

УДК 004.65

DOI: 10.32626/2308-5916.2025-27.149-164

В. А. Федорчук*, д-р техн. наук, професор,

О. Є. Коваленко**, д-р техн. наук

*Кам'янець-Подільський національний університет

імені Івана Огієнка, м. Кам'янець-Подільський,

**Національний університет біоресурсів

і природокористування України, м. Київ

БАЗА ДАНИХ ЯК ЗАСІБ ОТРИМАННЯ АДАПТИВНОЇ МОДЕЛІ ПРЕДМЕТНОЇ ОБЛАСТІ ЗІ СЛАБОФОРМАЛІЗОВАНИМИ ОБ'ЄКТАМИ

Зростаюча динамічність сучасних бізнес-середовищ вимагає розробки адаптивних інформаційних систем, особливо для предметних областей зі слабоформалізованими об'єктами. Традиційні статичні моделі даних часто виявляються неефективними для адекватної підтримки таких швидкозмінних умов, що призводить до значних втрат при інформаційному супроводі проєкту через потреби його переробки. Пропонується інноваційний підхід, орієнтований на використання баз даних в якості адаптивних моделей предметної області. Методологія базується на застосуванні баз даних з гнучкою схемою, зокрема NoSQL (документних та графових), які здатні обробляти великі обсяги швидкозмінних, неструктурованих або напівструктурованих даних, пропонуючи значно більшу масштабованість та адаптивність порівняно з реляційними базами даних. Додатково інтегруються принципи адаптивних об'єктних моделей, де класи, атрибути, зв'язки та поведінка представлені як метадані, що інтерпретуються під час виконання, дозволяючи експертам предметної області змінювати модель в реальному часі. Метою дослідження є розробка адаптивної моделі для предметної області «Видавництво університету», яка динамічно коригується відповідно до змін вартості матеріалів та норм їх споживання. Проблема полягає в тому, що традиційне моделювання не може адекватно відобразити притаманний динамізм та слабоформалізовану природу реальних бізнес-процесів, що призводить до дорогих та трудомістких модифікацій системи. Запропонований механізм реалізує неперервний цикл зворотного зв'язку, який автоматично виявляє та коригує розбіжності між розрахунковими та реальними витратами на витратні матеріали. Це є значним відходом від традиційних «жорстких» моделей, перетворюючи базу даних на модель предметної області, як активний, самооптимізуючий компонент адаптивної системи. Такий підхід підвищує гнучкість, точ-

ність та операційну ефективність в управлінні складними бізнес-процесами, що розвиваються, забезпечуючи неперервне уточнення внутрішніх параметрів системи та зменшуючи потребу в ручному втручанні.

Ключові слова: адаптивна модель, слабоформалізовані об'єкти, база даних, гнучка схема, еволюція системи, видавнича справа, NoSQL, предметна область.

Вступ. Сучасні бізнес-середовища характеризуються безпрецедентною складністю та динамічністю, що створює значні виклики для традиційних інформаційних систем. Організації все більше залежать від даних і ефективне управління та отримання достовірної інформації відіграють вирішальну роль в їхньому успіху [1]. Однак, традиційні статичні моделі предметної області часто виявляються недостатніми для адекватної підтримки таких швидкозмінних умов. Вимоги до програмного забезпечення постійно еволюціонують як під час розробки, так і після розгортання, що за оцінками експертів призводить до значних втрат зусиль при розробці і експлуатації проекту (приблизно 35-40%) через зміни вимог та необхідність переробки [2]. Ця ситуація підкреслює нагальну потребу в системах, здатних реагувати на зміни в предметній області з мінімальним втручанням людини. У цьому контексті особливу складність становлять так звані слабоформалізовані об'єкти. Це сутності, чії атрибути, зв'язки або поведінка не є повністю фіксованими або відомими на етапі проектування, або які часто змінюються протягом життєвого циклу системи. Традиційні реляційні бази даних, що дотримуються негнучких, строго визначених схем, стикаються з серйозними труднощами при представленні таких об'єктів [3, 4]. Ця притаманна жорсткість перешкоджає швидкому врахуванню мінливих вимог і змушує розробників витратити значний час на навігацію через велику кількість таблиць, рядків, стовпців та заздалегідь визначених зв'язків. Така невідповідність між статичною природою традиційних баз даних та динамічними вимогами бізнес-процесів безпосередньо призводить до згаданих вище значних витрат та переробок. Це вказує на потребу в системах баз даних, які самі можуть адаптуватися, а не просто виступати пасивним сховищем даних.

Для вирішення цих проблем доцільно скористатись концепцією адаптивних моделей. Адаптивні моделі – це статистичні моделі або моделі машинного навчання, які здатні коригувати свої параметри та поведінку у відповідь на нові дані, навчаючись та вдосконалюючись з часом без необхідності ручного втручання або перенавчання. Їхня здатність навчатися з появою нових даних, а також адаптуватися до зміни ситуації робить їх незамінними в таких сек-

торах, як фінансові послуги, Інтернет речей (IoT), предиктивне обслуговування тощо. На відміну від неадаптивних моделей, які мають фіксовані параметри та структуру, адаптивні моделі демонструють високу гнучкість, неперервне навчання та стійкість до дрейфу концепції, що є вирішальним для динамічних застосувань. Однак, слід зазначити, що адаптивні моделі, як правило, є більш складними та вимагають значних обчислювальних ресурсів через неперервну обробку даних та процеси оновлення.

Виходячи із зазначеного вище, актуальною задачею є розробка та демонстрація працездатності підходу, орієнтованого на застосування баз даних для створення адаптивної моделі предметної області, здатної працювати зі слабоформалізованими об'єктами. Як конкретний приклад застосування обрано предметну область «Видавництво університету» з особливим акцентом на динамічне коригування витрат та норм споживання. Проблема полягає в тому, що традиційне моделювання не може адекватно відобразити притаманний динамізм та слабоформалізовану природу реальних бізнес-процесів, що призводить до дорогих та трудомістких модифікацій системи. Метою дослідження є подолання цієї проблеми шляхом впровадження механізмів, які дозволяють моделі самостійно адаптуватися до змін.

Новизна запропонованого підходу полягає в інтеграції принципів адаптивного моделювання (таких як неперервне навчання та самокорекція) безпосередньо в схему бази даних та рівень управління даними для конкретного, складного бізнес-процесу – видання книг. Зокрема, ключовою інновацією є запропонований механізм зворотного зв'язку, який автоматично коригує розбіжності між розрахунковими та реальними витратами на матеріали та витратні матеріали. Це виходить за рамки простого зберігання даних, перетворюючи базу даних на активний компонент, що бере участь у адаптації моделі. Таке застосування відрізняється від типових застосувань адаптивних моделей (наприклад, предиктивне обслуговування або динамічне ціноутворення) тим, що воно фокусується на управлінні операційними витратами в рамках виробничого процесу, забезпечуючи постійне уточнення внутрішніх параметрів системи.

Для ефективного вирішення проблем, пов'язаних з динамічними предметними областями та слабоформалізованими об'єктами, необхідно глибоко розуміти концепції адаптивного моделювання та гнучких підходів до проектування баз даних.

Концепції адаптивних моделей у базах даних та інформаційних системах. Адаптивні моделі є фундаментальним елементом сучасних інформаційних систем, що дозволяє їм функціонувати в умо-

вах постійних змін. Ці моделі здатні коригувати свої параметри та поведінку у відповідь на надходження нових даних, навчаючись та вдосконалюючись без необхідності ручного втручання або перенавчання. Така здатність до навчання підвищує точність прогнозів з часом та забезпечує стійкість до так званого «дрейфу концепції», коли розподіл даних змінюється з плином часу.

Особливо потужною концепцією в цьому контексті є Адаптивні Об'єктні Моделі (АОМ). АОМ – це системи, які представляють класи, атрибути, зв'язки та поведінку як метадані, що інтерпретуються під час виконання. У таких системах об'єктна модель є частиною самої бази даних і може бути змінена експертами предметної області з негайним (хоча і контрольованим) впливом на поведінку системи, яка її інтерпретує. Це кардинально відрізняється від традиційних об'єктно-орієнтованих підходів, де зміна в бізнесі, зазвичай, вимагає зміни коду, що призводить до створення нової версії програми. Натомість в АОМ, коли є зміна в бізнес-процесі, змінюються лише описи (метадані), і вони негайно відображаються в працюючому додатку. Це забезпечує високий рівень гнучкості та дозволяє системі еволюціонувати разом з появою нової інформації, що є критично важливим для динамічних застосувань.

Підходи до моделювання слабоформалізованих об'єктів та гнучких схем. Для ефективного моделювання слабоформалізованих об'єктів та підтримки динамічних змін у структурі даних традиційні реляційні бази даних часто виявляються неефективними. На противагу цьому, бази даних NoSQL стали основним рішенням для обробки великих обсягів швидко змінюваних, неструктурованих або напівструктурованих даних, забезпечуючи значно більшу масштабованість та адаптивність порівняно з реляційними базами даних SQL.

Гнучкість схеми є ключовою перевагою NoSQL. Наприклад, MongoDB, хоча й іноді вона описується як «безсхемна», насправді забезпечує гнучкість схеми, а не її повну відсутність [3]. Це дозволяє динамічно проєктувати схему, що вміщує різні шаблони та структури, такі як вбудовані документи чи поліморфні шаблони та посилання. Така гнучкість ідеально підходить для стартапів, що розробляють мінімально життєздатні продукти (MVP) роботи зі швидкозмінними бізнес-доменами та створення додатків з динамічними вимогами до даних, наприклад, у сфері штучного інтелекту [3].

Серед різних типів моделей даних NoSQL документні та графові бази даних особливо виділяються своєю здатністю працювати зі слабоформалізованими об'єктами.

Документні бази даних (наприклад, MongoDB) зберігають інформацію як логічні документи, часто у форматі JSON. Вони підтриму-

ють вкладені пари ключ-значення та дозволяють запити до будь-якого атрибута в документі. Легкий, ключово-значущий формат JSON є кращим для його простоти та прямої сумісності з сучасними мовами програмування, такими як JavaScript, і чудово підходить для обробки динамічних або неструктурованих даних.

Графові бази даних використовують модель, засновану на вузлах та ребрах, для представлення взаємопов'язаних даних, оптимізованих для обходу зв'язків між сутностями. Вони є безсхемними або опціонально-схемними, що означає, що вони можуть легко адаптуватися до мінливих вимог до даних шляхом додавання нових типів вузлів, ребер або властивостей без модифікації існуючої структури [5]. Ця гнучкість є ключовою для довгострокового управління даними та розробки додатків.

Важливим аспектом роботи з гнучкими схемами є еволюція схеми. Це процес модифікації схеми бази даних зі збереженням існуючих даних та забезпеченням підтримки їх сумісності із старою схемою. Еволюція схеми дозволяє додавати нові стовпці, модифікувати існуючі, перейменовувати або навіть видаляти стовпці без необхідності вивантаження та перезавантаження існуючих даних. Це життєво важливо для адаптації баз даних до мінливих бізнес-вимог та забезпечення зворотної сумісності додатків [6].

Історично, для досягнення гнучкості в реляційних базах даних використовувалася модель **Entity-Attribute-Value (EAV)**. Вона дозволяє легко додавати або видаляти атрибути, що сприяє загальній архітектурі. Однак, EAV має значні недоліки: вона повільна, ресурсоємна, погано масштабується, а її реалізація для загальних речей є вкрай складною. Крім того, EAV ускладнює забезпечення цілісності даних, таких як зовнішні ключі, унікальність або навіть типи даних, і призводить до значного зниження продуктивності запитів у масштабі, оскільки вимагає великої кількості з'єднань. У багатьох випадках EAV вважається анти-шаблоном у реляційних СУБД, особливо для складних систем, оскільки вона обмінює довгострокову стабільність на короткострокову гнучкість.

Адаптивні об'єктні моделі (АОМ) є ключовим теоретичним фундаментом для побудови систем, які можуть динамічно змінювати свою поведінку. Як вже зазначалося, АОМ – це системи, де об'єктна модель, що цікавить користувача, є частиною бази даних і інтерпретується під час виконання. Це дозволяє експертам предметної області конфігурувати визначення доменної моделі та правила її цілісності зовнішньо, безпосередньо впливаючи на систему, що її інтерпретує. Ця можливість є критично важливою для сценаріїв,

подібних до «Видавництва університету», де бізнес-правила, такі як коригування розрахунку витрат, повинні бути динамічними та адаптуватися до реальних умов. Однак, незважаючи на їхню потужність, АОМ можуть бути заплутаними та складними для розробників через наявність кількох рівнів абстракції, що створює кілька місць, які можуть бути змінені.

Порівняльний аналіз підходів до представлення динамічних та слабоформалізованих даних. Вибір підходу до моделювання даних для динамічних предметних областей зі слабоформалізованими об'єктами вимагає ретельного аналізу компромісів. Традиційні реляційні схеми, хоча й забезпечують високу цілісність даних та чітко визначені зв'язки, є негнучкими та вимагають значних зусиль для адаптації до мінливих вимог [3]. Це призводить до високих витрат та часу на переробку, коли бізнес-логіка або структура даних еволюціонує [1].

На протипагу цьому, NoSQL бази даних пропонують значну гнучкість схеми. Вони дозволяють швидке прототипування, легшу адаптацію до мінливих вимог та зменшення накладних витрат на міграцію бази даних. Вони також підтримують поліморфні моделі даних, що є ідеальним для слабоформалізованих об'єктів. Однак, ця гнучкість вимагає ретельного проектування на рівні додатку для підтримки цілісності даних та оптимізації продуктивності.

Порівнюючи формати даних, JSON (JavaScript Object Notation) є новішим, більш гнучким та компактним форматом, який широко використовується для API та мобільних додатків. Його мінімалістичний синтаксис зменшує накладні витрати на парсинг, і він чудово підходить для обробки динамічних або неструктурованих даних. Натомість XML (Extensible Markup Language), хоча й є старшою технологією, все ще використовується для складних структур документів та суворих визначень схем. Однак, його багатослівність та жорстка система тегів можуть уповільнити розробку. Гнучкість схеми JSON краще узгоджується з потребами адаптивних моделей.

Аналіз показує, що хоча АОМ пропонують потужну адаптивність під час виконання, зберігаючи схему як метадані, їхня складність та багаторівнева абстракція можуть бути заплутаними та важкими для розробників. Подібним чином модель EAV, яка є реляційним підходом до гнучкості, страждає від низької продуктивності, проблем з масштабованістю, труднощів у забезпеченні цілісності даних та складності запитів. Фактично EAV часто вважається анти-шаблоном у реляційних СУБД для складних систем. Це спостереження підкреслює, що для реалізації справді адаптивних моделей

предметної області зі слабоформалізованими об'єктами, документні або графові бази даних NoSQL є найбільш практичним та продуктивним архітектурним вибором. Гнучкість NoSQL безпосередньо зменшує складність та накладні витрати на продуктивність, пов'язані зі спробами примусити адаптивну поведінку до жорстких реляційних моделей або високоабстрактних АОМ у традиційному контексті. Цей вибір безпосередньо підтримує неперервне навчання та адаптацію, які визначають новизну запропонованої моделі. В таблиці 1 наведено порівняння цих підходів.

Таблиця 1

Порівняння гнучких підходів до моделювання даних

Підхід	Гнучкість для слабоформалізованих об'єктів	Продуктивність	Цілісність даних	Масштабованість	Складність реалізації / підтримки	Придатність для видавничої справи
Реляційна модель з EAV	Висока (легко додавати/видаляти атрибути)	Низька при масштабуванні, складні запити	Вимагає логіки додатку, важко забезпечити	Низька, не масштабується	Помірна-Висока (анти-шаблон)	Низька (проблеми з продуктивністю та цілісністю для динамічних витрат)
NoSQL Документна база даних	Висока (динамічна схема, поліморфні дані)	Висока для вкладених даних, швидкі запити	Вбудована валідація схеми, на рівні додатку	Висока (горизонтальне масштабування)	Помірна (потребує проектування)	Висока (ідеально для динамічних витрат, гнучких видань)
NoSQL Графова база даних	Висока (безсхемна, легко додавати вузли/ребра)	Висока для зв'язків та обходу	Природна для зв'язків, на рівні додатку	Висока (горизонтальне масштабування)	Помірна-Висока (специфічна логіка)	Помірна-Висока (корисна для складних взаємозв'язків процесів)

Моделювання предметної області «Видавництво університету». Предметна область «Видавництво університету» є показовим прикладом для демонстрації адаптивної моделі даних завдяки її складним та динамічним процесам, особливо щодо управління витратами та ресурсами.

Видавництво університету відповідає за повний цикл видання книг та інших публікацій, починаючи від отримання замовлення або рукопису до розповсюдження готової продукції. Цей процес включає взаємодію з авторами, планування виробництва, управління матеріалами, використання обладнання, виконання різних видів робіт та контроль за витратами. Ключовою особливістю є мінливість як зовнішніх факторів (ціна на матеріали), так і внутрішніх (ефективність використання витратних матеріалів обладнанням).

Основними об'єктами предметної області є:

ЗАМОВНИК: (ID, Прізвище, Ім'я, По_Батькові, Посада) – *сутність, що представляє особу або організацію, яка ініціює видання.*

ВИДАННЯ: (ID, Назва, Автор1, Автор2, Автор3, Автор4, Автор5, Автор6, Вид_Видання, Формат_Видання, Формат_Рукопису, Кегель_Рукопису, Інтерліньяж_Рукопису, Обкладинка, Тираж) – *центральна сутність, що описує публікацію. Її атрибути, такі як кількість авторів (до 6), вид видання, формат, можуть бути слабоформалізованими, оскільки їх кількість та деталі можуть змінюватися від видання до видання.*

ПЕРЕЛІК ВИДУ ВИДАННЯ: (Вид) – *довідкова таблиця для стандартизації видів видань.*

ПЕРЕЛІК ФОРМАТУ ВИДАННЯ РУКОПISУ: (Формат) – *довідкова таблиця для форматів видань та рукописів.*

ПЕРЕЛІК ТИПУ ОБКЛАДИНКИ: (Обкладинка) – *довідкова таблиця для типів обкладинок.*

ШТАТ ВИДАВНИЦТВА: (Табельний_Номер, Прізвище, Ім'я, По_Батькові, Посада, Дата_Прийняття, ID_Посадової_Інструкції, Період_Відпустки, Дата_Звільнення) – *інформація про співробітників видавництва.*

ШТАТНИЙ РОЗПИС: (Посада, Кількість_Ставок) – *структура штатного розпису.*

ПОСАДОВІ ІНСТРУКЦІЇ: (ID, Посадова_Інструкція) – *деталі посадових інструкцій.*

ВИДИ РОБІТ: (ID_Видання, Сторінок_Макетування, Коефіцієнт_Складності, Розрахунковий_Час_Макетування, Табельний_Номер_Макетувальника, Дата_Початку, Дата_Завершення, Спосіб_Тиражування, Листопідбірка, Метод_Зшивки, Розсилка_Контрольних_Примірників) – *описує конкретні етапи роботи над виданням, такі як макетування, тиражування, зшивка. Ця сутність є критичною для відстеження виробничих кроків, часу та розподілу ресурсів.*

ПЕРЕЛІК СПОСОБІВ ТИРАЖУВАННЯ: (Спосіб_Тиражування) – *довідкова таблиця для методів тиражування.*

ПЕРЕЛІК_МЕТОДІВ_ЗШИВКИ: (Метод_Зшивки) – довідкова таблиця для методів зшивки.

ВИТРАТНІ_МАТЕРІАЛИ: (Код, Назва, Одиниці_Вимірювань, Ціна, Кількість, Дата_Закупки) – перелік витратних матеріалів (наприклад, папір, чорнило). Атрибути Ціна та Кількість є динамічними та підлягають адаптації.

ОБЛАДНАННЯ: (ID, Назва, Балансова_Вартість, Час_До_Проведення_Регламентних_Робіт, Одиниці_Вимірювань, Одиниць_До_Проведення_Регламентних_Робіт, Вартість_Проведення_Регламентних_Робіт, Одиниці_Вимірювань_Продуктивності, Продуктивність, Витратний_матеріал_1, Розхід_1,..., Витратний_матеріал_8, Розхід_8) – інформація про виробниче обладнання. Атрибути Розхід_X (норми споживання) для різних матеріалів є динамічними та потребують адаптації.

Розробка адаптивної моделі даних для видавничого процесу. Для реалізації адаптивної моделі даних у контексті «Видавництва університету» пропонується використовувати документну базу даних NoSQL, таку як MongoDB. Цей вибір обґрунтований її притаманною здатністю обробляти слабоформалізовані об'єкти та підтримувати еволюцію схеми [3]. Документна модель дозволяє представляти сутності як гнучкі документи, що може вміщувати змінну кількість полів, що ідеально підходить для таких об'єктів, як ВИДАННЯ, де кількість авторів може варіюватися, або можуть з'являтися нові атрибути без необхідності міграції схеми.

Наприклад, сутність ВИДАННЯ може бути представлена як єдиний документ, що містить вбудовані масиви для авторів (Автори: [{Прізвище: "...", Ім'я: "..."}, ...]) або деталі, що раніше зберігалися в окремих довідкових таблицях, такі як Вид_Видання або формат_Видання. Це зменшує потребу в складних з'єднаннях та спрощує доступ до даних.

Особливу увагу слід приділити проектуванню сутностей ВИТРАТНІ_МАТЕРІАЛИ та ОБЛАДНАННЯ, оскільки вони є центральними для механізмів адаптації. Замість фіксованих полів Розхід_X в ОБЛАДНАННЯ або простого оновлення Ціни в ВИТРАТНІ_МАТЕРІАЛИ, необхідно реалізувати механізм версіонування атрибутів та моделювання часових рядів даних. Це означає, що Ціна та Розхід_X не є єдиними статичними атрибутами, а скоріше серією значень, кожне з яких має позначку часу початку дії. Просте оновлення цих полів призвело б до втрати історичного контексту, що унеможливило б точний аудит, перерахунок минулих робіт або аналіз

тенденцій. Для забезпечення повної адаптації та коригування розбіжностей з часом, система повинна знати, коли конкретна ціна або норма споживання була дійсною. Це розширює концепцію гнучкої схеми за межі простого додавання нових полів до управління часовою еволюцією існуючих полів, що є глибшою формою адаптації.

Наприклад, у ВИТРАТНІ_МАТЕРІАЛИ атрибут Ціна може бути вбудованим масивом об'єктів, кожен з яких містить значення_ціни, дата_початку_дії та джерело_ціни (наприклад, [{ціна: 1098.5, дата_початку: "2025-01-01"}, {ціна: 1146.2, дата_початку: "2025-03-15"}]). Аналогічно, для ОБЛАДНАННЯ, норми споживання (Розхід_X) можуть бути представлені як вкладені піддокументи або масиви, що дозволяють динамічно додавати типи матеріалів та їх відповідні норми споживання, кожна з яких також може мати часову мітку.

Принципи Domain-Driven Design (DDD) також мають вирішальне значення для розробки цієї моделі [7]. Модель предметної області повинна бути концептуальним представленням бізнесу, що відображає його основні елементи, сутності та зв'язки [4]. Використання «універсальної мови» (наприклад, «Тираж», «Макетування», «Розхід») забезпечує спільне розуміння між розробниками та зацікавленими сторонами бізнесу. Агрегати, такі як «Замовлення» (що включає ЗАМОВНИК та ВИДАННЯ), можуть бути використані для групування пов'язаних сутностей та об'єктів значень, забезпечуючи їх цілісність та послідовність. В таблиці 2 наведено структуру ключових об'єктів предметної області з акцентом на адаптивні атрибути та їх представлення у гнучкій схемі.

Механізми адаптації моделі до зміни вартості та розходу витратних матеріалів. Ключова новизна запропонованої моделі полягає в реалізації неперервного адаптивного циклу зворотного зв'язку, який дозволяє системі автоматично коригувати розрахункові витрати та норми споживання матеріалів. Цей механізм перетворює базу даних зі статичного сховища записів на активний компонент адаптивної системи, постійно уточнюючи її внутрішні параметри на основі реальних операційних даних. Цей процес узгоджується з ширшою тенденцією «інтелекту, керованого даними» та «адаптивного III», де системи навчаються та вдосконалюються без постійного втручання людини. Ця автоматизація знаменує собою значний зсув в операційній ефективності, переходячи від реактивних, ручних коригувань до проактивної, автоматизованої та постійно вдосконалюючої системи управління ресурсами у видавництві. Це також зменшує залежність від «розрізнених знань, зосереджених у кількох експертів».

Таблиця 2

*Структура ключових об'єктів предметної області
з акцентом на адаптивні атрибути*

Об'єкт	Ключові Атрибути	Адаптивні Атрибути	Механізм Адаптації	Рекомендоване представлення у гнучкій схемі
ВИТРАТНІ_МАТЕРІАЛИ	Код, Назва, Одиниці_Вимірювань, Кількість, Дата_Закупки	Ціна	Оновлення на основі останньої закупки або середньозваженої ціни; корекція за розбіжністю з реальними витратами.	Документ: {_id: "код", назва: "...", ціни:}
ОБЛАДНАННЯ	ID, Назва, Продуктивність	Розхід_1 – Розхід_8 (для різних матеріалів)	Корекція за розбіжністю між розрахунковим та реальним споживанням матеріалів під час виконання робіт.	Документ: {_id: "ID", назва: "...", норми_споживання:}
ВИДИ_РОБІТ	ID_Видання, Сторінок_Макетування, Коefіцієнт_Складності, Розрахунковий_Час_Макетування, Табельний_Номер_Макетувальника, Дата_Початку, Дата_Завершення	Немає прямих адаптивних атрибутів, але є джерелом даних для адаптації	Фіксація реальних витрат для зворотного зв'язку.	Документ: {_id: "ID_роботи", видання_id: "...", реальні_витрати: {матеріал_1: {кількість: Z, ціна: P}, ...}}

Покроковий алгоритм адаптації витрат та споживання виглядає наступним чином:

- Збір даних про виконання робіт:** Для кожного завершеного запису ВИДИ_РОБІТ (наприклад, «Макетування» або «Тиражування»), система фіксує деталі про фактично використані ВИТРАТНІ_МАТЕРІАЛИ та ОБЛАДНАННЯ. Це включає реальну кількість спожитих матеріалів та їхню фактичну вартість на момент використання.
- Початковий розрахунок прогнозованих витрат:** Перед початком або після завершення роботи, система розраховує Розраху-

нкові_Витрати. Для цього використовуються Розрахунковий_Час_Макетування та Коефіцієнт_Складності з ВИДИ_РОБІТ, а також поточні (або останні відомі) Ціни з ВИТРАТНІ_МАТЕРІАЛИ та Розхід_Х з ОБЛАДНАННЯ.

3. **Виявлення розбіжностей:** Система порівнює Розрахункові_Витрати з Реальними_Витратами, зафіксованими на кроці 1. Визначається поріг «значної розбіжності» (наприклад, 5% або 10% відхилення).
4. **Механізм корекції:** Якщо виявлена розбіжність перевищує встановлений поріг, система ініціює автоматичне коригування:
 - **Для ВИТРАТНІ_МАТЕРІАЛИ.Ціна:** Якщо аналіз показує, що розбіжність в основному спричинена зміною ціни матеріалу, система оновлює запис про ціну для відповідного матеріалу (або додає новий запис ціни з поточною позначкою часу та джерелом, як описано в таблиці 2). Це може бути зроблено на основі останньої закупівлі або інтеграції з зовнішніми джерелами даних про ринкові ціни.
 - **Для ОБЛАДНАННЯ.Розхід_Х:** Якщо розбіжність вказує на неточну норму споживання обладнанням, система коригує значення Розхід_Х (або додає новий запис норми споживання з поточною позначкою часу) для конкретного матеріалу та обладнання на основі спостережуваного реального споживання. Це є прямим циклом зворотного зв'язку, який дозволяє системі навчатися на основі фактичних даних про використання.
5. **Навчання та уточнення:** Система постійно навчається на основі цих коригувань. З кожним циклом адаптації значення Розхід_Х та Ціна стають точнішими, що призводить до зменшення майбутніх розбіжностей. Це забезпечує неперервну оптимізацію продуктивності системи.
6. **Сповіщення та звітність:** За бажанням система може генерувати автоматичні сповіщення для менеджерів або відповідальних осіб, коли відбуваються значні розбіжності або коли виконуються автоматичні коригування. Це забезпечує прозорість процесу адаптації та дозволяє людське втручання на випадок аномалій.

Інтеграція з програмним забезпеченням. Для безшовної взаємодії між прикладним програмним забезпеченням та адаптивною моделлю бази даних використовуються інструменти Object-Relational Mapping (ORM) або Object-Document Mapping (ODM). Хоча ORM традиційно застосовуються для реляційних баз даних, подібні концепції та переваги поширюються і на NoSQL бази даних через ODM.

ORM/ODM спрощують взаємодію з базою даних, дозволяючи розробникам працювати з даними так, ніби це об'єкти мови програ-

мування, а не безпосередньо писати складні запити до бази даних. Ця абстракція значно покращує читабельність та зручність підтримки коду. У контексті NoSQL, ODM є особливо корисним, оскільки він долає розрив між документоорієнтованим сховищем даних та об'єктно-орієнтованою природою мов програмування, таких як Ruby. Це дозволяє розробникам визначати моделі даних як класи та атрибути, використовуючи знайомі методи та конвенції.

Інтерфейси прикладного програмування (API) є основним засобом взаємодії між різними компонентами системи та зовнішніми додатками. Завдяки використанню JSON як формату обміну даними, який має вбудовану сумісність з веб-API, інтеграція з іншими системами, такими як системи бухгалтерського обліку або управління запасами, стає більш ефективною та простою.

Запропонований підхід до використання баз даних як засобу отримання адаптивної моделі предметної області зі слабоформалізованими об'єктами, продемонстрований на прикладі «Видавництва університету», має значні переваги, але також стикається з певними викликами.

Використання гнучкої схеми NoSQL (документної або графової) є ключовим фактором, що забезпечує високий рівень адаптивності моделі. Об'єкт ВИДАННЯ, який за своєю природою може бути слабоформалізованим (наприклад, змінна кількість авторів, нові види видань, додаткові атрибути), може легко еволюціонувати без необхідності проведення руйнівних міграцій схеми. Це дозволяє видавництву швидко реагувати на зміни в асортименті продукції, виробничих процесах або вимогах ринку, додаючи нові атрибути для ВИТРАТНІ_МАТЕРІАЛИ або ОБЛАДНАННЯ без перепроєктування всієї бази даних. Ця притаманна гнучкість безпосередньо вирішує проблему представлення слабоформалізованих об'єктів.

Бази даних NoSQL, як правило, масштабуються горизонтально, розподіляючи дані між кількома серверами, що робить їх ідеальними для обробки великих навантажень та зростаючих обсягів даних. Це має вирішальне значення для університетського видавництва, яке може значно розширювати свою діяльність, збільшувати кількість видань або обробляти більші обсяги даних про витрати та споживання. Горизонтальне масштабування також забезпечує високу доступність та відмовостійкість системи.

Впровадження цієї адаптивної моделі приносить численні переваги для видавництва.

Неперервний цикл зворотного зв'язку, який автоматично коригує норми споживання та ціни матеріалів, забезпечує дедалі більшу точність розрахункових витрат на публікації. Це дозволяє видавництву мати більш реалістичне уявлення про собівартість продукції та ефективніше управляти фінансами. Точніші дані про споживання

матеріалів та їх вартість призводять до оптимізованого управління запасами та ефективнішого розподілу ресурсів. Це допомагає мінімізувати надлишкові запаси та зменшити ризики дефіциту.

Здатність моделі адаптуватися до мінливих бізнес-вимог (наприклад, впровадження нових типів видань, використання нових матеріалів або придбання нового обладнання) з мінімальним ручним втручанням значно зменшує обсяг переробок, пов'язаних зі змінами. Це дозволяє видавництву залишатися конкурентоспроможним у швидкозмінному середовищі.

Адаптивна модель надає більш надійні та актуальні дані, що є основою для стратегічних рішень. Це включає оптимізацію ціноутворення видань, вдосконалення виробничих процесів та більш обґрунтовані рішення щодо закупівель.

Успішна реалізація такої адаптивної моделі бази даних позиціонує видавництво університету не просто як ефективний операційний інструмент, а як стратегічний актив, що втілює неперервне організаційне навчання. Це виходить за рамки простої цифрової трансформації і приводить до адаптивної трансформації, де основна інформаційна інфраструктура активно бере участь і керує еволюцією бізнесу. Замість того, щоб реактивно реагувати на ринкові зміни, система проактивно коригується, дозволяючи видавництву адаптуватися до швидкозмінних галузевих середовищ.

Висновки. У статті представлено та обґрунтовано підхід, орієнтований на бази даних для створення адаптивної моделі предметної області зі слабоформалізованими об'єктами. Запропонована модель була успішно застосована до предметної області «Видавництво університету», демонструючи її ефективність у вирішенні проблем, пов'язаних з динамічними змінами вартості матеріалів та норм їх споживання. Використання баз даних з гнучкою схемою (зокрема, NoSQL документних баз даних) у поєднанні з інноваційним механізмом неперервного циклу зворотного зв'язку дозволяє системі автоматично виявляти та коригувати розбіжності між розрахунковими та реальними витратами. Цей підхід значно підвищує гнучкість, точність та операційну ефективність управління складними видавничими процесами, перетворюючи базу даних на активний, самооптимізуючий компонент бізнес-логіки.

Подальші дослідження можуть розвинути запропонований підхід у кількох напрямках. Можливе дослідження інтеграції більш складних алгоритмів машинного навчання для прогнозування майбутніх цін на матеріали або оптимізації норм споживання, виходячи за рамки простих циклів зворотного зв'язку. Це може включати використання моделей часових рядів для прогнозування цін або регресійних моделей для уточнення норм споживання.

Перспективним напрямком є розробка інтуїтивно зрозумілих інтерфейсів на основі підходів UI/UX, які дозволяють експертам предметної області (наприклад, менеджерам видавництва) легко взаємодіяти та модифікувати метадані та правила адаптивної моделі, як це пропонується принципами АОМ. Це демократизує процес адаптації та зменшить залежність від ІТ-фахівців.

Варто також зосередитись на дослідженні надійних моделей безпеки та контролю доступу на основі ролей (RBAC), спеціально розроблених для адаптивних баз даних з гнучкою схемою, забезпечуючи цілісність даних та запобігаючи несанкціонованим модифікаціям адаптивної моделі.

Список використаних джерел:

1. HPE. «Adaptive Models». URL: <http://www.hpe.com/us/en/what-is/adaptive-models.html>. Дата звернення: 20.05.2025.
2. Yoder J., Johnson, R. The adaptive object-model architectural style. *3rd IEEE/IFIP Conference on Software Architecture (WICSA3)*. 2002. P. 3-27.
3. Couchbase. «NoSQL Explained: What It Is, How It Works & Why It Matters». URL: <http://www.couchbase.com/resources/why-nosql/>. Дата звернення: 22.05.2025.
4. Смолій В. М., Коваленко О. Є., Смолій Н. В. Інформаційна модель підтримки процесу виробництва електронних пристроїв. *Математичні машини і системи*. 2024. № 2. С. 49-64. DOI: 10.34121/1028-9763-2024-2-49-64.
5. Franconi E., Grandi F., Mandreoli F. A Semantic Approach for Schema Evolution and Versioning In Object-Oriented Databases. *Proceedings of the 2000 International Workshop on Description Logics (DL2000)*. Aachen, Germany, August 17-19, 2000. P. 99-112. DOI: 10.1007/3-540-44957-4_70.
6. Olabiya W. et al., The Evolution of AI: From Rule-Based Systems to Data-Driven Intelligence. *ResearchGate*, January 2025 URL: http://www.researchgate.net/publication/388035967_The_Evolution_of_AI_From_Rule-Based_Systems_to_Data-Driven_Intelligence. Дата звернення: 28.05.2025.
7. Huang Lei. Graph Database Modeling Tools. URL: <http://www.puppygraph.com/blog/graph-database-modeling-tools>. Дата звернення: 20.05.2024.

DATABASE AS A MEANS OF OBTAINING ADAPTIVE MODEL OF A SUBJECT AREA WITH WEAKLY FORMALIZED OBJECTS

The increasing dynamism of modern business environments requires the development of adaptive information systems, especially for subject areas with weakly formalized objects. Traditional static data models often prove ineffective in adequately supporting such rapidly changing conditions, which leads to significant losses in information support of the project due to the need for its processing. An innovative approach is proposed, focused on the use of databases as adaptive models of a subject area. The methodology is based on the use of databases with a flexible schema, in particular NoSQL (document and graph), which are capable of processing large volumes of rapidly changing, unstructured or semi-structured data, offering significantly greater scalability and

adaptability compared to relational databases. Additionally, the principles of adaptive object models are integrated, where classes, attributes, relationships, and behavior are represented as metadata that are interpreted at runtime, allowing subject matter experts to modify the model in real time. The goal of the study is to develop an adaptive model for the subject matter «University Publishing House», which dynamically adjusts to changes in the cost of materials and their consumption rates. The problem is that traditional modeling cannot adequately reflect the inherent dynamism and poorly formalized nature of real business processes, which leads to expensive and time-consuming system modifications. The proposed mechanism implements a continuous feedback loop that automatically detects and corrects discrepancies between estimated and actual costs of consumables. This is a significant departure from traditional “rigid” models, transforming the database into a domain model as an active, self-optimizing component of an adaptive system. This approach increases flexibility, accuracy, and operational efficiency in managing complex, evolving business processes, providing continuous refinement of the system’s internal parameters and reducing the need for manual intervention.

Key words: *adaptive model, loosely formalized objects, database, flexible schema, system evolution, publishing, NoSQL, subject area.*

Отримано: 25.06.2025