
DOI: <https://doi.org/10.15407/emodel.44.02.038>
УДК 004.942 : 519.876.5

О.О.Супруненко, канд. техн. наук.,
Б.О.Онищенко, канд. физ.-мат. наук, **Ю.Є.Гребенович**
Черкаський національний університет ім. Б. Хмельницького
Україна, 18006, Черкаси, бульв. Шевченка, 81
тел. (0472) 330759, e-mail: ra-oks@vu.cdu.edu.ua

Аналіз прихованих помилок у моделях програмних систем на основі мереж Петрі

Описано основні характеристики моделі програмної системи (ПС), до яких належать працездатність та передбачуваність функціонування. Проаналізовано інструментальні засоби, що використовуються для моделювання ПС. Побудова моделей та аналіз динамічних властивостей ПС основано на виборі інтерпретацій та модифікацій мереж Петрі (Petri Nets (PN)). Вони дозволяють дотримуватись структурної подібності у модельованій системі, а також мають однозначний математичний опис. Аналіз властивостей PN дає змогу виявляти явні та приховані помилки у функціонуванні PN-моделі ПС, що не завжди можливо при застосуванні імітаційного моделювання. До таких властивостей належать живість, обмеженість, досяжність (покриваємість), збережуваність, безконфліктність, керованість. Виявляти дотримання цих характеристик дозволяють *T*- та *P*-інваріанти, а також характеристики матриці інцидентності моделі. Проведено матричний опис PN-моделі за допомогою основного рівняння мережі Петрі, визначення інваріантів та їх аналіз для виявлення динамічних властивостей моделі. Сформульовано правило відсутності прихованих тупиків та нескінченних циклів у PN-моделі. Проілюстровано визначення прихованих помилок на прикладі робочої, але не повністю керованої PN-моделі. Описано виявлення прихованих помилок — тупика та нескінченного циклу. Запропоновано варіант усунення тупика та перевірено його відповідність динамічним властивостям моделі.

К л ю ч о в і с л о в а: мережа Петрі, PN-модель програмної системи, аналіз прихованих помилок.

При проектуванні програмних систем (ПС) важливим є досягнення високої якості характеристик працездатності та передбачуваності поведінки розробленої системи, що забезпечує успіх програмного проекту. Для забезпечення цих характеристик насамперед потрібно визначити тип проектованої системи. Програмна система складається з сукупності взаємопов'язаних компонентів, які взаємодіють між собою для досягнення певних бізнес-цілей [1], і має численні паралельні та конкуруючі

© Супруненко О.О., Онищенко Б.О., Гребенович Ю.Є., 2022

процеси на рівні компонентів та міжкомпонентних зв'язків [2]. Її структура може бути як ієрархічною, так і горизонтально інтегрованою. Тому ПС належать до складних систем з паралелізмом.

На етапі проектування ПС доводиться формувати та аналізувати набір моделей, які пов'язані зі структурною, функціональною та інформаційною складністю [3]. Такі моделі зазвичай мають суттєвий паралелізм, що створює передумови для непередбачуваного функціонування розроблюваної системи. Тому визначення проблемних ділянок моделі є актуальним завданням. Для практичного застосування важливо не тільки виявити проблемні ділянки, а й запропонувати рішення, які допоможуть усунути чи суттєво зменшити проблему, а також перевірити ці рішення.

Підґрунтя для розробки інструментальних засобів моделювання програмного забезпечення (ПЗ). Розробка проектів сучасних ПС передбачає взаємодію з недетермінованими процесами, які обумовлені характеристиками даних систем. Водночас, реалізація цих проектів має забезпечити їх передбачувану поведінку [4]. Для формування моделей ПС та їх подальшого ефективного аналізу потрібні підходи та інструментальні засоби, що дозволятимуть успішно відображати структурні особливості ПЗ та перевіряти динамічні характеристики, що безпосередньо впливають на працездатність та керованість розроблюваного програмного продукту.

Існують численні інструментальні засоби, які успішно вирішують локальні підзадачі цієї загальної проблеми. До них належать, наприклад, діаграми UML [5]. Групи UML-діаграм структури та поведінки застосовуються для структурного та функціонального аналізу розроблених систем, UML-діаграми взаємодії дозволяють відтворити динамічні аспекти взаємодії складових системи. Але відсутність засобів динамічної імітації, а також неузгодженість рівнів деталізації не дозволяють проаналізувати динамічні властивості ні окремих компонентів, ні всієї системи, що є проблемою при проектуванні ПЗ [2, 6].

Підзадачі аналізу у моделях недетермінованих систем з паралелізмом можна вирішувати такими засобами, як числення взаємодіючих систем, π -числення [7], алгебра процесів [8]. Але π -числення та алгебра процесів дозволяють описувати паралельні моделі у вигляді текстового, досить лінеаризованого, опису, що приводить до формування моделей, у яких «процеси робляться більш послідовними, ніж вони повинні бути» [9].

Вирішувати загальну задачу формування підходів до створення інструментальних засобів пропонується на основі імітаційного моделювання з використанням мереж Петрі, що являють собою розвинений графоаналітичний засіб моделювання систем з паралелізмом [2], побудований на основі графів, та парадигми динамічного моделювання [10].

Інструментальні засоби моделювання ПС з паралелізмом. Задачі аналізу динамічних характеристик ПС є дуже складними, тому стандарт IDEF 2 (динамічне моделювання) [6] не реалізований до нині і застосовується лише для обмежених динамічних моделей, побудованих, наприклад, на основі кольорових мереж Петрі (CPN) [11] та часових стохастичних мереж [12], які є інтерпретаціями та модифікаціями мереж Петрі.

Інші інтерпретації мереж Петрі також можуть бути залучені до створення інструментальних засобів моделювання та аналізу ПС, оскільки основою для них є численні графові інструменти [13], які використовуються для перевірки дискретних моделей реальних систем та пристосовані до їх програмної реалізації. До таких графових інструментів належать зважені орієнтовані графи, що є підґрунтям для бінарних розв'язуючих діаграм, які використовуються у верифікації методом model checking [14]. Також до переліку графових інструментів слід віднести динамічні оргграфи [15], які дозволяють доповнити якісні характеристики моделей кількісними характеристиками, що є важливим для практичного застосування у проектах ПС.

Розмірність ПС є досить великою, тому на початковому етапі моделювання ПС розбивають на компоненти, що дозволяє досліджувати окремі цільові характеристики. Цей етап називається мінімізацією складностей. Він оснований на технологіях об'єктного моделювання (об'єктно-орієнтоване моделювання (ООМ)), для яких є характерними тісні зв'язки даних і процесів у системі, що дозволяє розробляти більш надійні рішення, стійкі до змін та пристосовані до функціонування системи в реальному світі [16].

При створенні сучасних ПС широко застосовується і компонентно-орієнтована технологія [16, 17], яка підтримує концепцію успадкування (з ООМ) та передбачає використання нових концепцій таких, як інтроспективність (здатність до самоопису), модульність, персистентність, здатність до повторного використання компонентів [18]. Ці технології підтримуються ієрархічними та іншими інтерпретаціями мереж Петрі [19], які дозволяють адекватно описати системи, що створюються з використанням сучасних технологічних досягнень. Персистентність фактично означає збереження та відновлення стану компонента, що співвідноситься з такими властивостями мереж Петрі, як збережуваність і повторюваність; повторне використання компонентів у моделях забезпечується правилами побудови та взаємодії підмоделей в ієрархічних мережах Петрі.

Аналітичні засоби мереж Петрі для систем з паралелізмом. Приклад аналізу. Мережі Петрі дозволяють описувати моделі систем з па-

ралелізмом подібні за структурою до реальних систем, що спрощує їх формування і попередній візуальний аналіз та сприяє самоперевірці. Вони також дозволяють проводити імітаційне моделювання з метою аналізу динамічних властивостей побудованих моделей, що спрямовано переважно на виявлення та усунення явних помилкових рішень.

Важливо також зазначити, що моделі на основі PN дозволяють виявляти приховані помилки [20], використовуючи їх графове подання, що має однозначний математичний опис [21] та дозволяє дослідити основні властивості PN-моделей. До таких властивостей PN-моделей належать живість, обмеженість, досяжність (покриваємість), зберезуваність, безконфліктність, керованість [2, 22]. Для їх аналізу використовується матричний опис PN рівняннями стану [21] та метод інваріантів [23].

Для отримання розв'язку задачі досяжності, яка полягає у перевірці досяжності цільової розмітки μ_k з початкової розмітки μ_0 , потрібно розв'язати основне рівняння PN — рівняння стану [21]:

$$\mu_k = \mu_0 + W^T X, \quad (1)$$

де W — матриця інцидентності моделі; $X = \{v_{t1}, v_{t2}, \dots, v_{tm}\}$ — вектор Паріха, що є лічильником [23] спрацювання вершин переходів t_i для досягнення розмітки μ_k . Існування розв'язку є доведенням обмеженості PN та досяжності розмітки μ_k з початкової розмітки. Але отриманий результат не дає однозначного розв'язання задачі про шляхи досягнення розмітки, оскільки не визначає порядку запуску (спрацювання) вершин переходів. Це зумовлено тим, що розв'язання рівняння (1) є необхідною, але не достатньою умовою досягнення визначеної розмітки μ_k [24].

Множину розв'язків системи лінійних однорідних діофантових рівнянь $W^T X = 0$ називають T -інваріантом [21]. Він має забезпечувати існування послідовності переходів $\sigma_i = t_1, t_2, \dots, t_n$ з розмітки μ_0 до цієї ж початкової розмітки. T -інваріанти застосовуються при аналізі моделей на основі PN, зокрема за ними визначають властивість повторюваності, якщо всі місця переходів покриваються ненульовими координатами векторів із множини T -інваріантів. Властивість обмеженості визначається існуванням нетривіального рішення наведеної системи лінійних однорідних діофантових рівнянь.

При інваріантному аналізі PN застосовують також S -інваріанти [23], які є розв'язками системи лінійних однорідних діофантових рівнянь $WY = 0$. Розв'язок даної системи рівнянь відомий як S -інваріант. Він

дозволяє визначити властивості обмеженості, збережуваності та безконфліктності (несуперечливості) [24]. Про обмеженість PN свідчить покриття всіх вершин місць ненульовими координатами векторів з множини S -інваріант. Несуперечливість полягає у досяжності будь-якого маркування із нього ж при спрацьовуванні відповідної послідовності вершин переходів σ_i . Звідки впливає досяжність розмітки μ_0 з неї ж самої за послідовності σ_0 .

Розгляд матриці інцидентності дозволяє визначити неповну та повну керованість PN, яка є важливою характеристикою для розроблюваної ПС з паралелізмом. Повна керованість полягає у досяжності будь-якої розмітки PN з будь-якої іншої розмітки цієї мережі, що забезпечується, якщо виконується рівність

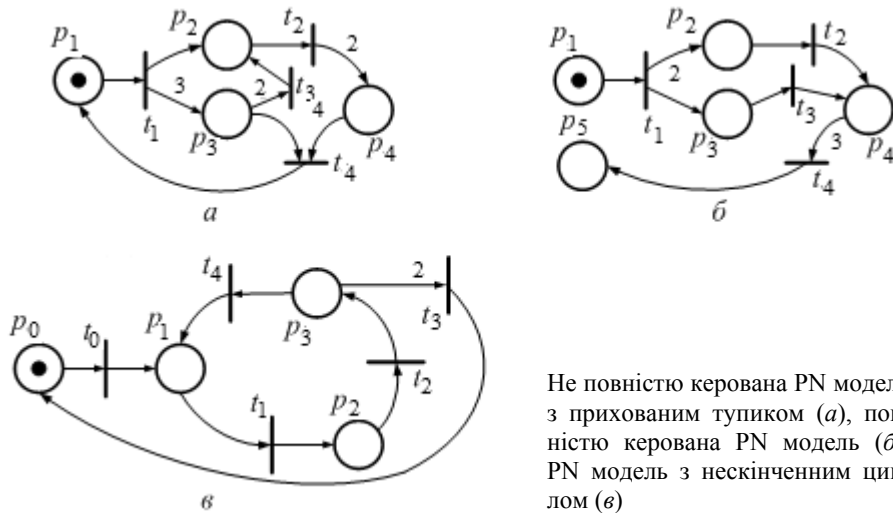
$$\text{rang}(W^T) = \min(n, m), \quad (2)$$

де n — кількість елементів у множині вершин місць; m — кількість елементів у множині вершин переходів PN. Якщо рівність не виконується, аналізована PN модель не є повністю керованою.

Приклад 1. Розв'яжемо та проаналізуємо рівняння стану для моделі, побудованої оціночною PN [19] (рисунок, а). Дана PN модель не має петель, що дозволяє проаналізувати всі варіанти її функціонування. Рівняння стану для неї має вигляд

$$\mu_k = \mu_0 + W^T X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 3 & 0 \\ 0 & -1 & 0 & 2 \\ 0 & 1 & -2 & 0 \\ 1 & 0 & -1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (3)$$

Починаючи з початкової розмітки $\mu_0 = [1 \ 0 \ 0 \ 0]^T$, отримаємо ряд послідовних розміток при спрацьовуванні: вершини переходу t_1 — $\mu_1 = [0 \ 1 \ 3 \ 0]^T$, дозволених з попередньої розмітки вершин переходів t_2, t_3 — $\mu_2 = [0 \ 1 \ 1 \ 2]^T$, вершини переходу t_2 вдруге — $\mu_3 = [0 \ 0 \ 1 \ 4]^T$ та вершини переходу t_4 — $\mu_4 = \mu_0 = [1 \ 0 \ 0 \ 0]^T$. Вектор X , який є лічильником кількості спрацьовувань переходів, поданий послідовністю $\sigma_4 = t_1, t_2, t_3, t_2, t_4$, знаходимо у вигляді $X = [1 \ 2 \ 1 \ 1]^T$. Вектор X є T -інваріантом, за яким досліджувана PN є обмеженою, оскільки всі складові інваріанта покриваються ненульовими значеннями. Вона також є повторюваною, оскільки з початкової розмітки μ_0 через чотири кроки досягається та ж сама початкова розмітка.



Не повністю керована PN модель з прихованим тупиком (а), повністю керована PN модель (б), PN модель з нескінченним циклом (в)

Для визначення інших властивостей PN знайдемо мінімальну множину S -інваріант для даного прикладу з системи рівнянь $WY = 0$. Skorистаємось методом TSS [23], за яким множину рішень вказаної системи рівнянь M'_j знаходять на основі множини векторів канонічного базису $M'_0 = \{e_1, \dots, e_4\}$, розглядаючи кожне рівняння системи. Елементи множини M'_0 розбивають на три групи: нульові M_j^0 , додатні M_j^+ та від'ємні M_j^- . Якщо хоча б дві з цих множин є непорожніми, існує нетривіальне рішення на множині натуральних чисел, інакше рівняння не має нетривіальних рішень з множини натуральних чисел. Елементи множини рішень для першого рівняння запишемо так [23]:

$$M'_1 = M_1^0 \cup \{y_{ij} \mid y_{ij} = -L_1(e_i) e_j + L_1(e_j) e_i, e_j \in M_1^+, e_i \in M_1^-\}. \quad (4)$$

Для другого рівняння будемо аналогічну формулу, якщо хоча б дві з множин M_2^0 , M_2^+ та M_2^- є непорожніми. Отже, знаходимо елементи множини M'_j , які є розв'язком системи рівнянь $WY = 0$. S -інваріант, який складає мінімальну множину рішень цієї системи рівнянь, отримано у вигляді $Y = [5 \ 2 \ 1 \ 1]^T$. Перевіримо отримане рішення, для чого запишемо послідовні розмітки, починаючи від знайденої за S -інваріантом, яку приймемо за початкову:

$$\mu'_0 \xrightarrow{t_1} \mu'_1 \xrightarrow{t_2} \mu'_2 \xrightarrow{t_3} \mu'_3 \xrightarrow{t_2} \mu'_4 \xrightarrow{t_4} \mu'_5;$$

$$\begin{bmatrix} 5 \\ 2 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{t_1} \begin{bmatrix} 4 \\ 3 \\ 4 \\ 1 \end{bmatrix} \xrightarrow{t_2} \begin{bmatrix} 4 \\ 2 \\ 4 \\ 3 \end{bmatrix} \xrightarrow{t_3} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 3 \end{bmatrix} \xrightarrow{t_2} \begin{bmatrix} 4 \\ 2 \\ 2 \\ 5 \end{bmatrix} \xrightarrow{t_4} \begin{bmatrix} 5 \\ 2 \\ 1 \\ 1 \end{bmatrix}. \quad (5)$$

З початкової розмітки μ'_0 через активізацію послідовності переходів отримаємо ту ж початкову розмітку. Аналіз S -інваріанту свідчить, що досліджувана мережа є обмеженою, оскільки всі її елементи є ненульовими; також вона є збережуваною і повторюваною, оскільки від початкової розмітки при активізації послідовності переходів $\sigma_4 = t_1, t_2, t_3, t_2, t_4$ дозволяє отримати ту ж початкову розмітку; вона є несуперечливою, оскільки у послідовності σ_4 кожен перехід спрацьовує хоча б один раз.

Розрахуємо ранг матриці W , який дорівнює трьом і є меншим за її розмірність (4×4): $\text{rang}(W) < 4$. Це значить, що розглянута РН є не повністю керованою. Тобто можуть існувати певні сценарії функціонування мережі, які включають безкінечні цикли чи тупики, що визначають непередбачувану поведінку моделі. Наприклад, якщо спрацьовування вершин переходів відбуватимуться не у тій послідовності, яка передбачена у лічильнику спрацьовування переходів (T -інваріанті), а, наприклад, з розмітки $\mu_{kr} = [5 \ 2 \ 1 \ 1]^T$ за наступною послідовністю спрацювання вершин переходів:

$$\sigma_{kr-t} = t_0, t_2, t_2, t_3, t_2, t_1, t_0, t_2, t_2, t_3, t_0, t_0, t_2, t_0, t_1, t_0, t_0, t_1, t_3, t_0, t_1, t_2, t_2, t_2, t_1, t_2, t_1, t_1, t_2, t_1, t_3, t_2, t_1, t_2, t_1, t_2, t_0, t_3, t_2, t_0, t_2, t_1, t_1, t_1, t_2, t_2, t_2, t_2, t_2, t_2, t_2, t_2, \quad (6)$$

то це приведе до розмітки $\mu_3 = [0 \ 0 \ 0 \ 31]^T$, яка фіксує досягнення тупика.

Для забезпечення відсутності тупика або нескінченного циклу у РН моделі без петель потрібно дотримуватись правила, яке має дві обов'язкові умови:

- 1) P -інваріанти покриті ненульовими елементами (покриття визначене для кожної вершини місця);
- 2) ранг матриці інцидентності W дорівнює мінімальній потужності множини вершин місць чи множини вершин переходів моделі: $\text{rang}(W) = \min(|T|, |P|)$.

На рисунку *б* показано варіант перебудованої моделі (див. рисунок *а*), для якої розраховано інваріанти та визначено ранг матриці інцидентності:

$$T_{5_7b} = [1 \ 2 \ 1 \ 1], \quad P_{5_7b} = [3 \ 1 \ 1 \ 1 \ 3], \quad W_{5_7b} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 2 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{rang}(W_{5_7b}) = 4.$$

При аналізі *T*- та *P*-інваріантів, а також рангу матриці інцидентності моделі, наведеної на рисунку *б*, визначено, що модель відповідає властивостям живості, повторюваності (за покриттям *T*-інваріанту), обмеженості та збережуваності (за покриттям *S*-інваріанту), а також є повністю керованою: $\text{rang}(W_{5_7b}) = \min(|T_{5_7b}|, |P_{5_7b}|) = 4$.

Приклад 2. При виявленні у PN-моделі нескінченного циклу кінцевої розмітки також не буде досягнуто. При імітаційному моделюванні такої моделі спостерігатиметься значне перевищення часу роботи без видачі кінцевого результату, наприклад моделі, поданій на рисунку *в*. Розраховані інваріанти та ранг матриці інцидентності для моделі з нескінченим циклом мають такий вигляд:

$$T_{nz} = [0 \ 1 \ 1 \ 0 \ 1], \quad S_{nz} = [0 \ 0 \ 0 \ 0 \ 0], \quad W_{nz} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -2 & -1 \end{bmatrix}, \quad \text{rang}(W_{nz}) = 4.$$

При аналізі *T*-інваріанту моделі з нескінченим циклом спостерігається покриття нульовими значеннями елементів, пов'язаними з вершиною переходу t_3 (що є вихідною вершиною з вершини місця p_3 , в якій визначено умову виходу з циклу, що не дозволяє запустити на виконання перехід t_3) та вершиною переходу t_0 (що вказує на неможливість відновлення початкової розмітки). Такі результати у *T*-інваріанті вказують на проблему з умовою виходу з циклу та проблему невідновлення початкової розмітки. Ранг матриці інцидентності $\text{rang}(W_{nz}) = \min(|T_{nz}|, |P_{nz}|) = 4$ дорівнює числу вершин місць, але при покритті нульовими елементами *P*-інваріанта. Це вказує на проблему, пов'язану з перетворенням розміток у мережі, включаючи відновлення початкової розмітки, тобто порушується вказане вище правило 1.

Таким чином, аналіз PN-моделей ПС на основі матричного опису та методу інваріантів дозволяє перевірити основні динамічні властивості побудованих моделей.

Висновки

Запропонований метод аналізу PN-моделей має практичне застосування у поетапному аналізі моделі ПС [2], оскільки актуальною задачею є виявлення ознак непередбачуваного функціонування компонентів програмних систем та перевірки проектних рішень, які усувають виявлені недоліки функціонування. На початковому етапі з його допомогою проводиться аналіз компонентів програмної системи на наявність явних та прихованих конфліктних ситуацій. Можливість поетапного аналізу моделі ПС обумовлена розповсюдженням компонентно-орієнтованої технології розробки ПС та досить широкими обмеженнями на кількість елементів PN-моделі. Обмеження у практичному застосуванні розглянутого методу аналізу накладає кількість елементів кожної PN-моделі, що може складати до 40 елементів для кожної множини вершин мережі (вершин місць та вершин переходів). Запропонований метод застосовувався як самостійно, так і у складі комбінованого підходу до імітаційного моделювання систем з паралелізмом. Метод аналізу прихованих помилок у моделях ПС дозволяє не тільки провести аналітичне дослідження побудованих PN-моделей, а й визначити конкретну локалізацію ділянок, які можуть містити приховані помилки чи ознаки непередбачуваного функціонування, а також перевірити запропоноване проектне рішення для усунення виявлених помилок.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Соммервилл И.* Инженерия программного обеспечения. 6-е изд. Пер. с англ. М.: Изд. дом «Вильямс», 2002, 624.
2. *Suprunenko O.* Combined approach architecture development to simulation modeling of systems with parallelism // *Eastern-European Journal of Enterprise Technologies*, 2021. 4(4(112)), pp. 74—82. Doi: <https://doi.org/10.15587/1729-4061.2021.239212>
3. *Стоян В.А.* Моделирование та ідентифікація динаміки систем з розподіленими параметрами. К.: Київський університет, 2008, 201 с.
4. *Супруненко О.О., Онищенко Б.О., Гребенович Ю.Є.* Аналітичний підхід при дослідженні властивостей графової моделі програмної системи // *Праці міжнародної науково-практичної конференції «Математичне моделювання процесів в економіці та управлінні проектами і програмами» (ММП-2020)*. Харків-Коблево, Україна, 14-18.09.2020. Харків: ХНУРЕ., 2020, с. 110-113.
5. *Брауде Э.* Технология разработки программного обеспечения. СПб: Питер, 2004, 655 с.
6. *Карпов Ю.Г.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб: БХВ-Петербург, 2005, 400 с.

7. Андреев А.М., Можаров Г.П., Сюзев В.В. Многопроцессорные вычислительные системы. Теоретический анализ, математические модели и применение. М.: Изд-во МГТУ им. Н.Э. Баумана, 2011, 332 с.
8. Application and Theory of Petri Nets and Concurrency. 38th International Conference, PETRI NETS 2017, Zaragoza, Spain, June 25–30, 2017. Ed. Will van der Aalst, Eike Best. Springer International Publishing AG. Available at: <<https://link.springer.com/book/10.1007/978-3-319-57861-3>>. [Accessed: December 25, 2021].
9. Van der Aalst, W.M.P. Pi calculus versus Petri nets: let us eat humble pie rather than further inflate the pi-hype. 2003. Available at: <tmitwww.tm.tue.nl/staff/wvdaalst/pihype.pdf>. [Accessed August 29, 2021].
10. Супруненко О.О. Парадигми імітаційного моделювання при дослідженні складних систем з паралелізмом // Східноєвропейський журнал передових технологій, 2013, 5/4 (65), с. 63—67.
11. Верников Г. Стандарты моделирования IDEF и ABC. 2009. [online]. Режим доступа: <<http://www.cfin.ru/vernikov/idef/idef0>>. [Дата звернення: 18 грудня 2014].
12. Сердюк А.Г. Компонентная модель распределённых информационных систем. Материалы 8-й Международной научно-методической конференции «Новые информационные технологии в университетском образовании: Компьютерные сети, системы телекоммуникаций и инструментальные средства». Новосибирск: СГУПС, 2001, 39 с.
13. Воеводин В.В. Математические модели и методы в параллельных процессах. М.: Наука. Гл. ред. физ.-мат. лит., 1986, 296 с.
14. Карпов Ю.Г. Model Checking. Верификация параллельных и распределённых программных систем. СПб.: БХВ-Петербург, 2010, 560 с.
15. Угольницкий Г.А. Имитационные и оптимизационные модели сложных систем с учётом их структуры // Управление большими системами. Сб. трудов, 30-1, 2010, с. 799—816. Режим доступа: <<https://cyberleninka.ru/article/n/imitatsionnye-i-optimizatsionnye-modeli-slozhnyh-sistem-s-uchetom-ih-struktury>>. [Дата звернення: 12 грудня 2021].
16. Лаврищева К.М., Слабоспицька О.О. Підхід до побудови об'єктно-компонентної моделі сімейства програмних продуктів // Проблеми програмування, 2013, 4, с. 14—24.
17. Череди́ченко Ю.О., Гонтар М.Ю., Иващенко О.В., Вовк М.А. Аналіз компонентно-орієнтованих методів розробки програмного забезпечення для електронного бізнесу // Вісник Вінницького політехнічного інституту, 2018, 2, с. 80—88.
18. Краліна Г.С., Смілій Е.Р. Компонентно-орієнтоване програмування сучасний підхід до розробки складних програмних систем // Тези ІХ міжнародної науково-практичної конференції «Інформаційно-комп'ютерні технології 2018». Житомир: Вид. Житомирського держ. ун-ту «Житомирська політехніка», 2018, 27 с.
19. Kuzmuk V.V., Suprunenko O.A. The means for the description of information flows in dynamic models of medical hardware-software systems // Theoretical and Applied Science, 2014, 7(15), с. 11—18.
20. Нестеренко Б.Б., Новотарський, М.А. Формальні засоби моделювання паралельних процесів та систем // Праці Інституту математики НАН України, 2012, Vol. 90, с. 334.
21. Murata T. Petri Nets: Properties, Analysis and Applications // Proceedings of the IEEE, 1989, 4(77), pp. 541—574.
22. Reisig W., Rozenberg G. Informal Introduction to Petri Nets. Lectures on Petri Nets I // Basic Models. Advances in Petri Nets. Series: Lecture Notes in Computer Science, 1998, Vol. 1491, pp. 1—12.

23. Крывый С.Л. О вычислении минимального множества инвариантов сети Петри // Искусственный интеллект, 2001, 3, с. 199—206.
24. Питерсон, Дж. Теория сетей Петри и моделирование систем. Пер. с англ. М.: Мир, 1984, 264 с.

Отримано 15.04.22

REFERENCES

1. Sommerwill, I. (2002), Software engineering, Williams House.
2. Suprunenko, O. (2021), “Combined approach architecture development to simulation modeling of systems with parallelism”, *Eastern-European Journal of Enterprise Technologies*, 4(4(112)), pp. 74-82. DOI: <https://doi.org/10.15587/1729-4061.2021.239212>.
3. Stojan, V.A. (2008), *Modelyuvannya ta identyfikatsiya dynamiky system z rozpodilenyimi parametramy* [Modeling and identification of dynamics of systems with distributed parameters], Kyivskyy universytet, Kyiv, Ukraine.
4. Suprunenko, O.O., Onyshchenko, B.O. and Grebenovich, J.E. (2020), “Analytical approach in the study of the properties of the graph model of a software system. Proceedings of the international scientific-practical conference”, *Pratsi mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Matematychni modelyuvannya protsesiv v ekonomitsi ta upravlinni proektamy i prohramamy» (MMP-2020)* [Mathematical modeling of processes in economics and project and program management (MMP-2020)], Kharkiv, KNURE, September 14-18, 2020, pp. 110-113.
5. Braude, E. (2004), *Tekhnologiya razrabotki programnogo obespecheniya* [Software engineering: Object-Oriented Perspective (Software development technology)], Piter, St. Petersburg, Russia.
6. Karpov, J.G. (2005), *Imitatsionnoye modelirovaniye sistem. Vvedeniye v modelirovaniye s AnyLogic 5* [Simulation modeling of systems. Introduction to modeling with AnyLogic 5], BHV-Peterburg, St. Petersburg, Russia.
7. Andreev, A.M., Mozarov, G.P. and Sjuzev, V.V. (2011), *Mnogoprotsessornyye vychislitel'nyye sistemy. Teoreticheskiy analiz, matematicheskiye modeli i primeneniye* [Multiprocessor computing systems. Theoretical analysis, mathematical models and applications], Izdatelstvo MGTU im. N.E. Bauman, Moscow, Russia.
8. Van der Aalst, W.M.P. and Best, E. (2017), Application and Theory of Petri Nets and Concurrency, *38th International Conference, PETRI NETS 2017, Zaragoza*, Springer, June 25–30, 2017, available at: <https://link.springer.com/book/10.1007/978-3-319-57861-3> (accessed: December 25, 2021).
9. Van der Aalst, W.M.P. (2003), Pi calculus versus Petri nets: let us eat humble pie rather than further inflate the pi-hype, available at: tmitwww.tm.tue.nl/staff/wvdaalst/pihype.pdf (accessed: August 29, 2021).
10. Suprunenko, O.O. (2013), “Paradigms of simulation modeling in the study of complex systems with parallelism”, *Shkhidnoyevropeyskiy zhurnal peredovykh tekhnolohiy*, Vol. 5, no. 4, pp. 63-67.
11. Vernikov, G. (2009), *Standarty modelirovaniya IDEF i ABC* [Modeling standards IDEF and ABC], available at: <http://www.cfin.ru/vernikov/idef/idef0> (accessed: December 18, 2014).
12. Serdjuk, A.G. (2001), “Component model of distributed information systems”, *Materialy 8-y Mezhnunarodnoy nauchno-metodicheskoy konferentsii «Novyye informatsionnyye tekhnologii v universitetskom obrazovanii: Kompyuternyye seti, sistemy telekommunikatsiy i instrumentalnyye sredstva»* [Materials of the 8th International Scientific and Methodo-

- logical Conference "New Information Technologies in University Education: Computer Networks, Telecommunications Systems and Tools", SGUPS, Novosibirsk, Russia.
13. Voevodin, V.V. (1986), *Matematicheskiye modeli i metody v parallel'nykh protsessakh* [Mathematical models and methods in parallel processes], Nauka, Moscow, Russia.
 14. Karpov, J.G. (2010), *Verifikatsiya parallel'nykh i raspredelennykh programmnykh sistem* [Model Checking. Verification of parallel and distributed software systems], BHV-Petersburg, St. Petersburg, Russia.
 15. Ugolnitskiy, G.A. (2010), "Simulation and optimization models of complex systems, taking into account their structure", *Upravleniye bolshimi sistemami*, Vol. 30, no. 1, pp. 799-816, available at: <https://cyberleninka.ru/article/n/imitatsionnye-i-optimizatsionnye-modeli-slozhnyh-sistem-s-uchetom-ih-struktury> (accessed: December 12, 2021).
 16. Lavrishcheva, K.M. and Slabospizka, O.O. (2013), "An approach to building an object-component model of a family of software products", *Problemy prohramuivannya*, Vol. 4, pp. 14–24.
 17. Cherednichenko, J.O., Hontar, M.Ju., Ivashchenko, O.V. and Vovk, M.A. (2018), "Analysis of component-oriented methods of e-business software development", *Visnyk Vinnytskoho politekhnichnoho instytutu*, Vol. 2, pp. 80-88.
 18. Kralina, G.S. and Smaglyi, E.R. (2018), "Component-oriented programming is a modern approach to the development of complex software systems", *Tezy IX mizhnarodnoyi naukovo-praktychnoyi konferentsiyi «Informatsiyno-kompyuterni tekhnolohiyi 2018»* [Abstracts of the IX International Scientific and Practical Conference «Information and Computer Technologies 2018»], Zhytomyr, ZVydavnytstvo Zhytomyrskoho derzhavnoho universytetu «Zhytomyrska politekhnika».
 19. Kuzmuk, V.V., Suprunenko, O.A. (2014), "The means for the description of information flows in dynamic models of medical hardware-software systems", *Theoretical and Applied Science*, Vol. 7, no. 15, pp. 11-18.
 20. Nesterenko, B.B. and Novotarskiy, M.A. (2012), *Formalni zasoby modelyuvannya parallel'nykh protsesiv ta sistem* [Formal tools for modeling parallel processes and systems], Pratsi Instytutu matematyky NAN Ukrainy, Kyiv, Ukraine.
 21. Murata, T. (1989), "Petri Nets: Properties, Analysis and Applications", *Proceedings of the IEEE*, Vol. 4, no. 77, pp. 541-574.
 22. Reisig, W. and Rozenberg, G. (1998), "Informal Introduction to Petri Nets" *Lectures on Petri Nets I: Basic Models. Advances in Petri Nets. Series: Lecture Notes in Computer Science*, Vol. 1491, pp. 1-12.
 23. Kryvyi, S.L. (2001), "On the calculation of the minimum set of Petri net invariants", *Iskusstvennyy intellekt*, Vol. 3, pp. 199–206.
 24. Peterson, D. (1984), *Teoriya setey Petri i modelirovaniye sistem* [Theory of Petri nets and system modeling], Mir, Moscow, USSR.

Received_15.04.22

O.O. Suprunenko, B.O. Onyshchenko, J.E. Grebenovych

ANALYSIS OF HIDDEN ERRORS IN THE MODELS OF SOFTWARE SYSTEMS BASED ON PETRI NETS

The basic characteristics of the model of the software system which belong to the working capacity and predictability, are described. Analyzed tools were used for the modeling of software systems. As a basis for the tools of modeling and analysis of the dynamic properties of soft-

ware systems, the choice of interpretations and modifications of Petri nets is justified. They allow to follow structural similarity in the model system and also have an unambiguous mathematical description. The analysis of the properties of Petri nets allows to reveal obvious and hidden errors in functioning of PN-model of software system, that is not always possible in the application of simulation modeling. To such properties belongs liveness, boundedness, reachability, preservation, conflictlessness, controllability. The T - and P -invariants, as well as the characteristics of the incidence matrix, allow to detect compliance with these characteristics. The matrix description of the PN-model was carried out with the help of the basic equation of the Petri net, definition identification of invariants and their analysis to reveal dynamic properties of the model. The rule of absence of hidden deadlocks and infinity loops in PN-model is formulated. The detection of hidden errors – deadlock and infinity loop are described. The solution of the deadlock elimination was proposed; its conformity to the dynamic properties was checked. The definition and correction of hidden errors on the example of not fully controlled PN-model is illustrated. The presented method of analysis of PN-models has practical application in step-by-step analysis of the software model. Phasing of the analysis of software model caused by the proliferation of component-oriented technology of software system development and the limiting the number of PN-model elements. It can be applied both independently and as a part of the combined approach to the simulation modeling of systems with parallelism.

Key words: Petri net, PN-model software system, analysis of hidden errors.

СУПРУНЕНКО Оксана Олександрівна, канд. техн. наук., доцент кафедри програмного забезпечення автоматизованих систем Черкаського національного університету ім. Б. Хмельницького. В 1993 р. закінчила Черкаський інженерно-технологічний інститут. Область наукових досліджень — моделювання паралельних процесів на основі мереж Петрі, інструментарій імітаційного моделювання програмних систем з паралелізмом, методи аналізу динамічних характеристик моделей програмних систем з використанням графоаналітичного інструментарію, гнучкі методології та технології розробки програмного забезпечення.

ОНИЩЕНКО Борис Олегович, канд. физ.-мат. наук, доцент кафедри програмного забезпечення автоматизованих систем Черкаського національного університету ім. Б. Хмельницького. В 1999 р. закінчив Черкаський державний університет ім. Б. Хмельницького. Область наукових досліджень — паралельні обчислювальні процеси, розробка паралельних високопродуктивних алгоритмів, дослідження операцій, глобальна оптимізація, стохастичне програмування, розробка методів стохастичної глобальної оптимізації, розробка програмного забезпечення для мобільних платформ.

ГРЕБЕНОВИЧ Юлія Євгенівна, старший викладач кафедри програмного забезпечення автоматизованих систем Черкаського національного університету ім. Б. Хмельницького. В 1995 р. закінчила Черкаський державний педагогічний інститут ім. 300-річчя возз'єднання України з Росією. Область наукових досліджень — апроксимаційні розв'язання інтегро-диференціальних рівнянь нецілого (дробового) порядку за методом кінцевих елементів, технології об'єктного моделювання, компонентно-орієнтовані технології розробки програмного забезпечення.