DOI: 10.32347/2412-9933.2024.60.136-145
УДК 004.6 005.8

Бугров Анатолій Анатолійович
Аспірант кафедри інформаційних технологій проєктування та прикладної математики,
https://orcid.org/0000-0001-6986-1595
Київський національний університет будівництва і архітектури, Київ
Волох Богдан Юрійович
Аспірант кафедри інформаційних технологій проєктування та прикладної математики,,
https://orcid.org/0000-0003-2846-2621
Київський національний університет будівництва і архітектури,Київ
Босенко Ігор Валерійович
Аспірант кафедри інформаційних технологій проєктування та прикладної математики,
https://orcid.org/0000-0002-9046-4380
Київський національний університет будівництва і архітектури, Київ
Теренчук Світлана Анатоліївна
Кандидат фізико-математичних наук, доцент кафедри технологій проєктування та прикладної математики,
https://orcid.org/0000-0001-6527-4123
Київський національний університет будівництва і архітектури, Київ

СИСТЕМА ПІДТРИМКИ ПРОЦЕСУ ВІДНОВЛЕННЯ ОБ'ЄКТІВ НЕРУХОМОСТІ: ОБРОБКА І ЗБЕРЕЖЕННЯ ДАНИХ

Анотація. *Стаття є продовженням робіт, що спрямовані на розробку інфокомунікаційної системи підтримки процесу відновлення об'єктів нерухомості, які зазнали пошкоджень і руйнувань внаслідок військових дій, що веде російська федерація за підтримки республіки білорусь, на території України з 24 лютого 2022 року. Сформовано цілісне уявлення про високонавантажену розподілену архітектуру масштабованої та стійкої до відмов системи підтримки процесу відновлення об'єктів нерухомості, що призначена для розв'язання багатьох задач будівельної галузі України. Запропоновано універсальну модель модуля обробки і збереження даних. Цей модуль розробляється для оперативної обробки, опрацювання і зберігання великої кількості інформації, представленої в різних форматах, що враховує потребу обробки нечіткої текстової інформації, яка надходить із засобів масової інформації і очевидців подій, та візуальних даних, які отримуються від геоінформаційних систем, систем супутникового спостереження і безпілотних літальних апаратів. Показано діаграму послідовності взаємодії сервісів і значення онтології у вирішенні проблем перевантаження системи і уникнення конфліктів у розумінні даних, що надходять із різних джерел. На основі аналізу низки досліджень у вигляді ідей оптимізації сформовано рекомендації щодо потенційної оптимізації формування онтології системи. При цьому показано можливість використання модуля в процесі формування онтології і шляхи вдосконалення онтологокерованих систем. Практичне значення роботи полягає в забезпеченні безперервного моніторингу і вчасного оновлення даних про об'єкти нерухомості, що важливо для підтримки актуальності інформації під час виконання відновлювальних робіт.*

Ключові слова: інформаційне моделювання; мікросервісна архітектура; розподілена система; онтологія

Вступ

Внаслідок повномасштабної збройної агресії, яку російська федерація за підтримки республіки білорусь веде на території України з 24 лютого 2022 року, пошкоджено і зруйновано багато об'єктів нерухомості.

Основними типами об'єктів нерухомості (об'єктів) є [1]:

1. Житлові об'єкти (будинки, котеджі, квартири, гуртожитки, мансарди, таунхауси).
2. Комерційні об'єкти (ресторани, кафе магазини, сервісні станції, торговельні центри, офіси, готелі).
3. Промислові об'єкти (логістичні центри, виробничі приміщення, складські приміщення заводи, фабрики).
4. Інфраструктурні об'єкти (автостради, аеропорти, мости, тунелі, залізничні станції, портові споруди).
5. Спеціалізовані об'єкти (наукові лабораторії, виставкові центри, церкви, храми, кладовища).

 © А. А. Бугров, Б. Ю. Волох, І. В. Босенко, С. А. Теренчук

6. Спортивні об’єкти (стадіони, басейни, фітнес-клуби, спортивні комплекси).

7. Культурні і освітні об’єкти (університети, школи, музеї, театри, бібліотеки, концертні зали).

8. Земельні ділянки (промислові зони, лісові угіддя, сільськогосподарські угіддя, приватні ділянки, ділянки під будівництво).

Великий обсяг руйнувань різних об’єктів і нагальна потреба багатьох громадян і організацій в прискоренні їх відновлення зумовлює актуальність розробки і впровадження в процес відбудови України спеціальної інформаційно-комунікаційної системи.

Аналіз останніх досліджень і публікацій

Модель інформаційно-комунікаційної системи підтримки процесу відновлення будівель, споруд і об’єктів інфраструктури (СППБСОІ), описана в [2], показана на рис. 1.

Користувачами системи можуть бути громадяни, громади, будівельні фірми, фахівці галузі, експерти, інвестори, центри надання адміністративних послуг (ЦНАП), органи державної влади (ОДВ), органи соціального захисту населення (ОСЗН), органи досудового розслідування (ОДР), спеціалізовані фонди відновлення України (СФВУ) [3].

У цій роботі і надалі ІКСППБСОІ називається системою підтримки процесу відновлення об’єктів нерухомості (СППВОН).

СППВОН призначена для розв’язання багатьох задач високонавантаженою системою, основні вимогами до якої такі [2; 4]:

– гнучкість, що забезпечує можливість швидко і легко вносити зміни в систему;

– продуктивність, що забезпечує потрібну швидкість обробки запитів і відповіді системи;

– безпеку, що полягає в здатності до захисту даних і забезпеченні конфіденційності;

– надійність, що полягає в здатності системи забезпечувати стабільне виконання роботи навіть при збої деяких компонентів;

– масштабованість, що полягає в здатності системи витримувати велике навантаження без великих втрат у продуктивності.

На рис. 2 показано схему комунікації користувачів з СППВОН.

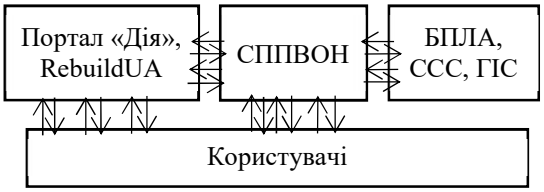

Рисунок 2 – Схема комунікації користувачів з СППВОН: БПЛА – безпілотний літальний апарат; ГІС – геоінформаційна система; ССС – система супутникового спостереження [2]

Для збирання і збереження первинних даних використано Amazon S3.

У [4] засвідчено, що швидкість і надійність роботи інфокомунікаційних систем значною мірою залежить від архітектури і системи керування БД.

Вибір MongoDB обґрунтовано в [5] вимогами до NoSQL, а саме здатністю швидко і ефективно зберігати і обробляти великі обсяги:

– даних, структуру яких не завжди можна визначити на початку;

– геопросторових динамічних даних.

Система керування MongoDB за короткий проміжок часу має обробляти велику кількість запитів користувачів і коректно відповідати на ці запити [7].

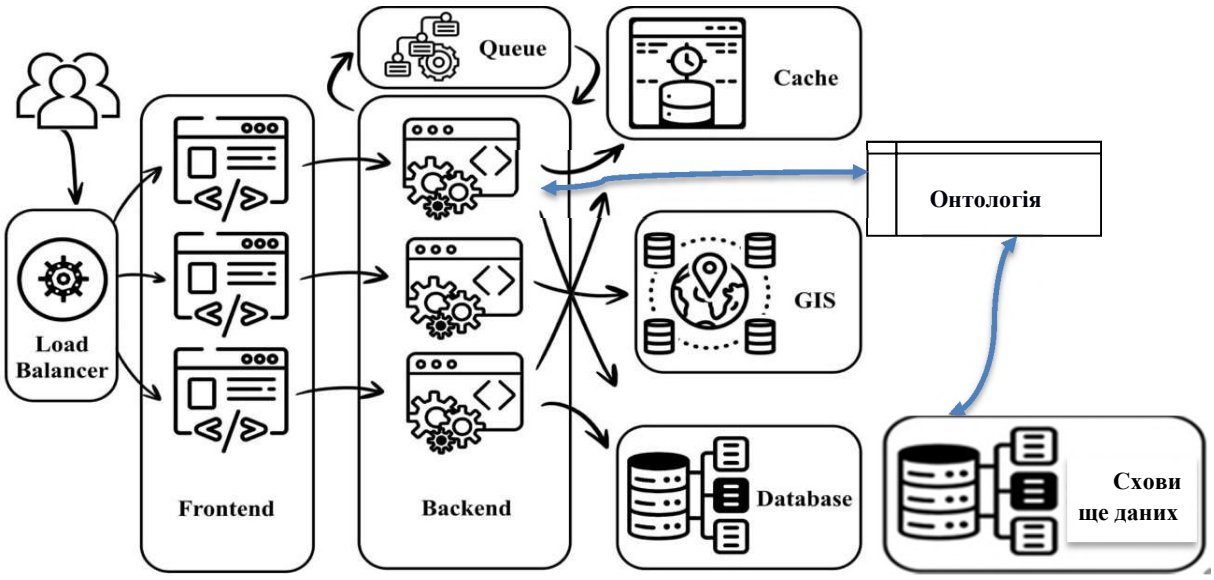

Рисунок 1 – Архітектура ІКСППБСОІ (СППВОН)

Фізичний рівень і схему формування логічного рівня абстракції MongoDB описано в [4]. У цьому дослідженні список полів у документах MongoDB розширено.

З операціями читання та запису застосовується кешування. Це уможливлює ефективно зберігати результати запитів, забезпечуючи швидкий доступ до них під час повторних звернень. Тому в СППВОН для зниження навантаження на MongoDB і скорочення часу відгуку при частих запитах використано технологію кешування. Це сприяє підвищенню продуктивності системи і зменшенню затримок, що особливо важливо для забезпечення відповідної швидкості і високого рівня доступності, враховуючи великий обсяг даних, що постійно збільшується.

Для читання кешування реалізується за допомогою Read-Aside. Коли додаток (Backend) запитує дані, то він спершу перевіряє їх наявність у кеші. Якщо даних немає, то додаток звертається до MongoDB і зберігає отриману інформацію для подальшого використання в кеші. Часовий ліміт для очищення кешу дає змогу швидко отримувати дані, мінімізуючи навантаження на MongoDB, і забезпечує актуальність інформації без накопичення застарілих даних у кеші.

Для запису використовується підхід Write-Aside, який надає додатку контроль над оновленням кешу. При зміні або додаванні нових даних додаток спочатку записує їх у MongoDB, а потім вирішує, чи потрібно оновити кеш. Якщо дані мають бути кешовані, додаток записує їх у кеш. У випадку великих файлів додаток пропускає кешування, залишаючи їх в MongoDB.

Для зберігання даних, що використовуються найчастіше, можуть бути застосовані Redis або Memcached.

Використання черг повідомлень допомагає ефективно керувати піковими навантаженнями. Рішення щодо балансування навантаження забезпечує рівномірний розподіл запитів між серверами. Це уможливює оптимізувати роботу системи і забезпечити вимоги, згадані вище.

Для реалізації вимог до СППВОН можуть бути використані різні архітектурні підходи, кожен з яких має свої переваги та недоліки [8].

Вибір архітектурного підходу залежить від конкретних потреб і обмежень (табл. 1).

Монолітна архітектура системи передбачає, що всі її компоненти зібрані в одному додатку. Це означає, що бізнес-логіка, доступ до даних і користувацький інтерфейс об'єднані в єдиній кодовій базі. Основною перевагою монолітної архітектури є простота, оскільки всі компоненти взаємодіють у рамках одного додатку, що спрощує розробку, тестування.

Таблиця 1 – ***Порівняння архітектурних підходів***

Критерій	Монолітна архітектура	Мікросервісна архітектура	Архітектура на основі подій
Масштабованість	Низька	Висока	Висока
Продуктивність	Висока при малих навантаженнях	Висока	Висока
Безпека	Помірна	Висока	Висока
Надійність	Помірна	Висока	Висока
Гнучкість	Помірна	Висока	Висока
Кількість мов програмування/ технологій	1-2	>3	>3
Час на розроблення	Короткий – швидке розгортання і розроблення	Середній – розроблення окремих сервісів може займати більше часу	Середній – розробка подій і обробників може займати більше часу
Узгодженість даних	Висока – всі дані в одному додатку	Середня – потребує взаємодії між сервісами	Низька – асинхронність може ускладнювати узгодженість
Резервування та відновлення	Просте –резервне копіювання всієї бази даних	Складне – кожен сервіс потребує окремого резервування	Складне – потребує управління подіями та їх станом
Використання	Невеликі проєкти з помірним навантаженням	Великі проєкти з високим навантаженням	Системи з високими вимогами до реального часу та асинхронної обробки

Однак цей підхід має обмеження коли додаток стає дедалі більшим, а його складність і розмір коду ускладнюють внесення змін і оновлень. Проблематичним є і масштабування моноліту, оскільки збільшення навантаження на один компонент може вплинути на всю систему [9].

Комунікація між компонентами в монолітній архітектурі здійснюється через виклики методів або функцій у рамках одного процесу. Це забезпечує високу швидкість взаємодії, але створює залежності, що ускладнюють підтримку і масштабування системи. Щоб зменшити ці проблеми, часто використовують модульний підхід.

Мікросервісна архітектура передбачає розподіл додатка на невеликі, незалежні сервіси, кожен з яких виконує певну функцію. Сервери розробляються, запускаються і розширюються окремо, що уможливлює значно підвищити гнучкість і масштабованість системи. Така архітектура забезпечує високу стійкість системи, оскільки вихід з ладу одного сервісу не впливає на роботу інших. Однак розроблення і підтримка мікросервісної архітектури має і складні моменти.

Ускладнення пов’язані з необхідністю організувати і забезпечити:

– стабільну і коректну комунікацію сервісів;

– узгодженість і координацію під час виконання складних операцій.

Незважаючи на ці виклики, архітектура такого типу дає змогу створювати гнучкі та масштабовані системи, здатні витримувати високе навантаження.

Комунікація між мікросервісами може бути синхронною і асинхронною. Синхронна комунікація здійснюється через такі протоколи, як HTTP/HTTPS (REST). Асинхронна комунікація використовує такі черги повідомлень, як Apache Kafka чи RabbitMQ. Це допомагає підтримувати надійну доставку повідомлень і розподілену обробку запитів [10]. Архітектура на основі подій передбачає, що компоненти системи взаємодіють через події. Коли один компонент генерує подію інші компоненти, що підписані на неї, реагують відповідним чином. Це уможливлює побудувати асинхронну систему, в якій всі елементи працюють незалежно один від одного, що підвищує її гнучкість і масштабованість.

Такий підхід добре підходить для систем, які потребують обробки великої кількості даних у режимі реального часу, оскільки забезпечує високу швидкість реакції на зміни і дає змогу додавати нові функції без значних змін у коді. Однак таку систему складніше налагоджувати і підтримувати, оскільки потрібно стежити за правильним порядком обробки і узгодженістю даних.

В архітектурі на основі подій компоненти взаємодіють через публікацію і підписку на повідомлення за допомогою брокерів повідомлень, таких як Apache Kafka чи Amazon Kinesis. Це забезпечує високу масштабованість і асинхронну обробку [10], що допомагає реалізувати складні сценарії обробки даних і підвищити надійність системи, оскільки відмова одного компонента не впливає на загальну роботу системи.

Отже, проведений аналіз архітектурних підходів до розроблення високонавантажених систем засвідчив, що мікросервісна архітектура є ефективним підходом для створення високонавантажених систем, які легко масштабуються та підтримуються. Це надає підстави вважати, що при розробці СППВОН доцільним є вибір мікросервісної архітектури системи, а першочерговою задачею, яку потрібно розв’язати, є розроблення модуля обробки і збереження даних. При цьому слід ураховувати, що наразі система працює з такими сутностями, як "Користувач", "Документ", "Фотознімок", тож користувачі СППВОН мають різний рівень доступу до різної інформації, а це також висуває відповідні вимоги до безпеки системи [9; 10].

Мета публікації

Метою дослідження є проєктування модуля обробки і збереження даних для системи підтримки процесу відновлення об’єктів нерухомості, що призначений забезпечити високу продуктивність, масштабованість, надійність і безпеку розподіленій системі підтримки процесу відновлення об’єктів нерухомості в умовах високих навантажень.

Виклад основного матеріалу

Універсальну схему мікросервісної архітектури модуля обробки і збереження даних, що уніфікує підхід до маніпуляції даними, забезпечуючи ефективну інтеграцію різних даних із різних джерел інформації, показано на рис. 3.

Взаємодія компонентів модуля (рис. 4) забезпечує комплексний підхід до управління даними, безпеки і продуктивності СППВОН.

Слід зауважити, що кожен компонент, взаємодіючи з іншими, виконує свою специфічну роль для забезпечення безперебійної роботи і високої ефективності системи в цілому, а саме:

– Сервіс авторизації, який відповідає за автентифікацію користувачів і управління їхніми правами доступу, запобігаючи несанкціонованому доступу до ресурсів і даних. Це забезпечує безпеку системи.

– Сервіс маршрутизації, що діє як єдина точка входу для всіх клієнтських запитів, забезпечує маршрутизацію запитів. При цьому балансування навантаження підвищує загальну продуктивність

системи і запобігає перевантаженню окремих компонентів, забезпечуючи рівномірний розподіл запитів між сервісами.

– Сервіс даних, який зберігає інформацію про користувачів і документи, а також забезпечує доступ до цих даних. Цей сервіс є критично важливим для системи, оскільки об'єднує всю основну бізнес-логіку і дані, що зберігаються в системі.

– Геоінформаційний сервіс, що надає доступ до супутникових знімків, телеметрії та інструментів просторового аналізу і просторової статистики використовуючи топологічні, геометричні чи географічні властивості геопросторових даних. Сервіс дає змогу отримувати детальну інформацію про об'єкти: розмір, форму, розташування, поверховість. Це полегшує створення ефективніших планів забудови, надаючи змогу оцінювати навантаження об'єктів на інфраструктуру і їх вплив на навколишнє середовище.

– Сервіс логування, що збирає і зберігає інформацію про всі помилки і збої, які виникають в системі. Це важливо для моніторингу стану системи і оперативного реагування на проблеми, а також для аналізу і вдосконалення системи.

На рис. 3 також показано схему створення онтології, доменом якої є галузь «Архітектура і будівництво».

На початковому етапі створення онтології збираються і зберігаються необроблені дані, що характеризують конкретні об'єкти. Потім зібрані дані аналізуються і обробляються для вилучення значущої інформації і здобуття знань. Цей етап включає курацію експертів, інтелектуальний аналіз даних або їх поєднання.

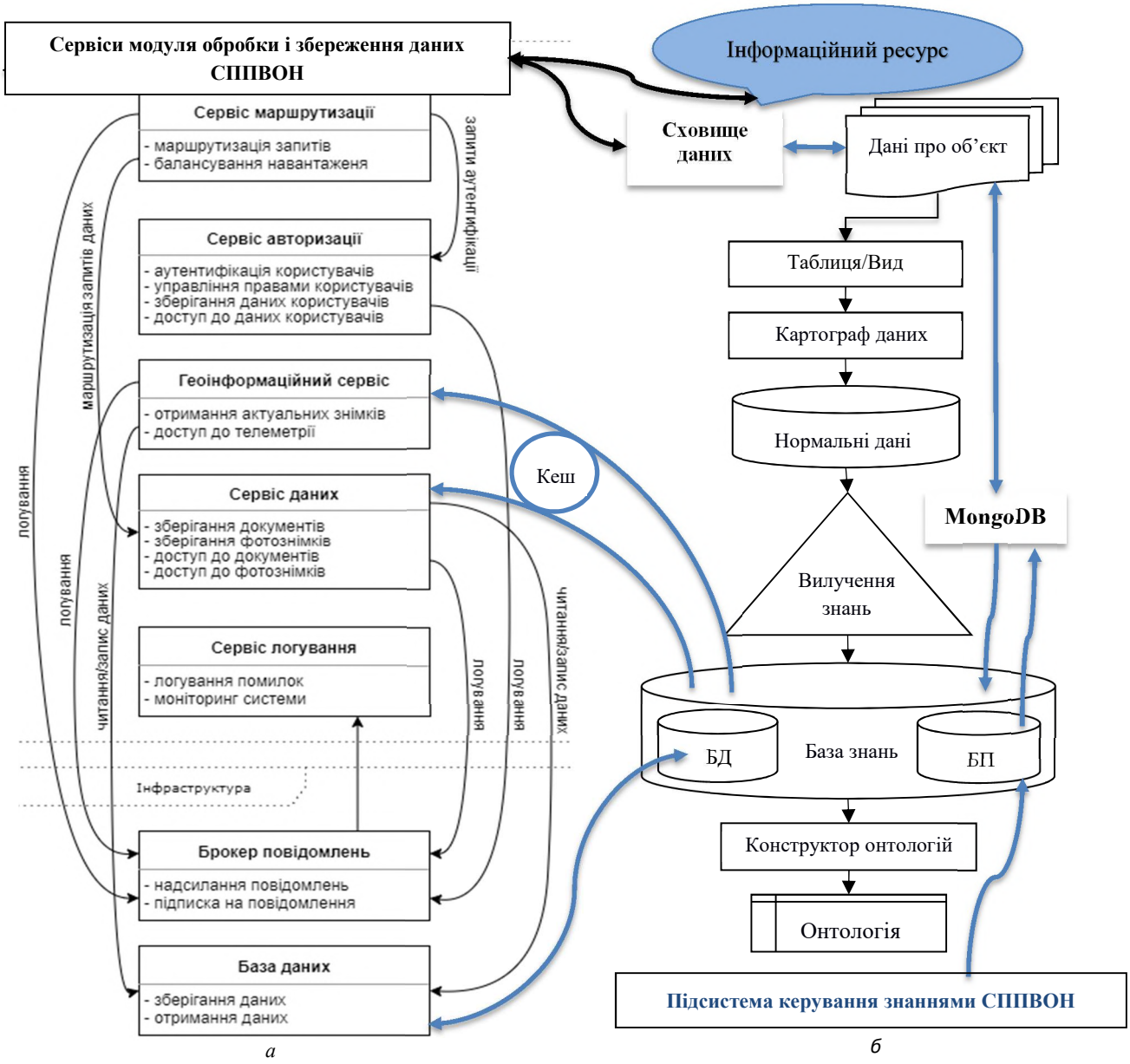

Рисунок 3 - Схема мікросервісної архітектури модуля обробки і збереження даних (а) та схема створення онтології (б)

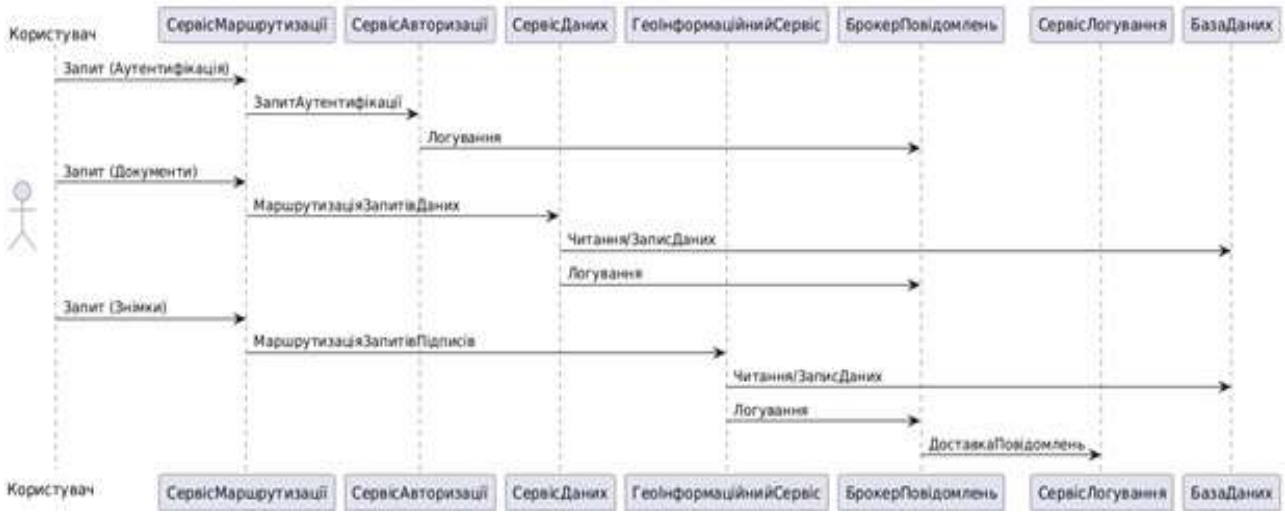

Рисунок 4 - Діаграма послідовності взаємодії сервісів

На етапі "Таблиця/Вид" попередньо оброблені дані організовують у таблиці чи інші структуровані формати, що полегшує їх візуалізацію, виявлення невідповідностей або закономірностей.

Модуль візуалізації даних переводить структуровані дані в уніфікований формат, забезпечуючи дотримання стандартів і роблячи дані однорідними для більш складної обробки.

Після нормалізації, на етапі вилучення знань, за допомогою методів машинного навчання, обробки природної мови чи статистичного аналізу [11; 12] визначаються взаємозв'язки і формулюються правила, які допомагають структурувати знання в цілісну систему. Ці правила і дані зберігаються в базі знань разом з правилами, за якими здійснюється підтримка складних запитів і аналітика.

На цьому етапі створення онтології СППВОН уже може взаємодіяти з базою знань через підсистему керування знаннями. Хоча на цьому етапі дослідження передбачається, що СППВОН через модуль обробки і збереження даних комунікує із зовнішніми джерелами інформації.

Конструктор онтологій використовує ці ресурси для створення онтології, що описує категорії і взаємозв'язки в даних.

Кінцевим продуктом є онтологія – інструмент, що покращує обмін і використання даних в експертних, інформаційно-пошукових системах та системах підтримки рішень різного призначення, забезпечуючи їх сумісність.

Кращого представлення знань можна досягти, якщо використати певні оптимізаційні стратегії ще на етапі проєктування архітектури програмного забезпечення (табл. 2).

Таблиця 2 - ***Рекомендації щодо покращення основних компонентів системи***

N	Компонент	Джерело	Ідеї оптимізації
1	Дані про об'єкт	[13]	Забезпечити адаптивність до нових розробок, розширюючи БД
2	База даних	[10]	Підвищити ефективність і масштабованість за допомогою асинхронного зв'язку і динамічного розподілу ресурсів.
3	Набуття знань	[14]	Покращити обробку даних за допомогою ітеративних методів модифікації моделей.
4	Таблиця/Вид	[13]	Структурувати складні архітектурні дані, покращуючи інтеграцію, нормалізацію та вирівнювання даних у системі.
5	Модуль візуалізації даних	[15]	Впровадити кластеризацію k-середніх для ефективної обробки даних та забезпечити структурованість даних.
6	Звичайні дані	[15]	Підвищити ефективність обробки даних, розставити пріоритети, відбираючи інформацію відповідно до визначених цілей і обмежень.
7	Видобуток знань	[16]	Вирішити потенційні проблеми оцінки якості динамічних даних за допомогою методів оцінки, що базуються на оцінці ризиків.
8	Формування правил	[12]	Знизити ризики та забезпечити надійні та стійкі правила і взаємозв'язки.
9	Знання (БД і БП)	[17]	Створити розширену базу знань, використовуючи прогнозне моделювання, щоб передбачити тенденції даних.
10	Конструктор онтологій	[11]	Використовувати алгоритми оптимізації для систематичного оцінювання і вдосконалення правил знань.

Проте, незважаючи на прогрес в автоматизації процесів розробки онтологій, створення еталонної онтології домену «Будівельна галузь» поки не може бути повністю автоматизовано.

Однією з основних причин є обмеження в охопленні складних зв'язків і контексту домену, які розрізняють фахівці галузі, але ще не можуть «розуміти» комп'ютерні системи внаслідок недостатньої формалізації і нюанси, що пов'язані з:

– створенням консенсусу даних з джерел, що використовують різну термінологію;

– конфліктами у розумінні та інтеграції даних внаслідок використання різних семантичних моделей.

Структурований фрагмент онтології, область застосування якої є СППВОН, показано на рис. 5.

Формування, інтеграція і формалізація об'єктів і концептів онтології на цьому етапі дослідження передбачає участь спеціалістів будівельної галузі.

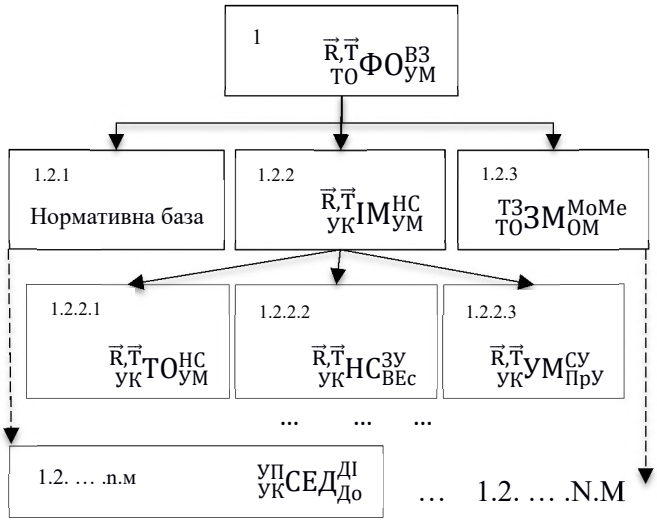

Рисунок 5 – Структурований фрагмент онтології

Тут і надалі прийнято такі скорочення і позначення: ФО – фізичний об'єкт; ІМ – інформаційна модель об'єкта; УК – універсальний класифікатор; ТО – тип об'єкта; НС – навколишнє середовище; МоМе – моделі і методи; ТЗ – технічне завдання; ВЗ – вимоги замовника; ЗМ – засоби моделювання; УМ – умови; ЗУ – зовнішні умови; СУ – суспільні умови; ВЕс – вплив на екосистему; ПрУ – природні умови; ОМ – обмеження; СЕД – система електронного документообігу; УП – учасники процесу; ДІ – джерела інформації; До – документи.

Кожен з цих компонентів і процесів відіграє важливу роль у перетворенні первинних даних щодо об'єктів нерухомості, які можуть бути використані в СППВОН у процесі вирішення таких завдань:

1. Обстеження об'єктів, які наразі проводяться з використанням БПЛА і ССС.

2. Обмін даними при виконанні будівельно-технічної експертизи.

3. Перевантаження і уникнення конфліктів у розумінні даних, що надходять із різних джерел.

4. Аналіз нечіткої текстової інформації від очевидців подій під час виконання судової будівельно-технічної експертизи.

5. Формування звіту щодо пошкоджень об'єктів та причин і наслідків для цих об'єктів і оточуючого середовища.

6. Внесення даних і звітів до спільної інформаційної системи і електронних БД.

7. Планування відновлювальних робіт.

8. Виконання відновлювальних робіт.

9. Обмін даними з BIM при плануванні та супроводі відновлювальних робіт.

BIM-технологія є ефективним інструментом роботи всіх учасників життєвого циклу об'єкта у спільному середовищі та являє частину комплексного процесу будівництва і експлуатації за умови існування єдиного для всіх суб'єктів цифрового простору [18].

Таким простором може бути Єдина державна електронна система у сфері будівництва [19; 20].

BIM дає змогу створювати детальні цифрові моделі наявних будівель, які включають повну інформацію про їхню структуру, матеріали та інженерні системи, а також зберігати і обробляти дані в хмарному середовищі, що забезпечує доступ до даних різним особам і організаціям, що зацікавлені у відновленні об'єктів нерухомості, включаючи архітекторів, інженерів, підрядників та державні органи. Це забезпечує швидкий обмін інформацією і дає можливість усім учасникам процесу працювати з однією актуальною версією даних, що запобігає непорозумінням і підвищує ефективність роботи.

BIM підтримує інтеграцію даних із різних сенсорів, ГІС і БПЛА. Це допомагає суттєво покращити ІМ з урахуванням впливу об'єкта на екосистему та навантажень і впливів на об'єкт навколишнього середовища, зовнішніх умов, що особливо важливо при великомасштабних проєктах відбудови територій після катастроф чи воєнних дій.

Окрім того, використання BIM може значно підтримати процеси оцінювання технічного стану і відновлення територій, зокрема завдяки можливостям модуля обробки і збереження даних СППВОН при:

– інтеграції даних із різних джерел, включаючи державні інформаційні бази і електронні бази нормативної документації;

– моделюванні пошкоджень об'єктів, що дає змогу оновлювати ІМ в процесі обстеження, забезпечуючи актуалізацію даних і прозорість процесу прийняття рішень;

– моделюванні потенційних ризиків та позапроєктних навантажень і впливів;

– плануванні відновлювальних робіт і виконанні експериментів з ІМ.

Висновок

1. Сформовано цілісне уявлення про високонавантажену розподілену архітектуру масштабованої та стійкої до відмов СППВОН.

2. Запропоновано модуль обробки і збереження даних СППВОН, що уможливлює централізовано збирати і зберігати інформацію в різних форматах, забезпечує її актуалізацію і доступ користувачів. Показано можливість використання модуля в процесі формування онтології домену «Будівельна галузь».

3. На прикладі структурованого фрагменту онтології показано схему інтеграції СППВОН з BIM, що в подальшому надасть можливість будувати онтологокеровані системи підтримки рішення, що призначені для розв'язання задач багатокритеріального експертного оцінювання.

4. Подальші роботи планується спрямувати на:

– розширення онтології в напрямі формалізації знань експертів щодо впливу на об'єкти техногенних факторів, які є наслідками повномасштабного вторгнення російської федерації на територію України;

– оптимізацію доступу до даних;

– зменшення навантаження на MongoDB шляхом використання технології кешування;

– роботи геоінформаційного сервісу модуля обробки і збереження даних СППВОН в напрямі підвищ.

Список літератури

1. Типи нерухомості і їх класифікація | 1realty. Перша рієлторська компанія. URL: https://1realty.ua/uk/tipy-nedvizhimosti-i-ih-klassifikacziya-1realty/

2. Terenchuk S., Pasko R., Buhrov A., Ploskyi V., Panko O. and Zapryvoda V., "Computerization of the process of reconstruction of damaged or destroyed real estate," 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2022, pp. 1–6, doi: 10.1109/KhPIWeek57572.2022.9916470.

3. Terenchuk S., Pasko R., Bosenko I., Buhrov A., Yaschenko A. and Volokh B., "Ontology Formation of Support System for Restoration of Buildings, Property and Infrastructure Objects," 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek), Kharkiv, Ukraine, 2023, pp. 1-5, doi: 10.1109/KhPIWeek61412.2023.10313006.

4. Modeling the Process of Assessing the Technical Condition of Damaged Real Estate Objects / B. Volokh et al. 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 4–6 May 2023. 2023. URL: https://doi.org/10.1109/sist58284.2023.10223547.

5. Gilbert S., Lynch N. Perspectives on the CAP Theorem. Computer. 2012. Vol. 45, no. 2. P. 30–36. URL: https://doi.org/10.1109/mc.2011.389

6. Campbell L., Majors C. Database Reliability Engineering: Designing and Operating Resilient Database Systems. O'Reilly Media, 2017. 294 p.

7. Burns B., Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services. O'Reilly Media, 2018. 166 p.

8. Gilbert J. Software Architecture Patterns for Serverless Systems: Architecting for Innovation with Events, Autonomous Services, and Micro Frontends. Packt Publishing, Limited, 2021. 432 p.

9. Blinowski G., Ojdowska A., Przybylek A. Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. IEEE Access. 2022. Vol. 10. P. 20357–20374. URL: https://doi.org/10.1109/access.2022.3152803

10. Pasieka N., Sheketa V., Romanyshyn Y., Pasieka M., Domska U. and Struk A., "Models, Methods and Algorithms of Web System Architecture Optimization," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 2019, pp. 147-152, URL: https://doi.org/10.1109/picst47496.2019.9061539

11. Li R., Etemaadi R., Emmerich M. T. M. and Chaudron M. R. V., "An evolutionary multiobjective optimization approach to component-based software architecture design," 2011 IEEE Congress of Evolutionary Computation (CEC), New Orleans, LA, USA, 2011, pp. 432-439, URL: https://doi.org/10.1109/cec.2011.5949650

12. Meedeniya I., Aleti A., Grunske L. Architecture-driven reliability optimization with uncertain model parameters. Journal of Systems and Software. 2012. Vol. 85, no. 10. P. 2340–2355. URL: https://doi.org/10.1016/j.jss.2012.04.056

13. Aleti A., Buhnova B., Grunske L., Koziolek A. and Meedeniya I., "Software Architecture Optimization Methods: A Systematic Literature Review", IEEE Transactions on Software Engineering. 2013. Vol. 39, no. 5. P. 658–683. URL: https://doi.org/10.1109/tse.2012.64.

14. Martens A., Koziolek H., Becker S., and Reussner R., "Automatically improve software architecture models for performance, reliability, and cost using evolutionary algorithms," in Proceedings of the first joint WOSP/SIPEW International Conference on Performance Engineering, 2010. ACM, 2010, c. 105-116, URL: https://doi.org/10.1145/1712605.1712624

15. Taboada, H. A., Baheranwala, F., Coit, D. W., & Wattanapongsakorn, N. Practical solutions for multi-objective optimization: An application to system reliability design problems. Reliability Engineering & System Safety, 2007, Vol. 92 no. 3, pp. 314–322, URL: https://doi.org/10.1016/j.ress.2006.04.014

16. Kokash N., D'Andrea V. Evaluating Quality of Web Services: A Risk-Driven Approach. Business Information Systems. Berlin, Heidelberg. P. 180–194. URL: https://doi.org/10.1007/978-3-540-72035-5_14.

17. Guo Y., Ge J., Guo P., Chai Y., Li T., Shi M., Tu Y. та Ouyang J., " PASS: Predictive Auto-Scaling System for Large-scale Enterprise Web Applications ", Proceedings of the ACM on Web Conference 2024, pp. 2747-2758, URL: https://doi.org/10.1145/3589334.3645330.

18. Клочко А. А. Цифрові технології в галузі архітектури і будівництва. *Управління розвитком складних систем*. Київ, 2021. № 48. С. 61 – 68, dx.doi.org\10.32347/2412-9933.2021.48.61-68.

19. Кабінет міністрів України розпорядження від 20 травня 2020 р. № 565-р Київ Про затвердження плану заходів щодо створення та запровадження Єдиної державної електронної системи у сфері будівництва. URL: https://zakon.rada.gov.ua/laws/show/565-2020-%D1%80#Text.

20. Кабінет міністрів України постанова від 23 червня 2021 р. № 681 Київ Деякі питання забезпечення функціонування Єдиної державної електронної системи у сфері будівництва. URL: https://zakon.rada.gov.ua/laws/show/681-2021-%D0%BF#Text.

Стаття надійшла до редколегії 12.11.2024

Buhrov Anatolii
Postgraduate student of Department of Information Technology Design and Applied Mathematics,
https://orcid.org/0000-0001-6986-1595
Kyiv National University of Construction and Architecture, Kyiv, Ukraine
Volokh Bohdan
Postgraduate student of Department of Information Technology Design and Applied Mathematics,
https://orcid.org/0000-0003-2846-2621
Kyiv National University of Construction and Architecture, Kyiv, Ukraine
Bosenko Ihor
Postgraduate student of Department of Information Technology Design and Applied Mathematics,
https://orcid.org/0000-0002-9046-4380
Kyiv National University of Construction and Architecture, Kyiv, Ukraine
Terenchuk Svitlana
PhD (Physical and Mathematical Sciences), Associate Professor, Associate Professor of Department of Information Technology Design and Applied Mathematics,
https://orcid.org/0000-0001-6527-4123
Kyiv National University of Construction and Architecture, Kyiv

SUPPORT SYSTEM REAL ESTATE RECONSTRUCTION PROCESS: PROCESSING AND STORAGE OF DATA

Abstract. *This article is a continuation of the work aimed at developing an information and communication system to support the process of restoring real estate objects that have been damaged and destroyed as a result of military operations conducted by the russian federation with the support of the republic of belarus on the territory of Ukraine since February 24, 2022. A holistic view of the highly loaded distributed architecture of a scalable and fault-tolerant system for supporting the process of restoring real estate objects, designed to solve many problems of the construction industry of Ukraine, has been formed. A universal model of the data processing and storage module is proposed. This module is designed for the rapid processing, analysis and storage of a large amount of information presented in various formats, taking into account the need to process fuzzy textual information from the media and eyewitnesses, and visual data obtained from geographic information systems, satellite surveillance systems and unmanned aerial vehicles. The article shows a diagram of the sequence of interaction between services and the importance of ontology in solving the problems of overload and avoiding conflicts in understanding data coming from different sources. Based on the analysis of a number of studies in the form of optimization ideas, recommendations for potential improvement of the system ontology formation are provided. At the same time, the possibility of using the module in the process of ontology formation and ways to improve ontology-managed systems are shown. The practical significance of the work is to ensure continuous monitoring and timely updating of data on real estate objects, which is important for maintaining the relevance of information when performing restoration work.*

Keywords: information modelling; microservice architecture; distributed system; ontology

References

1. Types of real estate and their classification | 1realty. The first real estate company, website. (2019). URL: https://1realty.ua/en/tipy-nedvizhimosti-i-ih-klassifikacziya-1realty/

2. Terenchuk, S., Pasko, R., Buhrov, A., Ploskyi, V., Panko, O. & Zapryvoda, V. (2022). Computerization of the process of reconstruction of damaged or destroyed real estate. In *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)* (pp. 1–6). IEEE. https://doi.org/10.1109/KhPIWeek57572.2022.9916470.

3. Terenchuk, S., Pasko, R., Bosenko, I., Buhrov, A., Yaschenko, A. & Volokh, B. (2023, October). Ontology Formation of Support System for Restoration of Buildings, Property and Infrastructure Objects. In *2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek)* (pp. 1–5). IEEE. https://doi.org/10.1109/KhPIWeek61412.2023.10313006.

4. Volokh, B., Bosenko, I., Pasko, R., Molodid, O., Zapryvoda, V. & Terenchuk, S. (2023, May). Modeling the Process of Assessing the Technical Condition of Damaged real estate objects. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)* (pp. 532–538). IEEE. https://doi.org/10.1109/sist58284.2023.10223547.

5. Gilbert, S. & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, 45 (2), 30–36. https://doi.org/10.1109/mc.2011.389.

6. Campbell, L. & Majors, C. (2017). *Database reliability engineering: designing and operating resilient database systems*. "O'Reilly Media, Inc.".

7. Burns, B. (2018). *Designing distributed systems: patterns and paradigms for scalable, reliable services*. " O'Reilly Media, Inc.".

8. Gilbert, J. & Price, E. (2021). *Software Architecture Patterns for Serverless Systems: Architecting for innovation with events, autonomous services, and micro frontends*. Packt Publishing Ltd.

9. Blinowski, G., Ojdowska, A. & Przybyłek, A. (2022). Monolithic vs. microservice architecture: A performance and scalability evaluation. *IEEE Access*, 10, 20357-20374. https://doi.org/10.1109/access.2022.3152803.

10. Pasieka, N., Sheketa, V., Romanyshyn, Y., Pasieka, M., Domska, U. & Struk, A. (2019, October). Models, methods and algorithms of web system architecture optimization. In *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 147–152). IEEE. https://doi.org/10.1109/picst47496.2019.9061539.

11. Li, R., Etemaadi, R., Emmerich, M. T. & Chaudron, M. R. (2011, June). An evolutionary multiobjective optimization approach to component-based software architecture design. In *2011 IEEE congress of evolutionary computation (CEC)* (pp. 432–439). IEEE. https://doi.org/10.1109/cec.2011.5949650.

12. Meedeniya, I., Aleti, A. & Grunske, L. (2012). Architecture-driven reliability optimization with uncertain model parameters. *Journal of Systems and Software*, 85(10), 2340-2355. https://doi.org/10.1016/j.jss.2012.04.056.

13. Aleti, A., Buhnova, B., Grunske, L., Koziolek, A. & Meedeniya, I. (2012). Software architecture optimization methods: A systematic literature review. *IEEE Transactions on Software Engineering*, 39(5), 658-683. https://doi.org/10.1109/tse.2012.64.

14. Martens, A., Koziolek, H., Becker, S. & Reussner, R. (2010, January). Automatically improve software architecture models for performance, reliability, and cost using evolutionary algorithms. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering* (pp. 105-116). https://doi.org/10.1145/1712605.1712624

15. Taboada, H. A., Baheranwala, F., Coit, D. W. & Wattanapongsakorn, N. (2007). Practical solutions for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering & System Safety*, 92(3), 314-322. https://doi.org/10.1016/j.ress.2006.04.014

16. Kokash, N. & D'Andrea, V. (2007). Evaluating quality of web services: A risk-driven approach. In *Business Information Systems: 10th International Conference, BIS 2007, Poznan, Poland, April 25-27, 2007. Proceedings 10* (pp. 180–194). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72035-5_14

17. Guo, Y., Ge, J., Guo, P., Chai, Y., Li, T., Shi, M., ... & Ouyang, J. (2024, May). PASS: Predictive Auto-Scaling System for Large-scale Enterprise Web Applications. In *Proceedings of the ACM on Web Conference 2024* (pp. 2747–2758). https://doi.org/10.1145/3589334.3645330

18. Klochko, A. (2021). Digital and Info-Communication Technologies in Branch of Architecture and Construction. *Management of Development of Complex Systems*, 48, 61–68, dx.doi.org\10.32347/2412-9933.2021.48.61-68.

19. Cabinet of Ministers of Ukraine Order of 20 May 2020 № 565-r Kyiv On approval of the action plan for the creation and implementation of the Unified State Electronic System in the field of construction. URL: https://zakon.rada.gov.ua/laws/show/565-2020-%D1%80#Text.

20. Cabinet of Ministers of Ukraine Resolution of 23 June 2021 № 681 Kyiv Some issues of ensuring the functioning of the Unified State Electronic System in the field of construction. URL: https://zakon.rada.gov.ua/laws/show/681-2021-%D0%BF#Text

---

Посилання на публікацію

APA Buhrov A., Volokh B., Bosenko I., & Terenchuk S. (2024). Support system real estate reconstruction process: processing and storage of data. *Management of Development of Complex Systems*, 60, 136–145, dx.doi.org\10.32347/2412-9933.2024.60.136-145.

ДСТУ Бугров А. А., Волох Б. Ю., Босенко І. В., Теренчук С. А. Система підтримки процесу відновлення об'єктів нерухомості: обробка і збереження даних. *Управління розвитком складних систем.* Київ, 2024. № 60. С. 136 – 145, dx.doi.org\10.32347/2412-9933.2024.60.136-145.