УДК 004.414.23.056.3:004.77
DOI https://doi.org/10.24144/2616-7700.2025.47(2).253-266

І. В. Рубан[1], В. М. Ткачов[2]

[1] Харківський національний університет радіоелектроніки,
професор кафедри електронних обчислювальних машин,
доктор технічних наук, професор
ihor.ruban@nure.ua
ORCID: https://orcid.org/0000-0002-4738-3286

[2] Харківський національний університет радіоелектроніки,
докторант кафедри електронних обчислювальних машин,
кандидат технічних наук, доцент
vitalii.tkachov@nure.ua
ORCID: https://orcid.org/0000-0002-6524-9937

МОДЕЛЬ ПОРУШЕНЬ, ПОЛІТИК ВІДНОВЛЕННЯ ТА ЛОКАЛІЗАЦІЇ У ПРОСТОРІ СТАНІВ ПРОЦЕСНОГО РІВНЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА МОБІЛЬНІЙ ПЛАТФОРМІ

У роботі запропоновано удосконалену модель процесного рівня з простором станів, що описується чотирма змінними: обсяг черги, часовий запас до дедлайну, валідність даних, дотримання причинно-часового порядку. Усі вимоги об'єднано в інтегральний бар'єрний показник, додатково введено показник локальності як радіус впливу відхилень. Порушення формалізовано як оператори переходів, що дає змогу кількісно оцінювати їхній внесок у деградацію. Розроблено R/L-політику, яка під частковою спостережуваністю обирає дії з гарантованим підвищенням бар'єра і пріоритетом локалізації. Сформульовано критерії коректності та умови здійсненності за заданих бюджетів часу і ресурсів. Реалізовано стенд і виконано експериментальні дослідження для двох профілів навантаження. Результати дослідження підтверджують застосовність і відтворюваність підходу та демонструють скорочення типового часу відновлення і кращу локалізацію відхилень.

Ключові слова: інформаційна система, процес, локалізація, критерій коректності, оператор переходу, живучість.

1. Вступ. Одна із проблем адаптації складних інформаційних систем до умов функціонування на базі мобільних платформ (БПЛА та інших рухомих платформ) полягає у забезпеченні безперебійного виконання критично важливих процесів в умовах переривчастої зв'язності, часткової спостережуваності та жорстких ресурсно-часових обмежень [1]. В таких умовах на процесному рівні інформаційної системи можуть виникати збурення у функціональності елементів системи, спричинені короткими локальними відхиленнями (необґрунтовані запити на збільшення ресурсів на тлі їхнього дефіциту, неадекватна конкуренція за ресурси, неконсистентні переходи станів процесів тощо), здатні каскадувати у макронаслідки та знижувати живучість інформаційної системи на мобільній платформі.

Аналіз класичних комплексних підходів до забезпечення живучості інформаційних систем показує задовільні результати на макрорівні інформаційної системи [2–3], але є непридатними для опису, локалізації та гарантованого відновлення процесів у разі деструкцій саме в просторі станів процесного рівня. Це пояснюється тим, що класичні підходи є ресурсоємними, інерційними і не враховують тонку причинно-часову структуру деструкцій мікрорівня інформаційної системи, що характерно в умовах її функціонування на мобільній платформі.

Мета роботи полягає у розробці моделі порушень та політик відновлення і локалізації у просторі станів процесного рівня інформаційної системи на мобільній платформі, що пов'язує деструкції з інваріантами причинно-часової консистентності та забезпечує існування політик, які повертають стан процесу в допустиму область з подальшою локалізацією негативних наслідків.

2. Постановка завдання. Для досягнення мети сформульовано наступні завдання, які необхідно вирішити в рамках цього дослідження:

– формалізувати простір станів процесного рівня інформаційної системи та визначити допустиму область і інваріанти причинно-часової консистентності;
– ввести поняття класів деструкцій (порушень) і формалізувати їх як оператори переходів станів;
– розробити політики відновлення та локалізації у вигляді керованих переходів і механізмів контролю за виконанням процесів в інформаційній системі;
– сформулювати критерії коректності та умови здійсненності за ресурсно-часових обмежень і часткової спостережуваності;
– дослідити застосовність моделі на стендовій інформаційній системі на мобільній платформі.

3. Аналіз досліджень і публікацій. У статті [4] запропоновано розподілену формальну модель самовідновних поведінок із верифікацією в просторі станів і статистичною перевіркою кількісних властивостей, що знімає архітектурну залежність специфікації політик відновлення. Втім, в роботі [4] не враховується специфіка мобільних платформ, яка більш жорстко визначає часові обмеження локалізації/відновлення.

У статті [5] показані результати досліджень щодо інтеграції підходів планування відновлення елементів інформаційних систем після збоїв і забезпечення безперервності виконання процесів, окреслює значимість отриманих результатів у стійкості множини інформаційних систем (мегасистеми). Автор систематизує різні стратегії — від резервного копіювання й відновлення до хмарних сценаріїв відновлення. Однак організаційно-технологічний рівень дослідження лишає поза уваги питання формування інваріантів та формування темпоральних меж локалізації/відновлення.

Стаття [6] присвячена питанням структурної живучості та надійності комп'ютерних мереж, які можна розглядати як цілісна система. Автори подають правила й формули для послідовних та паралельних з'єднань, еквівалентних перетворень, а також вводять показник живучості мережі у вигляді зваженої суми за станами (із ймовірностями їх настання), демонструючи числові приклади. Отримані мережні метрики можна розглядати як зовнішні умови для моделі, яка пропонується у даній роботі. Водночас процеси, описані у [6] лишаються на макрорівні мережної структури і не формалізують порушення процесно рівня з точки зору подання їх у вигляді операторів переходів у простори станів з інваріантами причинно-часової узгодженості.

За результатами аналізу досліджень і публікацій [4–6], варто зазначити, що вони частково дають формалізації порушень у просторі станів, не встановлюючи жорстких меж часу відновлення/локалізації та умов локальності впливу за переривчастої зв'язності й часткової спостережуваності.

4. Виклад основного матеріалу.

4.1. Формалізація простору станів процесного рівня та інваріантів причинно-часової консистентності. Моделювання процесного рівня інформаційної системи здійснюється у дискретному часі $t \in \mathbb{Z}^{\geq}$ з фіксованим кроком моделювання. Нехай у інформаційній системі одночасно виконується N процесів з ідентифікаторами i, $i = 1, 2, ..., N$. Стан i-го процесу на кожному із кроків t задається як:

$$x_i(t) = (q_i(t), \xi_i(t), \psi_i(t), \zeta_i(t)), \tag{1}$$

де:

– $q_i(t) \in \mathbb{Z}^{\geq}$ — одиниці буферизації (черги, кількість незавершених квантів обробки даних);
– $\xi_i(t) \in \mathbb{R}$ — часовий запас функціональної здатності процесу;
– $\psi_i(t) \in \mathbb{R}^{\geq}$ — актуальність процесних даних;
– $\zeta_i(t) \in \{0, 1\}$ — показник, який приймає значення «1», якщо дотримано причинно-часовий порядок подій для цього процесу. У іншому випадку було виявлено порушення, наприклад, коли порушено послідовність подій процесу.

Тоді вектор стану процесного рівня задається як поєднання станів усіх процесів:

$$s(t) = (x_1(t), \ldots, x_N(t)) \in S \subseteq \left(\mathbb{Z}^{\geq} \times \mathbb{R} \times \mathbb{R}^{\geq} \times \{0, 1\}\right)^N, \tag{2}$$

де:

– S – простір станів процесного рівня інформаційної системи.

Далі для кожного i-го процесу необхідно зафіксувати параметри допусків:

– $B_i' \in \mathbb{Z}^{\geq}$ — параметр граничної черги, який задає кількість одиниць роботи процесу, який можна накопичити без його деградації;
– $\Delta_i' \in \mathbb{R}^{\geq}$ — параметр максимально допустимого порогу актуальності даних (у розумінні втрати валідності або застарілості даних);
– поріг часового відрізку функціональної здатності процесу у контексті показника ξ_i. При значенні $\xi_i(t) \geq 0$інформаційна система вважає, що поріг не досягнуто.

Параметри допуску можуть задаватися з вимог предметної області на етапах розробки інформаційної системи.

Причинно-часова коректність на процесному рівні фіксується наступними інваріантами, що мають виконуватися для кожного i-го процесу та задаються як:

– черговий інваріант $I_q^{(i)}$ — необхідний для встановлення обмежень навантаження процесу:

$$0 \leq q_i(t) \leq B_i'; \tag{3}$$

– темпоральний інваріант $I_\xi^{(i)}$ — необхідний для перевірки виконання умови часових обмежень:

$$\xi_i(t) \geq 0; \tag{4}$$

– інваріант актуальності даних $I_\psi^{(i)}$ — необхідний для валідації даних за ознакою втрати їх актуальності у процесі виконання процесу:

$$\psi_i(t) \le \Delta'_t; \tag{5}$$

– порядковий інваріант $I_\zeta^{(i)}$ — необхідний для процесного локального причинно-часовий порядку:

$$\zeta_i(t) = 1. \tag{6}$$

Допустиму область процесного рівня, виходячи із (2)–(6) можна задати у вигляді деякої множини A:

$$A = \left\{ s \in S : \bigwedge_{i=1}^{N} \left(I_q^{(i)} \wedge I_\xi^{(i)} \wedge I_\psi^{(i)} \wedge I_\zeta^{(i)} \right) \right\}. \tag{7}$$

З (7) очевидно, що $s(t) \in A$ тоді і тільки, коли всі процеси інформаційної системи одночасно задовольняють свої чотири інваріанти (3)–(6).

Щоб уніфікувати аналіз і синтез політик, для i-го процесу доцільно ввести відстані до порушення заданого інваріанту (3)–(6):

$$\begin{aligned}
z_q^{(i)}(t) &:= B'_i - q_i(t),\\
z_\xi^{(i)}(t) &:= \xi_i(t),\\
z_\psi^{(i)}(t) &:= \Delta'_t - \psi_i(t),\\
z_\zeta^{(i)}(t) &:= \begin{cases} +1, & \zeta_i(t) = 1,\\ -1, & \zeta_i(t) = 0. \end{cases}
\end{aligned}$$

Тобто, $z < 0$ означає, що спостерігається порушення.

Локальний бар'єр (межа) i-го процесу можна задати як:

$$B_i(s(t)) := \min\left\{ z_q^{(i)}, z_\xi^{(i)}, z_\psi^{(i)}, z_\zeta^{(i)} \right\}, \tag{8}$$

а системний бар'єр всього процесного рівня інформаційної системи:

$$B(s(t)) := \min_{1 \le i \le N} B'(s(t)). \tag{9}$$

Тоді$s(t) \in A \Leftrightarrow B(s(t)) \ge 0$.

В масштабах інформаційної системи бар'єр $B(s)$ можна використовувати у якості узагальненого індикатора — чим він більший, тим віддаленою буде інформаційна система від будь-якого порогу втрати функціональності на процесному рівні. Відповідно, $B(s)$ може інтерпретуватися як цільова величина для вибору управляючих дій відновлення/локалізації процесів, тобто, дії підбираються так, щоб збільшити $B(s)$ і повернути стан у A.

4.2. Класи порушень і їх формалізація як операторів переходів. З урахуванням (1), (2), порушення класу c з параметрами інтенсивності/ жорсткості θ (прирости черг, швидкість втрати валідності даними («старіння» даних) тощо), тривалістю τ (кількість умовних процесних тактів, коли спостерігається порушення) та цільовою множиною процесів $\mathrm{I} \in \{1, \ldots, N\}$ (сукупність індексів пошкоджених процесів) можна змоделювати оператором переходів як:

$$D_{\theta,\tau,\mathrm{I}}^{(c)} : S \to S, \quad s(t+1) = D_{\theta,\tau,\mathrm{I}}^{(c)}\left(s(t)\right), \tag{10}$$

який протягом τ локально модифікує компоненти (1) (тільки для $i \in \mathrm{I}$, у разі потреби, і для їхніх безпосередніх сусідів у процесних залежностях. Після завершення тривалості τ дія порушення зникає.

Щоб задати «сусідство» в рамках процесного рівня інформаційної системи, спочатку треба ввести відношення залежностей $R \in \{1, \ldots, N\}^2$. Це відношення можна порівняти з організаційною схемою продюсерів (процеси, які є генерують нову інформації, залежності) та консюмерів (процеси, які обробляють інформацію та підпорядковуються залежностям). Для $i \in$ Іозначимо:

$$\Omega(i) = \{j : (i,j) \in R \quad 01 > \quad (i,j) \in R\}.$$

Порушення вражає ядро I і, навіть, окіл першого порядку $\bigcup_{i \in \mathrm{I}} \Omega(i)$ (наприклад, збій процесу-продюсера призведе до зростання черг даних у процесу-консюмера). Якщо уявити це у вигляді процесного графу інформаційної системи, то радіус впливу буде 1.

Доцільно ввести поняття бібліотеки базових класів порушень процесів, яка буде включати типові класи c з простими, але чіткими правилами оновлення компонент x_i (1) за один так порушення τ. Далі всі міжтактові події процесу інтерпретуються як такі, що додані до номінальної динаміки цього процесу. Тоді, у контексті $i \in$ I, вводимо наступні позначення:

– $\delta_q \geq 0$ — приріст до черги;
– $\delta_\xi \geq 0$ — зменшення запасу часового вікна (швидкість «виїдання» ξ);
– $\delta_\psi \geq 0$ — приріст віку даних («старіння» даних).

Відтак маємо наступні класи порушень:

– клас пікових навантажень на процес. Порушення у класі включають раптові напливи субзадач та, як наслідок, ресурсну конкуренцію, що має відображення суто на процесних змінних ($q_i \leftarrow q_i + \delta_q(\theta), \quad \xi_i \leftarrow \xi_i - \delta_\xi(\theta)$);
– клас тимчасових джиттерів. Порушення у класі включають локальне уповільнення виконання процесу ($\xi_i \leftarrow \xi_i - \delta_\xi(\theta)$);
– клас псевдовиключень взаємодії процесів. З темпоральної точки зору, недоступність взаємодії означає, що черга не розвантажується, а дані «старіють», тобто перестають бути валідними ($q_i \leftarrow q_i + \delta_q(\theta), \quad \psi_i \leftarrow \psi_i + \delta_\psi(\theta)$);
– клас порушень порядку дій. До класу входять порушення характеру причинно-часової невідповідності ($\zeta_i \leftarrow 0$);
– клас порушень актуальності даних. До класу входять порушення, які виникають тоді, коли спостережувані дані не оновлюються вчасно, відбувається старіння даних ($\psi_i \leftarrow \psi_i + \delta_\psi(\theta)$);
– клас дедлок-порушень. До класу входять порушення, які виникають тоді, коли виникають короткі ступори виконання процесу (дедлок), що «під'їдає» запас часу на виконання процесу та створює локальні згустки підвищених потреб у ресурсах ($q_i \leftarrow q_i + \delta_q(\theta), \quad \xi_i \leftarrow \xi_i - \delta_\xi(\theta)$).

Поширення на сусідні процеси (10) описується таким чином: для $j \in \Omega(i)$ дозволяються згладжені впливи (за малими значеннями коефіцієнтів, які можна ввести як $\alpha_q, \alpha_\xi, \alpha_\psi \in [0,1)$), що відображає процесну локальність, так як збій стадії виконання процесу впливає на суміжні стадії всього процесу:

$$q_j \leftarrow q_j + \alpha_q \delta_q, \quad \psi_j \leftarrow \psi_j + \alpha_\psi \delta_\psi, \quad \xi_j \leftarrow \xi_j - \alpha_\xi \delta_\xi.$$

Якщо на одному такті τ активні кілька класів $c \in C$, то вони мають застосовуватися послідовно до тих самих x_i та агрегуватися за найгіршим із них, коли:

прирости до q, ψ — адитивні, зменшення ξ — адитивне, ζ стає 0, якщо щонайменше одне порушення його перевело в це значення. Таким чином, формально, для кожного компоненту береться сума дельт, а для ζ — логічне об'єднання зі скиданням в нуль, якщо трапилося порушення порядку дій.

Враховуючи (7)–(9), для кожного класу порушень необхідно визначити мінімальний очікуваний спад локального бар'єра (8) за такт $\beta_\tau^{(A)}(\theta) \geq 0$ (наприклад, для класу пікових навантажень на процес спад визначається як $\min\{\delta_q, \delta_\xi\}$, для класу порушень актуальності даних — δ_ψ, а для класу порушень порядку дій — різкий перехід $z_\zeta^{(i)} : (+1 \to -1)$). Тоді:

$$B_i\left(D^{(c)}(s)\right) \leq B_i(s) - \beta_\tau^{(A)}(\theta), \quad B\left(D^{(c)}(s)\right) \leq B(s) - \beta_{\min}^{(A)}(\theta), \tag{11}$$

де:

– $\beta_{\min}^{(A)}$ — мінімум по процесах, уражених у τ-такт.

Це (11) дає прямий зв'язок між класом порушення та зменшенням запасу до інваріантів, з чим і будуть конкурувати політики відновлення процесу.

Процес порушень триває τ тактів (епізод порушень). Протягом кожного такту діють $\delta_q, \delta_\xi, \delta_\psi$ та/або скидання ζ. Після τ тактів оператор $D_{\theta,\tau,\mathrm{I}}^{(c)}$ (10) припиняє вплив (далі інформаційна система відпрацьовує за номінальними правилами). Якщо протягом епізоду $B(s) \to B(s) < 0$, то відбувається локальне порушення інваріантів (вихід із допустимої області A). І тут завдання політик відновлення — терміново повернути $B(s) \to B(s) \geq 0$.

В реальній інформаційній системі на мобільній платформі апріорі неможливо бачити точний стан (1) кожного процесу на кожному такті, так як інформація про стани процесів може містити пропуски, тому доводиться реально працювати з наближеними вимірюваннями станів процесів. Вектор спостережень (моніторингові агенти, лічильники тощо) $y(t) \in \mathbb{R}^p$:

$$y(t) = h\left(s(t)\right) + \eta(t), \quad \eta(t) \in \mathrm{E}, \tag{12}$$

де:

– $h(\cdot)$ – відображення доступних елементів (2);
– $\eta(t)$ – похибка вимірювання (з введеним обмеженням $\|\eta(t)\| \leq \delta_y$ для деякої відомої константи $\delta_y \geq 0$.

Частина елементів (12) може бути відсутня (пропуски). Для цього доцільно ввести поняття маски доступності $M(t) \in \{0,1\}^p$.

$M(t) = 1$, якщо пропусків немає; $M(t) = 0$, якщо пропуски є. Далі, коли здійснюється порівняння станів процесів, недоступні компоненти деактивуються покомпонентним множенням (операція Адамара [7]) $M(t) \odot (\cdot)$.

Замість однієї точки стану (2) вводиться множина можливих станів:

$$\Xi(t) = \left\{s \in S : M(t) \odot \left(h(s) - y(t)\right) \in [-\delta_y, \delta_y], \;\; s \text{ узгоджується з усіма } \Xi(t-1)\right\}.$$

$\Xi(t)$ включає усі стани, які могли породити спостереження в межах похибки δ_y, з урахуванням припустимої динаміки процесного рівня інформаційної системи (номінальний крок + можливі порушення, описані в бібліотеці базових класів порушень процесів).

На практиці, можно зберігати інтервали для кожної процесної змінної:

$$q_i \in \left[\underline{q}_i, \bar{q}_i\right], \quad \xi_i \in \left[\underline{\xi}_i, \bar{\xi}_i\right], \quad \psi_i \in \left[\underline{\psi}_i, \bar{\psi}_i\right], \quad \zeta_i(t) \in \{0,1\} \text{ або невідомо.}$$

Інтервали $(\underline{\cdot}, \bar{\cdot})$ необхідно оновлювати за наступним алгоритмом: спочатку прогнозуємо за номінальною динамікою та «найгіршим» впливом активних порушень, а потім корегуємо (12) з допусками δ_y та маскою доступності $M(t)$.

Далі, враховуючи (7)–(9), в умовах невизначеності, аби не допустити помилкового рішення про досягнення режиму відновлення, працюємо з інфіумоцінкою бар'єра $\underline{B}(t) := \inf\limits_{s\in\Xi(t)} B(s)$, яку просто обчислити через нижні запаси (7):

$$\begin{aligned}
\underline{z}_q^{(i)} &= B_i' - \bar{q}_i, \\
\underline{z}_\xi^{(i)} &= \underline{\xi}_i, \\
\underline{z}_\psi^{(i)} &= \Delta_t' - \bar{\psi}_i, \\
\underline{z}_\zeta^{(i)} &= \begin{cases} +1, & \zeta_i = 1, \\ -1, & \zeta_i = 0\ 01 > = 52V4 ><> . \end{cases}
\end{aligned}$$

Відповідно до (8): $\underline{B}(t) = \min_i \min\left\{\underline{z}_q^{(i)}, \underline{z}_\xi^{(i)}, \underline{z}_\psi^{(i)}, \underline{z}_\zeta^{(i)}\right\}$,якщо:

– $\underline{B}(t) \geq 0$, то точно всі інваріанти виконані не зважаючи на невизначеність;
– $\underline{B}(t) \in [-\varepsilon, 0)$, то процес у околі відновлення;
– $\underline{B}(t) < -\varepsilon$, то є місце гарантованому порушенню, тоді потрібно застосувати найсильніші дії відносно процесу, на кшталт, ізоляції.

Коли і параметри порушення точно невідомі, то замість $\beta_\tau^{(A)}(\theta)$можна використати допустимий діапазон Θ і визначити найгірший очікуваний спад бар'єру за такт (робастний підхід): $\beta_{robust}^{(A)}(\theta) = \min\limits_{\theta\in\Theta} \beta_\tau^{(A)}(\theta)$.

Додаємо ще «ціну невизначеності спостережень» η_{view}, яка лінійно пов'язана з δ_y та частотою пропусків, і переходимо до сумарного найгіршого просідання за тракт:

$$\beta' := \beta_{robust}^{(A)} + \eta_{view}. \tag{13}$$

Отримане визначення і є величиною, з якою має у подальшому конкурувати відновлювальна дія процесу.

Небезпека невизначеності породжує ймовірність неправильного вибору дії. Для виходу із ситуації пропонується наступне рішення. Нехай U — множина допустимих процесних дій, а D — оператор процесного переходу (10) під дією $u \in U$та реалізацією події середовища впливу на процес σ, причому її конкретика тут неважлива, — важливо лише, що вона не погіршує оцінки понад (13). Тоді застосовуємо таку $u(t)$, що максимізує найгірший приріст бар'єра:

$$u'(t) \in \arg\max_{u\in U} \inf_{s\in\Xi(t)} \left(B(D(s,\sigma,u) - B(s)\right),$$

та забезпечуємо існування гарантованого приросту бар'єру α з

$$\inf_{s\in\Xi(t)} \left(B(D(s,\sigma,u') - B(s)\right) \geq \alpha \quad V \quad \alpha > \beta'.$$

Це означає, що навіть у найгіршому стані процесу приріст бар'єра від дії u' буде більший, ніж найгірше «просідання» від порушень і невизначеності, тож

траєкторія виконання процесу гарантовано прямує назад у допустиму область A.

4.3. Політика відновлення та локалізації на процесному рівні. Задачу відновлення процесу в інформаційній системі можна формалізувати як якнайшвидше підняття бар'єру $B(s)$ до $\varepsilon^{\uparrow} > 0$, тобто повернення його в потрібну субобласть (R-задача). Задачу локалізації процесу в інформаційній системі можна формалізувати як стримання поширення відхилень (порушень) між процесами (L-задача).

Пропонується наступний каталог процесних дій (дії представлено таким чином, аби ефект від їх застосування одразу був означений):

Таблиця 1.

Каталог процесних дій

Легенда	**Зміст**	**Ефект**
retime(i)	Перепланування (збільшення ξ_i) (наприклад, перерозподіл у черзі планувальника; дрібні переноси у вікнах доступності)	$\xi_i \uparrow \Rightarrow z_{\xi}^{(i)} \uparrow$
clean(i)	Очистка черги (наприклад, видалення/злиття некритичних елементів, пріоритезація критичних)	$q_i \downarrow \Rightarrow z_q^{(i)} \uparrow$
rollback(i)	Локальний відкат кроку, що порушив порядок	$\zeta_i := 1, \quad \psi_i \downarrow \Rightarrow z_{\zeta}^{(i)} \uparrow$
fence(i)	Утворення примусового бар'єру перед наступним тактом (наприклад, увімкнення додаткової перевірки послідовності)	$\zeta_i := 1 \Rightarrow z_{\zeta}^{(i)} \uparrow$
throttle(i)	Стримування запуску (наприклад, тимчасове обмеження нових екземплярів процесу)	$q_i \searrow \Rightarrow z_q^{(i)} \uparrow$
degrade(i)	Контрольована деградація якості (наприклад, пропуск некритичних фаз процесу, толерантність до дефіциту ресурсів).	$q_i \downarrow, \xi_i \uparrow$
isolate(i)	Локальна ізоляція у графі процесних залежностей (наприклад, тимчасове розімкнення процесних потоків від/до сусідніх)	$\rho \downarrow$ (антикаскад)

Для того, аби підняти бар'єр та стиснути радіус поширення порушень, необхідно абстрагуватися від того, що на такті τ відомий точний стан процесу. Тут доцільно оперувати саме $\Xi(t)$. Тому виконується оцінка найгіршого можливого бар'єру через нижні межі запасів $\underline{B}(t) = \inf\limits_{s \in \Xi(t)} B(s)$.

Далі оцінюємо для кожної дії u найгірший приріст бар'єру:

$$\Upsilon_{\min}(u) := \inf_{s \in \Xi(t)} \left(B(D(s,u) - B(s)) \right),$$

Важливо теж відстежувати і значення радіусу впливу ρ за процесним графом залежностей.

Відтак формулюємо правило вибору дії:

– якщо наявний радіус впливу більше порогового значення, спершу необхідно виконати локалізацію, тобто обрати дію isolate/throttle на елементах інформаційної системи, які є джерелами відхилень, доки наявний радіус не зменшиться до порогового значення;

– у іншому випадку — виконуємо відновлення, тобто обираємо дію з найбільшим гарантованим підняттям бар'єра навіть у найгіршому сумісному стані. А у разі рівності пріоритетів — виконати дію rollback/fence (за наявності порушень порядку), далі — clean/retime, і на сам кінець — degrade.

Ідея правила — спершу зупини «розповзання» (L-частина), далі максимально швидко повернутись у нормальну субобласть A (R-частина).

Так як із (13) маємо «найгірший спад» від порушень та спостережень, то політика вмикає відновлення лише тоді, коли для обраної дії u' виконується:

$$\Upsilon_{\min}(u') \geq \alpha > \beta', \tag{14}$$

тобто підйом бар'єре (у гіршому випадку) перевищує гірше «просідання».

Координацію дій між кількома процесами у спрощеному вигляді можна подати через жадібні алгоритми [8]. Спочатку визначається множина критичних процесів інформаційної системи C'', потім оцінюється $\Delta^i_{\min}(u)$ для локальних дій, далі визначені дії виконуються для одного/двох найкритичніших процесів (щоб не перевантажити підсистему організації процесів інформаційної системи), решту — відкласти на наступну ітерацію.

У рамках одного такту можна побудувати наступний сценарій розвитку подій:

– здійснюється моніторинг за процесами → будуються інтервали → формуються $\Xi(t)$ та $\underline{B}(t)$;
– визначається ρ та застосовується правило вибору дії;
– здійснюється перехід до наступного кроку, цикл повторюється, поки $B(s) \geq \varepsilon^{\uparrow}$ не стабілізується.

Таким чином, маємо R/L-логіку для процесного рівня інформаційної системи, яка працює у ролі «щита», який не дає каскаду розповзатися і цілеспрямовано піднімає мінімальний запас інваріантів. Саме така комбінація робить відновлення процесів локально-швидким, що критично для інформаційної системи на мобільній платформі.

4.4. Критерії коректності та умови здійсненності за ресурсно-часових обмежень і часткової спостережуваності. В рамках цього дослідження пропонується введення наступних критеріїв коректності:

1. Безпековий критерій «*Safety Criterion*». Суть критерію полягає у тому, що навіть з урахуванням невизначеності вимірювань інформаційна система зберігає всі процесні інваріанти. Тобто, якщо у деякий квант часу t_0 на такті τ маємо $\underline{B}(t_0) \geq 0$, то політика на основі R/L-логіки не допустить виходів з допустимої області:$\underline{B}(t) \geq 0, \quad \forall t \geq t_0$.
2. Темпоральний критерій відновлення «*Recovery Criterion*». Суть критерію полягає у тому, що гарантований «позитивний дрейф» бар'єра забезпечує повернення в субобласть $^{\uparrow}$ за скінченне число тактів τ. Тобто, якщо початкові заходи з відновлення лежать у $\underline{B}(t_0) \geq -\varepsilon$ і у кожному кроці виконується (14) як:

$$\sum_{u \in U_t} \Delta_{\min}(u) \geq \alpha > \beta', \tag{15}$$

то існує верхня межа часу відновлення (повернення) $T_{recovery}$ до субобласті $\uparrow$:

$$T_{recovery} \leq \left\lceil \frac{\varepsilon^{\uparrow} - \underline{B}(t_0)}{\alpha - \beta'} \right\rceil. \tag{16}$$

3. Критерій локалізації «*Localization Criterion*». Суть критерію полягає у тому, щоб унеможливити безсконтрольний ріс каскаду порушень шляхом локалізації області впливу порушення до прийнятного радіуса. Тобто, якщо у тактах τ, коли $\rho(t) > \rho'$, політика застосовує хоча б одну «ізолюючу» дію (типу *isolate* або *throttle*) з ефективністю $\chi \geq 1$, то за не більш ніж:

$$T_{localization} \leq \left\lceil \frac{\rho(t_0) - \rho'}{\chi} \right\rceil \tag{17}$$

кроків матимемо $\rho(t) \leq \rho'$.

4. Критерій дальніх радіусів «Long-Range Criterion». Так як дія відновлення процесу – локальне явище, то критерієм забезпечується локалізація цієї дії без впливу на інші процеси в околі процесів, які не зазнали деструктивних впливів. Тобто, для процесів поза околом джерела відхилення (порушення, на процесному графі), дії з U_t не повинні зменшувати їхні локальні запаси.

Щоб вищенаведені критерії виконувалися в умовах функціонування інформаційної системи на мобільній платформі, треба, щоб запропоновані раніше дії вміщалися в бюджет (ресурсні можливості) і давали потрібний «підйом» бар'єра на кожному кроці. Для цього вводяться наступні умови:

1. Обмеження обчислювального бюджету. Виходимо з того, що кожна дія u описує свої потреби у вигляді обчислювального бюджету $c_{computing}(u)$ та затребуваного часу виконання $c_{time}(u)$. Відповідно, на кроці t такту τ доступні бюджети $C_{computing}(t)$ та $\Delta(t)$. Тоді умова формулюється як:

$$\sum_{u \in U_t} c_{computing}(u) \leq C_{computing}(t), \quad \max_{u \in U_t} c_{time}(u) \leq \Delta(t).$$

2. Доступність процесних вікон. Деякі дії (таблиця 1) мають передумови (наприклад, *rollback* потребує підтверджених даних). Позначимо це булевим предикатом $\wp(u,t)$. Тоді умова формулюється як: усі дії в U_t мають $\wp(u,t) = \text{true}$.
3. Вимога до позитивного дрейфу. З огляду на те, що умови 1-2 мають забезпечити (15), то сукупний гарантований приріст бар'єра за крок має перевищувати найгірший спад від порушень (з урахуванням невизначеності спостережень).
4. Контроль кількості одночасних дій над одним процесом. В умовах обмеженості ресурсів інформаційної системи на мобільній платформі є місце обмеженню кількості дій за такт. Відповідно, умова полягає у тому, аби брати топ процесів з мінімальним B_i і на кожному із них застосовувати дію з найбільшим $\Delta_{\min}$ протягом виконання умов 1-3. Таким чином, це можна інтерпретувати як жадібну координацію, що дає потрібний дрейф без складних методик оптимізації.

Далі враховуємо обмеження часткової спостережуваності. На підставі (13) беремо обмеження похибок вимірюваного стану процесу як δ_y та частку пропусків як $\phi \in [0,1)$ для «ціни невизначеності»: $\eta_{view} = \eta(\delta_y, \phi)$.

Тобто, чим більше похибка та кількість пропусків, тим більше η_{view}. Для забезпечення коректної роботи достатньо:

– забезпечити:

$$\eta_{view} \leq \eta_{\max}, \text{ при } \eta_{\max} < \alpha \quad 4; O \quad \alpha > \beta';$$

– забезпечити період оновлення моніторингових даних у діапазоні до $T_{\max}$, який не дозволяє інтервалам для q, ξ, ψ розповзтися так, щоб $\underline{B}$ впав нижче $-\varepsilon$ без відповідної реакції;
– забезпечити компенсацію ζ , яка отримує невідомі значення (та трактує їх як потенційні порушення) шляхом збільшення ваги дій fence або rollback (пріоритетно).

Тобто, якщо моніторингові заходи не ідеальні з точки зору похибок і при цьому не продукують велику кількість похибок («шум»), то точності нижньої оцінки $\underline{B}$ достатньо, щоб політика підняла її у субобласть $^{\uparrow}$.

Отже, узагальнюючи вищенаведене можна сформувати підсумковий блок достатніх умов коректності та здійсненності. Нехай у момент t_0 виконується $\underline{B}(t_0) \geq -\varepsilon$; нехай для всіх $t \geq t_0$ виконуються умови 1-4 та вищенаведені забезпечуючі дії, а на кожному кроці існує набір дій U_t з урахуванням (15). Тоді, виходячи з формулювання критеріїв коректності, маємо наступні твердження:

– для *Localization Criterion*: $\rho(t)$ не перевищує ρ' після (але не більше ніж) кількість кроків, визначену у (17);
– для *Recovery Criterion*: існує така кількість тактів (16), після чого $\underline{B}(t_0) \geq \varepsilon^{\uparrow}$ (входження в $A^{\uparrow}$);
– для *Safety Criterion*: $\underline{B}(t_0) \geq 0$ для всіх $t \geq t_0$.

На основі цих тверджень можна зробити висновок про нерівність на кожному кроці:

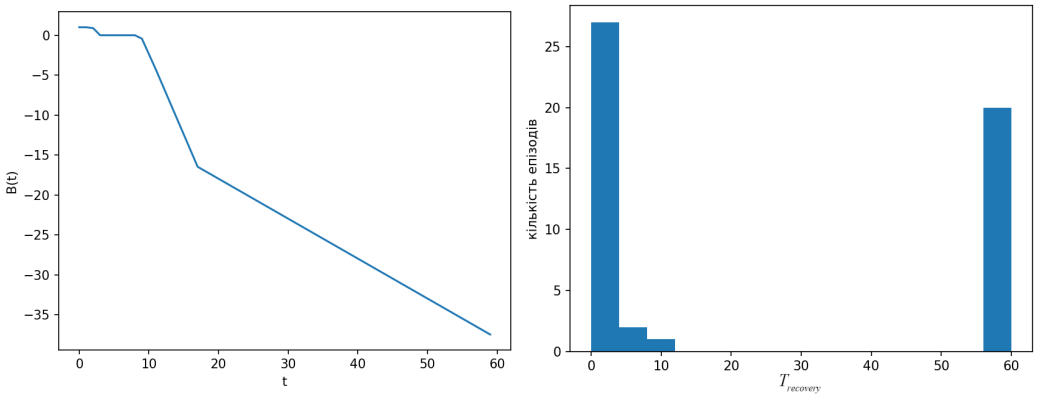
$$\underline{B}(t+1) \geq \underline{B}(t) + \left(\sum_{u \in U_t} \Delta_{\min}(u) \right) - \beta' \geq \underline{B}(t) + (\alpha - \beta').$$

Акумулюючи позитивний дрейф $(\alpha - \beta') > 0$ до рівня $\varepsilon^{\uparrow}$ отримуємо оцінку на $T_{recovery}$.Локалізація випливає зі зменшення ρ на χ за крок при $\rho > \rho'$.

Таким чином, поняття коректності вводиться як збереження / повернення процесних інваріантів і обмеження «розповзання» відхилень, а поняття здійсненності — як наявність на кожному кроці маленького набору нересурсозатратних дій, сукупний гарантований ефект яких більше β' за наявних ресурсів і якості спостережень (моніторингу).

4.5. Експериментальні дослідження застосовності моделі на стендовій інформаційній системі на мобільній платформі. Експеримент виконано у 50 епізодах на Windows 10, Python 3.7.9. Вихідний код розміщено на [9]. Для кожного епізоду було прологовано траєкторію бар'єра $B(t)$, час відновлення $T_{recovery}$, максимум радіусу впливу $\rho_{\max}$ і середнє значення B.

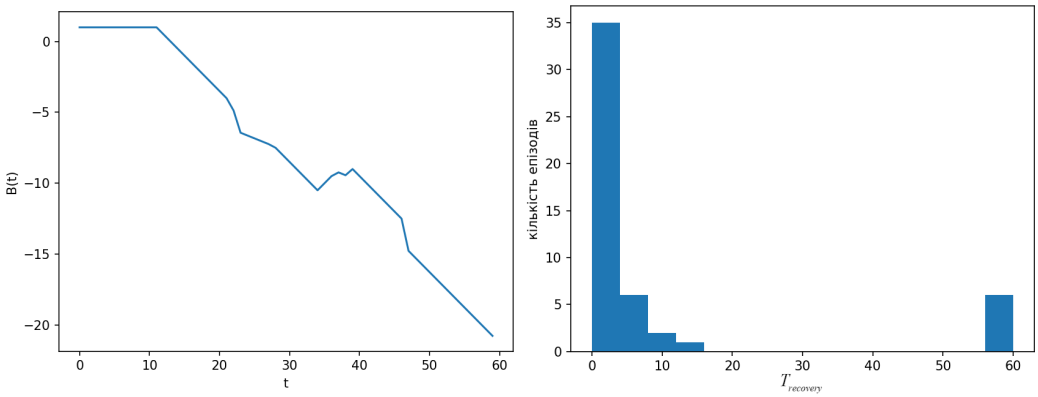
Для стресового профілю $T_{recovery} = 3$ кроки, 95 перцентиль = 60 (верхня межа горизонту), $\rho_{\max} = 3.28$, середній $B = -43.18$; епізодів, що повністю лишилися у A – не зафіксовано (0.0). Це відповідає графікам (рис. 1): швидкий вихід $B(t)$нижче нуля та завал $T_{recovery}$на рівні 60 кроків.



а) б)

Рис. 1. Візуалізація епізоду стресового профілю: а) бар'єр за часом; б) розподіл часу оновлення.

Для налаштованого профілю $T_{recovery} = 1$ крок, 95 перцентиль = 60, $\rho_{\max} =$ $= 2.36$, середній $B = -19.10$; епізодів, що весь час були у A, залишається нульовою. На графіках (рис. 2) видно більш «плаский» початок $B(t)$ та щільний кластер $T_{recovery}$ у зоні 1–5 кроків, хоча вир на 60 кроках зберігається.



а) б)

Рис. 2. Візуалізація епізоду налаштованого профілю: а) бар'єр за часом; б) розподіл часу оновлення.

Обидва набори профілів підтверджують відтворюваність метрик і застосовність моделі. Всі результати експериментів доступні на [10]. Перехід між профілями зменшує середній радіус розповсюдження відхилень (з 3.28 до 2.36) та підвищує «запас» інваріантів (середній B з 43.18 до 19.10), а також скорочує типову тривалість відновлення (з 3 до 1 кроку), що узгоджується з критеріями коректності за ресурсно-часових обмежень і часткової спостережуваності: гарантований приріст бар'єра від політики стає більшим від найгіршого просідання у більшій частині епізодів. Разом із тим, наявність «завалів» $T_{recovery} = 60$

свідчить про рідкісні, але важкі сценарії каскадування відмов, які можна цілеспрямовано зменшити ще агресивнішими кроками локалізації (*isolate / throttle*) та підвищенням пріоритету дій, що відновлюють порядок (*rollback / fence*).

5. Висновки та перспективи подальших досліджень. У роботі сформовано формалізацію простору станів процесів інформаційної системи, задано допустиму область та причинно-часові інваріанти, описано класи порушень як оператори переходів і синтезовано R/L-політику для відновлення та локалізації на рівні процесів із урахуванням ресурсно-часових обмежень і часткової спостережуваності. В роботі удосконалено модель процесного рівня інформаційної системи, яка, на відміну від відомих підходів «важких» архітектурних схем, інтегрує різнорідні інваріанти (черга, часовий запас, валідність даних, порядок подій) у єдиний бар'єр $B(s)$ вводить показник локальності ρ та «двошарову» область $A^{\varepsilon}/A^{\uparrow}$ та тим самим забезпечує вибір дій із гарантованим позитивним дрейфом і стримуванням каскадів при малих обчислювальних непрямих витратах. Стендові експерименти на мобільній платформі підтвердили застосовність і відтворюваність метрик – перехід від стресового профілю до налаштованого зменшує типовий час відновлення (за 3→1 крок), знижує середній радіус впливу та підвищує середній рівень бар'єра, що узгоджується з критеріями коректності та реалізує політику живучості на процесному рівні. Подальші дослідження спрямовуватимуться на звуження множини $\Xi(t)$ для підвищення точності $\underline{B}(t)$ в реальних інформаційних системах на мобільній платформі.

Список використаної літератури

1. Kompaniyets, O. M. (2025). Information technology for adaptive control of a swarm of unmanned aerial vehicles in an antagonistic environment. *Information processing systems,* 4(179), 43–48. https://doi.org/10.30748/soi.2024.179.04 [in Ukrainian].
2. Dodonov, O. G., Putyatin, V. G., Dodonov, V. O., Kutsenko, S. A., Germanyuk, A. P., Izvaryn, I. V., & Kravchuk, K. O. (2024). Technology for ensuring the survivability of territorially distributed information computer systems in a single information space. *Data registration, storage and processing,* 26(1), 121–143. https://doi.org/10.35681/1560-9189.2024.26.1.308659 [in Ukrainian].
3. Kolisnyk, M., Piskachova, I., & Kharchenko, V. (2024). Reliability Models of Diverse Majority-redundant Systems with Threshold Adaptation. In: *2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT).* IEEE, 1–7. https://doi.org/10.1109/dessert65323.2024.11122217
4. Hafaiedh, I. B., & Slimane, M. B. (2022). A distributed formal-based model for self-healing behaviors in autonomous systems: from failure detection to self-recovery. *The Journal of Supercomputing,* 78, 18725–18753. https://doi.org/10.1007/s11227-022-04614-0
5. Derick Musundi Kesa (2023). Ensuring resilience: Integrating IT disaster recovery planning and business continuity for sustainable information technology operations. *World Journal of Advanced Research and Reviews,* 18(3), 970–992. https://doi.org/10.30574/wjarr.2023.18.3.1166
6. Jawaddi, S. N. A., Johari, M. H., & Ismail, A. (2022). A review of microservices autoscaling with formal verification perspective. *Software: Practice and Experience,* 52(11), 2476–2495. https://doi.org/10.1002/spe.3135
7. Gehér, G. P., McLauchlan, C., Campbell, E. T., Moylett, A. E., & Crawford, O. (2024). Error-corrected Hadamard gate simulated at the circuit level. *Quantum,* 8, 1394. https://doi.org/10.22331/q-2024-07-02-1394
8. García, A. (2025). Greedy algorithms: a review and open problems. *Journal of Inequalities and Applications,* 2025(11). https://doi.org/10.1186/s13660-025-03254-1
9. Tkachov, V. (2025). GitHub - tikey/rlshield-sim-profiles. *GitHub.* Retrieved from https://github.com/tikey/rlshield-sim-profiles
10. Tkachov, V. (2025). RLShield-sim: experimental results (hard and tuned profiles, 50×2 epi-

sodes). *Zenodo.* https://doi.org/10.5281/zenodo.17180073 [in Ukrainian].

Ruban I. V., Tkachov V. M. A Model of Disruptions, Recovery and Localization Policies within the Process-Level State Space of a Mobile-Platform Information System.

The paper proposes an enhanced process-level model whose state space is described by four variables: queue backlog, time slack to the deadline, data validity, and adherence to causal–temporal order. All requirements are unified in a single barrier-type indicator, and an additional locality metric is introduced as the radius of deviation propagation. Disruptions are formalized as state-transition operators, enabling quantitative assessment of their contribution to degradation. An R/L policy (Recovery/Localization) is developed that, under partial observability, selects actions with guaranteed increase of the barrier while prioritizing localization. Correctness criteria and feasibility conditions are formulated under fixed time and resource budgets. A testbed implementation is provided and experimental studies are conducted for two workload profiles. The results confirm the applicability and reproducibility of the approach and demonstrate a reduction in typical recovery time together with improved localization of deviations.

Keywords: information system, process level, localization, correctness criteria; state-transition operator, survivability.

Одержано 24.09.2025