядків записується як Σ^+ , тоді $\Sigma^* = \Sigma^+ \cup \{\lambda\}$. У мережі Петрі події, що відбуваються в системі, описуються вершинами переходів, тому послідовністю запусків переходів можна описати функціонування побудованої моделі.

Мова мереж Петрі утворюється як множина рядків над абеткою Σ символів, що позначають кінцеву множину переходів функцією помічення $\sigma : T \rightarrow \Sigma$, а також початковий стан s_0 і множину кінцевих станів M_F . Таку мережу Петрі називають поміченою, а мову мережі Петрі визначають як

$$\gamma = (PN, \Sigma, \sigma, s_0, M_F), \quad (3)$$

де PN – мережа Петрі (1) з функціональними правилами S (2). Існують чотири класи мов мереж Петрі, що характеризуються множиною початкових станів, абеткою та множиною кінцевих станів [15]. Множина початкових станів визначається множиною маркувань $M = \{\mu_1, \mu_2, \dots, \mu_k\}$. Абетка мов мереж Петрі пов'язана з вершинами переходів, тому функція помічення $\sigma : T \rightarrow \Sigma$. Таким чином, послідовність запусків переходів мережі породжує рядок мови мереж Петрі.

Залежно від початкової розмітки розрізняють довільне маркування μ_0 PN, маркування початкової вершини $\mu(p_0) = \zeta$, при цьому всі інші вершини не мають міток, і множину початкових маркувань $M = \{\mu_1, \mu_2, \dots, \mu_k\}$, які у загальному вигляді можуть задавати початковий стан мережі. Ці способи розмітки можуть бути перетворені один в інший [14], зазвичай вважають, що початкова розмітка поміченої мережі Петрі (3) задається певним вектором маркувань μ_0 .

Абетка мережі Петрі пов'язана з вершинами переходів, оскільки саме вони визначають особливості функціонування мережі, тому зв'язок символів абетки з переходами задається функцією помічення $\sigma : T \rightarrow \Sigma$. Залежно від обмежень, які накладаються на функцію помічення σ , розрізняють вільні (вільно помічені) мережі Петрі, без λ - та з λ -переходами. У вільно поміче-

ній мережі Петрі всі переходи помічені порізно, тобто якщо $t_i = t_j$, то $\sigma(t_i) = \sigma(t_j)$. Якщо в мережі Петрі допускаються переходи з пустим символом $\sigma(t_\emptyset) = \lambda$, то мова цієї мережі Петрі допускає, що такий перехід не буде з'являтися у рядках мови мережі Петрі і, отже, не буде фіксуватися при її функціонуванні. Таким чином, розрізняють три класи функцій помічення: вільна, без λ - та з λ -переходами.

Множина підсумкових станів мережі Петрі визначена як кінцева множина підсумкових маркувань мережі M_F [15]. Визначені чотири множини підсумкових маркувань, залежно від яких розрізняють чотири класи мов мереж Петрі – мови L-типу, G-типу, T-типу і P-типу.

Мовою мереж Петрі L-типу є мова, що описує мережу Петрі $PN = (P, T, K, Q, R)$ з поміченням вершин переходів $\sigma : T \rightarrow \Sigma$, початковим маркуванням μ_0 та множиною підсумкових маркувань M_F , таких, що

$$L = \{\sigma(t_j) \in \Sigma^* \mid t_j \in T \wedge \delta(\mu_0, t_j) \in M_F\}.$$

Мову мереж Петрі L-типу називають мовою цілком закінчених мереж Петрі [18]. За визначенням мова L-типу є потужною, але найбільш обмеженою, що дає можливість більш просто описати нею основні властивості мереж Петрі. Її використовують для опису й аналізу бізнес-процесів, наприклад для опису WF-мереж [9]. Але мова мереж Петрі L-типу $L(\gamma)$ не дає змоги використати весь набір корисних властивостей цих мереж для моделювання програмних систем із паралелізмом і конкуруючими процесами, наприклад таких, як запуск моделі з довільної розмітки, яка відповідає проміжному стану цільового процесу, а також задачі синхронізації [15], одна з яких – організація безпечного доступу до критичної секції, тощо.

Більшу кількість моделей на основі мереж Петрі дає можливість описати клас мов мереж Петрі G-типу; цю мову ще називають вільною мовою мереж Петрі [18]. Мовою мереж Петрі G-типу є мова, для якої визначена мережа Петрі $PN = (P, T, K, Q, R)$ з поміченням вершин переходів $\sigma : T \rightarrow \Sigma$, початковим маркуванням μ_0 та множиною підсумкових маркувань M_F , таких, що

$$L = \{\sigma(t_j) \in \Sigma^* \mid t_j \in T \wedge \exists \mu_f \in M_F (\delta(\mu_f, t_j) \geq \mu'_f)\}.$$

Класи мов мереж Петрі тісно пов'язані між собою. Так наприклад, у мову мереж Петрі L-типу

з λ -переходами входять мова L-типу без λ -переходів, у яку входить мова вільно помічених мереж Петрі L-типу: $L^f \subseteq L \subseteq L^\lambda$. Також будь-яка мова P-типу входить у клас мов G-типу $P \subseteq G$, а будь-яка мова G-типу входить у клас мов L-типу $G \subseteq L$. Мови мереж Петрі L- і T-типу еквівалентні. Таким чином, найбільший клас мов мереж Петрі утворюють мови L-типу, які є замкненими відносно об'єднання, перетину, конкатенації, паралельної композиції та регулярної (кінцевої) підстановки, також для них розв'язуваною є задача приналежності. Це створює добрі передумови для використання мереж Петрі, що описуються мовами L-типу, для опису структури й аналізу складних програмних систем на основі примітивів [19]. Наприклад, це дає можливість формувати структуру модельованої системи в термінах оціночних мереж Петрі, які описуються мовами G-типу, що забезпечить компактне представлення алгоритмічних рішень. Як основу для побудови примітивів, що слугують для прискорення побудови моделі на основі стандартних паттернів, можна використовувати WF-мережі, які описуються мовою L-типу. Таке сполучення інтерпретацій мереж Петрі можливе, оскільки мови G-типу входить у клас мов L-типу: $G \subseteq L$.

Практичні приклади застосування комбінованого підходу до моделювання програмних систем

Мова мереж Петрі G-типу породжується, наприклад, безпечними та оціночними інтерпретаціями мереж Петрі. Оціночні мережі Петрі дають можливість у більш компактній формі представляти алгоритмічні конструкції, вони зводяться до безпечних мереж Петрі через еквівалентні перетворення [14], а в термінах безпечних мереж Петрі простіше довести основні властивості PN. У термінах мереж Петрі, які описуються мовою G-типу, можна відобразити доступ до критичної секції (рис. 1, а), ця конструкція може бути перетворена до інтерпретації, яка описується мовою L-типу для доведення безконфліктності [9] таких структур. На рис. 1, б представлена конструкція, яка еквівалентна наведеній на рис. 1, а, побудована в термінах WF-мереж, у якій безпечний доступ до критичної секції можна забезпечити такою ж структурою з доповненням: початкову розмітку не представляти в контролюючій вершині місця p_c , а обмежитись лише початковим маркуванням $\mu(p_0) = 1$,

яке дає змогу від додаткової вершини переходу t_0 маркуванням μ_1 відтворити в моделі початкове маркування мережі з прикладу на рис. 1, а. Це дає змогу позбавитись ситуації, в якій після відпрацювання моделі є можливість залишити певну кількість міток у деяких вершинах місць, що створює передумови для непередбачуваної поведінки побудованої моделі.

При аналізі критичних властивостей, таких як безпечність, обмежуваність, збережуваність, живість, досяжність і покриття, використовуються формальні мови мереж Петрі L-типу, в термінах яких доведення цих властивостей зводиться до доведення існування (досяжність одного стану досліджуваної системи з іншого заданого) та приналежності (можливості зазначеної послідовності дій у системі), що значно спрощує аналіз [15].

Якщо проаналізувати причини виникнення критичних властивостей, то потрібно розділити вершини переходів на вершини з обчислювальними функціями та вершини для прийняття рішень. Вершини місць розділимо на вершини, які передують обчисленням, та вершини місць, у яких описується прийняття рішень. Вершини, пов'язані з обчисленнями, мають одну вхідну та одну вихідну дуги, вони відповідають властивості безконфліктності, яка притаманна автоматним мережам Петрі [14]. Вершини, що описують прийняття рішень, мають кілька вихідних дуг, що і може породжувати критичні властивості.

Також, якщо в моделі є вершини, які мають кілька вихідних дуг, описуючи паралелізм, то

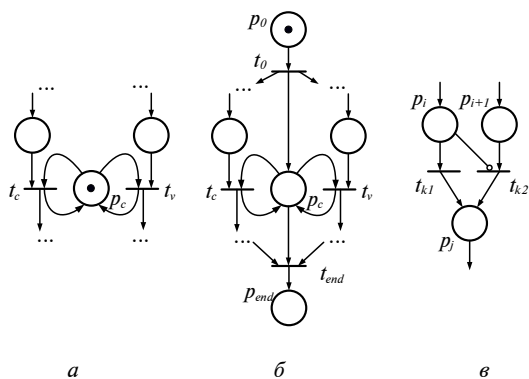


Рис. 1. Опис доступу до критичної секції засобами мереж Петрі, що описуються мовами G-типу (а) та L-типу (б); опис безпечного доступу до вершини з одиничним обмеженням (в)

мають бути і вершини, які мають кілька вхідних дуг для забезпечення властивості збережуваності мережі Петрі. Використання таких вершин є необхідним при моделюванні систем із паралелізмом, але додає можливість виникнення критичних ситуацій, тому потрібно дотримуватись низки правил, які забезпечать побудову моделі зі структурною безконфліктністю. Зокрема, слід враховувати, що безпечні мережі Петрі є обмеженням оціночних мереж Петрі, кожна вершина місця може містити не більше однієї мітки $m(p_\epsilon) = n_\nu, n_\nu \in \{0, 1\}$, вхідна дуга до p_ϵ має одиничну вагу $w(k_\epsilon) = 1$.

Для забезпечення збережуваності скористаємось властивістю нестрогої збережуваності, яка полягає в тому, що мережа Петрі з початковим маркуванням μ_0 називається збережуваною відносно вектора зважування $\omega = \{\omega_1, \omega_2, \dots, \omega_n\}$, де $\omega_i \in \mathbb{N}^+$ та $n = |P|$, якщо для всіх розміток μ' виконується умова

$$\sum_{i=\{1,2,\dots,n\}} \omega_i \mu'(p_i) = \sum_{i=\{1,2,\dots,n\}} \omega_i \mu(p_i).$$

Розглянемо модель інтерфейсу користувача, яка дає змогу працювати над спільним документом двом користувачам (рис. 2). На початку відбуваються вхід у систему (t_1) та вибір подальших дій: t_2 – створення документа, t_3 – відкриття документа, t_4 – налаштування середовища. Після створення документа його початкове наповнення відбувається у вершині пере-

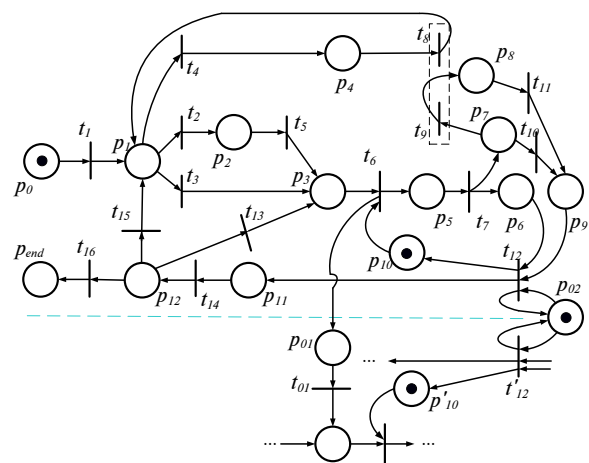


Рис. 2. Модель інтерфейсу користувачів для редагування спільного документа

ходу t_5 . Переміщення мітки у вершину місця p_3 свідчить про готовність першого користувача почати спільний перегляд та редагування документа. У вершині t_6 відбувається синхронізація перегляду документа двома користувачами, якщо другий користувач до цього готовий, інакше відбувається доступ до редагування документа одного користувача, а мітка для підключення другого користувача переміщується у вершину місця p_{01} , де перебуває до моменту, коли другий користувач буде готовий до спільної роботи над документом.

У вершині t_7 користувач проводить виділення фрагмента, до якого буде застосована певна функція. Якщо функція доступна в інтерфейсі, то переходимо до вибору функції (t_{10}), інакше заходимо у налаштування (t_9) і додаємо потрібну функцію (t_{11}) на панель інтерфейсу. У вершині переходу t_{12} має відбутися застосування функції до виділеного фрагмента, але така операція може паралельно здійснюватися і другим користувачем, тому для коректності внесення змін у документ забезпечимо послідовний доступ до змін у документі для двох користувачів із допомогою контрольної вершини місця p_{02} . Після цього зміни запам'ятовуються (t_{14}) та вибирається шлях (у вершині місця p_{12}) до подальшого редагування поточного документа (t_{13}), відкриття/створення для редагування іншого документа (t_{15}) чи завершення сеансу роботи ($t_{16} - p_{end}$).

Сформуємо ряд правил, які дадуть змогу побудувати мережну модель програмної системи з дотриманням структурної неконфліктності.

Правило 1. Сумарна вага вхідних дуг до кожної вершини місця не має перевищувати її місткості (обмеженості): $\sum_{p_e} w(k_{pe}) \leq v$. Це пра-

вило, застосоване до вершини місця з обмеженням $m(p_e) = n_v, n_v \in \{0, 1\}$, передбачає наявність у такої вершини однієї вхідної дуги з вагою $w(k_{pe}) = 1$. Якщо вхідних дуг з вагою 1 кілька, то потрібно застосувати примітив, який забезпечить безпечність цієї вершини (рис. 1, в). Якщо обмеженість вершини дорівнює певному цілому числу, як на рис. 3, а, то сумарна вага вхідних дуг не може її перевищувати, оскільки це призведе до порушення властивості обмеженості (на рис. 3, а – зліва).

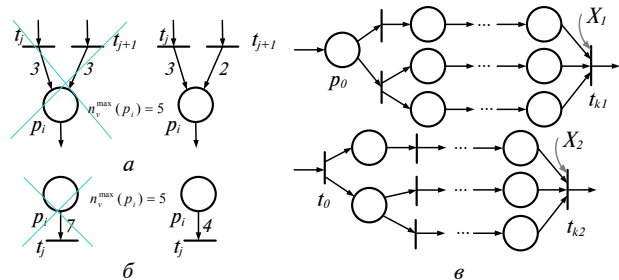


Рис. 3. Приклади застосування правил побудови PN при комбінованому підході

Правило 2. Дуга з вершини місця з максимальною вагою має підпорядковуватися нерівності $\max(w(k_{pe})) \leq n_v(p_e)$, тобто максимальна вага вихідної дуги не має перевищувати обмеження на число міток у цій вершині, інакше вона ніколи не зможе привести до спрацьовування вершини переходу, до якого вона є вхідною (рис. 3, б – зліва).

Правило 3. При утворенні кількох паралельних потоків від вершини переходу треба забезпечити їх об'єднання у певній вершині (певних вершинах) переходу, яка розміщується ближче до кінцевої вершини. У прикладі на рис. 2 паралельні потоки утворюються між парою вершин переходів t_7 і t_{12} .

Правило 4. При утворенні альтернативних потоків від вершини місця треба забезпечити їх об'єднання в іншій вершині місця, яка розміщується ближче до кінцевої вершини. У прикладі на рис. 2 альтернативні потоки утворюються між парою вершин місць p_1 і p_3 .

Правило 5. Коли в структурі моделі утворена комбінація паралельних і альтернативних потоків, потрібно їх об'єднувати у вершині переходу із зовнішнім керуючим вектором X_i . Це обумовлено необхідністю створення працездатного з'єднання при невикористанні певного альтернативного потоку, яке забезпечить X_i . Так, на рис. 3, в для спрацьовування вершини переходу t_{k1} потрібно, щоб хоча б один потік передав мітку в цей перехід (умова у X_1), а для спрацьовування вершини переходу t_{k2} потрібно, щоб хоча б два потоки з трьох передали мітки в цей перехід (умова у X_2). Злиття змішаних альтернативно-паралельних потоків зручно представляти однотипною конструкцією (рис. 3, в), оскільки це зменшить кількість побудов при вправленні логіки програмних додатків, що часто відбувається на початкових етапах проектування.

Правило 6. При створенні циклічних конструкцій необхідно передбачати контрольну вершину місця, яка належить до передкінцевих маркувань, у якій буде визначатись кількість граничних запусків циклу і яка при переповненні проінформує про виникнення зациклень.

На рис. 4. зображені варіанти циклів. У двох перших із них (рис. 4, а, б) контрольною вершиною місця є p_{i+1} , з якої виходить дуга вагою $w(k_{p_{i+1}}) = v$, що дає змогу перейти до кінцевої вершини p_{end} . По закінченні відпрацювання мережі, якщо в якихось вершинах, крім кінцевої, залишаться мітки, це буде свідчити про критичні властивості побудованої моделі, помилки у задаванні умови виходу з циклу чи в обчислювальному алгоритмі.

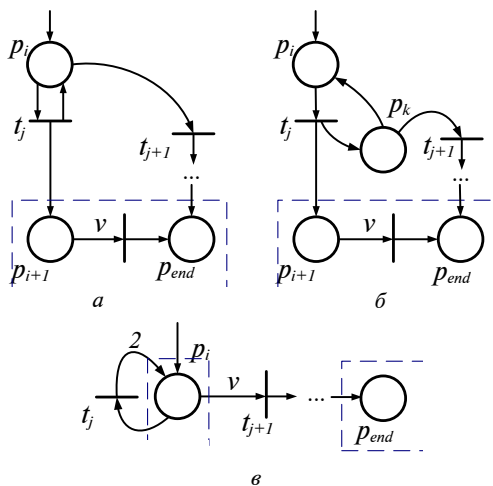


Рис. 4. Конструкції, які дають можливість контролювати скінченність циклів: а – циклу з передумовою, б – циклу з післяумовою, в – циклу типу *for*

У третьому виді циклу типу *for* (рис. 4, в) контролююча вершина p_i збігається з функціональною, вихідна дуга цієї вершини має вагу v , що забезпечує можливість закінчення поточного сеансу роботи. Але якщо по закінченні відпрацювання моделі (мітка у вершині p_{end}) у вершині p_i будуть залишатися мітки, це також свідчитиме про критичні властивості моделі та потребуватиме уточнення параметрів описаного алгоритму.

Правило 7. При утворенні конструкцій типу “буфер обміну” потрібно забезпечити рівноважність вхідних і вихідних міток, що досягається за рахунок врівноважування ваг вхідних і вихідних дуг:

$$\sum_{p_b} w(k_{p_b}) = \sum_{p_b} w(k_{p'_b}).$$

Одним із варіантів виникнення критичних ситуацій з причини структурної конфліктності є варіант задачі про виробника-споживача [15]. Буфер для передачі ресурсу від виробників до споживачів може накопичувати таку кількість міток, яка перевищує допустиму границю обмеження у вершині. Ця ситуація структурно може бути обумовлена неврівноваженістю ваг вхідних і вихідних дуг – сумарна вага вхідних дуг більша за сумарну вагу вихідних дуг $\sum_{p_b} w(k_{p_b}) > \sum_{p_b} w(k_{p'_b})$, –

що зумовлює поступове накопичення ресурсу вище визначених меж. Вирішення цієї критичної ситуації полягає у врівноваженні дуг чи в обмеженні буфера додатковою вершиною [15], що зробить функціонування виробників менш інтенсивним – при заповненні буфера вище допустимої межі виробник певний час чекатиме на звільнення його складових. Якщо продуктивність виробника вдвічі більша споживчих можливостей споживача, то рішенням може бути обслуговування двох споживачів через розподілення буфера.

Протилежна неврівноваженість ваги дуг $\sum_{p_b} w(k_{p_b}) < \sum_{p_b} w(k_{p'_b})$ викладає зниження інтенсивності роботи споживачів, які будуть чекати на вироблення ресурсу, і теж потребує врівноваження за першим варіантом. Якщо інтенсивність споживання вдвічі більша, ніж потужність виробника, то можливо запровадити двох виробників, які будуть наповнювати загальний буфер для передачі ресурсу.

При імітаційному моделюванні побудованих моделей на основі комбінації мереж Петрі можливо зосередити увагу на відлагодженні налаштування параметрів елементів моделі, оскільки структурні конфліктні ситуації будуть обмежені дотриманням наведених вище правил.

Розглянемо узагальнену модель розподіленої системи складання та поширення розкладу (рис. 5), застосуємо при її формуванні паттерни, наведені в [19]. На початку з допомогою паттерна “Дискримінатор” передбачимо можливість формування розкладу (з вершини місця p_0) або корекції складових початкових даних чи самого розкладу (з вершини місця p_{13}). З вершини t_1 , використовуючи паттерн “Паралельне розщеплення”, утворимо паралельні гілки, в яких будуть готуватися початкові дані. Такими даними

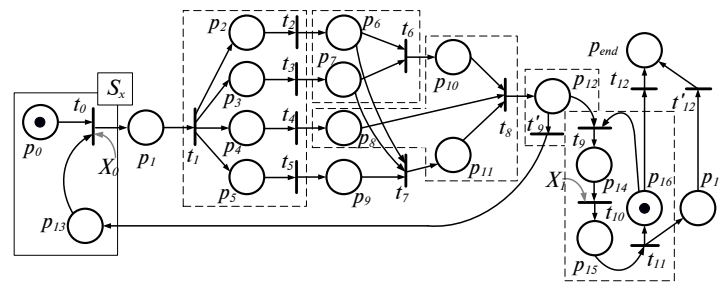


Рис. 5. Узагальнена модель розподіленої системи складання і поширення розкладу

для складання розкладу є формування робочих навчальних планів (p_3), штатний розпис викладачів (p_2), аудиторний фонд (p_4) і контингент студентів/груп (p_5). У вершині переходу t_6 буде відбуватися групування годин залежно від дисциплін і типу аудиторних занять, для відображення використовується паттерн “Синхронізація”. У вершині t_7 буде відбуватися формування навантаження викладачів залежно від контингенту студентів, навчальних планів спеціальностей і штатного розпису, що теж можливо відобразити паттерном “Синхронізація”, який включає вершини місць p_6 , p_7 , p_9 і вершину переходу t_7 . На рисунку він не обведений в окремий паттерн, оскільки зливається з деякими елементами попереднього паттерна. Його утворення буде потребувати роботи проектувальника з внутрішніми елементами паттерна. Для усунення таких ситуацій можливо застосувати дублювання вершин місць p_6 , p_7 , p_9 з входом від вершин переходів t_2 , t_3 , t_5 , що дасть змогу задіяти лише виходи попереднього паттерна і не вникати в особливості побудови паттернів усередині.

Ще одним паттерном “Синхронізація” відобразимо постановку занять у загальний розклад (вершина переходу t_8), на яку впливають не лише результати двох попередніх операцій (у p_{10} та p_{11}), але і завантаженість аудиторного фонду (p_8). У вершині p_{12} відбувається вибір – вносити правки у вхідні дані або розклад чи закінчити формування розкладу, що відображається паттерном “Відкладений вибір”.

В останньому паттерні “Цикл з післяумовою”, який починається від вершини переходу t_9 , відображена можливість корекції розкладу диспетчером (t_{10}) за наявності заявок від викладачів (у векторі X_1). Якщо заявок не надходить,

то відбувається звичайний моніторинг стану запитів на зміни; якщо до певного часу (перевірка у вершині місця p_{16}) заявок не надійшло, то перегляд розкладу для студентів і викладачів (через мобільний додаток) стає доступним (t_{12}). Контроль скінченності циклу відбувається у вершині місця p_{17} .

Проаналізувавши покриття представленої на рис. 5 моделі паттернами, можна помітити, що непокритими залишилися лише лінійні ділянки мережі (пов'язані з вершинами p_1 , t_{12} , $p_{17} - t'_{12}$), які можуть бути покриті автоматними мережами, що є безконфліктними. Це дає змогу стверджувати, що побудована модель є структурно безконфліктною. Якщо при імітаційному дослідженні виявляються помилки, то аналізують логіку процесів, особливу увагу приділяють умовам, які пов'язані з паттернами “Відкладений вибір” і “Цикл”.

Але не всі моделі так вдало покриваються паттернами. Зокрема, розглянемо ділянку мережі Петрі, яка відображає доступ до критичної секції (рис. 6, а). У цій ділянці теж можна виділити елементи паттернів “Відкладений вибір” і “Синхронізація”, але їх комбінація не є лінійною, тому така конструкція потребує додаткового аналізу живості та збережуваності.

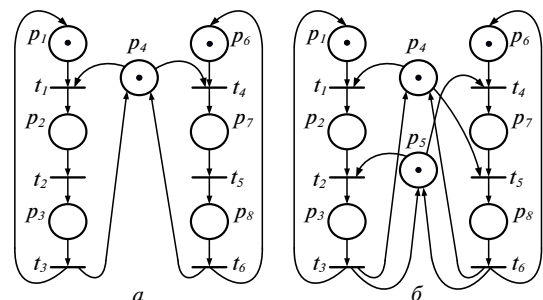


Рис. 6. Мережа Петрі, яка моделює доступ до критичної секції: а – доступ до критичної секції для двох процесів, б – використання двох ресурсів у різному порядку двома паралельними процесами

Так, наприклад, конструкція загального використання двох ресурсів двома паралельними процесами у різному порядку (рис. 6, б) має тупикову ситуацію при розмітці $\mu_i = (0, 1, 0, 0, 0, 0, 1, 0)$, яка виникає при одночасному спрацьовуванні вершин переходів t_1 і t_4 . Розв'язання цього тупика може критися у структурній надбудові, яка буде забезпечувати відставання одного з паралельних процесів від іншого не менше ніж на два такти, наприклад, вхід у всю критичну секцію має забезпечуватися одному з процесів, другий процес зможе увійти в неї лише тоді, коли попередній процес її покине. Можливе й інше рішення, яке стосуватиметься не структури, а особливостей розстановки початкового маркування, якщо в моделі можливо забезпечувати його надійність чи поновлюваність. Тому змішані конструкції потребують аналізу і розв'язання подібних критичних властивостей фахівцями, які можуть сформувати більш масивні конструкції, позбавлені критичних властивостей. Такі конструкції можуть застосовуватися при моделюванні проектувальниками.

Висновки

У статті дається підґрунтя для розробки комбінованого підходу до моделювання програмних систем, яке базується на сполученні властивостей WF-мереж та оціночних мереж Петрі. Розглянуто мови мереж Петрі, які породжуються цими типами інтерпретацій мереж Петрі, встановлено зв'язки між мовами L-типу та мовами G-типу, що дає змогу сформувати правила, які відображають обмеження для забезпе-

чення структурної безконфліктності при формуванні моделі програмної системи. Представлено правила побудови моделі системи з паралелізмом, які допомагають розширити базові паттерни [19], що запропоновані для моделювання бізнес-процесів, конструкціями, які враховують особливості програмних систем і дають можливість зберігати структурну безконфліктність і структуроподібність моделі при їх описі.

Перевагами запропонованого підходу є сполучення властивостей WF-мереж та оціночних мереж Петрі, які дають змогу проводити моделювання програмних систем із паралелізмом, враховуючи не тільки якісні, а й кількісні параметри, а також використовувати доведені властивості для WF-мереж і близьких інтерпретацій (як то однолічильникові PN) для доведення динамічних властивостей побудованих моделей, таких як обмежуваність, збережуваність, доступність і покриття.

До недоліків комбінованого підходу відноситься збільшення кількості обмежень при побудові моделей, що ускладнює використання методики та інструментів моделювання. Для зменшення цього недоліку можливо запропонувати використання базових паттернів [19], які дають змогу частково скомпенсувати цей недолік.

Перспективними є дослідження мереж Петрі вільного вибору, які дають можливість знизити кількість обмежень на модель, що слугує більш повному імітаційному моделюванню недетермінованих асинхронних програмних систем. Також перспективним є дослідження моделювання задач із необмеженими характеристиками засобами мереж Петрі, які мають кінцеву множину вершин.

References

- [1] A.I. Orlov, "Graphs for modeling industrial enterprise management processes", *Large Systems Management*, sp. iss. 30-1, pp. 62–75, 2010.
- [2] D. Phillips and A. Garcia-Diaz, *Network Analysis Methods (Fundamentals of network analysis)*. Moscow, Russia: Mir, 1984, 496 p.
- [3] G.A. Ugolnitskij, "Simulation and optimization models of complex systems, taking into account their structures", *Large Systems Management*, sp. iss. 30-1, pp. 799–816, 2010.
- [4] V.P. Strogalev and I.O. Tolkathova, *Simulation Modeling*. Moscow, Russia: Bauman Moscow State Technical University Publ., 2008, 280 p.
- [5] A.A. Kolobov et al., *High Tech Management. Integrated Production-Corporate Structures: Organization, Economics, Management, Design, Efficiency, Sustainability*. Moscow, Russia: Ekzamen, 2008, 621 p.
- [6] A. Prisker, *Modeling and Analysis Using Q-GERT Networks*. J. Willey & Sons, 1977, 312 p.
- [7] V.V. Kuzmuk and O.A. Suprunenko, "The means for the description of information flows in dynamic models of medical hardware-software systems", *Theor. Appl. Sci.*, vol. 15, no. 7, pp. 11–18, 2014.

- [8] B.B. Nesterenko and M.A. Novotarskii, "Process algebra for modeling complex systems with real workload", *Data Registration, Storage, and Processing*, vol. 9, no. 4, pp. 49–59, 2007.
- [9] W. van der Aalst and K. van Hee, *Workflow Management: Models, Methods and Systems*. Cambridge, Massachusetts, London: The MIT Press, 2002, 359 p.
- [10] J. Esparza *et al.* (2019). *Parameterized Analysis of Immediate Observation Petri Nets* [Online]. Available: <https://arxiv.org/pdf/1902.03025.pdf>
- [11] V.A. Bashkin, "On the resource equivalences in Petri nets with invisible transitions", in *Proc. Petri Nets and Software Engineering (PNSE'17)*, Zaragoza, Spain, June 25–30, 2017, pp. 51–68.
- [12] L.L. Mannel and W.M.P. van der Aalst, "Finding complex process-structures by exploiting the token-game", in *Proc. Int. Conf. Applications and Theory of Petri Nets and Concurrency*, vol. 11522, pp. 258–278, 2019. doi: 10/1007/978-3-030-21571-2_15
- [13] D.A. Zaicev and A.I. Slepov, "Equations of state and equivalent transformations of temporary Petri nets", *Cibernetika i Sistemnyi Analis*, no. 5, pp. 59–76, 1997.
- [14] V.V. Kuzmuk and O.O. Suprunenko, *Modified Petri nets and Devices for Modeling Parallel Processes*. Kyiv, Ukraine: Maclaut, 2010, 252 p.
- [15] J. Peterson, *Theory of Petri Nets and System Modeling*. Moscow, Russia: Mir, 1984, 264 p.
- [16] I.A. Lomazova, *Nested Petri Nets: Modeling and Analysis of Distributed Systems with an Object Structure*. Moscow, Russia: Nauchnyi Mir, 2004, 208 p.
- [17] P. Baldan *et al.*, "Domain and event structure semantics for petri nets with read and inhibitor arcs", *Theor. Comp. Sci.*, vol. 323, pp. 129–189, 2004. doi: 10.1016/j.tcs.2004.04.001
- [18] E.M. Bronshtein *et al.*, *Theory of Computational Processes*. Ufa, Russia: UGATU, 2012, 198 p.
- [19] O.A. Suprunenko, "Combined tools simulation workflow based graph models", *Theor. Appl. Sci.*, vol. 23, no. 3, pp. 153–158, 2015. doi: 10.15863/TAS.2015.03.23.26

O.A. Супруненко

КОМБИНИРОВАННЫЙ ПОДХОД К ИМИТАЦИОННОМУ МОДЕЛИРОВАНИЮ ДИНАМИКИ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ ИНТЕРПРЕТАЦИЙ СЕТЕЙ ПЕТРИ

Проблематика. Статья касается вопроса построения комбинированного подхода к имитационному моделированию динамики программных систем на основе WF-сетей и оценочных интерпретаций сетей Петри. Представлены исследования интерпретаций сетей Петри и формальных языков, которыми они описываются, для их адаптации к моделированию динамики программных систем.

Цель исследования. Обоснование применимости комбинированного подхода на основе WF-сетей и оценочных сетей Петри для моделирования программных систем; определение особенностей инструментария при описании программных систем сетями Петри; дополнение паттернов блочного моделирования конструкциями, которые позволяют описать задачи синхронизации, контроля, циклического функционирования и их комбинаций.

Методика реализации. Представлены доказательства возможности использования сетей Петри, которые описываются языком L-типа и языком G-типа в единой модели. Приведены и проанализированы структурные варианты использования конструкций сетей Петри с их параметрами для обеспечения жизнеспособности и бесконфликтного функционирования сформированной модели. Представлена модель интерфейса пользователя распределенной системы, которая обеспечивает совместное редактирование документа. На примере обобщенной модели составления и распространения расписания продемонстрировано использование паттернов при анализе сетевого представления модели.

Результаты исследования. Сформированы правила построения моделей программных систем, которые сохраняют структурную бесконфликтность и позволяют анализировать параметры элементов модели при имитационном моделировании.

Выводы. В статье представлены основания для разработки комбинированного подхода для моделирования программных систем, которые базируются на совмещении WF-сетей и оценочных сетей Петри, что позволяет учитывать особенности описания программных систем при формировании динамических моделей.

Ключевые слова: моделирование программной системы; сеть Петри; WF-сеть; формальные языки сетей Петри; задачи синхронизации.

O.O. Suprunenko

COMBINED APPROACH TO SIMULATION MODELING OF THE DYNAMICS OF SOFTWARE SYSTEMS BASED ON INTERPRETATIONS OF PETRI NETS

Background. The article presents the reasons for constructing a combined approach to simulation of the dynamics of software systems based on WF-nets and evaluative interpretations of Petri nets. The research of interpretations of Petri nets and formal languages with which they are described for their adaptation to modeling the dynamics of software systems is presented.

Objective. Justification of the applicability of the combined approach based on WF-nets and bounded Petri nets for modeling software systems, determination of the characteristics of tools for describing software systems using the Petri nets, additions of block modeling patterns with structures that allow describing the tasks of synchronization, control, cyclic functioning, and their combinations.

Methods. The paper presents the possibility of using Petri nets, which are described by the L-type and the G-type languages in a single model. The structural options for using the constructs of Petri nets with their parameters to ensure the liveness and safety of the

formed model are presented and analyzed. A model of the user interface of a distributed system that provides collaborative editing of a document is presented. Using a generalized model for schedule compilation and distribution, the use of patterns in the analysis of the network representation of a model is demonstrated.

Results. Rules for constructing models of software systems that maintain the structural conflict-free and allow you to analyze the parameters of the elements of the model during simulation were proposed.

Conclusions. The article presents the basis for the development of a combined approach for software systems modeling that are based on the combination of WF-nets and bounded Petri nets, which allows taking into account the features of the description of software systems when forming dynamic models.

Keywords: software system modeling; Petri net; WF-net; formal Petri nets languages; synchronization tasks.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
31 липня 2019 року

Прийнята до публікації
31 жовтня 2019 року