


Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук


КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА


на тему Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Виконав: студент 2 курсу, група КНм-20-1 
Підпис Р.Р. Гримак
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри ІІЗ 
Підпис О.М. Яшина
Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН 
Підпис Р.О. Багрій
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор 
30 листопада 2021 р. Підпис О.В. Бармак
Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь магістр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук



(підпис)

д.т.н., професор О.В. Бармак

« 1 » вересня 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

1. Тема кваліфікаційної роботи магістра: «Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень»

2. Завдання видано студенту Гримаку Роману Руслановичу

(прізвище, ім'я, по батькові)

3. Керівник роботи к.т.н., доцент Яшина Оксана Миколаївна

4. Затверджені наказом університету від « 25 » серпня 2021 р. № 102

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – розробка методу пониження розмірностей векторного простору для системи автоматизованого проектування Autodesk Revit, яка дозволить користувачам платформи архітектурної візуалізації використовувати функції перегляду, опрацювання, математичного перетворення та серіалізації над елементами будівельних 3D моделей. Для реалізації програмного продукту було обрано технологію .Net Framework 4.8, середовище розробки Microsoft Visual Studio Enterprise 2019, Revit API та Autodesk Revit 2021.

Реферат

Актуальність теми. У сучасному виробництві корисне поширення одержали системи автоматизованого проектування, які надають можливість створювати технологічні процеси з меншими витратами часу та інженерних засобів. Система автоматизованого проектування це система, що здатна на автоматизованому рівні реалізувати інформаційну технологію виконання функцій проектування, це організаційно-технічний комплекс програмних інструментів, призначених для автоматизації процесу проектування, що складається з персоналу і групи технічних, програмних і інших засобів автоматизації його діяльності.

Системи автоматизованого проектування є важливою ланкою в промисловому конструюванні, широко використовуваним в багатьох галузях, в тому числі в автомобільній, суднобудівній та аерокосмічній індустрії, промисловому і архітектурному проектуванні, протезуванні і багатьох інших. САД також широко використовується в створенні комп'ютерної анімації для спецефектів у фільмах, рекламних і технічних матеріалах, які часто називають цифровим контентом. Через свою економічну важливість, система автоматизованого проектування стала основною рушійною силою досліджень в області обчислювальної геометрії, комп'ютерної графіки (як апаратної, так і програмної) і дискретної диференціальної геометрії.

На сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення. Як правило, такі надбудови використовуються у функціональній інфраструктурі спеціалізованого комплексу рішень, що реалізують принцип інформаційного моделювання будівель (Building Information Modeling). Найпоширенішою системою такого типу є Autodesk Revit, тобто платформа, яка надає можливості тривимірного моделювання елементів будівлі і плоского креслення елементів оформлення, призначеного для архітекторів, конструкторів і інженерів-проектувальників.

Мета і задачі роботи. Мета кваліфікаційної роботи полягає в створенні методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору, що дозволить реалізувати алгоритм перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit.

Для досягнення поставленої мети визначені наступні задачі-дослідження:

- формулювання методики перетворення систем координат;
- реалізація фільтра для визначення об'єктів, що підлягають трансферу;
- визначення необхідного переліку функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- побудова інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- реалізація інструментарію для хмарного зберігання даних;
- реалізація перетворення даних до універсального формату зберігання та подальшого трансферу;
- виконати алгоритмічну та програмну реалізацію методу пониження розмірностей векторного простору.

Об'єкт дослідження – процес пониження розмірностей векторного простору для архітектурно-будівельних елементів при переході між інтегрованими системами координат в умовах автоматизованого проектування.

Предмет дослідження – моделі, алгоритми та засоби для побудови методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень.

Методи дослідження, застосовані для вирішення поставлених задач: формулювання методики перетворення систем координат – методи аналітичної геометрії; для реалізація фільтра об'єктів, що підлягають трансферу – методи функціонального аналізу; для визначення необхідного переліку функціональних

параметрів об'єктів, що підлягають трансферу – методи оптимізації; для побудови інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень – методи інтелектуального аналізу даних; для реалізація інструментарію щодо хмарного зберігання результуючих даних – методологія хмарних технологій; для реалізація перетворення даних до універсального формату – методи оптимізації; для програмної реалізації методу пониження розмірностей векторного простору – методології проектування програмних систем та об'єктно-орієнтований підхід;

Наукова новизна одержаних результатів. В результаті проведеної кваліфікаційної роботи були отримані наступні результати:

– вдосконалено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом зменшення розмірностей векторного простору, що дозволило оптимізувати склад об'єктів, які підлягають трансферу між системами візуалізації. Відмінність від відомих підходів полягає у попередній фільтрації переліку об'єктів у поєднанні з переліком необхідних їх функціональних параметрів;

– вдосконалено методику моделювання систем при перетворенні систем координат, використання яких, на відміну від відомих, дозволило у явному вигляді представити координати об'єктів в інструментальній надбудові архітектурно-будівельної платформи Autodesk Revit;

Практичне значення одержаних результатів. Розроблено метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень. В результаті проведених досліджень було створено застосування відносно перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit. Надбудова дає можливість:

– використовувати технології хмарного зберігання даних за допомогою програмного інструментарію.

– застосувати для збереження даних універсальний формат, зручний для розширеного та (або) віддаленого доступу.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові та практичні результати опубліковані в наукових виданнях МОН України:

– публікація на тему “Інформаційна технологія прийняття контрольованих критично-безпекових рішень перетворення параметрів моделей при трансфері між системами візуалізації” в науковому журналі “Вісник Хмельницького національного університету”.

За темою дипломної роботи магістра автором виконано одну наукову публікацію.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 20 найменувань та 4 додатків. Загальний обсяг дипломної роботи магістра становить 127 сторінок, з них 76 сторінок основного тексту та 51 сторінка додатків. У роботі наведено 4 формули та 48 рисунків.

Ключові слова: системи координат, системи візуалізації, Autodesk, Autodesk Revit, Building Information Modeling.

Зміст

| | |
|--|----|
| Перелік скорочень | 2 |
| Вступ..... | 3 |
| Розділ 1. Характеристика предметної області і постановка задачі..... | 7 |
| 1.1 Аналіз предметної області..... | 7 |
| 1.2 Аналіз систем архітектурної візуалізації..... | 12 |
| 1.3 Аналіз хмарних сервісів для зберігання інформації..... | 18 |
| 1.4 Постановка задачі..... | 20 |
| 1.5 Висновки до розділу 1 | 21 |
| Розділ 2. Проектування структури інформаційної системи..... | 22 |
| 2.1 Перетворення координат в системі візуалізації..... | 22 |
| 2.2 Аналіз та автоматизація обробки інформаційних потоків..... | 29 |
| 2.3 Розробка структури інформаційної системи | 36 |
| 2.4 Висновки до розділу 2 | 38 |
| Розділ 3. Програмна реалізація | 40 |
| 3.1 Структура і функціональне призначення модулів системи..... | 40 |
| 3.2 Розробка програмних модулів | 47 |
| 3.3 Висновки до розділу 3 | 52 |
| Розділ 4. Апробація програмної інформаційної системи..... | 53 |
| 4.1 Основні функції архітектурно-будівельної надбудови..... | 53 |
| 4.2 Апробація методу на прикладі моделі приватної власності..... | 59 |
| 4.3 Апробація методу на прикладі моделі житлового комплексу..... | 64 |
| 4.4 Апробація методу на прикладі моделі освітньої установи..... | 68 |
| 4.5 Висновки до розділу 4 | 72 |
| Висновки | 73 |
| Перелік посилань..... | 75 |
| Додатки | |

Перелік скорочень

| Скорочення, термін, позначення | Пояснення |
|---|-------------------------------|
| CAD | Computer-aided design |
| BIM | Building Information Modeling |
| ПЗ | Програмне забезпечення |
| ІС | Інформаційна система |
| Revit | CAD-система Autodesk Revit |

Вступ

Актуальність теми. У сучасному виробництві корисне поширення одержали системи автоматизованого проектування, які надають можливість створювати технологічні процеси з меншими витратами часу та інженерних засобів. Система автоматизованого проектування це система, що здатна на автоматизованому рівні реалізувати інформаційну технологію виконання функцій проектування, це організаційно-технічний комплекс програмних інструментів, призначених для автоматизації процесу проектування, що складається з персоналу і групи технічних, програмних і інших засобів автоматизації його діяльності.

Системи автоматизованого проектування є важливою ланкою в промисловому конструюванні, широко використовуваним в багатьох галузях, в тому числі в автомобільній, суднобудівній та аерокосмічній індустрії, промисловому і архітектурному проектуванні, протезуванні і багатьох інших. САД також широко використовується в створенні комп'ютерної анімації для спецефектів у фільмах, рекламних і технічних матеріалах, які часто називають цифровим контентом. Через свою економічну важливість, система автоматизованого проектування стала основною рушійною силою досліджень в області обчислювальної геометрії, комп'ютерної графіки (як апаратної, так і програмної) і дискретної диференціальної геометрії.

На сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення. Як правило, такі надбудови використовуються у функціональній інфраструктурі спеціалізованого комплексу рішень, що реалізують принцип інформаційного моделювання будівель (Building Information Modeling). Найпоширенішою системою такого типу є Autodesk Revit, тобто платформа, яка надає можливості тривимірного моделювання елементів будівлі і плоского креслення елементів оформлення, призначеного для архітекторів, конструкторів і інженерів-проектувальників.

Мета і задачі роботи. Мета кваліфікаційної роботи полягає в створенні методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору, що дозволить реалізувати алгоритм перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit.

Для досягнення поставленої мети визначені наступні задачі-дослідження:

- формулювання методики перетворення систем координат;
- реалізація фільтра для визначення об'єктів, що підлягають трансферу;
- визначення необхідного переліку функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- побудова інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- реалізація інструментарію для хмарного зберігання даних;
- реалізація перетворення даних до універсального формату зберігання та подальшого трансферу;
- виконати алгоритмічну та програмну реалізацію методу пониження розмірностей векторного простору.

Об'єкт дослідження – процес пониження розмірностей векторного простору для архітектурно-будівельних елементів при переході між інтегрованими системами координат в умовах автоматизованого проектування.

Предмет дослідження – моделі, алгоритми та засоби для побудови методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень.

Методи дослідження, застосовані для вирішення поставлених задач: формулювання методики перетворення систем координат – методи аналітичної геометрії; для реалізація фільтра об'єктів, що підлягають трансферу – методи функціонального аналізу; для визначення необхідного переліку функціональних

параметрів об'єктів, що підлягають трансферу – методи оптимізації; для побудови інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень – методи інтелектуального аналізу даних; для реалізація інструментарію щодо хмарного зберігання результуючих даних – методологія хмарних технологій; для реалізація перетворення даних до універсального формату – методи оптимізації; для програмної реалізації методу пониження розмірностей векторного простору – методології проектування програмних систем та об'єктно-орієнтований підхід;

Наукова новизна одержаних результатів. В результаті проведеної кваліфікаційної роботи були отримані наступні результати:

– вдосконалено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом зменшення розмірностей векторного простору, що дозволило оптимізувати склад об'єктів, які підлягають трансферу між системами візуалізації. Відмінність від відомих підходів полягає у попередній фільтрації переліку об'єктів у поєднанні з переліком необхідних їх функціональних параметрів;

– вдосконалено методику моделювання систем при перетворенні систем координат, використання яких, на відміну від відомих, дозволило у явному вигляді представити координати об'єктів в інструментальній надбудові архітектурно-будівельної платформи Autodesk Revit.

Практичне значення одержаних результатів. Розроблено метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень. В результаті проведених досліджень було створено застосування відносно перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit. Надбудова дає можливість:

– використовувати технології хмарного зберігання даних за допомогою програмного інструментарію.

– застосувати для збереження даних універсальний формат, зручний для

розширеного та (або) віддаленого доступу.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Основні наукові та практичні результати опубліковані в наукових виданнях МОН України:

– публікація на тему “Інформаційна технологія прийняття контрольованих критично-безпекових рішень перетворення параметрів моделей при трансфері між системами візуалізації” в науковому журналі “Вісник Хмельницького національного університету”.

За темою дипломної роботи магістра автором виконано одну наукову публікацію.

Структура та обсяг роботи. Дипломна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 20 найменувань та 4 додатків. Загальний обсяг дипломної роботи магістра становить 127 сторінок, з них 76 сторінок основного тексту та 51 сторінка додатків. У роботі наведено 4 формули та 48 рисунків.

Розділ 1

Характеристика предметної області і постановка задачі

1.1 Аналіз предметної області

Система координат називається комплексом визначень, що реалізує метод координат, тобто підхід визначати переміщення і положення точки або тіла за допомогою чисел чи інших символів. Сукупність чисел, що визначають положення конкретної точки, називається координатами цієї точки [1].

У математиці координати це сукупність чисел, зіставлених точкам різноманіття в деякій карті певного атласу. У елементарної геометрії координати це величини, що визначають положення точки в просторі і на площині (рис. 1.1). На площині положення точки найчастіше визначається відстанями від двох прямих, що перетинаються в одній точці під прямим кутом; одна з координат називається ординатою, а інша - абсцисою. У просторі за системою Декарта положення точки визначається дистанцією від трьох площин координат, що перетинаються в одній точці під прямими кутами один до одного, або сферичними координатами, де початок координат знаходиться в центрі сфери.

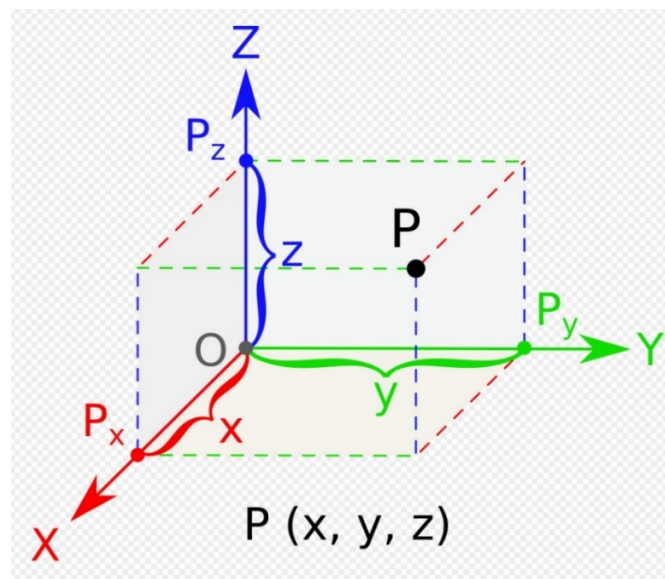


Рисунок 1.1 – Точка P і її координати в тривимірній системі координат

Найбільш вживана система координат представляється в форматі прямокутної системи координат, також відомої як декартова система координат [2]. Координати на площині і в просторі можна вводити нескінченним числом різних способів. Вирішуючи ту або іншу математичну або фізичну задачу методом координат, можна використовувати різні координатні системи, вибираючи ту з них, в якій завдання вирішується простіше або зручніше в даному конкретному випадку. Відомим узагальненням системи координат є системи відліку і системи референції.

Прямокутна система координат це прямолінійна система координат з перпендикулярними осями на площині або в просторі (рис. 1.2). Найбільш проста і тому часто використовувана система координат. Дуже легко і просто узагальнюється для просторів будь-якої розмірності, що також сприяє її широкому застосуванню.

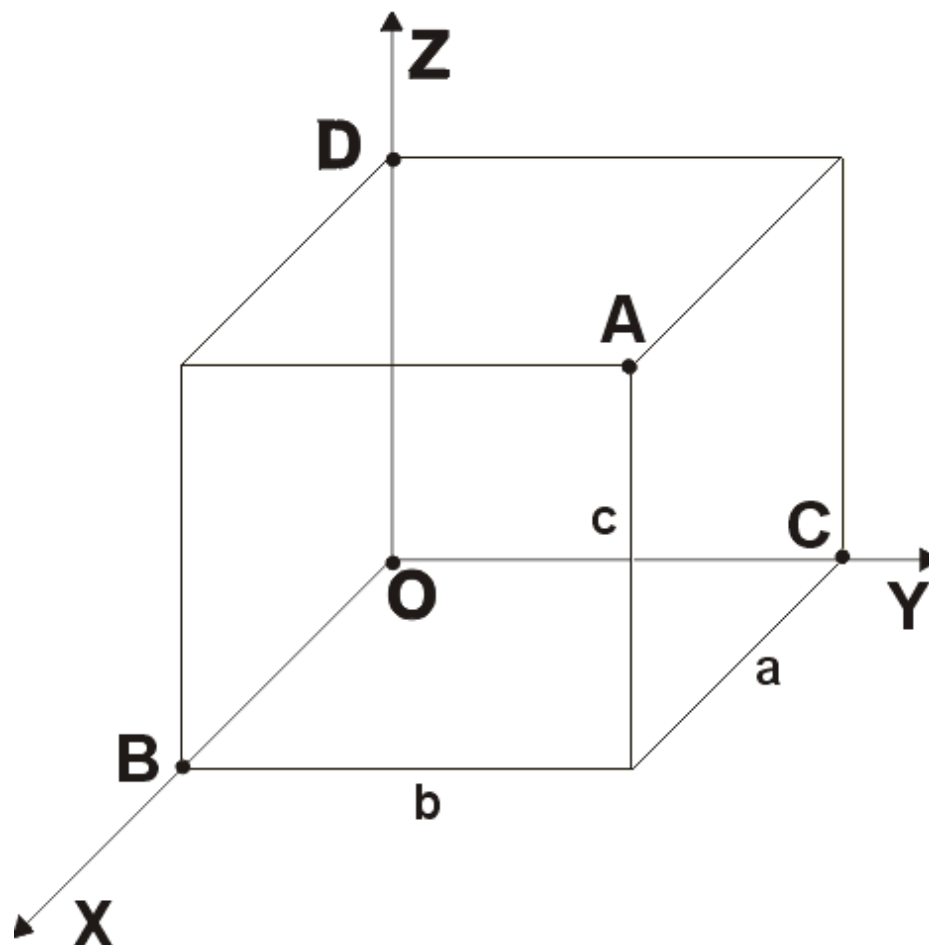


Рисунок 1.2 – прямокутна система координат

Всі прямокутні системи координат в тривимірному просторі діляться на два класи, а саме: ліві та праві. Зазвичай за замовчуванням намагаються використовувати праві координатні системи, а при їх графічному зображенні ще й розташовують їх, якщо можна, в одному з декількох звичайних положень. Праву і ліву системи координат неможливо поворотами поєднати так, щоб збіглися відповідні осі. Визначити, до якого класу належить будь-яка конкретно взята система координат, можна, використовуючи правило правої руки, правило гвинта і так далі. Позитивний напрямок осей вибирають так, щоб при повороті осі OX проти годинникової стрілки на 90° її позитивний напрямок збігався з позитивним напрямком осі OY , якщо цей поворот спостерігати з боку позитивного напрямку осі OZ .

В елементарній математиці виділяють наступні найбільш вживані системи координат:

- циліндрична система координат. Тривимірна система координат, яка є розширенням полярної системи координат шляхом додавання третьої координати, яка задає висоту точки над площиною. Циліндричні координати зручні при аналізі поверхонь, симетричних відносно будь-якої осі, якщо вісь Z взяти в якості осі симетрії.

- полярна система координат. Двовимірна система координат, в якій кожна точка на площині визначається двома числами - полярним радіусом і полярним кутом [3]. Полярна система координат особливо корисна у випадках, коли відносини між точками простіше зобразити у вигляді радіусів і кутів; в більш поширеною декартової, або прямокутної системи координат, такі відносини можна встановити тільки шляхом застосування тригонометричних рівнянь.

- сферична система координат (рис. 1.3). Тривимірна система координат, в якій кожна точка простору визначається трьома числами (r, Θ, φ) , де r це відстань до початку координат (радіальне відстань), а Θ та φ - зенітний і азимутальний кути відповідно [4]. Поняття зеніт і азимут широко використовуються в астрономії. Зеніт - напрямок вертикального підйому над довільно вибраним пунктом, що належить фундаментальній

площині. У якості фундаментальної площини в астрономії може бути обрана площина, в якій лежить екватор, або площина, в якій лежить горизонт, або площина екліптики і т. д., що породжує різні системи небесних координат. Азимут - кут між довільно обраним променем фундаментальної площини з початком в точці спостереження та іншим променем цієї площини, який має спільний початок з першим.

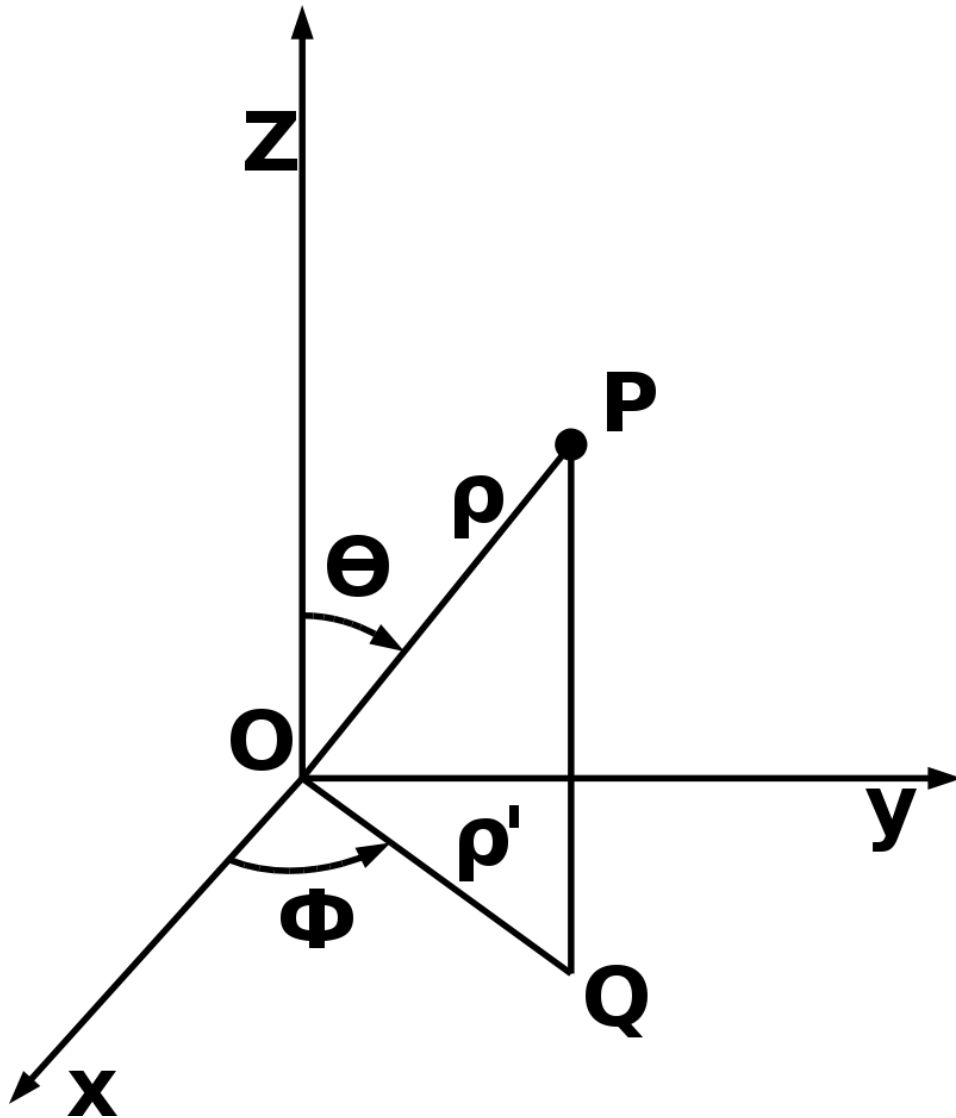


Рисунок 1.3 – сферична система координат

У науковому співтоваристві також використовуються інші поширені системи координат:

– афінна або косокутна система координат. Прямолінійна система координат в афінному просторі. На площині задається точкою початку координат O і двома упорядкованими неколінеарна векторами, які представляють собою афінний базис. Осями координат в даному випадку називаються прямі, що проходять через точку початку координат паралельно векторам базису, які, в свою чергу, задають позитивний напрямок осей. У тривимірному просторі, відповідно, афінна система координат задається трійкою лінійно незалежних векторів і точкою початку координат. Для визначення координат деякої точки M обчислюються коефіцієнти розкладання вектора OM по векторах базису;

– трикутні координати. Один із зразків однорідних координат, які мають своєю основою заданий трикутник, так що положення деякої точки визначається щодо сторін цього трикутника – головним чином ступенем віддаленості від них, хоча можливі й інші варіації. Трикутні координати можуть бути відносно просто перетворені в барицентричні; крім того, вони також конвертовані в двовимірні прямокутні координати, для чого використовуються відповідні формули;

– конічні координати. Тривимірна ортогональна система координат, що складається з концентричних сфер, які описуються за допомогою їх радіусу, і двох сімейств перпендикулярних конусів, розташованих уздовж осей x і z ;

– циліндричні параболічні координати. Тривимірна ортогональна система координат, що отримується в результаті просторового перетворення двовимірної параболічної системи координат. Координатними поверхнями, відповідно, служать конфокальні параболічні циліндри. Циліндричні параболічні координати пов'язані певним ставленням з прямокутними, можуть бути застосовані в ряді сфер наукових досліджень;

Отже, проаналізувавши предметну область та визначивши найважливіші умови для формування інформаційної системи для платформи автоматизованого проектування, можна дійти до конструктивного висновку, що інструментальна надбудова при

інформаційному моделюванні параметрів архітектурно-будівельних елементів буде використовувати базові математичні принципи прямокутної системи координат.

1.2 Аналіз систем архітектурної візуалізації

Autodesk Inc. це найбільший в світі постачальник програмного забезпечення для цивільного та промислового будівництва, індустрії машинобудування, ринку засобів інформації і розваг [5]. Компанією розроблений широкий спектр тиражованих програмних продуктів для конструкторів архітекторів та інженерів. На поточний момент налічується понад 10 млн користувачів Autodesk по всьому світу.

Всього Autodesk в даний час випускає близько ста програмних продуктів. Розробкою займаються три систематизованих підрозділи:

1. Платформи для машинобудування та промислового виробництва це комплексний набір для проектування і дизайну, який використовується в різних галузях промисловості, включаючи електромеханічне, машинобудівне, автомобільне виробництво промислового устаткування і споживчих товарів. Багато продуктів засновані на технології цифрових прототипів. До рішень цього сегмента відносяться:

– AutoCAD Mechanical і AutoCAD Electrical це спеціалізовані рішення для промисловості на основі AutoCAD, призначені для проектування механічних і електричних систем відповідно. Містять додаткові інструменти та бібліотеки компонентів, орієнтовані саме на використання в машинобудівних галузях;

– Autodesk Simulation це сімейство програм для розрахунків та інженерного аналізу. Сюди входять Autodesk Simulation Mechanical для кінематичного аналізу, Autodesk Simulation CFD для вирішення завдань обчислювальної гідрогазодинаміки, а також Autodesk Simulation MoldFlow для моделювання ливарних форм, виробів з пластмаси і процесу лиття під тиском;

– Autodesk Inventor це базове рішення на основі параметричного 3D моделювання для промисловості (рис. 1.4). Програма дозволяє моделювати, проектувати і візуалізувати різні тривимірні об'єкти в цифровому середовищі. В результаті виходить так званий «цифровий прототип», властивості якого повністю відповідають властивостям майбутнього фізичного прототипу аж до характеристик матеріалів;



Рисунок 1.4 – Autodesk Inventor 2019

2. Платформи для архітектурно-будівельної галузі. Програми цієї групи використовуються переважно різними архітектурними та проектними майстернями, а також іншими компаніями будівельної сфери для проектування різних будівель і споруд, моделювання та аналізу їх конструкцій і підсистем. Серед цих рішень системи параметричного проектування на основі технології інформаційного моделювання будівель Autodesk Revit Architecture і Autodesk Revit Structure, додатки для проектування підсистем будівель AutoCAD Architecture, AutoCAD Civil 3D і AutoCAD MEP, а також

аналітичні комплекси для вирішення задач екологічно раціонального проектування. До рішень цього сегмента відносяться:

– AutoCAD Civil 3D це програма для цивільного будівництва, заснована на технології AutoCAD. Використовується для розробки проектів в сфері транспорту, землеустрою та інфраструктури. Також підтримує технологію BIM;

– Autodesk Revit це сімейство програм, які виступають ядром технології інформаційного моделювання будівель. Вони дозволяють опрацьовувати і вивчати концепції майбутніх конструкцій і будівель.

– AutoCAD Map 3D це рішення для картографів, геодезистів і фахівців з геоінформаційних систем, яке надає можливості прямого доступу до різних форматів даних САПР і ГІС, їх редагування, візуалізації та аналізу в середовищі AutoCAD (рис. 1.5);

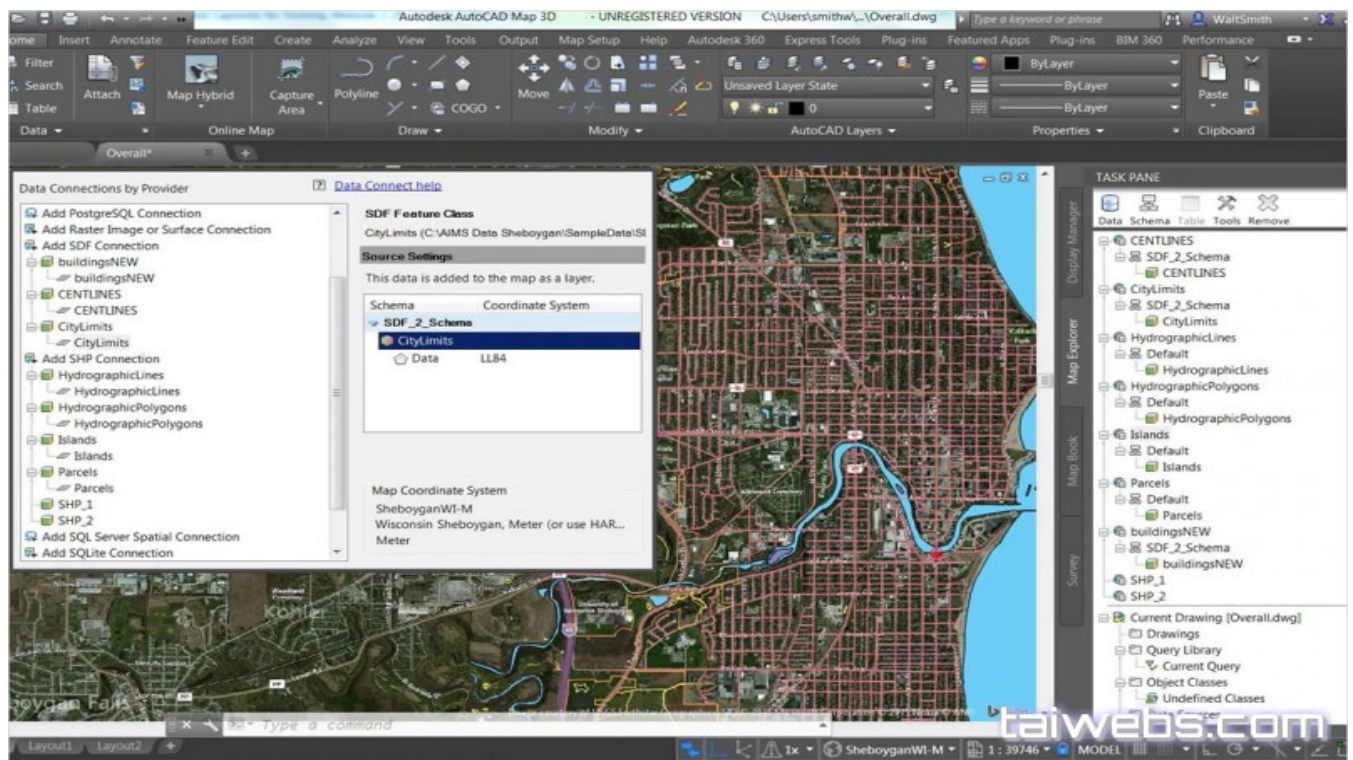


Рисунок 1.5 – AutoCAD Map 3D

3. Платформи для роботи з анімацією, графікою і створення віртуальної реальності. Інструменти для створення мультимедійних матеріалів у всіх сферах індустрії розваг, від візуальних ефектів в кіно і на телебаченні, і кольорокорекції до анімації, рендеринга, і розробки комп'ютерних ігор. Серед основних розробок редактори тривимірної графіки Autodesk 3ds Max, Autodesk Maya, і Autodesk Softimage, рішення для роботи з анімацією Motionbuilder, система «цифрове ліплення» зображень Mudbox, процесори спеціальних відеоефектів Burn, Flint, Inferno, Lustre, Flame, і Smoke.

Інформаційна система параметричних розрахунків на основі технології інформаційного моделювання будівель відноситься до категорії рішень для архітектурно-будівельної галузі, тому на етапі аналізу існуючого програмного забезпечення предметної області були обрані наступні системи автоматизованого проектування: Autodesk Revit та AutoCAD Civil 3D.

Autodesk Revit, або просто Revit це програмний комплекс для автоматизованого проектування, який реалізує принцип інформаційного моделювання будівель (рис. 1.6).



Рисунок 1.6 – Autodesk Revit 2021

Система автоматизованого проектування призначена для архітекторів, конструкторів і інженерів-проектувальників [6]. Платформа надає можливості тривимірного моделювання елементів будівлі і плоского креслення елементів оформлення, створення призначених для користувача об'єктів, організації спільної роботи над проектом, починаючи від концепції і закінчуючи випуском робочих креслень і специфікацій [7].

AutoCAD Civil 3D це програмний продукт компанії Autodesk для фахівців в області геодезії, землеустрою, проектування генплану і об'єктів інфраструктури (рис. 1.7). В його основі лежить використання BIM-технологій і тривимірної математичної моделі об'єктів [8]. AutoCAD Civil 3D дозволяє повністю автоматизувати проектування об'єктів інфраструктури, створення і випуск робочої документації, починаючи зі збору і обробки польових даних, геодезичних вишукувань і закінчуючи 3D-візуалізацією проектного задуму і зведенням самих об'єктів.



Рисунок 1.7 – AutoCAD Civil 3D

AutoCAD Civil 3D допомагає оптимізувати роботу над проектом завдяки таким можливостям, як геопросторовий аналіз для вибору найбільш оптимального

будівельного майданчика, аналіз зливових стоків для забезпечення дотримання екологічних норм, складання кошторису і динамічний розрахунок обсягів земляних робіт для оптимізації витрат матеріалів, а також 3D візуалізація для розуміння впливу проекту на навколишнє середовище [9].

На ринку автоматизованої продукції для рішень архітектурно-будівельної галузі існує велика кількість програмних продуктів для розробки BIM-моделі і подальшого оформлення робочої документації. Якщо розглядати інструменти для моделювання, в пакеті Revit є все необхідне для того, щоб швидко і точно побудувати 3D-модель [10].

При грамотній підготовці шаблону вся інформаційна частина вже буде закладена в BIM-модель. В рамках підготовки шаблону BIM-координатори налаштовують всі необхідні специфікації. У міру побудови BIM-моделі ці специфікації автоматично заповнюються і, в разі будь-яких змін, автоматично оновлюються. Дані настройки можна виконати і динамічними блоками в Autocad, але в деяких технічних сценаріях вони можуть виглядати не надто стабільними.

В системі автоматизованого проектування Revit модель і специфікації взаємопов'язані. Тобто можна вибрати будь-який елемент в специфікації і показати його на будь-якому вигляді, де даний елемент/елементи присутні. Це дозволяє: виключити зайві елементи, швидше заповнити специфікації і здійснити координацію по проекту.

Коли в Revit не вистачає стандартних інструментів, то існує велика база плагінів, які дозволяють виконувати різні види дій. Від найпростіших, таких як вирівнювання елементів по заданому закону або розширений вибір елементів моделі за параметрами, до більш складних - наприклад, автоматична побудова обробки приміщень.

Отже, проаналізувавши існуюче програмне забезпечення предметної області та визначивши всі необхідні умови для створення методу пониження розмірностей векторного простору по відношенню до 3D моделей, можна дійти до конструктивного висновку, що програмна надбудова буде сформована для системи автоматизованого проектування Autodesk Revit.

1.3 Аналіз хмарних сервісів для зберігання інформації

Хмарне сховище це модель хмарних, як правило, високопродуктивних обчислень, яка передбачає зберігання даних в мережі Інтернет за допомогою постачальника хмарних обчислювальних ресурсів, що надає сховище даних як сервіс і забезпечує методами управління вищезгаданими ресурсами [11]. Хмарне сховище надається на вимогу в необхідному обсязі, оплачується за фактом використання і позбавляє від необхідності купувати власну інфраструктуру для зберігання даних і керувати нею. Це забезпечує гнучкість, глобальну масштабованість і надійність. Дані доступні в будь-який час і в будь-якому місці.

Хмарне сховище купується у стороннього постачальника хмарних сервісів, який володіє ресурсами сховища даних, управляє ними і надає доступ до них через Інтернет з оплатою за фактом використання. Постачальники хмарних сховищ відповідають за стан ресурсів, безпеку і надійність, забезпечуючи доступність даних для додатків клієнтів по всьому світу.

Інформаційні системи отримують доступ до хмарного сховища через традиційні протоколи зберігання даних або безпосередньо через API. Багато постачальників пропонують додаткові сервіси, призначені для захисту, збору і аналізу даних у величезних масштабах, а також управління ними [12].

Так як інформаційна технологія перетворення параметрів конструктивних елементів передбачає можливість передачі результуючих даних між системами архітектурної візуалізації, тому на етапі аналізу існуючих хмарних сервісів були обрані наступні постачальники обчислювальних ресурсів: Microsoft Azure та Amazon Web Services.

Microsoft Azure це служба хмарних обчислень, створена корпорацією Майкрософт для створення, тестування, розгортання та керування програмами та службами за допомогою центрів обробки даних, якими керує Microsoft [13]. Поданий

сервіс надає програмне забезпечення як послугу (SaaS), платформу як послугу (PaaS) та інфраструктуру як послугу (IaaS) та підтримує безліч різних мов програмування, інструментів та фреймворків, включаючи як спеціальне програмне забезпечення, так і системи сторонніх розробників (рис. 1.8).

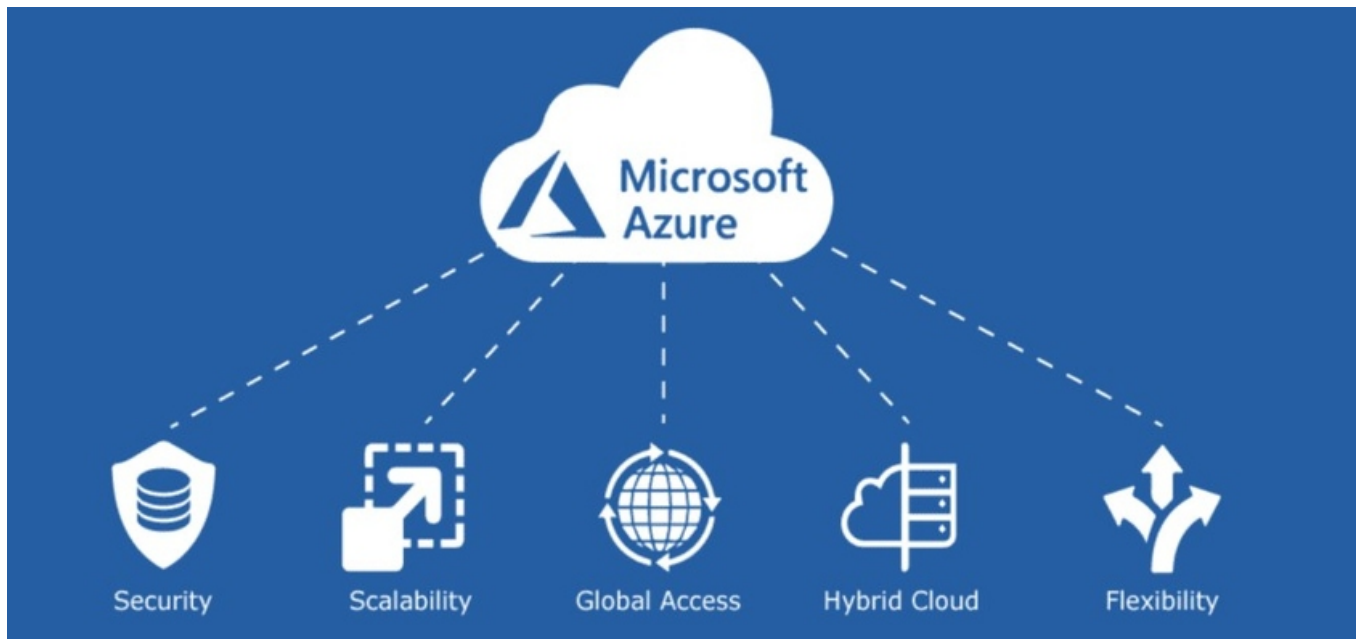


Рисунок 1.8 – Microsoft Azure Service

Amazon Web Services це комплексний сервіс хмарних обчислень, що надає обчислювальну потужність, сховища баз даних, доставку контенту і інші функціональні можливості для створення великих програм з високою гнучкістю, масштабованість і надійністю [14].

Технологія AWS впроваджується на фермах серверів по всьому світу та підтримується дочірньою компанією Amazon. Плата базується на поєднанні використання (відомого як модель "Оплата в процесі використання"), обладнання, операційної системи, програмного забезпечення або мережевих функцій, вибраних абонентом, необхідної доступності, резервування, безпеки та варіантів обслуговування. Абоненти можуть платити за один віртуальний комп'ютер AWS, спеціальний фізичний

комп'ютер або кластери будь-якого з них. В рамках угоди про підписку Amazon забезпечує безпеку систем абонентів (рис. 1.9).

AWS являє собою величезну хмарну інфраструктуру світового масштабу, що дозволяє швидко вводити нововведення, експериментувати і відтворювати свої дії.



Рисунок 1.9 – Amazon Web Services

Отже, проаналізувавши існуючі служби хмарних обчислень та визначивши функціональні умови для зберігання результатів методу пониження розмірностей векторного простору, можна дійти до висновку, що інженерна надбудова для системи автоматизованого проектування Autodesk Revit буде працювати з постачальником Microsoft Azure.

1.4 Постановка задачі

Отже, проаналізувавши характеристики предметної області, комплекс визначень для роботи з положенням математичних об'єктів та системи архітектурно-будівельної

візуалізації, для методу пониження розмірностей векторного простору щодо прийняття контрольованих критично-безпекових рішень при роботі з BIM 3D моделями передбачена реалізація наступних завдань досліджень:

1. Формулювання методики перетворення систем координат;
2. Реалізація фільтра для визначення об'єктів, що підлягають трансферу;
3. Визначення необхідного переліку функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
4. Побудова інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
5. Реалізація інструментарію для хмарного зберігання даних;
6. Реалізація перетворення даних до універсального формату зберігання та подальшого трансферу;
7. Виконання алгоритмічної та програмної організації методу пониження розмірностей векторного простору.

1.5 Висновки до розділу 1

В результаті проведеного аналізу предметної області було визначено, що для належного функціонування конструкторської надбудови необхідні наступні засоби архітектурної візуалізації та хмарного сервісу зберігання даних:

- програмне забезпечення для інформаційного моделювання будівель Autodesk Revit;
- сервіс хмарних обчислень Microsoft Azure.

Затверджено мету роботи та сформульована постановка задач досліджень для побудови методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору.

Розділ 2

Проектування структури інформаційної системи

2.1 Перетворення координат в системі візуалізації

Перетворення координат це заміна системи координат на площині, в просторі або, в найзагальнішому випадку, на заданому n -вимірному різноманітті [15].

Всі прямокутні системи координат в досліджуваному просторі рівноправні, тобто вибір однієї з них нітрохи не гірше і не краще вибору іншого. Ті чи інші переваги віддають, виходячи з особливостей конкретного завдання. Використання різних систем координат ставить задачу перетворення координат точки, тобто задачу обчислення її координат в одній системі координат по її координатам в іншій системі (рис. 2.1).

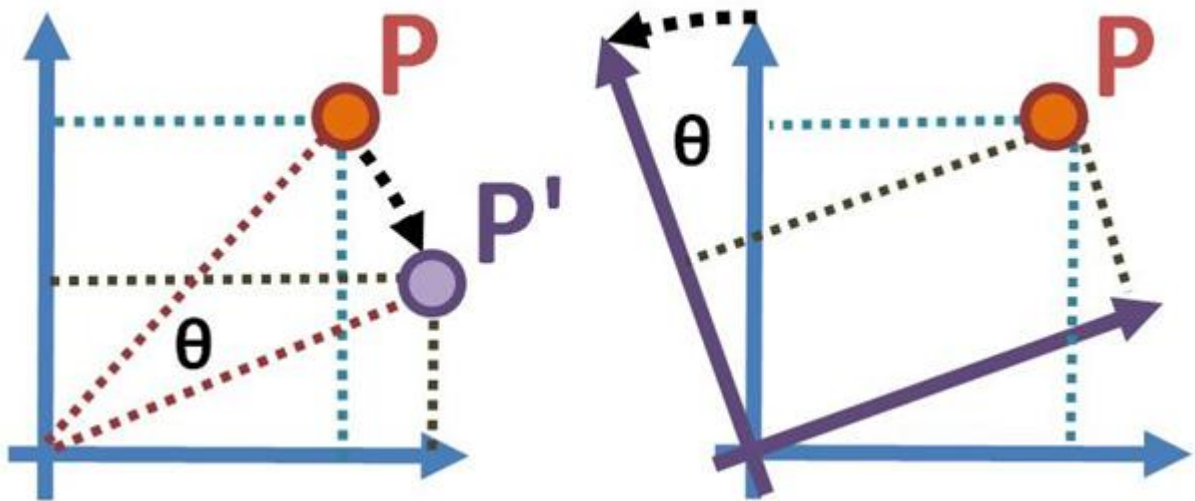


Рисунок 2.1 – перетворення координат

За типом формул всі перетворення координат можна згрупувати в різноманітні класи із загальними типовими властивостями. Далі перераховані деякі практично особливо важливі класи перетворень, які можуть комбінуватися один з іншим:

– ізометрія. Перетворення, що зберігає всі довжини. У тому числі: обертання

навколо точки або осі, паралельний перенос, відображення, поєднання цих трьох типів називається рухом;

– конформне відображення, що зберігає всі кути. В тому числі: подібність, кути зберігаються, але всі довжини множаться на деякий постійний коефіцієнт розтягування/стиснення;

– афінне перетворення. Відображення площини або простору в себе, при якому паралельні прямі переходять в паралельні прямі, криві, що перетинаються - у пересічні, перехресні - в перехресні.

В системі автоматизованого проектування Revit, що призначена для роботи з архітектурними рішеннями, використовуються дві системи координат: система координат зйомки і система координат проекту [16].

Система координат зйомки представляє собою реальний контекст для моделі будівлі. Вона призначена для опису розташування на поверхні землі. Система координат зйомки використовується для ідентифікації конкретного місця розташування на поверхні землі, де знаходиться модель Revit. Ця система координат створюється поза контекстом проекту.

Велика кількість систем координат зйомки стандартизовані. В одних системах використовується широта і довгота, в інших - координати точок на осях X, Y і Z. У системах координат зйомки застосовуються значно більші масштаби, ніж в системах координат проекту, і беруться до уваги такі деталі, як кривизна землі і рельєфу, несуттєві для систем координат систем проекту.

Система координат проекту описує розташування щодо моделі будівлі. Вона використовує обрану точку в межах ділянки в якості опорної точки для вимірювання відстаней і розташування об'єктів щодо моделі.

Система координат проекту використовується, щоб визначити розташування об'єктів щодо заданої точки поруч з моделлю. Дана система координат характерна

виключно для поточного проекту. Початок внутрішньої системи координат є основою для системи координат зйомки і системи координат проекту.

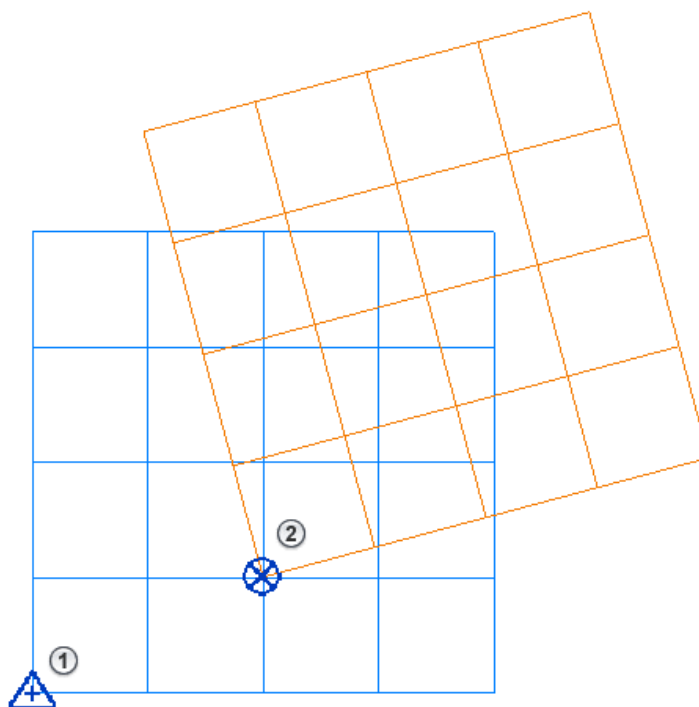


Рисунок 2.2 – системи координат в Autodesk Revit

Термін «система координат зйомки» близький за змістом до наступних термінів, що використовуються в інших додатках і контекстах: глобальні координати, координати ГІС, координати сітки, геодезичні координати, координати проекції, проекція штату, загальні координати.

Термін «система координат проекту» в Revit близький за змістом до наступних термінів, що використовуються в інших додатках і контекстах: локальні координати, світова система координат, світові координати (в контексті файлу DWG), призначена для користувача система координат, призначені для користувача координати, координати проекту, координати моделі, інженерні координати, внутрішні координати, координати САПР.

Інформаційна система щодо перетворення параметрів геометричних об'єктів

передбачає генерування єдиної цифрової моделі в контексті архітектурних компонентів. Для реалізації заданої ідеї потрібна систематизація математичних алгоритмів з метою перетворення координат між елементами BIM моделі та технічних листів.

Сутність вигляду або View являє собою базове представлення для всіх типів View в Autodesk Revit [17]. Вигляд може відображати зображення, створене за моделлю Revit. Відображення можуть бути графічними, наприклад, плани, висоти чи тривимірні подання, або текстовими, наприклад, графіки (рис. 2.3).

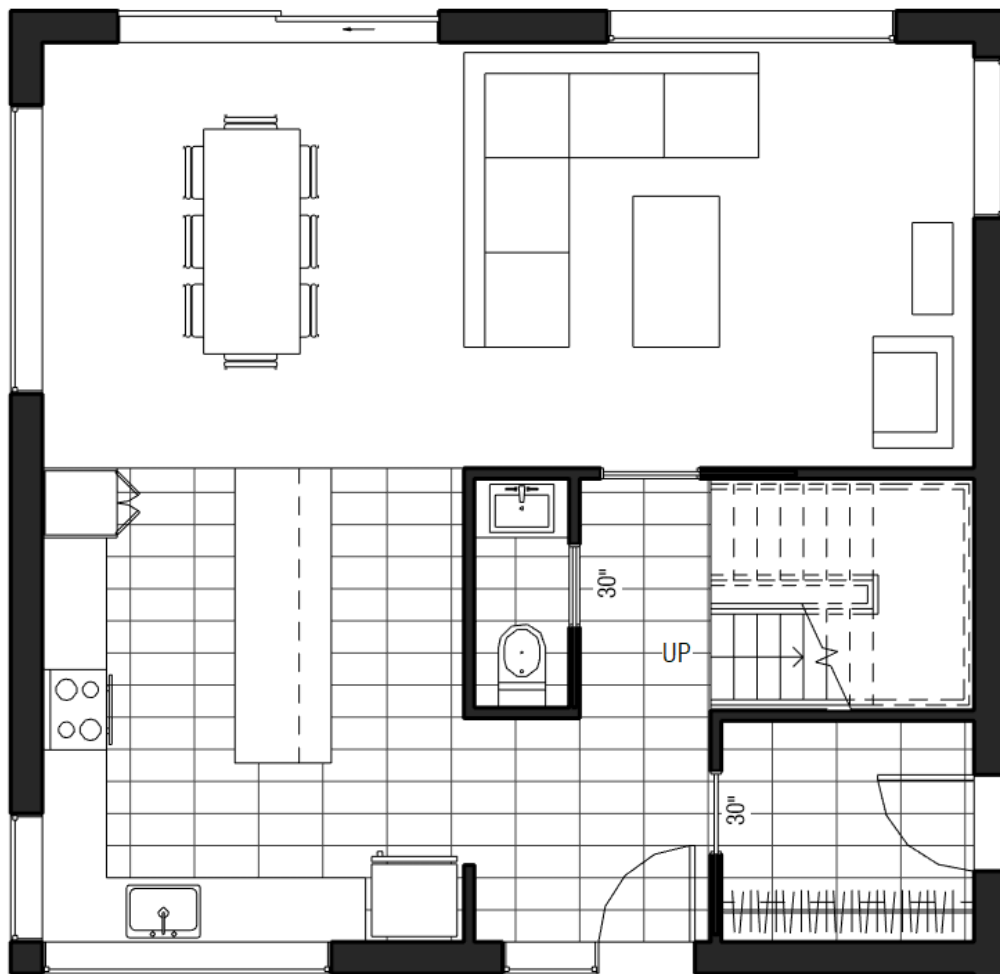


Рисунок 2.3 – вигляд типу Floor Plan в Autodesk Revit

Набір засобів програмного інтерфейсу від системи автоматизованого

проектування Autodesk Revit дозволяє оперувати архітектурними об'єктами на рівні активного View. Як правило, масив геометричних примітивів організований під систему координат проекту, тобто, центром математичних розрахунків виступає точка, яка задається автором документа.

Технічний лист або сутність Sheet/ViewSheet це різновид архітектурного відображення, що складається з модельного вигляду та заголовку (рис 2.4).

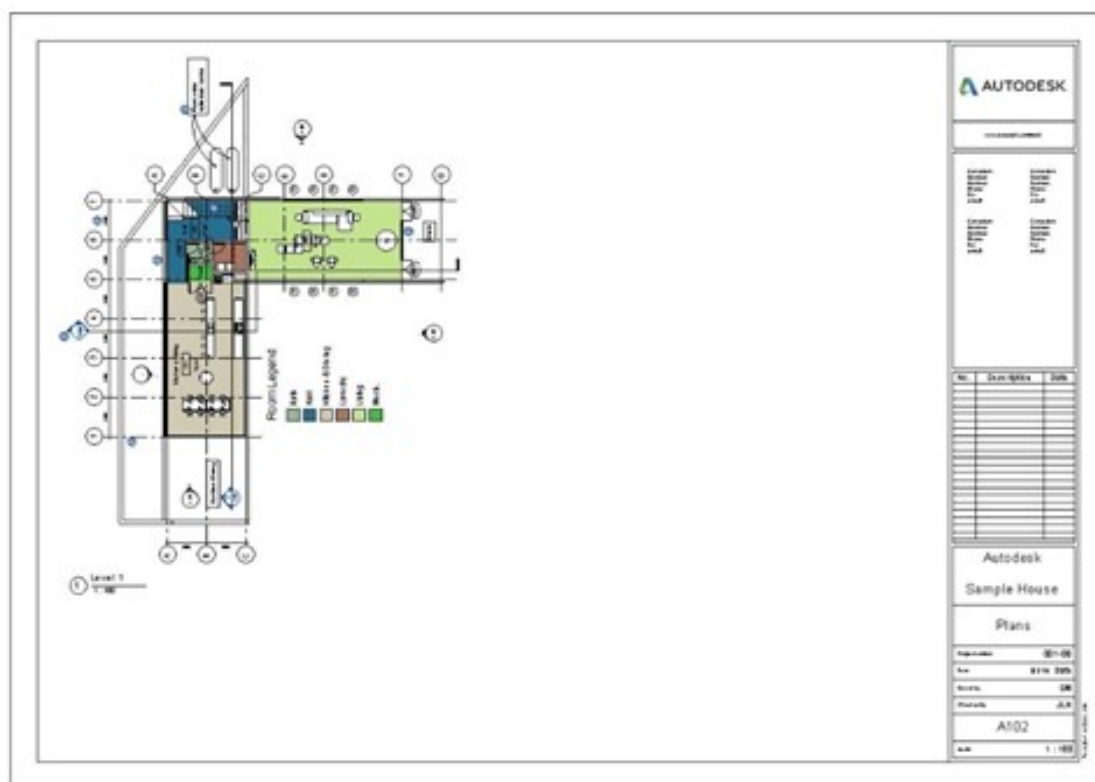


Рисунок 2.4 – вигляд типу Sheet в Autodesk Revit


Кожен лист має унікальний номер аркуша у повному наборі креслень. Номер відображається перед назвою аркуша у браузері проектів.

Основним структурним елементом для сутності ViewSheet є об'єкт під назвою Viewport, що відіграє роль в якості інформаційного посередника між конкретним відображенням поверхів та інженерними листами, на яких згадувані відображення розміщуються [18]. Система координат ViewSheet являє собою двовимірний паперовий

простір, де центром координат по стандарту є нижній лівий кут активного листа. У випадку, якщо користувач платформи для архітектурних рішень спробую видалити заголовок з представлення ViewSheet, Autodesk Revit автоматично виставить центр паперової системи координат в залежності від розташування компонентів Viewport.

Outline це загальний об'єкт, який надає обмежувальне поле/обмежувальний контур і підтримує операції масштабування та трансформації. Перетворення параметрів між інтегрованими системами координат в архітектурній платформі Autodesk Revit в першу чергу розпочинається з розрахунку центральної точки модельного вигляду в рамках двовимірного паперового простору:

$$\text{ViewModelOutlineCenter} = (\text{ViewModelOutlineMax.U} + \text{ViewModelOutlineMax.V}) / 2$$



(2.1)

$$\text{CenterOfModelInSheet} = \text{new UV}(\text{ViewPortCenter.U} - \text{ViewModelOutlineCenter.U}, \text{ViewPortCenter.V} - \text{ViewModelOutlineCenter.V})$$

Для переходу між геометричними характеристиками в системі автоматизованого проектування Revit використовується ознака під назвою scale. Scale являє собою відношення справжнього розміру моделі до розміру паперу [19].

Для того, щоб отримати паперові координати елементів будівельних 3D моделей, необхідно використовувати наступне математичне перетворення:

$$\text{PaperCoordinate} = \text{ModelCoordinate} * (1 / \text{ViewScale})$$
(2.2)

де ModelCoordinate – модельні координати будівельного елемента.

Для отримання модельних координат елементів будівельних 3D моделей, необхідно використовувати математичне перетворення, що зображено нижче:

$$\text{ModelCoordinate} = \text{PaperCoordinate} / (1 / \text{ViewScale}) \quad (2.3)$$

де PaperCoordinate – паперові координати елемента, що використовується в розрахунках.

Кінцевий результат від кількісного перетворення геометричних параметрів з модельного вигляду в систему координат технічного листа відображає інформаційний зв'язок між масштабованим переходом та додаванням центральної точки моделі приміщення на рівні двовимірного паперового простору.

$$\begin{aligned} \text{BoundingBox} = \text{new XYZ}(\text{modelPoint.X} * (1 / \text{ViewScale}) + \\ \text{ViewModelOutlineCenter.U}, \text{modelPoint.Y} * (1 / \text{ViewScale}) + \\ \text{ViewModelOutlineCenter.V}, 0) \end{aligned} \quad (2.4)$$

↓

$$\text{Result} = \text{BoundingBox} - \text{XYZ}(\text{Sheet.Outline.Min.U}, \text{sheet.Outline.Min.V}, 0)$$

де BoundingBox – обмежувальна рамка елемента; ViewScale – масштабування вигляду.

Отже, проаналізувавши основні вимоги до реалізації функціонального перетворення параметрів будівельних елементів та їх переходом від модельної системи координат до схеми технічної специфікації, можна дійти до конструктивного висновку, що для розробки математичної моделі в рамках платформи Autodesk Revit необхідне раціональне використання можливостей вбудованого програмного інтерфейсу Revit API.

2.2 Автоматизація обробки інформаційних потоків

Для проектування структури та систематизації методу пониження розмірностей векторного простору для роботи з перетворенням параметрів архітектурно-будівельних 3D моделей передбачаються аналіз та автоматизація обробки інформаційних потоків. Функціональні можливості інструментальної надбудови повинні задовільнити всі необхідні умови при опрацюванні даних з програмного інтерфейсу Revit API. Бізнес-процеси в роботі користувача Autodesk Revit можна розбити на наступні групи:

1. Бізнес-процес по роботі з відображеннями активного документу:

- вивести список відображень документа;
- перегляд даних щодо активного вигляду;
- інтерактивна взаємодія через Revit: функції підключення та масштабування відносно активного вигляду;
- пошук об'єктів зі списку доступних відображень.

2. Бізнес-процес по роботі з кресленням активного документу:

- вивести список інженерних листів документа;
- перегляд даних щодо активного креслення;
- інтерактивна взаємодія з Revit: функція підключення активного інженерного листа;
- пошук об'єктів зі списку доступних інженерних листів.

3. Бізнес-процес по роботі з кімнатами на рівні активного вигляду:

- вивести список доступних кімнат;
- перегляд даних щодо активної кімнати;
- інтерактивна взаємодія з Revit: функція масштабування до активної кімнати;
- пошук об'єктів зі списку доступних кімнат.

4. Бізнес-процес по роботі з дверима на рівні активного вигляду:

- вивести список доступних дверей;

- перегляд даних щодо активної двері;
- інтерактивна взаємодія з Revit: функція масштабування до активної двері;
- пошук об'єктів зі списку доступних дверей.

5. Бізнес-процес по роботі з підключеними документами:

- вивести дерево рішень для підключених документів;
- вивести кімнати і двері для кожного підключеного документа;

6. Бізнес-процес по роботі з генерацією метаданих:

- вивести список інженерних листів на рівні активного документа;
- надати користувачу можливість вибирати креслення перед відправкою на

ресурси хмарного сервісу Azure;

- перегляд актуальної інформації щодо створених даних;
- на основі вибраних інженерних листів згенерувати метадані та їх картинки.

7. Бізнес процес по роботі з відправленням даних на Azure File Storage:

- переглянути інформацію щодо списку метаданих для відправки;
- надати користувачу можливість видалити метадані;
- відправлення створених метаданих на Azure File Storage.

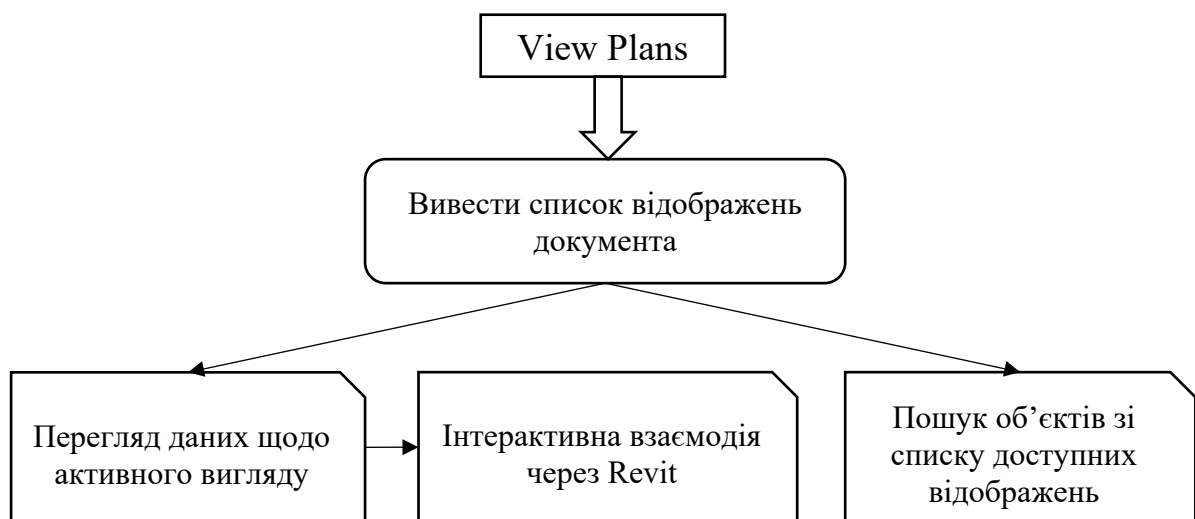


Рисунок 2.5 – Бізнес-процес «Робота користувача з представленнями активного документу»

Бізнес-процес «Робота користувача з відображеннями активного документу».

Даний бізнес-процес дозволяє кінцевому користувачу Autodesk Revit переглянути список відображень активного документу, переглядати інформацію щодо активного вигляду, взаємодіяти з відображеннями за допомогою інтегрованих функцій підключення та масштабування, проводити пошук об'єктів зі списку доступних відображень (рис 2.5).

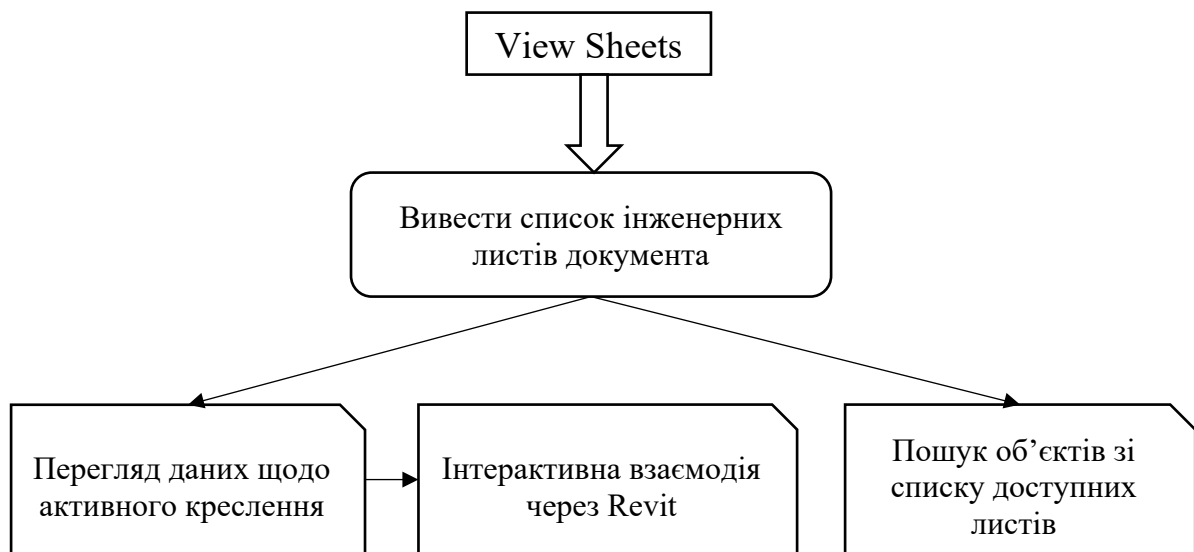


Рисунок 2.6 – Бізнес-процес «Робота користувача з кресленням активного документу»

Бізнес-процес «Робота користувача з кресленням активного документу». Даний бізнес-процес надає кінцевому користувачу Autodesk Revit можливість переглядати список доступних інженерних листів документа, переглядати інформацію щодо активного креслення, взаємодіяти з інженерними листами за допомогою функції підключення, проводити пошук об'єктів зі списку доступних інженерних листів (рис 2.6).

Бізнес-процес «Робота користувача з кімнатами на рівні активного вигляду».

Даний бізнес-процес надає користувачу Autodesk Revit змогу переглядати список доступних кімнат, переглядати дані щодо активної кімнати, взаємодіяти з кімнатами за

допомогою інтегрованої функції масштабування до елемента, проводити пошук об'єктів зі списку доступних кімнат (рис 2.7).

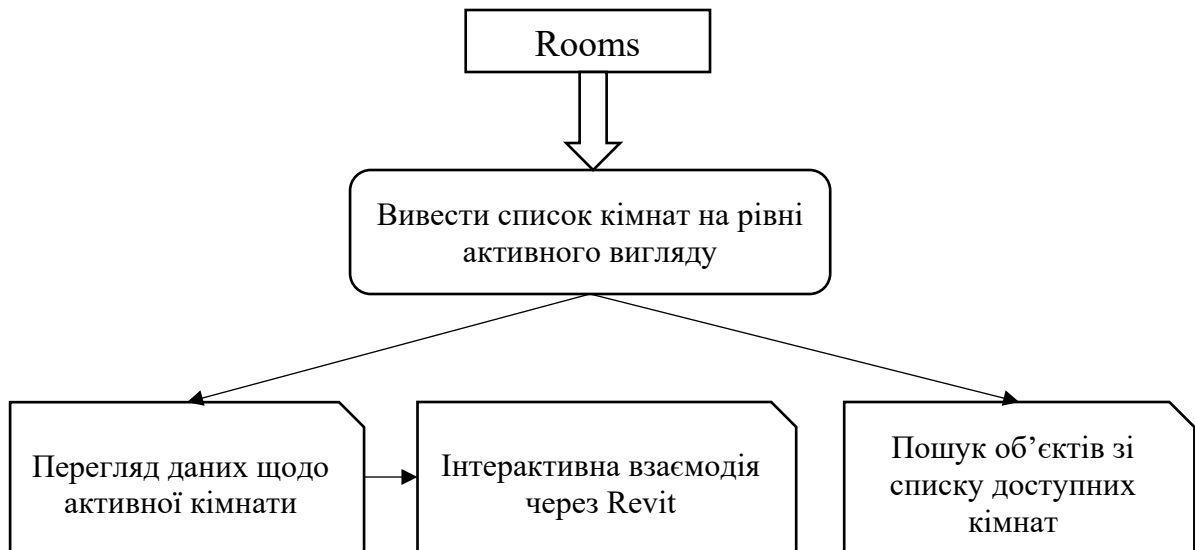


Рисунок 2.7 – Бізнес-процес «Робота користувача з кімнатами на рівні активного представлення»

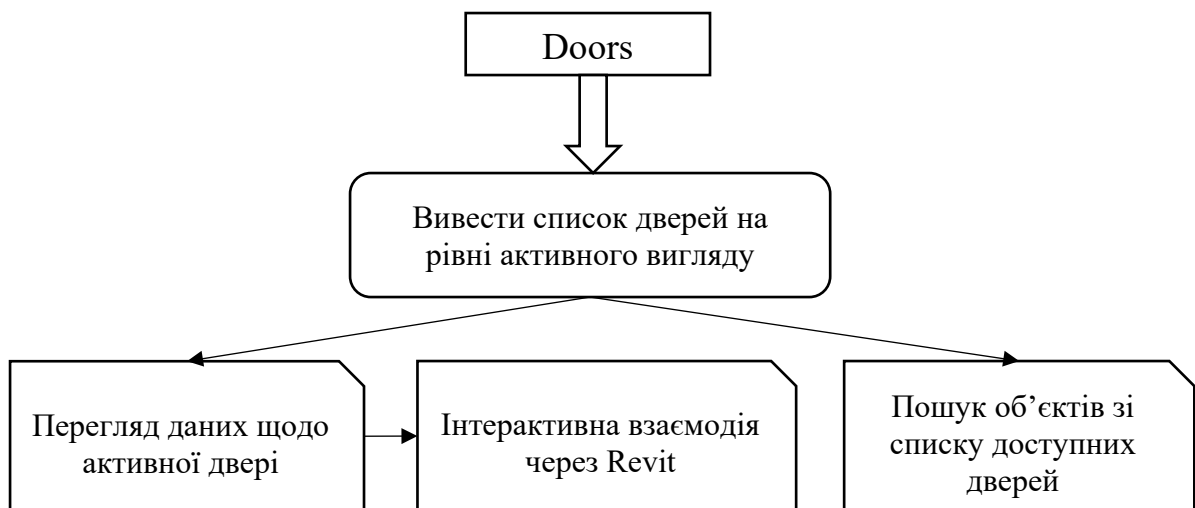


Рисунок 2.8 – Бізнес-процес «Робота з дверима на рівні активного представлення»

Бізнес-процес «Робота користувача з дверима на рівні активного вигляду».

Даний бізнес-процес дозволяє кінцевому користувачу Autodesk Revit переглядати

список доступних дверей, переглядати дані щодо активної двері, взаємодіяти з дверима за допомогою інтегрованої функції масштабування до елемента, проводити пошук об'єктів зі списку доступних дверей (рис 2.8).

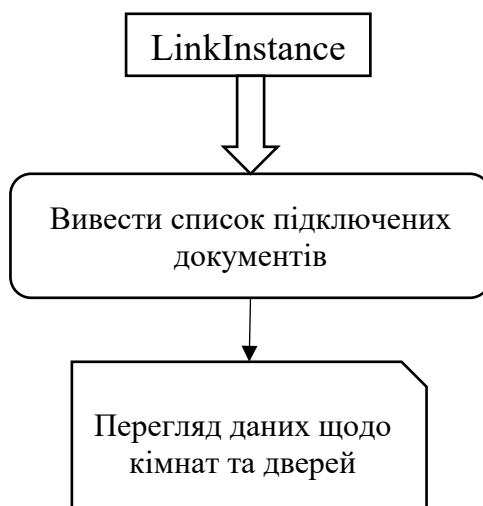


Рисунок 2.9 – Бізнес-процес «Робота користувача з підключеними документами»

Бізнес-процес «Робота користувача з підключеними документами». Даний бізнес-процес надає кінцевому користувачу Autodesk Revit можливість переглядати колекцію доступних підключених документів, переглядати список доступних кімнат та дверей на рівні підключеного документу (рис 2.9).

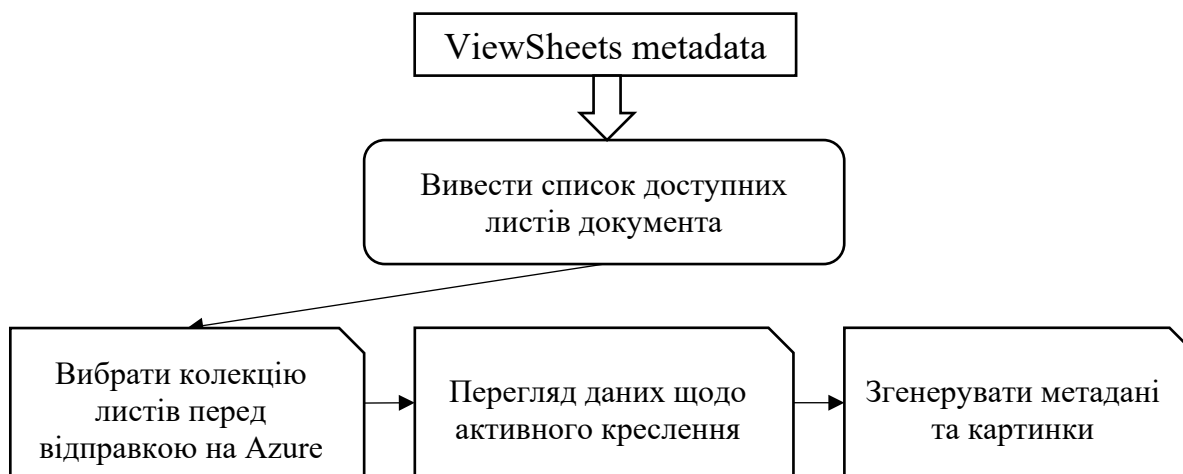


Рисунок 2.10 – Бізнес-процес «Робота користувача з генерацією метаданих»

Бізнес-процес «Робота користувача з генерацією метаданих». Даний бізнес-процес дозволяє кінцевому користувачу Autodesk Revit переглядати список доступних інженерних листів документа, вибрати колекцію доступних листів перед відправкою на Azure, переглядати дані щодо активного креслення, на основі вибраних інженерних листів згенерувати метадані та їх картинки (рис 2.10).

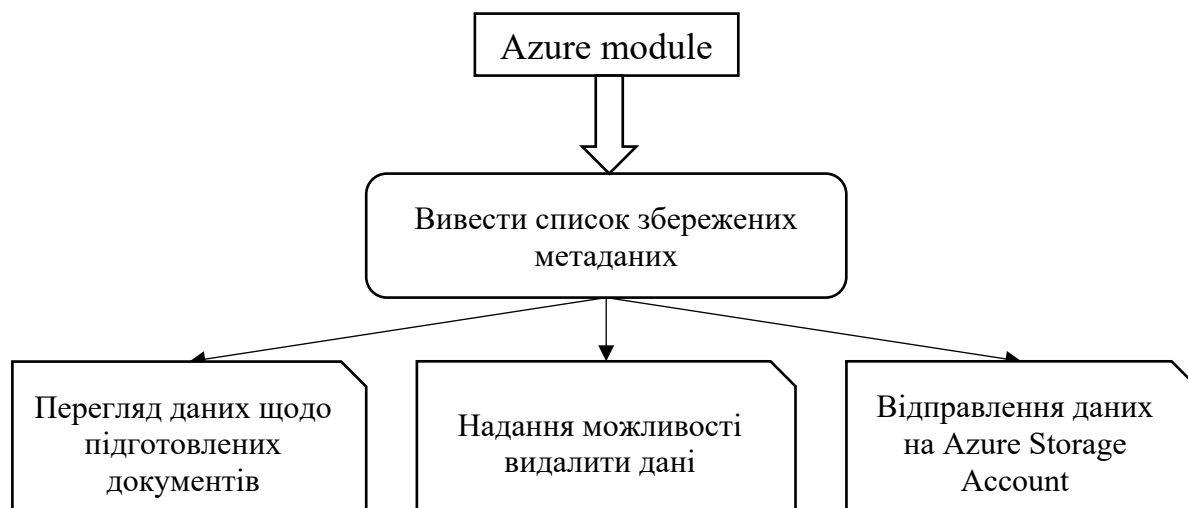


Рисунок 2.11 – Бізнес-процес «Робота користувача з відправленням даних на Azure File Storage»

Бізнес-процес «Робота користувача з відправленням даних на Azure File Storage». Даний бізнес-процес надає кінцевому користувачу Autodesk Revit змогу переглядати інформацію щодо списку метаданих для відправки на хмарний сервіс, видаляти метадані, відправляти метадані на Azure (рис 2.11).

Користувач системи автоматизованого проектування має виконати визначену послідовність дій для активації інженерної надбудови та подальшої роботи з даними відносно підключених архітектурно-будівельних 3D моделей (рисунок 2.12).

Запити з подальшим інформаційним моделюванням архітектурно-будівельних елементів 3D моделей інженерна надбудова в структурі Autodesk Revit повинна здійснювати за рахунок інтегрованого програмного інтерфейсу Revit API. Для роботи з

бізнес-процесами активного документу кінцевому користувачу в першу чергу необхідно затвердити встановлення головної бібліотеки надбудови в функціональну інфраструктуру системи автоматизованого проектування.

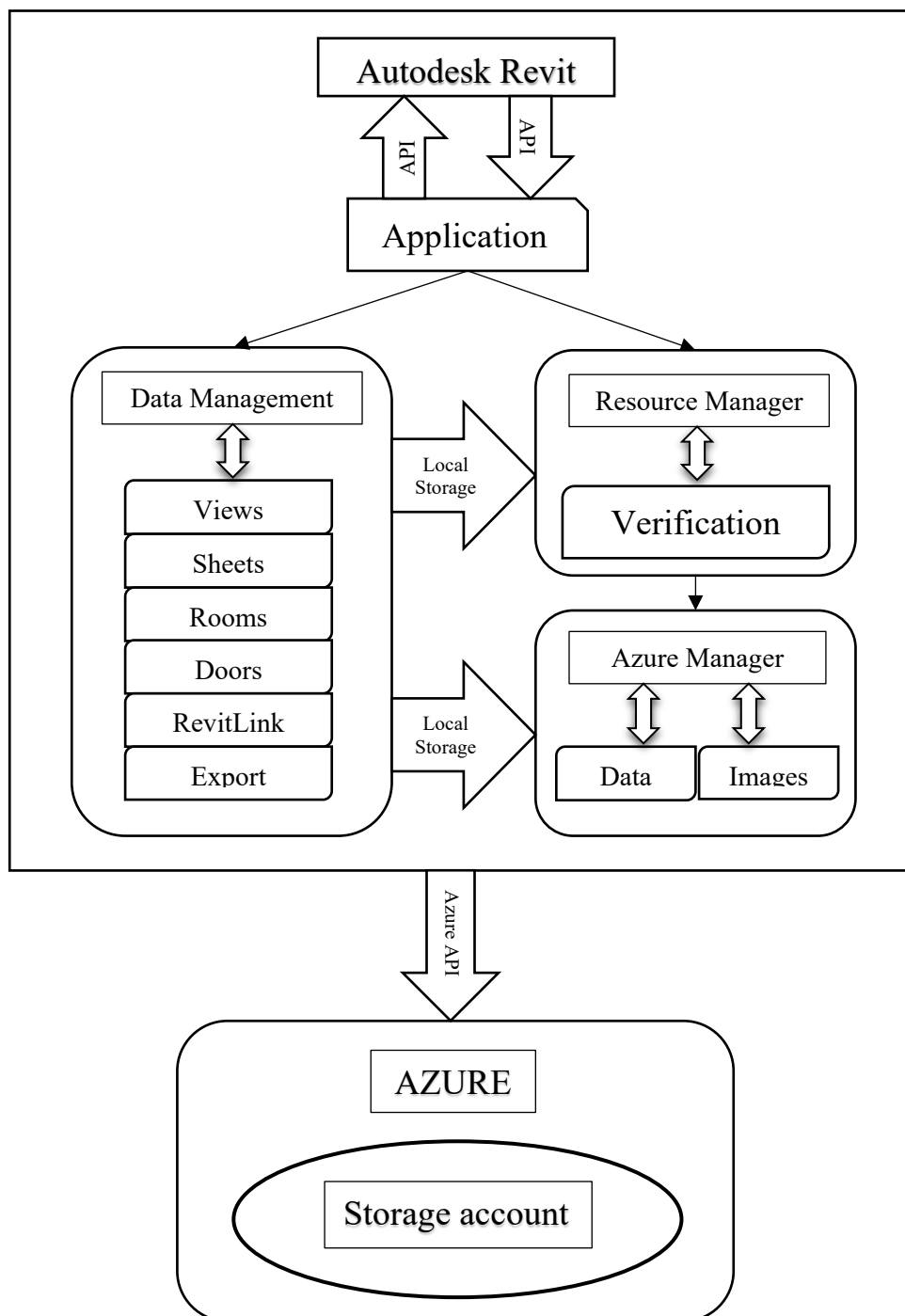


Рисунок 2.12 – функціональна діаграма користувача інженерної надбудови

Передбачається, що виконання інформаційного модулю, менеджера ресурсів та хмарного сервісу Azure буде реалізовано за рахунок компонентів користувацького інтерфейсу від операційної системи Windows. Вищезазначений набір елементів дозволить підготувати дані геометричної специфікації при трансфері між системами візуалізації.

Отже, проаналізувавши обробку інформаційних потоків, для архітектурного рішення щодо перетворення параметрів моделі був визначений повний перелік бізнес-процесів, які підлягають алгоритмам автоматизації. Зважаючи на наведені об'єктно-орієнтовані групи можливостей і організовану функціональну діаграму користувача Autodesk Revit, можна розробити структуру інформаційної системи, яка буде призначена для роботи з елементами приміщення з подальшою генерацією їх геометричних даних на рівні локальної системи координат креслення.

2.3 Розробка структури інформаційної системи

Для забезпечення автоматизації бізнес-процесів, необхідних для функціонування інженерної надбудови в системі архітектурного проектування Autodesk Revit, потрібна технологія зберігання даних. Її структура визначається обсягом інформації, необхідної для організації робочого процесу з документами та листами. Для роботи з будівельними елементами 3D моделей має бути використаний формат JSON.

JSON це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

Зважаючи на те, що основними бізнес-процесами при користуванні можливостей інформаційної технології є робота з представленнями активного документа та з

об'єктами приміщення, то головними параметрами для забезпечення процесу автоматизації слід визначити таблиці 2.1-2.5.

Таблиця 2.1 – Таблиця «RevitDocument» та її атрибути

| Атрибут | Тип даних | Опис |
|---------|-----------|-------------------------|
| Sheets | Object[] | Масив інженерних листів |

Таблиця 2.2 – Таблиця «RevitViewSheet» та її атрибути

| Атрибут | Тип даних | Опис |
|-----------------|-----------|----------------------------------|
| ViewSheetNumber | String | Інформація про номер листа |
| ViewSheetName | String | Інформація про назву листа |
| Scale | String | Інформація про масштабування |
| Unit | String | Інформація про одиниці координат |
| DoorsOfSheet | Object[] | Масив дверей |
| RoomsOfSheet | Object[] | Масив кімнат |

Таблиця 2.3 – Таблиця «RevitDoor» та її атрибути

| Атрибут | Тип даних | Опис |
|--------------|-----------|--------------------------------|
| Guid | String | Ідентифікатор елемента |
| Name | String | Інформація про назву дверей |
| Mark | String | Інформація про марку дверей |
| IsLinked | Boolean | Перевірка стосовно підключення |
| Level | String | Інформація про поверх дверей |
| Max | Object | Максимальна координата дверей |
| Min | Object | Мінімальна координата дверей |
| Tag | Object | Координата тега дверей |
| ToRoomGuid | String | Ідентифікатор вхідної кімнати |
| FromRoomGuid | String | Ідентифікатор вихідної кімнати |

Таблиця 2.4 – Таблиця «RevitRoom» та її атрибути

| Атрибут | Тип даних | Опис |
|------------------|-----------|--------------------------------|
| Guid | String | Ідентифікатор елемента |
| Name | String | Інформація про назву кімнат |
| Number | String | Інформація про номер кімнат |
| IsLinked | Boolean | Перевірка стосовно підключення |
| Tag | Object | Координата тега кімнат |
| BoundarySegments | Object[] | Масив координат кутів кімнати |

Таблиця 2.5 – Таблиця «JsonCoordinate» та її атрибути

| Атрибут | Тип даних | Опис |
|---------|-----------|-----------------------------|
| X | Decimal | Інформація про координату X |
| Y | Decimal | Інформація про координату Y |

Отже, для автоматизації бізнес-процесів, необхідних для функціональної роботи методу зменшення розмірностей векторного простору був обраний формат запису для зчитування даних JSON.

2.4 Висновки до розділу 2

Встановлено, що для автоматизації обробки інформаційних потоків в складі конструкторської надбудови передбачається реалізація наступного переліку бізнес-процесів:

- робота з відображеннями активного документа;
- робота з технічними листами активного документа;
- робота з кімнатами на рівні активного вигляду;
- робота з дверима на рівні активного вигляду;
- робота з підключеними документа типу Revit Links;

- робота з генерацією метаданих;
- робота з відправленням даних на Azure File Storage.

Розроблена структура інформаційної системи дозволить забезпечити надійний фундамент для передачі організованих даних щодо єдиної цифрової моделі поверхів на хмарний сервіс Azure.

Розділ 3

Програмна реалізація

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок

В межах створення практичної моделі надбудови було використано бібліотеку класів від платформи .Net Framework 4.8, що дозволить реалізувати програмно-логістичну структуру інформаційної системи прийняття контрольованих рішень за рахунок методу пониження розмірностей векторного простору. Тому слід зазначити призначення алгоритмічних модулів, елементів користувацького інтерфейсу та файлу маніфесту на рівні їх функціонального взаємозв'язку.

Зв'язок між бізнес-потокми інструментальної надбудови та платформою Revit, що реалізує принципи інформаційного моделювання будівель, розпочинається зі створення файлу маніфесту. Маніфест надбудови це файл XML, розташований у певному місці на жорсткому диску. Він може застосовуватися як до конкретного користувача, так і до всіх в рамках операційної системи Windows. Існують деякі необхідні теги, які важливо вказати як для зовнішньої програми, так і для зовнішньої команди, наприклад, Assembly, FullClassName, ClientId. Зовнішня команда також використовує тексти та описи. Ці атрибути надають ту саму інформацію, що і записи, які використовуються в застарілому механізмі реєстрації надбудови Revit.ini. Однак маніфест надбудови також пропонує велику кількість захоплюючих нових функцій, а тому багато нових необов'язкових тегів можна використовувати для визначення підказки, зображення, видимості, доступності, локалізації та інших функцій.

Файли маніфестів обробляються Revit автоматично, коли вони розміщуються в одному з наступних двох місць у папці «Дані програми» в системі користувача [20]. Усі файли з розширенням імені файлу «.addin», знайдені в цих місцях, будуть задіяні системою будівельного моделювання під час його запуску. Кілька надбудов можуть

завантажуватися одним файлом маніфесту. Синтаксис файлу маніфесту визначає теги, які дозволяють вказати всю інформацію, визначену раніше в реєстраційних даних Revit.ini, тобто шлях збірки, повну назву класу, текст та опис (рис. 3.1).

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RevitAddIns>
3  <AddIn Type="Application">
4    <Name>RevitAddinProject</Name>
5    <Assembly>C:\Users\roman\source\repos\RevitAddinProject\RevitProject.Main\bin\x64\Release\RevitProject.Main.dll</Assembly>
6    <FullName>RevitProject.Main.App</FullName>
7    <ClientId>537055e0-7f48-474a-845b-3e97cf5fded3</ClientId>
8    <VendorId>com.typepad.thebuildingcoder</VendorId>
9  </AddIn>
10 </RevitAddIns>

```

Рисунок 3.1 – структура файлу маніфесту для платформи Autodesk Revit

Основні та базові теги маніфесту архітектурного рішення включають наступні ознаки системних атрибутів:

- **Assembly**. Повний шлях до файлу збірки надбудови. Потрібно додавати для всіх зовнішніх команд та зовнішніх застосувань;

- **FullName**. Повна назва класу у файлі збірки, що реалізує інтерфейси `IExternalCommand` або `IExternalApplication`. Потрібно для всіх зовнішніх команд та зовнішніх застосувань;

- **ClientId**. GUID, який представляє ідентифікатор цієї конкретної програми. `ClientIds` повинні бути унікальними для певного сеансу Revit. Autodesk рекомендує створити унікальний GUID для кожної зареєстрованої програми або команди. Даний атрибут необхідно вказати для всіх зовнішніх команд та зовнішніх застосувань. Властивість `UIApplication.ActiveAddInId` надає програмний доступ до цього значення, якщо це необхідно;

- **Name**. Назва програми. Вимагається; лише для зовнішніх додатків;

- **Text**. Назва кнопки. Не обов'язковий тег, що використовується лише для зовнішніх команд. За замовчуванням «Зовнішній інструмент»;

- **Description**. Короткий опис команди, який застосовується в якості підказка для

генерованого компонента візуалізації. Необов'язковий атрибут, що використовується лише для зовнішніх команд. За замовчуванням являється підказкою з текстом команди.

Після завершення обробки реєстраційного файлу система автоматизованого проектування розпочинає виконання основної сесії, за рахунок якої викликається метод вбудованого інтерфейсу `IExternalApplication`, а саме `OnStartup`. Як правило, алгоритмічна схема даної реалізації відповідає за активацію елементів стартового модуля надбудови (рис. 3.2), що є невід'ємною частиною ініціалізацію компонентів користувацького інтерфейсу, таких як кнопки та одиниці тематичного розподілу.

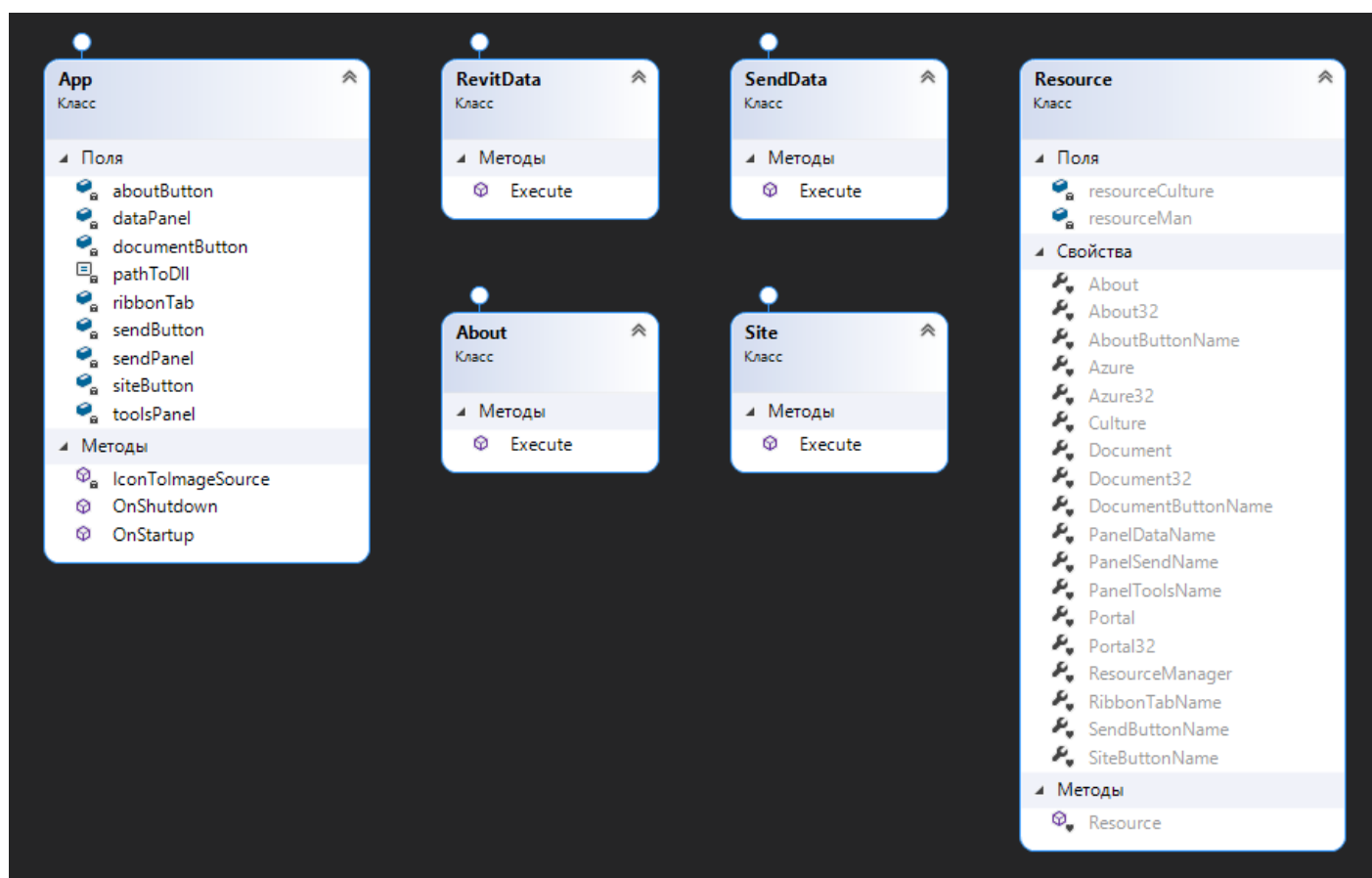


Рисунок 3.2 – діаграма класів стартового програмного модуля

В структурі стартового програмного модуля знаходяться наступні компоненти функціональної реєстрації:

1. «App» це основний клас надбудови, в якому відбувається запуск згаданого методу пониження розмірностей векторного простору. Поданий компонент відповідає за реалізацію глобальних методів `IExternalApplication` та зберігає доступ до готових шаблонів Revit команд;

2. «RevitData» являє собою логістичну команду для запуску бізнес-процесів по роботі з даними активної архітектурно-будівельної 3D моделі та подальшої генерації метаданих з результатом перетворення геометричних параметрів відносно приміщень та дверей. Даний клас є реалізацією інтерфейсу `IExternalCommand`, що викликається у разі натискання на кнопку «Document».

3. «SendData» в першу чергу це функціональний модуль для відправлення результуючих даних від «RevitData» в обліковий запис зберігання інформації File Storage, що відноситься до хмарного сервісу Microsoft Azure. Поданий функціонал є реалізацією інтерфейсу `IExternalCommand`, що викликається у разі натискання на кнопку «Azure».

4. Компоненти «About» та «Site» є частиною конфігурації надбудови, що відповідають за зберігання даних щодо версії підключеного забезпечення та маршрутизації користувача Autodesk Revit в простори хмарного сервісу Microsoft Azure.

Наступним модулем в структурі архітектурного рішення для роботи з даними геометричних примітивів є сукупність методів та їх об'єктно-орієнтованих залежностей, які дозволяють згенерувати елементи WPF, тобто компоненти графічного користувацького інтерфейсу (рис. 3.3). Такими в рамках існуючої програмної системи є:

– «MainWindow» це графічне вікно для відображення елементів активного документа, результат виконання команди «RevitData», що забезпечує безпосередню роботу з функціями перегляду, сортування та інтерактивної взаємодії з моделлю Revit;

– «AzureWindow» являється екземпляром графічних результатів виконання команди «RevitData», що дозволяє активному користувачу платформи інформаційного моделювання будівель здійснити відправлення вибраної інформації в обліковий запис

зберігання даних на Microsoft Azure.

– «Helpers», «Node» та «LinksModel» представлені в якості допоміжних інструментів для реалізації базових можливостей вікна «MainWindow».

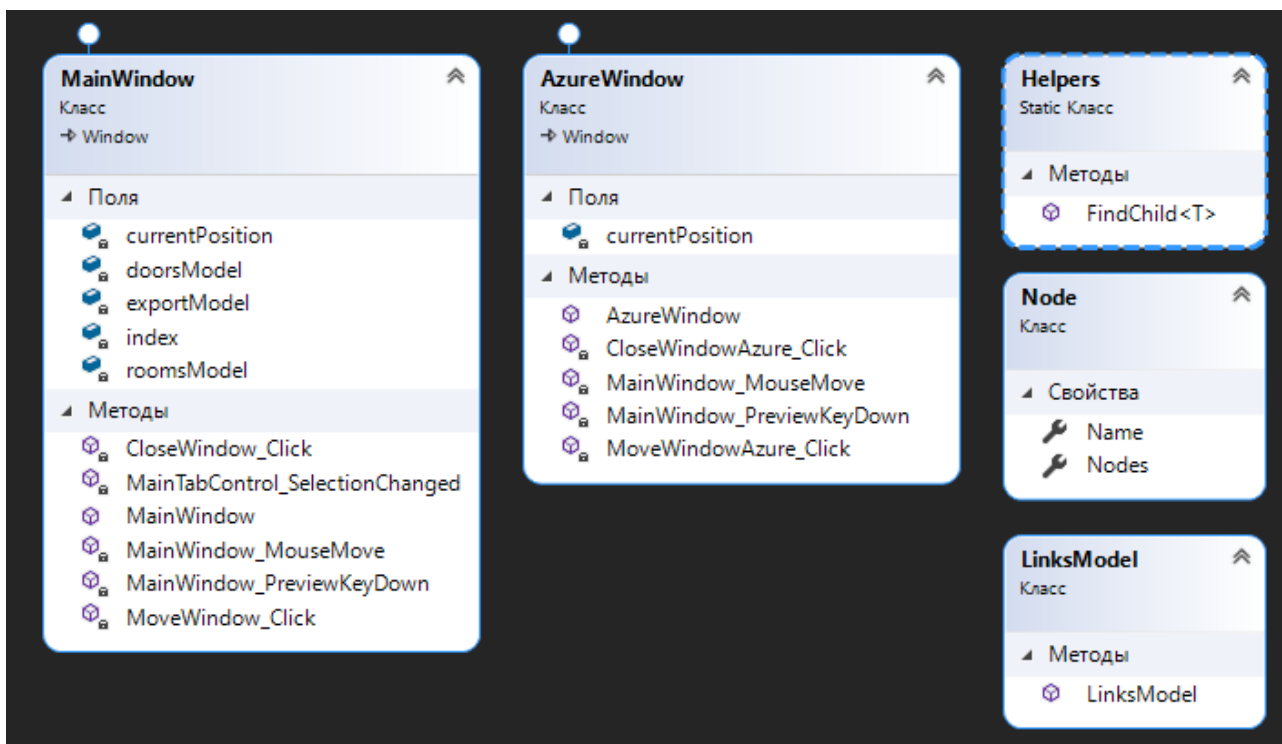


Рисунок 3.3 – діаграма класів WPF модуля

Основним інформаційним модулем в забезпеченні належного функціонування бізнес-потоків являється комплексний набір класів, який відповідає за автоматизацію обробки даних від програмного інтерфейсу Revit API (рис. 3.4). Базовим шаблоном в організації інформаційних пакетів являється абстрактний клас «RevitModel», що зберігає методи опрацювання над елементами активного документа, такими як вигляди, технічні листи, двері та кімнати. Класи, що успадковуються від вищезгаданого шаблону:

– «FloorPlansModel» це модель даних, яка забезпечує обробку даних відносно об'єктів типу Floor Plan View. На основі обраного вигляду здійснюється активація моделей «RoomsModel» та «DoorsModel»;

– «ViewSheetModel» представляється в якості функціонального посередника між

процесами інтерактивної взаємодії з листами документа Revit та запуску модуля зберігання та генерації метаданих;

– «RoomsModel» і «DoorsModel» це моделі даних для роботи з примітивами кімнат та дверей на рівні обраного вигляду, що дозволяють використовувати функції перегляду та сортування відповідних даних;

– «ExportModel» являється базовим компонентом для роботи з методами інформаційної технології перетворення координат та генерацією метаданих щодо задіяних на листах екземплярів дверей та кімнат, їх геометричного зв'язку;

– «AzureModel» це модель даних для роботи з результатами виконання «ExportModel», що забезпечує виконання функцій відправлення та видалення метаданих.

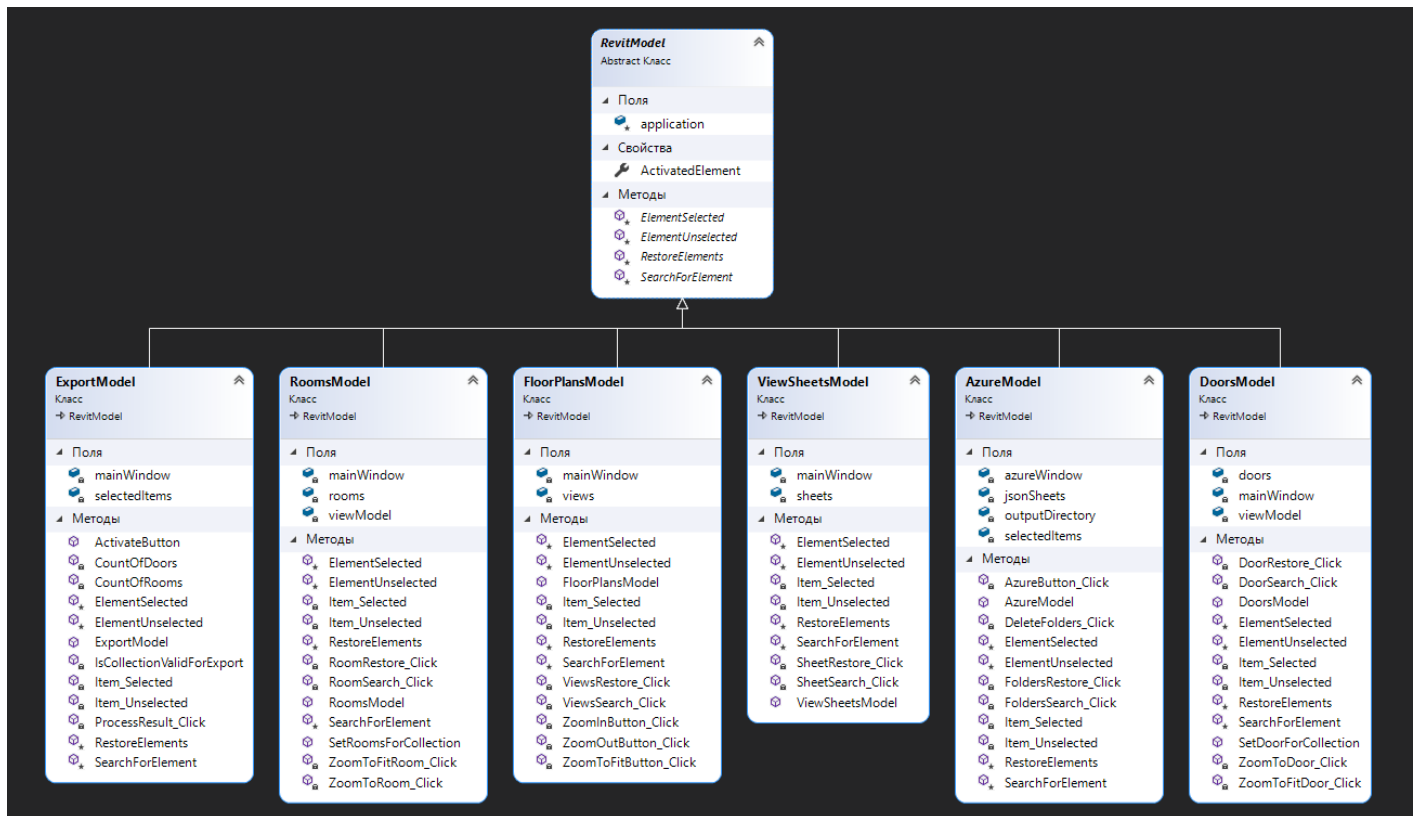


Рисунок 3.4 – діаграма класів моделі даних

Для забезпечення автоматизації функцій інструментальної надбудови в платформі архітектурного проектування Autodesk Revit був реалізований модуль зберігання та генерації метаданих (рис. 3.5).

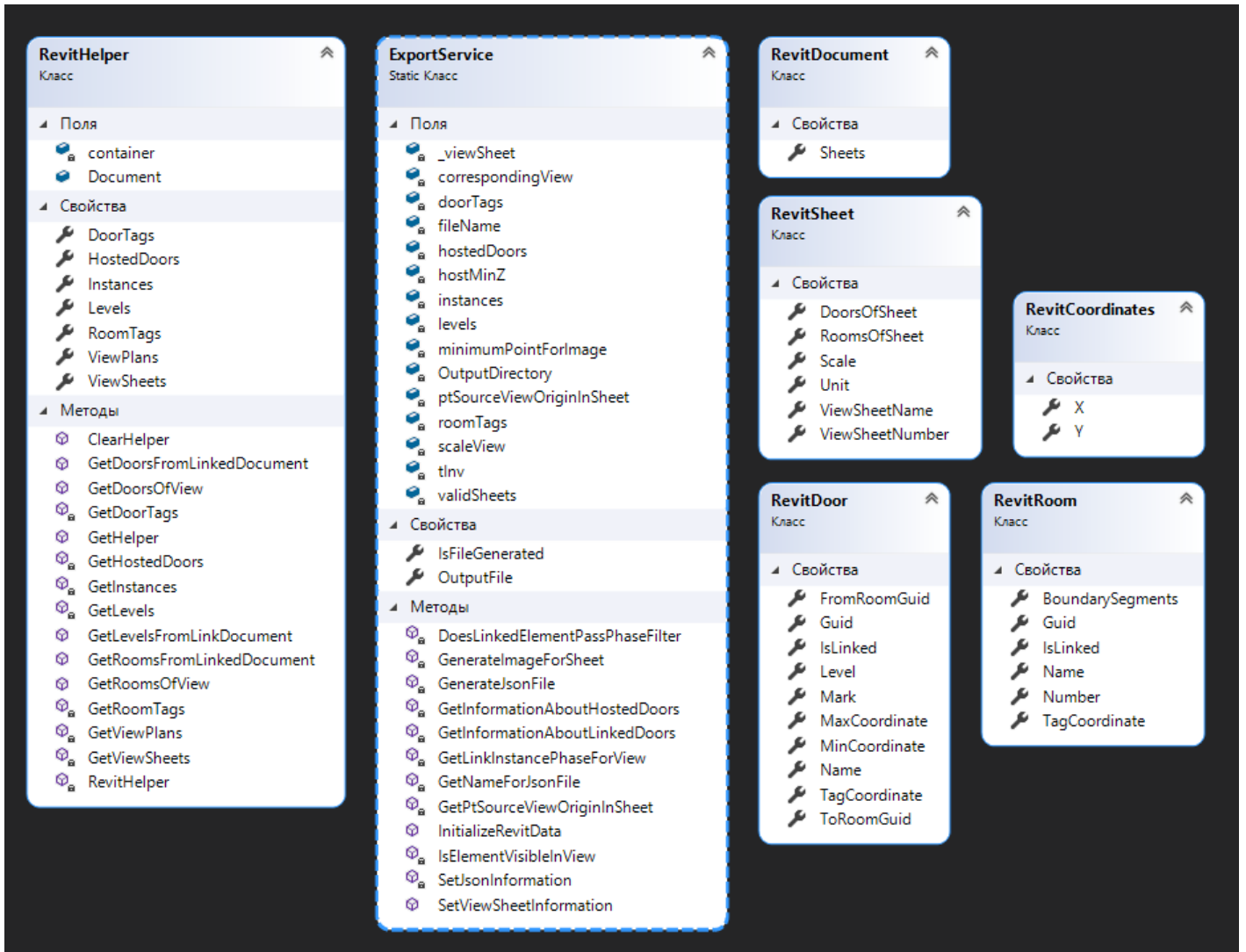


Рисунок 3.5 – діаграма класів модуля зберігання та генерації метаданих

В поданій структурі реалізовані наступні програмні компоненти:

– «RevitHelper» це клас-помічник, який дозволяє використовувати функції вбудованого програмного інтерфейсу Revit API. Наприклад, методи для виклику колекції дверей на рівні активного вигляду чи отримання інформації щодо поверхів;

– «RevitDocument», «RevitSheet», «RevitDoor», «RevitRoom», «RevitCoordinates» це базові шаблони серіалізації, що забезпечують можливості зберігання результуючих даних відносно математичного перетворення координат дверей та кімнат з модельного представлення до системи координат паперового простору;

– «ExportService» це клас, в якому реалізовані методи для роботи з перетворенням параметрів моделей при трансфері між системами візуалізації. Це набір викликів для обробки даних відносно кімнат та дверей, обрахунків їх взаємозв'язку з точки зору геометричної конфігурації та генерації оновлених координат на рівні об'єктів типу View Sheet.

Таким чином, проаналізувавши структуру та функціональне призначення фрагментів надбудови, можна дійти до висновку, що для формування методу пониження розмірностей векторного простору необхідно розробити задані програмні модулі та реалізувати можливості роботи з текстовим форматом зберігання та зчитування інформації, їх відправлення на ресурси хмарного сервісу Microsoft Azure.

3.2 Розробка програмних модулів

Розробка програмних модулів конструкторської надбудови для системи автоматизованого проектування Autodesk Revit передбачає деталізований виклад можливостей інформаційної забезпечення, а саме алгоритм взаємодії програмного забезпечення з BIM платформою, отримання та обробка даних Revit API відносно робочого документу, включаючи сутності виглядів та технічного листа, та генерації результуючих метаданих з можливістю їх відправлення в обліковий запис зберігання інформації.

За ознаками автоматизації інформаційних потоків функції архітектурного рішення можна розподілити за наступної класифікації: робота з відображеннями активного документа, робота з кресленням активного документа, робота з кімнатами на

рівні робочого вигляду, робота з дверима на рівні робочого вигляду, робота з підключеними документами, робота з генерацією метаданих та робота з відправленням даних на Azure File Storage.

1) бізнес-процес стосовно управління виглядами типу Floor Plan розпочинається з ініціалізації даних від програмного інтерфейсу Revit API, чий результат виконання зберігає колекцію об'єктів заданої класифікації елементів.

Для більш якісної роботи з архітектурно-будівельними відображеннями функціями програмного забезпечення розроблена можливість переглядати актуальну інформацію щодо вибраного об'єкта зі списку доступних виглядів. До числа таких нововведень відносяться наступні параметри:

- назва вигляду (Name of View);
- конкретний рівень вигляду (Detail Level);
- дисципліна вигляду (Discipline);
- назва робочої моделі будівлі або документа (Document Name);
- межі паперового простору (Outline of View);
- масштабування вигляду (Scale of View);
- тип активного відображення (Type of View);
- підтвердження, чи відноситься вигляд до типу assembly (IsAssemblyView);
- підтвердження, чи являється вигляд шаблонним (IsTemplate);
- підтвердження, чи є відображення перехідним або тимчасовим (IsTransient);
- назва поверху, до якого відноситься вигляд (Name of GenLevel).

2) наступним бізнес-процесом з набору можливостей надбудови представлена робота з технічними листами, які відносяться до категорії сутностей активного документу, що функціональні для подальшої печаті. Як і в випадку з відображеннями ініціалізація списку об'єктів типу ViewSheet розпочинається з виклику необхідного методу Revit API. Для більш комфортнішого управління елементами даної класифікації функціями архітектурного програмного забезпечення передбачена аналогічна

можливість переглядати інформацію відносно існуючих креслень на рівні робочої 3D моделі будівлі. Такими параметрами є наступні:

- назва технічного листа (Name of Sheet);
- кількість сутностей типу Viewport (Count of Viewports);
- початок паперової системи координат листа (Origin of Sheet);
- назва робочої моделі будівлі або документа (Document Name);
- номер креслення (Number of Sheet);
- масштабування активного листа (Scale of Sheet);
- категорія активного листа (Category of Sheet);
- підтвердження, чи дійсне креслення для генерації метаданих (JsonValid);
- підтвердження, чи відноситься вигляд до типу assembly (IsAssemblyView);
- підтвердження, чи являється вигляд шаблонним (IsTemplate);
- підтвердження, чи є відображення перехідним або тимчасовим (IsTransient);

3) при роботі з елементами приміщень, кімнат та геометричних секцій в рамках активної моделі враховується оптимізаційне використання методів сегментованого розподілення. На відміну від загального виклику Revit API по заданим класифікаціям колекція об'єктів типу Room проходить ініціалізацію за рахунок програмних засобів фільтрації на рівні обраного вигляду, що зберігає посилання на конкретний поверх чи логічний рівень.

Функціями даного бізнес-процесу запланований перегляд актуальних даних відносно активної кімнати. До списку такої інформації відносяться наступні змінні:

- назва активної кімнати (Name of Room);
- площа геометричної секції (Area of Room);
- рівень кімнати (Level of Room);
- назва робочої моделі будівлі або документа (Document Name);
- номер активної кімнати (Number of Room);
- об'єм обраної секції (Volume of Room);

- категорія активної кімнати (Category of Room);
- підтвердження, чи являється кімната дійсною (IsValid);
- фаза, в якій геометрична сутність була створена (Phase of Room);
- периметр обраної секції (Perimeter of Room);

4) бізнес-процес по роботі з дверима та їх різновидами на функціональному рівні дещо схожий і його колекція подібно з абстрактними сегментами відбувається за допомогою методів сортування на рівні логічного відображення.

Для більш якісного управління елементами заданої класифікації функціями конструкторської надбудови підготовлена можливість переглядати інформацію в такій послідовності:

- назва обраних дверей (Name of Door);
- рівень дверей (Level of Door);
- назва робочої моделі будівлі або документа (Document Name);
- назва кімнати, що є вихідною по відношенню до дверей (From Room);
- назва кімнати, що є вхідною по відношенню до дверей (To Room);
- категорія класифікаційної сутності (Category of Door);
- підтвердження, чи являються двері функціональними (IsValid);
- назва фази, в якій указані двері буди згенеровані (Phase of Door);
- підтвердження, чи невидима обрана геометрична сутність на відображенні (IsInvisible).

5) автоматизація інформаційних потоків програмної системи не обмежується функціональною структурою, що працює за рахунок обробки елементів активного робочого документа. У випадку, якщо користувач BIM платформи здійснить підключення інших моделей за посиланням, надбудова зможе їх відобразити в деревоподібному форматі.

6) основою надбудови задля використання функцій перетворення параметрів архітектурно-будівельних 3D моделей є бізнес-процес, що дозволяє побудувати

метадані елементів вибраних приміщень та відповідних складових з їх подальшим відправленням в хмарний сервіс зберігання інформації. На цьому етапі свою роль відіграє спосіб пониження розмірностей векторного простору, який відповідає за отримання паперових координат сутностей з оригінальної модельної системи координат. Основні методи для опрацювання вищезгаданих функцій наведено на рисунках 3.6 та 3.7.

```

87     ссылка: 1
88     private void ProcessResult_Click(object sender, RoutedEventArgs e)
89     {
90         var sheets = RevitHelper.GetHelper().ViewSheets.Where(sheet => selectedItems.Any(item => item.Content.Equals(sheet.Title))).ToList();
91         ExportService.InitializeRevitData(sheets.Count);
92         ExportService.SetViewSheetInformation(sheets);
93
94         if (ExportService.IsFileGenerated)
95             mainWindow.StatusInfo.Text = "Data generated successfully!";
96         else
97             mainWindow.StatusInfo.Text = "Something went wrong while generating data!";
98     }

```

Рисунок 3.6 – алгоритм початку роботи с сервісом експорту метаданих

```

150     ссылка: 1
151     private static List<RevitDoor> SetJsonInformation(Viewport viewport, out List<RevitRoom> jsonRooms)
152     {
153         jsonRooms = new List<RevitRoom>();
154
155         scaleView = correspondingView.Scale;
156         ptSourceViewOriginInSheet = GetPtSourceViewOriginInSheet(viewport);
157
158         List<RevitDoor> doorForJson = new List<RevitDoor>();
159
160         doorForJson.AddRange(GetInformationAboutHostedDoors(out List<RevitRoom> roomsJsonHosted));
161         if (roomsJsonHosted.Count != 0)
162             jsonRooms.AddRange(roomsJsonHosted);
163
164         doorForJson.AddRange(GetInformationAboutLinkedDoors(out List<RevitRoom> roomsJsonLinked));
165         if (roomsJsonLinked.Count != 0)
166             jsonRooms.AddRange(roomsJsonLinked);
167
168         return doorForJson;
169     }

```

Рисунок 3.7 – базовий метод для отримання результируючих метаданих

7) у випадку отримання метаданих щодо перетворених координат геометричних секцій та взаємопов'язаних дверей, що були опрацьовані на рівні паперового простору обраних користувачем технічних листів, функціями конструкторської надбудови

підготовлена можливість відправити локальні файли та зображення типу ViewSheet в обліковий запис зберігання інформації від платформи Microsoft Azure. Алгоритм роботи представлено на рисунках 3.8.

```

11  Ссылка: 0
12  public static void Main(string[] args)
13  {
14      var arguments = string.Join(" ", args).Split("~");
15      if (FileStorageManager.InitializeFileStorage("fileshare"))
16      {
17          string[] directories = Directory.GetDirectories(outputDirectory);
18          directories = directories.Where(directory => arguments.Contains(new DirectoryInfo(directory).Name)).ToArray();
19          foreach (string directory in directories)
20          {
21              if (FileStorageManager.DirectoryExists(new DirectoryInfo(directory).Name))
22                  FileStorageManager.DeleteDirectory(new DirectoryInfo(directory).Name);
23
24              FileStorageManager.CreateDirectory(new DirectoryInfo(directory).Name);
25              string[] files = Directory.GetFiles(directory);
26              foreach (string file in files)
27              {
28                  byte[] bytes = File.ReadAllBytes(file);
29                  FileStorageManager.Write(new DirectoryInfo(directory).Name, new FileInfo(file).Name, bytes);
30              }
31          }
32      }

```

Рисунок 3.8 – алгоритм генерації файлової інфраструктури в системі File Storage

3.3 Висновки до розділу 3

Отже, проаналізувавши розробку програмних модулів надбудови та виконавши конструктивний опис алгоритмічних конструкцій архітектурного рішення в інфраструктурі системи автоматизованого проектування Autodesk Revit, можна дійти до висновку, що інформаційна модель відповідає наступним вимогам:

1. Розширенню функціональних можливостей BIM платформи;
2. Вищезазначеним бізнес-процесам для автоматизації інформаційних потоків;
3. Можливості переходу до пункту експериментальної апробації.

Розділ 4

Апробація програмної інформаційної системи

4.1 Основні функції архітектурно-будівельної надбудови

З метою перевірки та підтвердження результатів виконання методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за допомогою пониження розмірностей векторного простору було проведено апробацію програмної системи. Згідно поставленого завдання та проектування структури архітектурного рішення для платформи інформаційного моделювання будівель Autodesk Revit було встановлено, що інструментальна надбудова має наступний перелік функціональних можливостей:

- робота з відображеннями активного документу (вивести список відображень документа, перегляд даних щодо активного вигляду, інтерактивна взаємодія через Revit: функції підключення та масштабування відносно активного вигляду та пошук об'єктів зі списку доступних відображень);

- робота з технічними листами активного документу (вивести список креслень документа, перегляд даних щодо активного креслення, інтерактивна взаємодія з Revit: функція підключення активного інженерного листа та пошук об'єктів зі списку доступних інженерних листів);

- робота з кімнатами на рівні активного вигляду (вивести список доступних кімнат, перегляд даних щодо активної кімнати, інтерактивна взаємодія з Revit: функція масштабування до активної кімнати та пошук об'єктів зі списку доступних кімнат);

- робота з дверима на рівні активного вигляду (вивести список доступних дверей, перегляд даних щодо активної двері, інтерактивна взаємодія з Revit: функція масштабування до активної двері та пошук об'єктів зі списку доступних дверей);

- робота з підключеними документами (вивести дерево рішень для підключених документів та вивести кімнати і двері для кожного підключеного документа);

- робота з генерацією метаданих (вивести список технічних листів на рівні активного документа, надати користувачу можливість вибирати креслення перед їх відправленням на ресурси хмарного сервісу Azure, перегляд актуальної інформації щодо створених даних та на основі вибраного креслення згенерувати метадані та їх картинки);
- робота з відправленням даних на Azure File Storage (переглянути інформацію щодо списку метаданих для відправки, надати користувачу можливість видалити метадані та відправлення створених метаданих на Azure File Storage).

Виконання бізнес-процесу по роботі з відображеннями будівельної 3D моделі показано на рисунку 4.1. Встановлено, що кінцевий користувач платформи Autodesk Revit отримує можливість переглядати параметричні дані відносно всіх існуючих об'єктів типу View Plan. Згадувані сутності відіграють значиму роль в відображенні геометричної конфігурації для сегментів та дверей на рівні модельної системи координат.

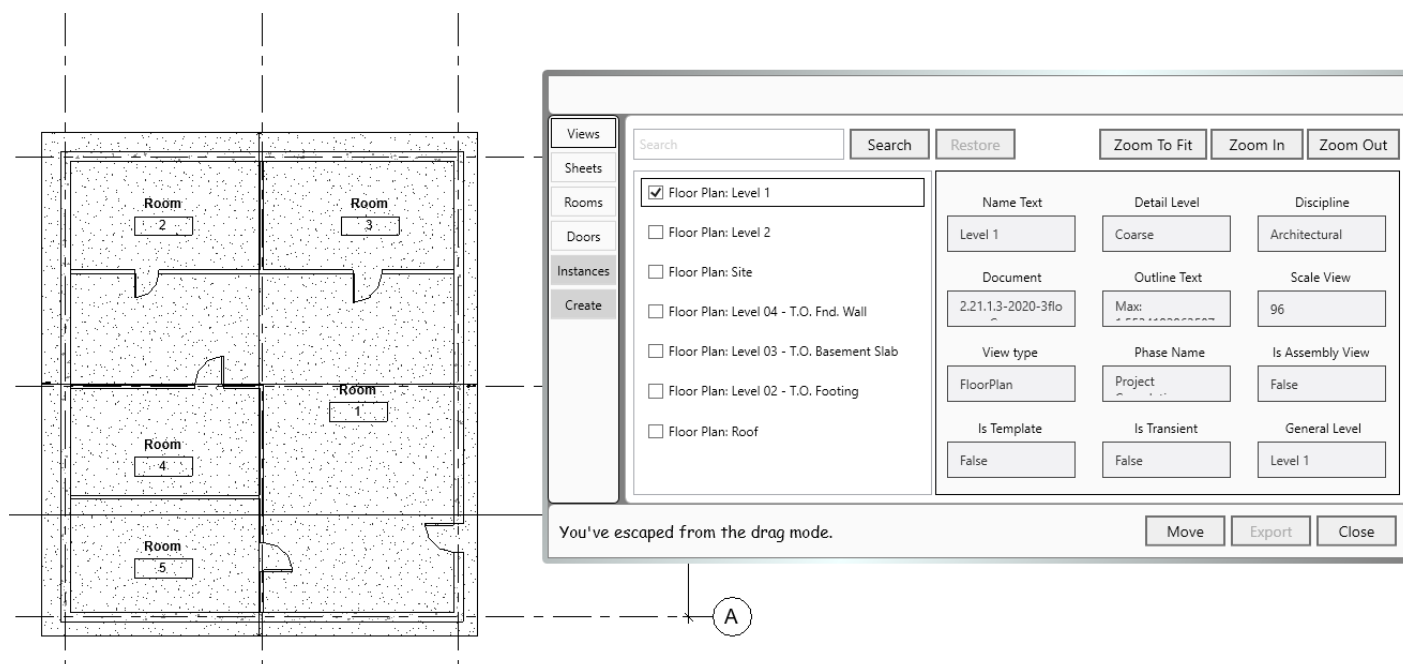


Рисунок 4.1 – результат виконання бізнес-процесу по роботі з відображеннями

Виконання бізнес-процесу по роботі з технічним кресленням на рівні активного документа зображено на рисунку 4.2. У випадку, якщо користувач архітектурної надбудови перейде в розділ існуючого креслення, тоді функціями програмного застосування стане відомо, чи підходить для генерації метаданих обраний View Sheet.

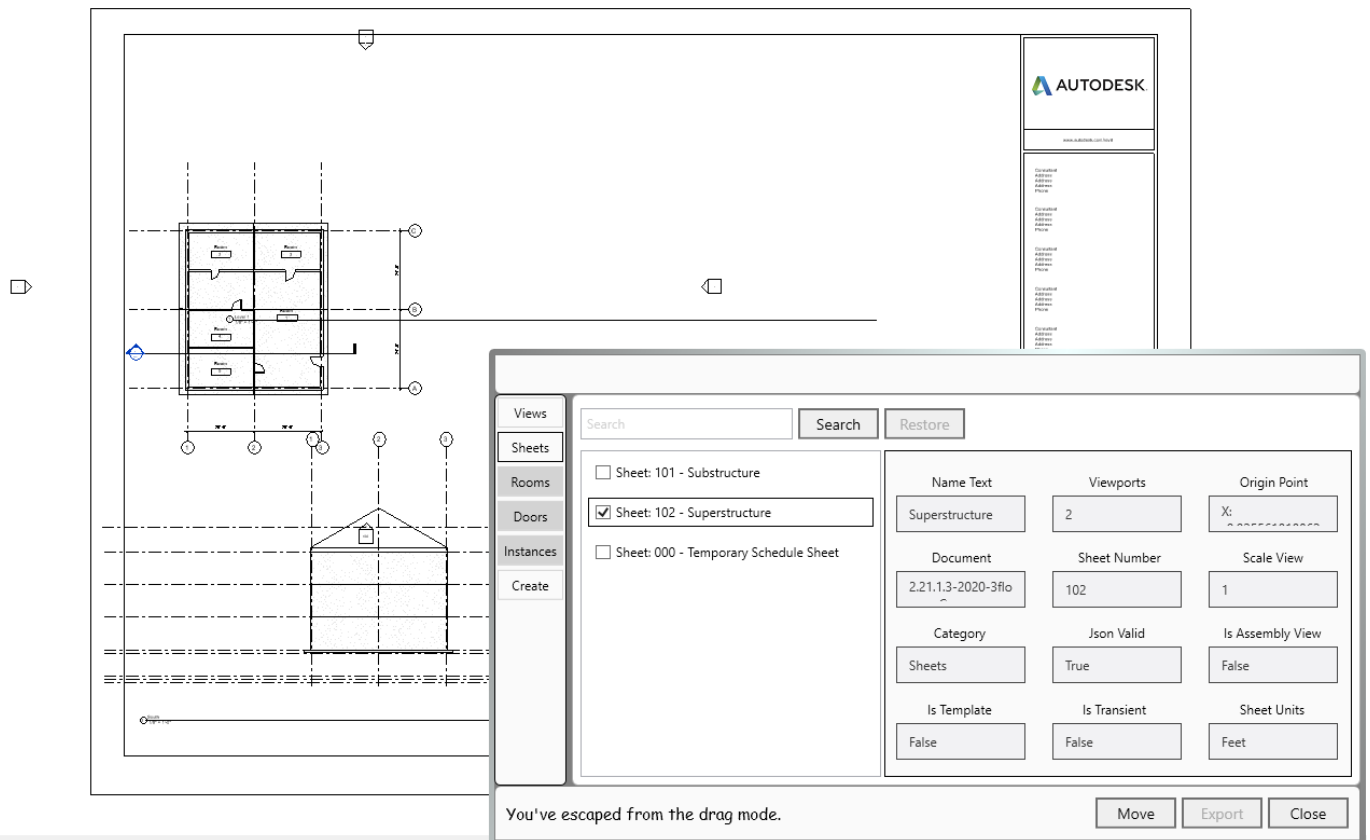


Рисунок 4.2 – результат виконання бізнес-процесу по роботі з кресленням документа

Виконання бізнес-процесу по роботі з кімнатами на рівні активної моделі показано на рисунку 4.3. Для того, щоб перейти в розділ для перегляду інформації стосовно геометричних секцій, кінцевому користувачу необхідно активувати вигляд при роботі з відображеннями та планами поверхів. Функціями надбудови передбачено, що колекція існуючих кімнат та їх параметричних даних створюється за умов вбудованих засобів фільтрації та сортування. При виборі конкретної кімнати встановлено, що

засобами інструментальної надбудови з об'єктом можна взаємодіяти за допомогою можливостей інтегрованого масштабування.

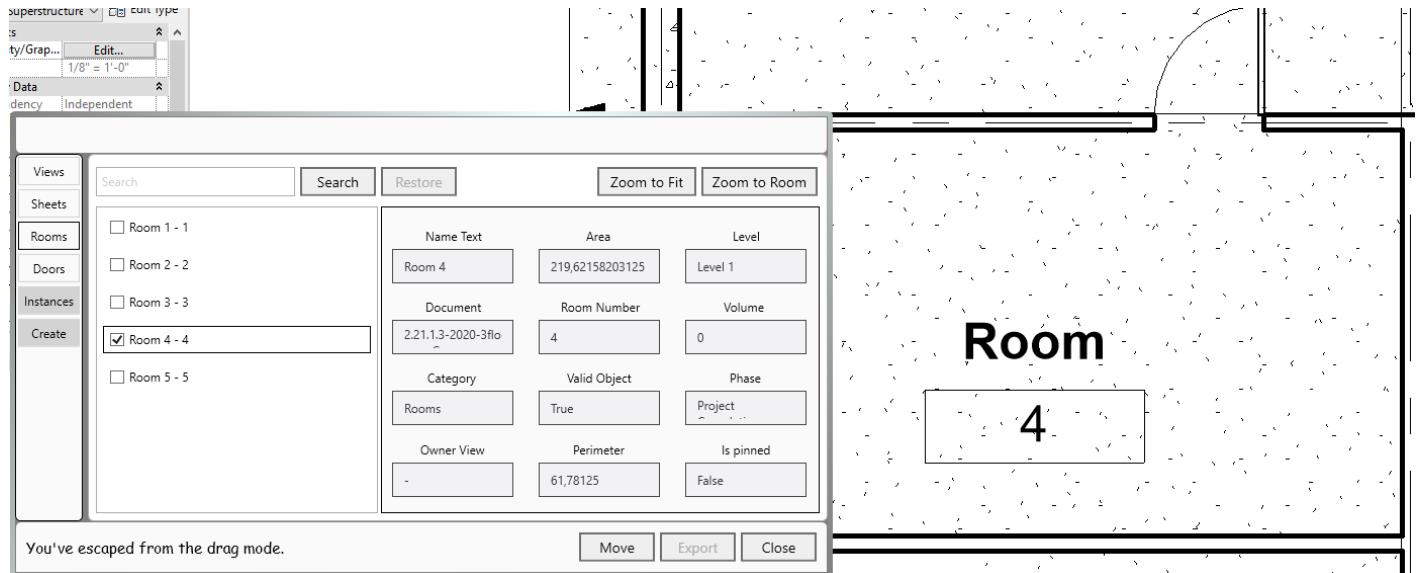


Рисунок 4.3 – результат виконання бізнес-процесу по роботі з кімнатами

Виконання бізнес-процесу по роботі з дверима на рівні активного відображення представлено на рисунку 4.4. Як і у випадку з геометричними секціями, користувачу системи для перегляду інформації щодо існуючих дверей потрібно активувати вигляд, на якому повинен бути задіяний хоча б один елемент типу Family Instance.

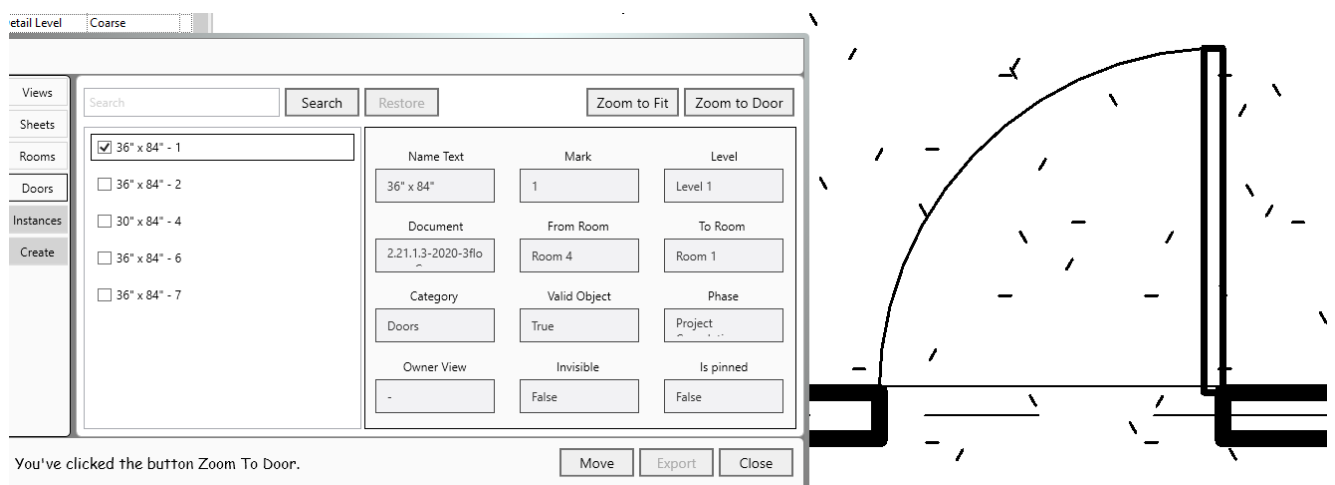


Рисунок 4.4 – результат виконання бізнес-процесу по роботі з дверима

Виконання бізнес-процесу по роботі з генерацією метаданих зображено на рисунку 4.5. Для переходу в згадуваний розділ кінцевому користувачу необхідно ознайомитися зі списком існуючого технічного креслення. На процеси математичних розрахунків та серіалізації результуючих даних можуть бути відправлені лише ті листи, де були добавлені сутності Viewport з елементами кімнат та дверей.

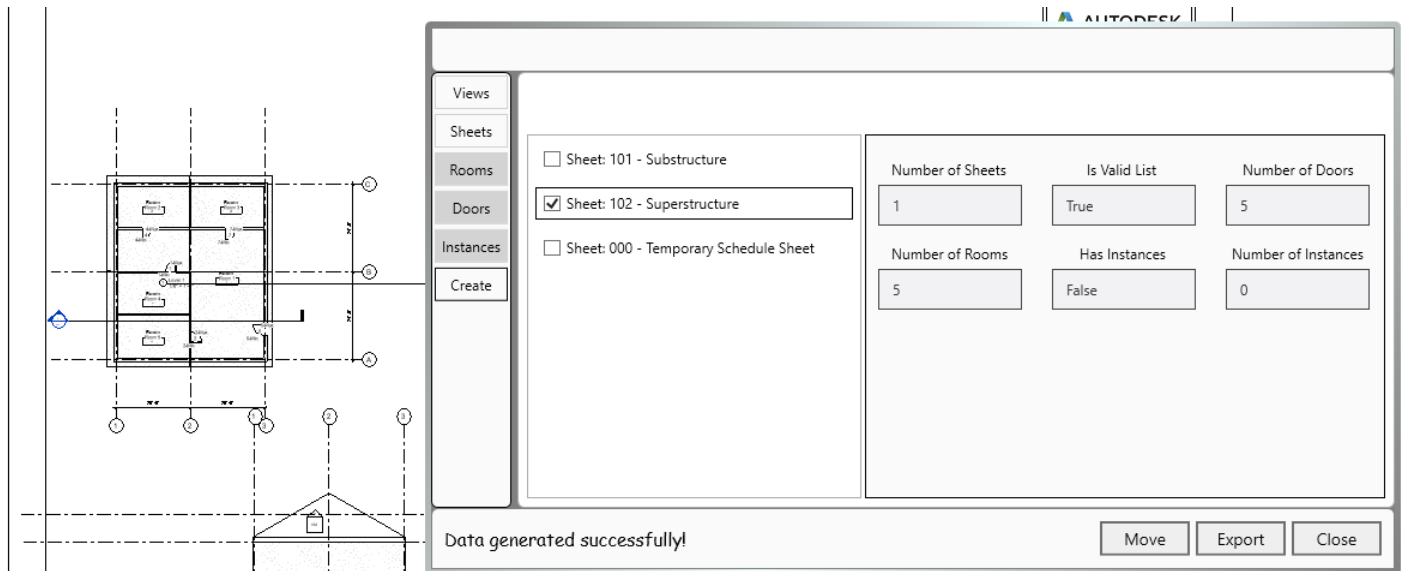


Рисунок 4.5 – результат виконання бізнес-процесу по роботі з генерацією метаданих

Бізнес-процес по роботі з відправленням метаданих в обліковий запис зберігання інформації Azure File Storage в складі хмарного сервісу Microsoft Azure показано на рисунках 4.5-4.6.

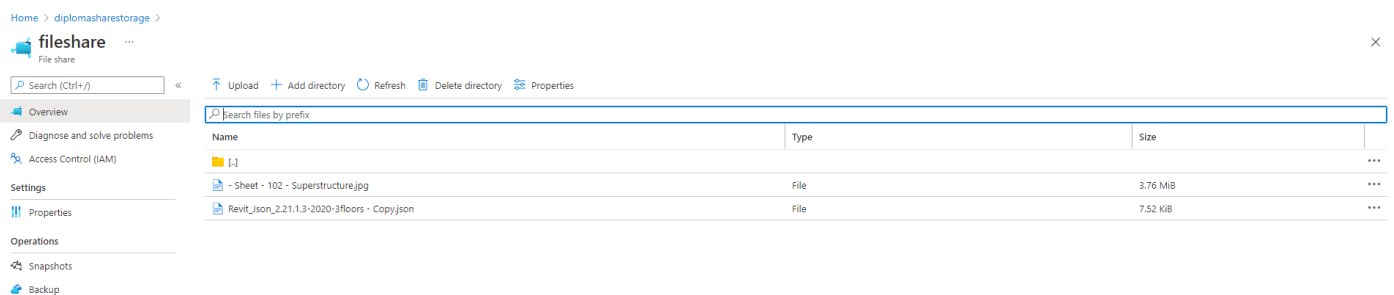


Рисунок 4.6 – відправлення результатів обробки на Azure

У випадку, якщо задача трансферу результуючої інформації завершиться успішно, кінцевий користувач отримує можливість дізнатися про це за рахунок відповідного статусу від модуля Azure. Для переходу в простір хмарного сервісу достатньо натиснути кнопку Portal.

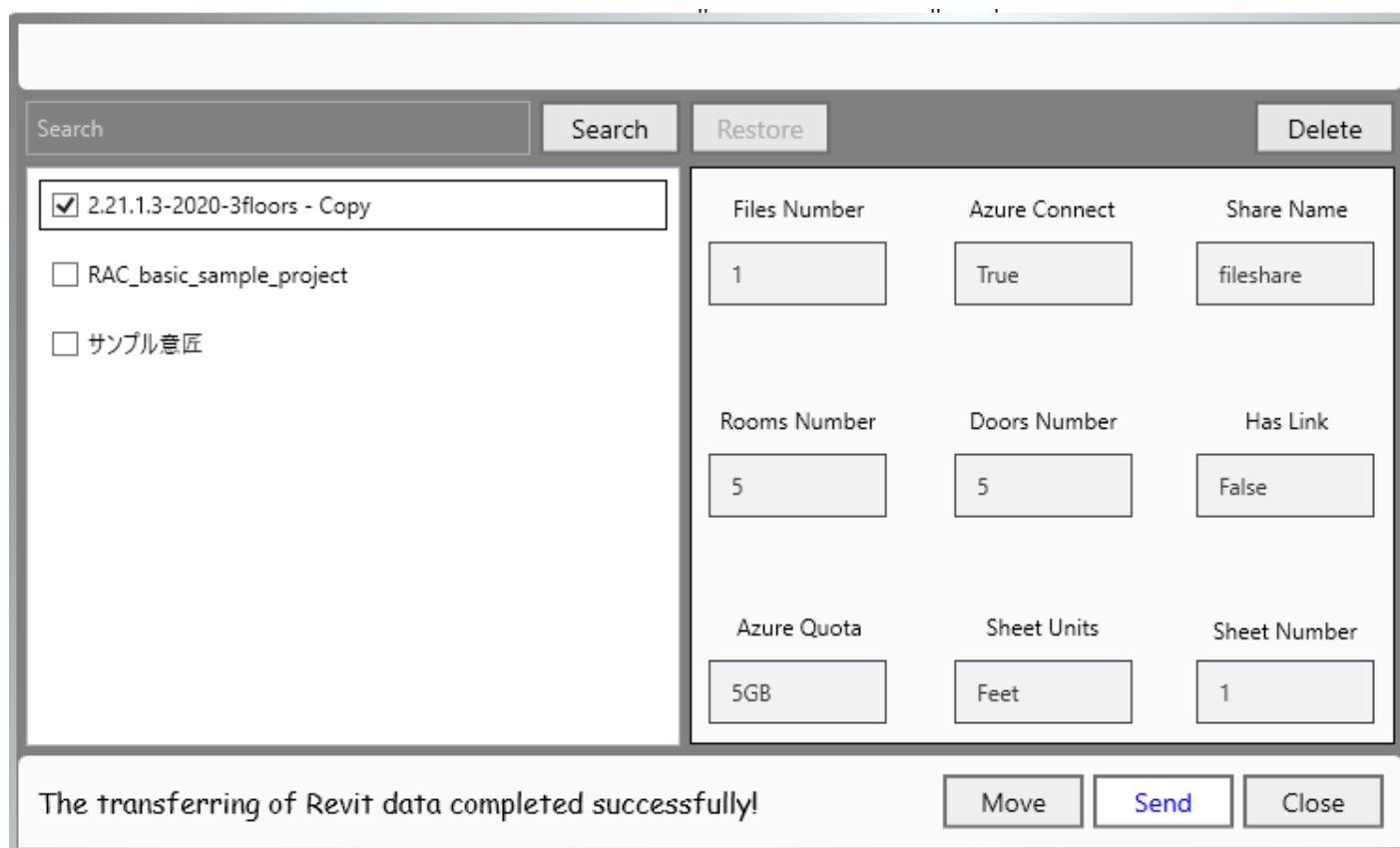


Рисунок 4.6 – результат виконання бізнес-процесу по роботі з відправленням даних на Azure File Storage

Отже, тестуванням інформаційних потоків в складі конструкторської надбудови, що слугує в якості кількісного інструментарію для роботи з характеристиками будівельних елементів, було встановлено, що функціонування програмного забезпечення повністю відповідає умовам користувацького використання.

Для перевірки методу пониження розмірностей векторного простору з подальшим перетворенням параметрів архітектурних 3D моделей було розглянуто три

типових проекти Autodesk Revit: модель приватного будинку, модель житлового комплексу та модель 3-х поверхового шкільного закладу. Статистичні результати інформаційного аналізу будуть подані у вигляді таблиць та зображень.

4.2 Апробація методу на прикладі моделі приватної власності

На рисунку 4.7 зображена модель приватної власності, яка відкрита в платформі Autodesk Revit версії 2021 року. Поданий проект будівлі поставляється за замовчуванням при інсталяції системи автоматизованого проектування в сфері архітектурного конструювання.



Рисунок 4.7 – 3D вигляд проекту «RAC_basic_sample_project.rvt»

В складі існуючого BIM рішення спробуємо виконати експорт даних для технічного листа під назвою «A102 - Plans». На рисунку 4.8 представлена статистика обраної специфікації. Встановлено, що до перетворення параметрів підготовлені наступні об'єкти:

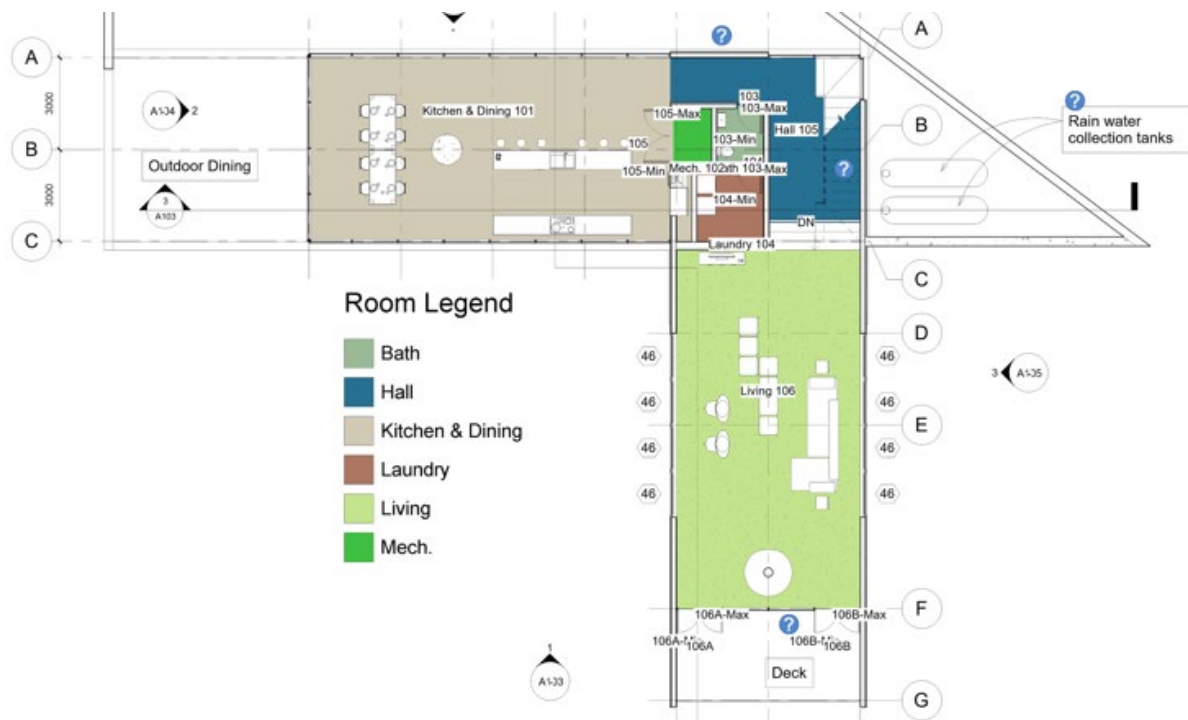


Рисунок 4.9 – розташування міток елементів на першому поверсі

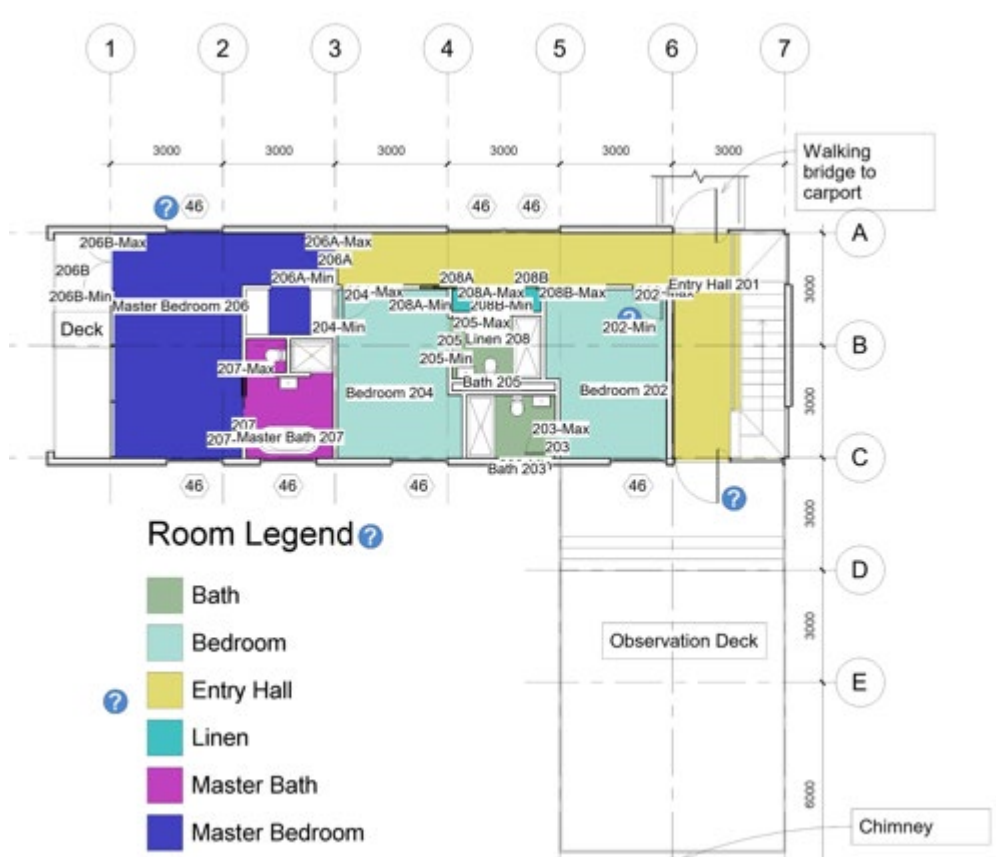


Рисунок 4.10 – розташування міток елементів на другому поверсі

Розглянемо структуру згенерованих метаданих щодо технічної специфікації, яка була оброблена для демонстраційного трансферу. Результати статистичного аналізу щодо деяких екземплярів типу Family Instance та Room представлено в таблицях 4.1-4.4.

Таблиця 4.1 – об'єкт з колекції «sheets» та його параметри

| Назва | Значення | Опис |
|-------------|----------|----------------------------------|
| sheetNumber | A102 | Інформація про номер листа |
| sheetName | Plans | Інформація про назву листа |
| scale | 1 | Інформація про масштабування |
| unit | Feet | Інформація про одиниці координат |

Таблиця 4.2 – об'єкт з колекції «doors» та його параметри

| Назва | Значення | Опис |
|----------|---|---|
| guid | 59371552-5800-43eb-9ba3-609565158fc5-00067242 | Ідентифікатор дверей |
| name | 800 x 2100 | Інформація про назву дверей |
| mark | 103 | Інформація про марку дверей |
| isLinked | false | Перевірка стосовно підключення |
| level | Level 1 | Інформація про поверх дверей |
| max | 0.909419254709962, 1.53374565318146 | Максимальні координати дверей в системі координат паперового простору |
| min | 0.877923191717836, 1.49988309630879 | Мінімальні координати дверей в системі координат паперового простору |
| tag | 0.893671223213899, 1.54627822813587 | Координати тега або центра дверей |
| toRoom | e6ac360b-aaed-4c3b-a130-36b4c2ac9d13-000d1502 | Ідентифікатор вхідної кімнати |

| | | |
|----------|---|--------------------------------|
| fromRoom | ебас360b-aaed-4c3b-a130-36b4c2ac9d13-000d15c9 | Ідентифікатор вихідної кімнати |
|----------|---|--------------------------------|

Таблиця 4.3 – об'єкт «toRoom» з колекції «rooms» та його параметри

| Назва | Значення | Опис |
|----------|---|---|
| guid | ебас360b-aaed-4c3b-a130-36b4c2ac9d13-000d1502 | Ідентифікатор кімнати |
| name | Bath 103 | Інформація про назву кімнати |
| number | 103 | Інформація про номер кімнати |
| isLinked | false | Перевірка стосовно підключення |
| tag | 0.881726464781197, 1.4694429179666 | Координати тега або локальної позиції геометричної секції |

Таблиця 4.4 – об'єкт «fromRoom» з колекції «rooms» та його параметри

| Назва | Значення | Опис |
|----------|---|---|
| guid | ебас360b-aaed-4c3b-a130-36b4c2ac9d13-000d15c9 | Ідентифікатор кімнати |
| name | Hall 105 | Інформація про назву кімнати |
| number | 105 | Інформація про номер кімнати |
| isLinked | false | Перевірка стосовно підключення |
| tag | 0.94418597660812, 1.51054302134542 | Координати тега або локальної позиції геометричної секції |

Таким чином, проаналізувавши результати виконання експорту геометричних даних для технічного листа «A102 - Plans» в складі 3D моделі приватного будинку «RAC_basic_sample_project.rvt», можна дійти до висновку, що конструкторська надбудова розрахувала координати паперового простору без жодної похибки.

4.3 Апробація методу на прикладі моделі житлового комплексу

Наступним проектом для тестування результатів виконання методу збору, відображення, опрацювання, зберігання, трансферу та доступу до технічної специфікації шляхом пониження розмірностей векторного простору являється 3D модель житлового будинку, що зображена на рисунку 4.11.

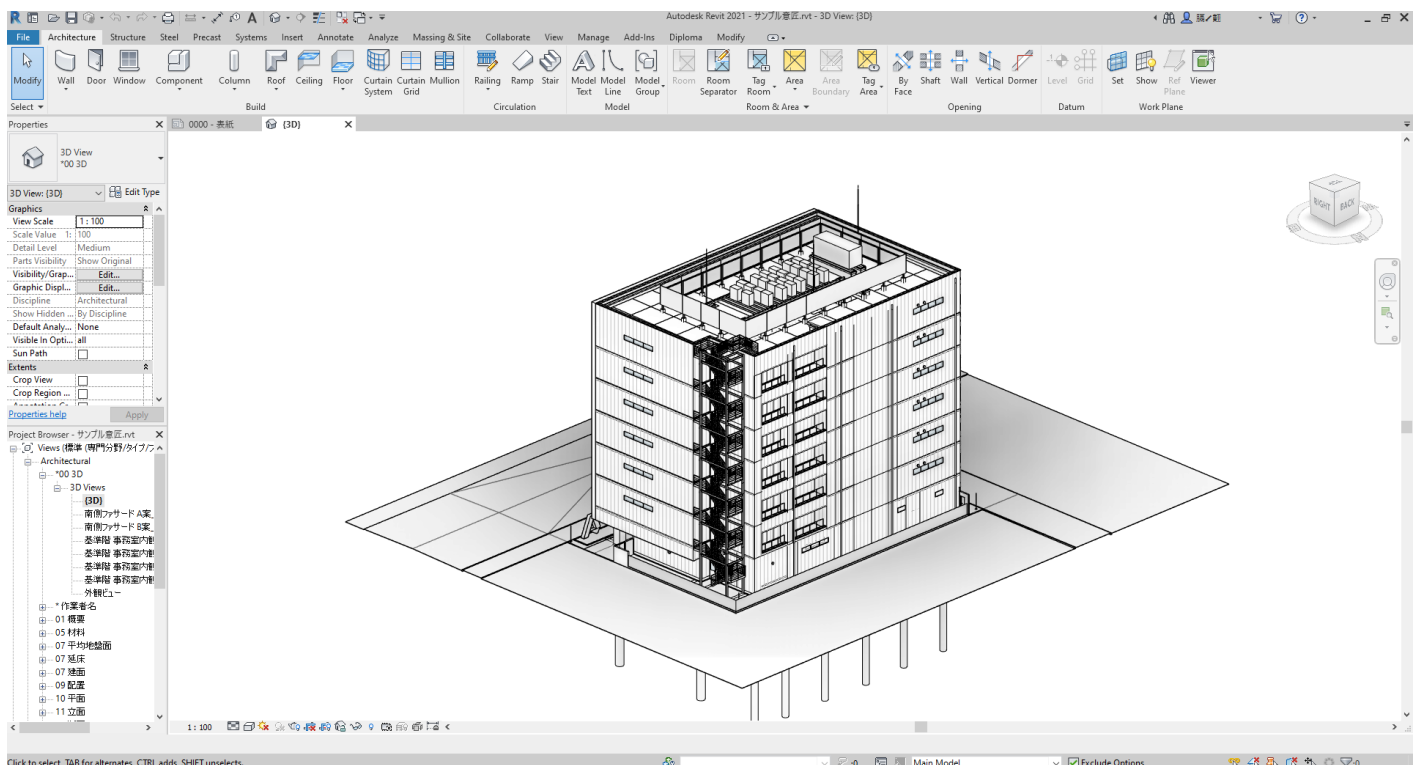


Рисунок 4.11 – 3D вигляд проекту «Sample design.rvt»

В рамках існуючого проекту активуємо експорт даних для декількох технічних листів, а саме: «1001», «1401», «2901» та «2302». На рисунку 4.12 зображена інформаційна статистика щодо вибраних специфікацій:

- загальна кількість листів для трансферу – 4;
- сумарна кількість дверей, що зображені на листі – 70;
- сумарна кількість кімнат, що підлягають обробці – 95;

Результати виконання функціоналу «Export» представлені на рисунках 4.13-4.16.

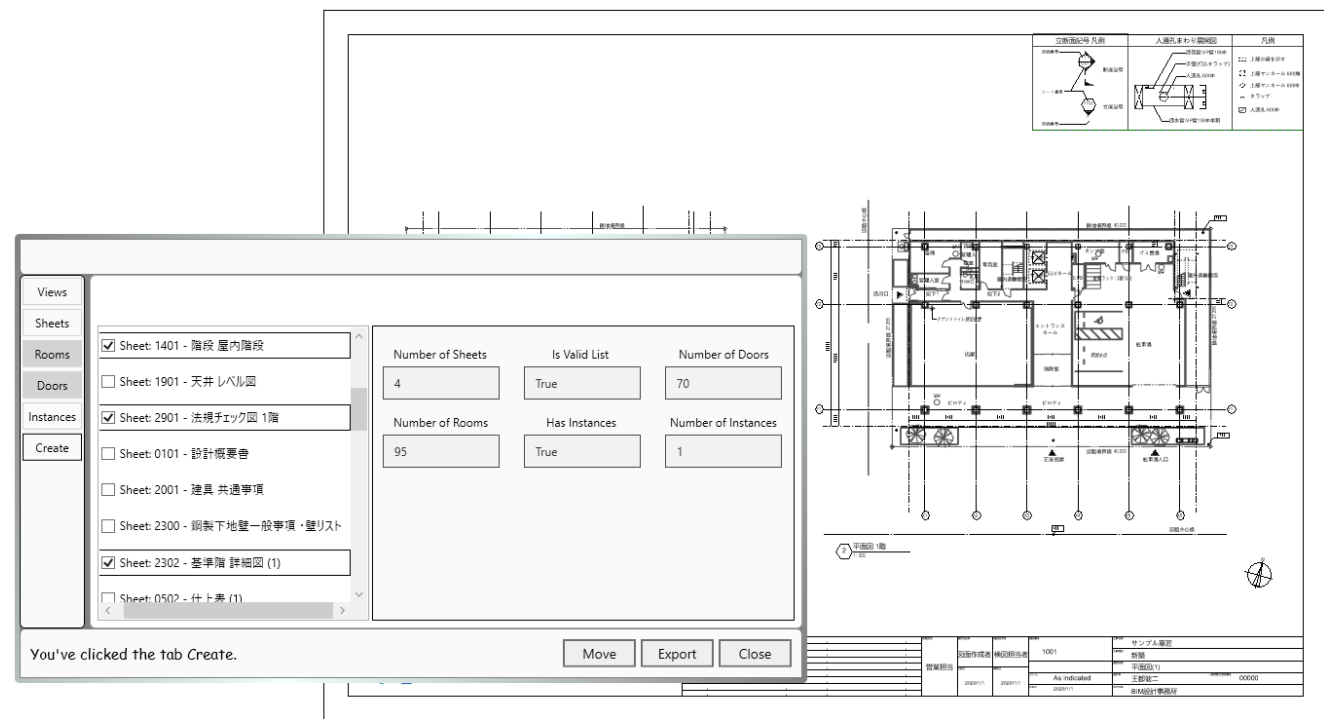


Рисунок 4.12 –інформація щодо обраних листів для проекту «Sample design.rvt»

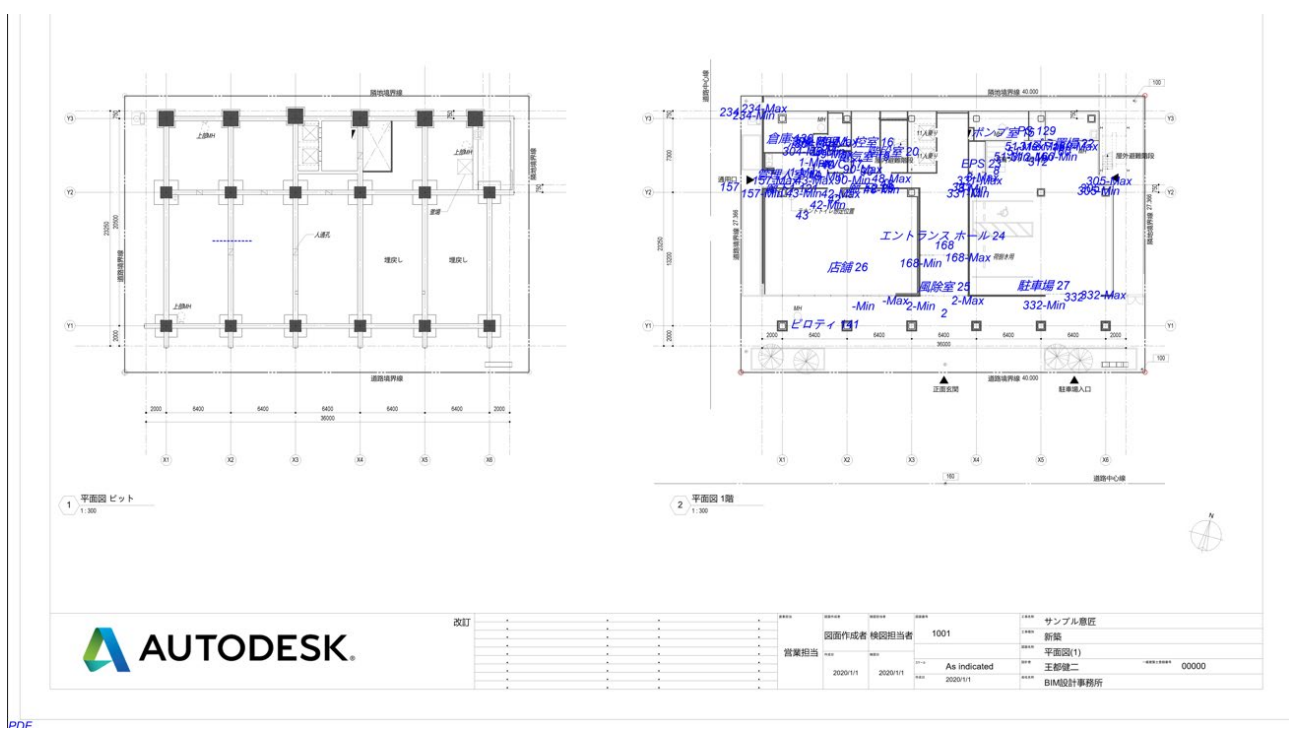


Рисунок 4.13 – розташування міток дверей та кімнат на листі «1001»

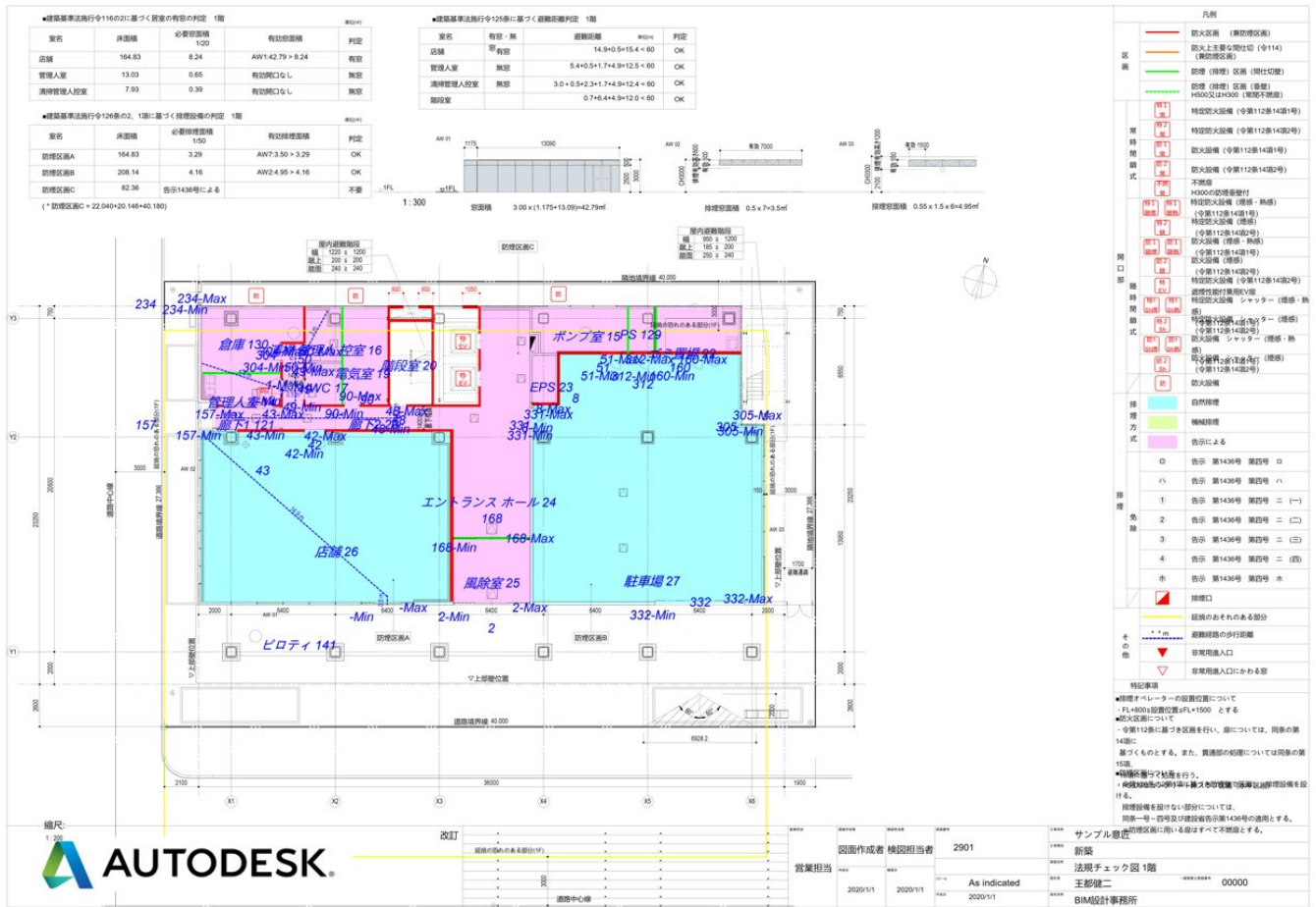


Рисунок 4.16 – розташування міток дверей та кімнат на листі «2901»

На відміну від архітектурного рішення з об’єктом приватної власності, що відноситься до будівель житлового класу, сутність багатоповерхового комплексу складається з більшої кількості екземплярів типу Family Instance та Room.

Розглянувши виконання функцій експорту при опрацюванні декількох технічних листів типу View Sheet в складі 3D моделі житлового комплексу «Sample design.rvt», можна дійти до конструктивного висновку, що розрахування паперових координат було здійснено майже без жодної похибки. Алгоритми сортування екземплярів Family Instance та Room спрацювали без помилок та непередбачуваних результатів.

4.4 Апробація методу на прикладі моделі освітньої установи

На рисунку 4.17 представлено модель 3-х поверхового шкільного закладу, що відноситься до категорії громадських будівель та споруд. За аналогією з проектом приватного будинку об'єкт школи поставляється за замовчуванням при установці системи архітектурного проектування і конструювання.

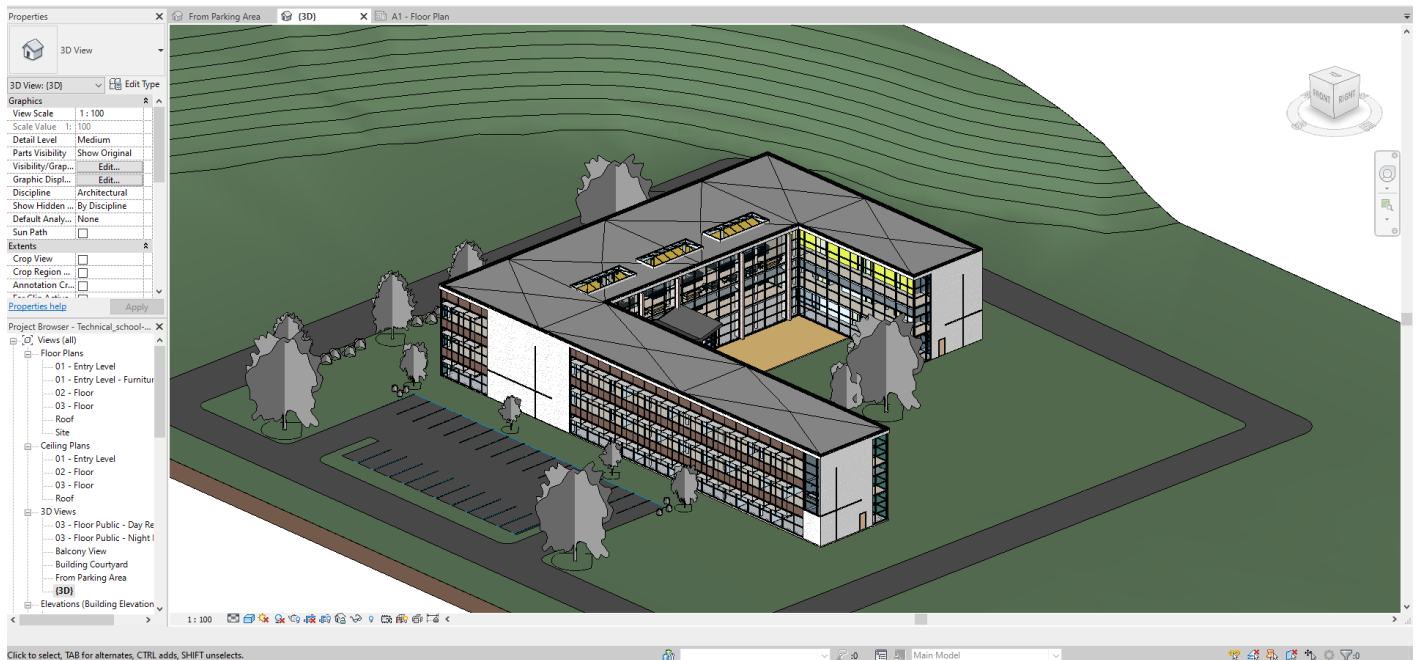


Рисунок 4.17 – 3D вигляд проекту «Technical_school-current_m.rvt»

Для існуючого архітектурно-будівельного рішення реалізуємо експорт геометричної специфікації для декількох технічного листа «A1 – Floor Plan». На рисунку 4.18 представлена попередня інформація щодо екземпляра типу View Sheet, що підлягає трансферу:

- загальна кількість листів для трансферу – 1;
- кількість дверей, що зображені на листі – 38;
- кількість кімнат, що підлягають обробці – 31;

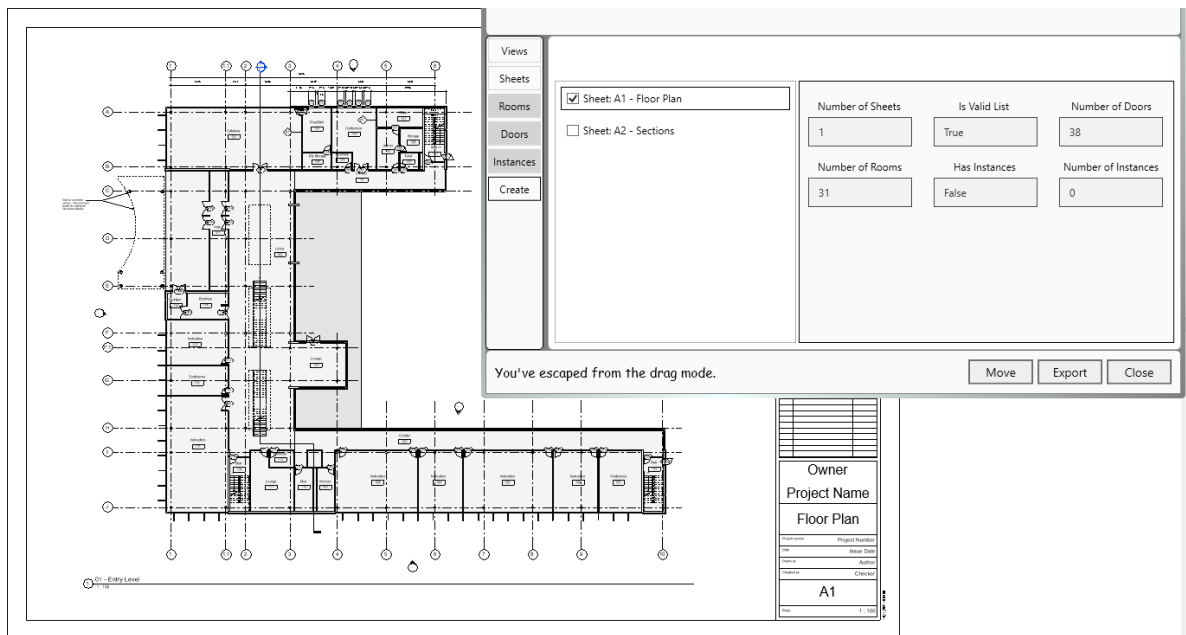


Рисунок 4.18 – інформація щодо листа для проекту «Technical_school-current_m.rvt»

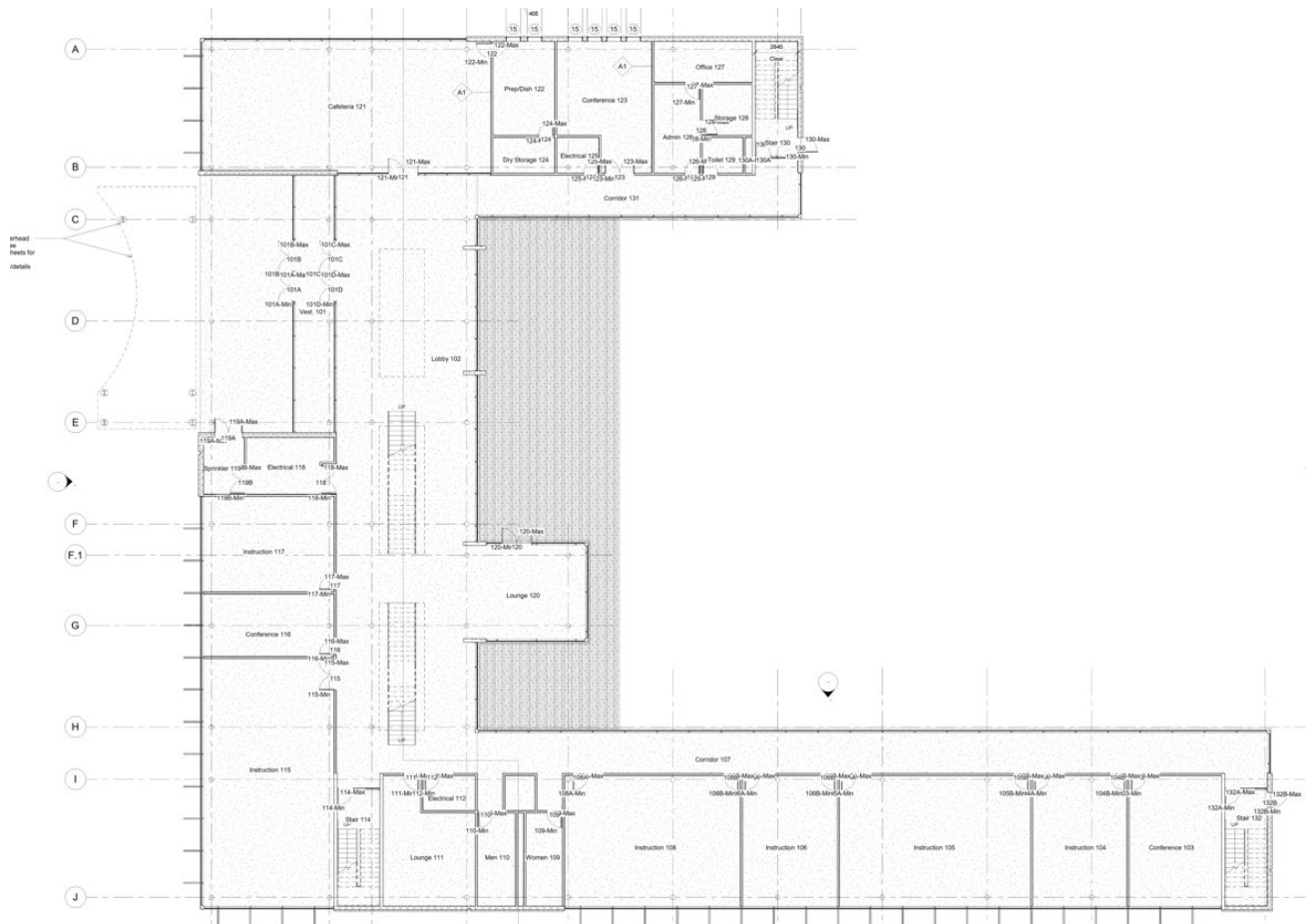


Рисунок 4.19 – розташування міток дверей та кімнат на листі «A1 – Floor Plan»

Результати виконання кнопки «Export» зображені на рисунку 4.19. Встановлено, що конструкторська надбудова розрахувала паперові координати для геометричних секцій та дверей на рівні першого поверху будівлі, як і було передбачено структурою обраної специфікації. Для перевірки сформованих координат були використані текстові мітки, які були успішно створені засобами програмного інтерфейсу Revit API.

Розглянемо зміст метаданих щодо обраного технічного листа типу View Sheet. Результати аналітичної структури представлено в таблицях 4.5-4.8.

Таблиця 4.5 – об’єкт з колекції «sheets» та його параметри

| Назва | Значення | Опис |
|-------------|------------|----------------------------------|
| sheetNumber | A1 | Інформація про номер листа |
| sheetName | Floor Plan | Інформація про назву листа |
| scale | 1 | Інформація про масштабування |
| unit | Feet | Інформація про одиниці координат |

Таблиця 4.6 – об’єкт з колекції «doors» та його параметри

| Назва | Значення | Опис |
|----------|---|---|
| guid | 1832d549-62c0-44df-812c-b19e8965ded5-00023a4f | Ідентифікатор дверей |
| name | 0915 x 2134mm | Інформація про назву дверей |
| mark | 122 | Інформація про марку дверей |
| isLinked | false | Перевірка стосовно підключення |
| level | 01 - Entry Level | Інформація про поверх дверей |
| max | 1.31557034885437, 2.31854830755232 | Максимальні координати дверей в системі координат паперового простору |
| min | 1.25385775042918, 2.28354174587253 | Мінімальні координати дверей в системі координат паперового простору |

| | | |
|----------|---|-----------------------------------|
| tag | 1.28614941709585, 2.30104502671243 | Координати тега або центра дверей |
| toRoom | 50383581-0df4-4a83-bafb- ea6cc3265f8b-0002b499 | Ідентифікатор вхідної кімнати |
| fromRoom | 50383581-0df4-4a83-bafb- ea6cc3265f8b-0002b49a | Ідентифікатор вихідної кімнати |

Таблиця 4.7 – об'єкт «toRoom» з колекції «rooms» та його параметри

| Назва | Значення | Опис |
|----------|---|--|
| guid | 50383581-0df4-4a83-bafb- ea6cc3265f8b-0002b499 | Ідентифікатор кімнати |
| name | Cafeteria 121 | Інформація про назву кімнати |
| number | 121 | Інформація про номер кімнати |
| isLinked | false | Перевірка стосовно підключення |
| tag | 0.990939896118436, 2.19304029445955 | Координати тега або локальної позиції геометричної секції |

Таблиця 4.8 – об'єкт «fromRoom» з колекції «rooms» та його параметри

| Назва | Значення | Опис |
|----------|---|--|
| guid | 50383581-0df4-4a83-bafb- ea6cc3265f8b-0002b49a | Ідентифікатор кімнати |
| name | Prep/Dish 122 | Інформація про назву кімнати |
| number | 122 | Інформація про номер кімнати |
| isLinked | false | Перевірка стосовно підключення |
| tag | 1.35240678718072, 2.22915534520189 | Координати тега або локальної позиції геометричної секції |

Отже, проаналізувавши результати генерації метаданих для технічного листа «A1 – Floor Plan», який є частиною 3D моделі шкільного закладу «Technical_school-current_m.rvt.rvt», визначено, що програмна система в інфраструктурі системи

автоматизованого проектування опрацьовує дані різної архітектурної класифікації без явних похибок та критичних помилок. Для тестування було обрано об'єкти двох типів: житлові будинки та громадський заклад.

4.5 Висновки до розділу 4

Виконано експериментальну апробацію результатів функціонування програмної інформаційної системи. Було перевірено роботу основних бізнес-процесів архітектурної надбудови та проведено тестування методу збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом пониження розмірностей векторного простору на прикладі трьох проектів Autodesk Revit, а саме:

- модель приватного будинку;
- модель багатоповерхового житлового комплексу;
- модель шкільного закладу.

Висновки

В результаті виконання кваліфікаційної роботи магістра було розроблено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору для системи автоматизованого проектування Autodesk Revit. Програмний продукт дозволить користувачам платформи архітектурної візуалізації використовувати функції перегляду, опрацювання, математичного перетворення та серіалізації над елементами будівельних 3D моделей. Для реалізації інструментальної надбудови було обрано технологію .Net Framework 4.8, середовище розробки Microsoft Visual Studio Enterprise 2019, Revit API та Autodesk Revit 2021.

Для досягнення поставленої мети були виконані наступні задачі-дослідження:

- сформульовано методику перетворення систем координат;
- реалізовано фільтр для визначення об'єктів, що підлягають трансферу;
- визначено необхідний перелік функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- побудовано інформаційну модель методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- реалізовано інструментарій для хмарного зберігання даних;
- реалізовано алгоритм перетворення даних до універсального формату зберігання та подальшого трансферу;
- виконана алгоритмічна та програмна реалізація методу пониження розмірностей векторного простору.

В рамках дослідження були визначені та охарактеризовані функціональні потоки інформаційної системи, описана взаємодія між програмними модулями.

Проведена експериментальна апробація основних функцій архітектурно-будівельної надбудови. Для тестування алгоритмів генерації метаданих були

проаналізовані три типових проекта Autodesk Revit, а саме моделі приватного будинку, багатоповерхового житлового комплексу та 3-х поверхового шкільного закладу. Результати перетворення координат для вищезгаданих будівель завершені без хибної ідентифікації та явних помилок.

Перелік посилань

1. Інформація про систему координат [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Sistema_koordinat
2. Прямокутна система координат [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Pryamougol'naya_sistema_koordinat
3. Полярна система координат [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Polyarnaya_sistema_koordinat
4. Сферична система координат [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Sfericheskaya_sistema_koordinat
5. What is Autodesk, Introduction to Autodesk [Електронний ресурс]. – режим доступу: <https://www.educba.com/what-is-autodesk/>
6. Yori R. Mastering Autodesk Revit 2020 1st Edition / Yori R., Kim M., Kirby L. // The best-selling Revit guide, now more complete than ever with all-new coverage on the 2020 release. – 2019. – P. 50-56.
7. Autodesk© Revit© Architecture 2013–2014 / James V., Eddie K., Phil R. // This book, written by renowned American experts, covers the basics of working with Autodesk Revit. – 2017. – P. 34-37.
8. Autodesk Civil 3D 2020: Fundamentals (Imperial Units): Autodesk Authorized Publisher. – 2019. – P. 130-131.
9. Призначення AutoCAD Civil 3D [Електронний ресурс]. – режим доступу: http://icad.spb.ru/software/item/civil_3d/
10. Переваги Revit для підготовки BIM-моделі [Електронний ресурс]. – режим доступу: <https://infars.ru/blog/preimuschestva-revit-dlya-podgotovki-bim-modeli/>
11. Що таке хмарне сховище? [Електронний ресурс]. – режим доступу: <https://aws.amazon.com/ru/what-is-cloud-storage/>

12. Список найкращих хмарних сервісів зберігання інформації [Електронний ресурс]. – режим доступу: <https://www.mojo.ua/news/10-luchshikh-oblachnykh-servisov-khraneniya-informatsii.html>
13. Microsoft Azure [Електронний ресурс]. – режим доступу: https://en.wikipedia.org/wiki/Microsoft_Azure
14. Amazon Web Services [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Amazon_Web_Services
15. Системи координат і перетворення координат [Електронний ресурс]. – режим доступу: https://ru.wikipedia.org/wiki/Preobrazovanie_koordinat
16. About Coordinate Systems in Autodesk Revit platform [Електронний ресурс]. – режим доступу: <https://knowledge.autodesk.com/support/revit/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Revit-Model/files/GUID-E67ED082-2556-475B-84A7-4605329F612F-htm.html>
17. Revit API 2021 [Електронний ресурс]. – режим доступу: <https://www.revitapidocs.com/2021.1/>
18. Autodesk Revit Architecture 2014: No Experience Required Autodesk Official Press 1st Edition / Eric W. // This Autodesk Official Press book helps you become proficient with Autodesk's popular building information modeling software using an innovative continuous tutorial. – 2015. – P. 132-133.
19. Autodesk® Revit Basics Training Manual First Edition / Brian W. Clayton // Revit, a type of Building Information Modeling or “BIM” software, is used widely in the U.S. to provide “real-life” information to clients. – 2008. – P. 45-46.
20. Маніфест і керівництво надбудови [Електронний ресурс]. – режим доступу: <https://thebuildingcoder.typepad.com/blog/2010/04/addin-manifest-and-guidize.html>

Додатки

Додаток А

Наукова стаття

Технічні науки

ISSN 2307-5732

DOI 10.31891/2307-5732-2021-297-3-35-42

УДК 004.5.02

ROMAN RUSLANOVYCH HRYMAK

Khmelnitskyi National University
ORCID ID: 0000-0001-6021-4334
e-mail: roman.gr.1998@gmail.com

OLEKSANDR ANATOLIYOVYCH PASICHNYK

Khmelnitskyi National University
ORCID ID: 0000-0002-8760-4688
e-mail: o.a.pasichnyk@gmail.com

TETIANA KAZYMYRIVNA SKRYPNYK

Khmelnitskyi National University
ORCID ID: 0000-0002-8531-5348
e-mail: marine_1996@ukr.net

EDUARD ANDRIIOVYCH MANZIUK

Khmelnitskyi National University
ORCID ID: 0000-0002-7310-2126
e-mail: eduard.em.km@gmail.com

INFORMATION TECHNOLOGY OF MAKING CONTROLLED CRITICALLY SAFE DECISIONS ABOUT MODEL PARAMETERS CONVERSION AT TRANSFER BETWEEN VISUALIZATION SYSTEMS

In modern production, computer-aided design systems have become widespread, which provide the opportunity to create technological processes with less time and engineering. Automated design system is a system capable of automated level to implement information technology to perform design functions, is an organizational and technical set of software tools designed to automate the design process, consisting of staff and a group of technical, software and other means of automating its activities.

Computer-aided design systems are an important link in industrial design, widely used in many industries, including the automotive, shipbuilding and aerospace industries, industrial and architectural design, prosthetics and many others. CAD is also widely used in computer animation for special effects in movies, commercials, and technical materials, often referred to as digital content. Due to its economic importance, the computer-aided design system has become the main driving force of research in the field of computational geometry, computer graphics (both hardware and software) and discrete differential geometry.

In today's automated manufacturing market, most constructors use additional engineering software. As a rule, such add-ins are used in the functional infrastructure of a specialized set of solutions that implement the principle of building information modeling (BIM). The most common system of this type is Autodesk Revit, a platform that provides three-dimensional modeling of building elements and flat drawing of design elements, designed for architects, designers and design engineers.

This research presents the results of information technology for Autodesk Revit computer-aided design system based on 2019-2021 packages, which will allow users of the architectural visualization platform to use the functions of viewing, processing, mathematical transformation and serialization on elements of 3D building models.

Keywords: coordinate systems, visualization systems, Autodesk, Autodesk Revit, Building Information Modeling.

РОМАН РУСЛАНОВИЧ ГРИМАК

Хмельницький національний університет

ОЛЕКСАНДР АНАТОЛІЙОВИЧ ПАСІЧНИК

Хмельницький національний університет

ТЕТЯНА КАЗИМИРІВНА СКРИПНИК

Хмельницький національний університет

ЕДУАРД АНДРІЙОВИЧ МАНЗЮК

Хмельницький національний університет

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРИЙНЯТТЯ КОНТРОЛЬОВАНИХ КРИТИЧНО-БЕЗПЕКОВИХ РІШЕНЬ ПЕРЕТВОРЕННЯ ПАРАМЕТРІВ МОДЕЛЕЙ ПРИ ТРАНСФЕРІ МІЖ СИСТЕМАМИ ВІЗУАЛІЗАЦІЇ

На сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення. Як правило, такі надбудови використовуються у функціональній інфраструктурі спеціалізованого комплексу рішень, що реалізують принципи інформаційного моделювання будівель. Найпоширенішою системою такого типу є Autodesk Revit, тобто платформа, яка надає можливості тривимірного моделювання елементів будівлі і плоского креслення елементів оформлення, призначеного для архітекторів, конструкторів і інженерів-проектувальників.

Ключові слова: системи координат, системи візуалізації, Autodesk, Autodesk Revit, Building Information Modeling.

FORMULATION OF THE PROBLEM

Computer-aided design system has a set of functional tasks that allow engineer to basically proceed the information modeling of architectural and construction 3D models. However, like any system of geometric visualization, these platforms are limited in the implementation of specialized processes, for example, in the calculations of local coordinates related to the view elements on the engineering sheets. Therefore, by the infrastructure of Autodesk platform and its application programming interface developers obtain the ability to

organize an unlimited number of engineering add-ins which will be used to solve a specified collection of tasks.

ANALYSIS OF RECENT RESEARCH

Autodesk Inc. is the world's largest provider of software for industrial and civil construction, mechanical engineering, media and entertainment markets. The company has developed a wide range of replicated software products for architects, engineers and designers. There are currently more than 9 million Autodesk users worldwide [1].

In total, Autodesk currently produces about a hundred software products. Four systematized divisions are engaged in development:

1. Solutions for industrial production and mechanical engineering is a comprehensive set for design and engineering, which is used in various industries, including mechanical engineering, electromechanical, automotive production of industrial equipment and consumer goods. Many products are based on digital prototype technology. The solutions of this segment include: Autodesk Inventor, AutoCAD Mechanical and AutoCAD Electrical, Autodesk Simulation.

2. Solutions for the architectural and construction industry [2]. The programs of this group are used mainly by various architectural and design workshops, as well as other construction companies for the design of various buildings and structures, modeling and analysis of their structures and subsystems. These solutions include parametric design systems based on building information modeling technology Autodesk Revit Architecture and Autodesk Revit Structure, applications for the design of building subsystems AutoCAD Architecture, AutoCAD Civil 3D and AutoCAD MEP, as well as analytical systems for solving problems of environmentally friendly design. The solutions of this segment include: Autodesk Revit, AutoCAD Civil 3D, AutoCAD Map 3D.

3. Solutions for working with animation, graphics and creating virtual reality. Tools for creating multimedia materials in all areas of the entertainment industry, from visual effects in film and television, and color correction to animation, rendering, and computer game development. Major developments include 3D graphics editors Autodesk Maya, Autodesk 3ds Max and Autodesk Softimage, Motionbuilder animation solutions, Mudbox digital image sculpting, Burn, Inferno, Flame, Flint, Luster and Smoke special video effects processors.

Information technology about parametric transformation based on the system of building information modeling belongs to the category of solutions for the architectural and construction industry, so at the stage of analysis of existing software in the subject area were selected the following computer-aided design systems: Autodesk Revit and AutoCAD Civil 3D.

Autodesk Revit or simply Revit is a software package for computer-aided design, which implements the principle of information modeling of buildings (figure 1).

The computer-aided design system is designed for architects, designers and design engineers [3]. The platform provides the opportunities for three-dimensional modeling of building elements and flat drawings of design elements, creation of custom objects, organization of joint work on the project, starting from the concept and ending with the release of working drawings and specifications.



Figure 1. Autodesk Revit

AutoCAD Civil 3D is a software product for professionals in the field of land management, surveying, master plan design and infrastructure [4]. It is based on the use of BIM-technologies and three-dimensional mathematical model of objects. AutoCAD Civil 3D allows you to fully automate the design of infrastructure objects, the creation and release of working documentation, ranging from the collection and processing of field data, geodetic surveys and ending with 3D-visualization of the design concept and the construction of the objects themselves.

In the market of automated products for solutions of the architectural and construction industry, there are a large number of software products for the development of BIM-model and subsequent design of working documentation. If you consider modeling tools, the Revit package has everything you need to quickly and accurately build a 3D model.

With proper preparation of the template, all the information part will already be embedded in the BIM-model. As part of the template preparation, BIM coordinators configure all the necessary specifications. As the BIM model is built, these specifications are automatically populated and, in the event of any changes, are automatically updated. These settings can also be made with dynamic blocks in Autocad, but in some technical scenarios they may not look very stable.

In the Revit computer-aided design system, the model and the specifications are interrelated [5]. That is, you can select any element in the specification and show it in any form where this element/elements are present. This allows you to: eliminate unnecessary elements, quickly fill in the specifications and coordinate the project.

When Revit lacks standard tools, there is a large database of plug-ins that allow you to perform various actions. From the simplest, such as alignment of elements according to the given law or the expanded choice of elements of model on parameters, to more difficult - for example, automatic construction of finishing of rooms.

Therefore, after analyzing the existing software of the subject area and determining all the necessary conditions for creating information technology for model parameters conversion, we can conclude that the software add-in will be implemented for the automated design system Autodesk Revit.

The purpose of the research: thus, having analyzed the characteristics of the subject area, a set of definitions for working with the position of mathematical objects and architectural and construction visualization system, for information technology about conversion of BIM 3D model parameters there is an implementation of the following functional stages:

1. Development and organization of technical means for creating a tool add-in in the computer-aided design system Autodesk Revit.
2. Definition of the basic parameters for work with components of building 3D models. Systematization of data on the elements of the room, obtaining standard information through the Revit API.
3. Formation of a mathematical model in order to convert the coordinates between the elements of the architectural and construction model and engineering sheets.
4. Organization of the results related to mathematical processing in the text format of data exchange. Generation of built-in elements of architectural and construction visualization.
5. Development of an auxiliary module for working with software resources of the Azure cloud service. Create and configure a storage account with a 5GB quota. Sending the resulting data with subsequent synchronization within the cloud service. Transfer of engineering specification between architectural visualization systems.

PRESENTATION OF THE MAIN MATERIAL

Coordinate transformation is the replacement of a coordinate system on a plane, in space, or, in the most general case, on a given n -dimensional variety [6].

All rectangular coordinate systems in the study space are equal, ie the choice of one of them is not worse or better than the choice of another. These or those preferences are given based on the specifics of a particular task. The use of different coordinate systems poses the problem of converting the coordinates of a point, ie the problem of calculating its coordinates in one coordinate system by its coordinates in another system.

The Revit computer-aided design system, designed to work with architectural solutions, uses two coordinate systems: the survey coordinate system and the project coordinate system [7].

The survey coordinate system is the real context for the building model. It is designed to describe the location on the earth's surface. The survey coordinate system is used to identify a specific location on the ground where the Revit model is located. This coordinate system is created outside the context of the project.

A large number of shooting coordinate systems are standardized. Some systems use latitude and longitude, while others use the coordinates of points on the X, Y, and Z axes. Surveying coordinate systems use much larger scales than project coordinate systems, and take into account details such as land and terrain curvature that are insignificant for coordinate systems of project systems.

The coordinate system of the project describes the location relative to the building model. It uses the selected point within the plot as a reference point to measure the distances and locations of objects relative to the model.

The project coordinate system is used to determine the location of objects relative to a given point next to the model. This coordinate system is unique to the current project. The beginning of the internal coordinate system is the basis for the survey coordinate system and the project coordinate system.

Information technology for converting the parameters of geometric objects involves the generation of a single digital model in the context of architectural components. To implement this idea there is a need in systematization of mathematical algorithms to convert the coordinates between the elements of the BIM model and engineering sheets.

The essence of the View is the basic view for all types of View in Autodesk Revit. The view can display an image created with the Revit model. Displays can be graphical, such as plans, heights, or 3D views, or textual, such as graphics (figure 2).

Autodesk Revit's suite of application programming interface allows you to operate architectural objects at

the Active View level. As a rule, the array of geometric primitives is organized under the coordinate system of the project, that is, the center of mathematical calculations is a point set by the author of the document.



Figure 2. Floor Plan in Autodesk Revit

Engineering sheet or Sheet/ViewSheet entity is a type of architectural mapping consisting of a model view and a header. Each sheet has a unique sheet number in a complete set of drawings. The number is displayed before the sheet name in the project browser (figure 3).

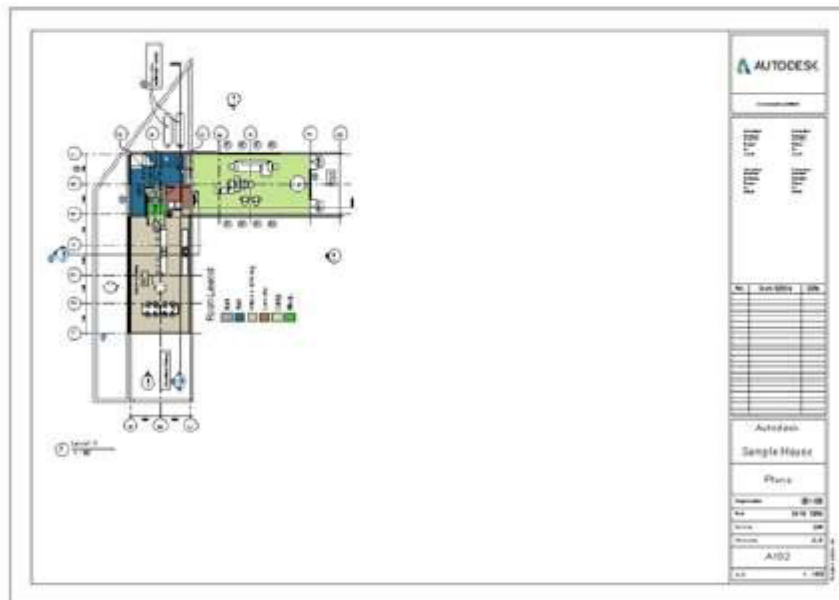


Figure 3. ViewSheet in Autodesk Revit

The main structural element for the ViewSheet entity is an object called Viewport, which acts as an information intermediary between the specific mapping of the floors and the engineering sheets on which these mappings are placed. The ViewSheet coordinate system is a two-dimensional paper space, where the center of coordinates by default is the lower left corner of the active sheet. In case the user of the platform for architectural solutions tries to remove the header from the ViewSheet view, Autodesk Revit will automatically set the center of

the paper coordinate system depending on the location of the Viewport components.

For design of structure and systematization of information technology regarding the parameters transformation of architectural and construction 3D models there is a need in the analysis and automation of information flows processing. The functionality of the engineering add-in must match all the necessary conditions for data processing from the Revit API. The business processes in Autodesk Revit can be divided into the following groups:

1. Business process for working with views of the active document (figure 4):

- display a list of document views;
- view data of the active view;
- interactions through Revit: connection and scaling functions for the relatively active view;
- search for objects from the list of available views.

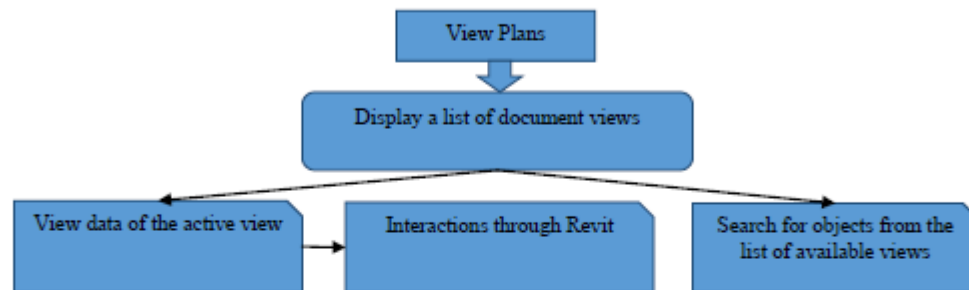


Figure 4. Business process for working with views of the active document

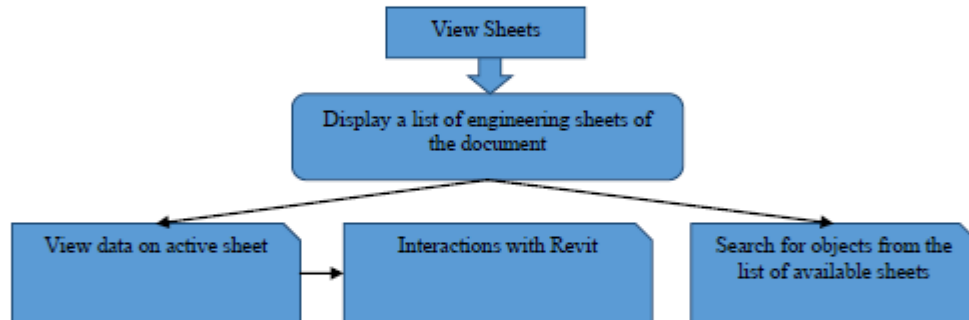


Figure 5. Business process for working with a sheet of an active document

2. Business process for working with a sheet of an active document (figure 5):

- display a list of engineering sheets of the document;
- view data on active sheet;
- interactions with Revit: function of connection of the active engineering sheet;
- search for objects from the list of available engineering sheets.

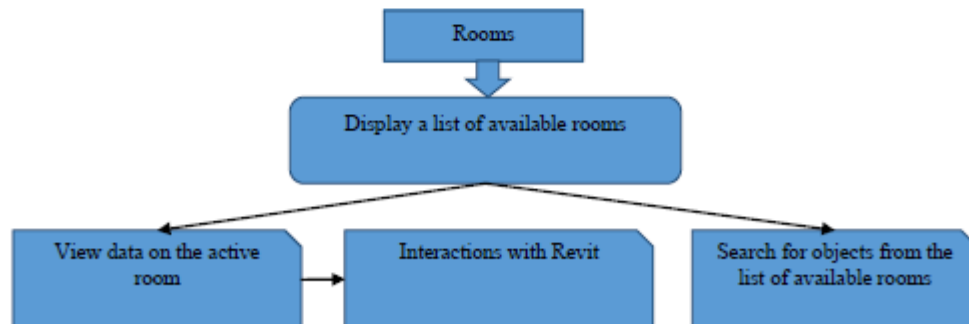


Figure 6. Business process for working with rooms at the level of active view

3. Business process for working with rooms at the level of active view (figure 6):

- display a list of available rooms;

- view data on the active room;
- interactions with Revit: zoom function to the active room;
- search for objects from the list of available rooms.

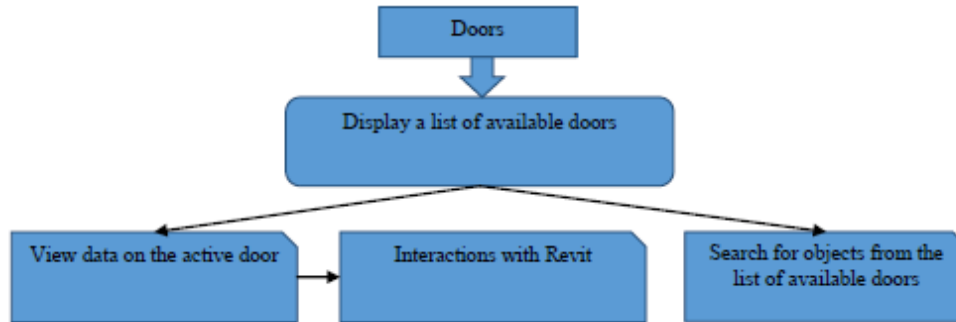


Figure 7. Business process for working with doors at the level of active view

4. Business process for working with doors at the level of active view (figure 7):
- display a list of available doors;
 - view data on the active door;
 - interactions with Revit: zoom function to the active door;
 - search for objects from the list of available doors.
5. Business process for working with linked documents (figure 8):
- display a decision tree for linked documents;
 - display rooms and doors for each linked document;

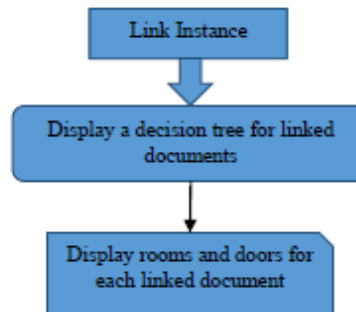


Figure 8. Business process for working with linked documents

6. Business process for working with metadata generation (figure 9):
- display a list of engineering sheets at the level of the active document;
 - give the user the opportunity to select sheets before sending to the resources of the Azure cloud service;
 - review of current information on the created data;
 - generate metadata and their images based on selected engineering sheets.

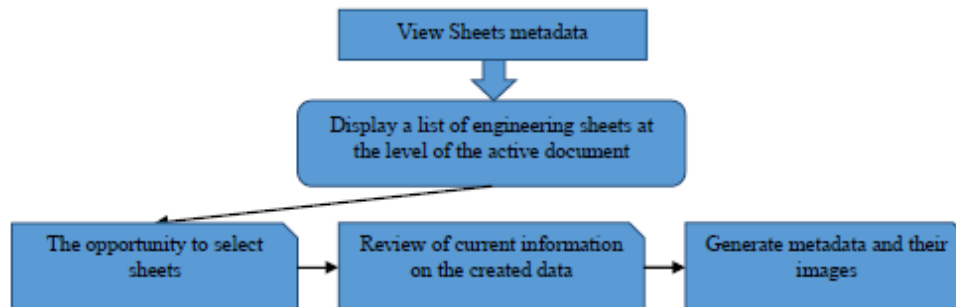


Figure 9. Business process for working with metadata generation

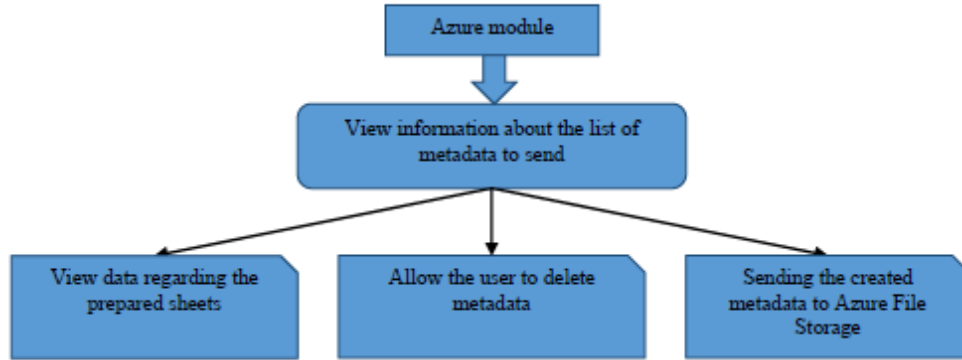


Figure 10. Business process for working with sending data to Azure File Storage

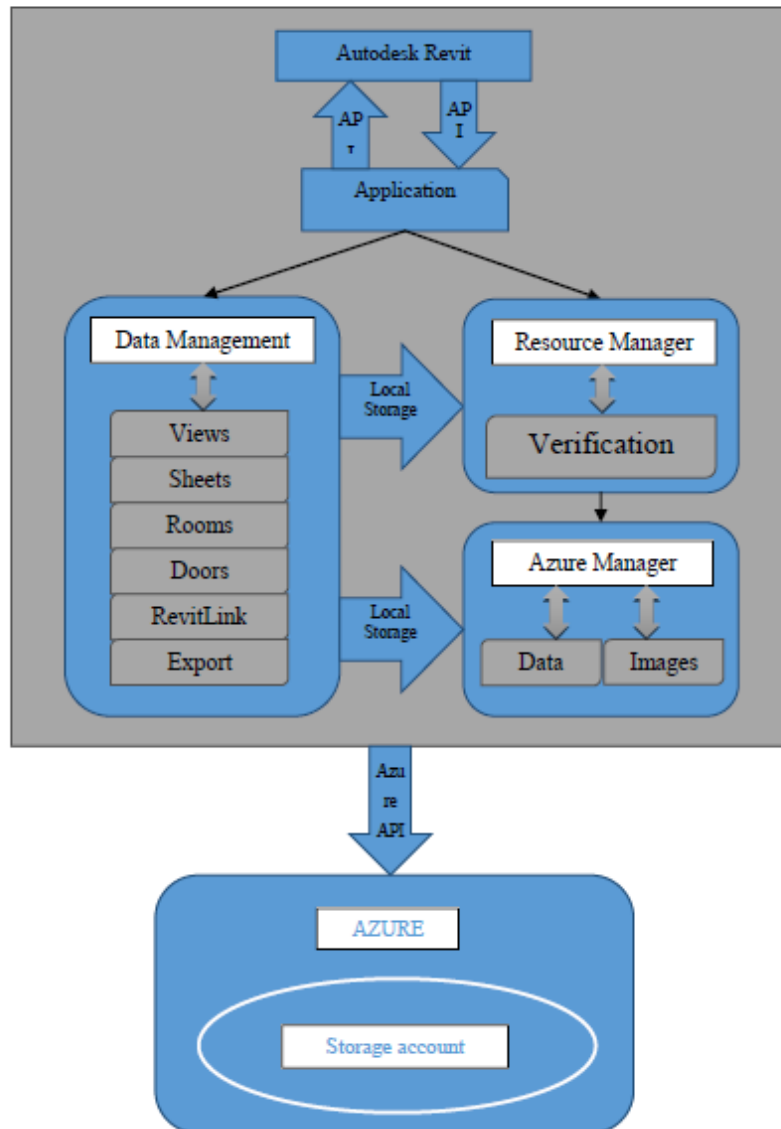


Figure 11. Functional user diagram of the engineering add-in

7. Business process for working with sending data to Azure File Storage (figure 10):

- view information about the list of metadata to send;
- allow the user to delete metadata;
- sending the created metadata to Azure File Storage.

The user of the computer-aided design system must perform a certain sequence of actions to activate the engineering superstructure and further work with the data on the connected architectural and construction 3D models.

Queries with subsequent information modeling of architectural and building elements of 3D models engineering add-in in the structure of Autodesk Revit should be carried out due to the integrated software interface Revit API. To work with the business processes of the active document, the end user must first approve the installation of the main library of the add-in in the functional infrastructure of the computer-aided design system.

It is expected that the implementation of the information module, resource manager and cloud service Azure will be implemented through the components of the user interface from the Windows operating system. The above set of elements will prepare the data of the geometric specification when transferring between visualization systems.

CONCLUSION

Thus, after analyzing the processing of information flows, the engineering technology of model parameters conversion was determined by a complete list of business processes that are subject to automation algorithms. Given the object-oriented feature groups and the organized functional diagram of the computer-aided design systems user, it is possible to configure the structure of the information system, which will be designed to work with the elements of the level with the subsequent generation of their geometric data at the local paper-space coordinate system.

References

1. Autodesk learning and training. URL: <https://www.autodesk.com/training>
2. What is Autodesk, Introduction to Autodesk. URL: <https://www.educba.com/what-is-autodesk/>
3. Yori R., Kim M., Kirby L. Mastering Autodesk Revit 2020 1st Edition. The best-selling Revit guide, now more complete than ever with all-new coverage on the 2020 release. 2019. P. 50–56.
4. Autodesk Civil 3D 2020: Fundamentals (Imperial Units): Autodesk Authorized Publisher. 2019. P. 130–131.
5. Preimushestva Revit dlya podgotovki BIM-modeli. URL: <https://infars.ru/blog/preimushchestva-revit-dlya-podgotovki-bim-modeli/>
6. Sistemi koordinat i preobrazovanie_koordinat URL: https://ru.wikipedia.org/wiki/Preobrazovanie_koordinat
7. About Coordinate Systems in Autodesk Revit platform. URL: <https://knowledge.autodesk.com/support/revit/learn-explore/caas/CloudHelp/cloudhelp/2020/ENU/Revit-Model/files/GUID-E67ED082-2556-475B-84A7-4605329F612F-htm.html>
8. Revit API 2021. URL: <https://www.revitapidocs.com/2021.1/>

Додаток Б

Тези

SCI-CONF.COM.UA

**PRIORITY DIRECTIONS
OF SCIENCE AND TECHNOLOGY
DEVELOPMENT**



**PROCEEDINGS OF IX INTERNATIONAL
SCIENTIFIC AND PRACTICAL CONFERENCE
MAY 16-18, 2021**

**KYIV
2021**

PRIORITY DIRECTIONS OF SCIENCE AND TECHNOLOGY DEVELOPMENT

Proceedings of IX International Scientific and Practical Conference

Kyiv, Ukraine

16-18 May 2021

Kyiv, Ukraine

2021

2

UDC 001.1

The 9th International scientific and practical conference “Priority directions of science and technology development” (May 16-18, 2021) SPC “Sci-conf.com.ua”, Kyiv, Ukraine. 2021. 1207 p.

ISBN 978-966-8219-84-9

The recommended citation for this publication is:

Ivanov I. Analysis of the phaumistic composition of Ukraine // Priority directions of science and technology development. Proceedings of the 9th International scientific and practical conference. SPC “Sci-conf.com.ua”. Kyiv, Ukraine. 2021. Pp. 21-27. URL: <https://sci-conf.com.ua/ix-mezhdunarodnaya-nauchno-prakticheskaya-konferentsiya-priority-directions-of-science-and-technology-development-16-18-maya-2021-goda-kiiev-ukraina-arhiv/>.

Editor**Komarytskyy M.L.***Ph.D. in Economics, Associate Professor*

Collection of scientific articles published is the scientific and practical publication, which contains scientific articles of students, graduate students, Candidates and Doctors of Sciences, research workers and practitioners from Europe, Ukraine, Russia and from neighbouring countries and beyond. The articles contain the study, reflecting the processes and changes in the structure of modern science. The collection of scientific articles is for students, postgraduate students, doctoral candidates, teachers, researchers, practitioners and people interested in the trends of modern science development.

e-mail: kyiv@sci-conf.com.ua

homepage: <https://sci-conf.com.ua>

©2021 Scientific Publishing Center “Sci-conf.com.ua” ®

©2021 Authors of the articles

| | | |
|-----|---|-----|
| | ВІД НЕБЕЗПЕЧНОГО МАГНІТНОГО ВПЛИВУ ЛІНІЙ ЕЛЕКТРОПЕРЕДАЧІ | |
| 80. | <i>Костів І. П.</i> КЛАСИФІКАЦІЯ ФІНАНСОВИХ ТРЕКЕРІВ ТА ВИКОРИСТАННЯ ТРЕКЕРІВ ВИТРАТ КОМПАНІЯМИ | 349 |
| 81. | <i>Кузнецов Ю. А.</i> МАТЕМАТИЧЕСКИЕ МОДЕЛИ ТЕМПЕРАТУРНЫХ ОШИБОК МАСШТАБНЫХ КОЭФФИЦИЕНТОВ ВОЛОКОННО-ОПТИЧЕСКИХ ГИРОСКОПОВ НАВИГАЦИОННЫХ СИСТЕМ | 354 |
| 82. | <i>Литовченко В. В.</i> ПІДКЛЮЧЕННЯ ТА ВИКОРИСТАННЯ ANDROID БІБЛІОТЕКИ RETROFIT | 361 |
| 83. | <i>Маленчак В. Б.</i> ІНТЕЛЕКТУАЛЬНА СИСТЕМА ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У НОВИНАХ | 364 |
| 84. | <i>Мосильний С. Б., Мініч М. А.</i> СИСТЕМА АНАЛІЗУ ІНТЕНСИВНОСТІ ТРАНСПОРТНИХ ПОТОКІВ НА RASPBERRY PI 4 | 370 |
| 85. | <i>Несук О. О., Потапова К. Р.</i> ЗВОРОТНИЙ АЛГОРИТМ РЕНДЕРИНГУ ОБ'ЄКТІВ ЗОБРАЖЕННЯ | 377 |
| 86. | <i>Ошовський В. Я.</i> ПІДВИЩЕННЯ ЗНОСОСТІЙКОСТІ І ЯКОСТІ ЦЕМЕНТОВАНИХ ПОВЕРХОНЬ СТАЛЬНИХ ДЕТАЛЕЙ ГЛИБОКИМ ОХОЛОДЖЕННЯМ | 383 |
| 87. | <i>Пасічник О. А., Гримак Р. Р.</i> ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕТВОРЕННЯ ПАРАМЕТРІВ МОДЕЛІ ПРИ ТРАНСФЕРІ МІЖ СИСТЕМАМИ ВІЗУАЛІЗАЦІЇ | 390 |
| 88. | <i>Радай Р. О.</i> ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА СТАТИСТИЧНОГО АНАЛІЗУ ТА ВІЗУАЛІЗАЦІЇ ФІНАНСОВИХ ПОКАЗНИКІВ КОМПАНІЙ | 394 |
| 89. | <i>Романчук Я. П.</i> ОПТИМАЛЬНИЙ ЛОКАЛЬНИЙ ПІДГРІВ ЗВАРЮВАНИХ ЕЛЕМЕНТІВ МАШИНОБУДІВНИХ КОНСТРУКЦІЙ | 399 |
| 90. | <i>Романюк О. Н., Романюк О. В., Кондрук Р. В.</i> ГАЛУЗІ ЗАСТОСУВАННЯ КРИВОЇ ГІЛЬБЕРТА | 403 |
| 91. | <i>Свідін О. Є., Білоус Н. В.</i> ПОШУК ТА ПОРІВНЯННЯ ПОЗ. НА ОСНОВІ 3D КООРДИНАТ СУГЛОБІВ ЛЮДИНИ | 409 |
| 92. | <i>Степанов Д. М., Великожон Б. М.</i> ПЕРЕВІРКА ЗАКОНУ РОЗПОДІЛУ ЧАСУ НАПРАЦЮВАННЯ НА ВІДМОВУ ВОЛОКОННО-ОПТИЧНИХ ЛІНІЙ ЗВ'ЯЗКУ КРИТЕРІЯМИ ЗГОДИ | 416 |

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПЕРЕТВОРЕННЯ ПАРАМЕТРІВ МОДЕЛІ ПРИ ТРАНСФЕРІ МІЖ СИСТЕМАМИ ВІЗУАЛІЗАЦІЇ

Пасічник Олександр Анатолійович

к. т. н., доцент,

Гримак Роман Русланович

Магістр

Хмельницький національний університет

м. Хмельницький, Україна

Вступ./Introduction. У сучасному виробництві корисне поширення одержали системи автоматизованого проектування, які надають можливість створювати технологічні процеси з меншими витратами часу та інженерних засобів. Система автоматизованого проектування це система, що здатна на автоматизованому рівні реалізувати інформаційну технологію виконання функцій проектування, це організаційно-технічний комплекс програмних інструментів, призначених для автоматизації процесу проектування, що складається з персоналу і групи технічних, програмних і інших засобів автоматизації його діяльності.

Системи автоматизованого проектування є важливою ланкою в промисловому конструюванні, широко використовуваним в багатьох галузях, в тому числі в автомобільній, суднобудівній та аерокосмічній індустрії, промисловому і архітектурному проектуванні, протезуванні і багатьох інших. CAD також широко використовується в створенні комп'ютерної анімації, рекламних і технічних матеріалах, які часто називають цифровим контентом. Через свою економічну важливість, система автоматизованого проектування стала основною рушійною силою досліджень в області обчислювальної геометрії, комп'ютерної графіки (як апаратної, так і програмної) і дискретної диференціальної геометрії.

Мета роботи./Aim. Метою роботи є розробка інформаційної технології для системи автоматизованого проектування Autodesk Revit на основі пакетів

2019-2021, яка дозволить користувачам платформи архітектурної візуалізації використовувати функції перегляду, опрацювання, математичного перетворення та серіалізації над елементами будівельних 3D моделей.

Матеріали та методи./Materials and methods. Autodesk Inc. це найбільший в світі постачальник програмного забезпечення для промислового і цивільного будівництва, індустрії машинобудування, ринку засобів інформації та розваг. Компанією розроблений широкий спектр тиражованих програмних продуктів для архітекторів, інженерів та конструкторів. На поточний момент налічується понад 9 млн користувачів Autodesk по всьому світу. Всього Autodesk в даний час випускає близько ста програмних продуктів. Розробкою займаються чотири систематизованих підрозділи.

Інформаційна технологія параметричного перетворення на основі системи інформаційного моделювання будівель відноситься до категорії рішення для архітектурно-будівельної галузі, тому на етапі аналізу існуючого програмного забезпечення предметної області були обрані наступні системи автоматизованого проектування: Autodesk Revit, AutoCAD Civil 3D і AutoCAD Map 3D. Autodesk Revit, або просто Revit це програмний комплекс для автоматизованого проектування, який реалізує принцип інформаційного моделювання будівель.

Для розробки програмного продукту, направлено на автоматизацію обробки елементів будівельних 3D моделей, перш за все потрібно обрати програмну платформу. Найбільш популярним є: .Net Framework і Java.

.NET Framework це програмна платформа, випущена компанією Microsoft в 2002 році. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), яка підходить для різних мов програмування. Функціональні можливості CLR доступні в будь-яких мовах програмування, що використовують подане середовище.

Середовища розробки, що підтримують .NET:

– Microsoft Visual Studio (C#, Visual Basic .NET, Managed C++, F#).

Комплекс продуктів компанії Microsoft, що включає інтегроване середовище

розробки програмного забезпечення і ряд інших інструментальних інструментів;

- SharpDevelop. Вільне середовище розробки для C #, Visual Basic .NET, Boo, IronPython, IronRuby, F#, C++;
- MonoDevelop. Середовище розробки є частиною проекту Mono. Вбудований в дистрибутив Unity3D як засіб написання скриптів;
- Embarcadero RAD Studio (Delphi for .NET); раніше Borland Developer Studio (Delphi for .NET, C#);
- A#. Портована версія мови програмування Ада на платформу Microsoft .NET. A# вільно поширюється кафедрою інформатики Військово-повітряної академії США як внесок в розвиток Ада-спільноти;
- Zonnon. Мова програмування загального призначення, заснований на мові Modula-2, і підтримує активні об'єкти, що з'явилися в Active Oberon;
- PascalABC.NET;
- JetBrains Rider. Кросплатформенне інтегроване середовище розробки програмного забезпечення для платформи .NET, що розробляється компанією JetBrains. Підтримуються мови програмування C#, VB.NET і F#.

Для реалізації програмного продукту було обрано технологію .Net Framework 4.8, середовище розробки Microsoft Visual Studio Enterprise 2019, Revit API та Autodesk Revit 2021.

Результати та обговорення./Results and discussion. За результатами аналізу предметної області, наявного програмного забезпечення предметної області та сучасних засобів реалізовано інженерну надбудову, що реалізує інформаційну технологію перетворення параметрів будівельних 3D моделей із підтримкою таких функцій:

- робота з представленнями активного документа (Views, вивести список представлень для документа, перегляд даних щодо активного представлення, інтерактивна взаємодія з Revit: Zoom In, Zoom Out і Zoom to Fit, пошук об'єктів зі списку доступних представлень);

- робота з ескізами/листами активного документа (Sheets, вивести список листів для документа, перегляд даних щодо активного листа, інтерактивна взаємодія з Revit, пошук об'єктів зі списку доступних листів);
- робота з кімнатами на рівні активного представлення (Rooms, вивести список кімнат для представлення, перегляд даних щодо активної кімнати, інтерактивна взаємодія з Revit: Zoom to Room, пошук потрібної кімнати);
- робота з дверима на рівні активного представлення (Doors, вивести список дверей для представлення, перегляд даних щодо активної двері, інтерактивна взаємодія з Revit: Zoom to Door, пошук потрібної двері);
- робота з підключеними документами (RevitLinkInstance, вивести дерево рішень для підключених документів, вивести кімнати і двері для кожного документа, перегляд даних щодо активного підключеного документа);
- робота з генерацією метаданих (вивести список листів на рівні активного документа, надати для користувача можливість самостійно вибирати листи перед відправкою на Azure, перегляд актуальної інформації щодо створених даних, на основі вибраних листів згенерувати метадані та їх картинки)
- робота з відправленням даних на Azure Storage Account (переглянути інформацію щодо списку метаданих для відправки, надати для користувача можливість видалити метадані, відправлення серіалізованої інформації на Azure).

Висновки./Conclusions. Результатом є інструментальна надбудова для системи Autodesk Revit 2019-2021, що реалізує інформаційну технологію перетворення параметрів моделі при трансфері між системами візуалізації.

Додаток В

Презентація

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА НА ТЕМУ

Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень

АКТУАЛЬНІСТЬ ТЕМИ



У сучасному виробництві корисне поширення одержали системи автоматизованого проектування, які надають можливість створювати технологічні процеси з меншими витратами часу та інженерних засобів.

Системи автоматизованого проектування є важливою ланкою в промисловому конструюванні, широко використовуваним в багатьох галузях, в тому числі в автомобільній, суднобудівній та аерокосмічній індустрії, промисловому і архітектурному проектуванні, протезуванні і багатьох інших.

На сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення. Як правило, такі надбудови використовуються у функціональній інфраструктурі спеціалізованого комплексу рішень, що реалізують принцип інформаційного моделювання будівель (Building Information Modeling).

МЕТА ТА ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Мета кваліфікаційної роботи полягає в створенні методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору, що дозволить реалізувати алгоритм перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit.

Для досягнення поставленої мети визначені наступні задачі-дослідження:

- ▶ формулювання методики перетворення систем координат;
- ▶ реалізація фільтра для визначення об'єктів, що підлягають трансферу;
- ▶ визначення необхідного переліку функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- ▶ побудова інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- ▶ реалізація інструментарію для хмарного зберігання даних;
- ▶ реалізація перетворення даних до універсального формату зберігання та подальшого трансферу;
- ▶ виконати алгоритмічну та програмну реалізацію методу пониження розмірностей векторного простору.

ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єкт дослідження – процес пониження розмірностей векторного простору для архітектурно-будівельних елементів при переході між інтегрованими системами координат в умовах автоматизованого проектування.

Предмет дослідження – моделі, алгоритми та засоби для побудови методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень.

НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

В результаті проведеної кваліфікаційної роботи були отримані наступні результати:

- ▶ вдосконалено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом зменшення розмірностей векторного простору, що дозволило оптимізувати склад об'єктів, які підлягають трансферу між системами візуалізації. Відмінність від відомих підходів полягає у попередній фільтрації переліку об'єктів у поєднанні з переліком необхідних їх функціональних параметрів;
- ▶ вдосконалено методику моделювання систем при перетворенні систем координат, використання яких, на відміну від відомих, дозволило у явному вигляді представити координати об'єктів в інструментальній надбудові архітектурно-будівельної платформи Autodesk Revit.

ПРАКТИЧНЕ ЗНАЧЕННЯ ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

Розроблено метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень. В результаті проведених досліджень було створено застосування відносно перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit. Надбудова дає можливість:

- ▶ використовувати технології хмарного зберігання даних за допомогою програмного інструментарію.
- ▶ застосувати для збереження даних універсальний формат, зручний для розширеного та (або) віддаленого доступу.

АНАЛІЗ ПО РОЗДІЛУ 1

Встановлено, що для роботи методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок зменшення розмірностей векторного простору буде використано платформу архітектурного проектування и конструювання Autodesk Revit.

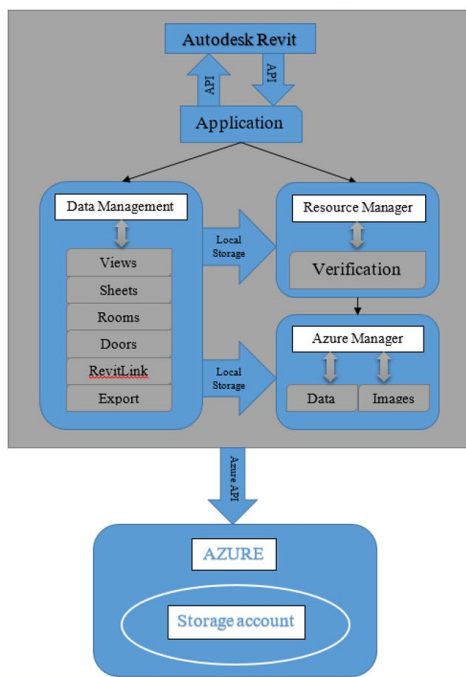
Результати виконання методу будуть зберігатися в обліковому записі зберігання даних в складі хмарного сервісу Microsoft Azure.



Microsoft Azure
Services



АНАЛІЗ ПО РОЗДІЛУ 2



Встановлено, що для автоматизації обробки інформаційних потоків в складі конструкторської надбудови передбачається реалізація наступного переліку бізнес-процесів:

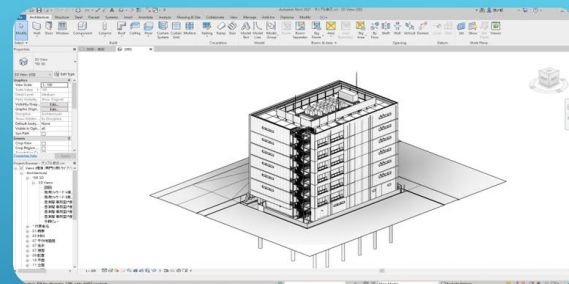
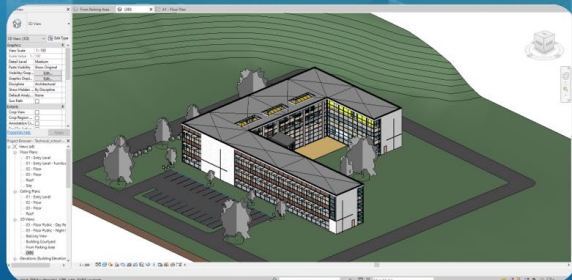
- ▶ робота з відображеннями активного документа;
- ▶ робота з технічними листами активного документа;
- ▶ робота з кімнатами на рівні активного вигляду;
- ▶ робота з дверима на рівні активного вигляду;
- ▶ робота з підключеними документа типу Revit Links;
- ▶ робота з генерацією метаданих;
- ▶ робота з відправленням даних на Azure File Storage.

АНАЛІЗ ПО РОЗДІЛУ 3

| Назва модуля | Функції | Опис |
|--|---|---|
| Стартовий програмний модуль | Запуск основної сесії інструментальної надбудови | Активується у випадку, якщо Autodesk Revit успішно зареєструє файл маніфесту. |
| WPF модуль | Засоби для генерування компонентів UI | Складається з двох основних вікон: інструмент для роботи з елементами активного документа та вікно для управління результатами виконання методу пониження розмірностей векторного простору. |
| Інформаційна модель даних | Автоматизацію обробки даних від програмного інтерфейсу Revit API. | Базовий програмний шаблон зберігає методи опрацювання над елементами активного документа, такими як вигляди, технічні листи, двері та кімнати. |
| Модуля зберігання та генерації метаданих | Запуск та зберігання результатів перетворення координат для архітектурно-будівельних сутностей. | Шаблони серіалізації, що забезпечують можливість зберігання результуючих даних відносно математичного перетворення координат дверей та кімнат з модельного представлення до системи координат паперового простору |

АНАЛІЗ ПО РОЗДІЛУ 4

З метою перевірки та підтвердження результатів виконання методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за допомогою пониження розмірностей векторного простору було проведено апробацію програмної системи.



Три типові проекти Autodesk Revit, а саме:

- ▶ модель приватного будинку;
- ▶ модель багатоповерхового житлового комплексу;
- ▶ модель шкільного закладу

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра було розроблено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору для системи автоматизованого проектування Autodesk Revit. Програмний продукт дозволить користувачам платформи архітектурної візуалізації використовувати функції перегляду, опрацювання, математичного перетворення та серіалізації над елементами будівельних 3D моделей. Для реалізації інструментальної надбудови було обрано технологію .Net Framework 4.8, середовище розробки Microsoft Visual Studio Enterprise 2019, Revit API та Autodesk Revit 2021.

Для досягнення поставленої мети були виконані наступні задачі-дослідження:

- ▶ сформульовано методу перетворення систем координат;
- ▶ реалізовано фільтр для визначення об'єктів, що підлягають трансферу;
- ▶ визначено необхідний перелік функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- ▶ побудовано інформаційну модель методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- ▶ реалізовано інструментарій для хмарного зберігання даних;
- ▶ реалізовано алгоритм перетворення даних до універсального формату зберігання та подальшого трансферу;
- ▶ алгоритмічна та програмна реалізація методу пониження розмірностей векторного простору.

В рамках дослідження були визначені та охарактеризовані функціональні потоки інформаційної системи, описана взаємодія між програмними модулями.

Проведена системна апробація основних функцій архітектурно-будівельної надбудови. Для тестування алгоритмів генерації метаданих були проаналізовані три типових проекта Autodesk Revit, а саме моделі приватного будинку, багатоповерхового житлового комплексу та 3-х поверхового шкільного закладу. Результати перетворення координат для вищезгаданих будівель завершені без хибної ідентифікації та явних помилок.

Додаток Г

Програмні коди

```

Файл App.cs
using Autodesk.Revit.Attributes;
using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
using RevitProject.Views;
using System.Diagnostics;
using System.Drawing;
using System.Reflection;
using System.Runtime.InteropServices;
using System.Windows;
using System.Windows.Interop;
using System.Windows.Media;
using System.Windows.Media.Imaging;

namespace RevitProject.Main
{
    public class App : IExternalApplication
    {
        private const string pathToDll =
@"C:\Users\roman\source\repos\RevitAddinProject\RevitPr
oject.Main\bin\x64\Release\RevitProject.Main.dll";

        private readonly string ribbonTab =
Properties.Resource.RibbonTabName;

        private readonly string dataPanel =
Properties.Resource.PanelDataName;
        private readonly string sendPanel =
Properties.Resource.PanelSendName;
        private readonly string toolsPanel =
Properties.Resource.PanelToolsName;

        private readonly string documentButton =
Properties.Resource.DocumentButtonName;
        private readonly string sendButton =
Properties.Resource.SendButtonName;
        private readonly string aboutButton =
Properties.Resource.AboutButtonName;
        private readonly string siteButton =
Properties.Resource.SiteButtonName;

        private ImageSource IconToImageSource(Icon
icon)
        {
            return Imaging.CreateBitmapSourceFromHIcon(
                icon.Handle,
                new Int32Rect(0, 0, icon.Width,
icon.Height),
                BitmapSizeOptions.FromEmptyOptions());
        }

        public Result OnStartup(UIControlledApplication
application)
        {
            application.CreateRibbonTab(ribbonTab);

            RibbonPanel ribbonPanelData =
application.CreateRibbonPanel(ribbonTab, dataPanel);

            PushButtonData docButton = new
PushButtonData(documentButton, documentButton,
pathToDll, typeof(RevitData).FullName);

            docButton.Image =
IconToImageSource(Properties.Resource.Document);
            docButton.LargeImage =
IconToImageSource(Properties.Resource.Document32);
            ribbonPanelData.AddItem(docButton);

            RibbonPanel ribbonPanelSend =
application.CreateRibbonPanel(ribbonTab, sendPanel);

            PushButtonData azureButton = new
PushButtonData(sendButton, sendButton, pathToDll,
typeof(SendData).FullName);
            azureButton.Image =
IconToImageSource(Properties.Resource.Azure);
            azureButton.LargeImage =
IconToImageSource(Properties.Resource.Azure32);
            ribbonPanelSend.AddItem(azureButton);

            RibbonPanel ribbonPanelTools =
application.CreateRibbonPanel(ribbonTab, toolsPanel);

            PushButtonData versionButton = new
PushButtonData(aboutButton, aboutButton, pathToDll,
typeof(About).FullName);
            versionButton.Image =
IconToImageSource(Properties.Resource.About);
            versionButton.LargeImage =
IconToImageSource(Properties.Resource.About32);
            ribbonPanelTools.AddItem(versionButton);

            PushButtonData portalButton = new
PushButtonData(siteButton, siteButton, pathToDll,
typeof(Site).FullName);
            portalButton.Image =
IconToImageSource(Properties.Resource.Portal);
            portalButton.LargeImage =
IconToImageSource(Properties.Resource.Portal32);
            ribbonPanelTools.AddItem(portalButton);

            return Result.Succeeded;
        }
    }

    public Result
OnShutdown(UIControlledApplication application) =>
Result.Succeeded;
}

[Transaction(TransactionMode.Manual)]
[Regeneration(RegenerationOption.Manual)]
public class RevitData : IExternalCommand
{
    public Result Execute(ExternalCommandData
commandData, ref string message, ElementSet elements)
    {
        if (new
MainWindow(commandData.Application).ShowDialog() ??
false)
            return Result.Succeeded;
        else
            return Result.Failed;
    }
}

```

```

[Transaction(TransactionMode.Manual)]
[Regeneration(RegenerationOption.Manual)]
public class SendData : IExternalCommand
{
    public Result Execute(ExternalCommandData
commandData, ref string message, ElementSet elements)
    {
        if (new AzureWindow().ShowDialog() ??
false)
            return Result.Succeeded;
        else
            return Result.Failed;
    }
}

[Transaction(TransactionMode.Manual)]
[Regeneration(RegenerationOption.Manual)]
public class About : IExternalCommand
{
    public Result Execute(ExternalCommandData
commandData, ref string message, ElementSet elements)
    {
        MessageBox.Show(
FileVersionInfo.GetVersionInfo(Assembly.GetExecutingAss
embly().Location).FileVersion,
        "Information",
        MessageBoxButton.OK,
        MessageBoxImage.Information);
        return Result.Succeeded;
    }
}

[Transaction(TransactionMode.Manual)]
[Regeneration(RegenerationOption.Manual)]
public class Site : IExternalCommand
{
    public Result Execute(ExternalCommandData
commandData, ref string message, ElementSet elements)
    {
        string url = "https://portal.azure.com";

        try
        {
            Process.Start(url);
        }
        catch
        {
            if
(RuntimeInformation.IsOSPlatform(OSPlatform.Windows))
            {
                url = url.Replace("&", "^&");
                Process.Start(new
ProcessStartInfo("cmd", $"/c start {url}") {
CreateNoWindow = true });
            }
            else if
(RuntimeInformation.IsOSPlatform(OSPlatform.Linux))
                Process.Start("xdg-open", url);
            else if
(RuntimeInformation.IsOSPlatform(OSPlatform.OSX))
                Process.Start("open", url);
        }

        return Result.Succeeded;
    }
}
}

```

Файл ExportService.cs

```

using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using Newtonsoft.Json;
using RevitProject.Views.Utils;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace RevitProject.Views.Export
{
    public static class ExportService
    {
        private static readonly string OutputDirectory
=
Path.Combine(Environment.GetFolderPath(Environment.Spec
ialFolder.ApplicationData), "Diploma");

        public static string OutputFile { get; set; }
        public static bool IsFileGenerated { get; set;
} = false;

        private static double hostMinZ;
        private static double scaleView;
        private static string fileName;
        private static View correspondingView;
        private static UV ptSourceViewOriginInSheet;
        private static Transform tInv;
        private static XYZ minimumPointForImage;

        private static IEnumerable<Level> levels;
        private static IEnumerable<RevitLinkInstance>
instances;
        private static IEnumerable<FamilyInstance>
hostedDoors;
        private static IEnumerable<IndependentTag>
doorTags;
        private static IEnumerable<RoomTag> roomTags;
        private static List<ViewSheet> validSheets;

        private static ViewSheet _viewSheet;

        private static string GetNameForJsonFile() =>
$"Revit_Json_{RevitHelper.Document.Title}.json";

        public static void InitializeRevitData(int
countOfSheets)
        {
            if (countOfSheets > 0)
            {
                levels =
RevitHelper.GetHelper().Levels;
                instances =
RevitHelper.GetHelper().Instances;
                hostedDoors =
RevitHelper.GetHelper().HostedDoors;
                doorTags =
RevitHelper.GetHelper().DoorTags;
                roomTags =
RevitHelper.GetHelper().RoomTags;
            }
        }

        public static void
SetViewSheetInformation(List<ViewSheet> sheets)
        {
            if
(Directory.Exists(Path.Combine(OutputDirectory,
RevitHelper.Document.Title)))

```

```

Directory.Delete(Path.Combine(OutputDirectory,
RevitHelper.Document.Title), true);

Directory.CreateDirectory(Path.Combine(OutputDirectory,
RevitHelper.Document.Title));

    validSheets = new List<ViewSheet>();
    fileName = GetNameForJsonFile();

    List<RevitSheet> sheetsJson = new
List<RevitSheet>();
    foreach (ViewSheet sheet in sheets)
    {
        _viewSheet = sheet;
        RevitSheet viewSheetInformation = new
RevitSheet
        {
            ViewSheetNumber =
sheet.SheetNumber,
            ViewSheetName = sheet.Name,
            Scale = sheet.Scale.ToString(),
            Unit = "Feet"
        };

        minimumPointForImage = new
XYZ(sheet.Outline.Min.U, sheet.Outline.Min.V, 0);
        using (Transaction t = new
Transaction(RevitHelper.Document, "Creating a Text
note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options = new
TextNoteOptions();
                options.HorizontalAlignment =
HorizontalTextAlignment.Center;
                options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementTyp
eGroup.TextNoteType);
                TextNote note =
TextNote.Create(RevitHelper.Document, sheet.Id,
minimumPointForImage, "PDF", options);

                TextElementType textType =
note.Symbol;
                BuiltInParameter paraIndex =
BuiltInParameter.TEXT_SIZE;
                Parameter textSize =
textType.get_Parameter(paraIndex);
                double newSize = (1.0 / 6.0) *
(1.0 / 18.0);
                textSize.Set(newSize);
                t.Commit();
            }
            catch { }
        }

        List<Viewport> viewports =
sheet.GetAllViewports().Select(port =>
RevitHelper.Document.GetElement(port) as
Viewport).Where(port =>
    {
        ViewPlan viewPlan =
RevitHelper.Document.GetElement(port.ViewId) as
ViewPlan;
        if (viewPlan != null)
            return
viewPlan.ViewType.Equals(ViewType.FloorPlan);

                return false;
            }).ToList();

        List<RevitDoor> doorsForExport = new
List<RevitDoor>();
        List<RevitRoom> roomsForExport = new
List<RevitRoom>();
        foreach (Viewport viewport in
viewports)
        {
            correspondingView =
RevitHelper.Document.GetElement(viewport.ViewId) as
View;

            using (Transaction t = new
Transaction(RevitHelper.Document, "Hide elements"))
            {
                try
                {
                    t.Start();

                    correspondingView.HideElements(doorTags.Select(doorTag
=> doorTag.Id).ToList());

                    correspondingView.HideElements(roomTags.Select(roomTag
=> roomTag.Id).ToList());

                    t.Commit();
                }
                catch
                {
                    // Nothing to do.
                }
            }

            tInv =
correspondingView.CropBox.Transform.Inverse;

            Parameter level =
correspondingView.LookupParameter("Associated Level");
            Level actualLevel =
levels.FirstOrDefault(lvl =>
lvl.Name.Equals(level.AsString()));
            hostMinZ = actualLevel.Elevation;

            doorsForExport.AddRange(SetJsonInformation(viewport,
out List<RevitRoom> jsonRooms));
            jsonRooms.ForEach(jsonRoom =>
            {
                if
(!roomsForExport.Any(roomForExport =>
roomForExport.Guid.Equals(jsonRoom.Guid)))
                roomsForExport.Add(jsonRoom);
            });

            viewSheetInformation.DoorsOfSheet =
doorsForExport;
            viewSheetInformation.RoomsOfSheet =
roomsForExport;
            if
(viewSheetInformation.DoorsOfSheet.Any()
            validSheets.Add(sheet);

            sheetsJson.Add(viewSheetInformation);
        }

        GenerateJsonFile(sheetsJson);

        if (validSheets.Any())

```

```

        GenerateImageForSheet();
    }

    private static List<RevitDoor>
    SetJsonInformation(Viewport viewport, out
    List<RevitRoom> jsonRooms)
    {
        jsonRooms = new List<RevitRoom>();

        scaleView = correspondingView.Scale;
        ptSourceViewOriginInSheet =
        GetPtSourceViewOriginInSheet(viewport);

        List<RevitDoor> doorForJson = new
        List<RevitDoor>();

        doorForJson.AddRange(GetInformationAboutHostedDoors(out
        List<RevitRoom> roomsJsonHosted));
        if (roomsJsonHosted.Count != 0)
            jsonRooms.AddRange(roomsJsonHosted);

        doorForJson.AddRange(GetInformationAboutLinkedDoors(out
        List<RevitRoom> roomsJsonLinked));
        if (roomsJsonLinked.Count != 0)
            jsonRooms.AddRange(roomsJsonLinked);

        return doorForJson;
    }

    private static UV
    GetPtSourceViewOriginInSheet(Viewport viewport)
    {
        UV outlineMax = new
        UV(correspondingView.Outline.Max.U,
        correspondingView.Outline.Max.V);
        UV outlineMin = new
        UV(correspondingView.Outline.Min.U,
        correspondingView.Outline.Min.V);
        UV centerOutline = (outlineMax +
        outlineMin) / 2;
        UV centerOutlineInModel = new
        UV(centerOutline.U / (1.0 / scaleView), centerOutline.V
        / (1.0 / scaleView));

        XYZ portOutlineCenter =
        viewport.GetBoxCenter();
        return new UV(portOutlineCenter.X -
        (centerOutlineInModel.U * (1.0 / scaleView)),
        portOutlineCenter.Y - (centerOutlineInModel.V * (1.0 /
        scaleView)));
    }

    private static void
    GenerateJsonFile(List<RevitSheet> sheetsJson)
    {
        try
        {
            string pathForJson =
            Path.Combine(Path.Combine(OutputDirectory,
            RevitHelper.Document.Title), fileName);

            RevitDocument documentJson = new
            RevitDocument { Sheets = sheetsJson };
            string json =
            JsonConvert.SerializeObject(documentJson,
            Formatting.Indented);
            File.WriteAllText(pathForJson, json);

            IsFileGenerated = true;
        }
    }

    OutputFile = pathForJson;
}
catch
{
    IsFileGenerated = false;
}
}

private static void GenerateImageForSheet()
{
    ImageExportOptions options = new
    ImageExportOptions()
    {
        ExportRange = ExportRange.SetOfViews,
        FilePath =
        Path.Combine(OutputDirectory,
        RevitHelper.Document.Title) + @"\ ",
        ImageResolution =
        ImageResolution.DPI_600,
        ZoomType = ZoomFitType.Zoom,
        Zoom = 100,
        ShadowViewsFileType = ImageFileType.PNG
    };

    options.SetViewsAndSheets(validSheets.Select(sheet =>
    sheet.Id).ToList());
    RevitHelper.Document.ExportImage(options);
}

private static List<RevitDoor>
GetInformationAboutHostedDoors(out List<RevitRoom>
roomsJson)
{
    roomsJson = new List<RevitRoom>();
    List<RevitDoor> doorForJson = new
    List<RevitDoor>();

    Parameter viewPhase =
    correspondingView.get_Parameter(BuiltInParameter.VIEW_P
    HASE);
    string currentPhaseName =
    viewPhase?.AsString();
    Phase currentPhase = null;
    foreach (Phase phase in
    RevitHelper.Document.Phases)
    {
        if (phase.Name == currentPhaseName)
        {
            currentPhase = phase;
            break;
        }
    }

    foreach (FamilyInstance doorVisible in
    hostedDoors)
    {
        try
        {
            var levelOfDoor =
            RevitHelper.Document.GetElement(doorVisible.LevelId) as
            Level;
            if (levelOfDoor is null)
                continue;

            if
            (!levelOfDoor.Elevation.Equals(hostMinZ) ||
            !IsElementVisibleInView(correspondingView,
            doorVisible))
                continue;
        }
    }
}

```

```

        BoundingBoxXYZ boxForDoor =
doorVisible.get_BoundingBox(correspondingView);
        XYZ pointMax =
tInv.OfPoint(boxForDoor.Max);
        XYZ pointMin =
tInv.OfPoint(boxForDoor.Min);
        XYZ boxForDoorMax = new
XYZ(pointMax.X * (1.0 / scaleView) +
ptSourceViewOriginInSheet.U, pointMax.Y * (1.0 /
scaleView) + ptSourceViewOriginInSheet.V, 0);
        XYZ boxForDoorMin = new
XYZ(pointMin.X * (1.0 / scaleView) +
ptSourceViewOriginInSheet.U, pointMin.Y * (1.0 /
scaleView) + ptSourceViewOriginInSheet.V, 0);

        using (Transaction t = new
Transaction(RevitHelper.Document, "Creating a Text
note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options =
new TextNoteOptions();
                options.HorizontalAlignment
= HorizontalTextAlignment.Center;
                options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementTyp
eGroup.TextNoteType);

                TextNote.Create(RevitHelper.Document, _viewSheet.Id,
boxForDoorMax,
doorVisible.get_Parameter(BuiltInParameter.DOOR_NUMBER)
.AsString() + "-Max", options);
                t.Commit();
            }
            catch { }
        }

        using (Transaction t = new
Transaction(RevitHelper.Document, "Creating a Text
note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options =
new TextNoteOptions();
                options.HorizontalAlignment
= HorizontalTextAlignment.Center;
                options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementTyp
eGroup.TextNoteType);

                TextNote.Create(RevitHelper.Document, _viewSheet.Id,
boxForDoorMin,
doorVisible.get_Parameter(BuiltInParameter.DOOR_NUMBER)
.AsString() + "-Min", options);
                t.Commit();
            }
            catch
            {
            }
        }

        XYZ doorTagPoint;
        IndependentTag doorTag =
doorTags?.FirstOrDefault(tag =>
tag.TaggedElementId.HostElementId.Equals(doorVisible.Id
));

        if (doorTag is null)
            doorTagPoint = (boxForDoorMax +
boxForDoorMin) / 2;
        else
        {
            XYZ tagHeadPositionTransformed
= tInv.OfPoint(doorTag.TagHeadPosition);
            doorTagPoint = new
XYZ(tagHeadPositionTransformed.X * (1 / scaleView) +
ptSourceViewOriginInSheet.U,
tagHeadPositionTransformed.Y * (1 / scaleView) +
ptSourceViewOriginInSheet.V, 0);
        }

        using (Transaction t = new
Transaction(RevitHelper.Document, "Creating a Text
note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options =
new TextNoteOptions();
                options.HorizontalAlignment
= HorizontalTextAlignment.Center;
                options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementTyp
eGroup.TextNoteType);

                TextNote.Create(RevitHelper.Document, _viewSheet.Id,
doorTagPoint,
doorVisible.get_Parameter(BuiltInParameter.DOOR_NUMBER)
.AsString(), options);
                t.Commit();
            }
            catch { }
        }

        doorTagPoint -=
minimumPointForImage;
        Room toRoom =
doorVisible.get_ToRoom(currentPhase);
        Room fromRoom =
doorVisible.get_FromRoom(currentPhase);
        List<Room> hostRooms = new
List<Room> { toRoom, fromRoom };
        RevitRoom toRoomJson = null;
        RevitRoom fromRoomJson = null;
        foreach (Room room in hostRooms)
        {
            if (room is null)
                continue;

            List<RevitCoordinates>
boundarySegments = new List<RevitCoordinates>();
            IList<IList<BoundarySegment>>
segments = room.GetBoundarySegments(new
SpatialElementBoundaryOptions());
            if (segments != null)
            {
                foreach
                (IList<BoundarySegment> segmentList in segments)
                {
                    foreach
                    (BoundarySegment boundarySegment in segmentList)
                    {
                        XYZ start =
boundarySegment.GetCurve().GetEndPoint(0);
                        XYZ
startTransformed = tInv.OfPoint(start);

```

```

        XYZ PointForRoom =
new XYZ(startTransformed.X * (1.0 / scaleView) +
ptSourceViewOriginInSheet.U, startTransformed.Y * (1.0
/ scaleView) + ptSourceViewOriginInSheet.V, 0);

        //using
(Transaction t = new Transaction(RevitHelper.Document,
"Creating a Text note"))
        //{
        //    try
        //    {
t.Start();
        //
        TextNoteOptions options = new TextNoteOptions();
        //
options.HorizontalAlignment =
HorizontalTextAlignment.Center;
        //
options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementType
eGroup.TextNoteType);
        //
TextNote.Create(RevitHelper.Document, _viewSheet.Id,
PointForRoom, room.Name, options);
        //
t.Commit();
        //    }
        //    catch { }
        //}
        PointForRoom -=
minimumPointForImage;

boundarySegments.Add(new RevitCoordinates { X =
Convert.ToDecimal(PointForRoom.X), Y =
Convert.ToDecimal(PointForRoom.Y) });
    }
}

        XYZ roomTagPoint;
RoomTag roomTag =
roomTags?.FirstOrDefault(tag =>
tag.TaggedRoomId.HostElementId.Equals(room.Id));
        if (roomTag is null)
        {
            LocationPoint infoPoint =
room.Location as LocationPoint;
            XYZ pointTransformed =
tInv.OfPoint(infoPoint.Point);
            roomTagPoint = new
XYZ(pointTransformed.X * (1 / scaleView) +
ptSourceViewOriginInSheet.U, pointTransformed.Y * (1 /
scaleView) + ptSourceViewOriginInSheet.V, 0);
        }
        else
        {
            XYZ
tagHeadPositionTransformed =
tInv.OfPoint(roomTag.TagHeadPosition);
            roomTagPoint = new
XYZ(tagHeadPositionTransformed.X * (1 / scaleView) +
ptSourceViewOriginInSheet.U,
tagHeadPositionTransformed.Y * (1 / scaleView) +
ptSourceViewOriginInSheet.V, 0);
        }

        using (Transaction t = new
Transaction(RevitHelper.Document, "Creating a Text
note"))
    {
        try
        {
            t.Start();
            TextNoteOptions options
= new TextNoteOptions();
            options.HorizontalAlignment =
HorizontalTextAlignment.Center;
            options.TypeId =
RevitHelper.Document.GetDefaultElementTypeId(ElementType
eGroup.TextNoteType);
            TextNote.Create(RevitHelper.Document, _viewSheet.Id,
roomTagPoint, room.Name, options);
            t.Commit();
        }
        catch { }
    }
}

roomTagPoint -=
minimumPointForImage;
if
(hostRooms.IndexOf(room).Equals(0))
{
    toRoomJson = new RevitRoom
    {
        Guid = room.UniqueId,
        Name = room.Name,
        Number = room.Number,
        IsLinked = false,
        TagCoordinate = new
RevitCoordinates
        {
            X =
Convert.ToDecimal(roomTagPoint.X),
            Y =
Convert.ToDecimal(roomTagPoint.Y)
        },
        BoundarySegments =
boundarySegments
    };
}
else
{
    fromRoomJson = new
RevitRoom
    {
        Guid = room.UniqueId,
        Name = room.Name,
        Number = room.Number,
        IsLinked = false,
        TagCoordinate = new
RevitCoordinates
        {
            X =
Convert.ToDecimal(roomTagPoint.X),
            Y =
Convert.ToDecimal(roomTagPoint.Y)
        },
        BoundarySegments =
boundarySegments
    };
}
}

boxForDoorMax -=
minimumPointForImage;
boxForDoorMin -=
minimumPointForImage;
doorForJson.Add(new RevitDoor

```

```

        {
            Guid = doorVisible.UniqueId,
            Name = doorVisible.Name,
            Mark =
doorVisible.get_Parameter(BuiltInParameter.DOOR_NUMBER)
.AsString(),
            IsLinked = false,
            Level =
RevitHelper.Document.GetElement(doorVisible.LevelId)?.Name ?? string.Empty,
            MaxCoordinate = new
RevitCoordinates
            {
                X =
Convert.ToDecimal(boxForDoorMax.X),
                Y =
Convert.ToDecimal(boxForDoorMax.Y)
            },
            MinCoordinate = new
RevitCoordinates
            {
                X =
Convert.ToDecimal(boxForDoorMin.X),
                Y =
Convert.ToDecimal(boxForDoorMin.Y)
            },
            TagCoordinate = new
RevitCoordinates
            {
                X =
Convert.ToDecimal(doorTagPoint.X),
                Y =
Convert.ToDecimal(doorTagPoint.Y)
            },
            ToRoomGuid = toRoomJson?.Guid
?? string.Empty,
            FromRoomGuid =
fromRoomJson?.Guid ?? string.Empty
        });
        if (!roomsJson.Any(roomJson =>
roomJson.Guid.Equals(toRoomJson?.Guid)) && toRoomJson
!= null)
            roomsJson.Add(toRoomJson);
        if (!roomsJson.Any(roomJson =>
roomJson.Guid.Equals(fromRoomJson?.Guid)) &&
fromRoomJson != null)
            roomsJson.Add(fromRoomJson);
    }
    catch
    {
        // If something went wrong with
getting the coordinates of the door in the model, we
will not include it currently
    }
}
return doorForJson;
}

private static List<RevitDoor>
GetInformationAboutLinkedDoors(out List<RevitRoom>
roomsJson)
{
    roomsJson = new List<RevitRoom>();
    List<RevitDoor> doorForJson = new
List<RevitDoor>();

    foreach (var instance in instances)
    {

```

```

        if
(instance.IsHidden(correspondingView)
|| instance.GetLinkDocument() is
null
||
((RevitLinkType)instance.Document.GetElement(instance.G
etTypeId())).IsNestedLink)
            continue;

        Phase phase =
GetLinkInstancePhaseForView(instance,
correspondingView);
        Document linkDocument =
instance.GetLinkDocument();

        IEnumerable<FamilyInstance>
actualLinkedDoors =
RevitHelper.GetDoorsFromLinkedDocument(linkDocument);
        IEnumerable<Level> linkedLevels =
RevitHelper.GetLevelsFromLinkDocument(linkDocument);

        var transform =
instance.GetTotalTransform();
        foreach (FamilyInstance doorLinked in
actualLinkedDoors)
        {
            try
            {
                if
(!DoesLinkedElementPassPhaseFilter(instance,
correspondingView, doorLinked))
                    continue;

                BoundingBoxXYZ boundingBox =
doorLinked.get_BoundingBox(correspondingView);
                Level actualLevel =
linkDocument.GetElement(doorLinked.LevelId) as Level;
                Level upLevel =
linkedLevels.FirstOrDefault(linkedLevel =>
linkedLevel.Elevation > actualLevel.Elevation);

                double actualLevelMax;
                if (upLevel is null)
                    actualLevelMax =
transform.OfPoint(boundingBox.Max).Z;
                else
                    actualLevelMax =
transform.OfPoint(new XYZ(0, 0, upLevel.Elevation)).Z;

                var viewCorner1 =
transform.OfPoint(boundingBox.Min);
                var viewCorner2 =
transform.OfPoint(boundingBox.Max);
                var numberIndex =
(viewCorner2.Z - viewCorner1.Z) / 2;
                var viewCorner1Z =
viewCorner1.Z - numberIndex;
                if (viewCorner1Z > hostMinZ ||
actualLevelMax < hostMinZ)
                    continue;

                XYZ pointMin =
tInv.OfPoint(viewCorner1);
                XYZ pointMax =
tInv.OfPoint(viewCorner2);

                XYZ boxForDoorMin = new
XYZ(pointMin.X * (1.0 / scaleView) +
ptSourceViewOriginInSheet.U, pointMin.Y * (1.0 /
scaleView) + ptSourceViewOriginInSheet.V, 0);

```

```

        XYZ boxForDoorMax = new
XYZ(pointMax.X * (1.0 / scaleView) +
ptSourceViewOriginInSheet.U, pointMax.Y * (1.0 /
scaleView) + ptSourceViewOriginInSheet.V, 0);

        using (Transaction t = new
Transaction(instance.Document, "Creating a Text note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options
= new TextNoteOptions();

options.HorizontalAlignment =
HorizontalTextAlignment.Center;
                options.TypeId =
instance.Document.GetDefaultElementTypeId(ElementTypeGr
oup.TextNoteType);

                TextNote.Create(instance.Document, _viewSheet.Id,
boxForDoorMin,
doorLinked.get_Parameter(BuiltInParameter.DOOR_NUMBER).
AsString() + "-Min", options);
                t.Commit();
            }
            catch { }
        }

        using (Transaction t = new
Transaction(instance.Document, "Creating a Text note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options
= new TextNoteOptions();

options.HorizontalAlignment =
HorizontalTextAlignment.Center;
                options.TypeId =
instance.Document.GetDefaultElementTypeId(ElementTypeGr
oup.TextNoteType);

                TextNote.Create(instance.Document, _viewSheet.Id,
boxForDoorMax,
doorLinked.get_Parameter(BuiltInParameter.DOOR_NUMBER).
AsString() + "-Max", options);
                t.Commit();
            }
            catch { }
        }

        XYZ doorLinkedTagPoint;
        IndependentTag doorLinkedTag =
doorTags?.FirstOrDefault(tag =>
tag.TaggedElementId.LinkInstanceId.Equals(instance.Id)
&&
tag.TaggedElementId.LinkElementId.Equals(doorLinked.I
d));

        if (doorLinkedTag is null)
            doorLinkedTagPoint =
(boxForDoorMax + boxForDoorMin) / 2;
        else
        {
            XYZ
tagHeadPositionTransformed =
tInv.OfPoint(doorLinkedTag.TagHeadPosition);
            doorLinkedTagPoint = new
XYZ(tagHeadPositionTransformed.X * (1 / scaleView) +

```

```

ptSourceViewOriginInSheet.U,
tagHeadPositionTransformed.Y * (1 / scaleView) +
ptSourceViewOriginInSheet.V, 0);
        }

        using (Transaction t = new
Transaction(instance.Document, "Creating a Text note"))
        {
            try
            {
                t.Start();
                TextNoteOptions options
= new TextNoteOptions();

options.HorizontalAlignment =
HorizontalTextAlignment.Center;
                options.TypeId =
instance.Document.GetDefaultElementTypeId(ElementTypeGr
oup.TextNoteType);

                TextNote.Create(instance.Document, _viewSheet.Id,
doorLinkedTagPoint,
doorLinked.get_Parameter(BuiltInParameter.DOOR_NUMBER).
AsString(), options);
                t.Commit();
            }
            catch { }
        }

        doorLinkedTagPoint -=
minimumPointForImage;
        Room toRoomLinked =
doorLinked.get_ToRoom(phase);
        Room fromRoomLinked =
doorLinked.get_FromRoom(phase);
        List<Room> linkedRooms = new
List<Room> { toRoomLinked, fromRoomLinked };
        RevitRoom toRoomJson = null;
        RevitRoom fromRoomJson = null;
        foreach (Room roomLinked in
linkedRooms)
        {
            if (roomLinked is null)
                continue;

            List<RevitCoordinates>
boundarySegments = new List<RevitCoordinates>();

            IList<IList<BoundarySegment>> segments =
roomLinked.GetBoundarySegments(new
SpatialElementBoundaryOptions());
            if (segments != null)
            {
                foreach
                (IList<BoundarySegment> segmentList in segments)
                {
                    foreach
                    (BoundarySegment boundarySegment in segmentList)
                    {
                        XYZ start =
boundarySegment.GetCurve().GetEndPoint(0);
                        XYZ
pointTrasformed = transform.OfPoint(start);
                        XYZ
pointTransformedFromCropBox =
tInv.OfPoint(pointTrasformed);
                        XYZ
PointForRoom = new XYZ(pointTransformedFromCropBox.X *
(1.0 / scaleView) + ptSourceViewOriginInSheet.U,
pointTransformedFromCropBox.Y * (1.0 / scaleView) +
ptSourceViewOriginInSheet.V, 0);

```

```

//using
(Transaction t = new Transaction(instance.Document,
"Creating a Text note"))
//{
//  try
//  {
t.Start();
//
TextNoteOptions options = new TextNoteOptions();
//
options.HorizontalAlignment =
HorizontalTextAlignment.Center;
//
options.TypeId =
instance.Document.GetDefaultElementTypeId(ElementTypeGr
oup.TextNoteType);
//
TextNote.Create(instance.Document, _viewSheet.Id,
PointForRoom, roomLinked.Name, options);
//
t.Commit();
//  }
//  catch { }
//}
PointForRoom -=
minimumPointForImage;
boundarySegments.Add(new RevitCoordinates { X =
Convert.ToDecimal(PointForRoom.X), Y =
Convert.ToDecimal(PointForRoom.Y) });
}
}
XYZ roomLinkedTagPoint;
RoomTag roomLinkedTag =
roomTags?.FirstOrDefault(tag =>
tag.TaggedRoomId.LinkInstanceId.Equals(instance.Id) &&
tag.TaggedRoomId.LinkedElementId.Equals(roomLinked.Id))
;
if (roomLinkedTag == null)
{
LocationPoint infoPoint
= roomLinked.Location as LocationPoint;
XYZ
infoPointTransformed =
transform.OfPoint(infoPoint.Point);
XYZ
infoPointTransformedCropBox =
tInv.OfPoint(infoPointTransformed);
roomLinkedTagPoint =
new XYZ(infoPointTransformedCropBox.X * (1 / scaleView)
+ ptSourceViewOriginInSheet.U,
infoPointTransformedCropBox.Y * (1 / scaleView) +
ptSourceViewOriginInSheet.V, 0);
}
else
{
XYZ
tagHeadPositionTransformed =
tInv.OfPoint(roomLinkedTag.TagHeadPosition);
roomLinkedTagPoint =
new XYZ(tagHeadPositionTransformed.X * (1 / scaleView)
+ ptSourceViewOriginInSheet.U,
tagHeadPositionTransformed.Y * (1 / scaleView) +
ptSourceViewOriginInSheet.V, 0);
}
}

```

```

using (Transaction t = new
Transaction(instance.Document, "Creating a Text note"))
{
try
{
t.Start();
TextNoteOptions
options = new TextNoteOptions();
options.HorizontalAlignment =
HorizontalTextAlignment.Center;
options.TypeId =
instance.Document.GetDefaultElementTypeId(ElementTypeGr
oup.TextNoteType);
TextNote.Create(instance.Document, _viewSheet.Id,
roomLinkedTagPoint, roomLinked.Name, options);
t.Commit();
}
catch { }
}
roomLinkedTagPoint -=
minimumPointForImage;
if
(linkedRooms.IndexOf(roomLinked).Equals(0))
{
toRoomJson = new
RevitRoom
{
Guid =
roomLinked.UniqueId,
Name =
roomLinked.Name,
Number =
roomLinked.Number,
IsLinked = true,
TagCoordinate = new
RevitCoordinates
{
X =
Convert.ToDecimal(roomLinkedTagPoint.X),
Y =
Convert.ToDecimal(roomLinkedTagPoint.Y)
},
BoundarySegments =
boundarySegments
};
}
else
{
fromRoomJson = new
RevitRoom
{
Guid =
roomLinked.UniqueId,
Name =
roomLinked.Name,
Number =
roomLinked.Number,
IsLinked = true,
TagCoordinate = new
RevitCoordinates
{
X =
Convert.ToDecimal(roomLinkedTagPoint.X),
Y =
Convert.ToDecimal(roomLinkedTagPoint.Y)
},
BoundarySegments =
boundarySegments
}
}
}
}

```



```

        return true;
        var status =
element.GetPhaseStatus(phase.Id);

        // Only New, Existing, Demolished, and
Temporary statuses are the valid values to get
PhaseStatusPresentation.
        var validStatuses = new
List<ElementOnPhaseStatus>()
        {
            ElementOnPhaseStatus.New,
            ElementOnPhaseStatus.Existing,
            ElementOnPhaseStatus.Demolished,
            ElementOnPhaseStatus.Temporary
        };
        if (!validStatuses.Contains(status))
            return false;
        return
phaseFilter.GetPhaseStatusPresentation(element.GetPhase
Status(phase.Id)) != PhaseStatusPresentation.DontShow;
    }
    catch (Exception)
    {
        return true;
    }
}

private static Phase
GetLinkInstancePhaseForView(RevitLinkInstance instance,
View view)
{
    Phase phase = null;
    Document document = instance.Document;
    Document linkDocument =
instance.GetLinkDocument();

    if
(typeof(RevitLinkType).Assembly.GetName().Version >=
Version.Parse("19.0.0.0"))
    {
        RevitLinkType type =
document.GetElement(instance.GetTypeId()) as
RevitLinkType;
        Type systemType =
typeof(RevitLinkType);
        var map =
systemType.GetMethod("GetPhaseMap").Invoke(type, null)
as IDictionary<ElementId, ElementId>;
        phase =
linkDocument.GetElement(map[view.get_Parameter(BuiltInP
arameter.VIEW_PHASE).AsElementId()]) as Phase;
    }

    return phase;
}
}
}

```

Файл RevitDocument.cs

```

using System.Collections.Generic;
using System.Runtime.Serialization;

namespace RevitProject.Views.Export
{
    [DataContract]
    public class RevitDocument
    {
        [DataMember(Name = "sheets", Order = 1)]
        public List<RevitSheet> Sheets { get; set; }
    }
}

```

```

}

```

Файл RevitSheet.cs

```

using System.Collections.Generic;
using System.Runtime.Serialization;

namespace RevitProject.Views.Export
{
    [DataContract]
    public class RevitSheet
    {
        [DataMember(Name = "sheetNumber", Order = 1)]
        public string ViewSheetNumber { get; set; }

        [DataMember(Name = "sheetName", Order = 2)]
        public string ViewSheetName { get; set; }

        [DataMember(Name = "scale", Order = 3)]
        public string Scale { get; set; }

        [DataMember(Name = "unit", Order = 4)]
        public string Unit { get; set; }

        [DataMember(Name = "doors", Order = 5)]
        public List<RevitDoor> DoorsOfSheet { get; set; }

        [DataMember(Name = "rooms", Order = 6)]
        public List<RevitRoom> RoomsOfSheet { get; set; }
    }
}
}
}

```

Файл RevitDoor.cs

```

using System.Runtime.Serialization;

namespace RevitProject.Views.Export
{
    [DataContract]
    public class RevitDoor
    {
        [DataMember(Name = "guid", Order = 1)]
        public string Guid { get; set; }

        [DataMember(Name = "name", Order = 2)]
        public string Name { get; set; }

        [DataMember(Name = "mark", Order = 3)]
        public string Mark { get; set; }

        [DataMember(Name = "isLinked", Order = 4)]
        public bool IsLinked { get; set; }

        [DataMember(Name = "level", Order = 5)]
        public string Level { get; set; }

        [DataMember(Name = "max", Order = 6)]
        public RevitCoordinates MaxCoordinate { get;
set; }

        [DataMember(Name = "min", Order = 7)]
        public RevitCoordinates MinCoordinate { get;
set; }

        [DataMember(Name = "tag", Order = 8)]
        public RevitCoordinates TagCoordinate { get;
set; }
    }
}
}

```

```

        [DataMember(Name = "toRoom", Order = 9)]
        public string ToRoomGuid { get; set; }

        [DataMember(Name = "fromRoom", Order = 10)]
        public string FromRoomGuid { get; set; }
    }
}

```

Файл RevitRoom.cs

```

using System.Collections.Generic;
using System.Runtime.Serialization;

namespace RevitProject.Views.Export
{
    [DataContract]
    public class RevitRoom
    {
        [DataMember(Name = "guid", Order = 1)]
        public string Guid { get; set; }

        [DataMember(Name = "name", Order = 2)]
        public string Name { get; set; }

        [DataMember(Name = "number", Order = 3)]
        public string Number { get; set; }

        [DataMember(Name = "isLinked", Order = 4)]
        public bool IsLinked { get; set; }

        [DataMember(Name = "tag", Order = 5)]
        public RevitCoordinates TagCoordinate { get;
set; }

        [DataMember(Name = "boundarySegments", Order =
6)]
        public List<RevitCoordinates> BoundarySegments
{ get; set; }
    }
}

```

Файл RevitCoordinates.cs

```

using System.Runtime.Serialization;

namespace RevitProject.Views.Export
{
    [DataContract]
    public class RevitCoordinates
    {
        [DataMember(Name = "x", Order = 1)]
        public decimal X { get; set; }

        [DataMember(Name = "y", Order = 2)]
        public decimal Y { get; set; }
    }
}

```

Файл AzureModel.cs

```

using Newtonsoft.Json;
using RevitProject.Views.Export;
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Windows;
using System.Windows.Controls;

```

```

namespace RevitProject.Views.Models
{
    public class AzureModel : RevitModel
    {
        private readonly string outputDirectory =
Path.Combine(Environment.GetFolderPath(Environment.Spec
ialFolder.ApplicationData), "Diploma");

        private AzureWindow azureWindow;

        private List<ListBoxItem> selectedItems;
        private Dictionary<string, List<RevitSheet>>
jsonSheets;

        public AzureModel(AzureWindow azureWindow)
        {
            this.azureWindow = azureWindow;
            jsonSheets = new Dictionary<string,
List<RevitSheet>>();

            selectedItems = new List<ListBoxItem>();
            string[] activeDirectories =
Directory.GetDirectories(outputDirectory);
            foreach (var activeDirectory in
activeDirectories)
            {
                ListBoxItem item = new ListBoxItem
                {
                    Content = new
DirectoryInfo(activeDirectory).Name
                };

                item.Selected += Item_Selected;
                item.Unselected += Item_Unselected;

                azureWindow.Folders.Items.Add(item);

                string[] files =
Directory.GetFiles(activeDirectory);

                string json = string.Empty;
                using (FileStream fs = new
FileStream(files.FirstOrDefault(file =>
file.Contains("json")), FileMode.Open,
FileAccess.Read))
                    using (StreamReader reader = new
StreamReader(fs))
                        json = reader.ReadToEnd();

                jsonSheets.Add(new
DirectoryInfo(activeDirectory).Name,
JsonConvert.DeserializeObject<RevitDocument>(json).Shee
ts);
            }

            azureWindow.DeleteFolders.Click +=
DeleteFolders_Click;
            azureWindow.AzureButton.Click +=
AzureButton_Click;
            azureWindow.FoldersSearch.Click +=
FoldersSearch_Click;
            azureWindow.FoldersRestore.Click +=
FoldersRestore_Click;
        }

        private void DeleteFolders_Click(object sender,
RoutedEventArgs e)
        {
            string[] activeDirectories =
Directory.GetDirectories(outputDirectory);

```

```

        foreach (var activeDirectory in
activeDirectories)
            if (selectedItems.Any(selectedItem =>
selectedItem.Content.Equals(new
DirectoryInfo(activeDirectory).Name)))
                Directory.Delete(activeDirectory,
true);

        var collection =
azureWindow.Folders.Items.SourceCollection
            .OfType<ListBoxItem>()
            .Where(box =>
!selectedItems.Any(selectedItem =>
selectedItem.Content.Equals(box.Content)));

        if (collection.Any())
        {
            ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];
collection.ToList().CopyTo(updatedArray);
            azureWindow.Folders.Items.Clear();
            foreach (ListBoxItem listBoxItem in
updatedArray)
                azureWindow.Folders.Items.Add(listBoxItem);
        }
        else
        {
            azureWindow.Folders.Items.Clear();
            azureWindow.DeleteFolders.IsEnabled =
false;
            azureWindow.AzureButton.IsEnabled =
false;
        }

        azureWindow.StatusInfoAzure.Text = "Local
deletion of Revit data completed successfully!";
    }

    private void AzureButton_Click(object sender,
RoutedEventArgs e)
    {
        var path =
Path.Combine(Path.GetDirectoryName(Assembly.GetExecutin
gAssembly().Location),
@"Deploy\RevitProject.Azure.exe");
        if (File.Exists(path))
        {
            var process = Process.Start(path,
string.Join("~", selectedItems.Select(selectedItem =>
selectedItem.Content as string)));
            process.WaitForExit();

            if (process.ExitCode == 0)
                azureWindow.StatusInfoAzure.Text =
"The transferring of Revit data completed
successfully!";
            else
                azureWindow.StatusInfoAzure.Text =
"The transferring of Revit data completed with
errors!";
        }
    }

    private void FoldersRestore_Click(object
sender, RoutedEventArgs e) => RestoreElements();

    private void FoldersSearch_Click(object sender,
RoutedEventArgs e) => SearchForElement();

```

```

        private void Item_Selected(object sender,
RoutedEventArgs e) => ElementSelected(sender);

        private void Item_Unselected(object sender,
RoutedEventArgs e) => ElementUnselected(sender);

        protected override void RestoreElements()
        {
            azureWindow.Folders.Items.Clear();
            jsonSheets.Clear();

            string[] activeDirectories =
Directory.GetDirectories(outputDirectory);
            foreach (var activeDirectory in
activeDirectories)
            {
                ListBoxItem item = new ListBoxItem
                {
                    Content = new
DirectoryInfo(activeDirectory).Name
                };

                item.Selected += Item_Selected;
                item.Unselected += Item_Unselected;

                azureWindow.Folders.Items.Add(item);

                string[] files =
Directory.GetFiles(activeDirectory);

                string json = string.Empty;
                using (FileStream fs = new
FileStream(files.FirstOrDefault(file =>
file.Contains("json")), FileMode.Open,
FileAccess.Read))
                    using (StreamReader reader = new
StreamReader(fs))
                        json = reader.ReadToEnd();

                jsonSheets.Add(new
DirectoryInfo(activeDirectory).Name,
JsonConvert.DeserializeObject<RevitDocument>(json).Shee
ts);
            }

            azureWindow.FoldersRestore.IsEnabled =
false;
        }

        protected override void SearchForElement()
        {
            string query =
azureWindow.fdsSearch.Text.Trim().ToLower();
            if (string.IsNullOrEmpty(query))
                return;

            var collection =
azureWindow.Folders.Items.SourceCollection
                .OfType<ListBoxItem>()
                .Where(box => (box.Content as
string).ToLower().Contains(query));
            ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];
            collection.ToList().CopyTo(updatedArray);
            azureWindow.Folders.Items.Clear();
            foreach (ListBoxItem listBoxItem in
updatedArray)
                azureWindow.Folders.Items.Add(listBoxItem);

```

```

        azureWindow.FoldersRestore.IsEnabled =
true;
    }

    protected override void ElementSelected(object
sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        if (!selectedItems.Contains(actualItem))
            selectedItems.Add(actualItem);

        if (selectedItems.Any())
        {
            azureWindow.DeleteFolders.IsEnabled =
true;
            azureWindow.AzureButton.IsEnabled =
true;

            azureWindow.TextBox_Files.Text =
selectedItems.Count.ToString();
            azureWindow.TextBox_Connection.Text =
true.ToString();
            azureWindow.TextBox_Share.Text =
"fileshare";

            var actualSheets = jsonSheets
                .Where(jsonSheet =>
selectedItems.Any(selectedItem =>
selectedItem.Content.Equals(jsonSheet.Key)))
                .SelectMany(pair => pair.Value);

            azureWindow.TextBox_Doors.Text =
actualSheets
                .SelectMany(sheet =>
sheet.DoorsOfSheet)
                .Count().ToString();
            azureWindow.TextBox_Rooms.Text =
actualSheets
                .SelectMany(sheet =>
sheet.RoomsOfSheet)
                .Count().ToString();
            azureWindow.TextBox_Link.Text =
actualSheets
                .Any(sheet =>
sheet.DoorsOfSheet.Any(door =>
door.IsLinked)).ToString();
            azureWindow.TextBox_Quota.Text = "5GB";
            azureWindow.TextBox_Units.Text =
"Feet";
            azureWindow.TextBox_Sheets.Text =
actualSheets.Count().ToString();
        }
    }

    protected override void
ElementUnselected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        selectedItems.Remove(actualItem);

        if (selectedItems.Any())
        {
            azureWindow.DeleteFolders.IsEnabled =
true;
            azureWindow.AzureButton.IsEnabled =
true;

            azureWindow.TextBox_Files.Text =
selectedItems.Count.ToString();

            azureWindow.TextBox_Connection.Text =
true.ToString();
            azureWindow.TextBox_Share.Text =
"fileshare";

            var actualSheets = jsonSheets
                .Where(jsonSheet =>
selectedItems.Any(selectedItem =>
selectedItem.Content.Equals(jsonSheet.Key)))
                .SelectMany(pair => pair.Value);

            azureWindow.TextBox_Doors.Text =
actualSheets
                .SelectMany(sheet =>
sheet.DoorsOfSheet)
                .Count().ToString();
            azureWindow.TextBox_Rooms.Text =
actualSheets
                .SelectMany(sheet =>
sheet.RoomsOfSheet)
                .Count().ToString();
            azureWindow.TextBox_Link.Text =
actualSheets
                .Any(sheet =>
sheet.DoorsOfSheet.Any(door =>
door.IsLinked)).ToString();
            azureWindow.TextBox_Quota.Text = "5GB";
            azureWindow.TextBox_Units.Text =
"Feet";
            azureWindow.TextBox_Sheets.Text =
actualSheets.Count().ToString();
        }
        else
        {
            azureWindow.DeleteFolders.IsEnabled =
false;
            azureWindow.AzureButton.IsEnabled =
false;

            azureWindow.TextBox_Files.Text =
string.Empty;
            azureWindow.TextBox_Connection.Text =
string.Empty;
            azureWindow.TextBox_Share.Text =
string.Empty;
            azureWindow.TextBox_Doors.Text =
string.Empty;
            azureWindow.TextBox_Rooms.Text =
string.Empty;
            azureWindow.TextBox_Link.Text =
string.Empty;
            azureWindow.TextBox_Quota.Text =
string.Empty;
            azureWindow.TextBox_Units.Text =
string.Empty;
            azureWindow.TextBox_Sheets.Text =
string.Empty;
        }
    }
}

Файл DoorsModel.cs

using System.Collections.Generic;
using System.Linq;
using System.Windows.Controls;
using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
using RevitProject.Views.Utills;

```

```

namespace RevitProject.Views.Models
{
    public class DoorsModel : RevitModel
    {
        private MainWindow mainWindow;

        private FloorPlansModel viewModel;
        private IEnumerable<FamilyInstance>
doors;

        public DoorsModel(UIApplication
application, FloorPlansModel viewModel,
MainWindow mainWindow)
        {
            this.application = application;
            this.mainWindow = mainWindow;

            this.viewModel = viewModel;

            mainWindow.ZoomToDoor.Click +=
ZoomToDoor_Click;
            mainWindow.ZoomToFitDoor.Click +=
ZoomToFitDoor_Click;
        }

        private void ZoomToFitDoor_Click(object
sender, System.Windows.RoutedEventArgs e)
        {
            UIView uIView =
application.ActiveUIDocument.GetOpenUIViews().F
irstOrDefault(uiView =>
uiView.ViewId.Equals(viewModel.ActivatedElement
));
            uIView.ZoomToFit();

            mainWindow.ZoomToFitDoor.IsEnabled
= false;
            mainWindow.StatusInfo.Text =
"You've clicked the button Zoom To Fit.";
        }

        private void ZoomToDoor_Click(object
sender, System.Windows.RoutedEventArgs e)
        {
            application.ActiveUIDocument.ShowElements(Activ
atedElement);

            mainWindow.ZoomToFitDoor.IsEnabled
= true;
            mainWindow.StatusInfo.Text =
"You've clicked the button Zoom To Door.";
        }

        public void SetDoorForCollection()
        {
            mainWindow.ViewsBoxDoors.Items.Clear();

            doors =
RevitHelper.GetDoorsOfView(RevitHelper.Document
, viewModel.ActivatedElement);

            if (!doors.Any())
                mainWindow.ZoomToDoor.IsEnabled
= false;

            if
(mainWindow.ViewsBoxDoors.Items.Count.Equals(do
ors.Count()))
                return;

            foreach (FamilyInstance door in
doors)
            {
                ListBoxItem item = new
ListBoxItem
                {
                    Content = door.Name + " - "
+
door.get_Parameter(BuiltInParameter.DOOR_NUMBER
).AsString()
                };
                item.Selected += Item_Selected;
                item.Unselected +=
Item_Unselected;

                mainWindow.ViewsBoxDoors.Items.Add(item);
            }

            if (doors.Any())
            {
                mainWindow.DoorSearch.Click +=
DoorSearch_Click;
                mainWindow.DoorRestore.Click +=
DoorRestore_Click;
            }
            else
                mainWindow.DoorSearch.IsEnabled
= false;

                mainWindow.DoorRestore.IsEnabled =
false;
        }

        private void DoorRestore_Click(object
sender, System.Windows.RoutedEventArgs e) =>
RestoreElements();

        private void DoorSearch_Click(object
sender, System.Windows.RoutedEventArgs e) =>
SearchForElement();

        private void Item_Selected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementSelected(sender);

```

```

        private void Item_Unselected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementUnselected(sender);

        protected override void
RestoreElements()
        {
mainWindow.ViewsBoxDoors.Items.Clear();
        foreach (FamilyInstance door in
doors)
        {
            ListBoxItem item = new
ListBoxItem
            {
                Content = door.Name + " - "
+
door.get_Parameter(BuiltInParameter.DOOR_NUMBER
).AsString()
            };
            item.Selected += Item_Selected;
            item.Unselected +=
Item_Unselected;

mainWindow.ViewsBoxDoors.Items.Add(item);
        }

        mainWindow.DoorRestore.IsEnabled =
false;
    }

    protected override void
SearchForElement()
    {
        string query =
mainWindow.tbDoors.Text.Trim().ToLower();
        if (string.IsNullOrEmpty(query) &&
mainWindow.ViewsBoxDoors.Items.Count.Equals(doo
rs.Count()))
            return;

        var collection =
mainWindow.ViewsBoxDoors.Items.SourceCollection
.OfType<ListBoxItem>()
.Where(box => (box.Content as
string).ToLower().Contains(query));
        ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];

collection.ToList().CopyTo(updatedArray);

mainWindow.ViewsBoxDoors.Items.Clear();
        foreach (ListBoxItem listBoxItem in
updatedArray)

mainWindow.ViewsBoxDoors.Items.Add(listBoxItem)
;

```

```

        mainWindow.DoorRestore.IsEnabled =
true;
    }

    protected override void
ElementSelected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        FamilyInstance activeDoor =
doors.FirstOrDefault(door =>
actualItem.Content.Equals(door.Name + " - " +
door.get_Parameter(BuiltInParameter.DOOR_NUMBER
).AsString()));
        ActivatedElement = activeDoor.Id;

        mainWindow.ZoomToDoor.IsEnabled =
true;

        mainWindow.NameTextDoor.Text =
activeDoor.Name;
        mainWindow.MarkDoorText.Text =
activeDoor.get_Parameter(BuiltInParameter.DOOR_
NUMBER).AsString();
        mainWindow.LevelDoorText.Text =
(RevitHelper.Document.GetElement(activeDoor.Lev
elId) as Level).Name;
        mainWindow.LevelDoorText.ToolTip =
new ToolTip { Content =
(RevitHelper.Document.GetElement(activeDoor.Lev
elId) as Level).Name };
        mainWindow.DocumentDoorText.Text =
activeDoor.Document.Title;
        mainWindow.DocumentDoorText.ToolTip
= new ToolTip { Content =
activeDoor.Document.Title };
        mainWindow.FromRoomText.Text =
activeDoor.FromRoom?.Name ?? "-";
        mainWindow.FromRoomText.ToolTip =
new ToolTip { Content =
activeDoor.FromRoom?.Name ?? "-" };
        mainWindow.ToRoomText.Text =
activeDoor.ToRoom?.Name ?? "-";
        mainWindow.ToRoomText.ToolTip = new
ToolTip { Content = activeDoor.ToRoom?.Name ??
"-" };

        mainWindow.CategoryDoorText.Text =
activeDoor.Category.Name;
        mainWindow.IsValidDoorText.Text =
activeDoor.IsValidObject.ToString();
        mainWindow.PhaseDoorText.Text =
(RevitHelper.Document.GetElement(activeDoor.Cre
atedPhaseId) as Phase)?.Name ?? "-";
        mainWindow.PhaseDoorText.ToolTip =
new ToolTip { Content =
(RevitHelper.Document.GetElement(activeDoor.Cre
atedPhaseId) as Phase)?.Name ?? "-" };
        mainWindow.OwnerViewDoorText.Text =
(RevitHelper.Document.GetElement(activeDoor.Own
erViewId) as View)?.Name ?? "-";

```

```

        mainWindow.InvisibleDoorText.Text =
activeDoor.Invisible.ToString();
        mainWindow.PinnedDoorText.Text =
activeDoor.Pinned.ToString();
    }

    protected override void
ElementUnselected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        FamilyInstance activeDoor =
doors.FirstOrDefault(door =>
actualItem.Content.Equals(door.Name + " - " +
door.get_Parameter(BuiltInParameter.DOOR_NUMBER
).AsString()));

        if
(ActivatedElement.Equals(activeDoor.Id))
        {
            ActivatedElement = default;

            mainWindow.ZoomToDoor.IsEnabled
= false;

            mainWindow.NameTextDoor.Text =
string.Empty;
            mainWindow.MarkDoorText.Text =
string.Empty;
            mainWindow.LevelDoorText.Text =
string.Empty;

            mainWindow.LevelDoorText.ToolTip = null;

            mainWindow.DocumentDoorText.Text =
string.Empty;

            mainWindow.DocumentDoorText.ToolTip = null;
            mainWindow.FromRoomText.Text =
string.Empty;
            mainWindow.FromRoomText.ToolTip
= null;
            mainWindow.ToRoomText.Text =
string.Empty;
            mainWindow.ToRoomText.ToolTip =
null;

            mainWindow.CategoryDoorText.Text =
string.Empty;
            mainWindow.IsValidDoorText.Text =
string.Empty;
            mainWindow.PhaseDoorText.Text =
string.Empty;

            mainWindow.PhaseDoorText.ToolTip = null;

            mainWindow.OwnerViewDoorText.Text =
string.Empty;

```

```

mainWindow.InvisibleDoorText.Text =
string.Empty;
        mainWindow.PinnedDoorText.Text
= string.Empty;
    }
}
}
}

```

Файл ExportModel.cs

```

using Autodesk.Revit.DB;
using RevitProject.Views.Export;
using RevitProject.Views.Utills;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using System.Windows.Controls;

namespace RevitProject.Views.Models
{
    public class ExportModel : RevitModel
    {
        private MainWindow mainWindow;

        private List<ListBoxItem>
selectedItems;

        public ExportModel(MainWindow
mainWindow)
        {
            this.mainWindow = mainWindow;

            selectedItems = new
List<ListBoxItem>();
            mainWindow.ProcessResult.Click +=
ProcessResult_Click;

            var sheets =
RevitHelper.GetHelper().ViewSheets;
            foreach (ViewSheet sheet in sheets)
            {
                ListBoxItem item = new
ListBoxItem
                {
                    Content = sheet.Title
                };
                item.Selected += Item_Selected;
                item.Unselected +=
Item_Unselected;

                mainWindow.JsonList.Items.Add(item);
            }

            private bool
IsCollectionValidForExport()
            {

```

```

        IEnumerable<ViewSheet> sheets =
RevitHelper.GetHelper().ViewSheets.Where(sheet
=> selectedItems.Any(item =>
item.Content.Equals(sheet.Title)));
        return sheets.Any(sheet =>
        {
            IEnumerable<Viewport> ports =
sheet.GetAllViewports().Select(port =>
RevitHelper.Document.GetElement(port) as
Viewport);
            return ports.Any(port =>
            {
                ViewPlan viewPlan =
RevitHelper.Document.GetElement(port.ViewId) as
ViewPlan;
                if (viewPlan != null)
                    return
viewPlan.ViewType.Equals(ViewType.FloorPlan);
            });
        });
    }

    private int CountOfDoors()
    {
        IEnumerable<ViewSheet> sheets =
RevitHelper.GetHelper().ViewSheets.Where(sheet
=> selectedItems.Any(item =>
item.Content.Equals(sheet.Title)));
        return sheets.SelectMany(sheet =>
        {
            IEnumerable<View>
collectionOfView =
sheet.GetAllViewports().Select(port =>
            {
                Viewport info =
RevitHelper.Document.GetElement(port) as
Viewport;
                return
RevitHelper.Document.GetElement(info.ViewId) as
View;
            }).Where(view => view is
ViewPlan &&
view.ViewType.Equals(ViewType.FloorPlan));
            return
collectionOfView.SelectMany(view =>
RevitHelper.GetDoorsOfView(RevitHelper.Document
, view.Id));
        }).Count();
    }

    private int CountOfRooms()
    {
        IEnumerable<ViewSheet> sheets =
RevitHelper.GetHelper().ViewSheets.Where(sheet
=> selectedItems.Any(item =>
item.Content.Equals(sheet.Title)));
        return sheets.SelectMany(sheet =>
        {
            IEnumerable<View>
collectionOfView =
sheet.GetAllViewports().Select(port =>
            {
                Viewport info =
RevitHelper.Document.GetElement(port) as
Viewport;
                return
RevitHelper.Document.GetElement(info.ViewId) as
View;
            }).Where(view => view is
ViewPlan &&
view.ViewType.Equals(ViewType.FloorPlan));
            return
collectionOfView.SelectMany(view =>
RevitHelper.GetRoomsOfView(RevitHelper.Document
, view.Id));
        }).Count();
    }

    private void Item_Selected(object
sender, RoutedEventArgs e) =>
ElementSelected(sender);

    private void Item_Unselected(object
sender, RoutedEventArgs e) =>
ElementUnselected(sender);

    private void ProcessResult_Click(object
sender, RoutedEventArgs e)
    {
        var sheets =
RevitHelper.GetHelper().ViewSheets.Where(sheet
=> selectedItems.Any(item =>
item.Content.Equals(sheet.Title))).ToList();
        ExportService.InitializeRevitData(sheets.Count)
;
        ExportService.SetViewSheetInformation(sheets);
        if (ExportService.IsFileGenerated)
            mainWindow.StatusInfo.Text =
"Data generated successfully!";
        else
            mainWindow.StatusInfo.Text =
"Something went wrong while generating data!";
    }

    public bool ActivateButton() =>
IsCollectionValidForExport();

    protected override void
RestoreElements()
    {
        throw new
System.NotImplementedException();
    }

```



```

        mainWindow.ZoomOutButton.IsEnabled
= false;
mainWindow.ZoomToFitButton.IsEnabled = false;

        mainWindow.ZoomInButton.Click +=
ZoomInButton_Click;
        mainWindow.ZoomOutButton.Click +=
ZoomOutButton_Click;
        mainWindow.ZoomToFitButton.Click +=
ZoomToFitButton_Click;

        views =
RevitHelper.GetHelper().ViewPlans;
        foreach (ViewPlan viewPlan in
views)
    {
        ListBoxItem item = new
ListBoxItem
    {
        Content = viewPlan.Title
    };
        item.Selected += Item_Selected;
        item.Unselected +=
Item_Unselected;

mainWindow.ViewsBoxView.Items.Add(item);
    }

        if (views.Any())
    {
        mainWindow.ViewsSearch.Click +=
ViewsSearch_Click;
        mainWindow.ViewsRestore.Click
+= ViewsRestore_Click;
    }
        else

mainWindow.ViewsSearch.IsEnabled = false;

        mainWindow.ViewsRestore.IsEnabled =
false;
    }

        private void ViewsRestore_Click(object
sender, RoutedEventArgs e) =>
RestoreElements();

        private void ViewsSearch_Click(object
sender, RoutedEventArgs e) =>
SearchForElement();

        private void Item_Selected(object
sender, RoutedEventArgs e) =>
ElementSelected(sender);

        private void Item_Unselected(object
sender, RoutedEventArgs e) =>
ElementUnselected(sender);

```

```

        private void
ZoomToFitButton_Click(object sender,
RoutedEventArgs e)
    {
        UIView uUIView =
application.ActiveUIDocument.GetOpenUIViews().F
irstOrDefault(uiView =>
uiView.ViewId.Equals(ActivatedElement));
        uUIView.ZoomToFit();

        mainWindow.StatusInfo.Text =
"You've clicked the button Zoom To Fit.";
    }

        private void ZoomOutButton_Click(object
sender, RoutedEventArgs e)
    {
        UIView uUIView =
application.ActiveUIDocument.GetOpenUIViews().F
irstOrDefault(uiView =>
uiView.ViewId.Equals(ActivatedElement));
        uUIView.Zoom(0.9);

        mainWindow.StatusInfo.Text =
"You've clicked the button Zoom Out.";
    }

        private void ZoomInButton_Click(object
sender, RoutedEventArgs e)
    {
        UIView uUIView =
application.ActiveUIDocument.GetOpenUIViews().F
irstOrDefault(uiView =>
uiView.ViewId.Equals(ActivatedElement));
        uUIView.Zoom(1.1);

        mainWindow.StatusInfo.Text =
"You've clicked the button Zoom In.";
    }

        protected override void
RestoreElements()
    {
mainWindow.ViewsBoxView.Items.Clear();
        foreach (ViewPlan viewPlan in
views)
    {
        ListBoxItem item = new
ListBoxItem
    {
        Content = viewPlan.Title
    };
        item.Selected += Item_Selected;
        item.Unselected +=
Item_Unselected;

mainWindow.ViewsBoxView.Items.Add(item);
    }

```

```

    }

    mainWindow.ViewsRestore.IsEnabled =
false;
    }

    protected override void
SearchForElement()
    {
        string query =
mainWindow.tbView.Text.Trim().ToLower();
        if (string.IsNullOrEmpty(query) &&
mainWindow.ViewsBoxView.Items.Count.Equals(view
s.Count()))
            return;

        var collection =
mainWindow.ViewsBoxView.Items.SourceCollection
.OfType<ListBoxItem>()
.Where(box => (box.Content as
string).ToLower().Contains(query));
        ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];

collection.ToList().CopyTo(updatedArray);

mainWindow.ViewsBoxView.Items.Clear();
        foreach (ListBoxItem listBoxItem in
updatedArray)
mainWindow.ViewsBoxView.Items.Add(listBoxItem);

        mainWindow.ViewsRestore.IsEnabled =
true;
    }

    protected override void
ElementSelected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        ViewPlan activeView =
views.FirstOrDefault(plan =>
plan.Title.Equals(actualItem.Content));
        ActivatedElement = activeView.Id;

        mainWindow.ZoomInButton.IsEnabled =
true;
        mainWindow.ZoomOutButton.IsEnabled
= true;

mainWindow.ZoomToFitButton.IsEnabled = true;
        mainWindow.RoomsTab.IsEnabled =
true;
        mainWindow.DoorsTab.IsEnabled =
true;

        mainWindow.NameText.Text =
activeView.Name;

```

```

        mainWindow.NameText.ToolTip = new
ToolTip { Content = activeView.Name };
        mainWindow.DetailText.Text =
activeView.DetailLevel.ToString();
        mainWindow.DisciplineText.Text =
activeView.Discipline.ToString();
        mainWindow.DocumentText.Text =
activeView.Document.Title;
        mainWindow.DocumentText.ToolTip =
new ToolTip { Content =
activeView.Document.Title };
        mainWindow.OutlineText.Text =
string.Format("Max: {0}, {1}; Min: {2}, {3}.",
activeView.Outline.Max.U,
activeView.Outline.Max.V,
activeView.Outline.Min.U,
activeView.Outline.Min.V);
        mainWindow.OutlineText.ToolTip =
new ToolTip { Content = string.Format("Max:
{0}, {1}; Min: {2}, {3}.",
activeView.Outline.Max.U,
activeView.Outline.Max.V,
activeView.Outline.Min.U,
activeView.Outline.Min.V) };
        mainWindow.ScaleText.Text =
Convert.ToString(activeView.Scale);
        mainWindow.ViewTypeText.Text =
activeView.ViewType.ToString();
        mainWindow.PhaseText.Text =
activeView.get_Parameter(BuiltInParameter.VIEW_
PHASE)?.AsValueString() ?? "-";
        mainWindow.PhaseText.ToolTip = new
ToolTip { Content =
activeView.get_Parameter(BuiltInParameter.VIEW_
PHASE)?.AsValueString() ?? "-" };
        mainWindow.AssemblyViewText.Text =
activeView.IsAssemblyView.ToString();
        mainWindow.TemplateText.Text =
activeView.IsTemplate.ToString();
        mainWindow.TransientText.Text =
activeView.IsTransient.ToString();
        mainWindow.GenLevelText.Text =
activeView.GenLevel.Name;
        mainWindow.GenLevelText.ToolTip =
new ToolTip { Content =
activeView.GenLevel.Name };

application.ActiveUIDocument.ActiveView =
activeView;
    }

    protected override void
ElementUnselected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        ViewPlan activeView =
views.FirstOrDefault(plan =>
plan.Title.Equals(actualItem.Content));

```

```

        if
(ActivatedElement.Equals(activeView.Id))
        {
            ActivatedElement = default;

mainWindow.ZoomInButton.IsEnabled = false;
mainWindow.ZoomOutButton.IsEnabled = false;
mainWindow.ZoomToFitButton.IsEnabled = false;
mainWindow.RoomsTab.IsEnabled =
false;
mainWindow.DoorsTab.IsEnabled =
false;

mainWindow.NameText.Text =
string.Empty;
mainWindow.NameText.ToolTip =
null;
mainWindow.DetailText.Text =
string.Empty;
mainWindow.DisciplineText.Text
= string.Empty;
mainWindow.DocumentText.Text =
string.Empty;
mainWindow.DocumentText.ToolTip
= null;
mainWindow.OutlineText.Text =
string.Empty;
mainWindow.OutlineText.ToolTip
= null;
mainWindow.ScaleText.Text =
string.Empty;
mainWindow.ViewTypeText.Text =
string.Empty;
mainWindow.PhaseText.Text =
string.Empty;
mainWindow.PhaseText.ToolTip =
null;

mainWindow.AssemblyViewText.Text =
string.Empty;
mainWindow.TemplateText.Text =
string.Empty;
mainWindow.TransientText.Text =
string.Empty;
mainWindow.GenLevelText.Text =
string.Empty;
mainWindow.GenLevelText.ToolTip
= null;
        }
    }
}

```

Файл LinksModel.cs

```
using System.Collections.Generic;
```

```
using System.Collections.ObjectModel;
using System.Linq;
using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using RevitProject.Views.Utils;
```

```
namespace RevitProject.Views.Models
```

```
{
    public class Node
    {
        public string Name { get; set; }
        public ObservableCollection<Node> Nodes
{ get; set; }
    }

    public class LinksModel
    {
        public
LinksModel(IEnumerable<RevitLinkInstance>
instances, MainWindow mainWindow)
        {
            ObservableCollection<Node>
instanceNodes = new
ObservableCollection<Node>();
            foreach (RevitLinkInstance instance
in instances)
            {
                if (instance.GetLinkDocument()
== null ||
((RevitLinkType)instance.Document.GetElement(in
stance.GetTypeId())).IsNestedLink)
                    continue;

                Document linkDocument =
instance.GetLinkDocument();
                IEnumerable<Room>
actualLinkedRooms =
RevitHelper.GetRoomsFromLinkedDocument(linkDocu
ment);
                List<Node> linkedRooms =
actualLinkedRooms.Select(linkedRoom => new Node
{ Name = linkedRoom.Name }).ToList();
                IEnumerable<FamilyInstance>
actualLinkedDoors =
RevitHelper.GetDoorsFromLinkedDocument(linkDocu
ment);
                List<Node> linkedDoors =
actualLinkedDoors.Select(linkedDoor => new Node
{ Name = linkedDoor.Name + " - " +
linkedDoor.get_Parameter(BuiltInParameter.DOOR_
NUMBER).AsString() }).ToList();
                instanceNodes.Add(new Node
                {
                    Name = instance.Name,
                    Nodes = new
ObservableCollection<Node>
                {
                    new Node
                    {
                        Name = "Rooms",

```

```

        Nodes = new
ObservableCollection<Node>(linkedRooms)
    },
    new Node
    {
        Name = "Doors",
        Nodes = new
ObservableCollection<Node>(linkedDoors)
    }
    });
}

mainWindow.Instances.ItemsSource =
instanceNodes;
}
}

```

Файл RevitModel.cs

```

using Autodesk.Revit.DB;
using Autodesk.Revit.UI;

namespace RevitProject.Views.Models
{
    public abstract class RevitModel
    {
        public ElementId ActivatedElement {
get; protected set; }

        protected UIApplication application;

        protected abstract void
RestoreElements();
        protected abstract void
SearchForElement();
        protected abstract void
ElementSelected(object sender);
        protected abstract void
ElementUnselected(object sender);
    }
}

```

Файл RoomsModel.cs

```

using System.Collections.Generic;
using System.Linq;
using System.Windows.Controls;
using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using Autodesk.Revit.UI;
using RevitProject.Views.Utils;

namespace RevitProject.Views.Models
{
    public class RoomsModel : RevitModel
    {
        private MainWindow mainWindow;

```

```

        private FloorPlansModel viewModel;
        private IEnumerable<Room> rooms;

        public RoomsModel(UIApplication
application, FloorPlansModel viewModel,
MainWindow mainWindow)
        {
            this.application = application;
            this.mainWindow = mainWindow;

            this.viewModel = viewModel;

            mainWindow.ZoomToRoom.Click +=
ZoomToRoom_Click;
            mainWindow.ZoomToFitRoom.Click +=
ZoomToFitRoom_Click;
        }

        private void ZoomToFitRoom_Click(object
sender, System.Windows.RoutedEventArgs e)
        {
            UIView uUIView =
application.ActiveUIDocument.GetOpenUIViews().F
irstOrDefault(uiView =>
uiView.ViewId.Equals(viewModel.ActivatedElement
));
            uUIView.ZoomToFit();

            mainWindow.ZoomToFitRoom.IsEnabled
= false;
            mainWindow.StatusInfo.Text =
"You've clicked the button Zoom To Fit.";
        }

        private void ZoomToRoom_Click(object
sender, System.Windows.RoutedEventArgs e)
        {
            application.ActiveUIDocument.ShowElements(Activ
atedElement);

            mainWindow.ZoomToFitRoom.IsEnabled
= true;
            mainWindow.StatusInfo.Text =
"You've clicked the button Zoom To Room.";
        }

        public void SetRoomsForCollection()
        {
            mainWindow.ViewsBoxRooms.Items.Clear();
            rooms =
RevitHelper.GetRoomsOfView(RevitHelper.Document
, viewModel.ActivatedElement);

            if (!rooms.Any())
                mainWindow.ZoomToRoom.IsEnabled
= false;

```

```

        if
(mainWindow.ViewsBoxRooms.Items.Count.Equals(ro
oms.Count()))
            return;

        foreach (Room room in rooms)
        {
            ListBoxItem item = new
ListBoxItem
            {
                Content = room.Name + " - "
+ room.Number
            };
            item.Selected += Item_Selected;
            item.Unselected +=
Item_Unselected;

            mainWindow.ViewsBoxRooms.Items.Add(item);
        }

        if (rooms.Any())
        {
            mainWindow.RoomSearch.Click +=
RoomSearch_Click;
            mainWindow.RoomRestore.Click +=
RoomRestore_Click;
        }
        else
            mainWindow.RoomSearch.IsEnabled
= false;

            mainWindow.RoomRestore.IsEnabled =
false;
        }

        private void RoomRestore_Click(object
sender, System.Windows.RoutedEventArgs e) =>
RestoreElements();

        private void RoomSearch_Click(object
sender, System.Windows.RoutedEventArgs e) =>
SearchForElement();

        private void Item_Selected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementSelected(sender);

        private void Item_Unselected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementUnselected(sender);

        protected override void
RestoreElements()
        {
            mainWindow.ViewsBoxRooms.Items.Clear();
            foreach (Room room in rooms)
            {
                ListBoxItem item = new
ListBoxItem
                {
                    Content = room.Name + " - "
+ room.Number
                };
                item.Selected += Item_Selected;
                item.Unselected +=
Item_Unselected;

                mainWindow.ViewsBoxRooms.Items.Add(item);
            }

            mainWindow.RoomRestore.IsEnabled =
false;
        }

        protected override void
SearchForElement()
        {
            string query =
mainWindow.tbRooms.Text.Trim().ToLower();
            if (string.IsNullOrEmpty(query) &&
mainWindow.ViewsBoxRooms.Items.Count.Equals(ro
ms.Count()))
                return;

            var collection =
mainWindow.ViewsBoxRooms.Items.SourceCollection
.OfType<ListBoxItem>()
.Where(box => (box.Content as
string).ToLower().Contains(query));
            ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];
            collection.ToList().CopyTo(updatedArray);
            mainWindow.ViewsBoxRooms.Items.Clear();
            foreach (ListBoxItem listBoxItem in
updatedArray)
                mainWindow.ViewsBoxRooms.Items.Add(listBoxItem)
;

            mainWindow.RoomRestore.IsEnabled =
true;
        }

        protected override void
ElementSelected(object sender)
        {
            ListBoxItem actualItem = sender as
ListBoxItem;
            Room activeRoom =
rooms.FirstOrDefault(room =>
actualItem.Content.Equals(room.Name + " - " +
room.Number));
            ActivatedElement = activeRoom.Id;
        }

```

```

        mainWindow.ZoomToRoom.IsEnabled =
true;

        mainWindow.NameTextRoom.Text =
activeRoom.Name;
        mainWindow.AreaRoomText.Text =
activeRoom.Area.ToString();
        mainWindow.AreaRoomText.ToolTip =
new ToolTip { Content =
activeRoom.Area.ToString() };
        mainWindow.LevelRoomText.Text =
(RevitHelper.Document.GetElement(activeRoom.Lev
elId) as Level).Name;
        mainWindow.LevelRoomText.ToolTip =
new ToolTip { Content =
(RevitHelper.Document.GetElement(activeRoom.Lev
elId) as Level).Name };
        mainWindow.DocumentRoomText.Text =
activeRoom.Document.Title;
        mainWindow.DocumentRoomText.ToolTip
= new ToolTip { Content =
activeRoom.Document.Title };
        mainWindow.NumberTextRoom.Text =
activeRoom.Number;
        mainWindow.VolumeRoomText.Text =
activeRoom.Volume.ToString();
        mainWindow.CategoryRoomText.Text =
activeRoom.Category.Name;
        mainWindow.IsValidRoomText.Text =
activeRoom.IsValidObject.ToString();
        mainWindow.PhaseRoomText.Text =
activeRoom.get_Parameter(BuiltInParameter.ROOM_
PHASE)?.AsString() ?? "-";
        mainWindow.PhaseRoomText.ToolTip =
new ToolTip { Content =
activeRoom.get_Parameter(BuiltInParameter.ROOM_
PHASE)?.AsString() ?? "-" };
        mainWindow.OwnerViewRoomText.Text =
(RevitHelper.Document.GetElement(activeRoom.Own
erViewId) as View)?.Name ?? "-";
        mainWindow.PerimeterRoomText.Text =
activeRoom.Perimeter.ToString();

mainWindow.PerimeterRoomText.ToolTip = new
ToolTip { Content =
activeRoom.Perimeter.ToString() };
        mainWindow.PinnedRoomText.Text =
activeRoom.Pinned.ToString();
    }

    protected override void
ElementUnselected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        Room activeRoom =
rooms.FirstOrDefault(room =>
actualItem.Content.Equals(room.Name + " - " +
room.Number));

```

```

        if
(ActivatedElement.Equals(activeRoom.Id))
        {
            ActivatedElement = default;
            mainWindow.ZoomToRoom.IsEnabled
= false;
            mainWindow.NameTextRoom.Text =
string.Empty;
            mainWindow.AreaRoomText.Text =
string.Empty;
            mainWindow.AreaRoomText.ToolTip
= null;
            mainWindow.LevelRoomText.Text =
string.Empty;
            mainWindow.LevelRoomText.ToolTip = null;
            mainWindow.DocumentRoomText.Text =
string.Empty;
            mainWindow.DocumentRoomText.ToolTip = null;
            mainWindow.NumberTextRoom.Text
= string.Empty;
            mainWindow.VolumeRoomText.Text
= string.Empty;
            mainWindow.CategoryRoomText.Text =
string.Empty;
            mainWindow.IsValidRoomText.Text
= string.Empty;
            mainWindow.PhaseRoomText.Text =
string.Empty;
            mainWindow.PhaseRoomText.ToolTip = null;
            mainWindow.OwnerViewRoomText.Text =
string.Empty;
            mainWindow.PerimeterRoomText.Text =
string.Empty;
            mainWindow.PerimeterRoomText.ToolTip = null;
            mainWindow.PinnedRoomText.Text
= string.Empty;
        }
    }
}

```

Файл ViewSheetsModel.cs

```

using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
using RevitProject.Views.Utills;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Controls;

```

```

namespace RevitProject.Views.Models
{
    public class ViewSheetsModel : RevitModel
    {
        private MainWindow mainWindow;

        private IEnumerable<ViewSheet> sheets;

        public ViewSheetsModel(UIApplication
application, MainWindow mainWindow)
        {
            this.application = application;
            this.mainWindow = mainWindow;

            sheets =
RevitHelper.GetHelper().ViewSheets;
            foreach (ViewSheet sheet in sheets)
            {
                ListBoxItem item = new
ListBoxItem
                {
                    Content = sheet.Title
                };
                item.Selected += Item_Selected;
                item.Unselected +=
Item_Unselected;

                mainWindow.ViewsBoxSheet.Items.Add(item);
            }

            if (sheets.Any())
            {
                mainWindow.SheetSearch.Click +=
SheetSearch_Click;
                mainWindow.SheetRestore.Click
+= SheetRestore_Click;
            }
            else
            {
                mainWindow.SheetSearch.IsEnabled = false;

                mainWindow.SheetRestore.IsEnabled =
false;
            }

            private void SheetRestore_Click(object
sender, System.Windows.RoutedEventArgs e) =>
RestoreElements();

            private void SheetSearch_Click(object
sender, System.Windows.RoutedEventArgs e) =>
SearchForElement();

            private void Item_Selected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementSelected(sender);

                private void Item_Unselected(object
sender, System.Windows.RoutedEventArgs e) =>
ElementUnselected(sender);

                protected override void
RestoreElements()
                {
                    mainWindow.ViewsBoxSheet.Items.Clear();
                    foreach (ViewSheet sheet in sheets)
                    {
                        ListBoxItem item = new
ListBoxItem
                        {
                            Content = sheet.Title
                        };
                        item.Selected += Item_Selected;
                        item.Unselected +=
Item_Unselected;

                        mainWindow.ViewsBoxSheet.Items.Add(item);
                    }

                    mainWindow.SheetRestore.IsEnabled =
false;
                }

                protected override void
SearchForElement()
                {
                    string query =
mainWindow.tbSheet.Text.Trim().ToLower();
                    if (string.IsNullOrEmpty(query) &&
mainWindow.ViewsBoxSheet.Items.Count.Equals(she
ets.Count()))
                        return;

                    var collection =
mainWindow.ViewsBoxSheet.Items.SourceCollection
.OfType<ListBoxItem>()
.Where(box => (box.Content as
string).ToLower().Contains(query));
                    ListBoxItem[] updatedArray = new
ListBoxItem[collection.Count()];

                    collection.ToList().CopyTo(updatedArray);

                    mainWindow.ViewsBoxSheet.Items.Clear();
                    foreach (ListBoxItem listBoxItem in
updatedArray)
                    {
                        mainWindow.ViewsBoxSheet.Items.Add(listBoxItem)
;

                        mainWindow.SheetRestore.IsEnabled =
true;
                    }
                }
            }
        }
    }
}

```

```

        protected override void
ElementSelected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        ViewSheet activeSheet =
sheets.FirstOrDefault(sheet =>
sheet.Title.Equals(actualItem.Content));
        ActivatedElement = activeSheet.Id;

        mainWindow.JsonTab.IsEnabled =
true;

        mainWindow.NameTextSheet.Text =
activeSheet.Name;
        mainWindow.NameTextSheet.ToolTip =
new ToolTip { Content = activeSheet.Name };
        mainWindow.VieportsSheetText.Text =
activeSheet.GetAllViewports().Count.ToString();
        mainWindow.OriginTextSheet.Text =
string.Format("X: {0}, Y: {1}.",
activeSheet.Outline.Min.U,
activeSheet.Outline.Min.V);
        mainWindow.OriginTextSheet.ToolTip
= new ToolTip { Content = string.Format("X:
{0}, Y: {1}.", activeSheet.Outline.Min.U,
activeSheet.Outline.Min.V) };
        mainWindow.DocumentTextSheet.Text =
activeSheet.Document.Title;

mainWindow.DocumentTextSheet.ToolTip = new
ToolTip { Content = activeSheet.Document.Title
};
        mainWindow.NumberTextSheet.Text =
activeSheet.SheetNumber;
        mainWindow.ScaleTextSheet.Text =
Convert.ToString(activeSheet.Scale);
        mainWindow.CategoryTextSheet.Text =
activeSheet.Category.Name;
        mainWindow.JsonValidTextSheet.Text
= activeSheet.GetAllViewports().Select(port =>
RevitHelper.Document.GetElement(port) as
Viewport).Any(port =>
    {
        ViewPlan viewPlan =
RevitHelper.Document.GetElement(port.ViewId) as
ViewPlan;
        if (viewPlan != null)
            return
viewPlan.ViewType.Equals(ViewType.FloorPlan);

        return false;
    }).ToString());

mainWindow.AssemblyViewTextSheet.Text =
activeSheet.IsAssemblyView.ToString();
        mainWindow.TemplateTextSheet.Text =
activeSheet.IsTemplate.ToString();
        mainWindow.TransientTextSheet.Text
= activeSheet.IsTransient.ToString();

        mainWindow.UnitsTextSheet.Text =
"Feet";

application.ActiveUIDocument.ActiveView =
activeSheet;
    }

    protected override void
ElementUnselected(object sender)
    {
        ListBoxItem actualItem = sender as
ListBoxItem;
        ViewSheet activeSheet =
sheets.FirstOrDefault(sheet =>
sheet.Title.Equals(actualItem.Content));

        if
(ActivatedElement.Equals(activeSheet.Id))
        {
            ActivatedElement = default;

            mainWindow.JsonTab.IsEnabled =
false;

            mainWindow.NameTextSheet.Text =
string.Empty;

            mainWindow.NameTextSheet.ToolTip = null;

            mainWindow.VieportsSheetText.Text =
string.Empty;
            mainWindow.OriginTextSheet.Text
= string.Empty;

            mainWindow.OriginTextSheet.ToolTip = null;

            mainWindow.DocumentTextSheet.Text =
string.Empty;

            mainWindow.DocumentTextSheet.ToolTip = null;
            mainWindow.NumberTextSheet.Text
= string.Empty;
            mainWindow.ScaleTextSheet.Text
= string.Empty;

            mainWindow.CategoryTextSheet.Text =
string.Empty;

            mainWindow.JsonValidTextSheet.Text =
string.Empty;

            mainWindow.AssemblyViewTextSheet.Text =
string.Empty;

            mainWindow.TemplateTextSheet.Text =
string.Empty;

            mainWindow.TransientTextSheet.Text =
string.Empty;

```

```
        mainWindow.UnitsTextSheet.Text  
= string.Empty;  
    }  
}  
}
```

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 15.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 11%

| | | | | |
|--|----------|---------|-------------------------------------|--------------|
| ID: 97108 Название: Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень Добавлено в БД: 2021-11-23 Авторы: Р.Р. Гримак Руководители: О.М. Яшина Консультанты: Опоненты: | Документ | | Суммарное совпадение по Базе Данных | |
| | Символы | Лексемы | Символы | Лексемы |
| | 84360 | 616 | 15552 (18%) | 159 (26%) |

Источник плагиата

| ID | Описание | Наличие плагиата в документе | |
|-------|---|------------------------------|----------------|
| | | Символы | Лексемы |
| 95886 | Название: ЗВІТ з науково-дослідної практики Добавлено в БД: 2021-09-29 Авторы: Гримак Р.Р. Руководители: Скрипник Т.К. Консультанты: Опоненты: | 12927 (15.0%) | 134 (22.0%) |



Ім'я користувача:
Кафедра КН

ID перевірки:
1009320313

Дата перевірки:
23.11.2021 20:00:04 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
24.11.2021 09:30:19 EET

ID користувача:
100005671

Назва документа: Гримак_Кваліфікаційна_робота_магістра

Кількість сторінок: 84 Кількість слів: 11629 Кількість символів: 96239 Розмір файлу: 7.35 MB ID файлу: 1009344525

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

7.79%
Схожість

Найбільша схожість: 1.8% з джерелом з Бібліотеки (ID файлу: 1009334714)

4.5% Джерела з Інтернету 102 Сторінка 86

3.63% Джерела з Бібліотеки 73 Сторінка 87

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 22

Підозріле форматування 15 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень

Автор: Гримак Роман Русланович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент Яшина Оксана Миколаївна

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок | Позначка про відповідність |
|---|---|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту. | відповідає |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи | |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат. | |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту. | |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 15% та вони, схожість виявлена зі звітом автора з науково-дослідної практики.

2) за програмою UNICHECK виявлені 7.79%; найбільша схожість: 1.8% з джерелом з Бібліотеки (ID файлу: 1009334714) яке є посиланням на працю автора; інші схожості є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 15% і 7.79% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

О.М. Яшина

Гарант ОП

Р. О. Багрій

Завідувач кафедри КН

О. В. Бармак



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНМ-20-1 Гримак Романа Руслановича за темою: Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень.

1. Актуальність обраної теми

У сучасному виробництві корисне поширення одержали системи автоматизованого проектування, які надають можливість створювати технологічні процеси з меншими витратами часу та інженерних засобів. Система автоматизованого проектування це система, що здатна на автоматизованому рівні реалізувати інформаційну технологію виконання функцій проектування, це організаційно-технічний комплекс програмних інструментів, призначених для автоматизації процесу проектування, що складається з персоналу і групи технічних, програмних і інших засобів автоматизації його діяльності.

На сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення. Як правило, такі надбудови використовуються у функціональній інфраструктурі спеціалізованого комплексу рішень, що реалізують принцип інформаційного моделювання будівель (Building Information Modeling). Найпоширенішою системою такого типу є Autodesk Revit, тобто платформа, яка надає можливості тривимірного моделювання елементів будівлі і плоского креслення елементів оформлення, призначеного для архітекторів, конструкторів і інженерів-проектувальників

Таким чином, актуальність теми кваліфікаційної роботи магістра є достатньо обгрунтованою.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Об'єктом дослідження є процес пониження розмірностей векторного простору для архітектурно-будівельних елементів при переході між інтегрованими системами координат в умовах автоматизованого проектування. Предмет дослідження – моделі, алгоритми та засоби для побудови методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень.

Таким чином, тема кваліфікаційної роботи відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра.

3. Повнота розкриття мети та завдань дослідження

Мета кваліфікаційної роботи полягає в створенні методу збору, представлення, обробки, зберігання, передачі та доступу до інформації за рахунок пониження розмірностей векторного простору, що дозволить реалізувати алгоритм перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit.

Для досягнення поставленої мети визначені наступні задачі-дослідження:

- формулювання методики перетворення систем координат;
- реалізація фільтра для визначення об'єктів, що підлягають трансферу;
- визначення необхідного переліку функціональних параметрів стосовно відсортованих об'єктів, що підлягають трансферу;
- побудова інформаційної моделі методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень;
- реалізація інструментарію для хмарного зберігання даних;
- реалізація перетворення даних до універсального формату зберігання та подальшого трансферу;
- виконати алгоритмічну та програмну реалізацію методу пониження розмірностей векторного простору.

Завдання наукового дослідження повністю розкривають тему кваліфікаційної роботи магістра.

4. Наявність наукової новизни

В результаті проведеної кваліфікаційної роботи були отримані наступні результати:

– вдосконалено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом зменшення розмірностей векторного простору, що дозволило оптимізувати склад об'єктів, які підлягають трансферу між системами візуалізації. Відмінність від відомих підходів полягає у попередній фільтрації переліку об'єктів у поєднанні з переліком необхідних їх функціональних параметрів;

– вдосконалено методику моделювання систем при перетворенні систем координат, використання яких, на відміну від відомих, дозволило у явному вигляді представити координати об'єктів в інструментальній надбудові архітектурно-будівельної платформи Autodesk Revit;

– розроблено метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень. В результаті проведених досліджень було створено застосування відносно перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit. Надбудова дає можливість використовувати технології хмарного зберігання даних за допомогою програмного інструментарію, застосувати для збереження даних універсальний формат, зручний для розширеного та (або) віддаленого доступу.

5. Зміст кожного розділу роботи

Зміст розділу 1: характеристика предметної області та постановка задачі (аналіз предметної області, аналіз систем архітектурної візуалізації, аналіз хмарних сервісів для зберігання даних та постановка задачі).

Зміст розділу 2: проектування структури інформаційної системи (перетворення координат в системі візуалізації, автоматизація обробки інформаційних потоків та розробка структури інформаційної системи).

Зміст розділу 3: програмна реалізація (структура і функціональне призначення модулів системи та розробка програмних модулів).

Зміст розділу 4: апробація програмної інформаційної системи (основні функції надбудови, апробація методу на моделі приватної власності, апробація методу на моделі житлового комплексу та апробація методу на моделі шкільного закладу).

6. Ступінь розкриття теми роботи

В кваліфікаційній роботі магістра продемонстрована здатність збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до комп'ютерної системи, здатність формалізувати предметну область у вигляді відповідної інформаційної моделі та використовувати математичні методи для аналізу формалізованих моделей предметної області, здатність розробляти програмне забезпечення з обранням належних засобів для розробки та оцінювати та забезпечувати якість ІТ-проектів.

7. Якість оформлення кваліфікаційної роботи


Кваліфікаційна робота магістра виконана із дотриманням усіх вимог щодо оформлення. Викладення матеріалу є логічним, послідовним та стилістично грамотним.

8. Недоліки кваліфікаційної роботи

В кваліфікаційній роботі магістра не наведено шляхи подальшого вдосконалення та пропозиції щодо інтеграції результатів дослідження в інші програмні додатки.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Кваліфікаційна робота магістра Гримака Р.Р. рекомендується до захисту та заслуговує оцінки «відмінно».

Опонент _____  _____ к.т.н., доцент, доцент кафедри комп'ютерної інженерії та інформаційних систем Гнатчук Є.Г.



ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНм-20-1 Гримак Романа Руслановича за темою: Метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень.

1. Актуальність теми

Актуальність теми кваліфікаційної роботи магістра обґрунтована достатньо, оскільки на сучасному ринку автоматизованого виробництва більшість інженерів-конструкторів мають потребу в застосуванні додаткового програмного забезпечення.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

В кваліфікаційній роботі магістра продемонстрована здатність збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до комп'ютерної системи, здатність формалізувати предметну область у вигляді відповідної інформаційної моделі та використовувати математичні методи для аналізу формалізованих моделей предметної області, здатність розробляти програмне забезпечення з обранням належних засобів для розробки та оцінювати та забезпечувати якість ІТ-проектів.

Кваліфікаційна робота магістра містить сучасні моделі, методи, алгоритми, технології, процеси та способи отримання, представлення, обробки, аналізу, передачі, зберігання даних в інформаційних системах, що реалізовано у методі пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень.

Таким чином, тема кваліфікаційної роботи відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра.

3. Професійні та особистісні якості магістранта

Кваліфікаційна робота магістра Гримак Р.Р. є результатом роботи по вивченню методу пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень. Ним запропоновані і реалізовані основні ідеї, що дозволили отримати достатньо просту та завершену методологію пониження розмірностей векторного простору для прийняття контрольованих критично-безпечових рішень для певного кола об'єктів.

Під час роботи над кваліфікаційною роботою магістра Гримак Р.Р. виявив себе старанним, працьовитим, сумлінним, кваліфікованим спеціалістом здатним генерувати і реалізовувати нові наукові та інженерні ідеї.

За результатами виконання кваліфікаційної роботи магістрант отримав низку наступних професійних якостей: здатність збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до комп'ютерної системи, що розробляється, експлуатується чи супроводжується; здатність формалізувати предметну область певного проекту у вигляді відповідної інформаційної моделі; здатність використовувати математичні методи для аналізу формалізованих моделей предметної області; здатність розробляти, описувати, аналізувати та оптимізувати архітектурні рішення комп'ютерних систем різного призначення.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

В переважній більшості кваліфікаційна робота студентом-магістром була виконана самостійно. Студент сформулював інформаційну модель методу пониження розмірностей векторного простору та розробив архітектурно-будівельну надбудову, що доводить ефективність роботи вказаного методу.

Спільно із науковим керівником: сформульовано мету роботи та завдання дослідження; проведено обговорення отриманих результатів.

5. Наукова новизна та оригінальність запропонованих підходів

В результаті проведеної кваліфікаційної роботи були отримані наступні результати: – вдосконалено метод збору, представлення, обробки, зберігання, передачі та доступу до інформації шляхом зменшення розмірностей векторного простору, що дозволило оптимізувати склад об'єктів, які підлягають трансферу між системами візуалізації. Відмінність від відомих підходів полягає у попередній фільтрації переліку об'єктів у поєднанні з переліком необхідних їх функціональних параметрів; – вдосконалено методику моделювання систем при перетворенні систем координат, використання яких, на відміну від відомих, дозволило у явному вигляді представити координати об'єктів в інструментальній надбудові архітектурнобудівельної платформи Autodesk Revit.

Основні наукові та практичні результати опубліковані в наукових виданнях МОН України: – публікація на тему “Інформаційна технологія прийняття контрольованих критично-безпечових рішень перетворення параметрів моделей при трансфері між системами візуалізації” в науковому журналі “Вісник Хмельницького національного університету”. За темою дипломної роботи магістра автором виконано одну наукову публікацію.

6. Ступінь оволодіння методами дослідження

Ступінь оволодіння методами дослідження, що були використані в кваліфікаційній роботі магістра, є високим та достатнім.

7. Повнота та якість розкриття теми роботи

Тема кваліфікаційної роботи магістра розкрита на високому рівні. Для досягнення поставленої меті були виконані усі задачі дослідження

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Структура кваліфікаційної роботи оформлена добре та без явних зауважень. Завдання дослідження описані інформативно та в логічній послідовності.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Розроблено метод пониження розмірностей векторного простору для прийняття контрольованих критично-безпекових рішень. В результаті проведених досліджень було створено застосування відносно перетворення параметрів архітектурної моделі в структурі автоматизованого рішення Autodesk Revit. Надбудова дає можливість:

– використовувати технології хмарного зберігання даних за допомогою програмного інструментарію.

– застосувати для збереження даних універсальний формат, зручний для розширеного та (або) віддаленого доступу.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Кваліфікаційна робота магістра Гримака Р.Р. може бути допущена до захисту та заслуговує оцінки «відмінно».

Науковий керівник _____



к.т.н., доцент Яшина О.М.

