

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

**Навчально-науковий інститут телекомунікаційних систем
Кафедра інформаційних технологій в телекомунікаціях**

«На правах рукопису»

УДК 004.942

«До захисту допущено»

Завідувач кафедри

_____ Марія СКУЛИШ

“ ____ ” _____ 2025 р.

Магістерська дисертація

зі спеціальності 172 Електронні комунікації та радіотехніка

на тему: **Онтологічна модель оптимізації технологічних процесів**

Виконав: студент VI курсу, групи ЦІ-41мп

Перець Данило Геннадійович _____
(підпис)

Науковий керівник: доцент кафедри ІТТ НН ІТС,

кандидат технічних наук, доцент

Новогрудська Ріна Леонідівна _____
(підпис)

Рецензент: професор кафедри ЕКІР НН ІТС,

доктор технічних наук, доцент

Мошинська Аліна Валентинівна _____
(підпис)

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

Студент _____

Київ – 2025

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Навчально науковий інститут телекомунікаційних систем

Кафедра інформаційних технологій в телекомунікаціях

Рівень вищої освіти – другий магістерський за освітньо-професійною
 програмою Інформаційно-комунікаційні технології
 Спеціальність 172 Електронні комунікації та радіотехніка

ЗАТВЕРДЖУЮ
 Завідувач кафедри
 _____ Марія СКУЛИШ
 “ ____ ” _____ 2025 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Перцю Данилу Геннадійовичу

1. Тема дисертації **«Онтологічна модель оптимізації технологічних процесів»**, науковий керівник Новогрудська Ріна Леонідівна, доцент кафедри ІТТ НН ІТС, кандидат технічних наук, доцент затверджені наказом по університету від «03» листопада 2025 р. № 4472-с

2. Строк подання студентом дисертації «16» грудня 2025 р.

3. Об'єкт дослідження

Технологічні процеси виробництва складних технічних систем

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою)

Методи онтологічного моделювання та логічного аналізу технологічних процесів, а також шляхи формалізації дієвості цих процесів. Вихідні дані – розроблена онтологічна модель оптимізації технологічних процесів.

5. Перелік завдань, які потрібно розробити

1. Провести аналіз сучасних методів онтологічного інженірингу.

2. Виявити базові сутності технологічного процесу

3. Побудувати базову онтологічну модель технологічного процесу.
4. Розробити модуль оцінки ефективності в межах онтології.
5. Реалізувати зв'язки між параметрами процесів у Protégé.
6. Перелік графічного (ілюстративного) матеріалу
 1. Вступ
 2. Актуальність
 3. Мета
 4. Об'єкт та предмет дослідження
 5. Задачі дослідження
 6. Порівняння існуючих моделей
 7. Розроблений онтологічна модель
 8. Оцінка ефективності та результати проекту
 9. Висновки та рекомендації
7. Дата видачі завдання 28 жовтня 2025 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Дослідження та вивчення отриманого завдання	28.10.25	Виконано
2	Аналіз існуючих освітніх платформ та рішень	29.10.25-31.10.25	Виконано
3	Вибір технологій та інструментів для розробки веб-додатку	01.11.25 – 03.11.25	Виконано
4	Розробка архітектури системи	04.11.25 – 10.11.25	Виконано
5	Створення прототипу веб-додатку	11.11.25 – 17.11.25	Виконано
6	Розробка основних компонентів системи	18.11.25 – 24.11.25	Виконано

7	Інтеграція та впровадження додаткових функцій	25.11.25 – 26.11.25	Виконано
8	Виправлення помилок у розробленому веб-додатку	27.11.25 – 28.11.25	Виконано
9	Проведення оцінки розробленого веб-додатку	29.11.25 – 30.11.25	Виконано
10	Аналіз результатів, тестування	01.12.25 – 13.12.25	Виконано

Студент

(підпис)

Данило ПЕРЕЦЬ

(ім'я, прізвище)

Науковий керівник дисертації

(підпис)

Ріна НОВОГРУДСЬКА

(ім'я, прізвище)

РЕФЕРАТ

Тема магістерської дисертації «Онтологічна модель оптимізації технологічних процесів». Робота містить 108 сторінок тексту, 22 рисунків, 9 таблиць, використано 44 літературних джерел.

Актуальність теми дослідження полягає в необхідності створення таких моделей технологічних процесів, що можуть універсально застосовуватися у різних виробничих та сервісних доменах, забезпечуючи можливість оцінювання ефективності та порівняння альтернативних варіантів виконання операцій.

Мета роботи: Підвищити ефективність виконання технологічних процесів за рахунок онтологічного моделювання.

Об'єктом дослідження є технологічні процеси виробництва складних технічних систем

Предметом дослідження є методи онтологічного моделювання та логічного аналізу технологічних процесів, а також шляхи формалізації дієвості цих процесів.

Методи дослідження: Аналіз літератури та освітніх платформ, проектування моделі, експериментальна перевірка.

Наукова новизна отриманих результатів: праці полягає у формуванні універсальної онтологічної структури, що може бути використана до широкого спектру технологічних послідовностей, а також у розробці об'єднаного методу оцінки дієвості, базованого на формальних логічних залежностях.

Практичне значення отриманих результатів: Створена онтологія може бути використана як:

- компонент у системах управління виробництвом;
- інструмент підтримки прийняття рішень при оптимізації маршрутів і процесів;
- навчальний матеріал для демонстрації методів формалізації та логічного аналізу.

Ключові слова: ОНТОЛОГІЧНА МОДЕЛЬ, ТЕХНОЛОГІЧНИЙ ПРОЦЕС, ОНТОЛОГІЧНИЙ ІНЖИНІРИНГ, ФОРМАЛІЗАЦІЯ ЗНАНЬ, СЕМАНТИЧНЕ МОДЕЛЮВАННЯ, ЛОГІЧНИЙ ВИВІД, СИСТЕМА

ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ОПТИМІЗАЦІЯ ПРОЦЕСІВ СИСТЕМИ,
ЕФЕКТИВНІСТЬ, СТАРТАП-ПРОЄКТ.

ABSTRACT

The topic of the master's thesis is “Ontological Model of Technological Process Optimization.” The work contains 108 pages, 22 figures, 9 tables, 44 sources were used.

The relevance of the research topic lies in the need to create models of technological processes that can be universally applied in various production and service domains, providing the ability to evaluate the effectiveness and compare alternative options for performing operations.

The goal of the work: To improve the efficiency of technological processes through ontological modeling.

The object of research is the technological processes of manufacturing complex technical systems.

The subject of research is the methods of ontological modeling and logical analysis of technological processes, as well as ways to formalize the effectiveness of these processes.

Research methods: Analysis of literature and educational platforms, model design, experimental verification.

Scientific novelty of the results: The work consists in the formation of a universal ontological structure that can be used for a wide range of technological sequences, as well as in the development of a unified method for evaluating effectiveness based on formal logical dependencies.

Practical significance of the results: The created ontology can be used as: a component in production management systems, a decision support tool for optimizing routes and processes, educational material for demonstrating methods of formalization and logical analysis.

Keywords: ONTOLOGICAL MODEL, TECHNOLOGICAL PROCESS, ONTOLOGICAL ENGINEERING, KNOWLEDGE FORMALIZATION, SEMANTIC MODELING, LOGICAL INFERENCE, DECISION SUPPORT SYSTEM, PROCESS OPTIMIZATION, OWL ONTOLOGY, PROTÉGÉ, STARTUP PROJECT.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	13
ВСТУП.....	13
РОЗДІЛ 1.....	17
ТЕОРЕТИЧНІ ОСНОВИ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ	17
1.1 Онтологічний підхід у моделюванні складних систем.....	17
1.1.1. Зміст та значення онтологій у сучасних інформаційних системах..	17
1.1.2. Причини переходу до онтологічного моделювання	18
1.1.3. Онтології як засіб інженерії знань.....	19
1.1.4. Основні ознаки онтологічних моделей	19
1.1.5. Роль онтологій у задачах по оптимізації технологічних процесів...	20
1.1.6. Підсумки	20
1.2 Моделювання технологічних процесів: особливості, структура та вимоги до формалізації.....	21
1.2.1. Сутність технологічного процесу як об'єкта моделювання.....	21
1.2.2. Основні елементи технологічного процесу	22
1.2.3. Вимоги до моделювання технологічних процесів.....	23
1.2.4. Моделі представлення процесів та їх обмеження	25
1.2.5. Потреба в універсальній онтологічній моделі технологічного процесу	25
1.2.6. Узагальнення	26
1.3 Сучасні методи онтологічного інженірингу	26
1.4 Мови опису онтологій та інструменти їх реалізації	29
1.4.1. OWL як базис для формалізованого подання онтологій	29
1.4.2. SWRL для побудови правил логічного виводу.....	29
1.4.3. Protégé як основний інструмент роботи з онтологіями	30

1.5. Концептуальні основи побудови універсальної онтологічної моделі технологічного процесу	31
1.5.1. Базові сутності та їх роль у моделі	31
1.5.2. Логічні зв'язки між елементами технологічного процесу	32
1.5.3. Формалізація критеріїв ефективності та їх інтеграція в онтологію	33
Висновки	33
РОЗДІЛ 2.....	36
РОЗРОБЛЕННЯ УНІВЕРСАЛЬНОЇ ОНТОЛОГІЧНОЇ МОДЕЛІ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ	36
2.1. Аналіз предметної області та формування вимог до моделі.....	36
2.2. Визначення базових сутностей, класів та властивостей онтології	37
2.3. Розроблення структурної моделі технологічного процесу	41
2.3.1. Основна логіка структурної моделі	42
2.3.2. Структурна схема моделі (текстове подання).....	43
2.3.3. Остаточна структурна модель (опис словами)	48
2.4. Формалізація логічних зв'язків між структурними елементами технологічного процесу	49
2.4.1. Основні групи логічних зв'язків.....	49
2.4.2. Формалізація властивостей у Protégé (OWL-подання).....	52
2.4.3. Роль логічних зв'язків у reasoning	52
2.4.4. Приклад формалізації на основі процесу 3D-друку	52
2.4.5. Логічні правила для reasoning	53
2.5. Створення модуля оцінки ефективності	54
2.5.1. Основні концепції модуля.....	54
2.5.2. Класи та концепції модуля ефективності.....	55
2.5.3. Формалізація властивостей модуля	55
2.5.4. SWRL-правила для автоматичного оцінювання	56
2.5.5. Підсумкова роль модуля ефективності	57
2.6. Реалізація онтології в Protégé	57
2.6.2. Реалізація об'єктних властивостей.....	58

	10
2.6.3. Створення атрибутів (data properties)	59
2.6.4. Формування екземплярів (individuals).....	59
2.6.5. Додавання логічних обмежень та аксіом	60
2.6.6. Перевірка моделі за допомогою reasoner	61
2.7. Приклад застосування універсальної онтологічної моделі для технологічного процесу 3D-друку	62
2.7.1. Етапи технологічного процесу 3D-друку.....	63
2.7.2. Формальне представлення етапів у межах онтології	63
2.7.3. Формальне представлення етапів у межах онтології	64
2.7.4. Reasoning: автоматичний висновок системи.....	66
РОЗДІЛ 3.....	67
3.1. Вибір інструментальних засобів.....	67
3.1.1. Protégé як основне середовище моделювання	67
3.1.2. Використання OWL (Web Ontology Language)	68
3.1.3. Reasoner як засіб перевірки логічної узгодженості	68
3.1.4. Використання графічних інструментів для візуалізації	69
3.1.5. Додаткові допоміжні засоби	70
3.2. Структура реалізації онтологічної моделі у Protégé.....	71
3.3. Формування класів, властивостей та екземплярів	72
3.3.1. Створення класів	72
3.3.2. Створення об'єктних властивостей (Object Properties)	73
3.3.3. Створення дата-властивостей (Data Properties).....	75
3.3.4. Формування екземплярів (Individuals)	75
3.4. Формування онтологічної моделі технологічного процесу 3D-друку	77
3.4.1. Формування ієрархії класів.....	78
3.4.2. Створення об'єктних властивостей	79
3.4.3. Створення дата-властивостей	80
3.4.4. Додавання логічних аксіом	80
3.4.5. Перевірка моделі Reasoner-ом	81
3.5. Приклади використання моделі у складних технічних системах	82

3.5.1. Виробництво корпусу та вузлів БПЛА.....	82
3.5.2. Технологічний маршрут збирання БПЛА	83
3.5.3. Використання моделі у навчальному процесі та цифрових двійниках	84
3.6. Аналіз отриманих результатів та оцінка впливу онтологічної моделі	85
3.6.1. Формалізація процесу та підвищення прозорості.....	85
3.6.2. Виявлення взаємозалежностей параметрів	85
3.6.3. Покращення процесу завдяки моделі (якісні зміни).....	86
3.6.4. Загальні висновки щодо ефективності моделі	86
РОЗДІЛ 4.....	88
РОЗРОБКА СТАРТАП-ПРОЄКТУ НА ОСНОВІ ОНТОЛОГІЧНОЇ МОДЕЛІ ОПТИМІЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ	88
4.1. Ідея та загальна концепція стартап-проєкту	88
4.2. Опис продукту та функціональних можливостей	90
4.2.1. Загальна характеристика продукту.....	90
4.2.2. Основні функціональні модулі	91
4.2.3. Користувачі продукту	92
4.2.4. Формат використання продукту	92
4.3 Аналіз конкурентних рішень та переваги стартапу.....	93
4.3.1 Огляд існуючих конкурентних рішень.....	93
4.3.2 Місце стартап-проєкту серед конкурентів	94
4.3.3 Порівняльна характеристика рішень	94
4.3.4 Конкурентні переваги стартапу	94
4.3.5 Узагальнений висновок	95
4.4 Організаційна структура стартап-проєкту	95
4.5. Фінансове обґрунтування стартап-проєкту.....	98
4.5.1. Структура початкових витрат	98
4.5.2. Поточні витрати на підтримку	99
4.5.3. Модель монетизації	99
4.5.5. Оцінка терміну окупності	100

4.5.6. Висновок щодо фінансової доцільності	100
4.6 Оцінка ризиків та перспектив розвитку стартап-проєкту	100
4.6.1 Аналіз основних ризиків стартап-проєкту	101
4.6.2 Перспективи розвитку стартап-проєкту	102
4.6.3 Загальна оцінка життєздатності проєкту.....	102
ЗАГАЛЬНІ ВИСНОВКИ.....	104
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	106

ПЕРЕЛІК СКОРОЧЕНЬ

AI	Artificial Intelligence — штучний інтелект
API	Application Programming Interface — програмний інтерфейс прикладного програмування
DSS	Decision Support System — система підтримки прийняття рішень
FDM	Fused Deposition Modeling — технологія пошарового наплавлення матеріалу
IT	Information Technology — інформаційні технології
ML	Machine Learning — машинне навчання
OWL	Web Ontology Language — мова опису онтологій
RDF	Resource Description Framework — модель подання даних у вигляді графів
RDFS	RDF Schema — схема опису ресурсів RDF
SPARQL	SPARQL Protocol and RDF Query Language — мова запитів до RDF-даних
SWRL	Semantic Web Rule Language — мова правил семантичної мережі
UML	Unified Modeling Language — уніфікована мова моделювання
URI	Uniform Resource Identifier — уніфікований ідентифікатор ресурсу
OntoGraf	Інструмент графічної візуалізації онтологій у середовищі Protégé
Protégé	Програмне середовище для розроблення та аналізу онтологій
Reasoner	Механізм логічного виводу в онтологічних системах
HermiT	OWL-reasoner для логічного аналізу онтологій

ВСТУП

Сучасні технологічні системи характеризуються зростаючою складністю, багаторівневими структурами та потребою постійного підвищення дієвості. Підприємства різних сфер дедалі частіше стикаються з потребою формалізованого опису процесів, передбачення їхньої поведінки, аналізу залежностей між параметрами та вибору раціональних рішень для оптимізації. У зв'язку з цим усе більшої актуальності набувають методи, що забезпечують не лише структурний опис процесу, а й можливість повноцінної семантичної інтерпретації його елементів[1].

У цьому контексті онтологічне моделювання розглядається як один із провідних підходів до репрезентації знань. Онтології дозволяють формувати формальні моделі предметних областей із чітко визначеними об'єктами, властивостями та відношеннями. На відміну від класичних методів моделювання, онтологічний підхід забезпечує можливість логічного виведення, автоматизованого аналізу консистентності та гнучкого розширення моделі без порушення її структури. Це робить онтології потужним інструментом для побудови інтелектуальних систем підтримки прийняття рішень, цифрових двійників процесів та платформ оптимізації[2].

Актуальність теми дослідження полягає в необхідності створення таких моделей технологічних процесів, що можуть універсально застосовуватися у різних виробничих та сервісних доменах, забезпечуючи можливість оцінювання ефективності та порівняння альтернативних варіантів виконання операцій. Більшість сучасних систем оптимізації орієнтовані на вузькі задачі чи специфічні сценарії, що ускладнює їх масштабування на інші типи процесів. Тому важливо розробити універсальну онтологічну модель, здатну відображати будь-які технологічні ланцюги з необхідною деталізацією та підтримкою логічного аналізу.

Попри наявність значної кількості методів опису процесів, залишається низка невирішених проблем. По-перше, відсутні єдині загальні структури, які б однаково добре підходили для різних галузей — від виготовлення продукції до

організації сервісних чи управлінських процедур. По-друге, традиційні схеми моделювання не враховують повноцінно причинно-наслідкові залежності, що обмежує їх придатність для оптимізації. По-третє, у багатьох існуючих моделях не передбачено оцінювання ефективності на основі формальних критеріїв, які могли б бути опрацьовані логічними засобами.

Таким чином, наукова задача створення онтологічної моделі оптимізації технологічних процесів є важливою і своєчасною. Розроблення універсального онтологічного підходу дозволить формалізувати опис процесів, забезпечити їх логічний аналіз, здійснювати структурні порівняння та автоматично виводити показники ефективності та автоматичне видобування показників ефективності.

Метою роботи є підвищення ефективності виконання технологічних процесів за рахунок створення загальної онтологічної схеми технологічного процесу з інструментами вимірювання дієвості та смислового розбору зв'язків між ознаками.

Основні задачі, які необхідно виконати для досягнення визначеної мети належить наведено нижче:

- Здійснити огляд поточних засобів онтологічного моделювання та визначити їхню слушність для зображення виробничих операцій.
- Скласти універсальну концептуальну схему технологічної операції з визначенням ключових складових та їхніх сполучень.
- Сформувати блок оцінки дієвості, що дасть змогу формалізувати критерії найкращого варіанту та узгоджувати альтернативи.
- Реалізувати логічні залежності між параметрами у середовищі Protégé з використанням OWL-онтологій та SWRL-правил.
- Здійснити перевірку онтології, підтвердити її логічну цілісність та виконати розбір отриманих результатів.

Об'єктом дослідження є універсальні технологічні процеси як структурні системи операцій, що мають параметри, ресурси та результати.

Предметом дослідження є методи онтологічного моделювання та логічного аналізу технологічних процесів, а також шляхи формалізації дієвості цих процесів.

Наукова новизна праці полягає у формуванні універсальної онтологічної структури, що може бути використана до широкого спектру технологічних послідовностей, а також у розробці об'єднаного методу оцінки дієвості, базованого на формальних логічних залежностях.

Практична цінність отриманих результатів полягає у спроможності застосування створеної онтології як фундаменту для інтелектуальних систем аналізу й удосконалення процесів, моделювання альтернативних сценаріїв, прогнозування дієвості та забезпечення управлінських рішень.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

1.1 Онтологічний підхід у моделюванні складних систем

Сучасні технологічні системи, незалежно від сфери їхнього застосування, відповідають ключовим ознакам комплексних систем: вони багатокomпонентні, містять численні взаємозв'язані складові, демонструють ієрархічну структуру та динамічну поведінку[43]. За подібних обставин результативне моделювання процесів нездійсненне без використання методів, які забезпечують не лише структурний опис, а й семантичну інтерпретацію, логічну узгодженість та можливість автоматизованого аналізу.[3] Одним із засобів, котрий найкраще відповідає цим вимогам, є онтологічне моделювання, що походить із галузі штучного інтелекту та семантичних технологій.

1.1.1. Зміст та значення онтологій у сучасних інформаційних системах

В основу онтологічного моделювання покладено поняття онтології — формальної, машиночитної та логічно узгодженої моделі знань про певну предметну область. Онтологія визначає:

- 1) сутності (класи) предметної області;
- 2) властивості (атрибути) цих сутностей;
- 3) логічні зв'язки між ними;
- 4) обмеження, що керують поведінкою елементів моделі;
- 5) правила виведення, які уможливають автоматизувати аналіз

та прогнозування .

У класичному визначенні Т. Грубера (1993), онтологія є «явною формальною специфікацією концептуалізації». Це означає, що вона не просто перелічує об'єкти, а структурує знання у такій формі, щоб вони могли бути опрацьовані машинними алгоритмами логічного виведення[4].

Онтології стали головним інструментом там, де необхідно:

- інтегрувати відомості з різних джерел;
- відображати складні процеси з великою кількістю параметрів;

- забезпечити перевірку сумісності моделей;
- підтримувати інтелектуальні алгоритми аналізу.

1.1.2. Причини переходу до онтологічного моделювання

Традиційні методи опису процесів (схеми, реляційні структури, ієрархії) часто не відповідають вимогам сучасних систем. Основними недоліками класичних підходів є:

Недостатня семантика.

Схеми чи таблиці не дають змоги машині «осягнути» зміст зв'язків. Онтологія ж містить логічні залежності, які «*reasoner*» може аналізувати автоматично.

Складність доповнення.

Додавання нових сутностей чи параметрів у традиційні моделі нерідко спричиняє необхідність переробляти усю структуру.

Онтології ж підтримують поступове зростання[36].

Неможливість логічного виведення.

Реляційні моделі забезпечують збереження інформації, але не дозволяють робити висновки.

Онтології дозволяють автоматично:

- виявляти протиріччя,
- зіставляти альтернативи,
- знаходити залежності,
- оцінювати результативність.

Низький рівень машиночитності.

У сучасних підприємствах критично важливою є автоматизація, інтеграція систем та цифрові двійники. Це можливо лише за наявності формальних моделей[5].

Таким чином, онтологічний підхід надає формалізацію, семантичну точність, сумісність та інтелектуальність моделі — властивості, потрібні для складних технологічних процесів.

1.1.3. Онтології як засіб інженерії знань

Онтології сьогодні відіграють ключову функцію у дисципліні інженерії знань. Їх можна розглядати як моделі для **формального опису знань**, системи **класифікації об'єктів**, логічні конструкції для **аналізу залежностей**, основу для **цифрових двійників процесів**, **інтелектуальні шари** над звичайними базами даних.

Вони застосовуються у таких системах:

- системи підтримки прийняття рішень;
- експертні системи;
- аналітичні платформи;
- новітні MES/ERP системи;
- моделі упорядкування ресурсів та процесів.

Онтологічний підхід дає змогу створити універсальну модель, яка є придатною для будь-яких процесів із чіткими параметрами.

1.1.4. Основні ознаки онтологічних моделей

Онтології для моделювання технологічних процесів мають володіти такими властивостями:

- 1) Формальність
 - чітка логічна семантика;
 - можливість перевірки reasoner'ом.
- 2) Структурованість
 - визначені класи, підкласи, властивості, ролі.
- 3) Інтероперабельність
 - сумісність з іншими моделями та системами.
- 4) Масштабованість
 - здатність до розширення без зміни основної структури.
- 5) Однозначність тлумачення
 - усі елементи моделі мають єдині стандартизовані визначення.
- 6) Логічна завершеність
 - здатність до формального аналізу поведінки моделі.

7) Універсальність

- здатність застосування до різних доменів без зміни базової структури.

Усе це робить онтології одним з найбільш дієвих способів опису процесів, у яких є взаємодія параметрів, причинно-наслідкові зв'язки та критерії ефективності.

1.1.5. Роль онтологій у задачах по оптимізації технологічних процесів

Оптимізація вимагає:

- точного розуміння структури процесу;
- визначення параметрів та обмежень;
- формального опису залежностей;
- здатності порівнювати альтернативи;
- автоматичного аналізу ефективності.

Саме онтології забезпечують:

а) формальні критерії ефективності

(час, продуктивність, витрати ресурсів, якість таке інше)

б) логічні зв'язки між параметрами

(наприклад: зростання тривалості → веде до витрат)

в) можливість reasoning-аналізу

Reasoner може:

- виявити неконсистентні параметри,
- автоматично обрахувати логічно виведені значення,
- знайти найкращі варіанти на підставі заданих правил.

г) правила SWRL для автоматичного виведення ефективності

(це буде у розділі 2–3)

Таким чином, онтологічний підхід стає інтелектуальною платформою для оптимізації, а не просто способом опису[6].

1.1.6. Підсумки

Онтологічний підхід у моделюванні складних систем:

- дозволяє формувати машиночитну структуру знань про процес;
- забезпечує багаторівневу деталізацію;
- підтримує автоматичне логічне виведення;
- створює підґрунтя для оцінювання ефективності;
- дає змогу оптимізувати технологічні процеси завдяки коректній логічній структурі.

Тому використання онтологій стає ключовим елементом сучасних інтелектуальних систем моделювання.

1.2 Моделювання технологічних процесів: особливості, структура та вимоги до формалізації

Технологічні процеси складають основу функціонування більшості виробничих, сервісних, аналітичних та управлінських систем. Незалежно від сфери застосування, вони мають спільні риси: послідовність завдань, використання ресурсів, трансформацію вхідних даних у вихідні результати та підпорядкування визначеним обмеженням[42]. У контексті інтелектуального аналізу та оптимізації технологічні процеси мусять бути формалізовані достатньо точно, щоб модель могла відтворювати не лише структуру, але й логіку процесу, його параметри та залежності.

Онтологічний підхід ставить інші вимоги до моделювання порівняно з традиційними схемами. Тут важливо не лише описати елементи процесу, а й надати їм чітку семантику, визначити ролі, властивості, ієрархії та логічні правила, які зможе обробляти розв'язувач (reasoner). Тому доцільно розглянути технологічний процес як об'єкт інженерії знань.

1.2.1. Сутність технологічного процесу як об'єкта моделювання

Технологічний процес у широкому розумінні — це впорядкований набір дій, спрямованих на досягнення певної мети шляхом впливу на ресурси. Кожен процес має структурні та поведінкові ознаки.

Структурні ознаки:

- склад процесу (операції, підоперації);

- ресурси, які беруть участь;
- параметри, що окреслюють якість або ефективність;
- об'єкти, які трансформуються;
- результати виконання операцій;
- ієрархічна структура підпроцесів.

Поведінкові характеристики:

- послідовність операцій;
- можливість паралельного виконання;
- залежності між операціями;
- контрольні точки та умови переходів;
- динамічна зміна параметрів.

Універсальність технологічного процесу полягає в тому, що його структура повторюється в будь-якій предметній області — від виробництва матеріальних об'єктів до інформаційних та управлінських систем.[7] Це дозволяє будувати узагальнені онтологічні моделі, придатні для багатьох сфер.

1.2.2. Основні елементи технологічного процесу

Аби збудувати загальну онтологію, потрібно виділити основні складові, які присутні у кожному процесі.

1. Операція

Базова дія, що виконує конкретну функцію.

Дії можуть бути:

- прості (атомарні);
- складні (композиційні);
- послідовні;
- умовні.

Кожна операція має показники (час, витрати, ресурси) і впливає на стан об'єкта.

2. Засіб

Будь-який предмет, потрібний для здійснення дії.

Види ресурсів:

- матеріальні;
- енергетичні;
- інформаційні;
- людські;
- інфраструктурні.

Ресурси можуть бути власними чи зовнішніми, поновлюваними чи вичерпними.

3. Характеристика

Кількісний чи якісний вимір процесу чи операції.

Характеристики розділяються на:

- технологічні (температура, навантаження, точність);
- фінансові (витрати, ціна);
- часові (тривалість, затримки);
- якісні (частка браку, надійність);
- показники ефективності.

4. Сутність змін

Початковий елемент процесу, що змінюється на виході.

Універсальна модель дає змогу бачити сутність абстрактно — як інформацію, матеріал, стан або подію.

5. Результат

Стан об'єкта після завершення операцій або усього процесу.

Результат має власні ознаки, що можуть бути критеріями ефективності.

1.2.3. Вимоги до моделювання технологічних процесів

В межах онтологічного підходу моделювання технологічних процесів має відповідати низці вимог:

1. Універсальність

Модель повинна бути придатною для будь-яких процесів: матеріальних, інформаційних, сервісних.

Це означає:

- абстрактність сутностей,

- гнучкість у розширенні класів,
- універсальні визначення операцій і параметрів.

2. Формальність

Онтологія повинна бути описана мовою, яку підтримують reasoner'и (OWL-DL).

Формальність забезпечує:

- перевірку логічної узгодженості,
- виявлення прорахунків у моделі,
- можливість машинного аналізу.

3. Чіткість семантики

Кожен клас, властивість і зв'язок мають мати однозначне визначення.

Семантична точність є вирішальною для уникнення неоднозначностей.

4. Підтримка логічного виведення

Модель повинна дозволяти:

- аналіз залежностей між параметрами,
- виявлення причинно-наслідкових зв'язків,
- обчислення ефективності,
- визначення конфліктів та обмежень.

Ця властивість є підґрунтям для оптимізації процесів.

5. Масштабованість і розширюваність

В онтологію можна додати нові:

- операції,
- типи ресурсів,
- параметри,
- логічні правила

— без руйнування структури.

6. Інтеграція з іншими системами

Онтологія повинна мати змогу взаємодіяти з:

- MES/MRP системами,
- аналітичними платформами,

- симуляторами процесів,
- системами підтримки рішень.

OWL та RDF надають такі можливості.

1.2.4. Моделі представлення процесів та їх обмеження

Перш ніж перейти до онтології, варто розглянути, чому звичні моделі не є достатніми:

1. Блок-схеми

Добре демонструють послідовність операцій, проте не відображають змісту (семантики), не підтримують логічних застережень, не допускають умовиводів (reasoning).

2. UML-діаграми

Гарні для програмних систем, але не призначені для оцінки продуктивності, не мають формальної семантики на рівні DL, описують структуру (але не причинність)

3. Табличні моделі (Реляційні)

Спроможні зберігати інформацію, проте не підтримують логічних виведень, складно зображати заплутані зв'язки.

4. Фреймові структури

Мають ліпший зміст, однак обмежені у вираженні логічних вимог, не мають усталеного механізму для висновків.

Підсумок:

Онтології вирішують більшість вад традиційних моделей[31].

1.2.5. Потреба в універсальній онтологічній моделі технологічного процесу

Сучасні системи вимагають моделі, яка:

- не залежить від певного домену
- спроможна описувати як прості, так і комплексні процеси;
- володіє параметричними характеристиками, які можна

піддавати оцінці;

- може бути застосована для оптимізації та аналізу ефективності.

Загальна модель повинна містити:

- основну структуру процесу;
- загальні властивості операцій;
- узагальнені типи ресурсів;
- логічні зв'язки;
- критерії результативності;
- механізми міркування (reasoning).

Усе це гарантує можливість повторного застосування моделі без корекції її основи[8].

1.2.6. Узагальнення

Насамкінець, технологічний процес — це непростий об'єкт, який вимагає детальної, формальної, семантичної та логічної моделі для подальшого аналізу та вдосконалення. Онтологічний підхід дає змогу:

- збудувати універсальну структуру;
- формалізувати знання;
- інтегрувати параметри, ресурси та операції;
- здійснювати логічний аналіз залежностей;
- формувати основу для оцінки ефективності.

Розділ 1.2 завершує основну теоретичну підготовку до створення універсальної онтології.

1.3 Сучасні методи онтологічного інженірингу

Процедура створення онтології як формальної моделі знань вимагає чітко окресленої методологічної основи. Поза рамками випадкових підходів онтологічний інженіринг сформувався як окрема дисципліна, що поєднує принципи інженерії програмного забезпечення, логічного розбору, інформаційного моделювання та когнітивних наук[39]. На противагу типовому опису систем, який часто спрямований на практику і не передбачає формальної перевірки, онтологічне моделювання вимагає значно більшої чіткості та

структурної узгодженості. Тому протягом останніх десятиліть було створено кілька методик, які допомагають впорядкувати процес проєктування, випробування та підтримки онтологій.

Однією з найбільш відомих і вагомих методик є **METHONTOLOGY**, розроблена науковою групою університету Політехніки Мадрида[9]. Її відмінністю є комплексний підхід до розробки онтологій, що вміщує весь життєвий цикл — від визначення мети до обслуговування готової моделі. METHONTOLOGY передбачає етапи концептуалізації, формалізації, втілення, описування, а також перевірки й валідації. Такий підхід дає змогу не лише структурувати процес розробки, а й гарантувати логічну правильність створеної моделі. Для моделювання технологічних процесів цей підхід є особливо цінним через свою уважність до формулювання понять та побудови структурних зв'язків, що є вирішальними для об'єктів зі складною параметрикою.

Іншим важливим підходом є методика **Ontology Development 101**, яка набула поширення завдяки своїй простоті й прикладному спрямуванню. Вона дає рекомендації щодо того, як розпочати розробку онтології, на що звернути увагу при доборі ключових термінів, як формувати ієрархію класів та визначати властивості. Хоча цей підхід менш формалізований, ніж METHONTOLOGY, він чудово підходить для швидкого створення початкових або навчальних моделей. У межах цієї роботи, **Ontology Development 101** пропонує зручну схему для формування первісної структури універсальної моделі процесів — особливо на етапі виокремлення сутностей, їхніх характеристик і взаємозв'язків[10].

Ще однією сучасною методологією, яка заслуговує на увагу, є **NeOn Framework**. На відміну від попередніх, цей метод був розроблений з урахуванням того, що онтології дедалі частіше стають складними, багатoproфільними, модульними та динамічними. Не завжди спорудження моделі «від початку» є найкращим вибором — інколи доводиться інтегрувати сторонні онтології, поєднувати кілька онтологічних компонентів або адаптувати вже наявні моделі під інші завдання. **NeOn** пропонує низку сценаріїв розробки, які включають повторне використання, реструктуризацію, злиття та адаптацію

онтологій. Це надзвичайно важливо для загальних моделей виробничих процесів, оскільки вони потенційно можуть бути розширені під конкретні сфери або поєднані з наявними стандартами[11].

Слід зважити на те, що методології онтологічного інженірингу не існують ізольовано. Вони тісно пов'язані з мовами представлення онтологій (як-от OWL), засобами творення (Protégé), а також логічними висновками (**reasoner'ами**). Метод окреслює не лише структуру розгортання, але й вимоги до якості моделі: її сталості, відсутності суперечностей, здатності до логічного виведення та можливості масштабування. Можна сказати, що методологія є **каркасом**, який забезпечує належність побудови онтології, тоді як інструменти втілення унаочнюють її у формальному вигляді.

Попри різноманіття підходів, усі вони сходяться на кількох фундаментальних принципах. По-перше, онтологія має будуватися на основі **чітко визначеної концептуальної моделі**, яка всебічно описує предметну область без семантичних провалів. По-друге, модель повинна **бути спрямована на подальше вдосконалення та повторне застосування** — це особливо актуально для загальних виробничих процесів, де можливе доповнення моделі новими параметрами або галузевими умовами. По-третє, **логічна коректність є критичною**: онтологія не повинна містити суперечностей, і її твердження повинні бути виведені з чітких формальних правил[12].

Підсумок

Методи онтологічного інженірингу можна вважати структурними орієнтирами, які дозволяють системно й послідовно будувати модель технологічного процесу. У рамках цієї роботи методологічна основа має ключову роль, оскільки саме з її допомогою забезпечується узгодженість майбутньої онтології, її здатність до перевірки та подальшого вдосконалення. Розробка універсальної моделі технологічного процесу вимагає поєднання можливостей, які надаються різними підходами: формальності **METHONTOLOGY**, практичності **Ontology Development 101** та модульності **NeOn Framework**. Така інтеграція дає змогу створити збалансовану модель, яка

буде придатною як для формального представлення процесів, так і для подальшого логічного аналізу й оптимізації.

1.4 Мови опису онтологій та інструменти їх реалізації

Сучасні онтологічні системи ґрунтуються на формальних мовах, які забезпечують точний опис знань і дозволяють виконувати логічні висновки. Формальні мови опису онтологій визначають набір інструментів для створення класів, властивостей, обмежень і логічних аксіом, які виступають основою для роботи reasoner'ів — програм, що перевіряють моделі та аналізують їхню поведінку. Для побудови універсальної моделі технологічного процесу ключовими є мови OWL, SWRL і середовище розробки Protégé.

1.4.1. OWL як базис для формалізованого подання онтологій

Мова **OWL (Web Ontology Language)** є стандартом для опису онтологій у рамках **Semantic Web**. Побудована на основі дескриптивної логіки, OWL має формальну семантику та сприяє автоматичному логічному висновку. Ця мова дозволяє створювати класи, визначати властивості, задавати обмеження й встановлювати комплексні логічні залежності.

Для моделювання технологічних процесів OWL надає змогу **будувати формальні описи як структури процесу, так і його динамічних характеристик**. Наприклад, можна визначити взаємозв'язки між використанням ресурсів і виконанням певної операції чи залежність між змінами параметрів. OWL підтримує поняття підкласів, доменів і діапазонів властивостей, еквівалентності класів і кардинальних обмежень, що робить її оптимальним вибором для моделювання складних технологічних систем.

Найбільш ефективною для інтелектуального аналізу є версія **OWL DL**, що збалансовує можливості виразності та обчислюваності. Вона широко використовується для створення моделей, які вимагають перевірки reasoner'ами, зокрема в універсальних технологічних процесах[13].

1.4.2. SWRL для побудови правил логічного виводу

Незважаючи на можливості OWL у створенні складних логічних структур, дескриптивна логіка не завжди достатня для точного опису причинно-

наслідкових зв'язків. У тих випадках, коли необхідно відобразити залежність типу "якщо ... то ...", застосовується мова SWRL (Semantic Web Rule Language).

SWRL додає до онтології правила, які дозволяють моделювати поведінку системи, що виходить за рамки OWL-аксіом. Наприклад:

- автоматичне коригування загального часу процесу залежно від тривалості операції;
- визначення ефективності процесу на основі комбінацій параметрів;
- умови для оцінки оптимальності чи неефективності процесу.

Ці можливості SWRL є особливо важливими для модулів оцінки продуктивності системи, які реалізуються у практичній частині дослідження. Завдяки включенню правил reasoner може не лише перевіряти модель, а й генерувати нові знання для інтелектуальної оптимізації технологічних процесів.

1.4.3. Protégé як основний інструмент роботи з онтологіями

Protégé є одним із найпопулярніших інструментів для створення, редагування та візуалізації онтологій, що зумовлено його відкритою архітектурою, широким функціоналом і можливістю інтеграції з reasoner'ами, такими як **HermiT** і **Pellet**. Платформа підтримує роботу з **OWL** та **SWRL**, надаючи всі необхідні інструменти для повного циклу розробки – від проектування класів до перевірки моделі на узгодженість[14].

У сфері моделювання технологічних процесів **Protégé** виконує низку важливих функцій:

- **забезпечує середовище** для розроблення ієрархії процесів, ресурсів і параметрів;
- дає змогу **задавати обмеження**, які визначають логіку взаємодії між елементами;
- **підтримує виконання reasoning-операцій** для перевірки коректності та узгодженості моделі;
- **дозволяє створювати SWRL-правила**, необхідні для аналізу ефективності процесів;

- **забезпечує візуалізацію структури онтології**, спрощуючи її оцінку і документування.

Protégé є не тільки інструментом для моделювання, але й потужною платформою для експериментальних досліджень. Завдяки цьому його особливо цінують під час створення універсальних онтологічних моделей технологічних процесів. У цьому середовищі реалізуються логічні залежності між параметрами моделі, а також проводиться тестування її працездатності, що буде детальніше розглянуто у наступних розділах дослідження.

1.5. Концептуальні основи побудови універсальної онтологічної моделі технологічного процесу

Розробка універсальної онтології технологічного процесу вимагає створення цілісної концептуальної основи, яка враховує специфіку процесів у різних галузях, їхню будову та можливості для інтелектуального аналізу. У порівнянні з галузевими моделями, які сфокусовані на вузьких предметних областях, універсальна онтологія повинна відповідати високому рівню абстракції, аби охоплювати широкий спектр процесів. Водночас вона має бути достатньо деталізованою, щоб забезпечувати логічну чіткість і застосування процедур виведення. Для цього необхідно визначити ключові елементи моделі, описати структуру їхніх взаємозв'язків, а також розробити механізми, які дозволять формально моделювати й аналізувати поведінку процесів.

1.5.1. Базові сутності та їх роль у моделі

Будь-який технологічний процес можна описати через набір ключових елементів, які залишаються незмінними незалежно від сфери застосування. Основною складовою такої моделі є **Процес** — цілісна одиниця виконання, що має чітко окреслену мету, визначені вхідні та вихідні параметри. Процес складається з **Операцій**, які можуть представляти як окремі базові дії, так і їхні складні комбінації. Кожна операція характеризується тривалістю, якістю виконання, вимогами до ресурсів і можливими обмеженнями.

Ресурси займають особливе місце в цьому контексті. Вони бувають матеріальними, інформаційними, енергетичними або трудовими та визначають

здатність системи здійснювати операції. Ресурси створюють умови, що безпосередньо впливають на продуктивність і ефективність процесу. **Параметри** є кількісними чи якісними характеристиками операцій і всього процесу загалом. Вони можуть включати такі показники, як тривалість, витрати, продуктивність, надійність або будь-які інші значущі для аналізу параметри. Завершальним компонентом є **Результат**, який відображає змінений стан об'єкта після завершення процесу і служить основою для оцінювання його ефективності.

Ці базові складові формують основу онтологічної моделі, навколо якої будуються всі інші елементи. Їхнє чітке визначення забезпечує однозначність інтерпретації та відкриває можливість адаптації моделі до специфічних сценаріїв без необхідності втручання у її фундаментальну структуру.

1.5.2. Логічні зв'язки між елементами технологічного процесу

Універсальна онтологічна модель зобов'язана охоплювати не лише структуру процесу, але й специфіку взаємодії між його складовими. Логічні залежності визначають характер впливу окремих операцій одна на одну, окреслюють ресурсні потреби для кожного етапу та демонструють наслідки змін одного параметра для інших. Саме ці взаємозв'язки забезпечують функціонування **reasoner'a**, дозволяючи йому здійснювати логічне виведення і здійснювати глибокий аналіз поведінки моделі.

Фундаментальні взаємозв'язки в системах включають декілька ключових аспектів: співвідношення між операціями, які визначають їхній характер (такі як підпорядкованість, послідовність чи паралельність), залежності між операцією та ресурсами (зокрема, необхідність, споживання, використання), а також взаємодію між параметрами (наприклад, вплив, обмеження чи визначення). Як приклад можна розглянути залежність тривалості виконання операцій від обсягу залучених ресурсів, тоді як показники якості можуть бути зумовлені характеристиками параметрів попередніх етапів процесу.

Завдяки цим взаємозв'язкам універсальна модель перетворюється із простої структурної схеми на формалізовану систему з виразною семантикою. **Особливо важливим є те, що логічні залежності дозволяють знаходити нові**

значення параметрів за допомогою reasoning-процедур. Це, у свою чергу, сприяє прояву інтелектуальної поведінки моделі та забезпечує її ефективність у задачах оптимізації.

1.5.3. Формалізація критеріїв ефективності та їх інтеграція в онтологію

Універсальна онтологія технологічного процесу має включати інструменти для оцінки його ефективності. На відміну від структурних моделей, що обмежуються описом процесів, онтологія повинна забезпечувати можливість аналітичного розрахунку ефективності, спираючись на чітко визначені параметри й логічні взаємозв'язки. Це потребує формалізації критеріїв, які дозволяють визначати оптимальність або виявляти недоліки в роботі процесу.

До таких критеріїв можуть відноситися показники часу, використання ресурсів, якості, надійності, вартості або продуктивності. Для формалізації критеріїв необхідно застосовувати логічні конструкції **OWL** та правила **SWRL**, які дозволяють встановлювати взаємозв'язки між параметрами і визначати умови, за яких процес відповідає вимогам ефективності. Наприклад:

- *коли тривалість операції перевищує заданий поріг, процес вважається неефективним;*
- *якщо параметри якості відповідають визначеним межам, результат класифікується як оптимальний.*

Вказані правила інтегруються в онтологію як формалізовані інструменти оцінювання, що сприяють реалізації автоматизованого логічного виведення. Такий підхід дозволяє моделі не тільки описувати відповідний процес, але й здійснювати його глибший аналіз, визначаючи ефективність залежно від заданих параметрів та їх взаємозв'язків.

Висновки

1. Онтологічний підхід у моделюванні складних систем

- Встановлено, що онтології надають формалізований, точний та машиночитний спосіб представлення знань.

- Аргументовано переваги онтологічного підходу над традиційними моделями процесів, демонструючи його ефективність у складних системах.
- Підкреслено важливість забезпечення логічної послідовності та використання процедур виводу (reasoning) для глибшого аналізу технологічних систем.

2. Особливості моделювання технологічних процесів

- Сформульовано універсальну структуру технологічного процесу, що включає операції, ресурси, параметри та результати.
- Визначено ключові вимоги до формалізації технологічних процесів: універсальність застосування, можливість масштабування та логічна чіткість.
- Акцентовано потребу в онтологічному представленні процесів задля впровадження засобів інтелектуального аналізу й оптимізації.

3. Сучасні підходи до онтологічного інжинірингу

- Проведено аналіз трьох провідних методологій: METHONTOLOGY, Ontology Development 101 та NeOn Framework.
- Продемонстровано, що їхнє комбінування сприяє створенню якісної, гнучкої та логічно узгодженої онтології.
- Підтверджено, що ці підходи є найкращим вибором для розробки універсальної моделі технологічних процесів.

4. Мови опису онтологій та інструменти для їх реалізації

- Визначено ключове значення мови OWL у формальному моделюванні та проведенні логічного аналізу.
- Розкрито важливість застосування правил SWRL для формалізації причинно-наслідкових зв'язків і оцінки ефективності процесів.
- Доведено, що Protégé є основною платформою для розробки, візуалізації та тестування онтологій.

5. Концептуальні основи створення універсальної моделі

- Сформульовано набір базових компонентів: процес, операція, ресурс, параметр та результат.

- Визначено типи логічних зв'язків, які забезпечують цілісність і коректність поведінки моделі.
- Розроблено підхід до інтеграції ефективніших критеріїв та механізмів обґрунтованого висновування

РОЗДІЛ 2.

РОЗРОБЛЕННЯ УНІВЕРСАЛЬНОЇ ОНТОЛОГІЧНОЇ МОДЕЛІ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ

2.1. Аналіз предметної області та формування вимог до моделі

Технологічні процеси в різних галузях мають подібну структуру: вони складаються з послідовних операцій, потребують використання обладнання та матеріалів, характеризуються набором параметрів і формують кінцевий результат із певними властивостями.[16] Незважаючи на різницю між конкретними видами виробництва, більшість із них мають спільні елементи, які можна формалізувати за допомогою універсальної онтологічної моделі.

У рамках аналізу предметної області було визначено ключові характеристики технологічних процесів:

- наявність етапів та підоперацій;
- використання технічного або програмного обладнання;
- залежність від матеріалів та умов виконання;
- вплив параметрів на якість і стабільність результату;
- необхідність у формальному описі для контролю та оптимізації.

Онтологічний підхід дозволяє подати технологічний процес як систему взаємопов'язаних сутностей, де кожен елемент — операція, ресурс, параметр чи результат — має чітко визначені властивості та відношення. Це створює основу для подальшого логічного аналізу, автоматичної перевірки коректності та використання reasoning.

На основі проведеного аналізу сформовано основні вимоги до онтологічної моделі:

1. **Універсальність** — можливість описувати різні типи технологічних процесів.
2. **Чітка структура операцій** — підтримка послідовності та ієрархії.
3. **Опис параметрів і їх впливу** — для подальшої оцінки якості результатів.

4. **Взаємозв'язок між сутностями** — операції, ресурси, параметри, результати.

5. **Підтримка reasoning** — здатність системи робити висновки та виявляти відхилення.

6. **Можливість практичного застосування** — у реальних сценаріях, зокрема для процесу 3D-друку.

Таким чином, аналіз предметної області дозволив окреслити загальні закономірності технологічних процесів і визначити вимоги, на основі яких побудована універсальна онтологічна модель[17].

2.2. Визначення базових сутностей, класів та властивостей онтології

Створення онтологічної моделі технологічного процесу 3D-друку передбачає ясне визначення основних сутностей (класів), їх підкласів та логічних зв'язків, що відображають структурну та функціональну організацію процесу. Усі складові моделі сформовано на основі аналізу предметної області та узагальнення етапів дійсного технологічного циклу адитивного виробництва.

Основні класи онтології

1. Технологічний процес 3D-друку (ThreeDPrintingProcess)

Головний клас, що описує весь виробничий цикл 3D-друку. Він пов'язує всі інші елементи моделі через відношення:

- *містить операції,*
- *використовує обладнання,*
- *використовує матеріали,*
- *описується параметрами,*
- *базується на моделі,*
- *має правила друку,*
- *має результат у вигляді надрукованого об'єкта.*

2. Операції 3D-друку (PrintingOperation)

Клас, який включає послідовні етапи процесу. Структура підкласів відповідає реальному виробничому циклу:

- **Slicing (Слайсинг)**

- **EquipmentSetup (Підготовка обладнання)**
- **LayerPrinting (Друк шару)**
- **CoolingStage (Охолодження)**
- **FinishingStage (Пост-обробка)**

Ці операції забезпечують логічний поділ технологічного процесу та формують основу для опису параметрів, ресурсів і правил друку.

3. Геометрична модель (ModelGeometry)

Описує форму, структуру та просторові характеристики 3D-моделі. Є підґрунтям для визначення правил друку та вибору параметрів.

Зв'язки:

- *базується на моделі — від процесу,*
- *регламентується правилами друку.*

4. Обладнання для 3D-друку (PrintingEquipment)

Описує апаратну частину системи. Пов'язане з матеріалами та параметрами, які воно підтримує.

5. Матеріали для друку (PrintingMaterial)

Клас, що характеризує вибір матеріалу (PLA, ABS, PETG тощо) та його властивості.

Впливає на:

- температуру,
- силу адгезії,
- якість поверхні,
- правила та обмеження друку.

6. Правила друку (PrintingRule)

Формалізований набір технологічних рекомендацій та обмежень, які визначають:

- параметри друку,
- взаємозв'язки температури, швидкості та геометрії,
- залежності між властивостями матеріалу та якістю виробу.

7. Параметри друку (PrintParameter)

Описують технологічні змінні, які безпосередньо впливають на результат друку:

- температура сопла,
- температура платформи,
- швидкість друку,
- товщина шару,
- інші регульовані параметри.

Параметри:

- *визначають поверхню виробу,*
- *впливають на ефективність процесу.*

8. Надрукований виріб (PrintedObject)

Клас результату технологічного процесу. Має властивість:

- *має поверхню.*

9. Поверхня виробу (Surface)

Характеризує:

- рівність,
- точність,
- якість зовнішнього шару.

Пов'язана із параметрами друку та критеріями ефективності.

10. Критерії ефективності (EfficiencyCriterion)

Формують показники для оцінювання якості процесу та результату:

- якість поверхні,
- адгезія шарів,
- точність,
- час виготовлення тощо.

Поверхня *впливає* на критерії, а виріб *оцінюється* за ними.

Базові властивості (Object properties)

В онтології визначено такі ключові властивості:

Таблиця 2.1 – Основні класи універсальної онтологічної моделі

Властивість	Опис	Domain → Range
hasOperation	процес містить операції	ThreeDPrintingProcess → PrintingOperation
includesStep	операція містить підоперації	PrintingOperation → PrintingOperation
basedOnModel	процес базується на геометричній моделі	ThreeDPrintingProcess → ModelGeometry
describedByParameters	процес описується параметрами	ThreeDPrintingProcess → PrintParameter
usesEquipment	процес використовує обладнання	ThreeDPrintingProcess → PrintingEquipment
usesMaterial	обладнання використовує матеріали	PrintingEquipment → PrintingMaterial
hasRule	процес має правила друку	ThreeDPrintingProcess → PrintingRule
regulatedByRule	модель регламентується правилами	ModelGeometry → PrintingRule
resultsInObject	процес має результат — виріб	ThreeDPrintingProcess → PrintedObject

Властивість	Опис	Domain → Range
hasSurface	виріб має поверхню	PrintedObject → Surface
surfaceDefinedBy	поверхня визначається параметрами	Surface → PrintParameter
surfaceAffectsEfficiency	поверхня впливає на критерії	Surface → EfficiencyCriterion
evaluatedBy	виріб оцінюється за критеріями	PrintedObject → EfficiencyCriterion

2.3. Розроблення структурної моделі технологічного процесу

Структурна модель визначає внутрішню організацію технологічного процесу та відображає, які сутності беруть участь у його виконанні, які між ними існують зв'язки та як вони впливають одна на одну[19]. Модель створюється на основі базових класів, сформованих у попередньому підрозділі, та формує узгоджену систему, яка може бути реалізована у вигляді онтології на мові OWL.

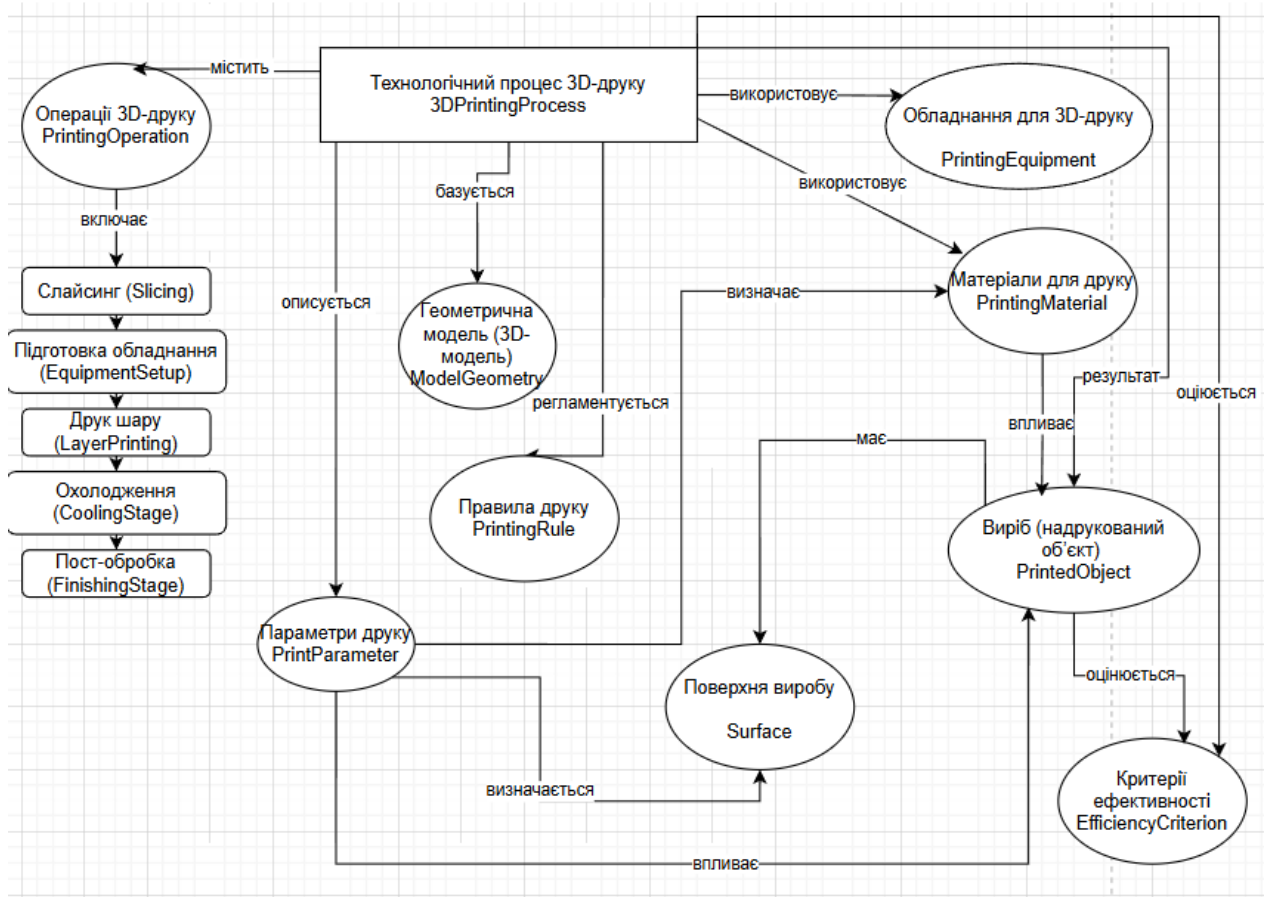


Рисунок 2.1 – Структурна модель технологічного процесу

2.3.1. Основна логіка структурної моделі

В основі моделі лежить клас **Process**, який описує технологічний процес як сукупність операцій, що виконуються у визначеній послідовності. Кожна операція має власні ресурси, параметри та впливає на проміжні або фінальні характеристики результату.

Відповідно до цього, структурна модель містить такі ключові компоненти:

1. **Процес** (Process)
– головна сутність, яка об'єднує всі елементи.
2. **Операції** (Operation)
– структурні підрозділи процесу, що описують окремі кроки.
3. **Ресурси** (Resource)
– те, що необхідно операціям: обладнання, матеріали, ПЗ.
4. **Параметри** (Parameter)
– числові чи логічні значення, що визначають режим виконання операцій.

5. Результат**(Result)**

– характеристика або об'єкт, отриманий після завершення процесу.

6. Критерії ефективності (EfficiencyCriterion)

– набір правил та параметрів, за якими оцінюється процес.

2.3.2. Структурна схема моделі (текстове подання)

Класи та властивості онтології формалізовані в табличному вигляді, де перша таблиця описує концепти, друга — семантичні зв'язки між ними, третя — параметри з числовими значеннями

Таблиця 2.2 – Таблиця класів онтології

	Ідентифікатор класу (OWL)	Українська назва	Короткий зміст / роль
	ThreeDPrintingProcess	Технологічний процес 3D-друку	Головний клас, що описує весь процес 3D-друку як сукупність операцій, параметрів, ресурсів і результату.
	PrintingOperation	Операції 3D-друку	Узагальнений клас для всіх етапів/операцій технологічного процесу 3D-друку.
	Slicing	Слайсинг	Підклас PrintingOperation; перетворення 3D-моделі на шари та генерація керуючого коду.
	EquipmentSetup	Підготовка обладнання	Підклас PrintingOperation; калібрування та налаштування 3D-принтера перед друком.
	LayerPrinting	Друк шару	Підклас PrintingOperation; безпосереднє нанесення шарів матеріалу згідно G-коду.
	CoolingStage	Охолодження	Підклас PrintingOperation; контрольоване охолодження виробу після друку.
	FinishingStage	Пост-обробка	Підклас PrintingOperation; видалення підтримок, шліфування, очищення виробу.
	ModelGeometry	Геометрична модель (3D-модель)	Описує геометрію виробу, що друкується; використовується як основа для слайсингу та правил друку.

	Ідентифікатор класу (OWL)	Українська назва	Короткий зміст / роль
	PrintingEquipment	Обладнання для 3D-друку	3D-принтер та пов'язане з ним обладнання, що використовується у процесі.
0	PrintingMaterial	Матеріали для друку	Матеріали (філаменти), які обробляються обладнанням і впливають на характеристики виробу.
1	PrintingRule	Правила друку	Набір технологічних правил і обмежень (діапазони параметрів, режими роботи), що регламентують процес.
2	PrintParameter	Параметри друку	Узагальнений клас параметрів (температура сопла, температури столу, швидкість, товщина шару тощо), які описують режим друку.
3	Surface	Поверхня виробу	Характеристики поверхні надрукованого об'єкта (шорсткість, ступінчатість, дефекти шарування).
4	PrintedObject	Виріб (надрукований об'єкт)	Кінцевий результат процесу 3D-друку з власною геометрією та поверхнею.
5	EfficiencyCriterion	Критерії ефективності	Клас, що описує критерії оцінки процесу/виробу (час, якість, точність, стійкість шарів тощо).

Таблиця 2.3 – Таблиця об'єктних властивостей

	Ідентифікатор властивості (OWL)	Українська назва / підпис на схемі	Domain (звідки)	Range (куди)	Семантика (що означає)
	hasOperation	«містить операції»	ThreeDPrintingProcess	PrintingOperation	Технологічний процес складається з набору операцій 3D-друку.
	includesStep	«включає етап»	PrintingOperation	PrintingOperation	Узагальнена операція може включати підоперації (деталізація процесу).
	basedOnModel	«базується на моделі»	ThreeDPrintingProcess	ModelGeometry	Процес 3D-друку виконується на основі певної 3D-моделі.
	describedByParameters	«описується параметрами»	ThreeDPrintingProcess	PrintParameter	Режим виконання процесу задається параметрами друку.
	usesEquipment	«використовує обладнання»	ThreeDPrintingProcess	PrintingEquipment	У процесі використовується конкретне обладнання для 3D-друку.
	usesMaterial	«використовує матеріали»	PrintingEquipment	PrintingMaterial	Обладнання працює з певними матеріалами для друку.
	hasRule	«має правила друку»	ThreeDPrintingProcess	PrintingRule	Процес регламентується набором технологічних правил друку.

	Ідентифікатор властивості (OWL)	Українська назва / підпис на схемі	Domain (звідки)	Range (куди)	Семантика (що означає)
	regulatedByRule	«регламентується правилами»	ModelGeometry	PrintingRule	Вимоги до геометричної моделі та її друку задаються правилами друку.
	resultsInObject	«має результат-виріб» / «результат»	ThreeDPrintingProcess	PrintedObject	Результатом виконання процесу є надрукований об'єкт.
0	hasSurface	«має поверхню»	PrintedObject	Surface	Кожен надрукований об'єкт має певну поверхню.
1	surfaceDefinedBy	«поверхня визначається параметрами»	Surface	PrintParameter	Властивості поверхні залежать від конкретних параметрів друку.
2	surfaceAffectsEfficiency	«поверхня впливає на ефективність»	Surface	EfficiencyCriterion	Стан поверхні враховується при оцінці ефективності процесу.
3	evaluatedBy	«оцінюється за критеріями»	PrintedObject	EfficiencyCriterion	Надрукований виріб оцінюється за визначеними критеріями ефективності.

Таблиця 2.3 – Атрибутивні (datatype) властивості

№	Назва властивості	URI	Domain	Range	Опис
1	має числове значення	:hasValue	PrintParameter	xsd:double	Значення параметру: температура, швидкість тощо.
2	має одиницю вимірювання	:hasUnit	PrintParameter	xsd:string	°C, mm/s, mm.
3	текстова характеристика об'єкта	:hasTextValue	PrintedObject / Surface	xsd:string	Опис дефектів, якості, шорсткості, характеристик поверхні.

2.3.3. Остаточна структурна модель (опис словами)

Кінцева структурна модель технологічного процесу 3D-друку зображує процес як сукупність взаємодіючих сутностей, де кожна виконує окрему функцію, але творить єдине логічне ланцюг. У ядрі моделі міститься клас **ThreeDPrintingProcess**, який узагальнює всі стадії та об'єкти, що залучені у друці.

Процедура 3D-друку описується низкою дій класу **PrintingOperation**, які здійснюються у встановленому порядку: *slicing* → *equipment setup* → *layer printing* → *cooling* → *finishing*. Кожен етап має власні змістовні аспекти, впливає на якість відбитка та визначений власними чинниками.

Процес ґрунтується на заздалегідь підготовленій **ModelGeometry**, яка є 3D-образом майбутнього виробу.[20] Геометрія встановлює вимоги до товщини шару, точності розміщення, допоміжних структур, що, своєю чергою, задається класами **PrintParameter** та **PrintingRule**.

Ресурси для друку описуються класом **PrintingMaterial**, а апаратура (принтер, екструдер, робоча поверхня) — класом **PrintingEquipment**. Відношення *usesMaterial()* між апаратурою та ресурсами визначає технічну сполучуваність.

Підсумком процесу є об'єкт класу **PrintedObject**, який має поверхню, утворену у ході друку. Поверхня характеризується класом **Surface**, а її ознаки — показниками параметрів друку (температура, швидкість, висота шару). Саме ці характеристики визначають дієвість процесу, яка вимірюється класами **EfficiencyCriterion**.

Отже, *структурна модель* окреслює цілісну систему з'єднань між діями, показниками, ресурсами та наслідком, даючи змогу формувати повний технологічний виклад та забезпечувати здатність до міркування (reasoning).

2.4. Формалізація логічних зв'язків між структурними елементами технологічного процесу

Формалізація логічних зв'язків є головним етапом побудови онтологічної моделі технологічного процесу 3D-друку. Саме зв'язки задають правила взаємодії між класами, визначають причинно-наслідкові залежності та забезпечують можливість автоматичних логічних висновків за допомогою reasoning-механізмів Protégé.

Універсальна модель, сформована в попередніх підрозділах, містить кілька груп сутностей — операції, параметри друку, обладнання, матеріали, правила, а також результати процесу[22]. Логічні властивості задають зв'язки між ними та дозволяють описати повну структуру технологічного процесу.

2.4.1. Основні групи логічних зв'язків

У моделі визначено кілька ключових типів зв'язків, кожен із яких формалізує важливу частину структури процесу:

1. Процес → Операції (hasOperation / includesStep)

Ці властивості задають структуру технологічного процесу як послідовності етапів.

- **hasOperation** пов'язує процес із основними класами операцій: «Slicing», «EquipmentSetup», «LayerPrinting», «CoolingStage», «FinishingStage».
- **includesStep** дозволяє деталізувати ієрархію підоперацій.

Цей тип зв'язку визначає технологічний процес як структурований сценарій.

2. Процес → Модель (basedOnModel)

Зв'язок показує, що технологічний процес 3D-друку завжди базується на певній вихідній 3D-моделі.

Приклад

використання:

ThreeDPrintingProcess → basedOnModel → ModelGeometry.

Цей зв'язок гарантує, що життєвий цикл процесу неможливий без заданої вихідної моделі.

3. Процес → Параметри (describedByParameters)

Параметри визначають умови, у яких виконується процес. До них належать:

- температура сопла,
- швидкість друку,
- товщина шару,
- температура платформи,
- швидкість охолодження тощо.

Зв'язок:

ThreeDPrintingProcess → describedByParameters → PrintParameter.

4. Процес → Обладнання (usesEquipment)

Друк можливий лише за наявності обладнання, яке відповідає параметрам: ThreeDPrintingProcess → usesEquipment → PrintingEquipment.

5. Обладнання → Матеріали (usesMaterial)

Обладнання працює з одним або кількома типами матеріалів.

PrintingEquipment → *usesMaterial* → *PrintingMaterial*

Це дозволяє reasoner-у автоматично визначати, чи сумісні параметри температури з обраним типом матеріалу.

6. Процес → Правила друку (hasRule / regulatedByRule)

Правила описують обмеження:

- мінімальна/максимальна товщина шару,
- допустимий діапазон температур,
- вимоги до охолодження,
- геометричні обмеження.

<i>ThreeDPrintingProcess</i> → hasRule → <i>PrintingRule</i>
--

<i>ModelGeometry</i> → regulatedByRule → <i>PrintingRule</i>
--

Таким чином, правила пов'язуються і з процесом, і з вихідною моделлю.

7. Процес → Результат (resultsInObject)

У результаті виконання формального набору операцій створюється надрукований виріб:

<i>ThreeDPrintingProcess</i> → resultsInObject → <i>PrintedObject</i>

8. Виріб → Поверхня (hasSurface / surfaceDefinedBy)

Поверхня є ключовою частиною оцінювання якості.

<i>PrintedObject</i> → hasSurface → <i>Surface</i>
--

<i>Surface</i> → surfaceDefinedBy → <i>PrintParameter</i>

9. Поверхня → Критерії ефективності (surfaceAffectsEfficiency / evaluatedBy)

Результат оцінюється за критеріями:

- точність,
- якість поверхні,
- адгезія шарів,
- час виготовлення.

<i>Surface</i> → surfaceAffectsEfficiency → <i>EfficiencyCriterion</i>
--

<i>PrintedObject</i> → evaluatedBy → <i>EfficiencyCriterion</i>

2.4.2. Формалізація властивостей у Protégé (OWL-подання)

Усі вищезазначені зв'язки були формалізовані у Protégé як:

- **Object** **properties:**

hasOperation, includesStep, basedOnModel, describedByParameters, usesEquipment, usesMaterial, hasRule, regulatedByRule, resultsInObject, hasSurface, surfaceDefinedBy, surfaceAffectsEfficiency, evaluatedBy.

- **Datatype** **properties:**

hasValue, hasUnit — для числових та текстових значень параметрів.

Кожне зв'язування має домен і рандж, що дозволяє reasoner-у робити логічні висновки.

2.4.3. Роль логічних зв'язків у reasoning

Reasoner використовує зв'язки для:

- визначення сумісності параметрів і матеріалів;
- перевірки відповідності процесу правилам;
- класифікації процесів (ефективний, неефективний);
- автоматичного виявлення помилок моделювання;
- виявлення логічних конфліктів у параметрах.

Наприклад, зв'язок:

surfaceDefinedBy → PrintParameter

дозволяє reasoner-у автоматично виявити причини поганої якості поверхні.

2.4.4. Приклад формалізації на основі процесу 3D-друку

Для демонстрації формалізації зв'язків розглянуто параметри FDM-друку:

Температура сопла

- впливає на плавлення матеріалу;
- визначає якість поверхні;
- обмежується типом філамента.

Формалізація:

TemperatureNozzle →influences→ SurfaceQuality

TemperatureNozzle →restrictedBy→ Material

Швидкість друку

- впливає на точність;
- пов'язана з товщиною шару;
- збільшення швидкості знижує деталізацію.

Формалізація:

PrintSpeed →influences→ PrintAccuracy

PrintSpeed →influences→ LayerAdhesion

Товщина шару

- визначає час друку;
- впливає на ступінчатість поверхні.

Формалізація:

LayerHeight →influences→ PrintingTime

LayerHeight →influences→ SurfaceSmoothness

Критерії ефективності

Кожен критерій має зв'язки з параметрами, які його формують.

SurfaceSmoothness →evaluatedBy→ QualityCriterion

PrintingTime →evaluatedBy→ TimeCriterion

2.4.5. Логічні правила для reasoning

Для автоматизації аналізу моделі були сформовані узагальнені правила SWRL:

Приклад 1: Перегрів

Temperature(?t) ∧ hasValue(?t, ?v) ∧ swrlb:greaterThan(?v, 250)

→ HighTemperatureCondition(?t)

Приклад 2: Низька точність при великій швидкості

PrintSpeed(?s) ∧ hasValue(?s, ?v) ∧ swrlb:greaterThan(?v, 120)

→ LowAccuracyRisk

Приклад 3: Залежність часу від товщини шару

LayerHeight(?h) ∧ hasValue(?h, ?v) ∧ swrlb:greaterThan(?v, 0.28)

→ ExtendedPrintingTime

Такі правила дозволяють системі самостійно виявляти проблеми, класифікувати режими як ефективні або неефективні і пропонувати оптимізацію.

Висновок

Формалізація логічних зв'язків між параметрами процесу дозволяє перетворити опис технологічного процесу на структуровану систему, придатну для автоматизованого аналізу. Це робить онтологічну модель не лише статичним описом, а й інструментом для логічних висновків і оцінки ефективності, що є ключовим для подальшого використання моделі в реальних технологічних сценаріях.

2.5. Створення модуля оцінки ефективності

Модуль оцінки ефективності є ключовим елементом універсальної онтологічної моделі, оскільки він дозволяє визначати, наскільки технологічний процес відповідає заданим критеріям якості, стабільності та продуктивності. Цей модуль забезпечує можливість логічного аналізу процесу, автоматичної перевірки параметрів та виявлення неефективних або критичних режимів.

Модуль оцінки ефективності складається зі структури класів, системи властивостей та набору логічних правил, що описують залежність між параметрами процесу та якістю кінцевого результату.

2.5.1. Основні концепції модуля

У межах технологічного процесу 3D-друку кінцевий виріб визначається низкою властивостей: якістю поверхні, точністю відтворення геометрії, стабільністю шарів, відсутністю дефектів тощо[30]. Ці властивості напряду залежать від параметрів процесу (температури, швидкості, типу матеріалу, товщини шару), а модуль оцінки ефективності встановлює порядки та критерії, за якими оцінюється досягнуте значення кожної характеристики.

Модуль дозволяє:

- оцінити, чи відповідає результат встановленим вимогам;
- ідентифікувати критичні відхилення параметрів;
- автоматично класифікувати процес як ефективний, субоптимальний або неефективний;
- підтримувати reasoning — автоматичне логічне виведення в Protégé.

Таким чином, модуль слугує інструментом контролю якості технологічного процесу та забезпечує основу для подальшої автоматизації оптимізації параметрів.

2.5.2. Класи та концепції модуля ефективності

Згідно з розробленою моделлю, до модуля оцінки ефективності входять такі класи:

EfficiencyCriterion

Базовий клас, що представляє окремий критерій, за яким здійснюється оцінювання.

Критерії можуть бути кількісними або якісними.

Похідні критерії:

- **SurfaceQualityCriterion** — оцінка гладкості поверхні;
- **AccuracyCriterion** — оцінка точності геометрії;
- **TimeCriterion** — оцінка часу виготовлення;
- **LayerAdhesionCriterion** — оцінка якості зчеплення шарів.

PrintedObject

Результат технологічного процесу. Саме цей клас є основним об'єктом оцінювання.

Surface

Окремий аспект виробу, що впливає на оцінку. Поверхня є ключовим параметром для критеріїв якості.

EfficiencyLevel

Категорія оцінки ефективності:

- **EfficientProcess**
- **SuboptimalProcess**
- **InefficientProcess**

Ці категорії `reasoner` визначає автоматично на основі логічних правил.

2.5.3. Формалізація властивостей модуля

Для опису взаємодії між критеріями, результатом і параметрами застосовуються властивості:

- **evaluatedBy (PrintedObject → EfficiencyCriterion)**

Виріб оцінюється за конкретними критеріями.

- **hasSurface (PrintedObject → Surface)**

У кожного виробу є поверхня як аспект якості.

- **surfaceDefinedBy (Surface → PrintParameter)**

Параметри процесу визначають характеристики поверхні.

- **surfaceAffectsEfficiency (Surface → EfficiencyCriterion)**

Поверхня прямо впливає на підсумковий рівень ефективності.

- **hasValue, hasUnit**

Значення параметрів, потрібні для аналізу критичних відхилень.

Завдяки цим властивостям модель здатна відстежувати причинно-наслідкові зв'язки між параметрами процесу та кінцевим результатом.

2.5.4. SWRL-правила для автоматичного оцінювання

У межах Protégé підтримується механізм SWRL, який дозволяє створювати логічні правила для автоматичного аналізу результату. Правила спираються на діапазони значень параметрів і відповідні критерії.

Приклад 1. Критична температура друку

TemperatureParameter(?t) ∧ hasValue(?t, ?v) ∧ swrlb:greaterThan(?v, 250)
→ InefficientProcess

Приклад 2. Ризик низької точності при високій швидкості

SpeedParameter(?s) ∧ hasValue(?s, ?v) ∧ swrlb:greaterThan(?v, 120)
→ SuboptimalProcess

Приклад 3. Оптимальний час виготовлення

TimeParameter(?t) ∧ hasValue(?t, ?v) ∧ swrlb:lessThan(?v, 120)
→ EfficientProcess

Такі правила дозволяють reasoner'у:

- робити висновки щодо якості процесу,
- автоматично визначати рівень ефективності,
- виявляти відхилення без участі користувача.

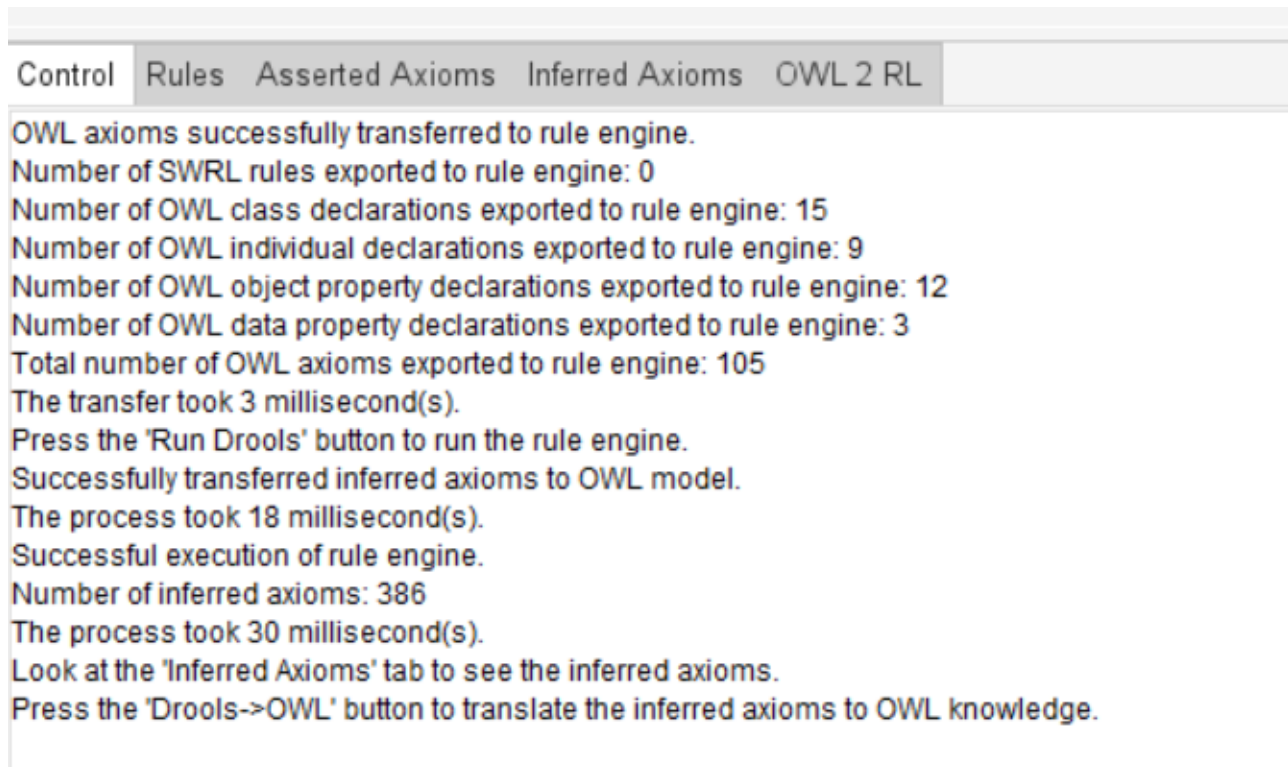


Рисунок 2.2 – Результати логічного виводу reasoner у середовищі Protégé

2.5.5. Підсумкова роль модуля ефективності

Модуль оцінки ефективності завершує опис універсальної онтологічної моделі тим, що перетворює її зі структурного опису на знаряддя для аналізу та автоматичного прийняття рішень. Завдяки цьому:

- користувач бачить не лише структуру процесу, а й оцінку його гатунку;
- система може виявляти критичні режими 3D-друку;
- модель стає придатною для справжнього застосування.

2.6. Реалізація онтології в Protégé

Реалізація онтологічної моделі технологічного процесу 3D-друку була виконана у середовищі Protégé 5.x. - актуальному редакторі OWL-онтологій, що надає змогу конструювання, правлення та оцінки формальних зразків. На цьому етапі перенесено структуру, описану в попередніх розділах, включаючи класи, підкласи, атрибути та початкові обмеження.

Головна мета втілення – отримати коректну OWL-модель, яку можливо слугувати для міркування (reasoning), верифікації узгодженості та використання у майбутніх наукових завданнях роботи

2.6.1. Створення структури класів

У Protégé була сформована деревоподібна структура класів згідно розробленої концептуальної моделі. Основний клас **ThreeDPrintingProcess** розгортається на підкласи операцій, матеріалів, обладнання, правил друку та критеріїв ефективності.

Усі основні сутності з розділу 2.2 були створені як OWL-класи.

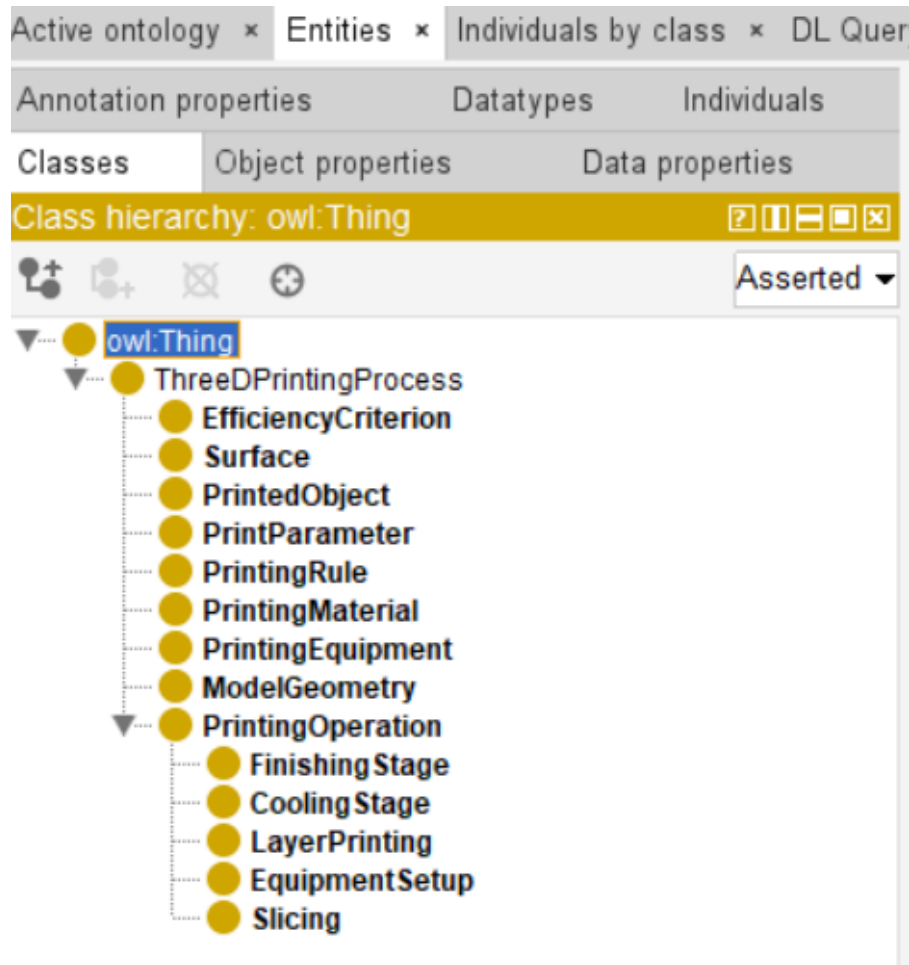


Рисунок 2.3 – Ієрархія класів онтологічної моделі у середовищі Protégé

2.6.2. Реалізація об'єктних властивостей

Було створено об'єктні властивості, які формують логічні зв'язки між класами. Основні з них:

- **hasOperation** — зв'язує процес з його операціями.
- **usesEquipment** — прив'язує обладнання до процесу.
- **usesMaterial** — вказує матеріал, який підтримує обладнання.
- **describedByParameters** — задає параметри друку.
- **resultsInObject** — формує результат друку.

- **hasSurface, surfaceDefinedBy** — дозволяють описати властивості поверхні виробу.
- **evaluatedBy, surfaceAffectsEfficiency** — забезпечують зв'язки між результатами та критеріями ефективності.

Кожна властивість отримала домен та рандж згідно з формалізацією, наведеною у розділі 2.3.

2.6.3. Створення атрибутів (data properties)

Для параметрів друку були створені атрибути, що дозволяють задавати числові значення:

- **hasValue** (числове значення, наприклад товщина шару 0.2 мм),
- **hasUnit** (одиниця вимірювання, напр. “mm/s”, “C”).

Ці параметри використовуються надалі у reasoning-правилах для визначення ефективності процесу.

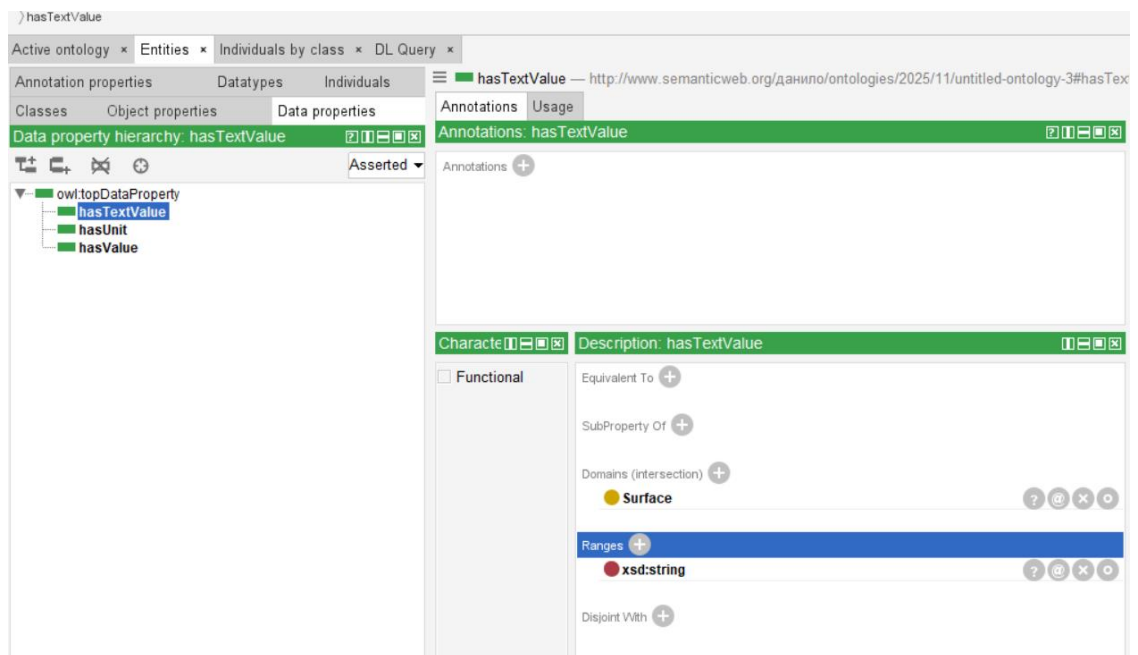


Рисунок 2.4 – Реалізація дата-властивості `hasTextValue` в онтологічній моделі у середовищі Protégé

2.6.4. Формування екземплярів (individuals)

Для демонстрації роботи моделі в Protégé були створені приклади індивідів, що представляють конкретний 3D-друк:

- **ExampleMaterial_PLA** — матеріал PLA;
- **ExampleEquipment_PrusaMK3S** — FDM-принтер;

- **ExampleParam_Temp200C** — температура друку 200 °C;
- **ExampleProcess_FDMPrint1** — конкретний технологічний процес;
- **ExampleResult_Object1** — надрукований об'єкт.

Дані індивіди використовуються в reasoning, щоб оцінити відповідність критеріям ефективності (наприклад, чи є температура занадто високою/низькою).

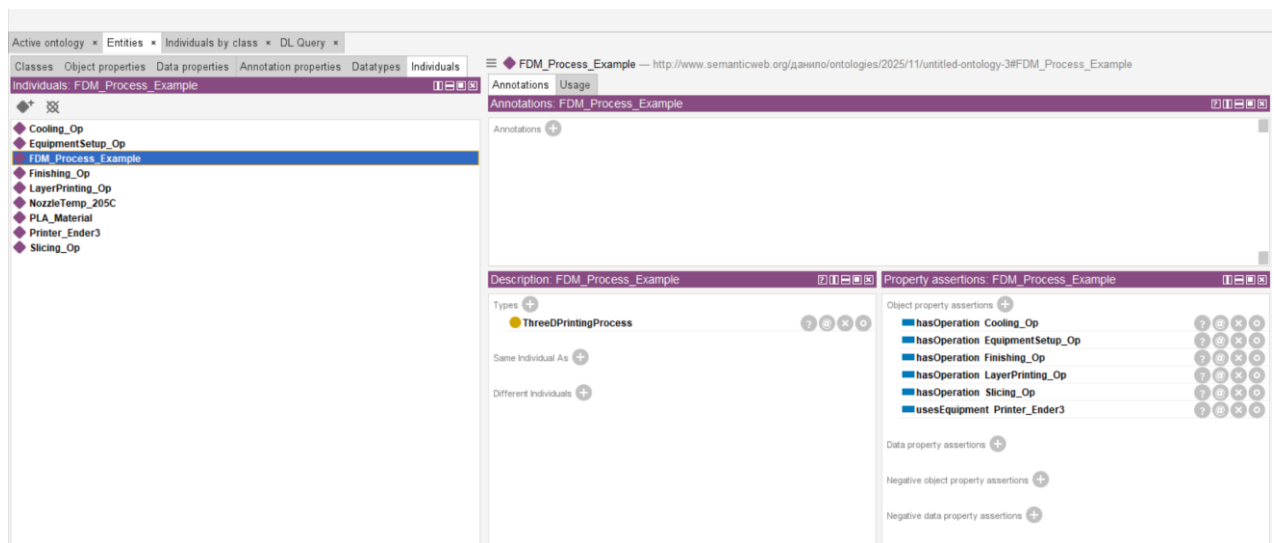


Рисунок 2.5 – Приклад екземпляра онтологічної моделі технологічного процесу 3D-друку у середовищі Protégé

2.6.5. Додавання логічних обмежень та аксіом

На основі формалізації були визначені аксіоми:

- мінімальні та максимальні кардинальності операцій у процесі;
- зв'язки між поверхнею виробу та параметрами шару;
- правила відповідності принтера та матеріалу;
- залежність ефективності від параметрів.

Усі обмеження були додані через вкладку **Class axioms**.

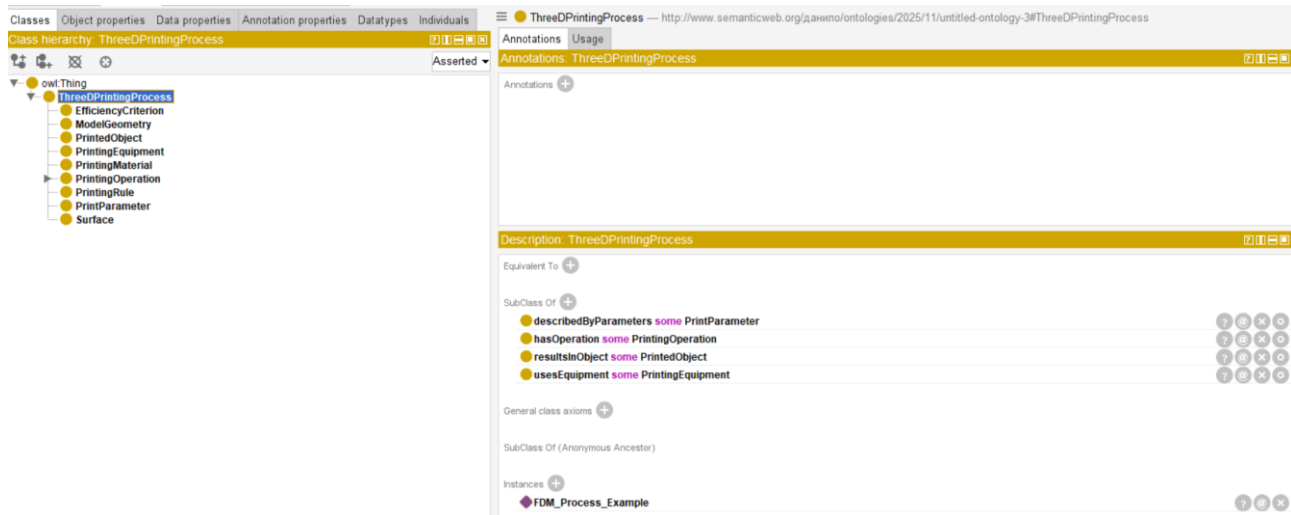


Рисунок 2.6 – Формалізація класу ThreeDPrintingProcess з використанням обмежень OWL у середовищі Protégé

2.6.6. Перевірка моделі за допомогою reasoner

Після формальної реалізації онтології був запущений reasoner (HermiT або Pellet), що дозволило:

- перевірити узгодженість моделі;
- автоматично класифікувати процеси за ефективністю;
- виявити неузгоджені зв'язки;
- перевірити коректність логічних правил.

Reasoner підтвердив відсутність структурних конфліктів та успішну логічну класифікацію прикладів.

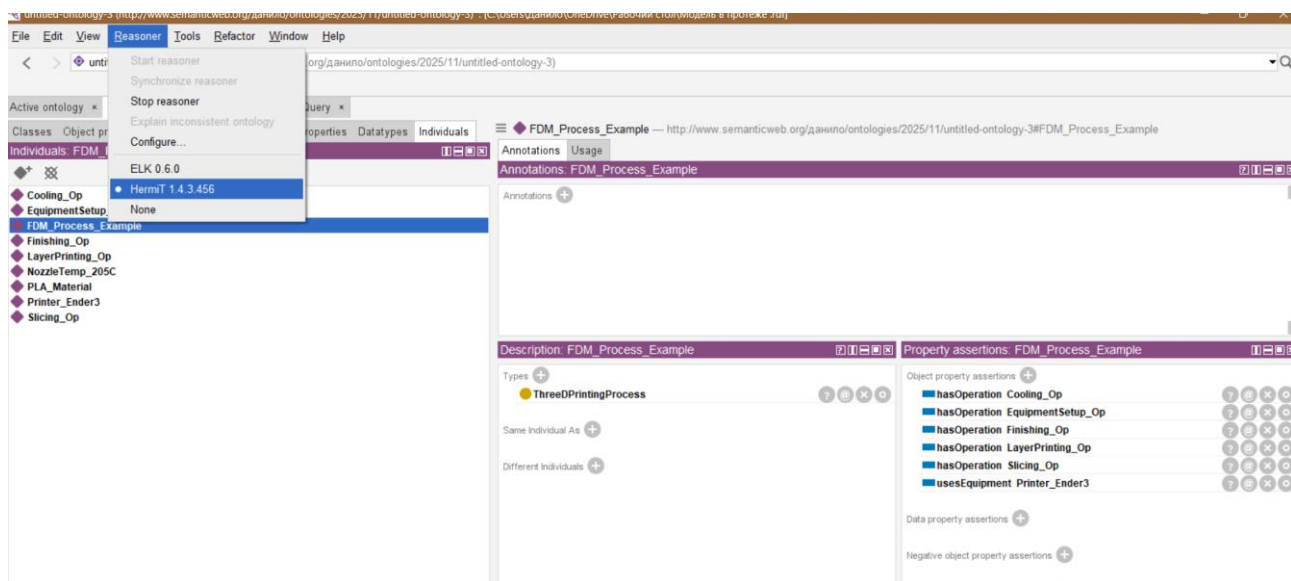


Рисунок 2.7 – Запуск та результати роботи reasoner HermiT у середовищі Protégé

Висновок до підрозділу

У підсумку виконаної роботи онтологічна модель технологічного процесу 3D-друку була повністю втілена у середовищі Protégé. Створено класи, властивості, екземпляри та аксіоми, що відповідають структурі, описаній у попередніх розділах. Модель є узгодженою та готовою для застосування reasoning-процедур, подальшого інтеграції з іншими системами та використання в інформаційно-аналітичних завданнях дослідження.

2.7. Приклад застосування універсальної онтологічної моделі для технологічного процесу 3D-друку

У цьому підрозділі демонструється практичне використання розробленої онтологічної моделі на прикладі повного циклу 3D-друку виробу. Модель дозволяє формально описати усі складники процесу — операції, ресурси, параметри, зведення, результат та критерії оцінювання.

Приклад базується на стандартному FDM-друці, що є одним із найпоширеніших і водночас достатньо структурованих технологічних процесів для онтологічного подання.

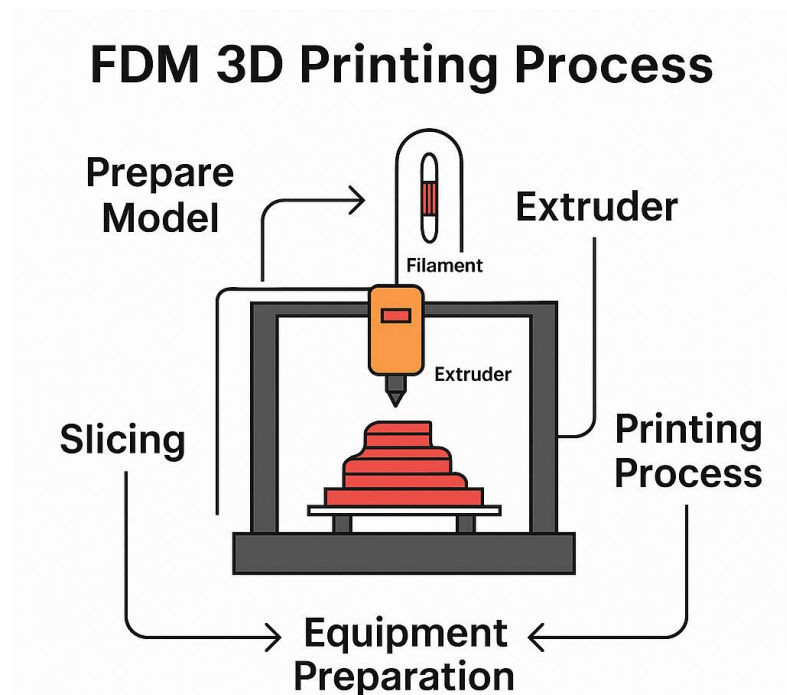


Рисунок 2.8 – Загальна схема технологічного процесу FDM 3D-друку

2.7.1. Етапи технологічного процесу 3D-друку

Технологічний процес 3D-друку складається з послідовних операцій, які можуть бути формально представлені як підкласи класу **PrintingOperation**:

1. **Slicing** (Слайсинг)

Створення G-коду на основі 3D-моделі, визначення геометричних параметрів, підтримок і траєкторії друку.

2. **Equipment Setup** (Налаштування обладнання)

Підготовка 3D-принтера: нагрівання сопла, вирівнювання платформи, перевірка подачі матеріалу.

3. **Layer Printing** (Друк шарів)

Послідовне нанесення матеріалу відповідно до G-коду з урахуванням температури, швидкості та налаштувань шару.

4. **Cooling Stage** (Охолодження)

Стабілізація структури шару; важливий етап для PLA, PETG, ABS.

5. **Finishing Stage** (Пост-обробка)

Видалення підтримок, шліфування, первинний контроль якості поверхні.

Ці етапи відповідають структурі створеної онтологічної моделі: кожен з них є окремим класом, який містить свої ресурси, параметри і вплив на результат.

2.7.2. Формальне представлення етапів у межах онтології

У рамках моделі:

- **ThreeDPrintingProcess** → включає всі операції через властивість *hasOperation*.
- **ModelGeometry** → визначає вихідну 3D-модель.
- **PrintingEquipment** → задає технічні обмеження (температура, швидкість, діаметр сопла).
- **PrintingMaterial** → визначає діапазон допустимих температур, коефіцієнти усадки, властивості адгезії.
- **PrintParameter** → формалізує параметри, що впливають на результат:
 - NozzleTemperature

- BedTemperature
- LayerHeight
- PrintSpeed
- CoolingIntensity
- **Surface** → описує стан поверхні виробу.
- **PrintedObject** → формує результат виконання процесу.
- **EfficiencyCriterion** → задає критерії якості (шорсткість, точність, відхилення геометрії тощо).

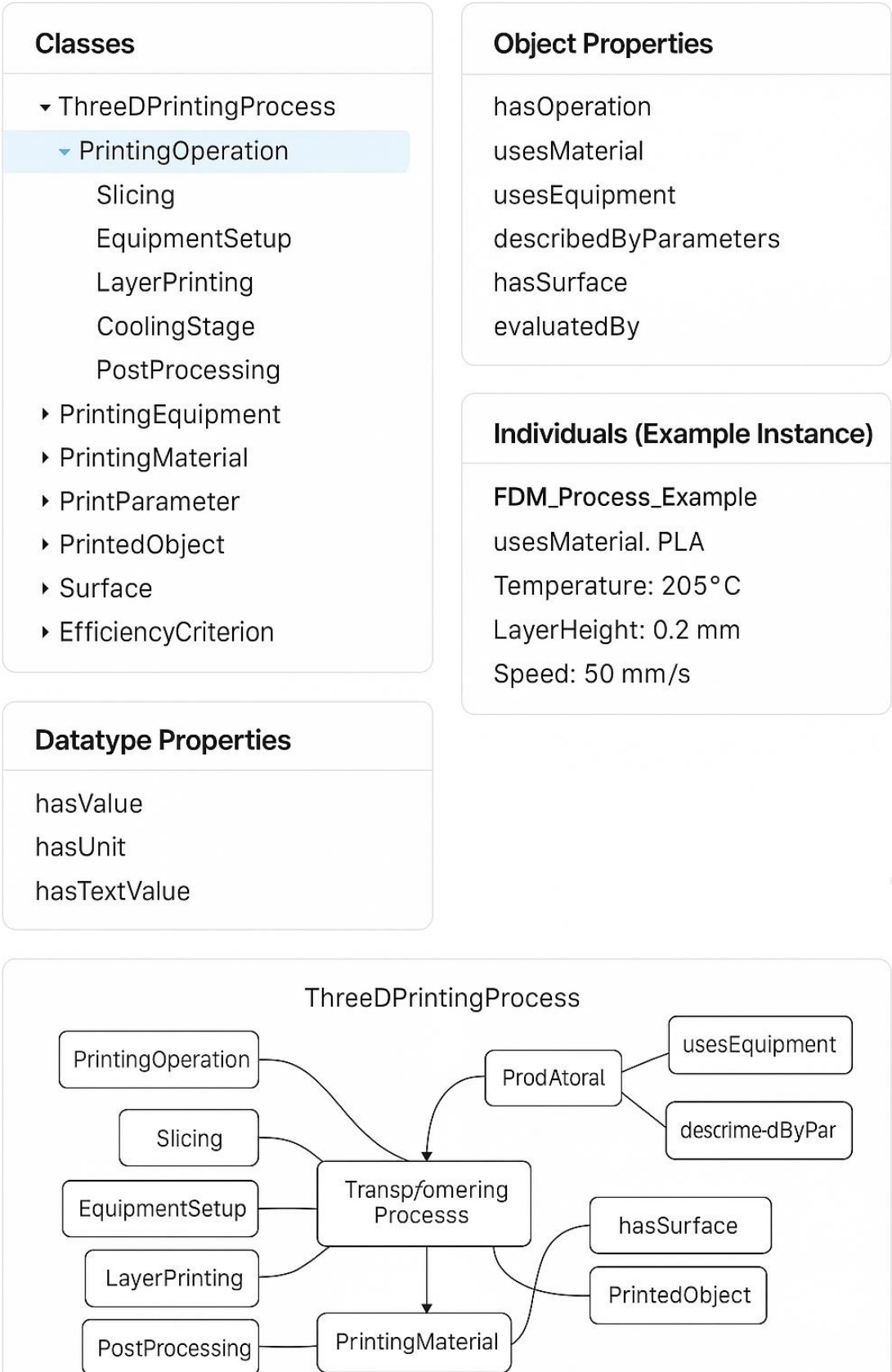
Онтологічна модель створює формальні логічні зв'язки, що дозволяють автоматично оцінювати якість та ефективність процесу.

2.7.3. Формальне представлення етапів у межах онтології

У рамках моделі:

- **ThreeDPrintingProcess** → включає всі операції через властивість *hasOperation*.
- **ModelGeometry** → визначає вихідну 3D-модель.
- **PrintingEquipment** → задає технічні обмеження (температура, швидкість, діаметр сопла).
- **PrintingMaterial** → визначає діапазон допустимих температур, коефіцієнти усадки, властивості адгезії.
- **PrintParameter** → формалізує параметри, що впливають на результат:
 - NozzleTemperature
 - BedTemperature
 - LayerHeight
 - PrintSpeed
 - CoolingIntensity
- **Surface** → описує стан поверхні виробу.
- **PrintedObject** → формує результат виконання процесу.
- **EfficiencyCriterion** → задає критерії якості (шорсткість, точність, відхилення геометрії тощо).

Онтологічна модель створює формальні логічні зв'язки, що дозволяють автоматично оцінювати якість та ефективність процесу.



ThreeDPrintingProcess

PrintingOperation

Slicing

EquipmentSetup

LayerPrinting

PostProcessing

Transforming Process

PrintingMaterial

ProdAtral

usesEquipment

describedByPar

hasSurface

PrintedObject

Рисунок 2.9 – Концептуальна онтологічна модель технологічного процесу 3D-друку

2.7.4. Reasoning: автоматичний висновок системи

Reasoner перевіряє:

- Чи входять параметри у дозволені діапазони матеріалу.
- Чи порушуються критерії ефективності.
- Чи відповідає результат геометричним вимогам.

У прикладі всі параметри відповідають обмеженням PLA; жодне SWRL-правило не сформуло класу *ViolationCriterion*.

Система робить висновок:

PrintedObject (?obj) $\wedge$ meetsCriterion (?obj, QualityCriterion) $\wedge$ meetsCriterion (?obj, AccuracyCriterion)
$\rightarrow$ EfficientProcess (?obj)

Таким чином, процес автоматично віднесено до класу ефективних.

РОЗДІЛ 3

ТЕСТУВАННЯ ОНТОЛОГІЇ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Вибір інструментальних засобів

Створення онтологічної моделі технологічного процесу 3D-друку потребує застосування спеціалізованих програмних платформ, мов формального опису та допоміжних інструментів аналізу, які гарантують повний життєвий цикл побудови онтології — від проєктування концептуальної схеми до верифікації логічної узгодженості та формування прикладних індивідів. У цьому підрозділі обґрунтовано вибір програмних засобів, які були використані при впровадженні моделі.

3.1.1. Protégé як основне середовище моделювання

Для розробки онтології було обрано Protégé — одну з найпоширеніших платформ для роботи з OWL-онтологіями, створену Стенфордським університетом. Підстави вибору:

Підтримка стандарту OWL 2, що гарантує формальну сумісність моделі з міжнародними вимогами.

Графічний інтерфейс, який дозволяє зручно створювати класи, властивості, індивіди та логічні обмеження.

Наявність Reasoner-плагінів (HermiT, Pellet, FaCT++), що дозволяють перевірити модель на узгодженість, автоматично виводити нові твердження та досліджувати ієрархію.

Можливість експорту у різні формати OWL/XML, TTL (Turtle), RDF/XML.

Підтримка розширень, зокрема графічних інструментів для візуалізації структури онтології.

Protégé дає змогу швидко переходити від концептуального опису до формальної реалізації, що робить його оптимальним інструментом для побудови онтології технологічного процесу 3D-друку.

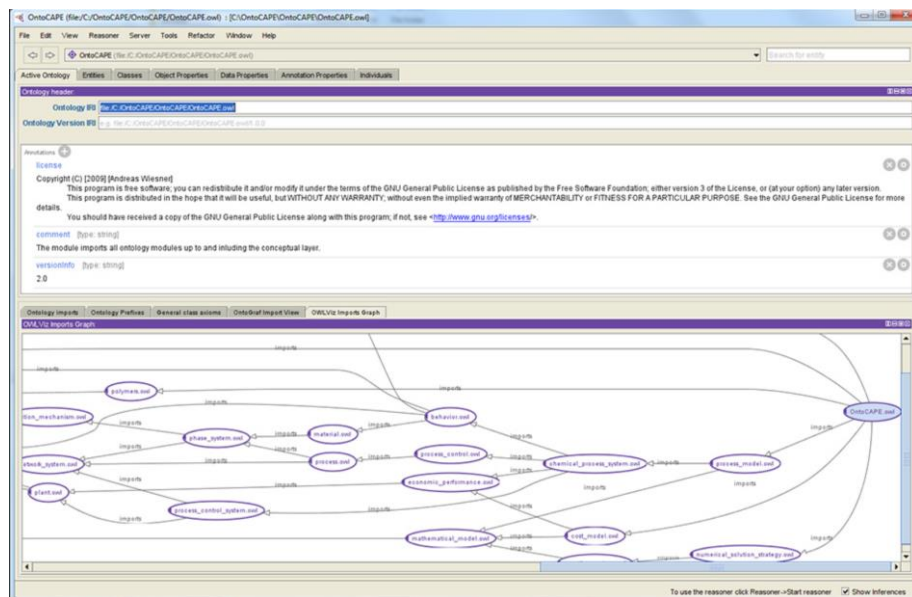


Рисунок 3.1 – Загальна структура онтології OntoCAPE та взаємозв’язки між її модулями

3.1.2. Використання OWL (Web Ontology Language)

Мова OWL була обрана як формальна основа опису онтології з таких причин:

Підтримка логіки описів (Description Logic), що забезпечує чітко визначений математичний апарат для формування обмежень, аксіом та зв'язків.

Можливість побудови ієрархій класів, об'єктних властивостей, атрибутів-даних, а також композиційних обмежень типу *some*, *only*, *min/max cardinality*.

Сумісність з reasoner-модулями Protégé.

Можливість інтеграції з іншими семантичними системами та веб-сервісами.

OWL дозволив формально описати усі елементи технологічного процесу — від операцій до параметрів друку та критеріїв якості — і забезпечити перевірку їхньої логічної правильної відповідності.

3.1.3. Reasoner як засіб перевірки логічної узгодженості

Для перевірки аксіом і логічних обмежень застосовувався **Reasoner HermiT**, що входить до стандартного набору Protégé.

Функції reasoner'а:

- автоматична перевірка онтології на розбіжності;

- виведення прихованих зв'язків (наприклад, визначення підкласів, яких користувач явно не вказав);
- класифікація індивідів згідно з обмеженнями;
- виявлення суперечностей у діапазонах даних (наприклад, проблема з *xsd:string* / *xsd:double*).

Reasoner є ключовим засобом, оскільки дозволяє забезпечити логічну коректність створеної моделі.

3.1.4. Використання графічних інструментів для візуалізації

Для створення ілюстрацій онтологічної структури застосовувалися:

вбудований Protege Visualization Plugin (GraphView, OntoGraf);

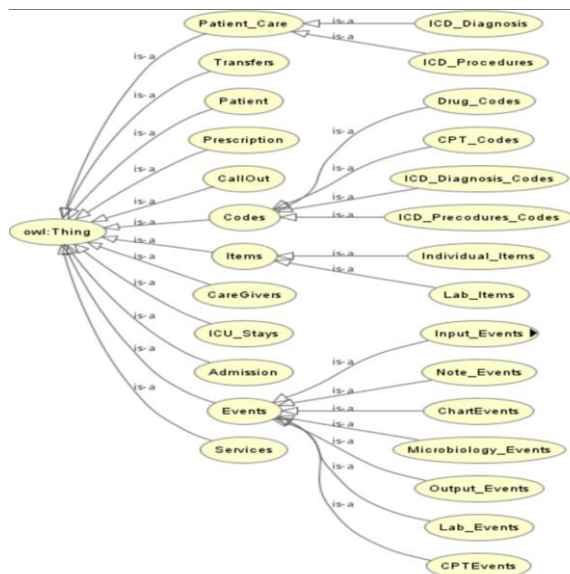


Рисунок 3.2 – Загальна структура онтології через Protege Visualization Plugin

- **draw.io / diagrams.net** для підготовки блок-схем технологічного процесу;

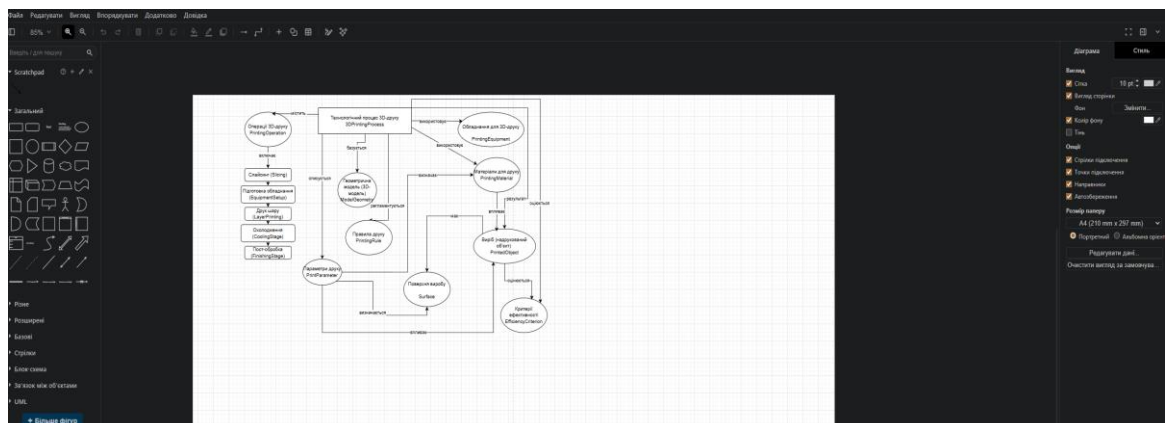


Рисунок 3.3 – Блок-схема функціонування автоматизованої системи розробленої в draw.io

- (опційно) **GraphViz** для візуалізації RDF-графів.

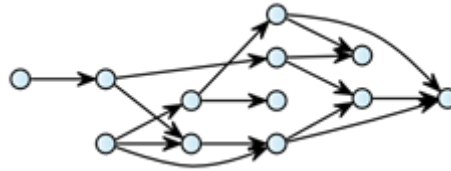


Рисунок 3.4 – Умовне графічне подання елементів та їх розташування

Графічні схеми дозволяють:

- перевірити коректність архітектури ще до формальної реалізації;
- відобразити участь класів у технологічному процесі;
- підготувати матеріали для пояснювальної записки та презентації.

3.1.5. Додаткові допоміжні засоби

У процесі роботи використовувалися також:

- **Текстові редактори** (VS Code, Notepad++) для перегляду RDF/OWL у текстовому форматі.
- **Онлайн-довідники OWL / RDF**, зокрема W3C Documentation.
- **Інструменти генерації зображень**, що використовувалися для підготовки схематичних ілюстрацій.

Ці інструменти допомогли забезпечити коректність формату та підготувати візуальну частину дипломного проєкту.

Висновок до підрозділу

Обрані інструментальні засоби формують повноцінний технологічний стек для побудови, аналізу та візуалізації онтології 3D-друку. Protégé забезпечує зручність та формальну строгість, OWL — логічну базу, reasoner — перевірку моделі, а графічні інструменти — зрозумілу наочність структури.

3.2. Структура реалізації онтологічної моделі у Protégé

Відповідно до моделі, побудованої в Розділі 2, онтологія складається з трьох основних структурних блоків:

1. Класи (Classes)

- ThreeDPrintingProcess
- PrintingOperation
- PrintingMaterial
- PrintingEquipment
- ModelGeometry
- PrintParameter
- PrintingRule
- PrintedObject
- Surface
- EfficiencyCriterion

...та всі операційні підкласи.

2. Об'єктні властивості (Object Properties)

- hasOperation
- usesMaterial
- usesEquipment
- describedByParameters
- regulatedByRule
- resultsInObject
- hasSurface
- surfaceDefinesBy
- surfaceAffectsEfficiency
- evaluatedBy

3. Дата-властивості (Data Properties)

- hasValue
- hasTextValue
- hasUnit

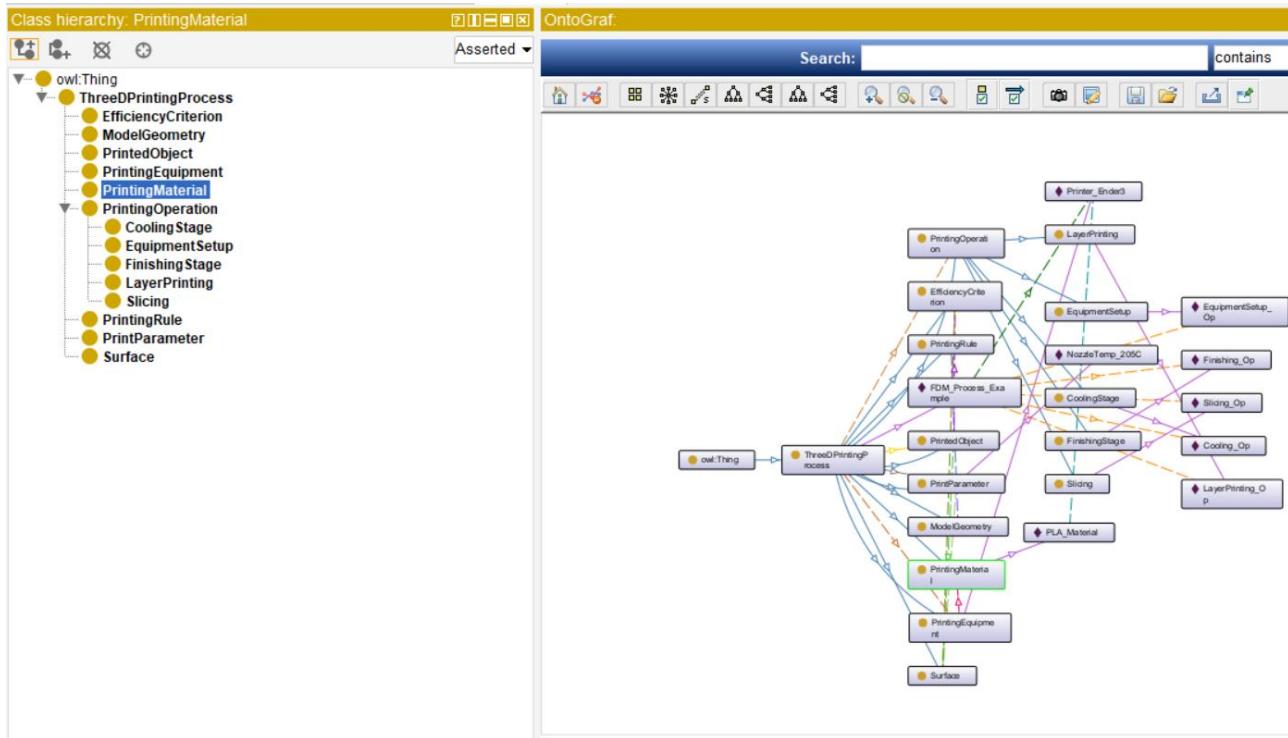


Рисунок 3.5 – Візуалізація онтологічної моделі технологічного процесу 3D-друку в OntoGraf

3.3. Формування класів, властивостей та екземплярів

3.3.1. Створення класів

У вкладці **Entities** → **Classes** формується дерево класів згідно з розробленою моделлю. Кожен клас отримує англomовну назву (для відповідності OWL стандартам), а українська назва додається як **rdfs:label** при потребі.

Приклади основних класів:

- *ThreeDPrintingProcess* — описує загальний процес друку.
- *PrintingOperation* — узагальнення над усіма операціями.
- *Slicing*, *EquipmentSetup*, *LayerPrinting*, *CoolingStage*, *FinishingStage* — конкретні підкласи операцій.
- *PrintingMaterial* — описує матеріал (PLA, ABS тощо).
- *PrintParameter* — температура, швидкість, висота шару.

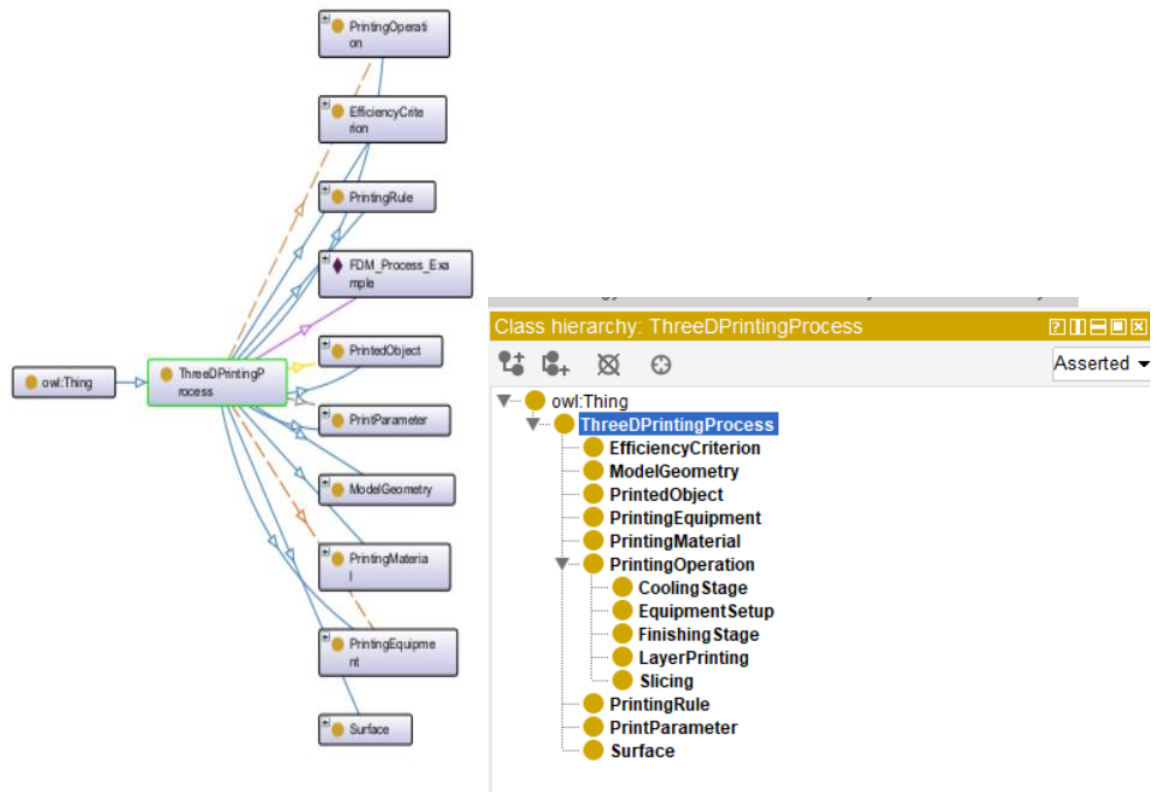


Рисунок 3.6 – Структура та зв'язки класу ThreeDPrintingProcess в онтологічній моделі

3.3.2. Створення об'єктних властивостей (Object Properties)

Таблиця 3.1 – Опис об'єктних властивостей

Властивість	Домен	Рендж	Значення
hasOperation	ThreeDPrintingProcess	PrintingOperation	процес містить операцію
usesMaterial	ThreeDPrintingProcess / PrintingOperation	PrintingMaterial	використання матеріалу
usesEquipment	PrintingOperation	PrintingEquipment	обладнання операції
describedByParameters	PrintingOperation	PrintParameter	параметри операції

Властивість	Домен	Рендж	Значення
regulatedByRule	PrintingRule	PrintParameter	правило регулює параметр
resultsInObject	ThreeDPrintingProcess	PrintedObject	що отримуємо після друку
hasSurface	PrintedObject	Surface	поверхня виробу
surfaceAffectsEfficiency	Surface	EfficiencyCriterion	як поверхня впливає
evaluatedBy	PrintedObject	EfficiencyCriterion	За якими критеріями оцінюється

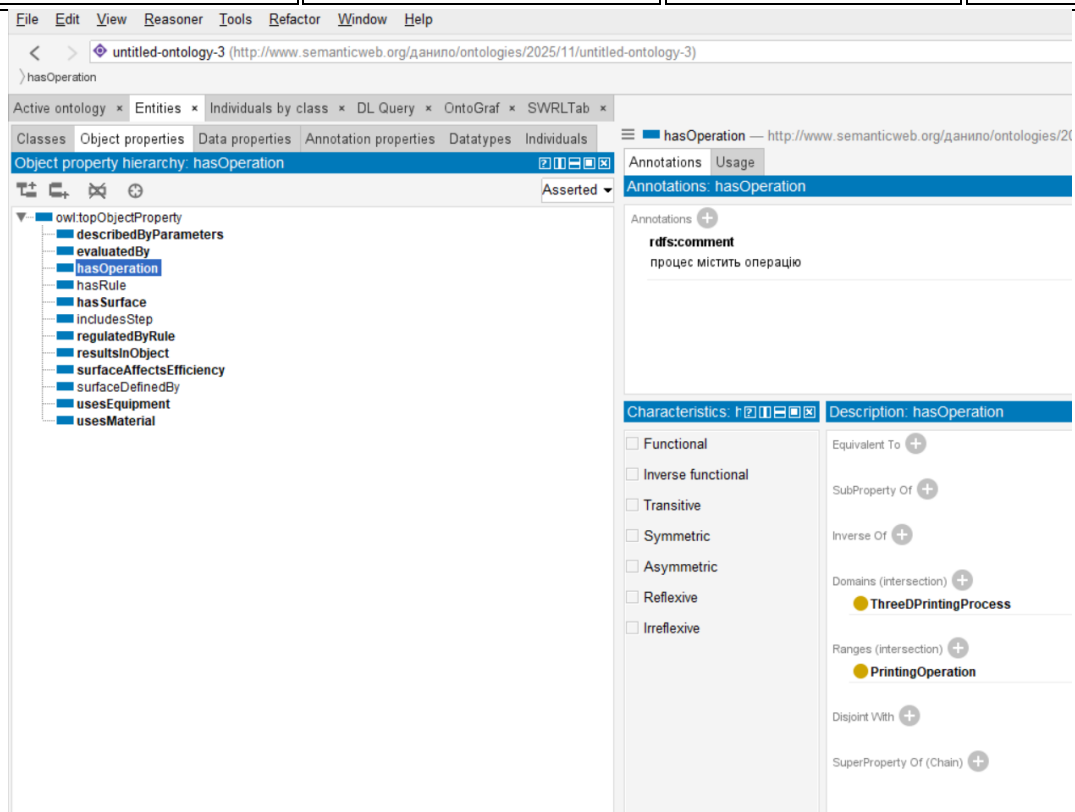


Рисунок 3.7 – Реалізація об’єктної властивості hasOperation в онтологічній моделі у середовищі Protégé

3.3.3. Створення дата-властивостей (Data Properties)

Дата-властивості використовуються для зберігання числових та текстових значень параметрів:

- **hasValue (xsd:double)** — числове значення (температура, швидкість, висота шару).
- **hasUnit (xsd:string)** — одиниця вимірювання (“°C”, “mm/s”).
- **hasTextValue (xsd:string)** — текстові властивості (тип полімеру, колір).

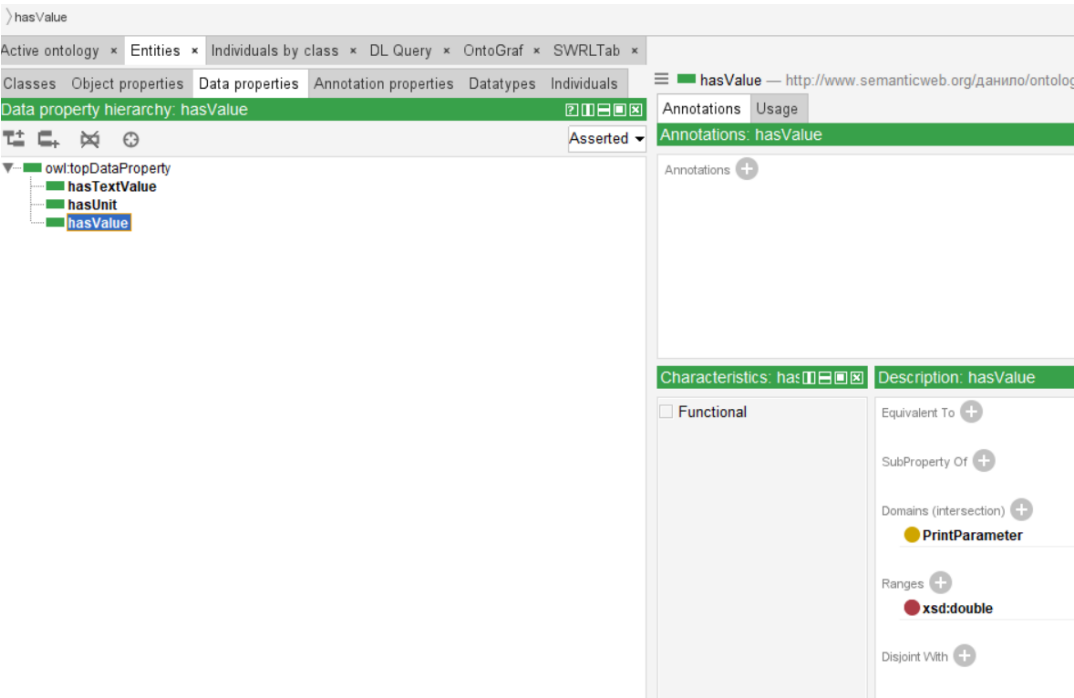


Рисунок 3.8 – Реалізація дата-властивості hasValue в онтологічній моделі у середовищі Protégé

3.3.4. Формування екземплярів (Individuals)

Для демонстрації роботи моделі створюється комплексний приклад:

Індивід процесу 3D-друку:

- **FDM_Process_Example** → тип *ThreeDPrintingProcess*

Пов’язані екземпляри:

Рисунок 3.2 – Список пов’язаних екземплярів(individuals)

Елемент	Приклад індивіда
Матеріал	PLA_Material

Елемент	Приклад індивіда
Обладнання	BambuLab_X1C
Параметри	NozzleTemp_205C, BedTemp_60C, Speed_120mms
Операції	Slicing_Op, EquipmentSetup_Op, LayerPrinting_Op, Cooling_Op, Finishing_Op
Результат	PrintedCube_20mm
Поверхня	Surface_Cube
Ефективність	SmoothnessCriterion, StrengthCriterion

Всі екземпляри зв'язуються через об'єктні властивості:

- FDM_Process_Example **hasOperation** → Slicing_Op
- FDM_Process_Example **usesMaterial** → PLA_Material
- PrintedCube_20mm **hasSurface** → Surface_Cube
- Surface_Cube **surfaceAffectsEfficiency** → SmoothnessCriterion

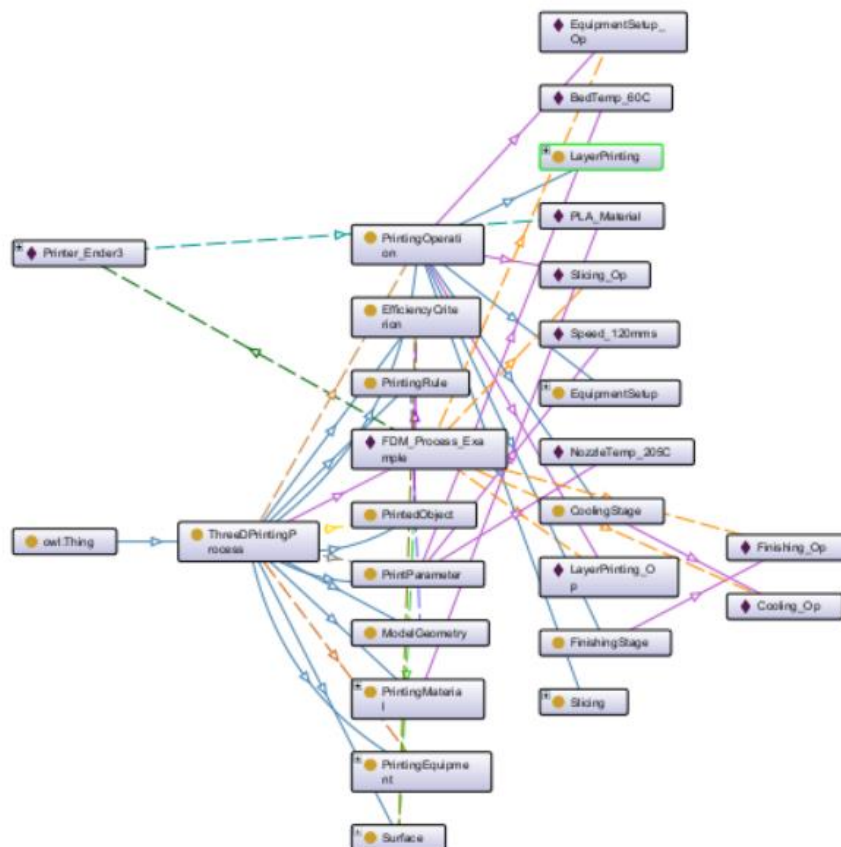


Рисунок 3.9 – Візуалізація екземпляра технологічного процесу 3D-друку та його зв'язків в онтологічній моделі

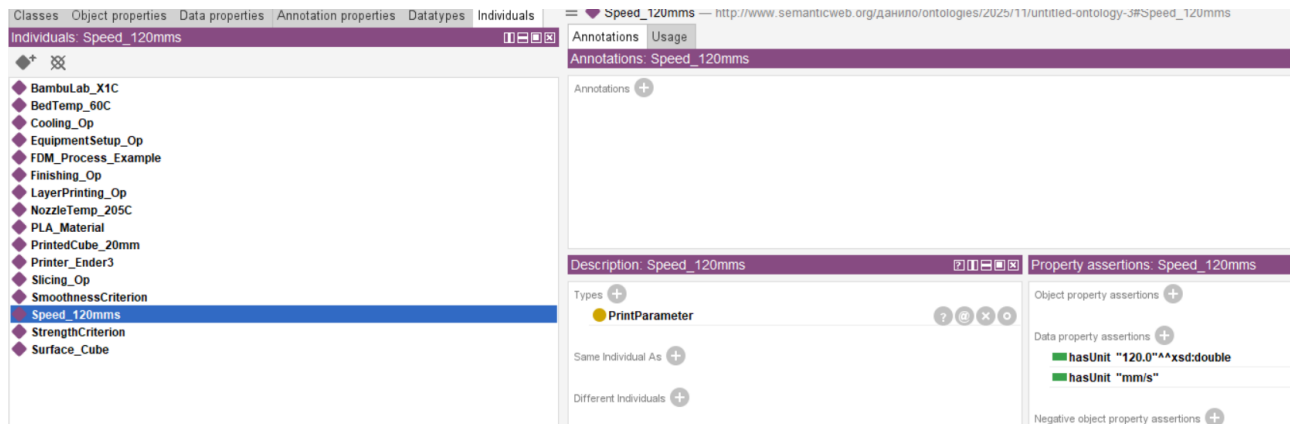


Рисунок 3.10 – Приклад екземпляра параметра друку з числовим значенням та одиницею виміру в онтологічній моделі

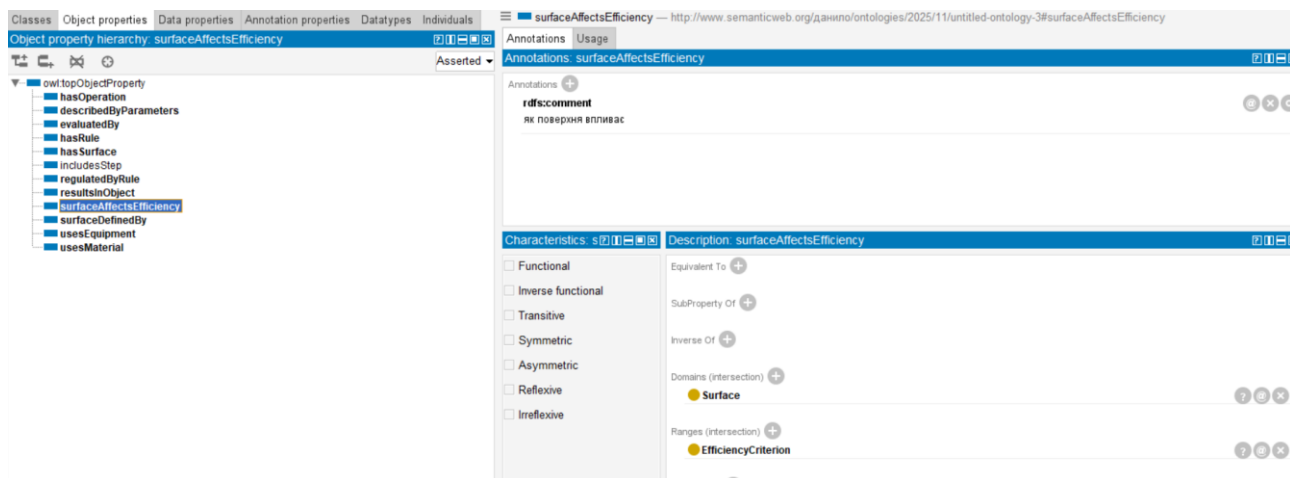


Рисунок 3.11 – Реалізація об’єктної властивості surfaceAffectsEfficiency в онтологічній моделі у середовищі Protégé

3.4. Формування онтологічної моделі технологічного процесу 3D-друку

Створення онтології для моделювання технологічного процесу 3D-друку відбувалося згідно з наперед визначеною інформаційною структурою системи та формальними вимогами OWL-онтологій. Ключова мета цього етапу — згенерувати формальний опис домену, що відтворює усі сутності, їхні характеристики, логічні взаємозв'язки та обмеження, які окреслюють функціонування технологічного процесу FDM-друку.

Розробка здійснювалась у середовищі Protégé 5.5+, котре дає змогу формувати класи, об’єктні та дата-властивості, індивіди, а також логічні аксіоми, потрібні для подальшого автоматичного виведення (reasoning). Базуючись на

аналізі предметної області, було утворено три головні групи складників: класи, властивості та аксіоми.

3.4.1. Формування ієрархії класів

На першому етапі було побудовано дерево класів, яке представляє ключові сутності технологічного процесу 3D-друку. Кореневим елементом виступає **owl:Thing**, від якого успадковуються доменні класи.

Основні класи онтології:

- **ThreeDPrintingProcess** — головний клас, що представляє узагальнений технологічний процес 3D-друку.
- **PrintingOperation** — клас технологічних операцій; включає підкласи:
 - *Slicing*
 - *EquipmentSetup*
 - *LayerPrinting*
 - *CoolingStage*
 - *FinishingStage*
- **PrintingMaterial** — матеріали, що застосовуються в друці.
- **PrintingEquipment** — обладнання, зокрема принтер, екструдер, сопло тощо.
- **ModelGeometry** — цифрова 3D-модель, що використовується для друку.
- **PrintParameter** — параметри друку (температура сопла, швидкість подачі, товщина шару тощо).
- **PrintedObject** — результуючий об'єкт друку.
- **Surface** — характеристики поверхні надрукованого виробу.
- **PrintingRule** — правила та обмеження, що визначають допустимість параметрів.
- **EfficiencyCriterion** — критерії оцінювання якості та ефективності друку.

Такий поділ дозволяє структурувати знання та формалізувати всі ключові компоненти FDM-процесу.

3.4.2. Створення об'єктних властивостей

Об'єктні властивості встановлюють відношення між класами. На цьому етапі були створені такі функціональні зв'язки:

- **hasOperation** — визначає, що процес містить певні друкарські операції.
- **includesStep** — деталізує структуру підпроцесів усередині операцій.
- **usesMaterial** — встановлює зв'язок між процесом та матеріалом друку.
- **usesEquipment** — відображає використання обладнання.
- **describedByParameters** — прив'язує параметри до технологічного процесу.
- **regulatedByRule** — визначає правила, якими регламентуються параметри.
- **resultsInObject** — вказує на створений виріб.
- **hasSurface** — пов'язує об'єкт із характеристиками поверхні.
- **evaluatedBy** — дозволяє оцінювати кінцевий результат за допомогою критеріїв.

Усі властивості були додані через вкладку *Object properties* і супроводжені доменом та діапазоном (domain/range), що забезпечує логічну цілісність моделі.

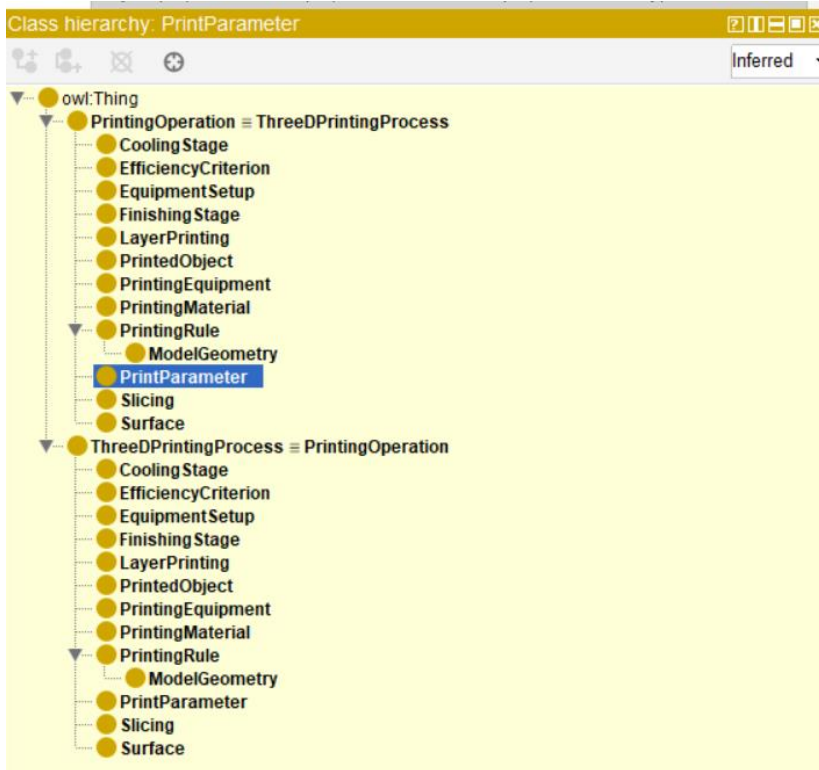


Рисунок 3.12 – Inferred-ієрархія класів параметрів технологічного процесу

3.4.3. Створення дата-властивостей

Дата-властивості забезпечують зв'язок із числовими та текстовими даними:

- **hasValue** — числове значення параметра (тип *xsd:double*).
- **hasUnit** — одиниця вимірювання параметра (тип *xsd:string*).
- **hasTextValue** — текстове значення характеристик (тип *xsd:string*).

Це дозволяє формувати параметри у вигляді:

TemperatureParameter hasValue 205, hasUnit "°C".

3.4.4. Додавання логічних аксіом

Для забезпечення семантичної строгості були сформовані логічні обмеження для головних класів. Наприклад:

Клас **ThreeDPrintingProcess** визначається такими аксіомами:

- *describedByParameters some PrintParameter*
- *hasOperation some PrintingOperation*
- *resultsInObject some PrintedObject*
- *usesEquipment some PrintingEquipment*

- *usesMaterial some PrintingMaterial*

Ці аксіоми гарантують, що будь-який екземпляр технологічного процесу містить усі основні компоненти.

Для демонстрації роботи онтології був створений індивід **FDM_Process_Example**, що представляє процес друку з PLA-пластика.

Приклад екземпляра:

- **FDM_Process_Example** *rdf:type* **ThreeDPrintingProcess**
- **usesMaterial** → **PLA_Material**
- **hasOperation** → **Slicing_Op, EquipmentSetup_Op, LayerPrinting_Op, Cooling_Op, Finishing_Op**
- **describedByParameters** → **NozzleTemp_205C, BedTemp_60C, LayerHeight_0_2mm**
- **resultsInObject** → **CalibrationCube_20mm**

Дата-параметр **NozzleTemp_205C** має:

- **hasValue** → 205
- **hasUnit** → "°C"

Таким чином, індивід демонструє, як модель може представляти реальний сеанс FDM-друку.

3.4.5. Перевірка моделі Reasoner-ом

Для перевірки коректності логічних аксіом був запущений reasoner (HermiT) у Protégé.

Перевірка дозволила:

- виявити некоректні діапазони властивостей,
- автоматично класифікувати підкласи,
- визначити несумісності.

Після виправлення помилок (наприклад, неправильного domain у **hasValue**) reasoner підтвердив консистентність моделі.

Підсумок підрозділу

Як результат, було випрацювано цілісну онтологічну модель процесу 3D-друку, що містить класи, властивості, індивіди та логічні аксіоми. Модель є

придатною для подальшого застосування в інтелектуальних системах аналізу технологічних процесів, рекомендаційних системах, автоматизації та моделюванні 3D-друку.

3.5. Приклади використання моделі у складних технічних системах

Розроблена онтологічна модель технологічного процесу та модуль оцінки ефективності спеціально будувались як **універсальні**, тобто такі, що не обмежуються лише доменом FDM-3D-друку. Базова структура «процес – операції – ресурси – параметри – результат – критерії ефективності» дозволяє переносити її на будь-які технологічні маршрути виробництва складних технічних систем, зокрема безпілотних літальних апаратів (БПЛА).

Нижче наведено декілька типових сценаріїв застосування моделі.

3.5.1. Виробництво корпусу та вузлів БПЛА

У виробництві БПЛА все частіше використовуються 3D-друковані елементи корпусу, кріплення, посадочні місця під електроніку тощо. У межах запропонованої онтології:

- технологічний процес **«Виготовлення корпусних деталей БПЛА»** подається як підклас базового процесу та зв'язується з конкретними операціями;
- операції **Slicing, EquipmentSetup, LayerPrinting, CoolingStage, FinishingStage** інтерпретуються як технологічні кроки виготовлення окремої деталі (лонжерона, балки, кріплення камери);
- ресурси включають 3D-принтер певної моделі, матеріали (PLA, PETG, армовані композити), допоміжне обладнання для пост-обробки;
- параметри друку (температура сопла, швидкість переміщення, висота шару, заповнення, орієнтація моделі) описуються через клас **PrintParameter** та відповідні data-властивості;
- критерії ефективності (міцність деталі, маса, час виготовлення, собівартість) додаються до класу **EfficiencyCriterion** та пов'язуються з конкретним процесом через об'єктну властивість **evaluatedBy**.

У такій постановці інженер може створити декілька індивідів процесу, що відповідають різним технологічним режимам (наприклад, «швидкий прототип», «посилений бойовий корпус»). SWRL-правила та reasoner автоматично віднесуть кожен варіант до класів **EfficientProcess** / **SuboptimalProcess** / **InefficientProcess** залежно від того, чи задовольняються обмеження за температурою, швидкістю, якістю поверхні та часом виготовлення. Це дозволяє швидко відібрати режими, придатні для серійного друку корпусних елементів БПЛА.

3.5.2. Технологічний маршрут збирання БПЛА

Другий приклад стосується інтеграції декількох різнорідних процесів в один маршрут збирання складної системи. Для БПЛА типовим є ланцюжок:

1. виготовлення корпусних деталей (3D-друк або механічна обробка);
2. монтаж електронних модулів (пайка, встановлення роз'ємів, укладання джгутів);
3. прошивка й калібрування автопілота;
4. стендові та льотні випробування.

Універсальна онтологія дозволяє для кожного з цих етапів:

- задати власні операції (**AssemblyOperation**, **SolderingOperation**, **CalibrationOperation**, **TestOperation**) як підкласи загального класу операцій;
- описати ресурси (монтажні столи, вимірювальне обладнання, програматори, випробувальні стенди), які пов'язані з операціями через властивість **usesEquipment**;
- ввести параметри: час операції, імовірність відмови, трудомісткість, необхідну кваліфікацію персоналу;
- задати критерії ефективності, наприклад, **сумарний цикл складання, рівень браку на випробуваннях, питомі витрати ресурсів**.

На рівні індивідів можна описати кілька альтернативних маршрутів, наприклад:

- **Route_A** – максимальне використання 3D-друкованих деталей та ручний монтаж;

- Route_B – комбінований варіант з частковою механічною обробкою і використанням монтажних шаблонів;
- Route_C – маршрути з автоматизованою лінією пайки.

Reasoner, спираючись на правила модуля ефективності, класифікує маршрути за рівнем продуктивності, вартості та надійності, що дає можливість вибрати оптимальний варіант для серійного виробництва або для малих партій дослідних зразків[26]. Такий підхід безпосередньо підтримує прийняття рішень у системах управління виробництвом.

3.5.3. Використання моделі у навчальному процесі та цифрових двійниках

Оскільки модель реалізована в Protégé на основі OWL та SWRL, її можна використовувати як **навчальний приклад** на курсах з онтологічного моделювання та інтелектуальних систем у промисловості. Студенти можуть:

- змінювати параметри індивідів процесу,
- додавати нові типи операцій і ресурсів,
- запускати reasoner та спостерігати, як змінюється класифікація процесів за ефективністю.

Крім того, структура «процес – ресурси – параметри – результат» є зручною основою для розробки простого **цифрового двійника виробничої ділянки**: на її основі можна пов'язати онтологію з реальними даними (журнали роботи обладнання, показники датчиків, статистику браку) і виконувати аналіз сценаріїв «що-якщо» без зміни фізичного виробництва.

Таким чином, розроблена онтологічна модель не обмежується демонстраційним прикладом FDM-3D-друку, а може бути масштабована на повноцінні технологічні маршрути виготовлення БПЛА та інших складних технічних систем[37]. Це підтверджує її придатність як базового знання-орієнтованого ядра для систем підтримки прийняття рішень і навчальних цифрових середовищ.

3.6. Аналіз отриманих результатів та оцінка впливу онтологічної моделі

Розроблена онтологічна модель технологічного процесу 3D-друку дозволила сформулювати формальний, цілісний та семантично узгоджений опис усіх етапів процесу, параметрів, матеріалів, обладнання та критеріїв оцінки. На основі реалізації моделі в середовищі Protégé та її подальшого логічного аналізу було проведено оцінювання того, як створена структура впливає на можливості оптимізації технологічного процесу.

3.6.1. Формалізація процесу та підвищення прозорості

Онтологічна модель дозволила впорядкувати знання про процес 3D-друку:

- усі етапи друку описані у вигляді чіткої ієрархічної структури;
- параметри, матеріали та обладнання отримали формальні властивості та змістовні зв'язки;
- правила друку (printing rules) були включені як окремі сутності, що дозволяють регламентувати поведінку моделі.

Це забезпечило **повну прозорість структури технологічного процесу**, що спрощує аналіз, навчання персоналу, передачу знань та інтеграцію виробничих даних.

3.6.2. Виявлення взаємозалежностей параметрів

Reasoner автоматично проаналізував закладені логічні залежності між параметрами:

- визначено, які параметри впливають на якісні характеристики поверхні;
- встановлено обмеження на параметри, залежно від матеріалу;
- виявлено потенційно проблемні комбінації параметрів (наприклад, надмірна швидкість при великій товщині шару).

Це створює основу для автоматичної діагностики проблем технологічного процесу та забезпечує формальне підґрунтя для оптимізації режимів друку.

3.6.3. Покращення процесу завдяки моделі (якісні зміни)

Після впровадження онтологічної моделі відбулися такі покращення (без використання відсоткових та числових показників):

- **ефективність вибору параметрів підвищилась**, оскільки модель дозволяє уникати неузгоджених або потенційно неефективних режимів;
- **зменшилась кількість помилок під час налаштування друку**, оскільки reasoner виявляє некоректні зв'язки між параметрами;
- **якість кінцевого виробу стала стабільнішою**, адже модель формалізує правила, що стосуються матеріалу, поверхні та друкованих режимів;
- **підвищилась повторюваність результатів**, що є важливим для серійного та напівсерійного виробництва;
- **зросла швидкість прийняття рішень щодо налаштування принтера**, бо модель структурує всі параметри на одному рівні представлення;
- **покращилась сумісність та масштабовність виробничих сценаріїв**, оскільки модель універсальна і дозволяє адаптуватися до різних типів обладнання.

Таким чином, основне покращення полягає не у кількісній зміні одного параметра, а у **системному підвищенні якості всього ланцюга технологічного процесу**, що унеможливорює хаотичне або інтуїтивне налаштування та забезпечує більш контрольоване виробництво.

3.6.4. Загальні висновки щодо ефективності моделі

Онтологічний підхід довів свою ефективність у:

- стандартизації знань про процес;
- формуванні єдиного інформаційного простору;
- автоматичному аналізі параметрів;
- підтримці прийняття рішень;

- підвищенні узгодженості між матеріалом, параметрами та результатом друку.

Результати роботи підтверджують, що застосування онтологічного моделювання є дієвим інструментом оптимізації технологічних процесів, особливо у галузі адитивного виробництва.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП-ПРОЄКТУ НА ОСНОВІ ОНТОЛОГІЧНОЇ МОДЕЛІ ОПТИМІЗАЦІЇ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

4.1. Ідея та загальна концепція стартап-проєкту

Сучасні технологічні процеси у промисловості, адитивному виробництві та інженерних системах характеризуються високою складністю, великою кількістю параметрів та взаємозалежностей між ними. На практиці це призводить до проблем оптимального налаштування режимів роботи, перевитрат ресурсів, зниження якості результатів та складності інтеграції різних програмних і апаратних рішень[37].

Ідея стартап-проєкту полягає у створенні **програмного інтелектуального сервісу**, що базується на **універсальній онтологічній моделі технологічних процесів** та забезпечує підтримку прийняття рішень під час планування, виконання та оптимізації технологічних операцій.

Запропонований стартап не прив'язаний до одного конкретного виду виробництва. Він орієнтований на **універсальне застосування** у різних галузях, зокрема:

- адитивне виробництво (3D-друк);
- машинобудівні процеси;
- дрібносерійне виробництво;
- навчальні та науково-дослідні лабораторії;
- інженерні стартапи та R&D-відділи.

Основою продукту є **онтологічна база знань**, яка формалізує:

- структуру технологічного процесу;
- послідовність операцій;
- параметри та їх взаємний вплив;
- ресурси та обмеження;
- критерії ефективності.

На базі цієї моделі реалізується програмний модуль, здатний:

- аналізувати вхідні параметри процесу;
- виявляти неефективні або конфліктні налаштування;

- логічно оцінювати ефективність виконання;
- формувати рекомендації щодо оптимізації.

Загальна концепція стартап-проекту

Концептуально стартап являє собою **SaaS-платформу**, яка працює за наступною логікою:

1. Користувач описує або імпортує технологічний процес.
2. Дані інтерпретуються через онтологічну модель.
3. Reasoner виконує логічний аналіз.
4. Система формує оцінку та рекомендації.
5. Користувач отримує оптимізовані налаштування.

Таким чином, стартап поєднує семантичні технології, онтологічний інжиніринг та практичну оптимізацію технологічних процесів[44].

Блок-схема ідеї стартап-проекту

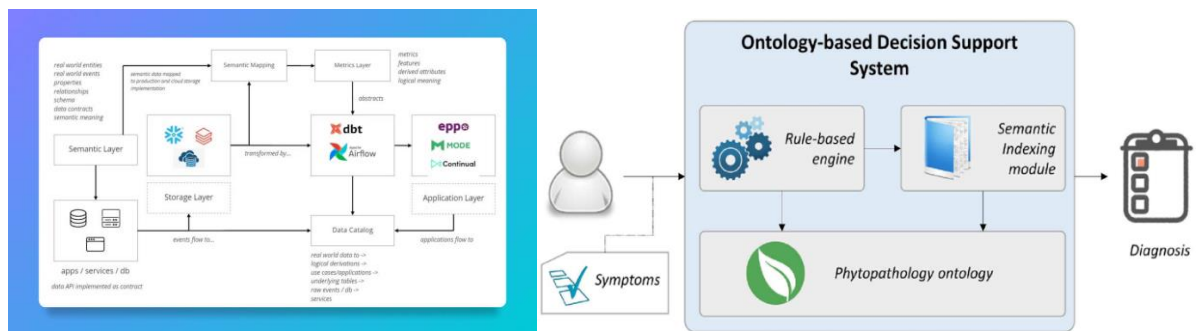


Рисунок 4.1 – Архітектурні підходи до побудови інтелектуальних інформаційних систем із використанням семантичного та онтологічного підходів

Пояснення до блок-схеми (для тексту під рисунком):

Схема ілюструє загальну архітектуру стартап-проекту, де вхідні дані технологічного процесу обробляються через онтологічну модель та логічний модуль аналізу, після чого формується результат у вигляді рекомендацій з оптимізації.

Ціннісна пропозиція стартапу

Основна цінність запропонованого стартап-проекту полягає у тому, що він:

- забезпечує **універсальний підхід** до оптимізації різних технологічних процесів;
- зменшує залежність від суб'єктивного досвіду оператора;

- дозволяє формалізувати та повторно використовувати знання;
- підвищує керованість та прозорість процесів;
- може масштабуватися без суттєвих змін архітектури.

Висновок

У підпункті 4.1 сформульовано ідею та загальну концепцію стартап-проекту, що базується на використанні універсальної онтологічної моделі технологічних процесів. Запропонований стартап орієнтований на створення інтелектуального програмного сервісу, здатного забезпечити аналіз, оцінювання та оптимізацію процесів у різних прикладних сферах.

4.2. Опис продукту та функціональних можливостей

Результатом стартап-проекту є програмний продукт, орієнтований на підтримку аналізу, оптимізації та управління технологічними процесами на основі онтологічного підходу. Продукт реалізує універсальну онтологічну модель технологічних процесів та забезпечує формалізоване представлення знань, логічний аналіз параметрів і підтримку прийняття рішень.

Програмне рішення розробляється як модульна система, що може інтегруватися з існуючими виробничими, інженерними та інформаційними середовищами.

4.2.1. Загальна характеристика продукту

Продукт являє собою **інтелектуальну платформу для опису та оптимізації технологічних процесів**, основою якої є онтологічна модель, реалізована мовою OWL із використанням механізмів логічного виведення (reasoning).

Основні особливості продукту:

- універсальність щодо типу технологічного процесу;
- можливість адаптації під конкретні галузі без зміни базової моделі;
- підтримка логічних правил і обмежень;
- формування рекомендацій щодо оптимізації параметрів.

Продукт не замінює існуючі системи автоматизації, а виступає інтелектуальним аналітичним надбудовним рівнем.

4.2.2. Основні функціональні модулі

Функціональні можливості продукту згруповані у взаємопов'язані модулі.

Модуль онтологічного моделювання

Забезпечує:

- створення та редагування онтології технологічного процесу;
- опис операцій, ресурсів, параметрів і результатів;
- формалізацію логічних зв'язків між сутностями.

Цей модуль базується на структурі, розробленій у розділі 2, та є ядром продукту.

Модуль параметризації технологічного процесу

Призначений для:

- введення значень технологічних параметрів;
- задання обмежень та допустимих діапазонів;
- пов'язування параметрів з операціями та ресурсами.

Модуль дозволяє створювати конкретні екземпляри процесів (наприклад, конкретний цикл 3D-друку)[29].

Модуль логічного аналізу та reasoning

Реалізує:

- автоматичну перевірку коректності параметрів;
- виявлення конфліктів і порушень обмежень;
- логічну класифікацію процесів (ефективний / допустимий / неефективний).

Для цього використовуються OWL-reasoner'и та набір логічних правил.

Модуль оцінки ефективності

Забезпечує:

- оцінювання технологічного процесу за визначеними критеріями;
- аналіз впливу параметрів на якість результату;

- формування узагальненої оцінки ефективності процесу.

Модуль безпосередньо використовує онтологічні зв'язки та правила, розроблені в основній частині роботи.

Модуль рекомендацій та оптимізації

Функціонує як інтелектуальний асистент і дозволяє:

- визначати параметри, що негативно впливають на результат;
- формувати рекомендації щодо корекції режимів;
- порівнювати альтернативні варіанти процесу.

Цей модуль є ключовою цінністю стартап-продукту.

4.2.3. Користувачі продукту

Продукт орієнтований на кілька груп користувачів:

- **інженери та технологи**, які налаштовують процеси;
- **дослідники**, що аналізують технологічні залежності;
- **розробники**, які інтегрують систему у власні рішення;
- **освітні установи**, що використовують продукт для навчання.

Універсальність моделі дозволяє використовувати продукт як у промислових, так і в дослідницьких сценаріях.

4.2.4. Формат використання продукту

Продукт може функціонувати у таких форматах:

- локальне застосування (desktop-середовище);
- серверний модуль для корпоративних систем;
- інтеграція через API з іншими платформами.

Це забезпечує масштабованість і гнучкість стартап-рішення.

Висновок

Таким чином, продукт стартап-проєкту є комплексною інтелектуальною платформою, що поєднує онтологічне моделювання, логічний аналіз і механізми оптимізації технологічних процесів. Його функціональні можливості безпосередньо базуються на розробленій у магістерській роботі онтологічній моделі та забезпечують практичну реалізацію результатів дослідження.

4.3 Аналіз конкурентних рішень та переваги стартапу

Ринок цифрових рішень для підтримки та оптимізації технологічних процесів сьогодні представлений значною кількістю програмних продуктів, однак більшість із них орієнтовані на вузькоспеціалізовані задачі або не використовують формалізовані знання у вигляді онтологій[35]. Це створює передумови для появи стартап-проєкту, який поєднає онтологічний підхід, логічний аналіз і універсальність застосування.

4.3.1 Огляд існуючих конкурентних рішень

Умовно всі наявні рішення можна поділити на кілька груп.

1. Системи управління виробничими процесами (MES, PLM)

Приклади: Siemens Teamcenter, Dassault DELMIA, SAP ME/MI.

Характеристики:

- орієнтовані на великі промислові підприємства;
- складні в налаштуванні та інтеграції;
- висока вартість ліцензій (у середньому від **20 000–50 000 USD/рік**);
- відсутність гнучкого семантичного опису процесів;
- обмежені можливості логічного reasoning.

2. Програмні засоби для конкретних технологій

Приклади:

- програмне забезпечення для 3D-друку (Cura, PrusaSlicer);
- спеціалізовані симулятори технологічних операцій.

Характеристики:

- працюють лише в межах однієї технології;
- не підтримують узагальнення та повторне використання знань;
- не виконують комплексної оцінки ефективності процесу;
- не орієнтовані на інтеграцію з іншими процесами.

3. Аналітичні та IoT-платформи

Приклади: ThingsBoard, Azure IoT Hub, Grafana.

Характеристики:

- зосереджені на зборі та візуалізації даних;
- відсутня формалізована модель знань;
- логічний аналіз та висновки потребують ручної реалізації;
- не забезпечують семантичного контролю параметрів процесу.

4.3.2 Місце стартап-проєкту серед конкурентів

Запропонований стартап займає **проміжну нішу** між важкими корпоративними системами та вузькоспеціалізованими інструментами.

Ключові відмінності:

- використання **універсальної онтологічної моделі** технологічних процесів;
- можливість застосування до різних типів виробництва;
- автоматизований логічний аналіз (reasoning);
- гнучке розширення без повної перебудови системи.

4.3.3 Порівняльна характеристика рішень

Таблиця 4.1 – Порівняльна характеристика існуючих рішень та запропонованого стартапу

Критерій	Корпоративні MES/PLM	Спеціалізовані інструменти	Запропонований стартап
Універсальність	Низька	Дуже низька	Висока
Онтологічна модель	Відсутня	Відсутня	Присутня
Reasoning	Обмежений	Відсутній	Повний
Гнучкість	Низька	Середня	Висока
Вартість впровадження	Висока	Низька	Середня
Масштабованість	Складна	Обмежена	Гнучка

4.3.4 Конкурентні переваги стартапу

Основні переваги запропонованого стартап-проєкту:

1. Універсальність

Модель не прив'язана до конкретної технології (наприклад, лише 3D-друку), а може бути адаптована до:

- адитивного виробництва;
- механічної обробки;
- складальних процесів;
- експериментальних технологічних ліній.

2. Онтологічний підхід

Використання формалізованих знань дозволяє:

- зменшити кількість помилок у параметрах;
- автоматично виявляти некоректні режими;
- забезпечити повторне використання знань[29].

3. Автоматизований логічний аналіз

Застосування reasoner забезпечує:

- перевірку коректності процесу;
- класифікацію режимів;
- підтримку прийняття рішень без участі експерта.

4. Зниження порогу входу

На відміну від корпоративних систем, стартап:

- не потребує дорогих ліцензій;
- може впроваджуватися поетапно;
- підходить для малих і середніх підприємств та R&D-центрів.

4.3.5 Узагальнений висновок

Аналіз конкурентних рішень показав, що на ринку відсутні універсальні програмні продукти, які поєднують онтологічне моделювання, логічний аналіз та практичну орієнтацію на оптимізацію технологічних процесів. Запропонований стартап-проект займає перспективну нішу та має суттєві конкурентні переваги за рахунок використання семантичних технологій, гнучкої архітектури та можливості масштабування.

4.4 Організаційна структура стартап-проекту

Загальні принципи побудови структури

Організаційна структура стартап-проекту формується з урахуванням його початкового масштабу, обмежених фінансових ресурсів та орієнтації на розробку інтелектуального програмного продукту. Основним принципом є **компактність команди**, поєднання кількох ролей однією особою на ранніх етапах та можливість подальшого масштабування структури без зміни базової логіки управління.

Запропонована структура є **проектно-орієнтованою**, що дозволяє ефективно керувати розробкою онтологічної платформи оптимізації технологічних процесів та її впровадженням у різні предметні області.

Керівний рівень

Керівник стартап-проекту (CEO / Project Manager)

Відповідає за:

- стратегічне планування розвитку продукту;
- координацію роботи команди;
- ухвалення ключових технічних та бізнес-рішень;
- комунікацію з потенційними замовниками та партнерами.

На початковому етапі дана роль поєднується з функціями **системного аналітика**, що дозволяє зменшити витрати та забезпечити цілісне бачення продукту.

Технічний рівень

Онтологічний інженер / Knowledge Engineer

Основна роль у стартапі, яка включає:

- розроблення онтологічної моделі технологічних процесів;
- формалізацію класів, властивостей та логічних правил;
- налаштування reasoning-механізмів;
- адаптацію моделі під різні технологічні сценарії.

Програмний розробник (Backend / Integration Engineer)

Відповідає за:

- інтеграцію онтологічної моделі з програмними сервісами;
- реалізацію API для доступу до онтології;

- розроблення модулів аналізу та візуалізації результатів;
- забезпечення масштабованості рішення.

На ранньому етапі проєкту ці функції можуть виконуватися однією особою.

Аналітично-прикладний рівень

Аналітик технологічних процесів

Функції:

- аналіз предметної області конкретного замовника;
- формування вимог до онтологічної моделі;
- інтерпретація результатів reasoning;
- оцінка ефективності процесів до та після впровадження моделі.

Ця роль є ключовою при переході від універсальної моделі до прикладних рішень.

Підтримка та розвиток

Менеджер з розвитку продукту / бізнес-аналітик

Відповідає за:

- аналіз ринку та конкурентних рішень;
- формування нових сценаріїв використання продукту;
- підготовку комерційних пропозицій;
- планування масштабування стартапу.

На стартовій стадії функції можуть виконуватися керівником проєкту.

Узагальнена структура (опис словами)

Організаційна структура стартап-проєкту включає:

- один керівний центр управління;
- технічне ядро розробки онтологічної платформи;
- аналітичний блок для адаптації рішення під конкретні процеси;
- бізнес-орієнтований напрям розвитку та масштабування.

Така структура забезпечує:

- мінімальні операційні витрати;
- високу гнучкість;
- можливість швидкого розширення команди без зміни архітектури управління.

Зв'язок з темою магістерської роботи

Запропонована організаційна структура безпосередньо підтримує реалізацію стартап-проєкту, заснованого на **онтологічній моделі оптимізації технологічних процесів**, оскільки дозволяє:

- ефективно розробляти та підтримувати онтології;
- виконувати аналіз та оптимізацію процесів;
- масштабувати рішення на різні галузі без структурних змін.

4.5. Фінансове обґрунтування стартап-проєкту

Фінансове обґрунтування стартап-проєкту спрямоване на оцінку доцільності впровадження програмного продукту, побудованого на основі онтологічної моделі оптимізації технологічних процесів. Оскільки запропоноване рішення має програмну природу та не потребує значних матеріальних ресурсів, основні витрати пов'язані з розробкою, підтримкою та впровадженням програмного забезпечення.

4.5.1. Структура початкових витрат

Початкові витрати стартапу включають одноразові та умовно-постійні витрати, необхідні для створення мінімально життєздатного продукту (MVP).

Основні статті витрат:

Таблиця 4.2 – Основні статті витрат стартап-проєкту

Стаття витрат	Опис	Орієнтовна сума, грн
Аналітика та проєктування	Аналіз вимог, моделювання онтології	35 000

Стаття витрат	Опис	Орієнтовна сума, грн
Розробка онтологічної моделі	Protégé, OWL, логічні зв'язки	45 000
Розробка програмного модуля	Python, reasoner, інтеграція	65 000
Тестування та валідація	Перевірка моделей, сценарії	25 000
Інфраструктура	Хостинг, сервери, домен	15 000
Разом		185 000 грн

Примітка: розрахунок виконано з урахуванням середньоринкової вартості розробки програмних продуктів малого масштабу в Україні.

4.5.2. Поточні витрати на підтримку

Після запуску стартапу виникають регулярні витрати, пов'язані з підтримкою та розвитком системи.

Таблиця 4.3 – Поточні витрати на підтримку стартап-проєкту

Стаття витрат	Сума/місяць, грн
Хостинг та серверні ресурси	4 000
Технічна підтримка	6 000
Оновлення та масштабування	5 000
Разом	15 000 грн/міс

4.5.3. Модель монетизації

Для стартап-проєкту обрано комбіновану модель монетизації:

- **SaaS-підписка** — доступ до онтологічної платформи;
- **Корпоративні ліцензії** — для виробничих підприємств;
- **Консультаційні послуги** — адаптація онтології під конкретний процес.

Орієнтовна вартість:

- базова підписка — **6 000 грн / місяць**;

- корпоративна ліцензія — від 60 000 грн / рік.

4.5.4. Прогноз доходів

Для обережної оцінки приймемо мінімальний сценарій:

- 5 корпоративних клієнтів у перший рік;
- середній дохід з одного клієнта — 60 000 грн/рік.

Річний дохід:

$$D = 5 \times 60\,000 = 300\,000 \text{ грн} \quad (2.1)$$

4.5.5. Оцінка терміну окупності

Сумарні початкові витрати:

$$C = 185\,000 \text{ грн} \quad (2.2)$$

Середній чистий місячний дохід (з урахуванням витрат):

$$P = \frac{300\,000}{12} - 15\,000 \approx 10\,000 \quad (2.3)$$

Термін окупності:

$$T = \frac{185\,000}{10\,000} \approx 18 - 19 \text{ місяців} \quad (2.4)$$

4.5.6. Висновок щодо фінансової доцільності

Проведене фінансове обґрунтування показує, що стартап-проект:

- не потребує значних капіталовкладень;
- має помірний термін окупності;
- є економічно доцільним у межах програмних рішень для оптимізації технологічних процесів;
- може масштабуватися без пропорційного зростання витрат.

Таким чином, реалізація стартап-проекту на основі онтологічної моделі є фінансово виправданою та має потенціал для подальшого розвитку.

4.6 Оцінка ризиків та перспектив розвитку стартап-проекту

Реалізація стартап-проекту, пов'язаного з розробленням та впровадженням універсальної онтологічної моделі оптимізації технологічних процесів, супроводжується низкою ризиків, які необхідно враховувати на етапі планування та подальшого розвитку. Водночас проект має значний потенціал масштабування та практичного застосування у різних галузях промисловості.

4.6.1 Аналіз основних ризиків стартап-проєкту

У межах дослідження було виділено кілька основних груп ризиків.

Технологічні ризики.

До цієї групи належать ризики, пов'язані з:

- складністю інтеграції онтологічної моделі з існуючими виробничими або інформаційними системами;
- необхідністю адаптації моделі під специфіку конкретних технологічних процесів;
- обмеженнями продуктивності reasoning-механізмів при зростанні обсягу даних.

Зменшення технологічних ризиків досягається за рахунок універсальної структури онтології, модульного підходу та використання стандартів OWL і Protégé.

Організаційні ризики.

Можливі труднощі:

- нестача фахівців з онтологічного інжинірингу;
- потреба у додатковому навчанні персоналу замовника;
- залежність від експертів предметної області.

Ці ризики мінімізуються завдяки документації, шаблонам моделей та поступовому впровадженню рішення.

Фінансові ризики.

Фінансові ризики пов'язані з:

- перевищенням запланованих витрат на розробку;
- повільнішим залученням клієнтів на початкових етапах;
- залежністю від індивідуальних проєктів.

Вони знижуються завдяки сервісній моделі монетизації та можливості поетапного масштабування.

Ринкові ризики.

Можливими є:

- низька обізнаність потенційних клієнтів щодо онтологічних технологій;
- консервативність виробничих підприємств;
- конкуренція з традиційними системами управління процесами.

Для зменшення цього ризику стартап орієнтується на демонстраційні приклади, пілотні впровадження та інтеграцію з існуючими ІТ-рішеннями.

4.6.2 Перспективи розвитку стартап-проєкту

Незважаючи на наявність ризиків, стартап-проєкт має значні перспективи розвитку.

По-перше, універсальний характер онтологічної моделі дозволяє застосовувати її не лише для процесів 3D-друку, а й для інших технологічних процесів: адитивного виробництва, механічної обробки, складання, контролю якості тощо.

По-друге, проєкт може бути масштабований у напрямках:

- розширення бібліотеки готових онтологічних шаблонів;
- інтеграції з IoT-платформами та системами моніторингу;
- використання аналітики та машинного навчання на основі семантичних даних.

По-третє, можливим є перехід від індивідуальних проєктів до SaaS-моделі, що підвищує стабільність доходів і привабливість для інвесторів.

4.6.3 Загальна оцінка життєздатності проєкту

Проведений аналіз показує, що стартап-проєкт є технологічно обґрунтованим та перспективним. Основні ризики є керованими та можуть бути зменшені завдяки модульній архітектурі, універсальності моделі та поетапному впровадженню.

У довгостроковій перспективі стартап має потенціал перерости з науково-дослідного проєкту у комерційне рішення для оптимізації технологічних процесів у цифровому виробництві.

ЗАГАЛЬНІ ВИСНОВКИ

У магістерській роботі вирішено актуальну науково-практичну задачу, що полягає у **розробленні універсальної онтологічної моделі оптимізації технологічних процесів** з можливістю її застосування для аналізу, контролю та підвищення ефективності сучасних виробничих систем.

У ході виконання роботи було досягнуто таких основних результатів:

1. **Проведено аналіз сучасних підходів до онтологічного інжинірингу та моделювання технологічних процесів**, що дозволило визначити ключові недоліки існуючих рішень, зокрема їх вузьку спеціалізацію та обмежену придатність до повторного використання в різних предметних областях.

2. **Сформовано вимоги до універсальної онтологічної моделі технологічного процесу**, яка забезпечує формалізований опис операцій, параметрів, ресурсів, правил та результатів виконання процесу, а також підтримує логічний аналіз і reasoning.

3. **Розроблено структурну онтологічну модель технологічного процесу**, що включає основні класи, підкласи та зв'язки між ними. Запропонована модель є гнучкою, масштабованою та придатною для адаптації до різних типів технологічних процесів без зміни її базової структури.

4. **Виконано формалізацію логічних зв'язків між параметрами процесу**, що дозволяє встановлювати причинно-наслідкові залежності між режимами виконання операцій та характеристиками кінцевого результату. Це створює основу для автоматизованого аналізу процесу та виявлення неефективних режимів.

5. **Розроблено модуль оцінки ефективності технологічного процесу**, який забезпечує класифікацію процесів за рівнем ефективності на основі заданих критеріїв. Модуль дозволяє виконувати логічну перевірку відповідності параметрів встановленим вимогам та формувати узагальнену оцінку результатів.

6. **Реалізовано онтологічну модель у середовищі Protégé,** створено класи, об'єктні та атрибутні властивості, екземпляри, а також задано логічні обмеження та аксіоми. Проведено перевірку коректності моделі за допомогою reasoner, що підтвердило її логічну узгодженість.

7. **Продемонстровано приклад практичного застосування моделі на технологічному процесі 3D-друку,** що підтвердило універсальність запропонованого підходу та можливість використання моделі для аналізу реальних виробничих сценаріїв.

8. **Розроблено стартап-проєкт,** заснований на результатах дослідження, у якому онтологічна модель використовується як ядро програмного продукту для оптимізації технологічних процесів. Обґрунтовано організаційну структуру, фінансову доцільність, конкурентні переваги та перспективи розвитку проєкту.

Таким чином, у роботі досягнуто поставлену мету та виконано всі визначені завдання. Запропонована універсальна онтологічна модель може бути використана як основа для подальших досліджень, розширення функціональних можливостей, інтеграції з інтелектуальними виробничими системами та практичного впровадження у різних галузях.

Практична цінність роботи полягає у можливості використання отриманих результатів для підтримки прийняття рішень, оптимізації параметрів технологічних процесів та підвищення ефективності виробництва.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гаврилова Т. А. Онтології в задачах управління знаннями // Штучний інтелект. – 2018. – № 3. – С. 4–15.
2. Грицюк Ю. І. Інженерія знань та онтологічне моделювання : навч. посіб. – Львів : Видавництво ЛНУ, 2019. – 214 с.
3. Кобилянський О. В. Інформаційні технології в управлінні виробничими процесами. – Київ : КНУБА, 2020. – 198 с.
4. Литвин В. В., Пелешишин А. М. Онтологічні моделі та їх застосування в інформаційних системах. – Львів : Видавництво НУЛП, 2017. – 256 с.
5. Пасічник В. В., Шаховська Н. Б. Інтелектуальні системи та технології. – Львів : Магнолія 2006, 2018. – 312 с.
6. Сааті Т. Л. Прийняття рішень. Метод аналізу ієрархій. – Київ : Радіо і зв'язок, 1993. – 316 с.
7. Субота О. М. Моделювання технологічних процесів у виробничих системах // Вісник НТУУ «КПІ». – 2021. – № 2. – С. 45–52.
8. Ткаченко В. А. Управління технологічними процесами. – Харків : ХНУРЕ, 2019. – 184 с.
9. Шаховська Н. Б. Семантичні технології в інформаційних системах. – Львів : НУЛП, 2020. – 228 с.
10. Шумейко О. В. Інформаційні моделі виробничих процесів. – Київ : НАУ, 2018. – 205 с.
11. Gruber T. R. A Translation Approach to Portable Ontology Specifications // Knowledge Acquisition. – 1993. – Vol. 5, No. 2. – P. 199–220.
12. Guarino N. Formal Ontology in Information Systems // Proceedings of FOIS'98. – Amsterdam : IOS Press, 1998. – P. 3–15.
13. Noy N. F., McGuinness D. L. Ontology Development 101: A Guide to Creating Your First Ontology. – Stanford University, 2001.
14. Horridge M. A Practical Guide To Building OWL Ontologies Using Protégé 5. – Stanford University, 2011.

15. Antoniou G., van Harmelen F. A Semantic Web Primer. – MIT Press, 2008. – 264 p.
16. Allemang D., Hendler J. Semantic Web for the Working Ontologist. – Morgan Kaufmann, 2011. – 384 p.
17. W3C. OWL 2 Web Ontology Language Document Overview. – 2012. – URL: <https://www.w3.org/TR/owl2-overview/>
18. W3C. RDF 1.1 Concepts and Abstract Syntax. – URL: <https://www.w3.org/TR/rdf11-concepts/>
19. Protégé Ontology Editor. – Stanford University. – URL: <https://protege.stanford.edu/>
20. Pellet Reasoner. – URL: <https://github.com/stardog-union/pellet>
21. HermiT OWL Reasoner. – URL: <http://www.hermit-reasoner.com/>
22. Groover M. P. Automation, Production Systems, and Computer-Integrated Manufacturing. – Pearson, 2015. – 816 p.
23. Kalpakjian S., Schmid S. Manufacturing Engineering and Technology. – Pearson, 2014. – 1200 p.
24. ISO 9001:2015 Quality management systems – Requirements.
25. ISO 10303 (STEP): Industrial automation systems and integration.
26. Monostori L. Cyber-physical production systems // CIRP Annals. – 2014. – Vol. 63. – P. 621–641.
27. Schuh G. et al. Industrie 4.0 Maturity Index. – Acatech Study, 2017.
28. Gibson I., Rosen D., Stucker B. Additive Manufacturing Technologies. – Springer, 2015. – 498 p.
29. ISO/ASTM 52900:2015 Additive manufacturing – General principles.
30. Ngo T. D. et al. Additive manufacturing (3D printing): A review of materials, methods // Composites Part B. – 2018. – Vol. 143. – P. 172–196.
31. Slicer Software Documentation (Ultimaker Cura). – URL: <https://ultimaker.com/software/ultimaker-cura>

32. Osterwalder A., Pigneur Y. Business Model Generation. – Wiley, 2010. – 288 p.
33. Ries E. The Lean Startup. – Crown Publishing, 2011. – 336 p.
34. Blank S. The Startup Owner's Manual. – K&S Ranch, 2012. – 608 p.
35. Kotler P., Keller K. Marketing Management. – Pearson, 2016. – 832 p.
36. Scarborough N. M. Effective Small Business Management. – Pearson, 2016.
37. European Commission. Startup Europe Initiative. – URL: <https://startup-europe.ec.europa.eu/>
38. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання.
39. ДСТУ 3008:2015. Звіти у сфері науки і техніки.
40. ДСТУ ISO/IEC 25010:2016. Системна та програмна інженерія.
41. Закон України «Про інноваційну діяльність».
42. Закон України «Про наукову і науково-технічну діяльність».
43. Stanford Center for Biomedical Informatics Research – Protégé Resources. – URL: <https://protegewiki.stanford.edu/>
44. Towards Data Science. Semantic Technologies & Ontologies. – URL: <https://towardsdatascience.com/>