

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.422.833

«До захисту допущено»
Завідувач кафедри
О.В. Коваль
(підпис) (ініціали, прізвище)
“ ” _____ 2018р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Програмне забезпечення розподілених систем
на тему «Програмне забезпечення для взаємодії PDM та CAD систем»

Виконав (-ла): студент (-ка) 6 курсу, групи ТВ-71мп
Орел Дмитро Сергійович
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, доцент, к.е.н. Левченко Л.О.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент ст. викл., к.т.н. Баранюк О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В.
(прізвище, ініціали) (підпис)
« » 2018р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Орлу Дмитру Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Програмне забезпечення для взаємодії PDM та CAD систем.

Науковий керівник Левченко Лариса Олексіївна, к.е.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 5 листопада 2018 року №4072-с

2. Строк подання студентом дисертації 11 грудня 2018

3. Об'єкт дослідження процес інтеграції систем автоматизованого проектування та контролю виробів

4. Предмет дослідження програмне забезпечення організації взаємодії між PDM та CAD системами

5. Перелік питань, які потрібно розробити

1) проаналізувати сучасні CAD та PDM систем;

2) проаналізувати методи інтеграції між CAD та PDM системами;

3) удосконалити спосіб інтеграції PDM та CAD систем;

4) розробити програмне забезпечення для взаємодії PDM та CAD систем.

7. Орієнтований перелік публікацій _____

1) Орел Д.С. Програмне забезпечення для взаємодії PDM та CAD систем / Д.С. Орел, Л.О. Левченко // XVI міжнародна науково-практична конференція аспірантів, магістрантів, студентів «Сучасні проблеми наукового забезпечення енергетики», 24-27 квітня, 2018 року. – м. Київ, 2018. – С. 298.

2) Орел Д.С. Розробка програмного забезпечення для взаємодії PDM та CAD систем / Д.С. Орел // Сталый розвиток — XXI століття: управління, технології, моделі. Дискусії 2018: колективна монографія / Міненко М.А., Бендюг В.І., Комариста Б.М. [та ін.]; НТУУ «Київський політехнічний інститут імені Ігоря Сікорського»; Національний університет «Києво-Могилянська академія»; Вища економіко-гуманітарна школа / за наук. ред. проф. Хлобистова Є.В. — Київ, 2018. — 639 с.

7. Дата видачі завдання «11» вересня 2017р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз проблеми інтеграції PDM та CAD систем	11.09.2017-12.11.2017	
2	Аналіз існуючих реалізацій та шляхів вирішення проблеми	13.11.2017-15.01.2018	
3	Модифікування традиційного способу інтеграції систем автоматизованого проектування та систем контролю виробів	16.01.2018-18.08.2018	
4	Моделювання схеми роботи майбутньої програми	19.08.2018-30.08.2018	
5	Розробка архітектури програмного забезпечення	01.09.2018-20.09.2018	
6	Розробка програмного забезпечення	21.09.2018-10.10.2018	
7	Тестування програмного забезпечення	11.10.2018-30.11.2018	
8	Оформлення документації	01.12.2018-07.12.2018	

Студент

(підпис)

Орел Д.С.
(прізвище та ініціали)

Науковий керівник

(підпис)

Левченко Л.О.
(прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація складається зі вступу, шести розділів, висновку, переліку посилань з 51 найменування, 2 додатків, і містить 33 рисунки, 24 таблиці. Повний обсяг магістерської дисертації складає 119 сторінок, з яких перелік посилань займає 5 сторінок, додатки – 16 сторінок.

В умовах ринкової конкуренції керівництво компаній зацікавлене в максимально ефективному використанні нового програмного забезпечення і устаткування. При цьому замовлення все ускладнюються, а необхідний час реакції виробництва на потреби ринку скорочується. Випуск нового виробу на ринок напряду залежить від темпів його розробки.

Дисертаційна робота магістра виконувалась у НТУУ «КПІ ім. Ігоря Сікорського» у відповідності з планом наукових досліджень кафедри АПЕПС.

Метою дослідження є удосконалення існуючих способів інтегрування між CAD та PDM системами та розробкою програмного забезпечення на основі удосконаленого методу.

Для досягнення поставленої задачі були сформульовані наступні завдання дослідження, що визначили логіку дослідження та його структуру:

- забезпечення можливості виконання колективного проектування нових виробів із паралельним використанням різних CAD систем;
- уніфікація принципів та підходів при розробці нових виробів незалежно від середовища автоматизованого проектування (CAD);
- аналіз та удосконалення способів інтегрування PDM та CAD систем;
- здійснення програмної реалізації удосконаленого способу інтегрування між PDM та CAD системами.

Об'єктом дослідження є процес інтеграції систем автоматизованого проектування та контролю виробів.

Предметом дослідження є програмне забезпечення організації взаємодії між CAD та PDM системами.

Розв'язання поставлених задач виконувалося з використанням наступного підходу щодо уніфікації інтерфейсу взаємодії між PDM та CAD системами.

Наукова новизна одержаних результатів полягає в удосконаленні способу інтегрування між PDM та CAD системами шляхом уніфікації інтерфейсу взаємодії між ними, що забезпечує колективну розробку нового виробу.

Практична значимість дослідження визначається тим, що універсальний інтегруючий комплекс, розроблений на основі запропонованого підходу, значно прискорює впровадження на підприємстві декількох різномірних систем автоматизованого проектування та використання однієї системи контролю виробами.

Основні результати атестаційної роботи обговорювалися на:

— XVI міжнародна науково-практична конференція аспірантів, магістрантів, студентів «Сучасні проблеми наукового забезпечення енергетики» (24-27 квітня 2018 року);

— V міжнародна науково-практична конференція «Сталий розвиток – XXI століття: управління, технології, моделі (наукові читання імені Ігоря Недіна)» (м. Київ, 23-24 жовтня 2018 року).

За матеріалами роботи опубліковано 2 наукові роботи.

Ключові слова: *ПРОЕКТУВАННЯ ТА КОНТРОЛЬ ВИГОТОВЛЕННЯ ВИРОБУ, CAD СИСТЕМИ, PDM СИСТЕМИ, ІНТЕГРАЦІЯ CAD ТА PDM СИСТЕМ.*

ЗМІСТ

Вступ.....	10
Розділ 1. Системи автоматизованого проектування та контролю виробів на промисловому підприємстві	12
1.1 Етапи розробки виробів на підприємстві	12
1.2 Характеристика автоматизованих систем, що беруть участь у ЖЦП на підприємстві.....	14
1.3 Призначення CAD та PDM систем.....	19
1.4 Аналіз проблем підходу до інтеграції PDM та CAD систем	23
1.5 Мета та завдання наукового дослідження	24
Висновки до розділу 1	25
Розділ 2. Методи інтеграції PDM та CAD систем.....	26
2.1 Підхід PLM до автоматизації систем зокрема PDM та CAD.....	26
2.2 Удосконалення підходу до взаємодії PDM та CAD систем	34
Висновки до розділу 2	36
Розділ 3. Засоби розробки програмного забезпечення для взаємодії PDM та CAD систем.....	37
3.1 Мова програмування Java.....	37
3.2. Мова програмування C#.....	42
3.3 Середовище розробки Visual Studio	45
3.4 Середовище розробки IntelliJ IDEA.....	47
3.5 XML	49
3.6 SolidWorks API.....	50
3.7 J-LINK API	53
Висновки до розділу 3	56
Розділ 4. Реалізація програмного забезпечення для взаємодії PDM та CAD систем.....	58

<u>4.1 Розширення для PDM системи IT-Enterprise.....</u>	<u>58</u>
<u>4.2 Розширення для CAD систем PTC Creo та SolidWorks</u>	<u>63</u>
<u>4.3 Функціональна структура системи.....</u>	<u>66</u>
<u>Висновки до розділу 4</u>	<u>67</u>
<u>Розділ 5. Методи роботи користувача з програмним забезпеченням для</u> <u>взаємодії PDM та CAD систем</u>	<u>68</u>
<u>5.1 Підключення програмного забезпечення для взаємодії PDM та CAD</u> <u>систем</u>	<u>68</u>
<u>5.2 Інструкція роботи користувача з програмним забезпеченням для</u> <u>взаємодії PDM та CAD систем.....</u>	<u>73</u>
<u>Висновки до розділу 5</u>	<u>81</u>
<u>Розділ 6. Розроблення стартап проекту.....</u>	<u>82</u>
<u>6.1 Опис ідеї проекту</u>	<u>82</u>
<u>6.2 Технологічний аудит ідеї проекту.....</u>	<u>84</u>
<u>6.3 Аналіз ринкових можливостей запуску стартап-проекту.....</u>	<u>85</u>
<u>6.4 Аналіз ринкової стратегії проекту.....</u>	<u>91</u>
<u>6.5 Розроблення маркетингової програми стартап-проекту</u>	<u>94</u>
<u>Висновки до розділу 6</u>	<u>97</u>
<u>Висновки</u>	<u>98</u>
<u>Список Використаних джерел.....</u>	<u>99</u>
<u>Додаток А.....</u>	<u>104</u>
<u>Додаток Б</u>	<u>118</u>

ВСТУП

В умовах ринкової конкуренції керівництво компаній зацікавлене в максимально ефективному використанні нового програмного забезпечення і устаткування. При цьому замовлення все ускладнюються, а необхідний час реакції виробництва на потреби ринку скорочується. Випуск нового виробу на ринок напряму залежить від темпів його розробки [1].

Реалії сьогодення вимагають від підприємств постійно відслідковувати за тенденціями ринку, як найшвидше реагувати на будь-які зміни у вимогах щодо якості та ціни продукції, що випускається, як з боку споживачів так і з боку конкурентів. Постійно шукати способи зменшення собівартості власної продукції чи залучення додаткових потужностей для збільшення виробництва, а отже й збільшення власних прибутків.

Все більше і більше підприємств намагаються оптимізувати свою роботу. У наш час це зазвичай відбувається шляхом впровадження автоматизованих систем у виробничі процеси. Автоматизація здійснюється шляхом встановлення нового сучасного обладнання, використанні новітніх технологій. Автоматизовані системи слугують для виконання різних спеціалізованих задач, пов'язаних з роботою працівника, наприклад, проектування виробу чи збереження даних в базу даних та управління нею тощо.

Розробка будь-якого виробу включає в себе багато етапів. Одним з таких є етап проектування технічного об'єкту. В реальних задачах на підприємстві проектування виконується колективно. Різні підрозділи конструкторів виконують проектування своїх частин виробу за допомогою автоматизованих систем проектування CAD. Кожна CAD система має свій напрям використання, тому зазвичай на підприємстві застосовуються CAD системи для різних цілей проектування.

Життєдіяльність будь-якого промислового підприємства передбачає наявність величезної кількості конструкторської та технологічної документації, внутрішніх

документів, регламентів та інших даних стосовно розроблюваних виробів. На кожну одиницю продукції, яка проектується в CAD, створюються десятки документів. Починаючи від розробки специфікації і закінчуючи випуском виробу, усі документи розробляються.

Саме PDM системи управління даними про вироби на підприємстві.

У роботі створено програмне забезпечення для взаємодії PDM та CAD систем, яке здійснює інтеграцію CAD систем різного призначення та PDM системи. Для прикладу наведено результати інтеграції CAD систем SolidWorks та PTC Creo Parametric з PDM системою IT-Enterprise.

Дане програмне забезпечення дозволяє на промисловому підприємстві здійснювати автоматизоване проектування та контроль документації на виготовлення виробу та запуску його в серійне виробництво.

РОЗДІЛ 1. СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ТА КОНТРОЛЮ ВИРОБІВ НА ПРОМИСЛОВОМУ ПІДПРИЄМСТВІ

Головною метою роботи промислових підприємств є розробка та випуск на ринок нових виробів. Для цього використовується низка різних систем, кожна з яких має свою вузьку спеціалізацію. Одними з таких систем є CAD та PDM системи. Стрімка динаміка сучасного життя у ринковій системі породжує нові дослідницькі проблеми, активізує методологічні пошуки, формує нові парадигми дослідження технологічних процесів. Серед таких проблем виділяється проблема інтеграції PDM та CAD систем.

1.1 Етапи розробки виробів на підприємстві

Життєвий цикл продукту (ЖЦП) – це набір загально визначених етапів у житті комерційних продуктів. Визначаються наступні стадії, протягом певного періоду життя яких проходить кожний продукт: маркетингові дослідження, проектування, технологічна підготовка виробництва (ТПП), виробництво, експлуатація та утилізація [2].

На кожній стадії є свої цілі та задачі, від ефективності виконання яких залежить зниження собівартості розроблюваного товару, зниження часу на проектування та розробку, підвищення якості використання покупцями тощо.

Розглянемо деякі з основних етапів життєвого циклу для розробки виробів на рисунку 1.1.

Кожен продукт починається з ідеї. У деяких випадках вона може бути досить простою, на основі подібного, що вже існує, а в інших - це може бути дещо революційним та унікальним. Підприємство може мати багато ідей для нового

продукту, але, коли воно обирає найкращі з них, наступним кроком буде початок дослідження ринку. Це дає їм можливість з'ясувати, чи існує потреба у цьому виді

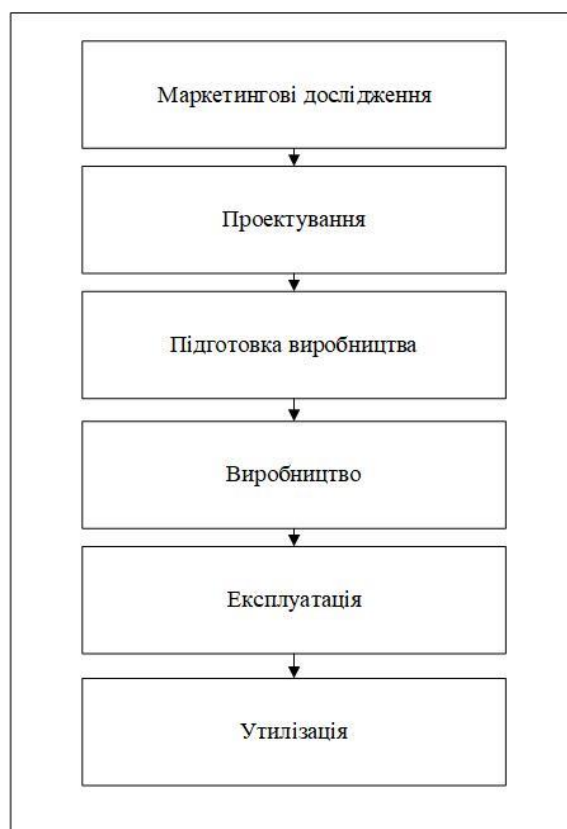


Рисунок 1.1 – Життєвий цикл розробки виробу

продукції, а також конкретні особливості, необхідні для розробки, щоб максимально відповідати потребам цього потенційного ринку. Таким чином, виконується маркетингове дослідження ринку на попит та пропозицію.

Етап проектування здійснює різні операції: формування різноманітних рішень, створення моделей і креслень, здійснення математичних розрахунків, моделювання процесів, поліпшення розробок. На ньому вирішуються технологічні та економічні питання, пов'язані з випуском нових виробів, де члени інженерно-конструкторської групи отримують уточнення специфікацій, розробляють концепт (вигляд) для виробів. При проектуванні приймаються рішення, якими саме властивостями повинен володіти даний товар для забезпечення необхідних потреб потенційних покупців. Проектування будь-якого технічного об'єкту здійснюється в два етапи: зовнішнє проектування та внутрішнє проектування. В процесі зовнішнього

проектування вирішуються функціональні та структурні питання системи в цілому. Внутрішнє проектування пов'язане з розробкою елементів (компонентів, підсистем) системи як фізичних одиниць обладнання, коли визначені цілі, створена модель об'єкта, обрані основні критерії ефективності та оптимальна стратегія управління. Закінчується цей етап випробуванням пробних зразків або партії виробів.

Технологічна підготовка виробництва (ТПВ) починається з поетапного планування термінів закінчення робіт. План оформлюється у вигляді загального графіка підготовки виробництва, який після затвердження стає єдиним вихідним документом для розгортання підготовки виробництва по кожному етапу та кожному виробу. Будь-які зміни як по номенклатурі виробів, так і за термінами закінчення етапів підготовки виробництва знаходять своє відображення у графіку. Всі його екземпляри враховуються. Для підготовки безпосереднього виготовлення деталей здійснюється маршрутна та операційна технології розробки, що реалізуються в програмах для верстаків числового програмного керування (ЧПК).

Етап безпосереднього виробництва включає в себе календарне й оперативне планування, придбання комплектуючих, механічна та інші необхідні види обробки виробу, збирання результатів обробки та тестування.

На останніх етапах виконуються упаковка, транспортування, встановлення у споживача, експлуатація, обслуговування, ремонт та утилізація товару.

Для досягнення поставлених цілей на сучасних підприємствах використовуються автоматизовані системи (АС), призначені для створення, переробки та використання всієї необхідної інформації про властивості виробів і процесів, кожна з яких виконує свої специфічні задачі.

1.2 Характеристика автоматизованих систем, що беруть участь у ЖЦП на підприємстві

Сучасні умови, в яких знаходиться практично будь-яке виробництво, особливо машинобудування чи приладобудування, - це постійне і значне ускладнення технічних та інженерних проектів. В таких умовах конкурентоспроможними

виявляються підприємства, що мають налагоджені процеси проектування, виробництва; поставки і підтримки виробів та здатні швидко реагувати на нові запити ринку.

На даному етапі розвитку виробництва актуальність питання автоматизації діяльності виробничих підприємств у всіх галузях промисловості не викликає сумнівів. На ринку представлений широкий вибір рішень, що реалізують максимальну кількість необхідних функцій, і численні компанії пропонують впровадження своїх систем, навчання і технічну підтримку. В даний час, розроблені десятки типів систем, кожна з яких відповідає за автоматизацію певної області (наприклад, системи автоматизації проектування, або системи автоматизації обліку на підприємстві).

На багатьох етапах життєвого циклу розробки товарів, починаючи з підприємств, які постачають вихідні матеріали і компоненти, та закінчуючи реалізацією товарів, потрібні послуги системи, що може здійснювати управління потоком товарів і послуг, передбачатиме переміщення та зберігання сировини, інвентарю, що переробляється, а також готової продукції з пункту походження до точки споживання. Такою системою є система планування управління ланцюгами поставок SCM (Supply Chain Management) [3]. Ланцюг поставок визначається як сукупність етапів підвищення вартості товарів при її русі від компаній-постачальників до компаній-споживачів. Управління даним ланцюгом представляє собою переміщення матеріального потоку з мінімальними затратами. В основному SCM використовує методи, спрямовані на координування всіх частин ланцюга управління від постачання сировини до постачання та / або відновлення продукції. SCM дозволяють мінімізувати загальні витрати у зв'язку з існуючими конфліктами між партнерами з мережі [4].

На сьогодні багато компаній, які розробляють програмні засоби автоматизованих систем, направлені на налагодження систем електронного бізнесу (E-commerce) [5]. Електронний бізнес (іноді називається комерційною мережею) - це термін, який використовується для опису діяльності, пов'язаної з веденням бізнесу в Інтернеті. Це стосується відносин бізнесу між бізнесом, бізнес-споживачем та навіть

споживачами-споживачами транзакцій, пов'язаних з купівлею та продажем товарів і послуг, передачі коштів та навіть обміну ідеями. Електронний бізнес включає в себе такі функції, як маркетинг, виробництво, фінанси, продаж та переговори [6].

У загальній структурі управління промисловістю виділяють кілька ієрархічних рівнів, наведених на рис. 1.2.

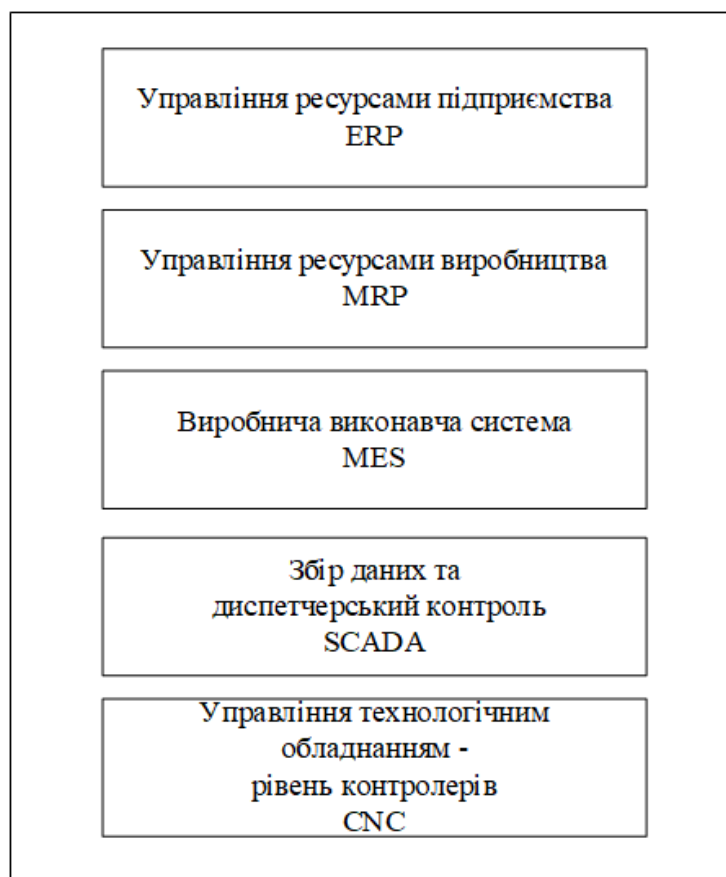


Рисунок 1.2 – Загальна структура управління промисловістю

Автоматизація управління підприємством здійснюється за допомогою автоматизованих систем управління (АСУ). Автоматизовані системи управління підприємством (АСУП) та автоматизовані системи управління технологічними процесами (АСУТП) - це загальний термін, що охоплює кілька типів систем керування та пов'язаних з ними інструментів, що використовуються для управління виробничими та технологічними процесами на підприємстві [7].

На першому рівні ієрархії виділяються АСУП для планування і управління підприємством ERP (Enterprise Resource Planning), яка використовується практично

на всіх етапах життєвого циклу [8]. Планування ресурсів підприємства (ERP) - інтегроване управління основними бізнес-процесами, часто в режимі реального часу з використанням відповідних програмного забезпечення та технологій. ERP зазвичай називають категорією програмного забезпечення для управління бізнесом, як правило, комплектом інтегрованих програм, які організація може використовувати для збирання, зберігання, управління та інтерпретації даних з цих численних бізнес-операцій [9].

Система ERP об'єднує різноманітні організаційні системи та полегшує безпомилкове здійснення транзакцій та виробництво, підвищуючи тим самим ефективність організації. ERP-системи працюють на різних комп'ютерних апаратних і мережних конфігураціях, зазвичай використовуючи базу даних як сховище інформації.

На другому рівні знаходяться системи планування ресурсами виробництва MRP II. Планування виробничих ресурсів визначається як спосіб ефективного планування всіх ресурсів виробничої компанії [10]. Планування виробничих ресурсів (MRP II) є інтегрованим методом операційного та фінансового планування для виробничих компаній. MRP II служить продовженням MRP (планування ресурсів виробництва замкнутого циклу) [11]. MRP II - це комп'ютерна система, яка може створювати мережні графіки покрокового виробництва з використанням даних у режимі реального часу. MRP II також використовується і як модуль більш масштабних систем управління ресурсами підприємства. Процес MRP II здійснюється синергетичною комбінацією комп'ютерних та людських ресурсів [12].

На третьому рівні використовуються MES системи. MES (Manufacturing Execution Systems) розглядається як проміжний крок між, з одного боку, системою планування ресурсів підприємства (ERP) та системою контролю та збору даних (SCADA) або системою керування процесом, з іншого. Система виробництва виробництва (MES) - це інформаційна система, яка об'єднує, контролює комплексні системи виробництва та потоки даних на підприємстві. Головною метою MES є забезпечення ефективного виконання виробничих операцій та покращення виробництва [13].

MES допомагає досягти цієї мети шляхом відстеження та збору точних даних у режимі реального часу про повний життєвий цикл виробництва, починаючи з випуску замовлення до етапу доставки готової продукції.

MES збирає дані про склад продукції, продуктивність, простежуваність, управління матеріалами, незавершену роботу та інші види діяльності підприємства. Ці дані, у свою чергу, дозволяють фахівцям у процесі прийняття рішень розуміти поточні параметри підприємства та краще удосконалити виробничий процес.

На четвертому рівні ієрархії здійснюється збір даних та диспетчерський контроль, який забезпечується за допомогою АСУТП SCADA (Supervisory Control and Data Acquisition) [14]. SCADA - це система програмних та апаратних елементів, що дозволяє промисловим організаціям здійснювати:

- контроль промислових процесів локально або в віддалених місцях;
- моніторинг, збирання та обробка даних у реальному часі;
- безпосередньо взаємодіяти з такими пристроями, як датчики, клапани, насоси, двигуни та інше, через програмне забезпечення Human-Machine Interface (HMI);
- запис інформації у журнал.

Системи SCADA є найбільш вживаними для промислових організацій, оскільки вони допомагають підтримувати ефективність контролю промислових процесів, що дозволяє зменшити час простою [15].

Програма SCADA обробляє та відображає дані, допомагаючи операторам та іншим працівникам аналізувати дані та приймати важливі рішення. Наприклад, система SCADA швидко повідомляє оператору, що партія продукту демонструє високий рівень помилок. Оператор призупиняє операцію та переглядає системні дані SCADA через HMI, щоб визначити причину виникнення проблеми. Оператор перевіряє дані та виявляє, що певна машина не працює. Можливість системи SCADA повідомляти оператора про проблему допомагає йому усунути цю проблему та запобігти подальшій втраті виробу [16].

На п'ятому рівні знаходяться системи CNC (Computer Numerical Control). Комп'ютерний числовий контроль (CNC) - це автоматизований контроль

інструментів обробки, а саме свердла, інструменти для свердління, токарні верстати за допомогою комп'ютера. Станок ЧПК створює виріб згідно специфікації, виконуючи запрограмовані інструкції [17].

ЧПК станки поєднують в собі моторизований маневровий інструмент і часто моторизовану маневрену платформу, які обидва контролюється ядром комп'ютера відповідно до запрограмованих команд. Команди передаються до ЧПК-станка у вигляді графічних файлів, які перетворені в послідовну програму інструкцій з автоматичного управління, а потім виконуються [18].

ЧПК є значним прогресом у механічній обробці. Отримані інструкції перетворюються (за допомогою "пост процесорного" програмного забезпечення) у конкретні команди, необхідні для комп'ютера для створення компонента, який завантажується в ЧПК [19].

Системи CNC отримують дані з систем автоматизованого проектування CAD (Computer-aided design), що використовуються на етапі проектування життєвого циклу розробки товарів. Для вирішення проблем збереження та управління даними про продукт, що проектується в CAD, розробляються системи, які отримали назву систем управління даними продукту PDM (Product Data Management). Системи PDM можуть бути як окремими, так і здійснювати інтеграцію з ERP системами.

1.3 Призначення CAD та PDM систем

Одним із засобів досягнення конкурентноспроможності в швидкій та якісній розробці виробу є впровадження засобів автоматизації проектування і виробництва. Завдання впровадження засобів автоматизації проектування у виробництві є особливо актуальним. Наприклад, в приладобудуванні через складність конструкцій приладів, наявності в них великої кількості різноманітних взаємодіючих деталей і елементів та особливо внаслідок високої частоти внесення змін до приладобудівних виробів виникає потреба у наявності систем автоматизованого проектування. До таких систем належать CAD.

CAD використовується для виконання попереднього дизайну та макетів, деталей дизайну та розрахунків, створення 3-D моделей, створення та випуску рисунків, а також взаємодії з аналізом, маркетингом, виробництвом та персоналом кінцевого користувача [20].

CAD сприяє виробничому процесу шляхом передачі детальної інформації про продукт у автоматизованій формі, що може бути універсально інтерпретований кваліфікованим персоналом. Він може бути використаний для отримання двовимірних або тривимірних діаграм. Використання інструментів CAD дозволяє переглядати об'єкт з будь-якого кута, навіть із зовнішнього вигляду. Одним з основних переваг моделювання з використанням CAD системи є можливість швидкого редагування у порівнянні з ручним методом. Крім детальної інженерії 2D або 3D моделей, CAD широко використовується - від концептуального дизайну та komponування продуктів до визначення виготовлення компонентів. CAD скорочує час проектування, підвищує точність моделювання виробу замість створення та тестування фізичних прототипів.

CAD широко використовується для виготовлення промислової продукції, анімаційних фільмів та інших програм. Спеціальний принтер або плоттер, як правило, потрібні для друку професійних дизайнерських оформлень. CAD-програми використовують векторну графіку або растрову графіку, що відображає зовнішній вигляд створюваного об'єкту.

Основними цілями CAD систем є [21]:

- скорочення трудомісткості проектування і планування;
- скорочення термінів проектування;
- скорочення собівартості проектування і виготовлення, зменшення витрат на експлуатацію;
- підвищення якості і техніко-економічного рівня результатів проектування;
- скорочення витрат на натурне моделювання та випробування.

Досягнення цих цілей забезпечується наступними методами:

- автоматизації оформлення документації;
- інформаційною підтримкою та автоматизацією процесу прийняття рішень;

- використанням технологій паралельного проектування;
- уніфікацією проектних рішень і процесів проектування;
- методом повторного використання проектних рішень, даних і напрацювань;
- методом стратегічного проектування;
- заміною натурних випробувань і методом макетування математичним моделюванням;
- підвищенням якості управління проектуванням;
- застосуванням методів варіантного проектування і оптимізації.

В реальних задачах на підприємстві проектування виробу виконується колективно. Різні підрозділи конструкторів виконують проектування своїх частин виробу засобами проектування, орієнтованими за конкретними напрямками. Тому для розробки одного продукту часто недостатньо однієї CAD системи і різні конструкторські підрозділи використовують різноманітні за призначенням CAD системи [22].

З метою забезпечення ефективного управління інформацією про створюваний виріб та вирішення проблем спільного функціонування компонентів CAD різного призначення використовуються PDM системи. PDM - це організаційно-технічні системи, що забезпечують управління всією інформацією про виріб. При цьому в якості виробів можуть розглядатися різні складні технічні об'єкти (кораблі і автомобілі, літаки і ракети, комп'ютерні мережі та ін.) [23].

Системи PDM фіксують та управляють інформацією про продукт, забезпечуючи користувачам інформацію про створюваний продукт протягом всього життєвого циклу розробки продукту. Вони підтримують електронний підпис файлів, контроль версій (реєстрація та перевірка файлів), управління версіями та статус випуску. Також для організації безпеки та адміністративної функціональності, захищення прав інтелектуальної власності застосовується управління ролями та відповідними привілеями доступу. Робочі процеси створення продукту дозволяють як внутрішнім учасникам розробки, так і зовнішнім партнерам брати участь у життєвому циклі продукту [24].

Переваги PDM:

- допомагають знайти необхідні дані учасникам підрозділів підприємства;
- поліпшують продуктивність та скорочують час розроблення;
- усувають помилки при розробленні продукту;
- покращують організацію ланцюга вартості;
- відповідають нормативним вимогам та розробленій документації;
- мінімізують операційні ресурси.

Впровадження PDM системи на підприємстві дозволяє створити єдине інтегроване середовище управління інженерними даними та проектами, яке забезпечує[25]:

- колективну роботу проектно-конструкторських підрозділів підприємства над проектами з розмежуванням прав доступу до його складових частин;
- управління інженерними даними, технічними даними, та різною інформацією, пов'язаною з виробом;
- надійне зберігання і швидкий пошук інформації в електронних архівах;
- максимальне використання в нових розробках відпрацьованих і перевірених технічних рішень, що зберігаються в базі даних (БД);
- уніфікація даних, які використовуються різними CAD системами.

Іншим актуальним питанням, що вирішують PDM системи, є одночасна робота з документом кількох відділів. Коли мова йде про технічну документацію, то до одного і того ж документу повинні мати доступ працівники відділу технологів та відділу конструкторів. За таких умов постійно виникає необхідність або копіювати документи, або узгоджувати, хто і коли буде працювати з документом. Це в свою чергу вимагає додаткових витрат часу та трудових і матеріальних ресурсів [26].

Як і для будь-якої справи, при розробці технічної документації дуже важливою є актуальність документу. Оскільки кожний варіант документу можуть підписати одні відповідальні особи і не завірити інші, виникає потреба версійності документів. Тобто може виникнути плутанина у виборі остаточного варіанту документу.

Будь-який підхід до керування професійними даними у виробництві вимагає інтеграції CAD та PDM систем. Технічно складні продукти не можуть бути оперативно оброблені та виведені на ринок своєчасно без PDM. Знання про вироби на підприємстві, в основному, містяться в його моделях CAD, а також у відповідних даних та документах, що мають контролюватися PDM системою [27].

Від ефективності і якості функціонування цих систем у виробництві, якості управління проектними роботами й інтеграції CAD в загальну архітектуру автоматизованої проектно-виробничого середовища багато в чому залежить ефективність самого виробництва і його здатність оперативно реагувати на вимоги ринку.

1.4 Аналіз проблем підходу до інтеграції PDM та CAD систем

До теперішнього часу інтегрування PDM та CAD систем здійснюється для кожного конкретного випадку за принципом «один на один»: конкретна PDM система і конкретна CAD система взаємодіють через API (Application Programming Interface - прикладні програмні інтерфейси обох систем). Принципова схема взаємодії представлена на рисунку 1.3.

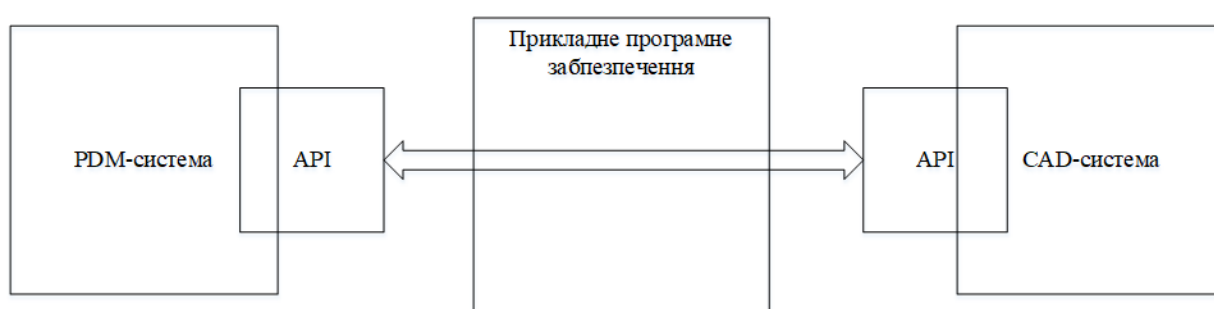


Рисунок 1.3 – Схема взаємодії між PDM та CAD системами

Основними перевагами цієї схеми є відносна простота та відносно низька вартість реалізації програмного забезпечення.

Даний спосіб взаємодії CAD та PDM систем має наступні недоліки [28]:

— обмежено застосування розробленого програмного забезпечення для інтеграції CAD системи і PDM системи, оскільки це рішення, як правило, специфічно для розв'язуваної задачі;

— для того, щоб замінити з будь-яких причин будь-яку з систем, необхідно повністю переробляти програмне забезпечення;

— у разі необхідності використання кількох систем одночасно необхідно істотно модифікувати програмне забезпечення;

— зміна API будь-якої з систем потребує за собою необхідність внесення змін у прикладне програмне забезпечення.

Таким чином, варіант інтегрування PDM системи та CAD системи через API безпосередньо, хоча і дозволяє створити ядро єдиного інформаційного простору підприємства, але робить його легко вразливим для будь-яких змін: від змін у функціях API до заміни систем.

Все вище сказане призводить до проблем використання сучасних інтеграцій систем автоматизованого проектування та систем управління даними про виріб.

1.5 Мета та завдання наукового дослідження

Основними завданнями дослідження інтеграції PDM та CAD систем є:

— забезпечення можливості виконання колективного проектування нових виробів із паралельним використанням різних CAD систем;

— уніфікація принципів та підходів при розробці нових виробів незалежно від середовища автоматизованого проектування (CAD);

— аналіз та удосконалення способів інтегрування PDM та CAD систем;

— здійснення програмної реалізації удосконаленого способу інтегрування між PDM та CAD системами.

Об'єктом дослідження є процес інтеграції систем автоматизованого проектування та контролю виробів.

Предметом дослідження є програмне забезпечення організації взаємодії між CAD та PDM системами.

Метою дослідження є розробка програмного забезпечення для взаємодії між CAD та PDM системами.

Висновки до розділу 1

Встановлено, що важливим фактором функціонування будь-якого виробничого підприємства є вдале поєднання різних автоматизованих систем, зокрема CAD та PDM. Від інтеграції PDM та CAD систем залежить якісний та швидкий контроль над розроблюваним виробом та його випуском у виробництво.

Аналіз проблем інтеграції PDM та CAD систем довів, що основною з проблем інтеграції таких систем є традиційний підхід «один до одного» за допомогою API. Тому для реалізації принципу «один до багатьох» було визначено перелік завдань щодо створення програмного забезпечення по взаємодії PDM та CAD систем. Таке програмне забезпечення надає можливості виконання колективного проектування нових виробів із паралельним використанням різних CAD систем та управлінні їх даними з PDM системи.

РОЗДІЛ 2. МЕТОДИ ІНТЕГРАЦІЇ PDM ТА CAD СИСТЕМ

На виробничих підприємствах використовується велика кількість систем для роботи з різного типу інформацією. Для того, щоб інтегрувати ці системи та створити так званий єдиний інформаційний простір використовуються загальноприйнятий підхід для автоматизації систем на підприємстві, а саме PLM (від англ. Product Lifecycle Management - управління життєвим циклом виробу). Саме цей підхід був обраний для удосконалення інтеграції PDM та CAD систем.

2.1 Підхід PLM до автоматизації систем зокрема PDM та CAD

До теперішнього часу розроблено досить велику кількість систем, що дозволяють автоматизувати роботу підприємства на різних ділянках виробництва: системи автоматизації проектування, системи зберігання та версійного контролю даних, автоматизованої технологічної підготовки виробництва, системи управління виробництвом та ін. При цьому основною проблемою є так звана «клаптева автоматизація» підприємств, тобто розрізненість систем автоматизації та їх несумісність.

В результаті такої «клаптевої автоматизації» підприємства отримують розрізнені прикладні системи, вартість інтеграції яких порівнянна із загальною вартістю комплексного рішення. Найчастіше відсутність єдиної стратегії розвитку автоматизованих систем призводить до створення незакінчених фрагментів інформаційної інфраструктури і систем, які не можуть ефективно застосовуватися на підприємстві. При цьому підприємство продовжує нести додаткові витрати на дублювання функцій, які могла б виконувати комплексна інформаційна система, і обслуговування трудомістких процедур обміну даними [29]. Цей так званий набір програм характеризується низькою доступністю інформації про підприємство: необхідна інформація є в базі даних, але вона часом суперечлива і недоступна безпосередньо для керівництва. Фахівці підрозділів можуть успішно користуватися

цими даними, але до керівництва вони доходять з затримками і у вигляді громіздких зведень, неприйнятних для оперативного аналізу та прийняття рішень.

На зміну цьому почали з'являтися технології організації єдиного інформаційного простору (ЄІП) підприємства з підходом PLM.

Підхід PLM полягає в тому, щоб забезпечити вирішення всіх завдань за допомогою набору взаємодоповнюючих програмних продуктів одного розробника програмного забезпечення. При такому підході користувач обмежений набором програмного забезпечення, що надається виробником, і залежить від розробника PLM-рішення.

За такою схемою розробники CAD-систем надають доступ до уже власно-створених PDM-систем, інтегрованих в CAD.

Найбільш розповсюдженими серед них є SolidWorks, PTC Creo, Catia, NX та інші.

SolidWorks – це сучасний програмний комплекс САПР, призначений для автоматизації робіт промислового підприємства на етапах конструкторської і технологічної підготовки підприємства. Він дозволяє проектувальникам створити математичну тверду модель об'єкту, яку можна зберегти в базі даних. SolidWorks є системою для проектування технічних об'єктів зі складною механікою, і використовує параметричний функціональний підхід, який спочатку розроблявся PTC (Creo / Pro-Engineer) для створення моделей та збірок. Параметри відносяться до обмежень, значення яких визначають форму чи геометрію моделі або складання. Параметри можуть бути числовими, такими як довжина рядка або діаметр кола, або геометричні параметри, такі як дотичні, паралельні, концентричні, горизонтальні або вертикальні тощо. Числові параметри можуть бути пов'язані один з одним через використання відносин. Побудова моделі в SolidWorks зазвичай починається з 2D чи 3D ескізу. Ескіз складається з геометрії, наприклад, точок, ліній, дуг, контурів (крім гіперболи) та сплайнів. Відносини використовуються для визначення таких атрибутів, як дотичність, паралельність, перпендикулярність і концентричність. Параметричний характер SolidWorks означає, що розміри та співвідношення керують геометрією, а не навпаки. Розміри в ескізі можна регулювати незалежно або

за взаємилн з іншими параметрами всередині або за межами ескізу [30]. Крім того, креслення нового виробу може бути створене або з деталей, або зі збірок. SolidWorks найбільш часто вживаний для проектування в сфері машинобудування та дозволяє перетворювати дані у різні формати.

PTC Creo - це система автоматизованого проектування (CAD), яка розроблена компанією PTC [31]. Така система підтримує розробку продукту, який виготовляється різними виробниками. CAD система складається з додатків, кожен з яких забезпечує чіткий набір можливостей щодо ролі користувача в процесі розробки продукту. Creo функціонує під управлінням системи Microsoft Windows і надає додатки для параметричної функції 3D CAD. Крім того, забезпечуються наступні функції: тверде моделювання, 3D-моделювання, 2D-орфографічні перегляди, аналіз та моделювання кінцевих елементів, схематичне оформлення, технічні ілюстрації, перегляд та візуалізація. Creo Elements / Pro та Creo Parametric конкурують безпосередньо з CATIA, Siemens NX / Solildge і SolidWorks. Набір додатків Creo замінює продукти PTC, раніше відомі як Pro / ENGINEER, CoCreate та ProductView. Creo має безліч різних рішень та функцій програмного забезпечення. Creo є частиною більш широкої системи розробки продуктів, розробленої PTC. Він з'єднується з іншими рішеннями PTC, що допомагають розробці продуктів, включаючи Mathcad для інженерних розрахунків та Arbortext для корпоративного програмного забезпечення. Він дозволяє групам конструкторів створювати, аналізувати, переглядати і максимально використовувати проекти виробів при подальшому конструюванні, використовуючи 2-х і 3-х вимірне моделювання САПР, параметричне та пряме моделювання. Використовується в машинобудуванні та аерокосмічній промисловості.

Catia – це багатоплатформенний набір програм для автоматизованого проектування (САПР), автоматизованого виробництва (CAM), автоматизованого інжинірингу (CAE), PLM та 3D, розроблений французькою компанією Dassault Systèmes. CATIA підтримує кілька етапів розробки продуктів (CAx), включаючи концептуалізацію, дизайн (CAD), інжиніринг (CAE) та виробництво (CAM), як правило, називається програмним комплексом для керування життєвим циклом

тривимірного продукту [32]. CATIA включає розробку поверхонь та форм, електричну, рідку і електронну систему проектування, машинобудування та системне проектування. CATIA дозволяє створювати 3D частини, від 2D ескізів, листових металів, композитів, формованих, підроблених або інструментальних деталей до визначення механічних зборів. Програмне забезпечення надає передові технології для механічного наплавлення. Вона надає інструменти для завершення визначення продукту, включаючи функціональні допуски, а також визначення кінематики. CATIA надає широкий спектр додатків для розробки інструментів, як для загальних інструментів, так і для прес-форм. У випадку аерокосмічної інженерії розроблений додатковий модуль, названий аерокосмічною листовою металевою конструкцією, який пропонує користувачу об'єднати можливості генеративного проектування листового металу та генеративного дизайну поверхні. CATIA пропонує рішення для формування дизайну, оформлення, наповнення робочого процесу та візуалізації для створення, модифікації та перевірки складних інноваційних форм промислового дизайну. CATIA підтримує кілька етапів дизайну продукту, починаючи з нуля або з 2D ескізів (креслення). CATIA спрощує розробку електронних, електричних та розподілених систем, до виробництва документації для виробництва.

NX - інтегроване рішення для спрощення і прискорення процесу розробки складних виробів, включаючи проектування електропроводки, деталей з листового металу, трубопроводів. NX включає в себе повний набір інструментів для розробки тривимірних моделей, складальних одиниць (вузлів), технічної документації та створення креслень. NX поєднує в собі каркасне, поверхнєве, параметричне, пряме і гібридне моделювання, що дає інженерам свободу у виборі підходу до проектування. Підтримуються технології проектування як «знизу-вгору», так і «зверху-вниз» з можливістю побудови наскрізних процесів розробки від вимог до виробу до етапу видачі даних для виробництва. Часто використовується при проектуванні деталей в енергомашинобудуванні [33].

Найбільш розповсюдженими PDM системами, які інтегруються з даними CAD, є SolidWorks PDM, WindChill, TeamCenter, IT-Enterprise та інші.

SolidWorks PDM від компанії Dassault Systemes дозволяє адмініструвати і синхронізувати інформацію про проект в межах всієї команди розробників виробу у вигляді єдиного і легко впроваджуваного рішення, тісно пов'язаного з усіма іншими додатками SolidWorks. Завдяки використанню захищеного сховища даних, можливо розширити доступ до середовища 3D-проектування і пов'язаним з ним файлів, як в локальному, так і у віддаленому режимі, для всіх учасників проекту, починаючи з проектування і закінчуючи виробництвом. Даний функціонал дозволяє всім учасникам проектів ділитися інформацією, співпрацювати з питань розробки, але при цьому забезпечує автоматичний захист вашої інтелектуальної власності за допомогою автоматизованих систем контролю версій і редакцій. SolidWorks PDM інтегрується з CAD системами того ж розробника, а саме з SolidWorks та Catia [34].

PDM система WindChill - програмний продукт для управління життєвим циклом продукту, пропонується PTC. WindChill надає індивідуальне робоче місце для роботи над виробом, в рамках якого користувач отримує доступ до всієї необхідної інформації, в залежності призначеної ролі, виконуваних завдань і прав доступу. Крім того, дані можуть надаватися користувачам залежно від етапу життєвого циклу і тих завдань, які виконуються на цьому етапі. WindChill пропонує наступні можливості:

- інтеграція з додатками MS Office;
- повнотекстовий пошук;
- автогенерація позначень для деталей і документів;
- контроль версій та ітерацій;
- перегляд структури виробів;
- управління обмеженнями застосування.

PDM система Teamcenter - пакет масштабованих програмних рішень для підтримки життєвого циклу виробів. Рішення Teamcenter призначені для інтенсифікації створення розробок, прискорення виведення виробів на ринок, забезпечення відповідності управлінським і законодавчим вимогам, оптимізації використання ресурсів підприємства і підтримки співпраці із суміжниками. Teamcenter можна використовувати для створення єдиної бази даних, процесів і

виробів, одержуваних з різних систем. Уповноважені співробітники отримують можливість використовувати цей ресурс для оперативного доступу до інформації, необхідної для виконання поставлених завдань. Система забезпечує спільну роботу у розподіленому середовищі: з її допомогою віддалені групи фахівців компанії встановлюють контакти, спілкуються і обмінюються інформацією в режимі реального часу. Завдяки наявності відкритого і функціонального інтерфейсу можна інтегрувати функції Teamcenter з уже наявними інформаційними системами і органічно вписатися в наявні процеси. Teamcenter заснований на гнучкій, чотирьохрівневій сервіс-орієнтованій архітектурі (SOA) і застосовується як у малому бізнесі, так і найбільшими світовими компаніями. На базі Teamcenter були розроблені спеціалізовані рішення, адаптовані для різних галузей - автомобільної, авіаційної, космічної та оборонної промисловості, високих технологій і електроніки, хімічної промисловості та фармацевтики, виробництва одягу та інших галузей. Основною CAD для інтеграції є NX.

Таким чином, маємо концептуальну схему інтеграції, за якою більшість розробників CAD систем постачають власні PDM. Дана схема наведена на рисунку 2.1.

У даного підходу є свої недоліки. З заміною CAD доведеться замінити і PDM. При розробці складного виробу можуть використовуватися декілька CAD-систем і збереження інформації в різних PDM-системах може призвести до того, що жодна з них не матиме даних про цілісний виріб [35]. Крім того, керівникам підприємств не вигідно купувати різні PDM системи у зв'язку з великою собівартістю. Ще однією проблемою є те, що документи та інженерні знання, які відносяться до створюваного виробу повинні бути доступними на всій території підприємства. За допомогою PDM працівники всіх підрозділів мають отримати доступ до будь-якої цієї інформації з відповідними дозволами, а налаштування доступу на різних PDM системах будуть затратувати велику кількість часу для кожного робітника. Перехід до однієї CAD-системи також практично неможливий, тому що кожна з них краще заточена під ту чи іншу форму розробки.

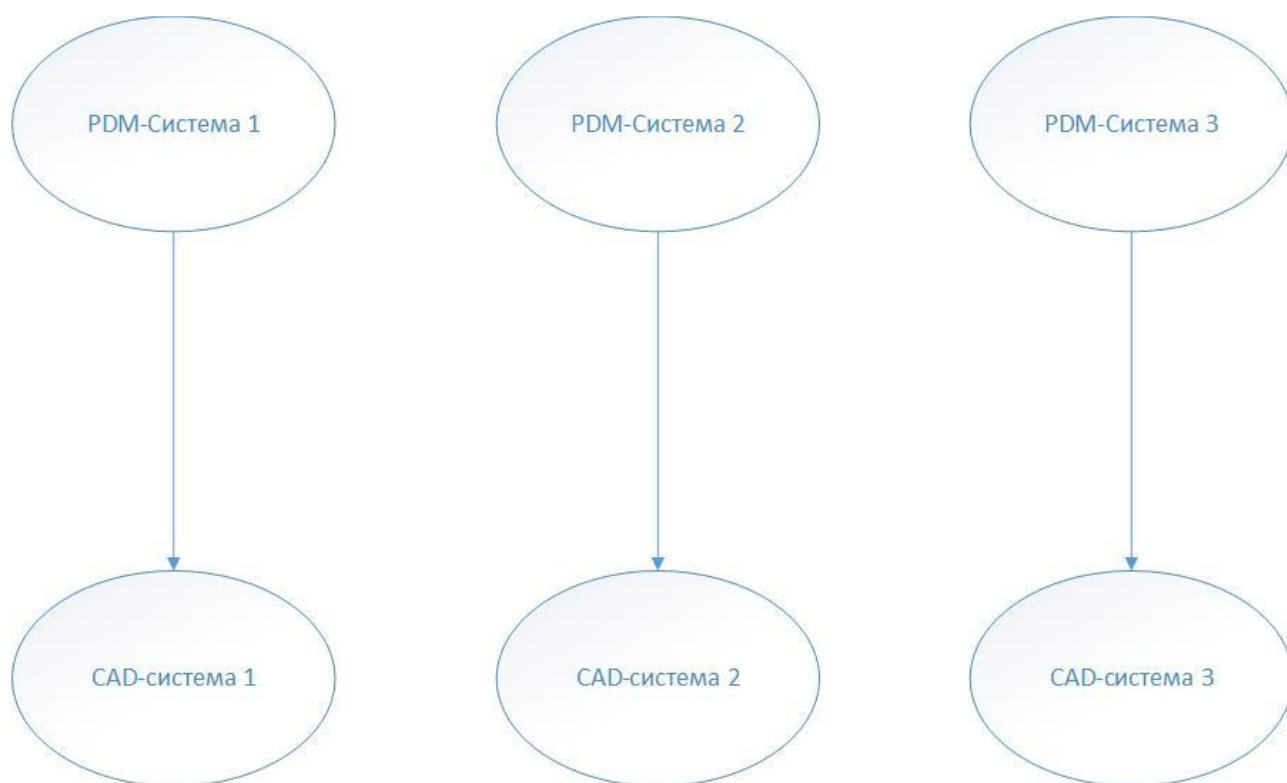


Рисунок 2.1– Загальна схема зв’язку CAD та PDM систем

IT-Enterprise - комплексна система управління забезпечує ефективне управління великим і середнім промисловим підприємством, групою підприємств, включає готові рішення для всіх відділів і служб підприємства.

IT-Enterprise- система класу ERP, MRPII, MES, APS, EAM, SCM, CRM, PDM. Система відповідає вимогам підприємств, сертифікованих по ISO 9001:2000 та вимогам законодавства України, враховує особливості ведення бізнесу в Україні.

Відповідно до класифікації APICS (American Production and Inventory Control Society) система IT-Enterprise повністю відповідає стандарту MRPII і концепції ERP.

Система «IT-Enterprise» функціонує в трьохрівневій архітектурі клієнт-сервер. Виділяються шари сервера бази даних, пулу серверів додатків (серверів додатків може бути скільки завгодно), клієнтського програмного забезпечення [36].

Адаптивність системи – більше 3000 параметрів налаштування системи, вбудовані генератори для кінцевого користувача – довільних запитів, табличних звітів, форм експорту інформації, документів звітності та аналізу [37].

В системі «IT-Enterprise» використовуються наступні технічні рішення і технології [38]:

- орієнтація на використання широких можливостей служб Windows Server 2008 R2;

- орієнтація і підтримка відкритих галузевих стандартів: PDF, HTML, XML;
- тісна інтеграція з продуктами Microsoft Office (Word, Excel, Outlook);
- інтеграція з безкоштовними офісними продуктами Open Office;
- широке використання технологій .NET Framework .
- ERP, MRP II - скорочення витрат, управління матеріальними;
- виробничими і фінансовими ресурсами, планування закупівель;
- виробництва і продажів, опис виробничих технологій і нормативів;
- CRM, SCM - управління взаємовідносинами з клієнтами, управління ланцюжками поставок;
- MES - внутрішньоцехове планування виробництва на рівні обладнання;
- APS - синхронне планування виробництва на рівні замовлень;
- ЕСКД, ЕСТД - формування конструкторської та технологічної документації;

- OLAP - багатовимірний бізнес-аналіз даних для управління;
- ISO 9001: 2000 - контроль якості в повному життєвому циклі продукції;
- процесний документообіг;
- WEB-сервіси, технології .NET та ASP.NET;
- інтеграція з MS Word, MS Excel, OpenOffice, Crystal Reports, MS Outlook;
- інтеграція з CAD, CAM, PDM. Інтеграція з АСУ ТП і обладнанням.

Модуль IT-Enterprise PDM вирішує багато проблем, пов'язаних з обліком і контролем електронного документообігу, полегшує пошук необхідних документів, якщо вони зареєстровані в системі. База даних модуля є гнучкою і налаштовується під вимоги конкретного підприємства.

В частині ведення архіву технічної документації модуль забезпечує:

- прийом електронних документів в архів (реєстрація, контроль, облік).
- Ведення картотек архівів;
- проведення змін в архівних документах;
 - пошук оригіналу в архівній базі даних;

- оповіщення користувачів про приміщенні в архів нового документа;
- облік видачі копій абонентам.

Однією з частин модулю являється класифікатор ресурсів. Класифікатор ресурсів в даному контексті – це всі доступні матеріальні ресурси, якими управляє підприємство та класифіковані за державними стандартами або стандартами НАТО або стандартам, що підтримує дане підприємство. Наприклад, ресурсом може бути звичайна шайба або цілий двигун, які відносяться до певних категорій розподілення на підприємстві.

Таким чином, необхідно інтегрувати декілька CAD систем з однією PDM-системою. Для цього необхідно розробити програмне забезпечення, яке дозволяє уніфіковано працювати з різними CAD-системами та інтегруватися з єдиною довільною PDM системою. Для реалізації цього програмного забезпечення було обрано інтегрувати CAD системи SolidWorks та PTC Creo, бо вони використовуються для різних типів проектування та PDM систему IT-Enterprise через її доступність та легкість у впровадженні.

2.2 Удосконалення підходу до взаємодії PDM та CAD систем

Виходячи із завдань, що були поставлені раніше, необхідно забезпечити можливість виконання колективного проектування нових виробів із паралельним використанням різних CAD систем. Для цього необхідно удосконалити принцип інтегрування між CAD та PDM таким чином, щоб розробка нових виробів була незалежною від середовища автоматизованого проектування. Це може бути досягнуто за рахунок уніфікації інтерфейсу взаємодії між PDM та CAD системами. Попропонується замість підходу «один на один» використовувати програмний інтерфейс для доступу до операцій PDM системи з CAD систем як наведено на рисунку 2.2.

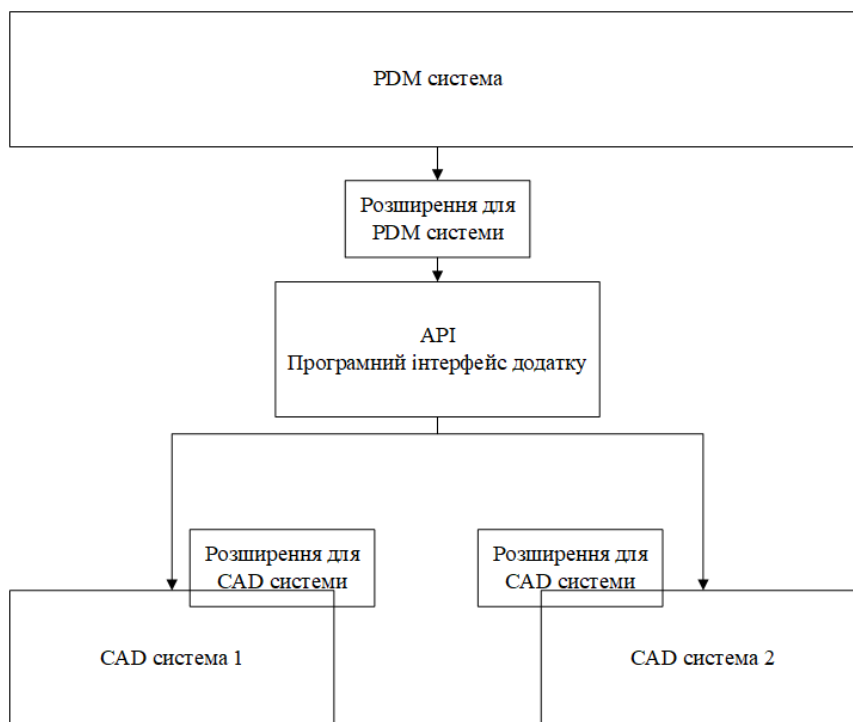


Рисунок 2.2 – Схема зв'язку CAD та PDM систем

Розширення для CAD систем слугують для виклику операцій, які визначені в програмному інтерфейсі. Розширення для PDM системи реалізує цей програмний інтерфейс і здійснює необхідні операції такі як завантаження моделі виробу, збереження моделі, що проектується в CAD до PDM системи тощо. Дане API може бути вдосконалене та доповнене у зв'язку з потребами користувача.

За допомогою запропонованого підходу всі дані, незалежно від того, є це моделі CAD, креслення, виготовлення та складання креслень, основні дані моделі, матеріали або документи із зображеннями, текстовими або розрахунковими таблицями, надійно зберігатимуться у сховищах даних системи PDM незалежно від типу CAD системи, з якої вони надходять. Це гарантує, що всі дані про продукт завжди оновлюватимуться та узгоджуватимуться між собою та між підрозділами.

Висновки до розділу 2

Розглянуто підхід PLM, який пропонується для інтеграції PDM та CAD систем. Для удосконалено підходу обрано інтегрувати CAD системи SolidWorks , PTC Creo та PDM систему IT-Enterprise.

Наукова новизна дослідження полягає у вдосконаленні традиційного підходу інтеграції PDM та CAD шляхом уніфікації інтерфейсу взаємодії, що забезпечує колективну розробку нового виробу.

Практична значимість дослідження визначається тим, що універсальний інтегруючий комплекс, розроблений на основі запропонованого підходу, значно прискорює впровадження нової CAD системи проектування без заміни існуючих або впровадження нових додаткових систем контролю виробів.

РОЗДІЛ 3. ЗАСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЗАЄМОДІЇ PDM ТА CAD СИСТЕМ

Програмне забезпечення для взаємодії складається з API для уніфікації операцій з боку PDM системи, розширень з боку CAD систем SolidWorks та PTC Creo та розширення з боку PDM системи IT-Enterprise, що реалізує вище зазначене API. Кожен з цих елементів написаний за допомогою різних технологій, кожна з яких базується на використаній платформі.

Для розробки розширення для CAD системи PTC Creo було використано мову програмування Java та J-LINK API.

Для розробки уніфікованого API операцій, розширень для CAD системи SolidWorks та PDM системи IT-Enterprise було обрано мову C#.

Для розробки розширення CAD системи SolidWorks використано також SolidWorks API.

Розробка розширення для PDM системи здійснювалася на платформі IT-Enterprise.

Розробка на мові Java середовищі IntelliJ IDEA, а на C# - Visual Studio.

Передача даних між PDM та CAD системами здійснюється за допомогою універсального формату XML.

Основні засоби розробки наведено нижче.

3.1 Мова програмування Java

Мова програмування Java - це загальноприйнята, об'єктно-орієнтована мова класу. Вона розроблена таким чином, щоб бути досить простою, щоб багато хто з програмістів міг досконало володіти мовою. Мова програмування Java пов'язана з C і C ++, але організована досить різноманітно, з рядом аспектів із C і C ++, що були опущені, а також кілька ідей з інших мов. Вона призначена бути виробничою

мовою, а не мовою дослідження, і тому, дизайн уникнув у тому числі нові та неперевірені функції [39].

Мова програмування Java статично типізована. Тому вона чітко виділяє помилки часу компіляції, які можуть і повинні бути виявлені під час компіляції, і ті, які трапляються під час виконання. Час компіляції зазвичай складається з перекладу програм в автономне байтове представлення коду. Заходи під час виконання включають завантаження та зв'язування класів, необхідних для виконання програми, необов'язкове створення машинного коду та динамічна оптимізація програми, а також фактичне виконання програми. Мова програмування Java є відносно мовою високого рівня, тому машинні представлення не доступні через мову. Це включає управління автоматичним сховищем, зазвичай за допомогою збирача сміття, щоб уникнути проблеми безпеки явного вилучення (як в C або C++) [40]. Високопродуктивні збирачі сміття можуть мати обмежені паузи для підтримки системного програмування та застосування в режимі реального часу. Мова не включає будь-які небезпечні конструкції, такі як масив-доступ без перевірки індексу, оскільки такі небезпечні конструкції призведуть до того, що програма поводитиметься неузгоджено.

Мова програмування Java, як правило, складається з інструкцією байт-коду і двійкового формату, що визначені в специфікації Java Virtual Machine [41].

На початку 90-х років Java, яка спочатку йшла під назвою Oak, а потім Грін, була створена командою під керівництвом компанії Sun Microsystems, яка зараз належить Oracle. Java була спочатку розроблена для використання на цифрових мобільних пристроях, таких як мобільні телефони. Проте, коли перша версія Java був випущена широкому загалу в 1996 році, основна увага була перенесена на використання в Інтернеті, забезпечуючи інтерактивність користувачів, надаючи розробникам можливість створювати анімовані веб-сторінки.

Сама мова Java дуже проста. Тим не менш, Java поставляється з бібліотекою класів, які надають найпоширеніші функції утиліт, без яких неможливо обійтися в більшості програм Java. Ця бібліотека класів, яка називається API Java, належить до Java як до самої мови. Мова Java складається лише з 50 ключових слів, але Java API

має кілька тисяч класів - з десятками тисяч методів, які можна використовувати у ваших програмах.

Java можна використовувати для розробки різних типів програм:

- автономні програми;
- аплети;
- веб-додатки;
- розподілені програми.

Автономний додаток - це програма, що працює на вашому комп'ютері. Він більш-менш схожий на програму C або C ++.

Аплет - це програма, призначена для передачі даних через Інтернет та для виконання на клієнтській машині веб-браузером, сумісним з Java, як Internet Explorer або Netscape. Аплети також є програмами Java, але вони знаходяться на серверах. Аплет не може бути виконаний як автономна програма. Аплет може бути виконаний тільки шляхом вставки його в HTML-сторінку, як зображення або звуковий файл. Для запуску аплету вам потрібно отримати доступ до HTML-сторінки, в якій вбудований аплет. Коли веб-браузер завантажує таку HTML-сторінку, вона згодом завантажує виконуваний файл, який містить Applet, а потім виконує його на локальному комп'ютері.

Веб-додатки запускаються на веб-сервері. Веб-застосунки доступні через веб-клієнти, тобто веб-переглядачі, такі як Internet Explorer або Netscape. Кожного разу, коли отримується доступ до деякого веб-сайту, вказавши URL-адресу (Universal Resource Locator), ви отримуєте доступ до деякої веб-програми. Основні компоненти веб-додатків, написані на Java, є Java Servlets Pages (JSP), і HTML.

Java Servlets також є програмами Java, які запускаються на сервері, а потім надсилають результат / відповідь клієнту. Сторінки JSP можна розглядати як комбінацію HTML та Java-коду. Веб-сервер перетворює сторінки JSP в Java Servlets перед виконанням.

Щоб отримати доступ до веб-додатків, треба вказати URL-адресу. Якщо URL-адреса відповідає HTML-сторінці, веб-сервер просто повертає сторінку HTML клієнту, яка потім відображає її. Якщо URL-адреса відповідає сервлет або JSP, то

вона виконується на сервері, а результат / відповідь повертається клієнту, який потім відображається клієнтом.

Розподілені додатки. Додаток Java можна розділити на декілька модулів / компонентів (Java-програм), які можуть працювати на різних машинах. Програми / додатки Java, що працюють на різних машинах, можуть взаємодіяти один з одним. Точніше, об'єкти Java на одній машині можуть викликати методи на об'єкти Java, що працюють на іншій машині. Таким чином, Java підтримує розподілену обробку в самому мові.

Особливості Java:

1) проста та потужна: якщо користувач вже розуміє основні поняття об'єктно-орієнтованого програмування, вивчення Java буде набагато простішим, оскільки Java успадковує синтаксис C / C ++ та багато об'єктно-орієнтованих функцій C ++, тому ми можемо сказати що Java була розроблена так, що її легко навчитися та використовувати. Java надає небагато чітких способів досягнення заданого завдання. На відміну від інших систем програмування, вони забезпечують десятки складних способів виконання простих завдань.

2) безпека. Використовуючи Java-сумісний браузер, кожен може безпечно завантажувати аплети Java без страху перед вірусною інфекцією чи зловмисним наміром через принцип його розробки. Таким чином, кожен може завантажувати аплети з упевненістю, що ніякої шкоди не буде зроблено, а безпека не буде порушена. Java забезпечує такий захист, обмежуючи програму Java у середовищі виконання Java, і робить її недоступною для інших частин комп'ютера.

3) портативна: багато типів комп'ютерів та операційних систем використовуються у всьому світі, і багато з них підключені до Інтернету. Java дає можливість гарантувати, що будь-який результат на одному комп'ютері з Java може бути відтворено на іншому. Тому код, запущений на різних платформах, має той самий результат.

4) об'єктно-орієнтована: Java підтримує всі особливості об'єктно-орієнтованої мови програмування, такі як абстракція, інкапсуляція, успадкування, поліморфізм та динамічне зв'язування тощо. Таким чином, за допомогою цих функцій користувач

може зменшити складність програми, що розвивається в Java. Java дала чистий, корисний, реалістичний підхід до об'єктів, тому ми можемо сказати, що об'єктна модель в Java проста і проста у розширенні.

5) надійна: більшість програм, що використовуються сьогодні, не можуть бути виконані з декількох причин. Першою є помилки в управлінні пам'яттю. Наприклад, в C / C ++ програміст повинен вручну виділити і виключити всю динамічну пам'ять. Це іноді призводить до проблем, тому що програмісти або забудуть вільну пам'ять, яка раніше була виділена, або іноді намагаються врятувати деяку пам'ять, яку ще використовує частина їх коду. Java практично усуває ці проблеми, керуючи розподілом пам'яті (за допомогою нового оператора) та видаленням. (Видалення повністю автоматичне, оскільки Java надає сміттєві збірки для невикористаних об'єктів.) Другою - невиправдані виняткові умови. За допомогою обробки виключень програміст може легко обробляти помилку або виняток, щоб користувач міг запобігти програмі, автоматично зупинивши виконання, коли знайдено виняток. Таким чином, при розробці Java була надана висока пріоритетність у створенні надійних програм.

6) багатопотокова: Java підтримує програмування, що дозволяє користувачеві писати програми, які виконують багато функцій одночасно. Дві або більше частини програми можуть працювати одночасно, тоді кожна частина такої програми називається Thread, і цей тип програмування називається багатопоточним програмуванням. Кожна нитка визначає окремий шлях виконання. Таким чином, багатопоточність - це спеціалізована форма багатозадачності.

7) нейтральна архітектура: дизайнери Java працювали над досягненням своєї мети "написати один раз; працювати в будь-якому місці, в будь-який час, назавжди", і в результаті була розроблена віртуальна машина Java. Java є нейтральною для архітектури, вона створює байтовий код, що нагадує машинний код, і не є специфічним для будь-якого процесора.

8) інтерпретована та висока продуктивність: вихідний код спочатку компілює і генерує код у проміжне представлення, яке називається байтовим кодом Java, який є високо оптимізованим набором коду інструкцій. Цей код можна інтерпретувати на

будь-якій системі, яка має віртуальну машину Java і генерує машинний код. Код байтів Java був ретельно розроблений, використовуючи компілятор, щойно у своєму розпорядженні часу, так що його можна легко перекласти в нативний машинний код для дуже високої продуктивності. Більшість попередніх крос-платформних рішень запускаються за рахунок продуктивності.

9) розподілена: Java дозволяє об'єкту отримати доступ до інформації через мережу за допомогою RMI (Remote Method Invocation) - це дозволені об'єкти на двох різних комп'ютерах для виконання процедур віддалено. Таким чином ця функція підтримує програмування клієнта / сервера.

10) динамічна: Java-програми містять значні обсяги інформації про тип часу, які використовуються для перевірки та вирішення доступу до об'єктів під час виконання. Це дає змогу динамічно зв'язати код безпечним та досконалим способом.

Протягом багатьох років Java розвивається як успішна мова для використання в Інтернеті та і не тільки. За даними Oracle, по всьому світу більше 9 мільйонів розробників Java, а на Java запускаються більше 3 мільярдів мобільних телефонів.

3.2. Мова програмування C#

C# - це загальна мова об'єктно-орієнтованого програмування (ООР) для мереж та веб-розробок. C# вказується як загальна мова інфраструктури (CLI) [42].

У січні 1999 року голландський інженер-програміст Андерс Хейлсберг створив команду для розробки C# як доповнення до системи Microsoft's NET. Спочатку C# було розроблено як об'єктно-орієнтовану мову C-Like. Фактичне ім'я було змінено, щоб уникнути потенційних проблем із товарними знаками. У січні 2000 року NET був випущений як C#. Його СУБД сприяє розвитку кількох веб-технологій. Цей термін іноді пишеться як C Sharp або C-Sharp.

C# покращено та оновлено багато функцій C і C++, включаючи наступне:

— C# має суворе значення типу булевих даних, наприклад `bool`, тоді як типи змінних `bool` C++ можуть бути повернені як цілі числа або покажчики, щоб уникнути спільних помилок програмування;

— C# автоматично управляє недоступною пам'яттю об'єктів за допомогою збирача сміття, що усуває проблеми розробника та витоку пам'яті;

— Тип C# безпечніший, ніж C++, і має безпечні конверсії лише за умовчанням (наприклад, розширення цілого числа), які реалізуються під час компіляції або виконання.

Жодна неявна конверсія між булевими, членами переліку та цілими числами (крім 0) може бути перетворена в перерахований тип. Визначені користувачем конверсії повинні бути вказані як явні або неявні, порівняно з операторами неявного конвертування за замовчуванням C++ та конструкторами копіювання.

C# призначений для роботи з платформою Microsoft .Net. Мета Microsoft - полегшити обмін інформацією та послугами через Інтернет, а також дозволити розробникам створювати надзвичайно портативні програми. C# спрощує програмування завдяки використанню розширюваної мови розмітки (XML) та протоколу простий доступ до об'єктів (SOAP), який дозволяє отримати доступ до об'єкта програмування або методу, не вимагаючи від програміста запису додаткового коду для кожного кроку. Оскільки програмісти можуть будувати існуючий код, а не багаторазово дублювати його, C#, як очікується, зробить його більш швидким та менш дорогим, щоб продавати нові продукти та послуги.

Microsoft співпрацює з ECMA, органом міжнародних стандартів, для створення стандарту для C#.

Програми C# запускаються на платформі .NET Framework, що є невід'ємною частиною Windows, яка включає в себе віртуальну систему виконання, яка називається загальною мовою виконання (CLR) та уніфікованим набором бібліотек класів [43]. CLR - це комерційна реалізація корпорацією Microsoft спільної мовної інфраструктури (CLI), міжнародного стандарту, що є основою для створення середовища виконання та розробки, в якій мови та бібліотеки працюють разом.

Вихідний код, написаний в C #, складається на проміжну мову (IL), що відповідає специфікації CLI. Код IL та ресурси, такі як растрові зображення та рядки, зберігаються на диску у виконуваному файлі, який називається збіркою,

зазвичай з розширенням .exe або .dll. Збірка містить маніфест, який надає інформацію про типи, версії, культуру та вимоги безпеки асамблеї[44].

Коли програма C # виконується, збірка завантажується в CLR, яка може виконувати різні дії на основі інформації в маніфесті. Тоді, якщо вимоги до безпеки відповідають вимогам, CLR виконує компіляцію в режимі реального часу (JIT), щоб перетворити IL-код на власні інструкції машини. CLR також надає інші послуги, пов'язані з автоматичною збіркою сміття, обробкою виключень та управління ресурсами. Код, який виконує CLR, іноді називають "керованим кодом", на відміну від "некерованого коду", який складається в нативній машинній мові, яка націлена на конкретну систему. Наступна діаграма ілюструє співвідношення часів і часу виконання файлів вихідного коду C #, бібліотек класів .NET Framework, збірок та CLR.

Мова взаємодії є ключовою особливістю .NET Framework. Оскільки IL-код, вироблений компанією C #, відповідає загальним специфікаціям типу (CTS), код IL, створений з C #, може взаємодіяти з кодом, який був згенерований з версій .NET для Visual Basic, Visual C ++ або будь-якого з більш ніж 20 інших CTS-сумісних мови. Один збічник може містити кілька модулів, написаних на різних мовах. NET, і типи можуть посилатися один на одного, так, якби вони були написані однією мовою[45].

Окрім служб часу, .NET Framework також включає в себе велику бібліотеку понад 4000 класів, організовану в просторах імен, які надають широкий спектр корисних функцій для всього, починаючи від введення та виводу файлів до обробки рядків до синтаксичного аналізу, до елементів управління Windows Forms . Типова програма C # використовує бібліотеку класів .NET Framework, яка широко використовується для обробки звичайних "сантехнічних" завдань.

Перехід на .NET платформу пропонує багато варіантів розробки. Деякі з переваг, що пропонує C#:

- сучасна мова програмування високого рівня;
- тип безпеки;
- об'єктно-орієнтований;
- спрощений синтаксис та прості асинхронні схеми;

- xml-коментарі;
- велика кількість платформ для створення: мобільні пристрої, веб, хмарні технології, настільні комп'ютери, ігри, IoT тощо;
- автоматичне керування пам'яттю;
- знайомий синтаксис для розробників Java і C-стилі;
- існує можливість користуватися API, що написане на іншій мові, що підтримує .NET.

3.3 Середовище розробки Visual Studio

Вбудоване середовище розробки Visual Studio - це панель запуску, яка використовується для редагування, налагодження та створення коду, а потім публікації додатку. Інтегроване середовище розробки (IDE) - це багатофункціональна програма, яка може використовуватися для багатьох аспектів розробки програмного забезпечення. Понад стандартний редактор і налагоджувач, який забезпечують більшість IDE, Visual Studio включає в себе компілятори, інструменти для завершення коду, графічні розробники та багато інших функцій для полегшення процесу розробки програмного забезпечення[46].

Visual Studio використовується для запису власного коду та керованого коду, підтримуваного Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Редактор коду Visual Studio .NET підтримує IntelliSense та рефакторинг коду, тоді як вбудований налагоджувач Visual Studio .NET підтримує як вихідну, так і налагоджену на рівні машини. Visual Studio .NET включає в себе інші вбудовані інструменти, такі як конструктор форм, що корисно при побудові графічних інтерфейсів; веб-дизайнер, який створює динамічні веб-сторінки; дизайнер класів, який використовується для створення власних бібліотек, і дизайнер схем підтримки баз даних.

Для покращення функціональності на багатьох рівнях можуть бути додані плагіни для підтримки системи керування джерелами, а редактори мови та

візуальних дизайнерів для домену є прикладами нових наборів інструментів, які можуть бути додані за допомогою цих плагінів[47].

Вбудовані мовні послуги пропонують підтримку різних відомих мов (наприклад, C#, C++, VB та F#), а мовні сервіси можуть бути інстальовані окремо для підтримки мов, включаючи M, Python і Ruby, серед інших підтримуваних мов.

Існують версії Visual Studio, які підтримують індивідуальні мови, які є економічно ефективним рішенням для початківців, яким потрібен тільки Visual Basic, або досвідченим користувачам, які працюють над швидкими додатками обробки, і постійно потребують візуальних C# або візуальних C++, але не зацікавлені в низькій ефективності, швидкості Visual Basic.

Наведемо деякі основні вікна Visual Studio[48]:

- вікно Solution дозволяє переглядати, переміщатись та керувати кодами файлів. Solution Explorer може допомогти організувати код, об'єднавши файли в рішення та проекти;

- вікно редактора, де можна редагувати код або спроектувати користувацький інтерфейс, наприклад вікно з кнопками та текстовими полями;

- вікно Output - це місце, на якій Visual Studio надсилає сповіщення, такі як налагодження та повідомлення про помилки, попередження компілятора, публікації повідомлень про статус тощо. Кожне джерело повідомлень має свою вкладку;

- вікно Team Explorer дозволяє відстежувати робочі елементи та спільно використовувати код з іншими користувачами, використовуючи технології керування версіями, такі як Git та Team Version Version Control (TFVC).

Деякі популярні функції Visual Studio, які допомагають бути продуктивнішими під час розробки програмного забезпечення, включають:

- рефакторинг включає в себе такі операції, як інтелектуальне перейменування змінних, витягування одного чи декількох рядків коду в новий метод, зміна порядку параметрів методу тощо;

- IntelliSense - це термін для набору функцій, який відображає інформацію про код безпосередньо в редакторі та, у деяких випадках, для вас пише невеликі коди. Це схоже на наявність базової документації в редакторі, яка заощаджує вас від

необхідності шукати інформацію про тип в іншому місці. Функції IntelliSense залежать від мови. Щоб отримати додаткові відомості, див. C # IntelliSense, Visual C++ IntelliSense, JavaScript IntelliSense та Visual Basic IntelliSense;

— швидкий запуск. Visual Studio може здаватися надзвичайно часом з таким численним меню, параметрами та властивостями. Вікно швидкого запуску - це відмінний спосіб швидко знайти те, що потрібно в Visual Studio. Коли починається вводиться шукана назва того, Visual Studio списує результати, які виконують саме ту інформацію, куди потрібно. Якщо потрібно додати функціональність до Visual Studio, наприклад, для того, щоб додати підтримку для додаткової мови програмування, Quick Launch пропонує результати, які відкривають Installer для встановлення робочого навантаження або окремих компонентів;

— squiggles і швидкі дії. Squiggles - хвилясті підкреслення, які попереджають про помилки або можливі проблеми у коді при введенні. Ці візуальні підказки дозволяють швидко виправляти неполадки, не чекаючи виявлення помилки при складанні або запуску програми. Якщо навести курсор на шпигун, ви побачите додаткову інформацію про помилку. Лампочка може також з'явитися в лівій частині поля за допомогою дій, які називаються "Швидкі дії", щоб виправити помилку;

— виклик ієрархії. Вікно ієрархії викликів показує методи виклику вибраного методу. Це може бути корисною інформацією, коли треба змінити або видалити метод, або відстежити помилку;

— codeLens допомагає знаходити посилання на код та зміни у ньому, помилки, робочі елементи, огляди коду та тести модулів, все не виходячи з редактора.

3.4 Середовище розробки IntelliJ IDEA

IntelliJ IDEA - це спеціальне середовище програмування або інтегроване середовище розробки (IDE), яке в основному призначене для Java. Це середовище використовується особливо для розробки програм. Він розроблений компанією JetBrains, яка офіційно називалася IntelliJ. Він доступний у двох виданнях:

Community Edition, який має ліцензію Apache 2.0, та комерційне видання Ultimate Edition. Обидва вони можуть бути використані для створення програмного забезпечення, яке можна продати. Те, що робить IntelliJ IDEA таким, що відрізняється від аналогів, - це простота використання, гнучкість та надійний дизайн.

IntelliJ IDEA була розроблена JetBrains, раніше відома як IntelliJ. Вона була вперше випущена в 2001 році, і похвалилася такими функціями, як вдосконалена кодова навігація та можливість редагувати коди, що зробило її дуже популярним. Середовище розробки з відкритим кодом для Android, випущене компанією Google у 2014 році, також базується на IntelliJ IDEA. IDE підтримує багато інших мов програмування, таких як Python, Lua і Scala.

Найбільша причина, що розглядається як один з найкращих інструментів програмування на базі Java, - це його допоміжні функції, які полегшують використання та створюють дуже добре розроблені програми. Вона також має розширені функції перевірки помилок, що дозволяє швидше і простіше перевіряти помилки.

IntelliJ IDEA поставляється з величезним вибором інтегрованих інструментів, які роблять розробку програм більш продуктивними та точнішими. Вона також має розширений редактор, що надає користувачам інструмент для плавного та бездоганного кодування для роботи з HTML та CSS.

IntelliJ IDEA включає в себе:

- інтелектуальний редактор коду, який має всі смахи для розуміння коду Java, XML та Groovy;
- рефакторинг, перевірка кодів та наміри, надшвидка навігація та пошук;
- інтеграція тестових рамок: JUnit і TestNG;
- інструменти для створення інструментів: Ant і Maven;
- популярні версії систем управління інтеграцією: CVS, Subversion і Git;
- Swing дизайнер інтерфейсу.

3.5 XML

XML - це розширення файлу для формату файлу Extensible Markup Language (XML), що використовується для створення загальних форм інформації та обміну форматом та даними у Всесвітній павутині, у внутрішніх мережах та інших країнах, використовуючи стандартний текст ASCII[48].

XML подібний до HTML. Обидва XML та HTML містять символи розмітки для опису вмісту сторінки або файлу. HTML, однак, описує вміст веб-сторінки (головним чином текстових та графічних зображень) лише в тому, як його відображати та взаємодіяти. Наприклад, буква "p" поміщається в теги розмітки, починається новий абзац тощо[49].

XML описує зміст з точки зору того, які дані описуються. Наприклад, слово "phonenum", розміщене в тегах розбиття, може вказувати на те, що дані, що були за ними, були номер телефону. XML-файл може оброблятися виключно як дані програми, або його можна зберігати з аналогічними даними на іншому комп'ютері або відображати, як файл HTML. Наприклад, залежно від того, як програма на приймальному комп'ютері хотіла б обробляти номер телефону, вона може бути збережена, відображена або набрана.

XML вважається розширюваним, оскільки, на відміну від HTML, символи розмітки необмежені та самостійно визначаються. Очікується, що HTML та XML будуть використовуватися разом у багатьох веб-додатках. Розмітка XML, наприклад, може з'явитися на сторінці HTML.

Деякі мови відстежують своє коріння окремим особам, які намагаються вирішувати малі чи локалізовані проблеми. XML був зібраний командою, яка працювала над розробкою продукту з широкомасштабним застосуванням. Проблема, яку вони вирішували - забезпечення доступу до Інтернету.

Команда не працювала з нуля. SGML забезпечив більшу частину фундаменту. Розвиток розпочато в 1996 році, а робоча копія була випущена протягом п'яти місяців. XML 1.0 був схвалений W3 в 1998 році. Версія 1.1 була використана в 2004 році. XML підтримується W3C.

3.6 SolidWorks API

SolidWorks API (Application Programming Interface) - це інтерфейс, що дозволяє розробляти власні додатки для системи SolidWorks. API - інтерфейс містить безліч функцій, які можна викликати з програм Microsoft Visual Basic, Microsoft Visual C ++, Microsoft Visual Studio або з файлів-макросів SolidWorks. Ці функції надають програмісту прямий доступ до функціональних можливостей SolidWorks[51].

За допомогою API - додатків можна вирішувати безліч різних завдань, наприклад такі як: інтеграція SolidWorks з іншими програмними пакетами, розробка спеціалізованих модулів, що додають до базових можливостей SolidWorks додаткову функціональність і різні інші завдання. API - додатки дозволяють отримати безліч конфігурацій однієї деталі або збірки, тим самим виграти величезну кількість часу при прийнятті конструкторських рішень.

Розробка API - додатки може здійснюватися на рівні створення макросу в SolidWorks, або на рівні окремого додатка, написаного на мові C # або VisualBasic. Всі динамічні бібліотеки, необхідні для роботи з API - додатками автоматично встановлюються разом з SolidWorks.

Використання SolidWorks API обумовлено тим, що це найбільш дешевий і зручний спосіб гнучко налаштувати інформаційну систему на вирішення завдань конкретного підприємства. Завдяки цьому, в даний момент SolidWorks є однією з найпопулярніших систем проектування в усьому світі.

Найбільш важливими областями застосування інтерфейсу прикладного програмування SolidWorks є:

- інтеграція SolidWorks з різними Windows-додатками;
- розробка модулів, що додають до базових можливостей SolidWorks додатковий функціонал і т.п.

Інтерфейс прикладного програмування поставляється в складі базової конфігурації САПР SolidWorks. Варто відзначити, що крім базового конструкторського рішення API є у всіх основних модулів, що входять в пакет

SolidWorks, до яких насамперед належать: Toolbox, FeatureWorks, Utilities, PhotoWorks, eDrawings, Routing та інші. Динамічні бібліотеки типів і констант, що відповідають за роботу API автоматично встановлюються на комп'ютер при установці програми. Таким чином, кожне робоче місце САПР SolidWorks за замовчуванням оснащений інтерфейсом прикладного програмування, що дає розробникам широке поле для діяльності.

Абсолютно всі динамічні бібліотеки, які відповідають за роботу API, за замовчуванням включені в дистрибутив SolidWorks і автоматично встановлюються на комп'ютер при установці програми. Використання API - найбільш дешевий і зручний спосіб ідеально налаштувати SolidWorks на вирішення завдань саме вашого підприємства.

У довідковій системі SolidWorks API можна знайти докладний опис структури API, а також список всіх нових функцій, доданих в API до моменту виходу останньої версії SolidWorks.

Програмний інтерфейс SolidWorks використовує об'єктно-орієнтовану модель. Всі функції SolidWorks API - це методи або властивості, які застосовуються до об'єктів SolidWorks. Діаграма об'єктів SolidWorks є ієрархічну структуру, яка відображає зв'язки між об'єктами SolidWorks і показує, в який спосіб можна отримати доступ до того чи іншого об'єкту SolidWorks API.

API містить сотні функцій, які можна викликати за допомогою VBA, VB.NET, Visual C #, Visual C ++ 6.0 і Visual C ++ / CLI. Ці функції забезпечують прямий доступ до функціональності SolidWorks, такий як створення лінії, вставка існуючої частини в документ частини або підтвердження параметрів поверхні. На рисунку 3.1 показано схему API програмного комплексу Solidworks.

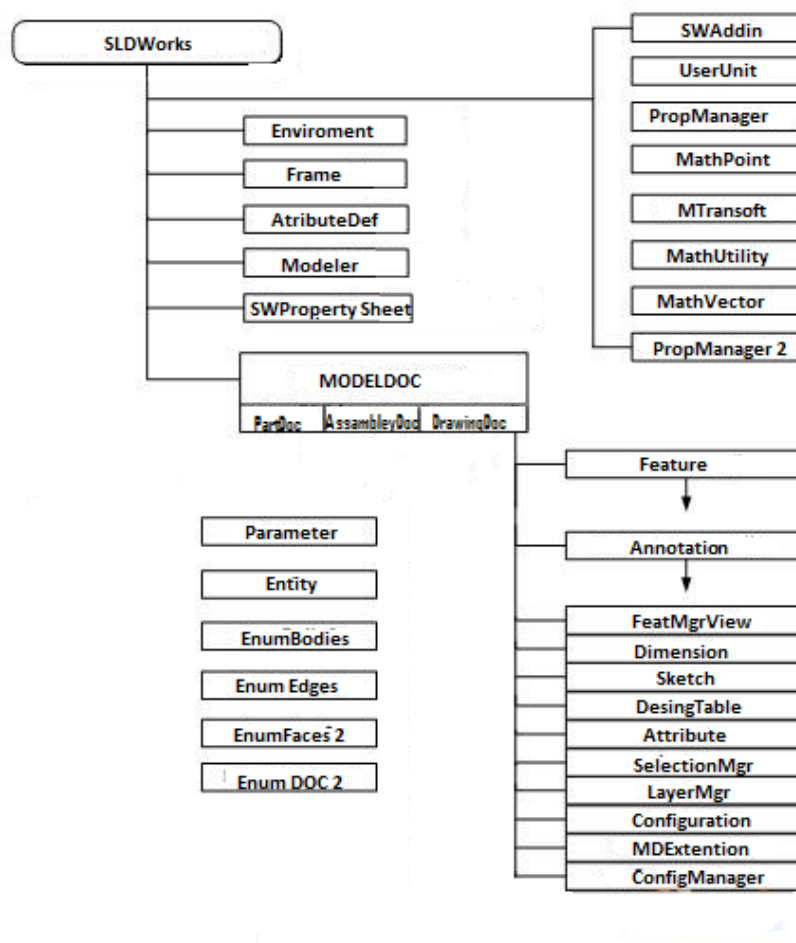


Рисунок 3.1 – Схема API Solidworks

Кожен інтерфейс, доступний в SolidWorks API, включаючи його пов'язані властивості і методи, зареєстрований; недокументовані інтерфейси SolidWorks API і функції не підтримуються і не повинні використовуватися. SolidWorks рекомендує, щоб використовували останню версію API.

Використання SolidWorks API припускає, що користувач знайомий з VBA, VB.NET, Visual C #, Visual C ++ 6.0 або Visual C ++ / CLI.

SolidWorks не документує свій API з повною діаграмою ієрархії класів, як відомі MFC і.NET / WinFX діаграми ієрархії класів. Причина цього полягає в тому, що MFC і.NET покладаються в великій мірі на: класи, спадкування впровадження, відбір між класами, використовуючи статичні кидки (C ++).

Ці типи бібліотек класів і API можуть часто бути корисні з діаграми ієрархії. Однак, SolidWorks - заснований на COM API, який використовує:

— інтерфейси;

- інтерфейсне спадкування;
- стандартні методи, щоб повернути інтерфейси на існуючих і нових об'єктах.

Взаємодія між інтерфейсами через: `QueryInterface` (C++), який повертає покажчик на вказаний інтерфейс на об'єкті, на який клієнт в даний час тримає інтерфейсний покажчик. У той час як графічна діаграма ієрархії часто вже не передає стільки ж інформації в цьому стилі API, є кілька місць, де це може бути корисно.

Взаємодія з призначеним для користувача інтерфейсом здійснюється завдяки `Feature Interfaces`, `User-interface` і `Drawing interface` які дозволяє підлаштувати меню програми і розташування всіх основних інструментів в зручному для користувача розташуванні.

Комунікація забезпечується за допомогою інструментарію `SolidWorks Toolbox` який являє собою бібліотеку стандартних компонентів. `Toolbox` розроблений спеціально щоб дозволити стороннім додатки взаємодія з `SolidWorks`.

3.7 J-LINK API

J-Link - це програмний інтерфейс (API) для CAD-системи `PTC Creo Parametric`. Він надає тільки частину функціональності, доступної в більш складному `Creo / TOOLKIT`, заснованому на C. Проте доступні виклики досить для автоматизації багатьох задач, що виникають при роботі в CAD-системах.

Той факт, що J-Link API, заснований на Java, дає розробнику можливість застосовувати велику колекцію бібліотек та фреймворків. Код, розроблений з використанням J-Link, в більшості випадків легше підтримувати в порівнянні з кодом у `Creo / TOOLKIT`. Додатковою перевагою J-Link API є те, що він не вимагає окремої ліцензії.

Синхронний додаток J-Link запускається та управляється `PTC Creo`. Контроль належить або `PTC Creo Parametric`, або програмі, але не обом одночасно.

Асинхронний додаток запускається незалежно від PTC Creo Parametric зможливістю запуску або підключення до PTC Creo Parametric процесів.

Можна запускати синхронні програми J-Link як окремі додатки або моделі, специфічні програми. Більшість необхідних параметрів для цих двох програм є незалежно від самих програм. Це дає змогу конвертувати прикладну програму до типової програми або навпаки.

Для розуміння як програма підключається за допомогою API J-LINK введемо такі поняття, які знаходяться в таблиці 3.1.

Таблиця 3.1. Таблиця понять API J-LINK

Поняття	Опис
<CREO_LOADPOINT>	Місце, де встановлені програми Creo.
<DATECODE>	Дата-код CREO, що доповнює поточну версію програми
<JLINK_LOADPOINT>	Місце, де встановлено J-Link: <CREO_LOADPOINT> \ Загальні файли \ <DATECODE> \ jlink
<JLINK_APP_TEXT>	Каталог ресурсів програми J-Link

Бібліотека для роботи з J-LINK API знаходиться на комп'ютері за адресою <CREO_LOADPOINT> / <DATE CODE> /text/java/pfc.jar (або otk.jar з версії PTC Creo 4.0).

J-Link застосунок Java має забезпечити клас двома загальнодоступними статичними методами:

- метод запуску: це параметр Creo для запуску програми.
- метод зупинки: викликається додатком Creo, щоб зупинити додаток.

Обидва методи повинні міститись у одному загальнодоступному класі за допомогою такої декларації публічний статичний xxxx (), де xxxx - це назва методу.

Додаток J-Link повинен забезпечувати каталог, що містить набори текстових ресурсів, зображення тощо. Цей каталог позначено як <JLINK_APP_TEXT>.

Текстові файли повинні знаходитись під пунктом <JLINK_APP_TEXT>, а зображення повинні знаходитись під <JLINK_APP_TEXT> \ resource.

Набір текстових ресурсів полягає в текстовому файлі, який визначає ключі та значення, як показано нижче на рисунку 3.3.

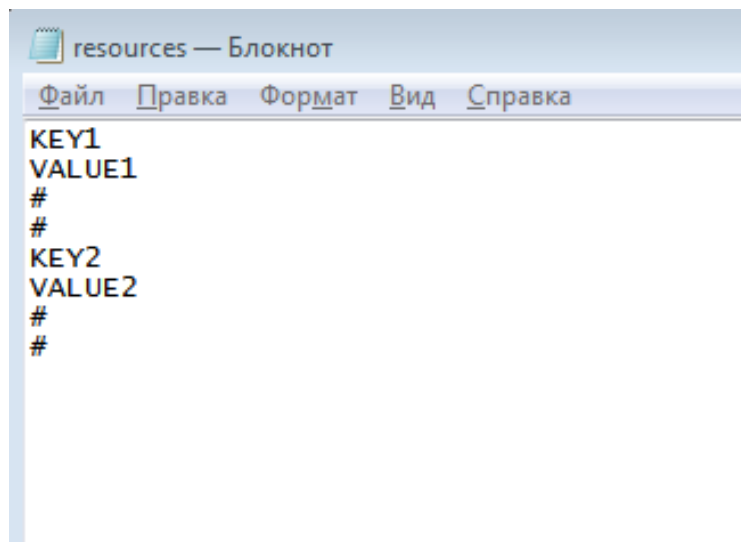


Рисунок 3.3 – Схема заповнення текстового файлу ресурсів для програмного додатку з API J-LINK

Компоненти текстових ресурсів використовуються API J-Link для перетворення тексту.

Для того, щоб бути завантаженою програмою Creo, програма з J-Link повинна бути описана в файлі реєстру, це простий текст, де кожен рядок складається з попередньо визначеного ключа та значення. Зазвичай цей файл має назву protk.dat і знаходиться в робочому каталозі Creo (налаштовується в самій програмі Creo). Структура файлу описана в таблиці 3.2.

Таблиця 3.2. Таблиця структури файлу реєстрації додатку для PTC Creo

Заздалегіть визначений ключ	Опис
name	Назва програми
startup	Для програми J-Link значення має бути «java»
java_app_class	Клас Java для програми

java_app_classpath	Шлях де знаходяться класи Java.
java_app_start	Назва методу початку. Сигнатура має бути статичною.
java_app_stop	Назва методу зупинки. Сигнатура має бути статичною.
allow_stop	Булеве визначення, якщо додаток може бути зупинено.
delay_start	Визначає, чи завантаження програми при запуску, чи ні (значення може бути істинним чи невірним)
text_dir	Каталог ресурсів
end	Кінцеве ключове слово, воно не потребує значення.

Щоб Creo знайшов файл реєстрації додатків треба:

- 1) визначити файл реєстру (повний шлях) у виписці creotkdat, protkdat, prodevdat або toolkit_registry_file у файлі конфігурації Creo.
- 2) покласти файл реєстру в розділ <CREO_LOADPOINT> \ Common Files \ <DATECODE> \ text та перейменувати його в creotk.dat, protk.dat або prodev.dat.
- 3) зберегти файл із розширенням dat та зареєструйте його вручну за допомогою користувацького інтерфейсу (Утиліти >> Допоміжні програми >> Реєстрація).

Висновки до розділу 3

Однією з переваг використання Java в розробці розширення для CAD системи Creo можна навести статичну типізацію, великий розвиток мови за останні роки, що пропонує ціли низку необхідних функцій в розробці якісного програмного забезпечення, а також можливість працювати з веб технологіями. За допомогою J-LINK API здійснюється зв'язок з програмою Creo під час її запуску, таким чином,

що запускається створюване розширення. Для забезпечення якісного та комфортного створення на Java було використано середовище розробки IntelliJ IDEA.

Розробка на мові C# інших компонентів, таких як уніфіковане API операцій для PDM систем, розширень для CAD системи SolidWorks та PDM системи IT-Enterprise обумовлюється тим, що це високорівнева сучасна об'єктно-орієнтована мова з великою кількістю вбудованих типів з .NET Framework та особливостями платформ для яких створювалися дані розширення. Розробка даних компонентів відбувалася в середовищі Visual Studio від Microsoft.

XML – універсальна мова обміну даними в світі в даному контексті використовується для передачі даних між PDM та CAD системами.

Обґрунтовано основні засоби розробки програмного забезпечення для взаємодії PDM та CAD систем.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВЗАЄМОДІЇ PDM ТА CAD СИСТЕМ

Програмне забезпечення для взаємодії PDM та CAD систем складається з API для уніфікації операцій з боку PDM системи, розширень з боку CAD систем SolidWorks та PTC Creo та розширення з боку PDM системи IT-Enterprise, що реалізує вище зазначене API. Кожен з цих елементів має свої особливості та певну архітектуру. Нижче проведено аналіз цих компонентів та їх взаємодії.

4.1 Розширення для PDM системи IT-Enterprise

Розширення для PDM системи IT-Enterprise виконує функцію обробника команд, що поступають з CAD систем. Для уніфікації операцій було створено API для можливості реалізації його з боку PDM. Основою API є інтерфейс для забезпечення контракту виклику функцій з певної PDM системи, який показано на рисунку 4.1.

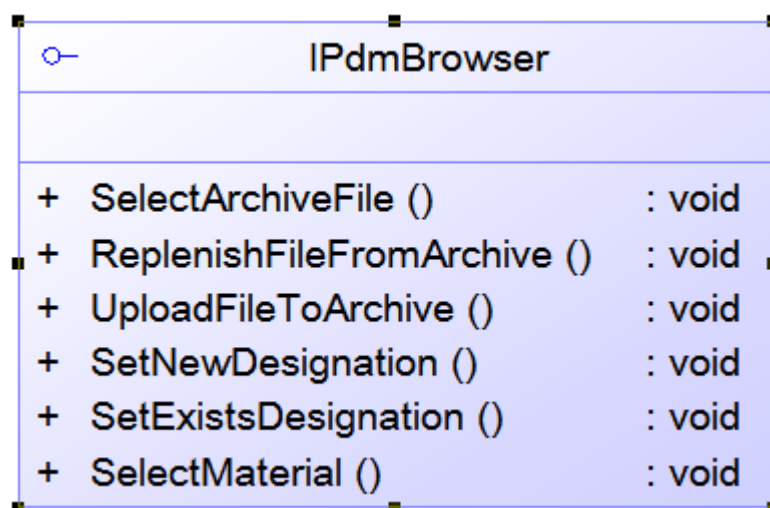


Рисунок 4.1 – Інтерфейс API зв'язку з PDM системою

Основними функціями цього API є:

- SelectArchiveFile - метод для вибору файлу з PDM системи;
- ReplenishFileFromArchive – метод поповнення збірки файлами з PDM системи;
- UploadFileToArchive – метод завантаження файлу в PDM систему;
- SetNewDesignation – метод присвоєння нового позначення;
- SetExistsDesignation – метод присвоєння вже існуючого позначення;
- SelectMaterial – метод вибору матеріалу.

Для реалізації даного API було створено наступну класову структуру як показано на рисунку 4.2.

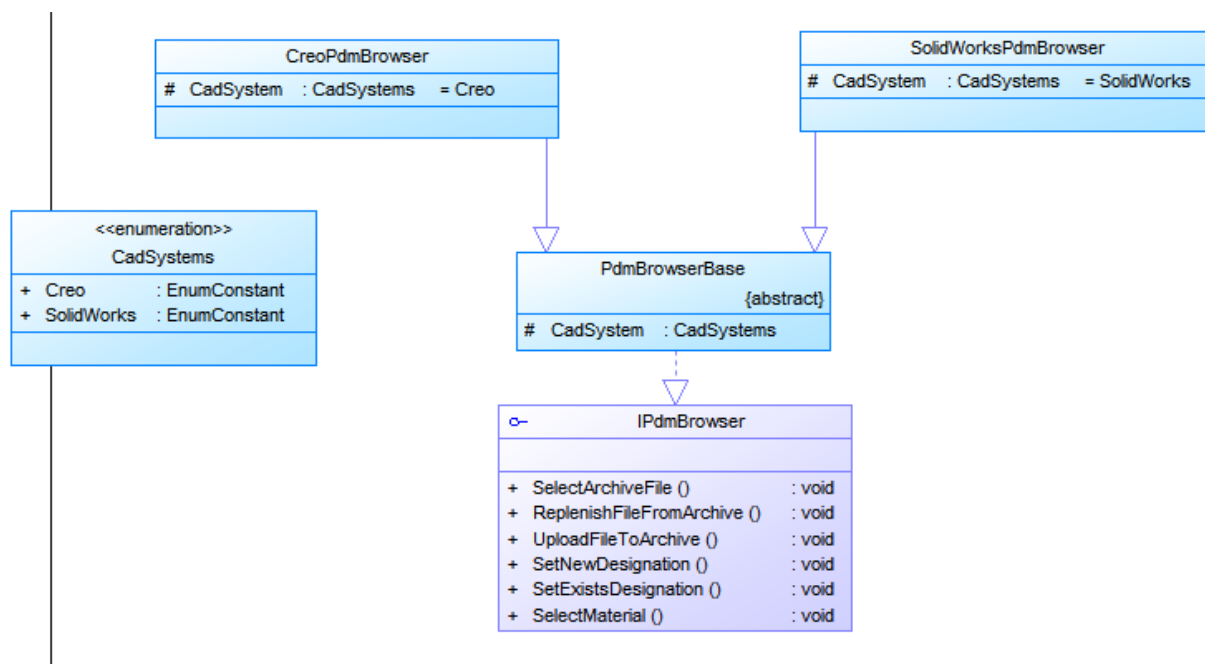


Рисунок 4.2 – Діаграма класів реалізації API зв'язку з PDM системою

В даному випадку API реалізовується за допомогою класу `PdmBrowserBase`, що є абстрактним та має поле типу `enum CadSystems`. Цей клас має двох спадкоємців `CreoPdmBrowser` та `SolidWorksPdmBrowser`, в яких значення поля `CadSystem` відповідно `Creo` та `SolidWorks`.

В залежності від типу CAD системи з PDM системи IT-Enterprise будуть завантажуватися різні налаштування, а саме, імена параметрів коду ресурсу, позначення, найменування, а також категорії та розширення для деталей та збірок.

Для виконання операцій завантаження файлу з архіву або поповнення файлу з архіву спочатку перевіряється, що наведені вище параметри не пусті. Якщо це так, то з параметрів запуску розширення береться ім'я файлу для передачі даними між PDM та CAD. З файлу зчитується інформація у вигляді XML та намагається параметр робочої папки куди треба відправити обраний в PDM системі файл. При отриманні необхідного значення, запускається вікно виробу документу в PDM системі, відфільтроване за документами, які мають файли сумісні з даною CAD розширеннями. Також відбувається перевірка необхідних прав доступу на копіювання файлу на клієнтську машину. Далі здійснюється вибір документу та за допомогою стратегії формується XML файл з даними про модель. Файл моделі та зв'язаних з нею моделей відправляється на клієнт у робочу папку.

Серед даних, що пересилаються клієнтові, присутні дані з документу такі як поля коду ресурсу, позначення та найменування, шлях файлу на клієнтській машині, код матеріалу для деталі. Сформований XML завантажують в файл, що завантажений з клієнта на початку роботи, таким чином забезпечуючи передачу даних в обидві сторони.

Для виконання операцій присвоєння позначення або присвоєння нового позначення розширення PDM системи відкриває класифікатор ресурсів в режимі вибору або в режимі додавання відповідно. Після вибору якогось з ресурсів відбувається передача його коду, позначення та найменування через XML файл. Операція присвоєння матеріалу має той же характер дії.

Для завантаження файлу в PDM систему знаходиться чи створюється, якщо немає, документ з кодом ресурсу та категорією, що надійшли з CAD системи.

При цьому заповнюються наступні таблиці: TDD, TDDS, TDFILES, ITTDPACK, ITTDPROP.

TDD – таблиця незмінних реквізитів документу від версії до версії, або як ще її називають карточка документів.

Далі приведено структуру таблиці карточки документів в таблиці 4.1.

Таблиця 4.1. Структура таблиці «TDD»

Назва	Тип поля	Опис
untdd	varchar (10)	Унікальний ідентифікатор карточки документу
tdk	varchar(10)	Категорія документу
kmat	varchar(15)	Код ресурсу
ndoc_full	varchar(255)	Найменування документу
ndoc_sfull	varchar(100)	Позначення

TDDS – таблиця змінних реквізитів документу в залежності від версії. Її структура зображена в таблиці 4.2.

Таблиця 4.2. Структура таблиці «TDDS»

Назва	Тип поля	Опис
untdds	varchar (10)	Унікальний ідентифікатор документу
untdd	varchar (10)	Унікальний ідентифікатор карточки документу
ttdv	varchar(10)	Вид архіву

TDFILES – таблиця файлів документів PDM системи. Її структура зображена в таблиці 4.3.

Таблиця 4.3. Структура таблиці «TDFILES»

Назва	Тип поля	Опис
id	varchar (20)	Унікальний ідентифікатор
filename	varchar(255)	Ім'я файлу

filesize	varchar(15)	Розмір файлу
extension	varchar(15)	Розширення
untdds	varchar(10)	Унікальний ідентифікатор документу

ITTDPACK – таблиця зв’язаних документів в PDM системі. Її структура зображена в таблиці 4.4.

Таблиця 4.4. Структура таблиці «ITTDPACK»

Назва	Тип поля	Опис
id	varchar (10)	Унікальний ідентифікатор
untdds1	varchar(10)	Унікальний ідентифікатор документ до якого прив’язаний поточний документ
untdds2	varchar(10)	Унікальний ідентифікатор поточного документу
relation	varchar(15)	Тип зв’язку

ITTDPROP – таблиця властивостей документу в PDM системі. Її структура зображена в таблиці 4.5.

Таблиця 4.5. Структура таблиці «ITTDPROP»

Назва	Тип поля	Опис
id	varchar (10)	Унікальний ідентифікатор
untdds	varchar(10)	Унікальний ідентифікатор документу
name	varchar(120)	Ім’я властивості
value	varchar(255)	Значення

Таким чином, структура вище зазначених таблиць заповнюється при завантаженні файлу в архів.

4.2 Розширення для CAD систем PTC Creo та SolidWorks

Розширення для CAD систем PTC Creo та SolidWorks пропонують набір кнопок в графічному інтерфейсі користувача для взаємодії з PDM системою.

Використовуючи J-LINK API для Creo та SolidWorks API для SolidWorks можна отримати доступ до моделей створюваних в CAD та їх властивостей.

Внутрішня структура класів представляє собою обробники натискання на кнопки в інтерфейсі користувача, визначення стратегій серіалізації та десеріалізації з/в XML, конвертування даних моделі тощо.

Таким чином, ми можемо представити структуру класів для розширень для CAD систем як показано на рисунку 4.3.

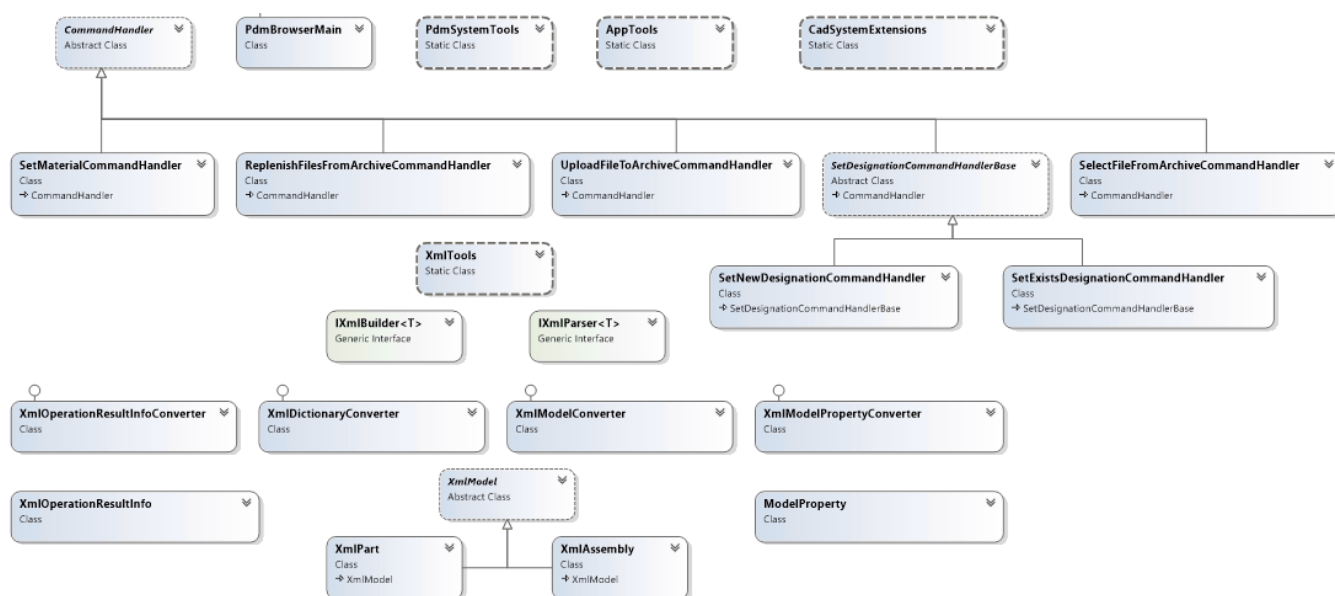


Рисунок 4.3 – Схематична діаграма класів розширень CAD систем

Операція «Взяти файл з архіву» представлена класом SelectFileFromArchiveCommandHandler. Для цієї операції алгоритм роботи наступний:

- вибрати робочу папку для завантаження в неї вибраної моделі із PDM;
- послати запит на IT-Enterprise, вказавши в параметрах файл, де в форматі XML лежить робоча папка;
- отримавши результат, розпарсити його використовуючи стратегії XML. Якщо результат прийшов помилковий, видати повідомлення. В іншому випадку створити модель в CAD за даними моделі з XML та заповнити відсутні властивості коду ресурсу, позначення, найменування;
- відкрити модель у новому вікні.

Для операції «Поповнити файл з архіву» (клас `ReplenishFilesFromArchiveCommandHandler`) алгоритм роботи дещо відрізняється. Ця операція доступна тільки для збірок, а кінцевим її результатом слугує поповнення збірки моделями з PDM системи замість відкриття їх у новому вікні.

Операція «Вибрати матеріал» (клас `SetMaterialComandHandler`) має наступний алгоритм:

- взяти модель і перевірити, що це деталь;
- послати запит в PDM систему та отримати відповідь. Якщо результат – помилка, видати повідомлення. В іншому випадку – записати у властивості моделі параметр із значенням матеріалу.

Операції «Присвоїти позначення» (клас `SetExistsDesignationCommandHandler`) та «Присвоїти нове позначення» (клас `SetNewDesignationCommandHandler`) практично не відрізняються один від одного:

- взяти модель і у випадку присвоєння нового позначення витягнути із властивостей моделі найменування та позначення якщо вони є;
- передати властивості разом з запитом в IT-Enterprise та отримати відповідь. Якщо результат – помилка, видати повідомлення. В іншому випадку – записати у властивості моделі параметр із значенням коду ресурсу, позначення та найменування.

Останньою операцією є «Завантаження файлу в архів» (клас `UploadFileToArchiveCommandHandler`). Для цього в XML вивантажується все дерево виробу із CAD системи разом з його залежностями, якщо такі є. Дані моделі в

форматі XML посилаються в IT-Enterprise та проходить завантаження файлів в архів. Назад в CAD систему передається ознака успішності виконаної операції.

Клас PdmBrowserMain відповідає за точку входу в додаток та завантажує користувацький інтерфейс. Клас AppTools слугує для службових операцій таких як перейменування чи видалення файлу тощо. PdmSystemTools слугує для зв'язку з конфігураційним файлом налаштування з якого береться інформація про властивості коду ресурсу, найменування, позначення та параметри запуску розширення для PDM системи. Клас CadSystemsExtensions працює з API CAD системи оперуючи необхідними даними для відображення користувачу. XmlTools здійснює конвертацію в XML необхідних даних для відправки в PDM систему.

Моделі CAD поділяють на моделі деталей та моделі збірок. Моделі деталей представляють собою просту геометричну модель без вхідних моделей, наприклад, гайки, болта тощо. Моделі збірок представляють собою ряд взаємодіючих між собою ієрархічних геометричних організованих об'єктів, складних за структурою, наприклад двигун тощо. Загальна схема моделі показано на рисунку 4.4.

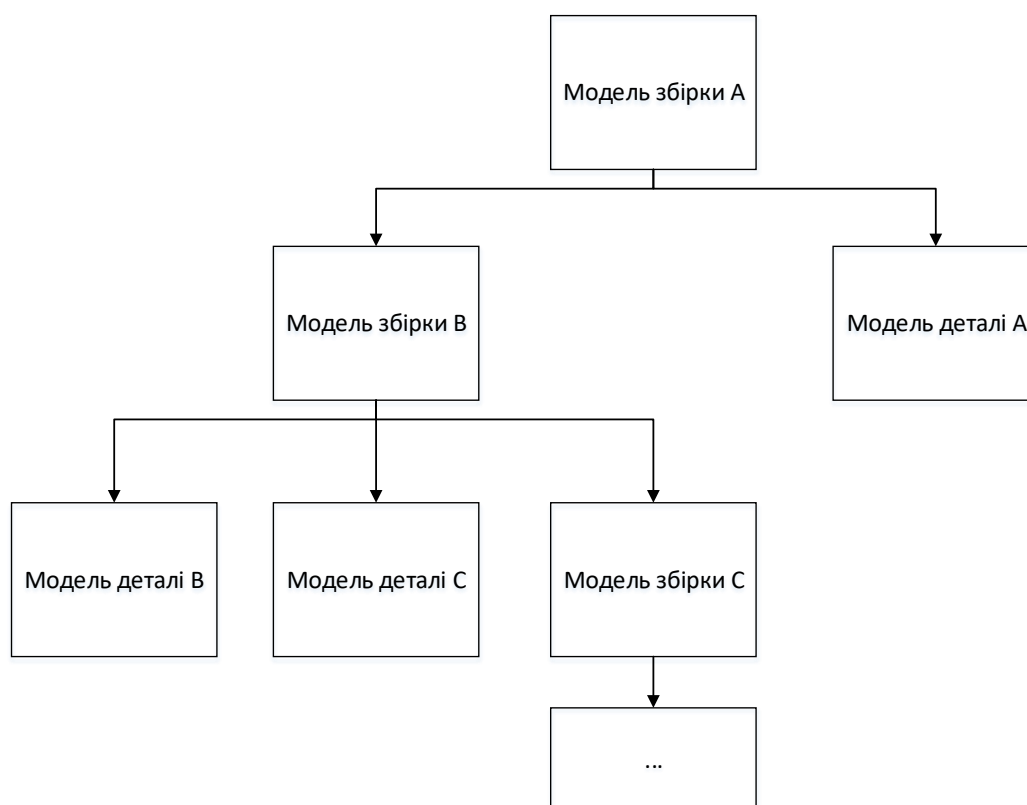
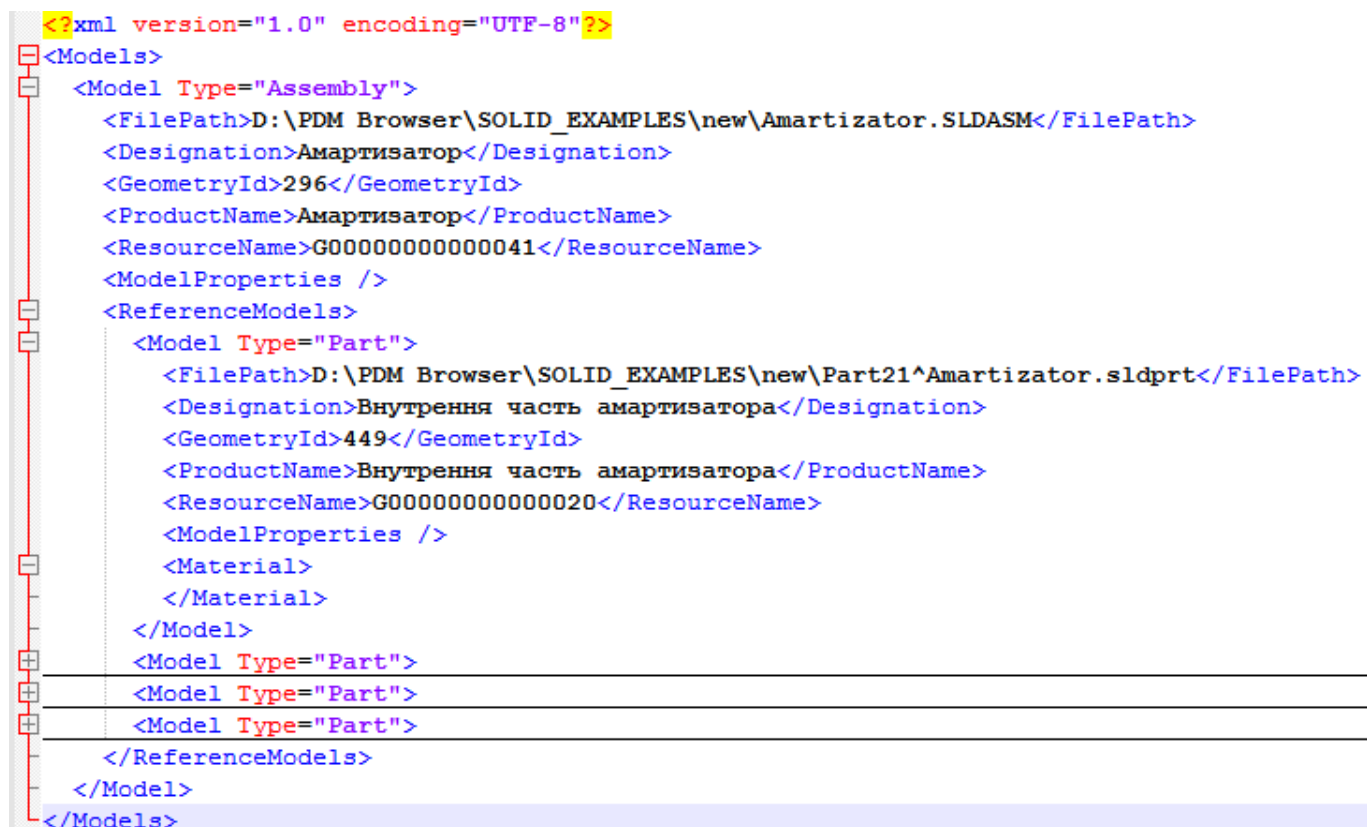


Рисунок 4.4 – Загальна схема відображення 3D моделі CAD

На рисунку 4.5 показано вигляд XML файлу моделі, яка передається між CAD та PDM системами.



```
<?xml version="1.0" encoding="UTF-8"?>
<Models>
  <Model Type="Assembly">
    <FilePath>D:\PDM Browser\SOLID_EXAMPLES\new\Amartizator.SLDASM</FilePath>
    <Designation>Амартизатор</Designation>
    <GeometryId>296</GeometryId>
    <ProductName>Амартизатор</ProductName>
    <ResourceName>G000000000000041</ResourceName>
    <ModelProperties />
    <ReferenceModels>
      <Model Type="Part">
        <FilePath>D:\PDM Browser\SOLID_EXAMPLES\new\Part21^Amartizator.sldprt</FilePath>
        <Designation>Внутренняя часть амартизатора</Designation>
        <GeometryId>449</GeometryId>
        <ProductName>Внутренняя часть амартизатора</ProductName>
        <ResourceName>G000000000000020</ResourceName>
        <ModelProperties />
        <Material>
        </Material>
      </Model>
      <Model Type="Part">
      </Model>
      <Model Type="Part">
      </Model>
    </ReferenceModels>
  </Model>
</Models>
```

Рисунок 4.5 – Приклад XML файлу з передачею 3D моделі між CAD та PDM системою

Основними атрибутами при передачі моделі через XML є шлях моделі на машині клієнта, основні властивості коду ресурсу, позначення та найменування, а також інші властивості, що необхідно передати з системи в систему. Для збірки передаються всі вхідні в неї моделі. Таким чином моделі CAD передаються між PDM та CAD системами.

4.3 Функціональна структура системи

Основні задачі, які може виконувати користувач, представлені на рисунку 4.6 на діаграмі прецедентів.

У користувача програми є наступні можливості:

- вибрати файл з PDM системи;
- поповнити збірку файлами з PDM системи;
- присвоїти позначення;
- присвоїти нове позначення
- вибрати матеріал;
- вивантажити файли в PDM систему.

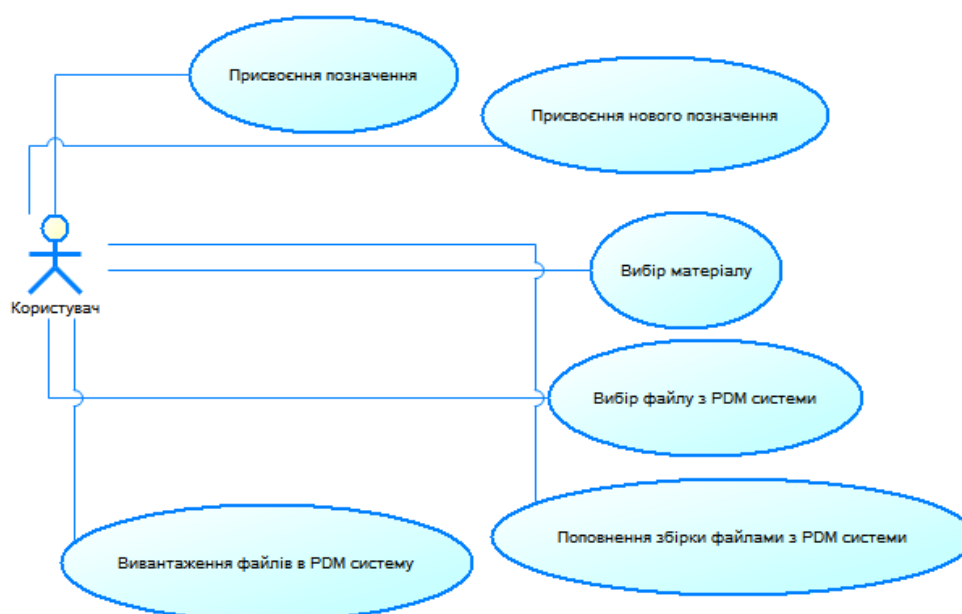


Рисунок 4.6 – Основні задачі, які може виконувати користувач

Висновки до розділу 4

Створено програмне забезпечення для взаємодії PDM та CAD систем.

Розглянуто основні архітектурні рішення, які використовувалися при створенні програмного забезпечення для взаємодії PDM та CAD систем. Було описано алгоритм дії всіх операцій як з боку CAD системи так і з боку PDM. Розглянуті основні таблиці при роботі з базою даних.

Програмне забезпечення для взаємодії PDM та CAD систем уніфікує інтерфейс взаємодії між системами автоматизованого проектування та контролю виробів.

РОЗДІЛ 5. МЕТОДИ РОБОТИ КОРИСТУВАЧА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ ДЛЯ ВЗАЄМОДІЇ PDM ТА CAD СИСТЕМ

Основними компонентами роботи користувача з програмним забезпеченням для взаємодії PDM та CAD систем є інтерфейси користувача зі сторони CAD систем SolidWorks та PTC Creo. За допомогою них користувач взаємодіє з PDM системою за допомогою набору уніфікованих операцій.

В наступних розділах буде описано як підключити програмне забезпечення для взаємодії PDM та CAD систем та використовувати вище зазначені операції.

5.1 Підключення програмного забезпечення для взаємодії PDM та CAD систем

Для роботи програмного забезпечення для взаємодії треба мати встановлені ліцензійні CAD системи SolidWorks або/і PTC Creo та систему IT-Enterprise з модулем PDM. Далі наведемо необхідні кроки для підключення та налаштування компонентів програмного забезпечення для взаємодії PDM та CAD систем.

Для того, щоб підключити розширення для PDM системи IT-Enterprise необхідно зайти в систему та знайти пункт «Бібліотека коду C#» за таким шляхом «Адміністрування системи → Адміністрування системи і управління доступом → Бібліотеки коду → Бібліотеки коду C #». Додати файл розширення PDMBrowser.dll натиснувши F2 і вибравши пункт випадаючого меню «Імпорт». Таким чином файл розширення буде успішно додано в систему як показано на рисунку 5.1.

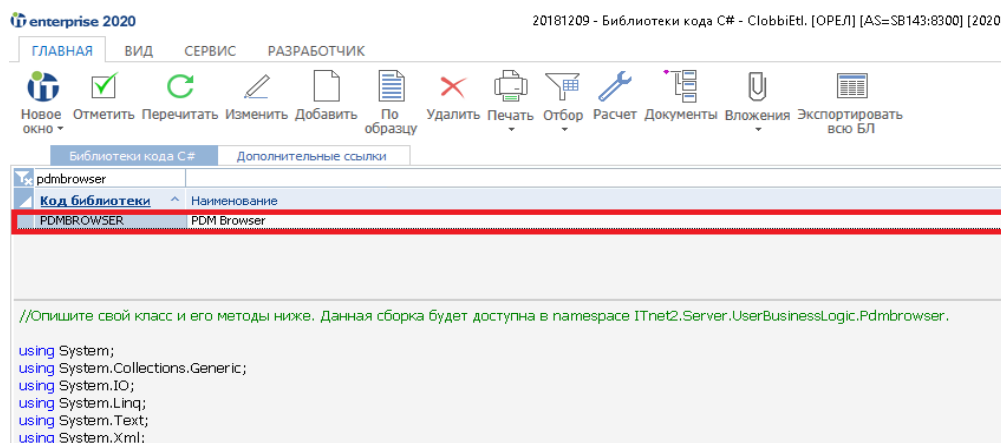


Рисунок 5.1 – Розширення для PDM системи IT-Enterprise

Для встановлення розширення для SolidWorks необхідно розпакувати вміст архіву SolidWorksPdmBrowser в папку на машині клієнта. В розпакованій папці будуть міститися файл розширення для SolidWorks SolidWorksPdmBrowser.dll, файл конфігурацій SolidWorksPdmBrowser.dll.config, збірки C# з API SolidWorks та ресурсні файли. Для запуску в SolidWorks даного розширення необхідно зареєструвати його в реєстрі Windows. Для цього необхідно запустити системну утиліту RegAsm.exe з консолі Windows під іменем адміністратора передавши в параметри шлях до збірки SolidWorksPdmBrowser.dll як показано на рисунку 5.2.

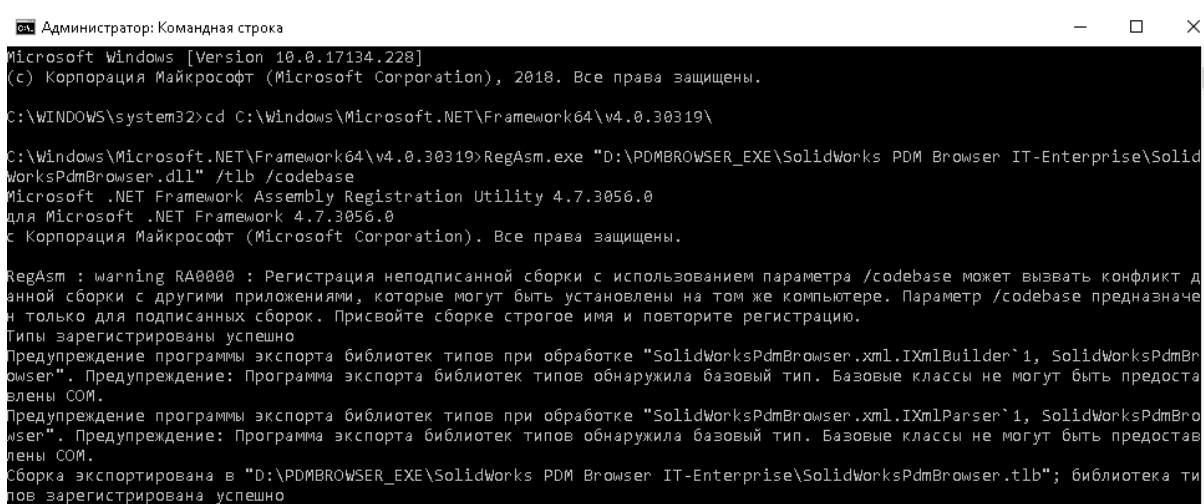


Рисунок 5.2 – Реєстрація розширення для CAD системи SolidWorks

Після успішного встановлення знаходимо розпаковану папку з розширенням та заходимо в файл конфігурації, який показано на рисунку 5.3. В параметр «ITClientExe» треба прописати повний шлях на машині клієнта, де у нього

встановлений PDM система IT-Enterprise. Параметр «IT2ClientProjParam» відповідає за те з якого тестового клієнта IT-Enterprise запущене дане розширення, якщо на машині клієнта (для запуску на підприємствах цей параметр завжди залишається пустим). В параметрі «ObjKey» встановлюється ID клієнта IT-Enterprise, якщо їх на машині декілька.



```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3    <appSettings>
4      <add key="It2ClientExe" value="C:\Users\orel\AppData\Roaming\IT-CurrentClient\IT2Client.exe" />
5      <add key="It2ClientProjParam" value="K_CLOBBI_RU" />
6      <add key="ObjKey" value="00001" />
7      <add key="SetNewDesignation" value="SETNSWDES" />
8      <add key="SetExistsDesignation" value="SETESWDES" />
9      <add key="SelectArchiveFile" value="SELASWFILE" />
10     <add key="ReplenishFileFromArchive" value="REPLSWASM" />
11     <add key="UploadFileToArchive" value="UPLSWFILE" />
12     <add key="GetUserInfo" value="USERINFO" />
13     <add key="NameResourceIdentifier" value="RESCODE" />
14     <add key="NameIdentification" value="DESIGNATION" />
15     <add key="NameProductName" value="NAMEPROD" />
16     <add key="NeedMaterial" value="no" />
17     <add key="MaterialDataBase" value="itMaterialDataBase" />
18   </appSettings>
19 </configuration>

```

Рисунок 5.3 – Конфігураційний файл для розширення CAD системи SolidWorks

Щоб активувати розширення для SolidWorks треба зайти в CAD систему в пункт меню «Інструменти → Додавання». Далі треба знайти розширення «PDM Browser» та поставити галочки для його запуску. Галочка зліва означає, що розширення буде одразу ж запущено. Галочка справа означає, що розширення буде запускатися кожного разу при вході в CAD систему. Таким чином, після відмічення цих пунктів розширення буде готове до роботи з інтерфейсу CAD системи SolidWorks. Вікно активації розширень показано на рисунку 5.4.

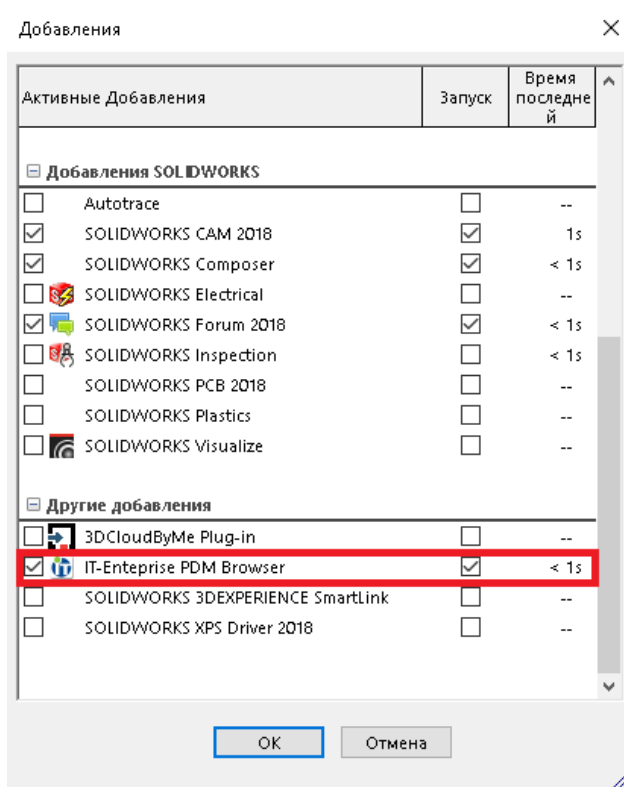


Рисунок 5.4 – Активация розширення для CAD системи SolidWorks

Для встановлення розширення для CAD системи PTC Creo необхідно розпакувати вміст архіву CreoPdmBrowser в папку на машині клієнта. В розпакованій папці знаходимо файли config.pro та protk.dat. Копіюємо їх в робочу папку Creo. Після цього заходимо в реєстраційний файл розширення protk.dat і змінюємо шлях в параметрі java_app_classpath на шлях до щойно розпакованого розширення, а в text_dir – на «шлях до щойно розпакованого розширення\text» як показано на рисунку 5.5.

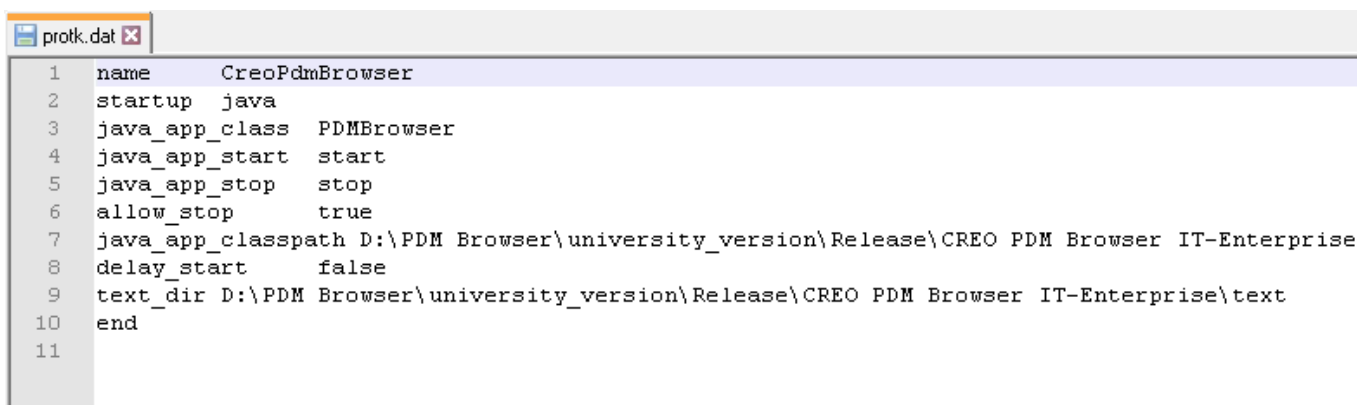


Рисунок 5.5 – Реєстраційний файл для розширення CAD системи PTC Creo

Щоб розширення запустилося при загрузці PTC Creo треба додати в змінні середовища змінні «CLASSPATH» та «PATH» зі значенням шляху до J-LINK API, що знаходиться в папка PTC Creo при встановленні, а також в змінну «PRO_JAVA_COMMAND» треба додати шлях до java.exe (рисунок 5.6).

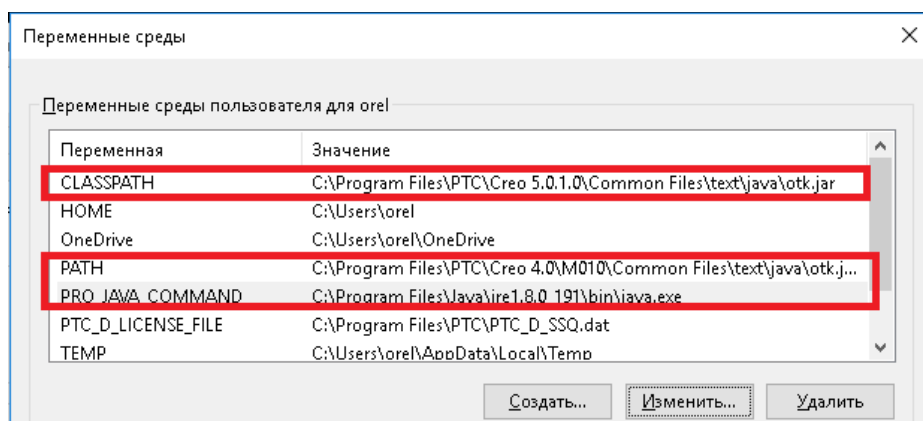


Рисунок 5.6 – Заповнення програмних змінних середовища Windows

В папці розширення можна знайти конфігураційний файл, в якому треба змінити параметри аналогічно як було в SolidWorks (рисунок 5.7).

```

config.txt — Блокнот
Файл Правка Формат Вид Справка
-It2ClientExe
\\it\dfs\dev\itnet\Versions\Current\Binaries\Client\IT2Client_.exe
#
#
-It2ClientProjectParam
K_CLOBBI_RU /srv=localhost:8300 /secure /encrypt
#
#
-ObjectKey
00001
#
#
-NeedMaterial
no
#
#
-SetMaterialKeyMenu
SELMAT
#
#

```

Рисунок 5.7 – Конфігураційний файл для розширення CAD системи PTC Creo

Після цього розширення готове до роботи з інтерфейсу CAD PTC Creo.

5.2 Інструкція роботи користувача з програмним забезпеченням для взаємодії PDM та CAD систем

Робота користувача з програмним забезпеченням для взаємодії PDM та CAD систем заключає в собі роботу в CAD системах за допомогою графічних інтерфейсів користувача, що реалізуються їх розширеннями. На рисунку 5.8 можна побачити даний інтерфейс користувача в SolidWorks.

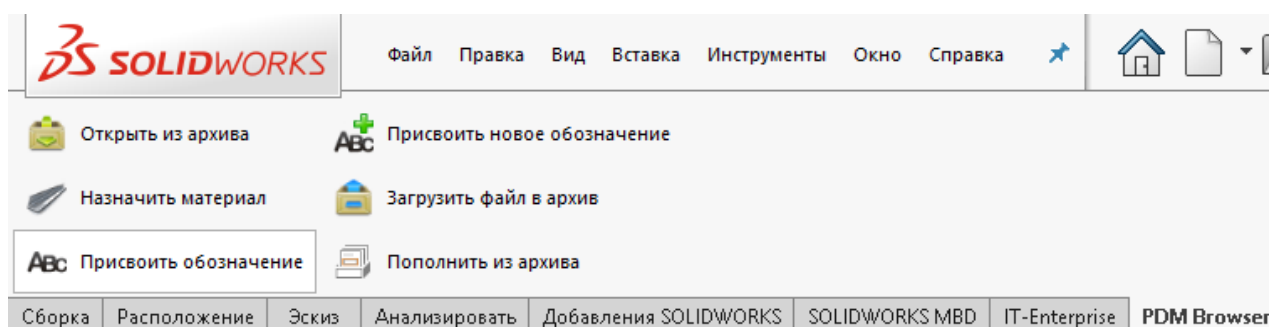


Рисунок 5.8 – Інтерфейс програмного забезпечення для взаємодії PDM та CAD в SolidWorks

При розробці інтерфейсів користувача було притримано уніфікованого вигляду для полегшення порогу входження користувачем використання нової системи. На рисунку 5.8 показано інтерфейс користувача в CAD системі PTC Creo.

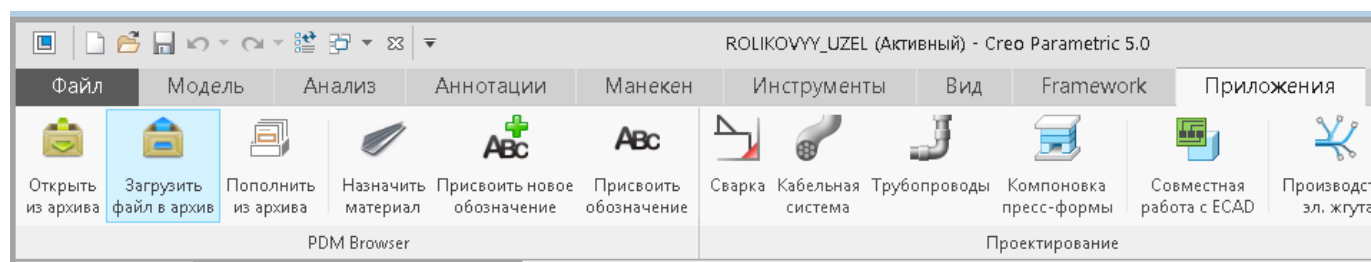


Рисунок 5.9 – Інтерфейс програмного забезпечення для взаємодії PDM та CAD в PTC Creo

В момент першого входу в PDM систему необхідно пройти аутентифікацію користувача. Перевіряється доступ користувача до PDM системи та можливості

виконання тих чи інших операцій. На рисунку 5.10 можна побачити як виглядає екран аутентифікації користувача в CAD системі SolidWorks.

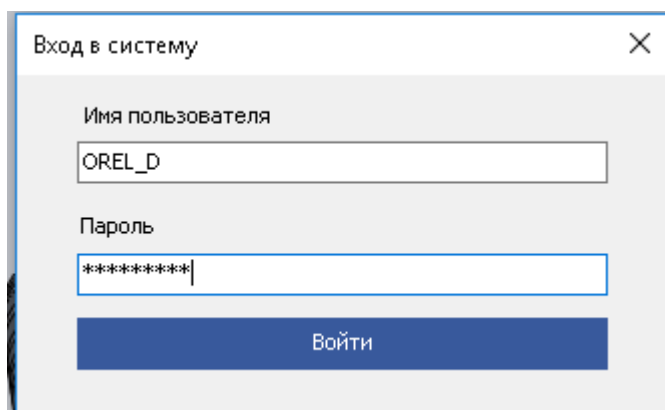


Рисунок 5.10 – Екран аутентифікації користувача в PDM систему з SolidWorks

На рисунку 5.11 знаходиться екран аутентифікації користувача в CAD системі PTC Creo.

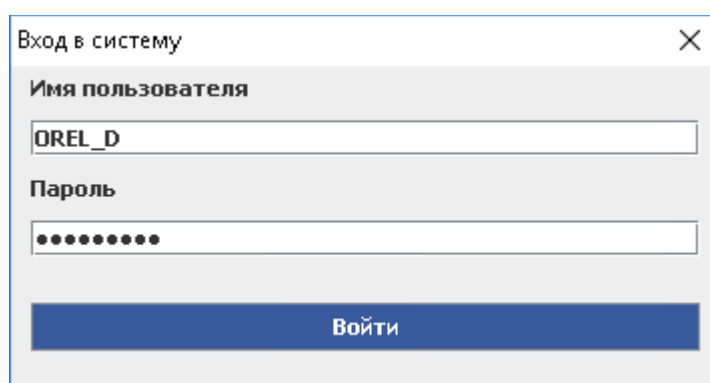


Рисунок 5.11 – Екран аутентифікації користувача в PDM систему з PTC Creo

Кожен співробітник конструкторського підрозділу проектує певну частину виробу: деталь чи збірку. Для того, щоб дати зрозуміти PDM системі з яким саме ресурсом працює проектувальник необхідно зв'язати цей ресурс із PDM системи з моделлю CAD системи.

Для цієї задачі існує класифікатор ресурсів в PDM системі IT-Enterprise, який містить довідкову інформацію про ресурси в системі. Цей інструмент допомагає

ідентифікувати ресурси, якими потім будуть використовуватися в CAD системі та при створенні документів безпосередньо в PDM системі.

Щоб обрати необхідний ресурс в PDM системі з CAD системи необхідно на вкладці «PDM Browser» викликати операцію «Присвоїти позначення». Викликається класифікатор ресурсів з фільтром на готову продукцію.

Для заповнення класифікатору ресурсів можна відкрити IT-Enterprise та перейти в пункт меню за шляхом «Адміністрування системи → Управління основними даними (MDM) → Ресурси → Групи, підгрупи ресурсів».

В ньому можна додати нові ресурси для різних груп, наприклад матеріалів, готової продукції тощо. Після цього користувач маючи ресурс в PDM системі може скористатися ним з CAD системи.

На рисунку 5.12 можна побачити інтерфейс класифікатору ресурсів відкритий з CAD системи SolidWorks.

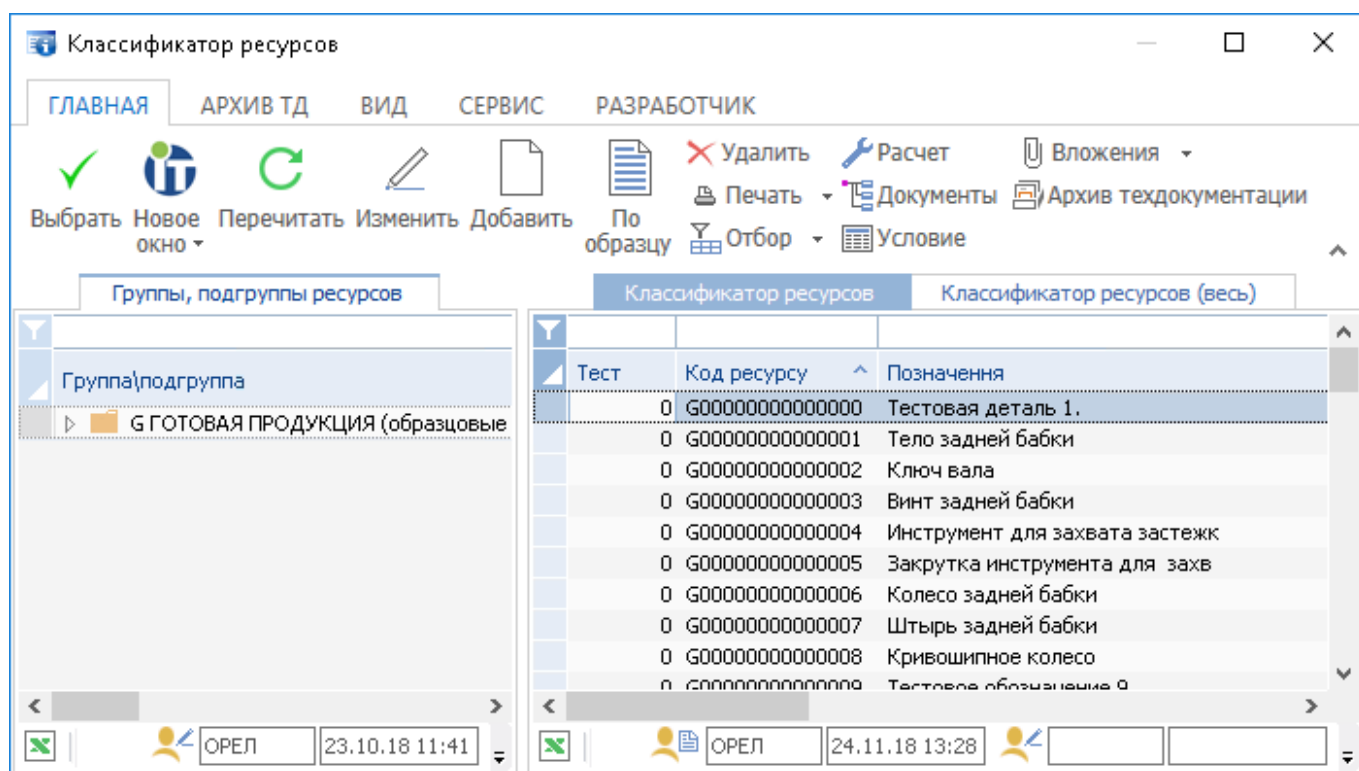


Рисунок 5.12 – Інтерфейс класифікатора ресурсів PDM системи з IT-Enterprise

Після вибору необхідного ресурсу дані надходять до CAD системи. У властивостях моделі можна знайти такі параметри як «Код ресурсу», «Позначення»,

«Найменування», імена яких беруться з конфігураційних файлів розширень до CAD систем. Зокрема ця операція викликає зміну імені файла на позначення ресурсу. На рисунку 5.13 можна побачити властивості моделі в SolidWorks після присвоєння позначення.

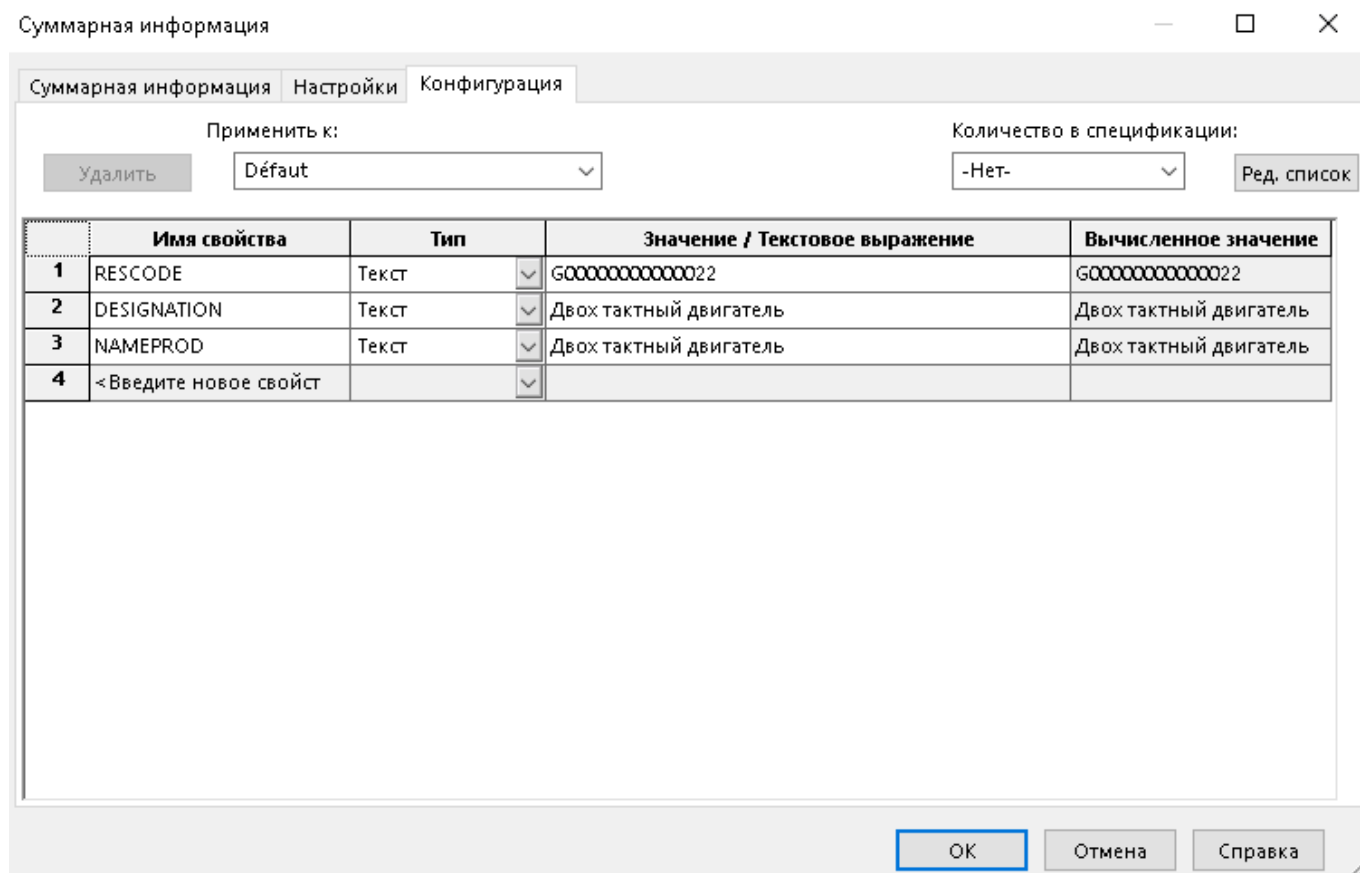


Рисунок 5.13 – Властивості моделі SolidWorks після виконання операції «Присвоїти позначення»

Подібним чином працює операція «Присвоїти матеріал». Вона доступна тільки для типу моделі «деталь», так як збірка є набором деталей і тому матеріалу у неї немає. Після виклику програма відкриває класифікатор ресурсів з фільтром на матеріали та користувач вибирає необхідний матеріал.

Після вибору код матеріалу зберігається в властивості моделі з визначений ім'ям. На рисунку 5.14 можна побачити властивості моделі Creo після присвоєння деталі матеріалу.

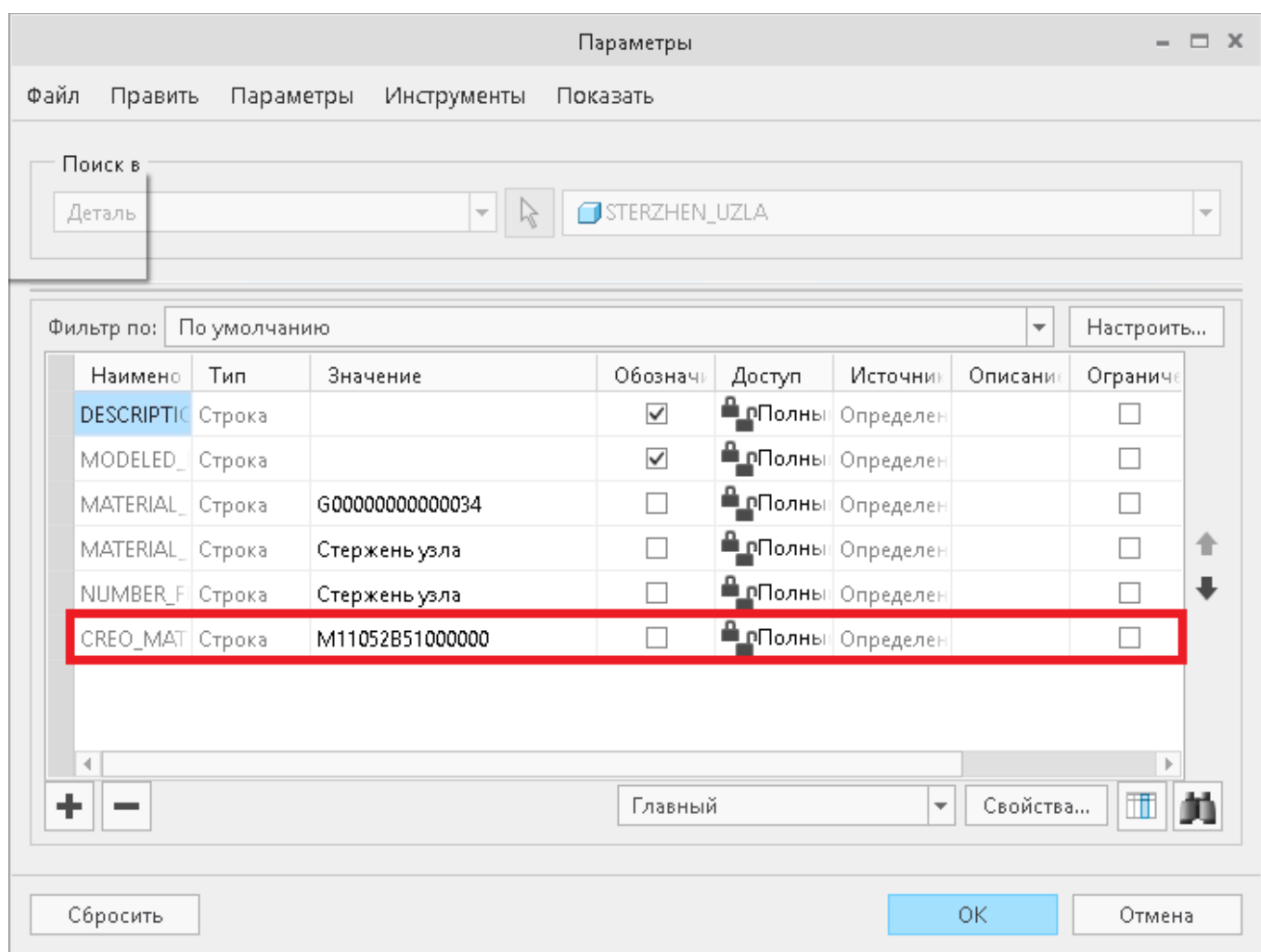


Рисунок 5.14 – Властивості моделі PTC Creo після виконання операції
«Присвоїти матеріал»

Замість того, щоб заходити в PDM систему і створювати новий ресурс, коли він відсутній в класифікаторі, можна скористатися операцією «Присвоїти нове позначення». За допомогою цієї операції відкривається класифікатор ресурсів в режимі додавання.

Перед користувачем з'являється вікно коригування, в якому, ввівши всі необхідні реквізити, він може додати новий ресурс до класифікатора та присвоїти відповідні параметри в моделі CAD системи.

Зокрема ця операція викликає зміну імені файла на позначення ресурсу як і операція «Присвоїти позначення».

На рисунку 5.15 можна побачити екран додавання ресурсу при виклику операції «Присвоїти нове позначення».

Рисунок 5.15 – Экран додавання нового ресурсу в класифікаторі ресурсів PDM системи

Операції «Відкрити з архіву» та «Поповнити з архіву» є схожими. Для початку користувач обирає папку для вигради туди моделей з PDM системи. Далі обидві операції звертаються до PDM системи для вибору документу, який містить моделі поточної CAD системи.

В першому випадку користувач обирає документ в якому міститься модель, вона передається клієнту на машину та відкривається в новому вікні CAD системи.

Друга ж операція використовується тільки для поповнення збірок. Користувач може обрати декілька документів чиї моделі будуть завантажені на машину користувача та по чергово додані до збірки відкритої в поточному вікні.

Обидві операції дозволяють проектувальникам використовувати вже створені деталі та збірки замість створення їх власноруч, що прискорює швидкість розробки та зменшує кількість помилок при проектуванні однієї і тієї ж моделі. Розглянемо операцію «Поповнити з архіву».

На рисунку 5.16 можна побачити модель двигуна в системі автоматизованого проектування SolidWorks, яка буде поповнюватися моделлю із PDM системи.

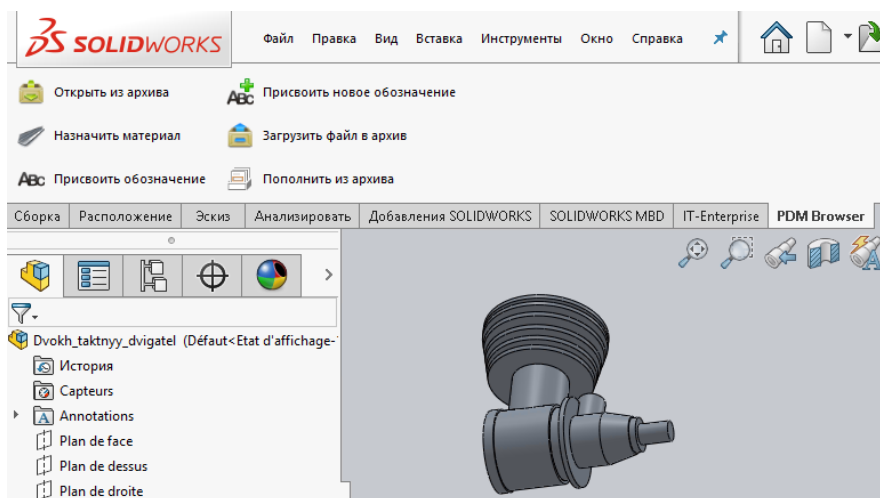


Рисунок 5.16 – Збірка в CAD системі SolidWorks

На рисунку 5.17 можна побачити вибір документу з моделлю амортизатору для поповнення збірки.

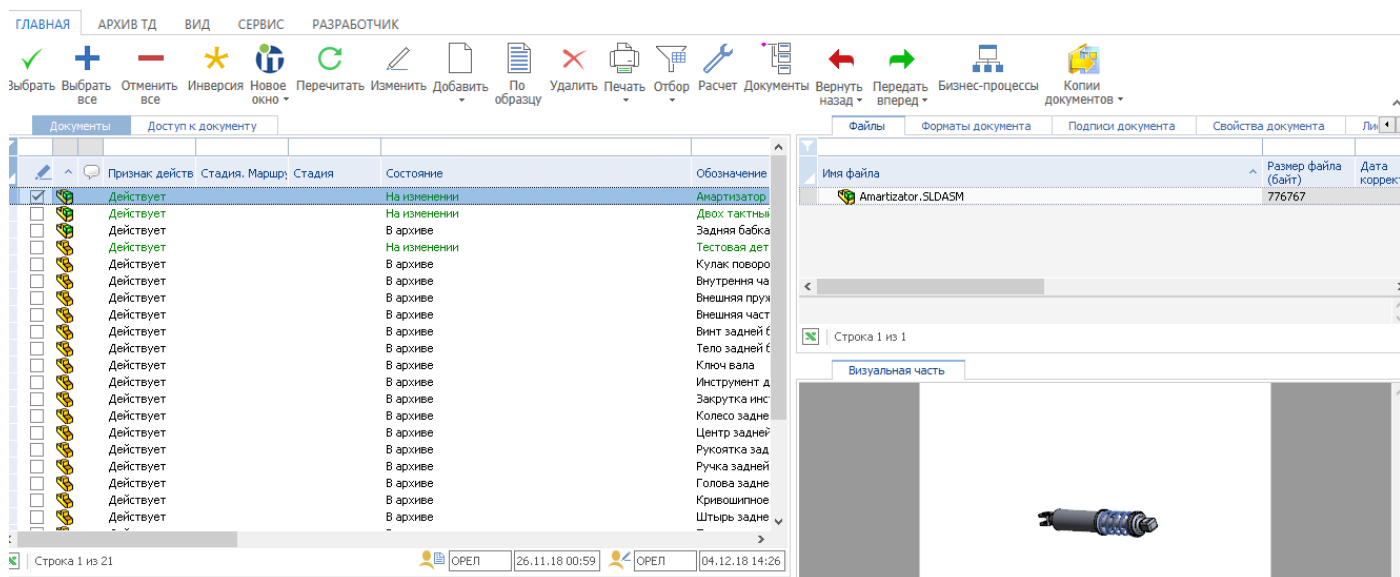


Рисунок 5.17 – Вибір моделі SolidWorks з PDM системи для можливості поповнення поточної збірки

На рисунку 5.18 можна побачити, що обрана модель додається до збірки при поповненні збірки цією моделлю.

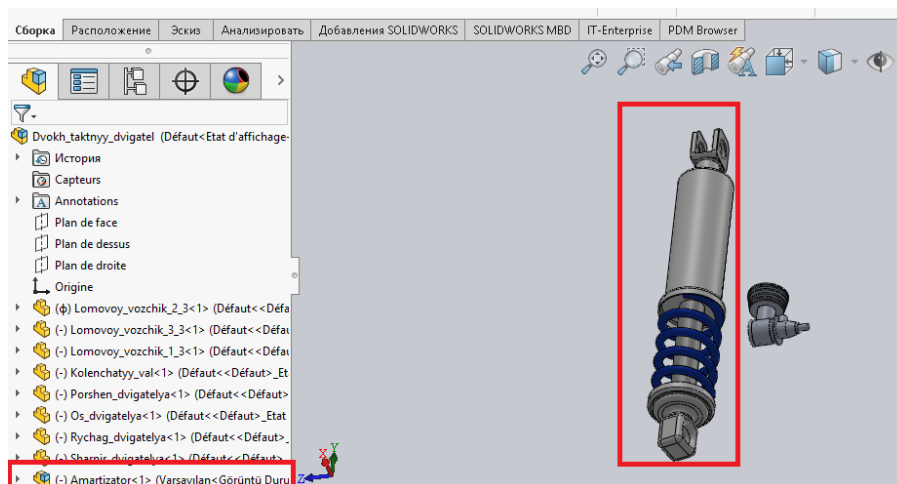


Рисунок 5.18 – Поповнена збірка моделлю з PDM системи

Кінцевою операцією є вивантаження моделі в архів за допомогою операції «Завантажити в архів». Проектувальник в будь-який момент може вивантажити в архів створену модель для загального доступу. Результатом цієї операції є вивантаження зв'язаних файлів моделей в PDM систему.

В PDM системі створюються або знаходяться відповідні документи з кодом ресурсу, позначенням та найменуванням, що беруться з властивостей моделі. Ці документи беруться «На зміну» даному користувачеві, що означає, що тільки він їх може правити. Файли моделі прив'язуються до цих документів, а матеріал у властивостях деталі переходить до властивостей документів. Документи моделей, що входять один до одного стають зв'язаними документами, що забезпечує можливість не втратити дані при виграді файлів верхнього рівня (тобто з файлами збірок на машину клієнта будуть вивантажуватися файли моделей, що входять в цю збірку).

На рисунку 5.19 зображено створені документи після вивантаження їх в PDM систему.

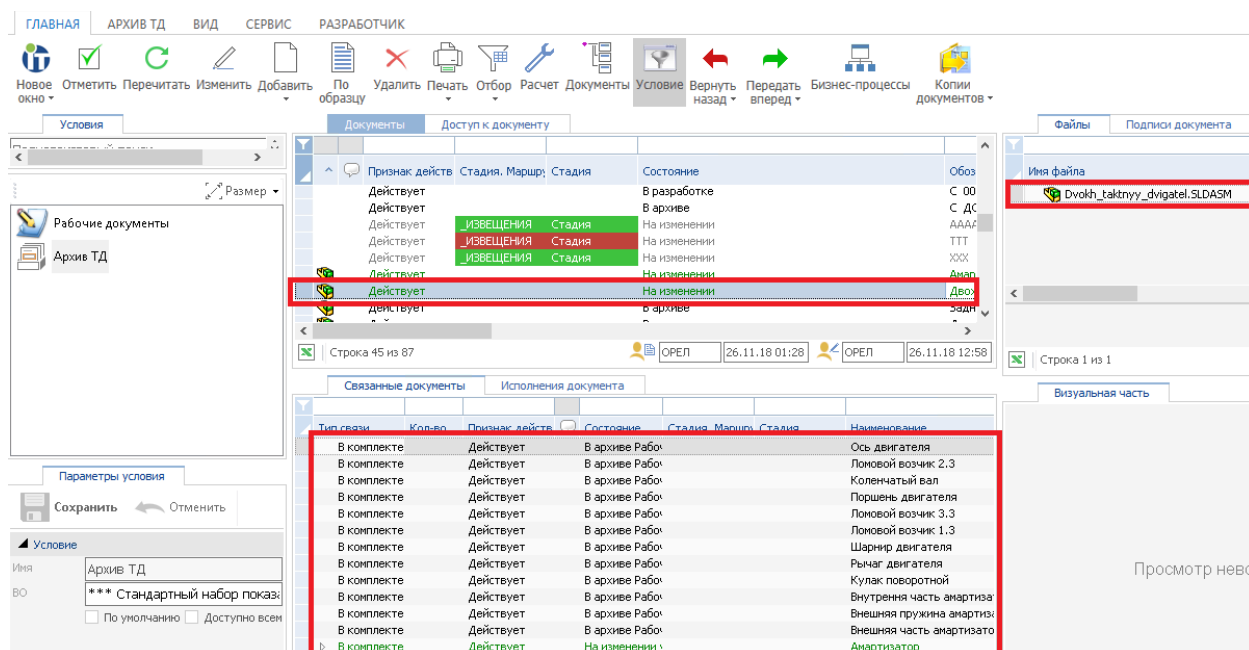


Рисунок 5.19 – Вивантажені в архів моделі з CAD системи SolidWorks за допомогою операції «Завантажити в архів»

Висновки до розділу 5

Розглянуто методику роботи користувача з системою, а також представлено один із сценаріїв роботи розробленого програмного забезпечення для взаємодії PDM та CAD систем.

Продемонстровано функціонал можливості «Присвоєння позначення» з ресурсів PDM системи, що в свою чергу скорочує час на пошук необхідних даних в системі. За допомогою операції «Присвоєння нового позначення» в ресурсах PDM системи можна доповнювати довідникову інформацію працюючи в CAD системі. Операція «Присвоєння матеріалу» необхідна для присвоєння матеріалу для деталей з подальшим використанням цих даних в CAD системі. За допомогою операцій «Взяття файлу з архіву» та «Поповнення файлом з архіву» можна отримати доступ до вже зареєстрованих в PDM системі моделей CAD та додати їх до свого проекту. Продемонстровано вивантаження файлів моделей в PDM систему з подальшим створенням документів на базі інформації моделей CAD.

РОЗДІЛ 6. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

Розробка стартап проекту означає встановлення цілей високого рівня, які підприємець хоче досягти у компанії. Приймаючи проект та об'єднуючи елементи дій, великі та малі для досягнення ваших цілей, є початком реалізації ідеї.

Частина досягнення якої-небудь цілі полягає в тому, щоб розбити її на менші, керовані частини, які легко вирішити.

Таким чином опис стартап проекту буде розбито на такі етапи:

- 1) розробляється опис ідеї проекту та визначаються напрями використання товару чи послуги та їх відмінність від конкурентів;
- 2) аналізуються ринкові можливості;
- 3) на базі аналізу ринку розробляється стратегія ринкового впровадження.

6.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- 1) зміст ідеї (що пропонується);
- 2) можливі напрями застосування;
- 3) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- 4) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці таблиця 6.1.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Забезпечення інтеграції декількох САД	1. Колективне проектування нових виробів із паралельним	1. Оптимізація та прискорення часу розробки.

систем та PDM системи	використанням різних CAD систем.	
	2.Збереження даних в єдину PDM систему	2. Не залежить від PDM системи, гарантує цілісність та повноту даних про створюваний виріб.

В таблиці 6.2 проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 6.2).

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Значення показників	Мій проект	Інтегровані PDM та CAD одного виробника	Інтегровані PDM та CAD за допомогою API
1	W слабка сторона	Відносно складна схема налаштувань	При зміні однієї системи треба міняти і іншу.	Клаптева автоматизація
2	N нейтральна сторона	Відносно помірний час роботи	Налаштування відносно помірної складності	Можливість інтегрувати будь-які PDM та CAD
3	S сильна сторона	Можливість інтеграції будь-якої PDM та CAD	Швидкий час роботи з сумісними системами	Ціленаправлені на вирішення невеликих задач

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту наводиться у таблиці 6.3.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Інтерфейс користувача	Мова програмування C#, Java	Наявна	Умовна безкоштовно
2	Розширювана база даних	SQL, MS Server	Наявна	Умовна безкоштовно
3	Алгоритм передачі даних між PDM та CAD системами	Мова програмування C#, Java	Наявна	Умовна безкоштовно
4	Алгоритм збереження та контролю даних	Мова програмування C#	Наявна	Умовна безкоштовно

Висновок: проект реалізувати можливо.

Обрана технологія реалізації ідеї проекту: удосконалення способу інтегрування між PDM та CAD системами за рахунок уніфікації інтерфейсу взаємодії між ними.

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як SolidWorks API, J-LINK API та XML через їх доступність та безкоштовність.

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку, який показано в таблиці 6.4.

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	400 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	50 %

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу (таблиця 6.4) було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та зформовано орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/ п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Удосконалення способу інтегрування між PDM та CAD системами	PDM та CAD системи	Компанії заключають довготривалі договори, а стартапери віддають перевагу пробному терміну	стабільність роботи; невисока ціна; наявність випробувального періоду; наявність документації; підтримка необхідних платформ оптимізований час;

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиця 6.6, 6.7).

Таблиця 6.6. Фактори загроз

Но п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Обмеженість функцій	наявними функціями і не має деяких функцій, які мають конкуренти	Додавання нових функцій за потреби

Таблиця 6.7. Фактори можливостей

Но п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Незалежність від CAD та PDM	Можна використовувати з різними CAD та PDM	Вихід на широкий ринок підприємств розробки складної продукції
2	Недоліки в існуючих альтернативах	Існуючі альтернативи не гнучкі в плані інтеграції, або не є орієнтованими на конкретну предметну область	Модифікація існуючих методів інтеграції

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 6.8).

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>
1. Вказати тип конкуренції - олігополія	На ринку присутні декілька компаній- конкурентів, але їх товар дещо відрізняється між	Підтримка якості продукту та постійні нововведення, вдосконалення.

	собою.	
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - марочна	Бренд присутній і його роль значна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (таблиця 6.9).

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	-	Dassault Systemes	Визначити фактори сили в постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників

Висновки:	Прямі конкуренти відсутні	Є можливості виходу на ринок, оскільки існуючі рішення не надають потрібних переваг	Постачальники підлаштовуються під ринок	Клієнти диктують вимоги згідно з умовами експлуатації	Обмеження для роботи на ринку через товари заміники
-----------	---------------------------	---	---	---	---

За результатами аналізу таблиця 6.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентноспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентноспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в таблиці 6.9, а також із урахуванням характеристик ідеї проекту (таблиця 6.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиці 6.6, 6.7) визначається та обґрунтовується перелік факторів конкурентноспроможності. Аналіз оформлено у таблиці 6.10.

Таблиця 6.10. Обґрунтування факторів конкурентноспроможності

№ п/п	Фактор конкурентноспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Використання сучасних технологій	Використання сучасних технологій збільшить привабливість та конкурентноспроможність продукту
2	Пропозиція можливостей яких не має у конкурентів	Інтеграція між різними PDM та CAD принесе середнім та великим підприємствам приріст у швидкості та якості вироблення продукції.

За визначеними факторами конкурентоспроможності (таблиця 6.10) проведено аналіз сильних та слабких сторін стартап-проекту (таблиця 6.11).

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1- 20	Рейтинг товарів-конкурентів у порівнянні з даним продуктом						
			-3	-2	-1	0	1	2	3
1	Використання сучасних технологій	15				+			
2	Пропозиція можливостей яких не має у конкурентів	15					+		

Завершальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 6.11).

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: Можливість модифікації Швидкий економічний ефект Інтеграція між різними PDM та CAD	Слабкі сторони: Середній поріг входження на ринку, потрібна промо-кампанія.
Можливості: Вихід на нові ринки	Загрози: Обмежена кількість клієнтів Поява зовнішніх конкурентів Банкрутство

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний

оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 6.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 6.13).

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Зміна цільового сектору	30 %	3 місяці
2	Пошук нових потенційних споживачів	20 %	3 місяці
3	Орієнтація поточної моделі на ринок Enterprise	50 %	168 год

Після аналізу було обрано альтернативу №3.

6.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 6.14).

Таблиця 6.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
3	Потенційними споживачами є великий та середній бізнес (заводи)	висока	1-100	пасивна	Відсутність конкурентів в Україні
Які цільові групи обрано: великий та середній бізнес					

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентос- проможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
--	------------------------------	---	-------------------------------

Стратегія диференціації	Стратегія диференціації	Можливість налаштування системи під користувача	Стратегія спеціалізації
-------------------------	-------------------------	---	-------------------------

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 6.16).

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів	Чи буде компанія копіювати основні характеристики конкурента	Стратегія конкурентної поведінки
Так	Шукати нових споживачів, забирати існуючих у конкурентів	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 6.5), а також в залежності від обраної базової стратегії розвитку (таблиця 6.15) та стратегії конкурентної поведінки (таблиця 6.16) розроблено стратегію позиціонування (таблиця 6.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
	Доступна інформація про товари на всьому життєвому циклі розробки	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Споживачі – заводи по виготовленню технічних виробів.

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки компанії, яка визначає напрями роботи стартап-компанії на ринку.

6.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 6.18 підсумовано результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
----------------	-----------------------------------	---

Швидкий економічний ефект	Швидкий економічний ефект	Оптимізація з урахуванням багатьох факторів впливу
---------------------------	---------------------------	--

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання. Ці дані показано в таблиці 6.19.

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення для взаємодії PDM та CAD систем		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Розмір програмного забезпечення	Нм	Тх
	2. Інтерфейс користувача	Нм	Тх
	3. Модульність	Нм	Вр
	Якість: стандарти, нормативи, параметри тестування тощо		
	Пакування: відсутнє		
	Марка: PDM Browser		
III. Товар із підкріпленням	До продажу: відсутнє		
	Після продажу: персональна підтримка в обслуговуванні за додаткову платню.		
На ідею зареєстровано патент.			

В таблиці 6.20 проводиться визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар. Аналіз проведено експертним методом.

Таблиця 6.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
-------	--------------------------------	------------------------------	--	---

1	300000-500000	високий	високий	250000-300000
---	---------------	---------	---------	---------------

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 6.21)

Таблиця 6.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цілових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Визнання наявності потреби. Прийняття рішення про конкретні шляхи та джерела задоволення потреби	Вихід на ринок з новим товаром	Кількість модулів	Збільшення ринкової частки підприємства

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 6.22).

Таблиця 6.22. Концепція маркетингових комунікацій

Специфіка поведінки цілових клієнтів	Канали комунікацій, якими користуються цілові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення

Прямий маркетинг	особистий продаж; продаж за каталогами; та інші засоби ЗМІ (телевізійний маркетинг); Інтернет.	Доступність та об'єктивність інформації про фірму та товар Пресс-event Семінар Круглий стіл Виставки	Дослідження ринку, Виявлення потенційних клієнтів	Пропозиція
------------------	--	--	--	------------

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 6

Проведено аналіз програмного продукту у якості стартап проекту.

Використання продукту для інтеграції PDM та CAD систем дозволить значно поліпшити позиції виробника на ринку. Даний продукт здатний досить швидко продемонструвати економічний ефект для виробництва, що робить його більш привабливим для клієнтів.

Подальший розвиток проекту є перспективним, оскільки він знайде свою цільову аудиторію.

Висновки

В ході виконання магістерської дисертації встановлено, що важливим фактором функціонування будь-якого виробничого підприємства є вдале поєднання систем автоматизованого проектування (CAD) та контролю виробів (PDM). Від інтеграції PDM та CAD систем залежить якісний та швидкий контроль над розроблюваним виробом та його випуском у виробництво.

Було досліджено проблеми інтеграції між системами CAD та PDM, основною з яких є традиційний підхід інтеграції «один до одного». Для реалізації принципу «один до багатьох» було встановлено, що необхідно уніфіковано працювати з різними CAD-системами та інтегруватися з єдиною довільною PDM системою. Для цього було удосконалено підхід до автоматизації систем PLM за рахунок уніфікації інтерфейсу взаємодії між PDM та CAD системами.

Було створено програмне забезпечення для взаємодії PDM та CAD систем, яке забезпечує виконання колективного проектування нових виробів із паралельним використанням різних CAD систем та дозволяє уніфікувати принципи та підходи при розробці нових виробів незалежно від середовища автоматизованого проектування (CAD).

Програмне забезпечення створене з використанням засобів розробки Visual Studio, IntelliJ Idea, мов програмування C# та Java.

Для тестування програмного забезпечення було обрано інтегрувати CAD системи SolidWorks, PTC Creo та PDM систему IT-Enterprise.

Проаналізувавши можливості стартап проекту на основі цього програмного забезпечення було встановлено, що воно дозволить багатьом промисловим підприємствам пришвидшити розробку та випуск нової продукції на ринок.

Програмне забезпечення для взаємодії PDM та CAD систем впроваджене та використовується працівниками компанії ТОВ “ІТ-Софт”.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Орел Д.С. Програмне забезпечення для взаємодії PDM та CAD систем / Д.С. Орел, Л.О. Левченко // XVI міжнародна науково-практична конференція аспірантів, магістрантів, студентів «Сучасні проблеми наукового забезпечення енергетики», 24-27 квітня, 2018 року. – м. Київ, 2018. – С. 298.
- 2) Столбов В. Ю. Модель интеллектуальной системы управления производством / В. Ю. Столбов, С. А. Федосеев. // Проблемы управления. – 2006. – №5. – С. 86 –99.
- 3) Шапиро Дж. Моделирование цепи поставок. / Дж. Шапиро // Вестник Мурманского государственного технического университета. – 2006. – №2. — 720 с.
- 4) Ivanov D. & Sokolov B., (2010), Adaptive Supply Chain Management. — L.: Springer. — 200 p.
- 5) Юрасов А. В. Основы электронной коммерции / А. В. Юрасов // Горячая линия-Телеком, Москва. – 2007. – С. 124.
- 6) Gary Schneider, (2008), Electronic Commerce / Schneider Gary. — Course Technology. — 2008. — 330 p.
- 7) Втюрин В. А. Основы АСУ ТП. / В. А. Втюрин. — СПб.: Питер. — 2006. — 154 с.
- 8) Абросимова А. Технологии и люди. Сложности внедрения ERP-систем / Анна Абросимова // Управление человеческим потенциалом. – 2006. – №1. – С. 25–30.
- 9) Власов А. П. Анализ современных ERP-систем / А. П. Власов, П. Б. Сергей, Я. С. Борис. // Современные наукоемкие технологии. – 2009. – №2. – с. 50–54.
- 10) Верников Г. Г. Стандарт MRPII. Структура и основные принципы работы систем поддерживающих этот стандарт / Геннадий Геннадьевич Верников. – М.: Высшая школа, 1999. – 489 с.

- 11) Гаврилов Д. А. Управление производством на базе стандарта MRP / Д.А. Гаврилов. – СПб: Питер, 2002. – 416 с.
- 12) Лирин О. Д. ERP системы. Современное планирование и управление ресурсами предприятия. — СПб.: Питер – 2005. - 376 с.
- 13) Загидуллин Р. Управление машиностроительным производством с помощью систем MES, APS, ERP. / Загидуллин Р. Р. — Старый Оскол: ТНТ, 2011. — 372 с.
- 14) Куцевич Н.А. «SCADA-системы: взгляд изнутри». / Н.А. Куцевич, О.В. Синенко, Б.Е. Андреев. — Издательство «РТСофт». - 2004 г. – 231 с.
- 15) Петров И.В. Программируемые контроллеры: стандартные языки и приемы прикладного проектирования». / И.В. Петров. — Москва: «Солон-Пресс». - 2007 г. – 333 с.
- 16) Boyer, Stuart A. (2010). SCADA Supervisory Control and Data Acquisition. USA: ISA - International Society of Automation. p. 179.
- 17) Втюрин В.А. Автоматизированные системы управления технологическими процессами. Основы АСУТП. / В.А. Втюрин. — Москва: «Солон-Пресс». – 2006г. – 124 с.
- 18) Босинзон М. А. Современные системы ЧПУ и их эксплуатация для нач. проф. Образования / М. А. Босинзон. — Издательство Academia, 2009 г. – 231С.
- 19) Ловыгин А. А., Васильев А. В. Современный станок с ЧПУ и CAD/CAM система / А. А. Ловыгин, А.В. Васильев. — Издательство Эльф ИПР, 2006 г. – 240 с.
- 20) ГОСТ 34.003-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения»
- 21) ГОСТ 23501.101-87 «Системы автоматизированного проектирования. Основные положения»
- 22) Phiri, Michael. (1999) Information Technology in Construction Design. - London, p. 228
- 23) Щеглов Д. К. [Электронный ресурс] Методы и средства выбора системы управления проектными данными конструкций летательных аппаратов. - СПб.:

- ГНЦ РФ ЦНИИ «Электроприбор». – 2009. – Режим доступа до ресурсу:
<http://www.elektropribor.spb.ru/cnf/kmu10/text/s3-032.rar>
- 24) Козловский В. А. Производственный менеджмент / В. А. Козловский. - М.:Инфра-М., 2003. — 574 с.
 - 25) Голенищев Э. П. Информационное обеспечение систем управления : учеб. пособие / Э. П. Голенищев, И. В. Клименко. — Ростов н/Д. : Феникс, 2010. — 315 с.
 - 26) Коршунова Н.И. Информационные технологии в управлении предприятием. / Н.И. Коршунова // Вестник Мурманского государственного технического университета. – 2003. – №2. – С. 215–221.
 - 27) Кисіль Н. М. Класифікація інформаційних систем / Н. М. Кисіль, З. П. Гаталяк, Н. І. Горбаль [Електронний ресурс] – Режим доступу:http://base.dnsgb.com.ua/files/journal/Lisove-gospodarstvo-l-p-d-promyslovist/2004_29/242_Kysil_LG_29.pdf (назва з екрану).
 - 28) Новицкий Н. И. Организация и планирование производства / Н. И. Новицкий. – Минск: Новое знание, 2004. – 256 с.
 - 29) Отрасли Дассо Систем [Електронний ресурс] – Режим доступу до ресурсу:<https://www.3ds.com/ru/otrasli/>
 - 30) PTC Creo [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.ptc.com/ru/products/cad/creo/parametric>
 - 31) САПР Catia [Електронний ресурс] – Режим доступу до ресурсу:
<http://bourabai.ru/cm/catia.htm>
 - 32) NX CAD [Електронний ресурс] – Режим доступу до ресурсу:
https://borlas.ru/plm_nx_cad.html
 - 33) Современный рынок систем PDM [Електронний ресурс] – Режим доступу до ресурсу: <https://www.osp.ru/os/2000/09/178132/>.
 - 34) Орел Д.С. Розробка програмного забезпечення для взаємодії PDM та CAD систем / Д.С. Орел // Сталій розвиток — XXI століття: управління, технології, моделі. Дискусії 2018: колективна монографія / Міненко М.А., Бендюг В.І., Комариста Б.М. [та ін.]; НТУУ “Київський політехнічний інститут імені Ігоря

- Сікорського”; Національний університет “Києво-Могилянська академія”; Вища економіко-гуманітарна школа / за наук. ред. проф. Хлобистова Є.В. — Київ, 2018. — 639 с.
- 35) J. Gosling, B. Joy, G. Steele, G. Brachda, (1996), The Java Language Specification, 2nd Edition, Kinkle Edition . — 320 p.
- 36) Эффективное управление производством [Электронный ресурс] – Режим доступа до ресурсу: <http://www.it.ua/>
- 37) Балдин А. В. Информационные системы управления ресурсами / А. В. Балдин, О. Н. Григорович, А. А. Пичугин. // Научный вестник Московского государственного технического университета гражданской авиации. – 2005. – №92. – С. 53–60.
- 38) Голенищев Э. П. Информационное обеспечение систем управления : учеб. пособие / Э. П. Голенищев, И. В. Клименко. – Ростов н/Д. : Феникс, 2010. – 315 с.
- 39) Бьярн Страуструп. Язык программирования C++ = The C++ Programming Language. — М.—СПб.: Бином, Невский диалект, 2008. — 1104 с.
- 40) Ахо Альфред В. Структуры данных и алгоритмы / Ахо, Альфред В., Джон Э. Хопкрофт, and Джеффри Д. Ульман. — М. : Издательский дом "Вильямс", 2000. - 384 с.
- 41) Язык C# (инструкции по C#) [Электронный ресурс] – Режим доступа: [http://msdn.microsoft.com/ru-ru/library/ms186214\(v=vs.90\).aspx](http://msdn.microsoft.com/ru-ru/library/ms186214(v=vs.90).aspx)
- 42) Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4.0. — М.: ООО «И.Д. Вильямс», 2011. — 1392 с.
- 43) Ник Рендольф Visual Studio 2010 для профессионалов. / Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. — М.: «Диалектика», 2011. — 1184 с.
- 44) Алекс Макки. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов / Макки Алекс. — М.: «Вильямс», 2010. — 416 с.
- 45) Visual C# 2008: базовый курс. / [Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, и др.]. — М.: «Вильямс» - 2009. — 1216 с.

- 46) Пауэрс Л. Microsoft Visual Studio 2008 Unleashed by Lars Powers and Mike Snell / Л. Пауэрс М. Снелл . — С.: «БХВ-Петербург», 2008. — 1200 с.
- 47) Майо Д. Самоучитель Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide) / Д.Майо. — С.: «БХВ-Петербург», 2010. — 464 с.
- 48) Давыдов С. IntelliJ IDEA. Профессиональное программирование на Java / С. Давыдов. А. Ефимов .— СПб.: БХВ, 2005. — 800 с.
- 49) Sikos L. Web Standards: Mastering HTML5, CSS3, and XML. Apress, 2011. — 524 с.
- 50) Kay M. (2008) XSLT 2.0 and XPath 2.0 Programmer's Reference. — Wrox. — 1368 с.
- 51) Аведьян А. Б., SolidWorks API – универсальная платформа для интеграции инженерных и бизнес-приложений / А. Б. Аведьян, Е. Е. Викентьев // САПР и графика. – 2006.- № 6. С.32–40.