

DOI: <https://doi.org/10.15276/ict.03.2026.57>

УДК 004.65:004.94

Відновлення траєкторій платіжних зобов'язань за журналами подій цифрової сервісної платформи

Руденко Олександр Володимирович¹⁾

ORCID: <https://orcid.org/0009-0004-2944-4040>; alexo.rudenko@gmail.com

Арсірій Олена Олександрівна¹⁾

ORCID: <https://orcid.org/0000-0001-8130-9613>; e.arsiriy@gmail.com. Scopus Author ID: 54419480900

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

У статті розв'язано науково-прикладне завдання відновлення траєкторій платіжних зобов'язань клієнтів цифрових сервісних платформ за операційними журналами та агрегованими записами. Актуальність дослідження зумовлена тим, що початкові дані платформи, як правило, не утворюють готових послідовностей станів: відомості про виникнення зобов'язання, платіжні запити, повторні звернення, строки, оплати та завершення спостереження можуть зберігатися в різних полях, мати різний рівень часової деталізації та не завжди відповідати семантиці технічних назв атрибутів. Запропоновано процедуру відновлення, яка охоплює ідентифікацію записів одного платіжного зобов'язання, семантичну перевірку часових полів, перетворення записів на події, їх впорядкування, визначення станів і допустимих переходів, обробку агрегованих повторних звернень, розмежування підтвердженого кінцевого результату та цензурування, а також контроль якості сформованої траєкторії. Принциповою вимогою є недопущення штучного створення часових міток для подій, момент настання яких не спостерігався безпосередньо. Експериментальну перевірку виконано на вибірці з 15566 платіжних зобов'язань цифрової сервісної платформи. Для 5580 записів були доступні кількість і часові межі платіжних звернень. Сумарна кількість звернень за полем `sent_count` становила 7320; для 6390 з них, або 87,3 %, вдалося відновити точні часові мітки першого, єдиного або останнього звернення, тоді як 930 проміжних повторів залишилися часово невизначеними. Для 1979 зобов'язань зафіксовано точний момент оплати. Перевірка показала, що поле `created_at` не може автоматично трактуватися як початок траєкторії: у 5564 із 5580 випадків із відомим першим зверненням його значення було не ранішим за `min_sent_at`. Два випадки містили оплату раніше останнього зафіксованого звернення та були позначені як конфліктні. Наукова новизна полягає у формалізації відновлення траєкторій платіжних зобов'язань із частково агрегованих журналів із явним збереженням ступеня часової визначеності подій і відокремленням цензурованих спостережень від підтвердженого завершення без оплати. Практичне значення результатів полягає у формуванні узгодженого набору траєкторій, придатного для багатостанового моделювання, аналізу часу до події та подальшого динамічного прогнозування.

Ключові слова: журнали подій, траєкторії процесу, якість даних, платіжні зобов'язання, цифрові сервісні платформи, часові мітки, цензуровані дані, багатостанове моделювання.

Для цитування: Руденко О. В., Арсірій О. О. «Відновлення траєкторій платіжних зобов'язань за журналами подій цифрової сервісної платформи». *Інформатика. Культура. Техніка.* 2026; Том 3 № 2 (4): 745–755. DOI: <https://doi.org/10.15276/ict.03.2026.57>

Актуальність. Цифрові сервісні платформи поєднують оформлення послуги, фіксацію додаткових нарахунків, платіжні операції та комунікацію з клієнтом у єдиному інформаційному середовищі. У межах такого середовища окреме платіжне зобов'язання може виникнути як під час основного розрахунку, так і пізніше – внаслідок додаткового нарахування, штрафу, компенсації витрат або іншої асинхронної події. Подальша обробка зобов'язання розвивається у часі та може включати первинний платіжний запит, повторні звернення, очікування, оплату або завершення внутрішнього циклу без підтвердженого позитивного результату. Тому для аналізу такого процесу недостатньо мати лише підсумковий статус або одну строку табличного набору даних.

Ключовим джерелом інформації про фактичний перебіг процесу є операційні журнали цифрової платформи. У загальному випадку вони містять записи різних підсистем: реєстру платіжних вимог, модуля інвойсингу, сервісу повідомлень, платіжного шлюзу та внутрішніх засобів завершення обробки. У процесному аналізі такі записи можуть бути перетворені на журнал подій, де кожна подія повинна належати конкретному екземпляру процесу, мати тип і часову мітку [2], [3]. Однак на практиці початкові дані рідко відповідають цій структурі без додаткової підготовки. Частина подій може бути агрегована, окремі часові мітки можуть бути відсутні, а технічні поля мати семантику, відмінну від тієї, що впливає з їх назв [4], [5].

Для платіжних зобов'язань ця проблема є особливо суттєвою. Якщо один клієнт має декілька незалежних вимог щодо оплати, об'єднання даних лише за ідентифікатором клієнта

змішує різні траєкторії. Якщо поле `created_at` без перевірки використовується як момент виникнення зобов'язання, а воно фактично відображає створення технічного запису або пізніший етап обробки, усі розрахунки тривалості стають зміщеними. Якщо агрегований показник `sent_count` перетворюється на довільно розставлені у часі події, у даних з'являються факти, яких платформа насправді не спостерігала. Нарешті, відсутність оплати на момент експорту не означає автоматичного завершення без оплати; для незавершеного процесу коректною інтерпретацією є цензурування.

Тому перед побудовою багатостанової моделі життєвого циклу, оцінюванням інтенсивностей переходів або формуванням динамічного прогнозного профілю необхідно отримати відтворювані траєкторії окремих платіжних зобов'язань. У попередній роботі було показано доцільність переходу від статичного прогнозування кінцевого статусу до багатостанового подання та динамічного прогнозування [7]. Проте якість такого прогнозування безпосередньо залежить від того, наскільки коректно з початкових журналів відновлено послідовність подій і часові інтервали. Саме задача формального відновлення траєкторій із неповних та частково агрегованих даних є предметом цієї роботи.

Аналіз останніх досліджень і публікацій. Задача перетворення операційних записів на структурований журнал подій є однією з базових для інтелектуального аналізу процесів. У роботі [2] журнал подій розглядається як інформаційна основа, що пов'язує фактичні дані інформаційної системи з методами виявлення, перевірки відповідності та вдосконалення процесних моделей. У роботі [3] якість, повнота та коректна прив'язка подій до окремих екземплярів процесу визначаються як необхідні передумови достовірного аналізу. Отже, проблема полягає не лише в наявності достатнього обсягу даних, а й у правильному визначенні предметного змісту кожного запису та його належності до конкретної траєкторії.

У роботі [4] систематизовано типові дефекти журналів подій, до яких віднесено відсутні або помилкові часові мітки, неправильну прив'язку подій до екземплярів процесу, дублювання записів, неоднакову деталізацію та неповноту реєстрації. Показано, що спосіб усунення таких дефектів безпосередньо впливає на результати подальшого аналізу та побудову процесної моделі. У роботі [5] серед найбільш суттєвих проблем якості журналів також виділено помилки часової прив'язки, пропущені події та надмірну складність початкових даних.

Окремий напрям досліджень пов'язаний із відновленням неповних журналів подій. У роботі [6] запропоновано підхід до відновлення пропущених подій і часових міток із використанням процесної моделі та ймовірнісного оцінювання. Такий підхід доцільний, коли необхідно реконструювати приховані або відсутні елементи процесу. Водночас для подальшого статистичного прогнозування платіжних зобов'язань принциповим є розмежування безпосередньо спостережуваних фактів і значень, відновлених за допомогою моделі. Якщо точний момент події невідомий, його довільне відновлення може створити штучну точність і вплинути на подальше оцінювання часових характеристик процесу.

У дослідженнях платіжної поведінки переважно використовуються вже підготовлені табличні вибірки. У роботах [8], [9], [10] для прогнозування статусу оплати застосовуються характеристики клієнта, платіжного зобов'язання, попередніх платежів і зафіксованих взаємодій. Для оцінювання часу до оплати використовуються методи аналізу часу до події, які дають змогу враховувати незавершені спостереження [11]. Багатостанові моделі дозволяють описувати послідовні переходи між проміжними та кінцевими станами та враховувати конкуруючі результати [12], [13]. Проте застосування таких методів потребує попереднього формування коректної подієво-часової структури даних.

Таким чином, у наявних дослідженнях достатньо повно розглянуто загальні проблеми якості журналів подій, інтелектуального аналізу процесів, прогнозування платежів та аналізу часу до події.

Водночас для цифрових сервісних платформ залишається актуальною задача відновлення траєкторії окремого платіжного зобов'язання за операційними даними, у яких

частина звернень може бути представлена агрегованими лічильниками та часовими межами, а окремі технічні поля не можуть без додаткової перевірки трактуватися як предметні події. Це зумовлює необхідність розроблення процедури, яка забезпечує формування траєкторії без штучного відновлення невідомих подій, зберігає інформацію про ступінь часової визначеності та розмежовує незавершене спостереження і підтверджений кінцевий результат.

Метою дослідження є розроблення та експериментальна перевірка процедури відновлення траєкторій платіжних зобов'язань за операційними журналами і частково агрегованими записами цифрової сервісної платформи. Для досягнення мети необхідно: формалізувати одиницю відновлення та структуру вихідних подій; визначити правила ідентифікації записів одного зобов'язання; встановити порядок семантичної перевірки часових полів; сформувати правила перетворення точних і агрегованих відомостей на події та стани; визначити обробку кінцевих результатів і цензурування; перевірити процедуру на реальному наборі даних платформи та оцінити частку точно і частково відновлюваної подієвої інформації.

Вихідне подання платіжного зобов'язання та його даних. Одиницею відновлення прийнято окреме платіжне зобов'язання, а не клієнта, замовлення або сукупність усіх фінансових взаємодій. Це принципово, оскільки один клієнт може мати декілька зобов'язань з різними підставами виникнення, сумами, строками та історіями звернень. Аналогічно одне замовлення може породжувати декілька окремих нарахувань.

Тому належність записів до спільної траєкторії повинна підтверджуватися ідентифікатором зобов'язання або предметно обґрунтованим правилом зіставлення:

$$O = \{o_1, o_2, \dots, o_Q\}. \quad (1)$$

Для i -го зобов'язання сукупність початкових записів позначимо B_i . Якщо інформація надходить з Q джерел, вона може бути подана як об'єднання записів реєстру зобов'язань, журналу платіжних запитів, журналу комунікацій, таблиці оплат та інших джерел:

$$B_i = \cup B_i^{(q)}, \quad q = 1, \dots, Q. \quad (2)$$

Початковий запис не ототожнюється з подією автоматично. Одна подія може породити декілька технічних записів, а один агрегований запис може містити відомості про декілька подій. Тому після предметної перевірки подію доцільно подати кортежем:

$$e_{ik} = (\tau_{ik}, y_{ik}, v_{ik}, c_{ik}), \quad (3)$$

де τ_{ik} – час події або підтверджена часова межа; y_{ik} – тип події; v_{ik} – додаткові атрибути; c_{ik} – ознака часової визначеності. У найпростішому випадку c_{ik} може набувати значень *exact*, *interval* або *aggregate*. Значення *exact* означає, що час події спостерігався безпосередньо; *interval* – що відомий лише інтервал її можливого настання; *aggregate* – що відома кількість або факт подій без індивідуальних часових міток. Таке маркування не змінює предметну модель життєвого циклу, але запобігає використанню неповної інформації як точної.

Основні групи вихідних даних, що використовуються для ідентифікації платіжного зобов'язання, формування його подієвої історії та відновлення часової структури траєкторії, узагальнено в Таблиці 1.

Процедура відновлення траєкторії. Запропонована процедура передбачає послідовне перетворення початкових операційних записів цифрової сервісної платформи на структуровану траєкторію окремого платіжного зобов'язання. Процедура не передбачає імовірнісного домислювання невідомих подій – якщо наявних даних недостатньо для однозначного визначення події або її часу, відповідна невизначеність зберігається у вихідному представленні.

Відновлення траєкторії виконується у такій послідовності:

1) ідентифікація записів, що належать одному платіжному зобов'язанню;

- 2) семантична перевірка часових полів і приведення часових міток до узгодженої шкали;
- 3) формування множини підтверджених подій та їх упорядкування за часом;
- 4) визначення станів платіжного зобов'язання і переходів між ними відповідно до зафіксованих подій;
- 5) встановлення кінцевого результату або ознаки цензурування для незавершеного спостереження;
- 6) контроль часової та предметної узгодженості сформованої траєкторії й маркування ступеня визначеності її елементів.

Результатом процедури є впорядкована траєкторія платіжного зобов'язання, яка містить лише підтверджені стани, події та часові характеристики і може використовуватися для подальшого статистичного аналізу, оцінювання переходів та прогнозування.

Таблиця 1. Основні групи даних, що використовуються під час відновлення траєкторії

Група даних	Приклади полів	Роль у відновленні
Ідентифікаційні	document_number, order_id, тип нарахування	Віднесення записів до одного платіжного зобов'язання
Характеристики зобов'язання	product_name, product_type, gross_amount, currency	Контекст та перевірка узгодженості записів
Часові поля	payment_due_date, min_sent_at, max_sent_at, paid_at	Формування подій і часових меж
Агреговані поля	sent_count, due_days, pay_due_days	Уточнення кількості подій та часових умов без створення штучних міток
Кінцеві ознаки	paid_status, paid_at, явна подія завершення	Визначення підтвердженого результату або цензурування

Джерело: розроблено авторами

Ідентифікація записів одного зобов'язання. На першому етапі встановлюється відповідність кожного початкового запису конкретному платіжному зобов'язанню. Найнадійнішим є прямий стабільний ідентифікатор. Якщо він відсутній, допускається складене правило зіставлення за номером документа, замовленням, типом нарахування, сумою, валютою та часовою близькістю. Таке правило повинно бути підтверджене предметною семантикою. Збіг лише ідентифікатора клієнта, суми або замовлення не вважається достатнім, оскільки кожний з цих атрибутів може повторюватися для незалежних зобов'язань.

Результат ідентифікації доцільно поділяти на чотири рівні. Прямий рівень використовується за наявності стабільного id зобов'язання. Складений підтверджений рівень – коли зв'язок відновлюється за предметно перевіреною комбінацією полів. Імовірний рівень означає, що існує частковий збіг, але його недостатньо для автоматичного об'єднання. Неідентифіковані записи зберігаються окремо та не включаються до вибірки точних траєкторій. Така консервативна логіка важлива, оскільки помилкове злиття двох зобов'язань створює хибну послідовність подій, яку вже неможливо виправити на пізніших етапах лише статистичними засобами.

Семантична перевірка часових полів. Назва технічного атрибута не використовується як єдине підтвердження його процесного змісту. Для кожного поля перевіряються формат, часовий пояс, діапазон значень, точність, взаємне розташування відносно інших подій та відповідність предметній семантиці. Наприклад, поле created_at може означати створення зобов'язання, створення запису в окремому сервісі, момент реплікації або формування експорту. Тому його використання як початку життєвого циклу допускається лише після перевірки того, що воно систематично передують процесним подіям, які не могли відбутися до виникнення зобов'язання.

Часові мітки приводяться до спільної шкали з урахуванням UTC-зсуву. Перетворення формату не повинно змінювати порядок подій. Для кожної пари подій застосовуються не універсальні, а предметно допустимі логічні співвідношення.

Наприклад, оплата не може передувати виникненню самого зобов'язання, але може відбутися після встановленого строку. Повторне звернення не повинно передувати первинному, однак нагадування після строку оплати є нормальним випадком, а не помилкою. Таким чином, часовий контроль має відрізнити реальну аномалію від допустимої особливості процесу.

Формування та впорядкування подій. Після семантичної перевірки вихідні записи перетворюються на події предметного рівня. Для досліджуваного внутрішнього циклу достатньо розрізнити виникнення або підтвердження зобов'язання, первинний платіжний запит, повторне звернення, оплату, підтверджене завершення внутрішнього циклу без оплати та кінець доступного спостереження. Події впорядковуються за часом. Якщо дві події мають однакову мітку і немає додаткового поля послідовності, довільне встановлення порядку не допускається, вони зберігаються як одночасні або як такі, чия відносна послідовність не визначена.

Особливу увагу необхідно приділяти технічним дублям. Наприклад, одне повідомлення може створювати записи «сформовано», «передано провайдеру» і «доставлено». Якщо для багатостанової моделі предметною подією є саме звернення до клієнта, ці технічні записи не повинні породжувати три повторні етапи. Навпаки, якщо доставка має самостійне значення для прогнозу, вона може залишатися окремою подією в історії, але не обов'язково створювати новий процесний стан. Таке розмежування між станом і подією запобігає надмірному розширенню простору станів.

Відновлення станів і переходів. Після отримання впорядкованої історії формується траєкторія перебування у станах. Для узгодження з багатостановим поданням життєвого циклу використовуються стан формування зобов'язання, первинна внутрішня обробка, k -й етап повторної обробки та два предметно різні кінцеві результати – оплата і підтверджене завершення без оплати.

Траєкторію можна подати як послідовність станів і часових меж:

$$L_i = ((s_i 0, t_i 0^{in}, t_i 0^{out}), \dots, (s_i m, t_i m^{in}, t_i m^{out})). \quad (4)$$

Момент входження до наступного стану визначається підтвердженою подією, а не умовним календарним правилом. Настання строку оплати, наприклад, змінює часовий контекст, але не повинно автоматично створювати новий стан, якщо сама логіка внутрішньої обробки не змінилася.

Подібно, збільшення кількості днів без реакції є важливою ознакою для прогнозу, однак не є подією. Такий підхід дозволяє окремо зберігати процесну структуру та часові характеристики перебування в ній.

Обробка агрегованих повторних звернень. Найбільш суттєва відмінність запропонованої процедури від прямого перетворення таблиці на event log полягає в роботі з агрегованими полями. Якщо запис містить sent_count, min_sent_at і max_sent_at, ці атрибути задають кількість звернень і часові межі, але не завжди дозволяють відновити момент кожного повтору.

За sent_count = 1 значення min_sent_at і max_sent_at повинні збігатися та задають єдину точну подію. За sent_count = 2 обидві межі можуть бути інтерпретовані як точні моменти першого і другого звернення. За sent_count > 2 точними залишаються лише перше та останнє звернення, кількість проміжних подій відома, але їх часові мітки не відновлюються без додаткового журналу.

Приклад подання такої частково визначеної послідовності повторних звернень наведено на Рис. 1.

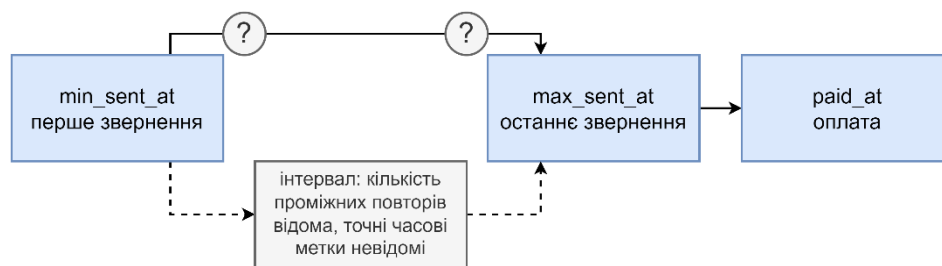


Рис. 1. Подання частково агрегованої послідовності повторних звернень

Джерело: розроблено авторами

Як показано на Рис. 1, за наявності більше двох повторних звернень достовірно визначеними залишаються часові мітки першого та останнього звернення, тоді як моменти проміжних подій залишаються невідомими. Довільний розподіл таких повторів рівномірно між `min_sent_at` і `max_sent_at` є небажаним, оскільки він створює штучні тривалості перебування у станах. У подальшому це може безпосередньо змінити оцінені інтенсивності переходів, `survival`-функції та ймовірності оплати на горизонті. Тому для часткової траєкторії зберігаються дві підтверджені події, кількість невідомих проміжних повторів та інтервал, у якому вони відбулися. Якщо конкретний метод моделювання вимагає точних часових міток, такі епізоди або аналізуються окремо, або використовуються лише до рівня доступної точності.

Правила інтерпретації агрегованих відомостей про повторні звернення залежно від кількості зафіксованих звернень і доступності часових меж наведено в Таблиці 2.

Таблиця 2. Правила інтерпретації агрегованих полів повторних звернень

Умова	Відновлювані події	Рівень визначеності
<code>sent_count</code> відсутній	Точні події звернень не формуються	Інформації недостатньо
<code>sent_count</code> = 1, <code>min</code> = <code>max</code>	Одне звернення у відомий момент	Точна подія
<code>sent_count</code> = 2, <code>min</code> < <code>max</code>	Перше і друге звернення	Дві точні події
<code>sent_count</code> > 2, <code>min</code> < <code>max</code>	Перше та останнє звернення; відома кількість проміжних	Часткова траєкторія
<code>sent_count</code> > 1, <code>min</code> = <code>max</code>	Автоматичне відновлення заборонено	Конфлікт даних

Джерело: розроблено авторами

Кінцеві результати та цензурування. Подія оплати є підтвердженням позитивним результатом лише за наявності надійного `paid_at` або еквівалентної події платіжної системи. Завершення внутрішнього циклу без оплати повинно мати окрему предметну підставу: явний статус закриття, передачу до зовнішнього контуру, списання або іншу подію, яку правила платформи визначають як завершальну. Відсутність `paid_at` у момент формування набору даних не є такою підставою. Якщо кінцевий результат не спостерігався, траєкторія вважається правосторонньо цензурованою на момент зрізу.

Це розмежування критичне для подальшого аналізу часу до події. Якщо всі записи без оплати кодувати як завершення без оплати, у вибірку потрапляють активні зобов'язання, які могли бути оплачені пізніше. У `survival analysis` та багатостанових моделях такі спостереження повинні робити внесок до моменту останнього достовірного спостереження, але не створювати хибний поглинальний перехід [11], [12], [13].

Контроль якості сформованої траєкторії. Після відновлення виконується набір структурних і часових перевірок. Ідентифікатор зобов'язання має бути однозначним; усі

точні події повинні бути впорядковані; кількість відновлених звернень не може перевищувати `sent_count`; поглинальний стан не повинен мати наступних переходів у межах того самого внутрішнього циклу; цензурування не може бути перетворене на завершення без оплати. Додатково контролюється, чи не використана пізніша інформація для формування більш ранньої частини траєкторії.

За результатами контролю траєкторія отримує статус `exact`, `partial` або `conflict`. `Exact` означає, що всі події, потрібні для вибраного рівня моделі, мають підтверджені часові мітки. `Partial` використовується, коли частина послідовності відома лише через кількість або часовий інтервал. `Conflict` означає порушення предметної послідовності, яке неможливо усунути без додаткового джерела. Такий статус не є оцінкою «якості клієнта» або «ризиків оплати», він описує лише надійність відновленої процесної інформації.

Наукова новизна запропонованого підходу полягає не у відновленні «найімовірнішої» прихованої історії, а у формальному розмежуванні того, що спостерігалось точно, що відомо лише агреговано і що взагалі не може бути встановлено з наявних даних. Для платіжних зобов'язань це дозволяє зберегти часову структуру процесу без підміни невідомих фактів штучними подіями та сформувані дані, придатні для подальшого багатостанового прогнозування.

Експериментальна перевірка. Процедура перевірено на наборі даних цифрової сервісної платформи, який містив 15566 записів платіжних зобов'язань. Кожний запис мав унікальний `document_number` і містив характеристики продукту, замовлення, клієнтського профілю, суми, валюти, строку оплати, агреговану інформацію про відправлення платіжних звернень та, за наявності оплати, точний `paid_at`. У вибірці були представлені чотири групи продуктів: `RENT`, `TRAFFIC_TOLLS`, `TRAFFIC_OFFENCES` і `DAMAGE`. Загальну характеристику використаної експериментальної вибірки та повноту основних полів, необхідних для відновлення траєкторій, наведено в Таблиці 3.

Таблиця 3. Загальна характеристика експериментальної вибірки

Показник	Значення
Кількість платіжних зобов'язань	15566
Унікальні <code>document_number</code>	15566 (100 %)
<code>RENT</code>	9421 (60,52 %)
<code>TRAFFIC_TOLLS</code>	4414 (28,36 %)
<code>TRAFFIC_OFFENCES</code>	1059 (6,80 %)
<code>DAMAGE</code>	672 (4,32 %)
Записи з <code>min_sent_at</code> , <code>max_sent_at</code> і <code>sent_count</code>	5580 (35,85 %)
Підтверджені події оплати <code>paid_at</code>	1979 (12,71 %)
Кількість валют	5

Джерело: розроблено авторами

Перша перевірка стосувалася ідентифікації одиниці аналізу. `document_number` був заповнений та унікальний для всіх 15566 записів, тому для експерименту використовувався як прямий ідентифікатор платіжного зобов'язання. Водночас `order_id` не був унікальним: п'ять значень зустрічалися двічі. Це підтверджує, що навіть у межах одного експорту ідентифікатор замовлення не слід автоматично використовувати як ідентифікатор зобов'язання.

Друга перевірка стосувалася повторних звернень. Для 5580 зобов'язань були одночасно доступні `sent_count`, `min_sent_at` і `max_sent_at`. У 4770 випадках `sent_count` дорівнював 1; для всіх них `min_sent_at` і `max_sent_at` збігалися, що дозволило однозначно відновити одну подію. Ще 140 випадків мали `sent_count` = 2 і різні часові межі, тому обидві події були відновлені точно. Для 670 зобов'язань `sent_count` перевищував 2. У цих випадках перше й останнє звернення мали точні мітки, але час проміжних повторів у вихідному наборі відсутній.

Сумарне значення `sent_count` для записів, де воно було наявне, становило 7320 звернень. За правилами процедури 6390 часових міток, або 87,3 % від цієї кількості, були безпосередньо відновлюваними: одна мітка для `sent_count = 1`, дві - для `sent_count = 2`, а також перша і остання мітки для випадків з більшою кількістю повторів. Для 930 проміжних звернень, або 12,7 %, точний час не був відомий. Вони зберігалися як агрегована кількість усередині підтвердженого інтервалу, без генерації синтетичних міток.

Третя перевірка продемонструвала необхідність семантичного контролю часових полів. Для 5580 записів із відомим `min_sent_at` лише у 16 випадках `created_at` був ранішим за перше звернення. У 5564 випадках, або 99,71 %, `created_at` був не ранішим за `min_sent_at`. Отже, у цьому наборі поле `created_at` не може універсально трактуватися як момент виникнення платіжного зобов'язання. Його автоматичне використання як `to` призвело б до траєкторій, у яких перша процесна подія відбувається до формального початку об'єкта. Процедура тому не включала `created_at` до траєкторії як початкову подію без додаткового предметного підтвердження.

Четверта перевірка стосувалася порядку звернень та оплат. Для всіх випадків `sent_count > 1` виконувалося `min_sent_at < max_sent_at`; конфлікту типу «декілька звернень в одну й ту саму агреговану мітку» не виявлено. Серед 1979 оплачених зобов'язань з доступним `max_sent_at` у двох випадках `paid_at` був ранішим за `max_sent_at`. Такі записи не виправлялися автоматично: вони отримали статус `conflict`, оскільки останнє звернення могло бути технічно сформоване після фактичної оплати, належати іншому етапу або відображати особливість агрегації.

П'ята перевірка стосувалася кінцевих результатів. Точний `paid_at` був наявний для 1979 зобов'язань. Для решти 13587 записів у використаному експорті не було окремої підтвердженої події завершення внутрішнього циклу без оплати. Тому процедура не переводила їх автоматично у стан нерезультативного завершення. За відсутності додаткового реєстру закриття ці спостереження мають розглядатися як незавершені або цензуровані на даті зрізу. Цей результат є принципово важливим: проста заміна порожнього `paid_status` класом «не оплачено» змішала б фактичне нерезультативне завершення з випадками, де результат ще не спостерігався.

Зведені результати перевірки відновлюваності подій, часової узгодженості записів та коректності визначення кінцевих результатів наведено в Таблиці 4.

Таблиця 4. Результати контролю відновлюваності та часової узгодженості

Перевірка	Результат	Інтерпретація
Унікальність <code>document_number</code>	15566 / 15566	Пряме групування записів допустиме
<code>sent_count = 1</code> та <code>min_sent_at = max_sent_at</code>	4770 / 4770	Одна точна подія звернення
<code>sent_count = 2</code>	140	Дві точні події за <code>min/max</code>
<code>sent_count > 2</code>	670	Проміжні повтори відновлюються частково
Точно локалізовані звернення	6390 із 7320 (87,3 %)	Можуть використовуватися як точні події
Проміжні повтори без точної мітки	930 із 7320 (12,7 %)	Не повинні штучно розподілятися у часі
<code>created_at</code> раніше першого звернення	16 із 5580 (0,29 %)	<code>created_at</code> не є універсальним початком циклу
<code>paid_at</code> раніше <code>max_sent_at</code>	2 із 1979 (0,10 %)	Конфліктна послідовність, потрібна перевірка
Записи без <code>confirmed paid_at</code>	13587 (87,29 %)	Не ототожнюються із завершенням без оплати

Джерело: розроблено авторами

У результаті експерименту сформовано два рівні вихідних даних. Перший містить точні події із підтвердженими часовими мітками: ідентифіковані звернення та оплати. Другий зберігає часткову інформацію для випадків із `sent_count > 2`: кількість проміжних повторів і часовий інтервал між першою та останньою точними подіями. Такий поділ дозволяє використовувати повністю спостережувані траєкторії для оцінювання часових параметрів і водночас не втрачати часткові спостереження, які можуть бути корисними для агрегованого

аналізу кількості звернень або для моделей, здатних працювати з інтервальною невизначеністю.

Обговорення результатів. Отримані результати показують, що основна складність підготовки траєкторій полягає не у технічному сортуванні записів за часом, а у відновленні предметної семантики. Навіть повністю заповнене поле часу не гарантує, що воно відповідає потрібній події. Приклад `created_at` демонструє це безпосередньо: формально валідна часова мітка виявилася несумісною з роллю універсального початку життєвого циклу. Тому процедура повинна починатися з перевірки семантики, а не з механічного перетворення назви поля на тип події.

Другий важливий результат стосується агрегованих повторів. З одного боку, 87,3 % усіх звернень, кількість яких відома через `sent_count`, мають часові мітки, що можуть бути використані без додаткових припущень. З іншого боку, 12,7 % повторів знаходяться лише між `min_sent_at` і `max_sent_at`. На великій вибірці навіть така частка може суттєво впливати на оцінювання тривалостей. Якщо розміщувати ці події рівномірно або вважати, що всі вони відбулися в один момент, модель отримає штучну структуру часу. Збереження невизначеності є менш зручним з обчислювальної точки зору, але методологічно коректнішим.

Третій результат пов'язаний із цензуруванням. У використаному експорті лише 12,71 % зобов'язань мали підтверджений момент оплати, а для решти не було окремого поля, яке однозначно означало завершення внутрішнього циклу без оплати. У задачі звичайної бінарної класифікації порожнє значення інколи спрощено перетворюють на негативний клас. Для аналізу життєвого циклу це некоректно: відсутність події на момент зрізу є характеристикою спостереження, а не обов'язково кінцевого стану. Відокремлення цензурування дозволяє надалі застосовувати *survival analysis* і багатостанові моделі без зміщення, спричиненого передчасним присвоєнням негативного результату.

Запропонована процедура не замінює *process mining* і не намагається реконструювати повний бізнес-процес платформи. Її задача вужча: сформувати надійну подієво-станову історію конкретного платіжного зобов'язання для подальшого прогностичного аналізу. Саме тому до траєкторії включаються лише ті події, які змінюють стан, уточнюють доступну історію або підтверджують кінцевий результат, тоді як технічні записи можуть залишатися додатковими атрибутами.

Обмеження експерименту визначаються структурою доступного набору даних. У ньому не було індивідуальних часових міток для всіх проміжних повторних звернень і не було окремої підтвердженої ознаки завершення без оплати. Тому робота свідомо не оцінює точність «відновлення» невідомих подій за еталонним журналом: такого еталону в наборі немає. Натомість оцінюється частка інформації, яка може бути відновлена без припущень, та виявляються поля, що потребують предметної валідації. Подальша перевірка може бути виконана на повному `event log`, де кожне звернення має власний ідентифікатор і `timestamp`, що дозволить кількісно порівняти агреговану та повну реконструкцію.

Висновки. У роботі запропоновано процедуру відновлення траєкторій платіжних зобов'язань клієнтів цифрових сервісних платформ за операційними журналами і частково агрегованими записами. Процедура охоплює ідентифікацію окремого зобов'язання, семантичну перевірку часових полів, формування та впорядкування подій, відновлення станів і переходів, обробку агрегованих повторів, розмежування кінцевих результатів і цензурування та контроль якості сформованої траєкторії.

Експеримент на 15566 платіжних зобов'язаннях показав, що механічне використання технічних полів може призводити до некоректної часової структури. Для 99,71 % записів із доступним першим зверненням `created_at` не передувало `min_sent_at`, тому його не можна автоматично використовувати як момент виникнення зобов'язання. Із 7320 звернень, зафіксованих через `sent_count`, 6390 подій мали відновлювані точні часові мітки, а 930 проміжних повторів залишалися часово невизначеними. Такі події були збережені у вигляді агрегованої інформації, а не штучно розподілені між відомими межами.

Для 1979 зобов'язань було підтверджено точний момент оплати. Відсутність paid_at для інших записів не інтерпретувалася як завершення без оплати за відсутності окремої події закриття; такі спостереження повинні оброблятися як незавершені або цензуровані. Запропонований підхід формує інформаційну основу для подальшого багатостанового моделювання, аналізу часу до події та динамічного прогнозування, зберігаючи відмінність між спостережуваними фактами, агрегованими відомостями та невідомими елементами траєкторії.

Конфлікт інтересів: Автори декларують, що не мають конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті

Фінансування: Дослідження проводилося без фінансової підтримки

Доступність даних: Усі дані доступні в цифровій або графічній формі в основному тексті рукопису

Використання засобів штучного інтелекту (ШІ): Автори підтверджують, що не застосовували інструментальні засоби ШІ для написання цієї роботи

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. de Reuver, M., Sørensen, C. & Basole, R. C. “The digital platform: A research agenda”. *Journal of Information Technology*. 2018; 33 (2): 124–135. DOI: <https://doi.org/10.1057/s41265-016-0033-3>.
2. van der Aalst, W. M. P. “Process Mining: Data Science in Action”. 2nd ed. Berlin: Springer, 2016. DOI: <https://doi.org/10.1007/978-3-662-49851-4>.
3. van der Aalst, W. M. P., Adriansyah, A., de Medeiros, A. K. A., et al. “Process Mining Manifesto”. *Business Process Management Workshops. Lecture Notes in Business Information Processing*. 2012; 99: 169–194. DOI: https://doi.org/10.1007/978-3-642-28108-2_19.
4. Suriadi, S., Andrews, R., ter Hofstede, A. H. M. & Wynn, M. T. “Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs”. *Information Systems*. 2017; 64: 132–150. DOI: <https://doi.org/10.1016/j.is.2016.07.011>.
5. Dakic, D., Stefanovic, D., Vuckovic, T., Zizakov, M. & Stevanov, B. “Event Log Data Quality Issues and Solutions”. *Mathematics*. 2023; 11 (13): 2858. DOI: <https://doi.org/10.3390/math11132858>.
6. Rogge-Solti, A., Mans, R. S., van der Aalst, W. M. P. & Weske, M. “Repairing event logs using timed process models”. *On the Move to Meaningful Internet Systems: OTM 2013 Workshops. Lecture Notes in Computer Science*. 2013; 8186: 705–708. DOI: https://doi.org/10.1007/978-3-642-41033-8_89.
7. Rudenko, O. V. «Модель життєвого циклу та метод динамічного прогнозування платіжних зобов'язань цифрових сервісних платформ». *Інформатика. Культура. Техніка*. 2026; 3 (1): 482–492. DOI: <https://doi.org/10.15276/ict.03.2026.37>.
8. Arsirii, O. O., Krantovskyi, I. O., Rudenko, O. V. & Glava, M. G. “Machine learning models for predicting payment status on an online car rental platform”. *Applied Aspects of Information Technology*. 2025; 8 (1): 13–23. DOI: <https://doi.org/10.15276/aait.08.2025.1>.
9. Zeng, S., Melville, P., Lang, C. A., Boier-Martin, I. & Murphy, C. “Using predictive analysis to improve invoice-to-cash collection”. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008: 1043–1050. DOI: <https://doi.org/10.1145/1401890.1402014>.
10. Moore, W. R. & van Vuuren, J. H. “A framework for modelling customer invoice payment predictions”. *Machine Learning with Applications*. 2024; 17: 100578. DOI: <https://doi.org/10.1016/j.mlwa.2024.100578>.
11. Saini, S., Manai, G., van den Boom, W., et al. “Invoice level forecasting with discrete survival methods for effective forecasting of account receivables in supply chain”. *Discover Analytics*. 2024; 2: 5. DOI: <https://doi.org/10.1007/s44257-024-00013-2>.
12. Andersen, P. K. & Keiding, N. “Multi-state models for event history analysis”. *Statistical Methods in Medical Research*. 2002; 11 (2): 91–115. DOI: <https://doi.org/10.1191/0962280202SM276ra>.

13. Putter, H., Fiocco, M. & Geskus, R. B. “Tutorial in biostatistics: Competing risks and multi-state models”. *Statistics in Medicine*. 2007; 26 (11): 2389–2430. DOI: <https://doi.org/10.1002/sim.2712>.

Отримано 26.08.2026

Отримано після перегляду 25.09.2026

Прийнято 08.10.2026

DOI: <https://doi.org/10.15276/ict.03.2026.57>

UDC 004.65:004.94

Reconstruction of payment obligation trajectories from event logs of a digital service platform

Olexandr V. Rudenko¹⁾

ORCID: <https://orcid.org/0009-0004-2944-4040>; alexo.rudenko@gmail.com

Olena O. Arsirii¹⁾

ORCID: <https://orcid.org/0000-0001-8130-9613>; e.arsirii@gmail.com. Scopus Author ID: 54419480900

¹⁾ Odesa Polytechnic National University, 1, Shevchenko Av. Odesa, 65044, Ukraine

ABSTRACT

This paper addresses the scientific and applied problem of reconstructing payment-obligation trajectories on digital service platforms from operational logs and partially aggregated records. Raw platform data do not normally form ready-to-use state sequences: information about obligation creation, payment requests, repeated contacts, due dates, payments, and observation termination may be stored in different fields, have different temporal granularity, and may not match the business meaning suggested by technical attribute names. A reconstruction procedure is proposed that includes identifying records belonging to the same payment obligation, semantic validation of time fields, conversion of source records into events, event ordering, reconstruction of states and admissible transitions, handling of aggregated repeated contacts, separation of confirmed terminal outcomes from censoring, and quality control of the resulting trajectory. A key principle is that timestamps are not synthetically generated for events whose occurrence times were not directly observed. The procedure was evaluated on a dataset of 15,566 payment obligations from a digital service platform. For 5,580 records, the number and temporal boundaries of payment requests were available. The sent_count field represented 7,320 requests in total. Exact timestamps could be reconstructed for 6,390 requests (87.3%), whereas 930 intermediate repeated requests remained temporally unresolved. An exact payment time was available for 1,979 obligations. The analysis also showed that created_at cannot be automatically interpreted as the trajectory start: in 5,564 of 5,580 cases with a known first request, created_at was not earlier than min_sent_at. Two cases contained payment timestamps earlier than the last recorded request and were marked as conflicting rather than automatically corrected. The scientific contribution is the formal reconstruction of payment-obligation trajectories from partially aggregated event data while explicitly preserving the degree of temporal certainty and separating censored observations from confirmed completion without payment. The resulting trajectories can serve as a consistent data basis for multi-state modelling, time-to-event analysis, and subsequent dynamic prediction.

Keywords: Event logs; process trajectories; data quality; payment obligations; digital service platforms; timestamps; censored data; multi-state modelling

ІНФОРМАЦІЯ ПРО АВТОРІВ



Руденко Олександр Володимирович - аспірант, кафедра Інформаційних систем, Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Галузь досліджень: інформаційні технології; штучний інтелект; машинне навчання.

Olexandr V. Rudenko - postgraduate, Department of Information Systems. Odesa Polytechnic National University, 1, Shevchenko Av. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0009-0004-2944-4040>; alexo.rudenko@gmail.com

Research field: Information technology; artificial intelligence; machine learning.



Арсірій Олена Олександрівна - доктор технічних наук, професор, завідувач кафедри Інформаційних систем. Національний університет «Одеська Політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Галузь досліджень: інформаційні технології; штучний інтелект; системи підтримки прийняття рішень; машинне навчання; нейронні мережі.

Olena O. Arsirii - Doctor of Engineering Sciences, Professor, Head of Department of Information Systems. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID: <https://orcid.org/0000-0001-8130-9613>; e.arsirii@gmail.com. Scopus Author ID 54419480900

Research field: Information technology; artificial intelligence; decision support systems; machine learning; neural networks