

**БАКАЛАВРСЬКА РОБОТА**

**БР. ІІІ - 35.00.00.000 ІІЗ**

**Група ІІІ-21-2**

**Тиховецький Ярослав**

**2025**

**Івано-Франківський національний технічний університет нафти і газу**

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

**Тиховецький Ярослав Степанович**

(прізвище, ім'я, по батькові)

УДК 004  
(індекс)

## **БАКАЛАВРСЬКА РОБОТА**

**Реалізація DSS системи на основі концепції семантичної мережі**

(назва роботи)

**Інженерія програмного забезпечення**

(назва освітньої програми)

**121 - Інженерія програмного забезпечення**

(шифр і назва спеціальності)

**Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело**

Здобувач освітнього рівня Тиховецький Я.С.  
(підпис, ініціали та прізвище здобувача)

Науковий керівник Шекета Василь Іванович, д.т.н., професор  
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту  
Завідувач кафедри

доц. Бандура В.В.  
(посада) (підпис) (дата) (ініціали та прізвище)

**Івано-Франківськ – 2025**

**Івано-Франківський національний технічний університет нафти і газу**

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ:**

Зав. кафедрою ІІЗ

доц.

В.В. Бандура

“     ”     2025 р.

## **ЗАВДАННЯ**

### **НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ**

**Тиховецькому Ярославу Степановичу**

(прізвище, ім'я, по-батькові)

**1. Тема проекту (роботи) “ Реалізація DSS системи на основі концепції семантичної мережі ”**

керівник проекту (роботи) Шекета Василь Іванович, професор

затвержені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

**2. Строк подання студентом проекту (роботи) 10 червня 2025 р.**

**3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики**

**4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)**

1. Аналіз предметної області використання семантичних мереж для DSS систем

2. Концептуальний контекст аналізу рішень, семантики, онтологій та мови веб-онтологій

3. Архітектура інтероперабельної DSS системи на базі семантичних веб-технологій

4. Побудова онтологій DSS системи на основі концепції семантичної мережі

5. Концептуальна модель середовища DSS з використанням технологій семантичного вебу

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

1. Схематичний вигляд DSS системи (рис. 1.1)

2. Архітектура DSS на основі багатоагентних систем (рис. 1.2)

3. Багатошарова модель семантичного вебу (рис. 1.3)

4. Приклад діаграми впливу, яка моделює процес прийняття рішення щодо продукту (рис. 2.1)

5. Архітектура інтероперабельної DSS системи (рис. 2.2)

## 6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 28 квітня 2025 р.

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз предметної області використання семантичних мереж для DSS систем	03.05.2025	виконано
2	Концептуальний контекст аналізу рішень, семантики, онтологій та мови веб-онтологій	15.05.2025	виконано
3	Архітектура інтегрованої DSS системи на базі семантичних веб-технологій	25.05.2025	виконано
4	Побудова онтологій DSS системи на основі концепції семантичної мережі	01.06.2025	виконано
5	Концептуальна модель середовища DSS з використанням технологій семантичного вебу	07.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник \_\_\_\_\_

(підпис)

Керівник роботи \_\_\_\_\_

(підпис)

## АНОТАЦІЯ

Бакалаврська робота містить 76 сторінок, 26 рисунків, список використаних джерел із 36 найменуваннями, 1 додаток.

**Мета роботи:** розробка та реалізація DSS-системи на основі концепції семантичної мережі з використанням сучасних інструментів розробки онтологій, логічного виведення та візуалізації знань

**Об'єкт дослідження:** системи підтримки прийняття рішень у контексті управління знаннями

**Предмет дослідження:** семантичні мережі, онтології та методи логічного виведення в DSS-системах

**Результати дослідження:** отримані результати демонструють потенціал семантичного підходу для створення інтелектуальних інформаційних систем, що забезпечують підтримку прийняття рішень.

**В першому розділі** визначено основні характеристики архітектури DSS-систем та обґрунтовано необхідність застосування семантичних технологій для розширення функціональності

**В другому розділі** проаналізовано концепти семантики, онтологій і мови OWL, розглянуто архітектуру семантично орієнтованої DSS-системи та описано послідовність її реалізації.

**В третьому розділі** здійснено побудову онтологій, реалізацію логічного виведення та створено концептуальну модель DSS-середовища на основі семантичного вебу

**Висновок:** реалізовано програмний прототип DSS-системи з логічним reasoner-компонентом, що дозволяє робити висновки на основі формалізованих знань

**КЛЮЧОВІ СЛОВА:** СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, СЕМАНТИЧНА МЕРЕЖА, ОНТОЛОГІЯ, ЛОГІЧНЕ ВИВЕДЕННЯ, PROTÉGÉ, СЕМАНТИЧНИЙ ВЕБ

## ANNOTATION

The bachelor's thesis comprises 76 pages, 26 figures, a list of 36 references, and 1 appendix.

**Objective of the study:** Development and implementation of a DSS system based on the concept of a semantic network using modern tools for ontology development, logical inference, and knowledge visualization.

**Object of the study:** Decision Support Systems in the context of knowledge management.

**Subject of the study:** Semantic networks, ontologies, and methods of logical inference in DSS systems.

**Research results:** The obtained results demonstrate the potential of the semantic approach for building intelligent information systems that support decision-making.

**In the first chapter,** the main characteristics of DSS system architecture are defined, and the necessity of applying semantic technologies to expand functionality is substantiated.

**In the second chapter,** concepts of semantics, ontologies, and the OWL language are analyzed; the architecture of a semantically-oriented DSS system is examined, and the implementation sequence is described.

**In the third chapter,** the construction of ontologies, the implementation of logical inference, and the creation of a conceptual model of a DSS environment based on the semantic web are carried out.

**Conclusion:** A software prototype of a DSS system with a logical reasoner component was implemented, enabling inference based on formalized knowledge.

**KEYWORDS:** DECISION SUPPORT SYSTEM, SEMANTIC NETWORK, ONTOLOGY, LOGICAL INFERENCE, PROTÉGÉ, SEMANTIC WEB,

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП .....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ СЕМАНТИЧНИХ МЕРЕЖ ДЛЯ DSS СИСТЕМ .....	12
1.1. Особливості архітектури DSS систем.....	12
1.2. Опис пропонованої розробки дипломної роботи .....	17
1.2.1. Обґрунтування розробки .....	17
1.2.2. Вплив семантичної технології .....	18
1.2.3 Розробка онтологій та анотування .....	19
1.3. Семантичний підхід до управління знаннями при розробці ПЗ .....	19
1.4. Випадки використання семантичного підходу до управління рішеннями.....	21
1.4.1. Процес розробки.....	21
1.4.2. Бізнес-процес .....	22
1.4.3. Документація .....	23
1.4.4. Сумісність та повторне використання .....	24
1.5. Роль онтологій та метаданих у підтримці рішень на основі семантичного вебу .....	24
1.5.1. Онтологія та метадані .....	26
1.5.2. Багатошарова модель семантичного вебу ("Layer Cake") .....	27
РОЗДІЛ 2. КОНЦЕПТУАЛЬНИЙ КОНТЕКСТ АНАЛІЗУ РІШЕНЬ, СЕМАНТИКИ, ОНТОЛОГІЙ ТА МОВИ ВЕБ-ОНТОЛОГІЙ.....	31
2.1. Семантика .....	31

					БР.ІІ – 35.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Реалізація DSS системи на основі концепції семантичної мережі  <b>Пояснювальна записка</b>	Літ.	Арк.	Акрушіє
Розроб.		Тиховецький Я						
Перевір.		Шекета В.І.					6	
Реценз.						ІФНТУНГ ІІ-21-2		
Н. Контр.		Піх М.М.						
Затверд.		Бандура В.В.						

2.2. Онтологія .....	31
2.3. Мова веб-онтологій OWL .....	32
2.4. Eclipse Process Framework .....	33
2.5. Представлення діаграми впливу .....	35
2.6. Архітектура інтероперабельної DSS системи на базі семантичних веб-технологій .....	39
2.7. Композиція веб-сервісів DSS системи .....	42
2.8. Аналіз рішень .....	44
2.9. Порядок виконання проекту .....	45

### РОЗДІЛ 3. ПОБУДОВА ОНТОЛОГІЙ DSS СИСТЕМИ НА ОСНОВІ

КОНЦЕПЦІЇ СЕМАНТИЧНОЇ МЕРЕЖІ .....	47
3.1. Використання онтологічного редактора Protege .....	47
3.2. Використання Стар для побудови діаграм .....	48
3.3. Методологія для формалізації та інтеграції аналізу рішень .....	51
3.4. Алгоритмічна реалізація логічного виведення (реазонатора) в DSS системі.....	55
3.4.1. Практичний сценарій .....	56
3.5. Концептуальна модель середовища DSS з використанням технологій семантичного вебу .....	64

ВИСНОВКИ.....	70
---------------	----

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	73
---------------------------------------	----

ДОДАТКИ

БІБЛІОГРАФІЧНА ДОВІДКА

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

W3C - World Wide Web Consortium (Консорціум Всесвітньої Павутини)

RDF - Resource Description Framework (Фреймворк опису ресурсів)

DAML+OIL - DARPA Agent Markup Language combined with the Ontology Interchange Layer

DSS - Decision Support System (Система підтримки рішень)

IDSS - Intelligent Decision Support System (Інтелектуальна система підтримки рішень)

SIDSS - Semantic Web technologies based IDSS (Інтелектуальна система підтримки рішень на основі технологій Семантичного Вебу)

SWSM - Semantic Web Service Manager (Менеджер Веб-сервісів Семантичного Вебу)

AI - Artificial Intelligence (Штучний інтелект)

IAs - Intelligent Agents (Інтелектуальні Агенти)

ANNs - Artificial Neural Networks (Штучні Нейронні Мережі)

ESs - Expert Systems (Експертні Системи)

SOA - Service-Oriented Architecture (Сервіс-орієнтована архітектура)

HTML - HyperText Markup Language (Мова гіпертекстової розмітки)

GUI - Graphical User Interface (Графічний Користувацький Інтерфейс)

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

У сучасному інформаційному суспільстві постійно зростає обсяг даних, що потребують обробки, аналізу та трансформації у знання для ефективного прийняття рішень. Традиційні системи підтримки прийняття рішень (DSS) не завжди забезпечують достатній рівень адаптивності, масштабованості та інтерпретованості знань, особливо в умовах складних і динамічних предметних областей. Семантичні технології, зокрема онтології та семантичні мережі, відкривають нові можливості для створення інтелектуальних DSS-систем, що здатні здійснювати логічне виведення та працювати з формалізованими знаннями на концептуальному рівні.

Таким чином, реалізація DSS-системи з використанням семантичної мережі є актуальним напрямом, який поєднує підходи штучного інтелекту, інженерії знань та інформаційних технологій. Такий підхід дозволяє підвищити якість рішень, що приймаються, забезпечити їх обґрунтованість, прозорість і повторне використання знань у різних контекстах.

### **Актуальність роботи**

Зростання складності управлінських процесів вимагає впровадження інтелектуальних систем підтримки рішень, які здатні інтерпретувати та логічно пов'язувати знання — саме це забезпечують семантичні мережі.

Системи підтримки прийняття рішень (DSS) є важливим інструментом сучасних інформаційних технологій, що спрямовані на підвищення ефективності управлінських, виробничих та інженерних процесів. Проте традиційні підходи до реалізації таких систем часто стикаються з труднощами, пов'язаними з неоднорідністю даних, недостатнім рівнем формалізації знань, обмеженою здатністю до масштабування та слабкою адаптацією до зміни контексту.

В останні роки стрімко розвивається підхід, який базується на використанні семантичних мереж і онтологій як засобів представлення знань.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці технології забезпечують глибший рівень семантичного розуміння інформації, підвищують точність та обґрунтованість висновків, що робляться DSS-системами. Семантичні веб-технології дозволяють інтегрувати розподілені джерела знань, забезпечити інтероперабельність між компонентами системи та реалізувати логічне виведення на основі заданих правил.

**Метою даної дипломної роботи** є розробка та реалізація DSS-системи на основі концепції семантичної мережі з використанням сучасних інструментів розробки онтологій, логічного виведення та візуалізації знань.

У межах роботи досліджено теоретичні засади функціонування семантичних DSS, здійснено побудову онтологічної моделі предметної області, реалізовано програмну архітектуру DSS-системи з можливістю логічного виведення, а також протестовано практичний сценарій її застосування.

#### **Завдання дослідження**

1. Проаналізувати архітектурні особливості DSS-систем.
2. Вивчити роль онтологій, метаданих і семантичних технологій у підтримці рішень.
3. Розробити концептуальну модель DSS-системи на основі OWL-онтологій.
4. Реалізувати логіку логічного виведення за допомогою реанонатора.
5. Створити практичний приклад реалізації системи з використанням редакторів Protege та Смар.
6. Провести аналіз ефективності побудованої системи у прикладному сценарії.

**Об'єкт дослідження** - системи підтримки прийняття рішень у контексті управління знаннями.

**Предмет дослідження** - семантичні мережі, онтології та методи логічного виведення в DSS-системах.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

### **Методи дослідження**

- Аналіз архітектурних підходів DSS-систем
- Онтологічне моделювання (Protege, OWL)
- Методи семантичного вебу та логічного виведення
- Формалізація знань та моделювання семантичних структур (Сmap)
- Моделювання та побудова інтероперабельної архітектури систем
- Сценарний підхід для перевірки ефективності

### **Наукова новизна**

Запропоновано багаторівневу архітектуру DSS-системи на основі семантичного вебу, що поєднує логічне виведення, онтологічне моделювання та композицію веб-сервісів для забезпечення адаптивної та інтероперабельної підтримки прийняття рішень.

### **Практичне застосування**

Результати дослідження можуть бути використані при створенні DSS-систем у сферах бізнес-аналітики, управління проєктами, документообігу та в освітніх або медичних інформаційних системах, що потребують гнучкого управління знаннями.

Бакалаврська робота містить 76 сторінок, 26 рисунків, 3 розділи список використаних джерел із 36 найменуваннями, 1 додаток.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИКОРИСТАННЯ СЕМАНТИЧНИХ МЕРЕЖ ДЛЯ DSS СИСТЕМ

## 1.1. Особливості архітектури DSS систем

DSS система (Decision Support System, тобто Система підтримки прийняття рішень) — це комп'ютеризована інформаційна система, яка допомагає людям у прийнятті рішень в організації. Вона не приймає рішення самостійно, а надає інструменти та інформацію, щоб допомогти користувачам аналізувати дані, моделювати різні сценарії та робити обґрунтований вибір.

Основні характеристики DSS:

### 1. Підтримка, а не заміна рішень

DSS системи доповнюють людські здібності, а не замінюють їх. Вони допомагають менеджерам та іншим користувачам краще зрозуміти складні проблеми.

### 2. Гнучкість та інтерактивність.

Користувачі можуть легко взаємодіяти з системою, змінювати параметри, запускати різні моделі та отримувати нові результати.

### 3. Орієнтація на неструктуровані та напівструктуровані проблеми.

DSS особливо корисні для вирішення проблем, які не мають чіткого алгоритму розв'язання і потребують творчого підходу та аналізу великого обсягу даних.

### 4. Використання даних та моделей.

Системи використовують як історичні дані, так і аналітичні моделі (статистичні, імітаційні) для підтримки процесу прийняття рішень.

### 5. Підтримка різних етапів прийняття рішень.

Від ідентифікації проблеми та збору інформації до оцінки альтернатив та вибору найкращого рішення.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Зазвичай DSS складається з трьох основних компонентів:

- Підсистема управління даними (Data Management Subsystem).

Включає в себе базу даних (як внутрішніх, так і зовнішніх джерел), систему управління базами даних (СУБД) та механізми для вилучення, оновлення та маніпулювання даними.

- Підсистема управління моделями (Model Management Subsystem).

Містить різні аналітичні моделі (фінансові, статистичні, операційні, оптимізаційні), які дозволяють аналізувати дані, прогнозувати результати та моделювати різні сценарії.

- Підсистема інтерфейсу користувача (User Interface Subsystem).

Забезпечує взаємодію між користувачем і системою, дозволяючи легко вводити запити, отримувати звіти та візуалізувати дані.

Переваги використання DSS полягають в тому, що вони допомагають приймати більш обґрунтовані та ефективні рішення за рахунок доступу до релевантної інформації та аналітичних інструментів, відбувається підвищення операційної ефективності, оптимізація бізнес-процесів та ресурсів, збільшення конкурентоспроможності, швидше реагування на зміни ринку та прийняття стратегічних рішень.

DSS системи є потужним інструментом, що допомагає організаціям приймати кращі рішення в умовах зростаючої складності та обсягу даних.

З точки зору складу, DSS системи зазвичай поділяються на 4 основні типи компонентів: підсистему користувацького інтерфейсу, підсистему управління моделями, підсистему управління даними та підсистему управління знаннями (або базовану на знаннях підсистему управління). DSS з четвертим компонентом, тобто базованими на знаннях підсистемами, забезпечують інтелектуальну підтримку для інших компонентів, і, як визначено раніше, такі DSS називаються IDSS/ Взаємозв'язок чотирьох основних компонентів між собою та з особою, що приймає рішення, можна побачити на рисунку 1.1.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

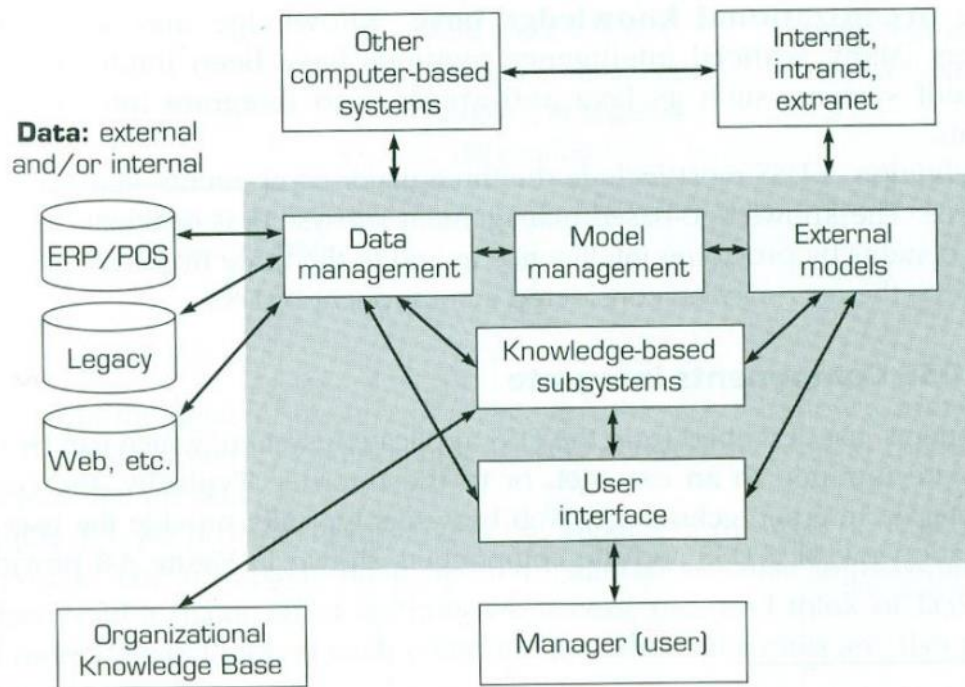


Рисунок 1.1 - Схематичний вигляд DSS системи

На Рисунку 1.1 менеджер взаємодіє лише з користувацьким інтерфейсом, який обробляє функції введення/виведення від трьох інших компонентів. Сіра рамка на рисунку 1.1 представляє всю систему, тоді як компоненти поза рамкою є зовнішніми джерелами, з яких DSS отримують інформацію. Компоненти з'єднуються між собою за допомогою інтернет-технологій. Користувацький інтерфейс забезпечує послідовний графічний користувацький інтерфейс (ГКІ) для взаємодії особи, що приймає рішення, зі DSS та їхнього контролю. Користувач також вважається частиною системи, оскільки відомо, що ефективність DSS походить від інтенсивної взаємодії між користувачем і комп'ютером.

Підсистема управління даними включає програмне забезпечення системи управління базами даних для обробки баз даних, що містять відповідні дані для DSS. Бази даних зазвичай є веб-серверами і можуть бути внутрішніми або зовнішніми для DSS. Підсистема бази моделей має подібне програмне забезпечення, яке називається системою управління базами

моделей, що включає фінансові, статистичні або інші кількісні моделі, і навіть мови моделювання, щоб надавати DSS аналітичні можливості та відповідне управління програмним забезпеченням. Моделі працюють на серверах застосунків і можуть бути або локально збережені, або отримані з інтернет-джерел.

Останнім компонентом є базована на знаннях підсистема, яка є унікальною для DSS, може надавати інтелектуальну підтримку іншим компонентам або діяти як окремий компонент. Вона може бути підключена до локального сховища знань або до зовнішніх веб-серверів. Вона також може складатися з методів та інструментів штучного інтелекту, розроблених у системах веб-розробки, таких як Java, для допомоги іншим компонентам DSS.

DSS система може проводити глибокий аналіз даних який допомагає виявити приховані тенденції та можливості для кращого розуміння бізнесу. Автоматизація збору та аналізу даних дозволяє менеджерам зосередитися на вищих рівнях прийняття рішень.

Інтелектуальні Агенти (IA) розглядаються як ключовий фактор, що забезпечує функціонування інтелектуальних DSS систем. IA являють собою програмні сутності, що демонструють інтелектуальну поведінку через інтеграцію властивостей програмних агентів та інтелектуальних систем. У широкому сенсі, агент може бути визначений як будь-яка сутність, що здатна автономно взаємодіяти зі своїм середовищем, при цьому інтелектуальний агент сприймає своє оточення, приймає обґрунтовані рішення на основі своїх сприйнять та діє відповідно. Агент характеризується наступними ознаками:

- а. Автономність: Агенти функціонують без прямого втручання людини.
- б. Кооперативність: Агенти співпрацюють з іншими агентами для досягнення спільних цілей.
- в. Реактивність: Агенти не лише реагують на своє середовище, але й здатні проявляти ініціативу та демонструвати цілеспрямовану поведінку.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

г. Мобільність: Агенти здатні переміщатися комп'ютерними мережами. Один агент може ініціювати створення іншого агента на віддаленому комп'ютері для виконання завдань. Крім того, агенти можуть мігрувати між комп'ютерами під час виконання, переносячи з собою накопичені знання та дані.

З огляду на зазначені характеристики агентів, зокрема інтелектуальних агентів, інтеграція агентних технологій в IDSS може потенційно призвести до скорочення втручання людини та часу, необхідного для прийняття рішень, за умови керування системою через мережі агентів.

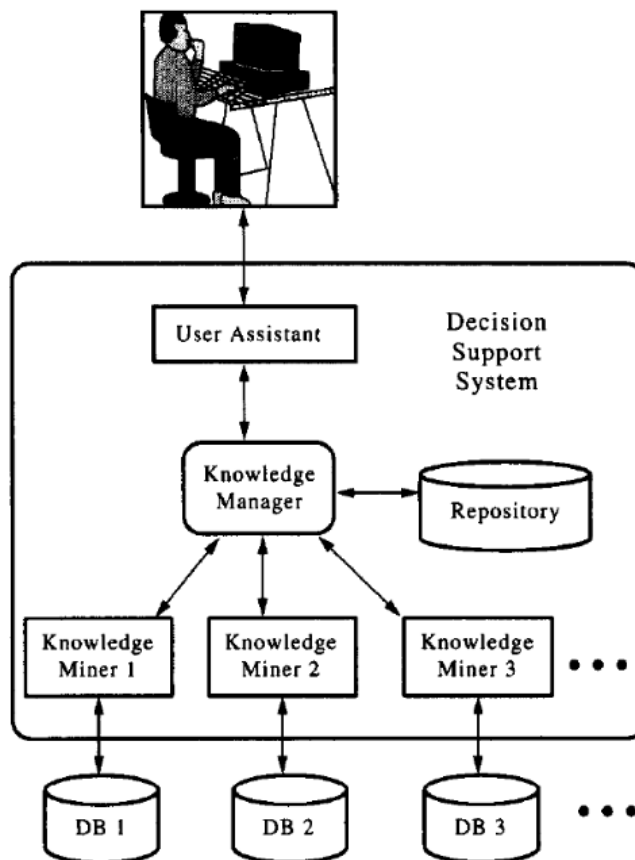


Рисунок 1.2 - Архітектура DSS на основі багатоагентних систем

Рисунок 1.2 ілюструє архітектуру DSS на основі багатоагентних систем де представлено три типи агентів підтримки рішень: агент видобування знань, який виявляє приховані зв'язки даних у джерелах

інформації; агенти-асистенти користувача, що виступають інтелектуальним інтерфейсом між особою, що приймає рішення, та DSS, що підтримуються ІА; і менеджер знань з підтримкою сховища, який забезпечує координацію системи та сприяє комунікації знань.

## 1.2. Опис пропонованої розробки дипломної роботи

У даній дипломній роботі досліджується застосування методів семантичного вебу для формального представлення аналізу рішень у контексті розробки програмного забезпечення. Метою дослідження є інтеграція процесу прийняття рішень у життєвий цикл розробки програмного забезпечення (ЖЦРПЗ) за допомогою фреймворку Eclipse Process Framework (EPF). Такий підхід дозволяє створювати формалізовані артефакти рішень, які не тільки чітко документують сутність, обґрунтування та причини прийняття конкретних рішень, а й можуть бути безпосередньо пов'язані з численними іншими артефактами процесу розробки.

### 1.2.1. Обґрунтування розробки

Традиційні підходи до документування рішень часто страждають від неоднозначності, відсутності структури та складності у відстеженні зв'язків, що перешкоджає ефективному повторному використанню знань та трасуванню рішень. Використання семантичного вебу, зокрема таких технологій як онтології Web Ontology Language (OWL), дозволяє подолати ці обмеження шляхом надання машинозчитуваного та структурованого представлення інформації про рішення. Це забезпечує точне визначення сутностей, властивостей та взаємозв'язків, що лежать в основі аналізу рішень.

Eclipse Process Framework (EPF), як гнучкий та розширюваний фреймворк для визначення, управління та розгортання процесів розробки програмного забезпечення, є ідеальною платформою для інтеграції процесу

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

прийняття рішень. EPF дозволяє моделювати діяльність, ролі, результати та інструменти, пов'язані з прийняттям рішень, тим самим вбудовуючи їх безпосередньо у визначені фази та ітерації ЖЦРПЗ.

Інтеграція цих двох парадигм дозволяє:

- Створити формальний артефакт. Рішення, їх обґрунтування, альтернативи, зацікавлені сторони та вплив можуть бути представлені як семантичні трійки, що робить їх доступними для автоматизованої обробки та запитів.

- Забезпечити трасування. Формальні артефакти рішень можуть бути пов'язані з вимогами, архітектурними елементами, дизайн-документами, тестовими сценаріями та іншими артефактами розробки, забезпечуючи повну трасувальність від кінцевого продукту до вихідного рішення.

### *1.2.2. Вплив семантичної технології*

Застосування семантичної технології у сфері аналізу рішень має значний потенціал для покращення:

- Інструменти пошуку Семантичного Вебу дозволяють виконувати складні запити на основі концептуальної подібності та контексту, що значно перевершує можливості ключового пошуку. Це полегшує знаходження подібних рішень, прийнятих у минулому, з метою їх повторного використання.

- Можливість ідентифікувати та повторно використовувати обґрунтовані рішення, а не переробляти їх, призводить до економії часу, ресурсів та підвищення якості прийнятих рішень.

- Стандарти семантичного вебу (RDF, OWL) сприяють сумісності даних про рішення між різними інструментами та платформами.

- Формальне представлення дозволяє здійснювати автоматизовану перевірку дотримання встановлених організаційних політик та стандартів у процесі прийняття рішень.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.2.3 Розробка онтологій та анотування

У межах даної роботи була розроблена онтологія OWL для моделювання базового процесу прийняття рішень. Ця онтологія формалізує ключові концепції, такі як "Рішення", "Альтернатива", "Критерій", "Обґрунтування", "Зацікавлена Сторона", "Етап Процесу Прийняття Рішень" тощо, відображаючи їх взаємозв'язки. Архітектура онтології була розроблена з урахуванням можливості представлення процесу за допомогою ERF, забезпечуючи безшовну інтеграцію між концептуальною моделлю рішення та його процесуальним контекстом.

Розроблена онтологія процесу була інтегрована з існуючою онтологією анотації аналізу рішень OWL, яка фокусується на змістовному аспекті самого рішення. Ця інтеграція дозволила створити єдину, всеосяжну онтологічну модель, яка охоплює як "що" (зміст рішення), так і "як" (процес прийняття рішення).

На заключному етапі, отримана інтегрована онтологія була успішно використана для анотування зразків рішень та відповідних процесів, за допомогою яких ці рішення були прийняті. Це практичне застосування продемонструвало ефективність запропонованого підходу у формалізації та управлінні знаннями, пов'язаними з прийняттям рішень у розробці програмного забезпечення.

### 1.3. Семантичний підхід до управління знаннями при розробці ПЗ

У цьому розділі дипломної роботи розглядається критична роль прийняття рішень у процесах розробки програмного забезпечення, а також виклики, пов'язані з їх формалізацією та управлінням.

Прийняття рішень є невід'ємною частиною життєвого циклу розробки програмного забезпечення, часто стосуючись питань високого рівня ризику та значного впливу на проект. Прикладом такого фундаментального рішення

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

може бути вибір конкретної моделі ЖЦРПЗ для проекту. Процес прийняття подібних рішень є системним і багатофакторним, вимагаючи ретельного аналізу проблеми, інтенсивних обговорень альтернативних варіантів, а також врахування множинних критеріїв і зацікавлених сторін. Результатом цього процесу є формування висновків, які лягають в основу остаточного рішення.

Документація таких рішень зазвичай включає інформацію про метод, час, причини їх прийняття, а також розглянуті альтернативи, висунуті аргументи та використані критерії. Однак, незважаючи на значний обсяг аналітичної роботи, що передує прийняттю рішень, зв'язок між самим рішенням та контекстом його розробки в процесі ЖЦРПЗ часто залишається неформальним або повністю відсутнім у сучасних системах підтримки.

Ця відсутність формального зв'язку створює низку істотних проблем:

- Складність перегляду та повторного використання.

Аналіз, що привів до рішення, нерідко відкидається після його прийняття. Навіть якщо він зберігається, недостатня документація контексту, в якому проводився аналіз, ускладнює його ефективний перегляд у разі зміни обставин або повторне використання для аналогічних рішень.

- Обмежена масштабованість та адаптивність.

Неформальний характер зв'язку з процесом обмежує масштабованість, адаптованість та гнучкість систем управління рішеннями. Це призводить до необхідності розробки кожного нового рішення "з нуля", навіть якщо воно є лише незначною варіацією попереднього, що суттєво знижує ефективність та продуктивність.

Семантичні технології мають потенціал значно підвищити ефективність пошуку, повторного використання та верифікації аналізу рішень. У рамках даної роботи була розроблена онтологія OWL для формального представлення базового процесу прийняття рішень, яка була інтегрована з моделями процесу, визначеними за допомогою Eclipse Process Framework (EPF). Крім того, онтологія процесу була суміщена з існуючою

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

онтологією анотації аналізу рішень OWL. Отримана інтегрована онтологія була застосована для анотування вибірки рішень та пов'язаних з ними процесів, що забезпечило формалізоване представлення їх контексту та змісту.

Основне завдання полягає у визначенні ступеня, в якому семантичні технології додають цінність для пошуку, повторного використання та верифікації аналізу рішень на розробленому зразку.

#### **1.4. Випадки використання семантичного підходу до управління рішеннями**

У цьому підрозділі представлено типові сценарії використання, які демонструють значущість формалізації та управління знаннями про рішення, особливо в контексті розробки програмного забезпечення та ширших бізнес-процесів.

##### *1.4.1. Процес розробки*

У будь-якому процесі розробки виникає необхідність прийняття численних рішень, кожне з яких має під собою певне обґрунтування. У випадку залучення великих команд, процес прийняття рішень може ускладнюватися через велику кількість запропонованих варіантів. Хоча це сприяє всебічному розгляду проблеми, відсутність формального запису про те, як і чому команда або окрема особа дійшла до певного рішення, значно ускладнює подальше відтворення цих причин або перегляд відхилених альтернатив. Таким чином, документування обґрунтування рішення суттєво сприяє ефективності процесу розробки, усуваючи необхідність повторного аналізу вже прийнятих рішень.

Приклад.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Під час фази проєктування апаратного продукту було прийнято рішення щодо довжини циліндричного валу (123 см), оскільки менша довжина могла призвести до пошкодження обладнання та створити загрозу безпеці. Пізніше, на завершальному етапі проєктування, член команди запропонував зменшити довжину до 110 см для зниження вартості на 10%. Без документування початкового обґрунтування для довжини 123 см, команда була б змушена повторно досліджувати всі аспекти рішення або ризикувати прийняттям некоректного рішення, наслідки якого могли б виявитися лише після виготовлення продукту.

#### *1.4.2. Бізнес-процес*

Аналогічно процесам розробки, кожен бізнес-процес структурується на основі множинних факторів, таких як вартість, прибутковість, якість, вимоги клієнтів, динаміка ринку тощо. У межах бізнес-процесів часто виникає потреба у прийнятті критичних рішень. Відсутність належної документації цих рішень або їхніх обґрунтувань створює значне навантаження при необхідності їх перегляду.

У будь-якому процесі, особливо в бізнесі, існує безліч альтернатив, і якщо обрана альтернатива не забезпечує очікуваних результатів, рішення вимагає перегляду. За наявності документованих рішень та розглянутих альтернатив проблема може бути виправлена або усунена за значно менший час порівняно з ситуацією, коли доводиться повторно аналізувати ситуацію.

Приклад.

Чотири місяці тому постачальник X був обраний серед дев'яти інших через найнижчу вартість, незважаючи на те, що постачальник Y пропонував безкоштовну доставку. Сьогодні постачальник X збільшив свої ціни, але вони все ще залишаються нижчими за ціни інших постачальників. Однак, з урахуванням безкоштовної доставки від постачальника Y, загальні витрати на закупівлю у нього можуть виявитися нижчими. Наявність

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

формалізованого запису про рішення, включаючи розглянуті альтернативи та їх обґрунтування, дозволила б оперативно переглянути вибір постачальника без необхідності повторного аналізу.

Крім того, критично важливим аспектом бізнес-процесів є дотримання політик, включаючи ті, що регулюються законодавством. Формалізація рішень та їхніх обґрунтувань є значно надійнішим та економічно ефективнішим способом забезпечення дотримання політик порівняно з неформальною експертизою або аудитом.

#### *1.4.3. Документація*

Усі процеси проектування та розробки в інженерній та академічній сферах передбачають створення документації майже на кожному етапі. Зокрема, розробка програмного забезпечення значною мірою залежить від детальної документації. Документування всіх рішень, прийнятих протягом процесу, спрощує можливість вибору інших альтернатив, якщо початковий вибір виявиться неоптимальним.

Також це забезпечує прозорість для керівництва або команди, які бажають зрозуміти обґрунтування певного рішення. Систематичне документування кожного рішення як обґрунтування суттєво спрощує ці процеси. Економія часу в таких ситуаціях є критичним фактором для успіху проекту.

#### *Приклад.*

У певних сценаріях, залежно від прийнятого рішення, команда або керівництво можуть обрати одну з кількох можливих гілок розвитку проекту. Якщо згодом обрана гілка виявиться непродуктивною або навіть такою, що ставить під загрозу весь проект, виникає потреба повернутися до точки розгалуження. За наявності документованого обґрунтування початкового рішення, цей процес буде значно спрощеним і швидшим.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 1.4.4. Сумісність та повторне використання

У більшості процесів повторне використання частини або всього робочого продукту призводить до значної економії часу та зусиль. Час, що найбільш інтенсивно витрачається, особливо при залученні групи людей, це прийняття рішень. Формалізована колекція всіх рішень, прийнятих у минулому, може бути ефективно повторно використана. Обґрунтування можуть бути застосовані в аналогічному майбутньому контексті або адаптовані для інших контекстів шляхом вилучення відповідних елементів.

Приклад.

Юридичні фірми регулярно використовують прецедентне право, повторно використовуючи рішення, прийняті раніше різними або тими ж особами, адаптуючи їх до поточних потреб. Навіть в існуючих системах, наявність формалізованого обґрунтування, чому була цитована певна справа або приклад, та які інші варіанти розглядалися, надало б цінні дані для майбутнього використання фірмою.

Здатність формалізувати рішення за допомогою семантичних технологій також значно покращує сумісність даних про рішення, дозволяючи різним системам та інструментам обмінюватися та інтерпретувати цю інформацію, що є фундаментальним для розширеного повторного використання.

### 1.5. Роль онтологій та метаданих у підтримці рішень на основі семантичного вебу

Семантичний Веб — це міграція від сучасного Вебу, що читається людиною та базується на документах, до Вебу майбутнього, що читається машиною та базується на даних. Веб виник перш за все як засіб для публікації документів людьми. Потужні пошукові системи, такі як Google ([www.google.com](http://www.google.com)), постійно індексують веб-документи та надають швидкі,

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

точні результати на текстові запити. Інформацію можна знайти в Інтернеті практично на будь-яку тему, маючи лише кілька ключових слів. Але як ми шукаємо в Інтернеті дані або сервіси? Веб у його нинішньому вигляді має дуже мало метаданих і не має засобів для кодування семантики. Суть Семантичного Вебу полягає в здатності виражати значення та встановлювати взаємозв'язки між ресурсами (документами, даними та сервісами).

Здатність обробляти запити та пошуки описаного вище типу вимагатиме методологій, запозичених зі штучного інтелекту, представлення знань та багатьох інших предметно-орієнтованих спільнот. Безумовно, Семантичний Веб не з'явиться за одну ніч, і його початкові реалізації не працюватимуть ідеально. Окреслюють чотири основні компоненти, необхідні для побудови Семантичного Вебу:

- Машинозчитуваний. Першим компонентом буде машинозчитувана розмітка для веб-контенту. XML (eXensible Markup Language) вже використовується для створення "документів, що самоописуються". Для кодування семантики веб-контенту будуть потрібні нові мови розмітки.

- Семантичне Індексування. Другим компонентом будуть інструменти, які можуть читати та індексувати семантичну розмітку. Як і у випадку з сучасним Вебом, Семантичний Веб покладатиметься на розподілені ресурси, індексовані централізованою базою знань. Агенти "скануватимуть" Семантичний Веб у пошуках метаданих – розмітки, яка описуватиме взаємозв'язки між ресурсами.

- Семантичні Сервіси. Третім компонентом буде розробка інструментів та сервісів, які обробляють семантичну інформацію та виводять нові факти. Цей компонент можна розглядати як наступне покоління сайтів для подорожей, новин, брокерських та інших сервісів – готових спілкуватися з агентами, а не лише з людьми.

- Міркування. Останній компонент зацікавлений у можливостях семантичних агентів міркувати та підтримувати прийняття рішень.

					БР.ІІІ – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Консорціум Всесвітньої Павутини (W3C, [www.w3c.org](http://www.w3c.org)) був піонером у розробці технологічних будівельних блоків Семантичного Вебу, включаючи XML та RDF (Resource Description Framework). Зовсім недавно вони створили робочу групу з веб-онтологій, яка розроблятиме та просуватиме мову веб-онтологій (див. нижче) для семантичної розмітки веб-даних та сервісів. Різноманітна спільнота дослідників, представлена академічними колами, урядом, промисловістю та незалежними розробниками, активно підтримує розвиток Семантичного Вебу.

### *1.5.1. Онтологія та метадані*

Онтологія визначається як "та галузь філософії, яка займається природою та організацією реальності". Онтологія визначається як "логічна теорія, яка дає явне, часткове представлення концептуалізації". Ще одне визначення, що підходить для Семантичного Вебу, – це "набір термінів знань, що включає словник, семантичні взаємозв'язки та деякі прості правила виведення та логіки, для певної теми". Чому нам потрібні онтології? Онтології захоплюють модель знань для певної предметної області. Вони дозволяють нам описувати ресурси в Інтернеті та взаємозв'язки між цими ресурсами. Наприклад, геопросторова онтологія включатиме концепції місцезнаходження, суміжності, включення та навігації (пошуку шляху). Особиста веб-сторінка може містити поштову адресу, написану в HTML, але це не має значення для агента. Однак, якщо адреса розмічена екземплярами геопросторової онтології, агент здатен вивести географічне розташування індивіда як в абсолютних термінах (координати), так і у відносних термінах (країна, округ, місто, район тощо).

Онтології — це метадані для вебу. Але наскільки метадані релевантні для підтримки рішень? СППР можуть надавати засоби для генерування ряду альтернативних рішень для порівняння та оцінки різних цілей. Для того, щоб СППР виконували свої задані функції в майбутньому, вони покладатимуться

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26



уському світі. Універсальні ідентифікатори ресурсів (URI) є глобально унікальними адресами ресурсів у Вебі і є надмножиною більш знайомих URL-адрес. Цьому шару можна віднести більшість сучасного Вебу. Наступний шар містить технології, що швидко розвиваються, для додавання структури до документів у Вебі; це XML, простори імен та XML-схеми. Ці технології об'єднуються для структурування документів як ієрархічних дерев на основі тегів, специфічних для предметної області. XML-схема, подібна до схеми реляційної бази даних, формалізує структуру та дозволяє машині читати та розбирати документи. Формальна структура документа дозволяє йому бути "самоописним".

Шар RDF та RDF-схем будується на шарі XML шляхом визначення мови опису метаданих. RDF — це проста мова представлення знань для вираження предикатної логіки. Вона складається з слабозв'язаних "трійок" суб'єкт-дієслово-об'єкт, які дозволяють творцям метаданих встановлювати взаємозв'язки між ресурсами, що охоплюють увесь Веб. Наприклад, "Меріленд розташований у Сполучених Штатах", "Я працюю над Проектом А" або "вода тече вниз" — це концептуальні твердження, які можуть бути виражені за допомогою посилань на відповідні суб'єкти (наприклад, "Меріленд"), дієслова (наприклад, "розташований у") та об'єкти ("Сполучені Штати"). Екземпляри відповідних суб'єктів, дієслів та об'єктів (колективно ресурсів) доступні через URI, які глобально та унікально їх ідентифікують. Отже, RDF дозволяє нам описувати дані (зображення, звуки, бази даних рельєфу тощо) у серіалізованому, машинозчитуваному вигляді.

Ґрунтуючись на виразних можливостях RDF, існують онтологічні словники, які описують специфічні асоціативні, причинно-наслідкові та функціональні взаємозв'язки між ресурсами певної предметної області. Семантика однієї онтології може значно відрізнятися від іншої онтології, особливо для омонімічних термінів. Наприклад, термін "поле" у спортивній онтології може посилатися на ігрове покриття, тоді як у фізичній онтології

						БР.ІП – 35.00.00.000 ПЗ	Арк.
							28
Змн.	Арк.	№ докум.	Підпис	Дата			

він може посылатися на електромагнітну властивість. Для кодування відповідної семантики з взаємозв'язками вищого порядку, такими як кардинальність та булеві операції, необхідні виразні онтологічні мови. Одним із прикладів такої мови є DARPA Agent Markup Language у поєднанні з Ontology Interchange Layer (DAML+OIL, див. [www.daml.org](http://www.daml.org)). Як розширення RDF, DAML+OIL включає ці конструкції з функціями для семантичного опису як даних, так і сервісів. DAML+OIL використовується як модель для мови веб-онтологій, яка буде розроблена W3C (згадана раніше), і була використана при створенні прототипів застосунків семантичного вебу.

Три найвищі шари представляють бачення можливостей Семантичного Вебу. Семантичні агенти, керовані індексованою онтологічною розміткою, повинні бути здатні виконувати базові міркування щодо фактів, закодованих у метаданих. Мови онтологій, такі як DAML+OIL, описуватимуть правила, застосовні до ресурсів з певної предметної області. Методології, запозичені зі спільнот штучного інтелекту та представлення знань (тобто логічне програмування та доводи теорем), дозволять семантичним агентам міркувати на основі правил та фактів, враховуючи семантичний контекст проблеми. Прикладом засобу міркування Семантичного Вебу є CWM. CWM здатний читати та розбирати синтаксис RDF та DAML+OIL і виконувати прямий вивід за допомогою простих правил Горна. В [5] автор надає ілюстративний приклад того, як взаємозв'язок довіри та доведення між агентами може бути виконаний для організації платежів для онлайн-транзакції. При покладанні на агентів для ведення бізнесу один з одним або прийняття рішень виникає низка проблем. Не останньою з цих проблем є безпека між комунікаціями агентів. Цифрові підписи пропонують певну міру захисту того, що факти, подані агенту міркування, є автентичними і не призначені для створення парадоксальних або іншим чином зловмисних ситуацій.

Модель багат шарового Семантичного Вебу узгоджується з процесами, що використовуються СППР (розвідка, дизайн та вибір). Можливості,

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

реалізовані на найвищих рівнях "багатошарового пирога", дозволять семантичним веб-агентам збирати дані на основі онтологічної розмітки, фільтрувати їх на основі атрибутів проблеми та цілей, міркувати, а потім висувати альтернативи на основі уподобань особи, що приймає рішення. Семантичний Веб пропонує можливість для підтримки рішень у неявно розподіленому, гетерогенному середовищі, де напівструктуровані проблеми прийняття рішень можуть бути вирішені ефективно та результативно за допомогою семантичних агентів.

У технічному документі W3C описано сценарій, коли особа зацікавлена у реєстрації на конференцію. У документі стверджується, що інформацію, таку як розклад, місце проведення та деталі конференції, важко зафіксувати за допомогою сучасних веб-інструментів. Наприклад, спроба навігації до конференції за допомогою портативного пристрою глобального позиціонування (GPS) є складною, оскільки користувач змушений вводити інформацію про пункт призначення та маршрутні точки вручну та заздалегідь. Семантичний опис назв місць та маршрутів уможливить навігацію від точки до точки у відкритому, інтероперабельному фреймворку.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. КОНЦЕПТУАЛЬНИЙ КОНТЕКСТ АНАЛІЗУ РІШЕНЬ, СЕМАНТИКИ, ОНТОЛОГІЙ ТА МОВИ ВЕБ-ОНТОЛОГІЙ

### 2.1. Семантика

Семантика відноситься до вивчення значення. Концепція може бути описана різними способами залежно від контексту, для якого вона визначена. Два поняття вважаються семантично еквівалентними, якщо вони мають ідентичне значення. Традиційним методом встановлення відношення тотожності є використання математичного апарату. У цьому підході термінологія та твердження відображаються на абстрактну математичну структуру, що зазвичай називається моделлю. Два терміни або твердження є однаковими, якщо вони відображаються на одну й ту ж модель.

Формальна семантика використовується для надання точного, однозначного значення знанням у певній предметній галузі. Визначення "точно" означає, що семантика не базується на суб'єктивних інтуїціях і не допускає множинних інтерпретацій різними особами чи машинами. Важливість формальної семантики добре встановлена в галузях, таких як математична логіка.

### 2.2. Онтологія

Однією з цілей даної роботи є документування аналізу рішень таким чином, щоб його можна було легко повторно використовувати як людьми, так і програмними агентами. Це досягається шляхом формального представлення термінології, відомого як онтологія. Онтологія визначає терміни, що використовуються для опису та представлення галузі знань. Онтології слугують спільним словником для людей, баз даних та програм, які потребують обміну доменною інформацією (домен — це конкретна

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

предметна область або галузь знань, наприклад, медицина, виробництво, нерухомість, фінансовий менеджмент тощо).

Онтології включають визначення основних концепцій у домені та взаємозв'язків між ними, які можуть бути оброблені комп'ютерами. Вони кодують знання в межах домену, а також знання, що охоплюють кілька доменів, тим самим забезпечуючи їх повторне використання. Інструменти онтологій Семантичного Вебу здатні виконувати автоматизоване міркування з використанням онтологій, надаючи розширені послуги інтелектуальним застосункам, зокрема: концептуальний/семантичний пошук та вибірка, програмні агенти, підтримка прийняття рішень, розуміння природної мови, управління знаннями, інтелектуальні бази даних та електронна комерція.

### 2.3. Мова веб-онтологій OWL

Мова Веб-Онтологій (OWL) була стандартизована консорціумом всесвітньої павутини (W3C) і наразі є широко прийнятою мовою для розробки онтологій.

OWL поділяється на три різні підмови (профілі), кожна з яких розроблена для задоволення потреб конкретних спільнот розробників та користувачів:

#### 1. OWL Full.

Цей профіль включає повний набір примітивів мови OWL і дозволяє їх довільне комбінування з RDF та RDF Schema. Перевагою OWL Full є його повна сумісність з RDF як синтаксично, так і семантично: будь-який дійсний документ RDF є дійсним документом OWL Full, і будь-який дійсний висновок RDF/RDF Schema також є дійсним висновком OWL Full. Це забезпечує максимальну виразність та синтаксичну свободу RDF, але без обчислювальних гарантій. Наприклад, в OWL Full клас може одночасно розглядатися як колекція індивідів і як індивід сам по собі [3].

						Арк.
					БР.ІП – 35.00.00.000 ПЗ	32
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2. OWL DL.

Для досягнення обчислювальної ефективності, OWL DL (скорочення від Description Logic) була створена як підмножина OWL Full. Вона обмежує використання конструкторів OWL та RDF, забезпечуючи відповідність мови добре вивченій логіці описів. Це дозволяє ефективну підтримку міркувань. Недоліком є часткова втрата повної сумісності з RDF: документ RDF зазвичай потребує розширення в одних аспектах та обмеження в інших, перш ніж він стане дійсним документом OWL DL. Водночас, кожен дійсний документ OWL DL є дійсним документом RDF.

## 3. OWL Lite.

Цей профіль є подальшою підмножиною OWL DL і підтримує ієрархію класифікації та прості обмеження. Наприклад, хоча він підтримує обмеження кардинальності, дозволяються лише значення кардинальності 0 або 1. Передбачається, що для OWL Lite буде простіше реалізувати інструментальну підтримку порівняно з двома іншими профілями.

### 2.4. Eclipse Process Framework

У даному дослідженні модель процесу реалізована з використанням Eclipse Process Framework (EPF) Composer. EPF є гнучким інструментом для конструювання та налаштування інженерних рамок процесу розробки програмного забезпечення, що включає прикладовий вміст процесу та інструментальну підтримку, яка адаптується до широкого спектру типів проектів та стилів розробки. Доступні розширювані рамки процесів в EPF Composer включають OpenUP/Basic, OpenUP/MDD та нещодавно інтегровані "інші agile компоненти". У цій роботі не передбачається розширення існуючих фреймворків; натомість, буде створена нова модель процесу.

EPF еволюціонував з Rational Unified Process (RUP). Підмножина RUP, відома як OpenUP, була опублікована IBM як проєкт з відкритим вихідним

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

кодом. OpenUP є мінімально достатнім процесом розробки програмного забезпечення, що означає, що він включає лише фундаментальний зміст. Відповідно, він не надає керівництва з багатьох тем, що можуть виникнути в проектах, таких як великі команди, питання відповідності, контрактні ситуації, безпека, критично важливі для місії програми, або технологічно-специфічні аспекти.

RUP та OpenUP є ітеративними процесами розробки програмного забезпечення. Це не лінійні, покрокові життєві цикли, а адаптивні рамки процесів, які дозволяють організації або команді розробників компонувати модель процесу відповідно до їхніх конкретних потреб. Така гнучка рамка є вкрай необхідною, оскільки жоден універсальний життєвий цикл розробки програмного забезпечення не є оптимальним для всіх видів розробки.

Фундаментальний принцип EPF полягає в розділенні вмісту методу та процесів. Вміст методу (Method Content) описує цілі, необхідні навички, різноманітні артефакти та результати без прив'язки до їх конкретного використання у життєвому циклі процесу. Після визначення вмісту методу, процеси (Processes) конструюються, описуючи послідовність виконання робіт різними ролями та deliverables, що будуть створені протягом життєвого циклу. Важливо, що роботи, ролі та результати визначені на рівні вмісту методу.

Вміст методу та процеси можуть бути:

- Повторно використані "як є".
- Налаштовані відповідно до вимог проекту, а потім повторно використані.
- Розширені у разі виникнення такої потреби.
- Адаптовані шляхом незначних модифікацій для відповідності специфічним вимогам.

Таке розділення визначення вмісту методу та процесів забезпечує незалежність їх повторного використання, розширення та адаптації.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2.5. Представлення діаграми впливу

Невизначеність є невід'ємною складовою будь-якого процесу прийняття рішень. Вона проявляється у ситуаціях, де обставини можуть змінюватися, наприклад, через коригування вимог замовником, що є частим явищем. У таких випадках виникає необхідність перегляду пов'язаних рішень. Повністю усунути невизначеність, пов'язану з прийняттям рішень, неможливо через її стохастичну природу. Однак, у міру зменшення невизначеності, може виникнути потреба у коригуванні початкового аналізу.

Діаграми впливу (Influence Diagrams - IDs) є систематичним, емпіричним методом для представлення невизначеності у рішенні. Формальною основою для діаграм впливу є теорія ймовірностей, і вони репрезентуються за допомогою байєсівських мереж (Bayesian Networks - BNs). Байєсівська мережа – це графічний механізм для визначення спільного розподілу ймовірностей набору випадкових змінних [7]. Таким чином, BN є фундаментальним механізмом представлення ймовірнісних моделей для стохастичних систем. Використання графів забезпечує інтуїтивно зрозумілий та візуально привабливий інтерфейс, за допомогою якого можна виражати складні стохастичні моделі. Ця графова структура також слугує основою для формату обміну стохастичними моделями і може бути використана при проєктуванні ефективних алгоритмів для видобутку даних, навчання та висновків. Діапазон потенційного застосування BNs є значним, і їхня популярність стрімко зростає, зокрема у біомедичних застосуваннях для діагностики захворювань.

Випадкові змінні у байєсівській мережі представлені у вигляді вузлів графа, а ребра позначають залежності між цими змінними. Це досягається шляхом визначення умовного розподілу ймовірностей для кожного вузла. Важливою умовою є те, що ребра BN не повинні утворювати спрямованого циклу, тобто BN є ациклічною. Якщо два вузли не пов'язані ребром, вони є

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

умовно незалежними. Цю властивість незалежності можна розглядати як наслідок наступної властивості BN: спільний розподіл ймовірностей (JPD) вузлів BN є добутком умовних розподілів ймовірностей (CPD) вузлів BN. Ця властивість також відома як правило ланцюгової ймовірності. Саме через це BN вважається ациклічною: правило ланцюгової ймовірності не може бути застосоване при наявності циклу. Коли BN є ациклічною, можна впорядкувати CPD таким чином, щоб визначення умовної ймовірності та статистичної незалежності могли бути застосовані для отримання серії скорочень, що зводить їх до JPD.

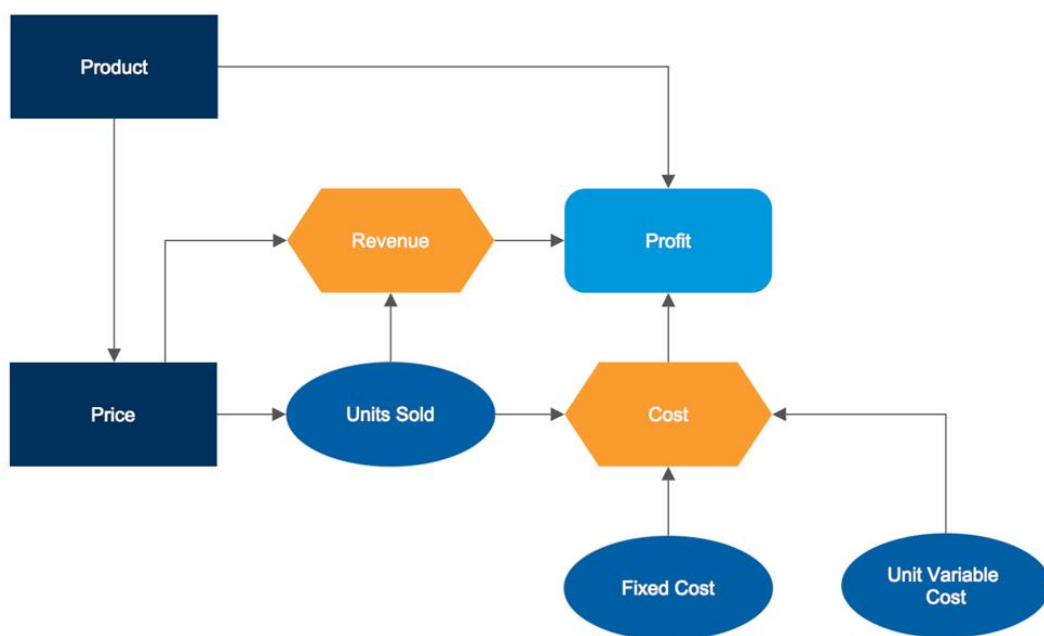


Рисунок 2.1 – Приклад діаграми впливу, яка моделює процес прийняття рішення щодо продукту

Ця діаграма (рис. 2.1) моделює процес прийняття рішення щодо продукту з метою оптимізації прибутку. Діаграми впливу використовуються для візуалізації залежностей між рішеннями, невизначеними подіями (шансовими змінними) та кінцевими результатами.

Діаграма складається з різних типів вузлів, які представляють різні елементи процесу прийняття рішень:

- Прямокутники (сині): Позначають вузли рішень, які контролюються особою, що приймає рішення. У цій діаграмі це:

- Product (Продукт): Рішення щодо самого продукту (можливо, його характеристик, типу тощо).

- Price (Ціна): Рішення щодо ціни продукту.

- Овали (сині): Позначають вузли невизначеності або детерміновані вузли (якщо їх значення повністю визначається вхідними даними). Тут це:

- Units Sold (Продані одиниці): Кількість проданих одиниць, яка залежить від рішень щодо продукту та ціни.

- Fixed Cost (Фіксовані витрати): Витрати, які не змінюються залежно від обсягу виробництва.

- Unit Variable Cost (Змінні витрати на одиницю): Витрати, що припадають на кожну вироблену одиницю.

- Шестикутники (помаранчеві): Позначають вузли проміжного значення або проміжні результати, які впливають на кінцевий вузол корисності. Тут це:

- Revenue (Виручка): Загальний дохід від продажу.

- Cost (Вартість/Витрати): Загальні витрати, пов'язані з продуктом.

- Округло-прямокутник (синій): Позначає вузол корисності (кінцевий результат), який є цільовою функцією для оптимізації. У даному випадку це:

-Profit (Прибуток): Кінцевий показник, який прагне максимізувати особа, що приймає рішення.

Стрілки вказують на прямий вплив або залежність одного вузла від іншого:

- Рішення щодо Product (Продукту) впливає на Price (Ціну) та Units Sold (Продані одиниці).

- Рішення щодо Price (Ціни) також впливає на Units Sold (Продані одиниці).

- Units Sold (Продані одиниці) прямо впливає на Revenue (Виручку).

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

- Fixed Cost (Фіксовані витрати) та Unit Variable Cost (Змінні витрати на одиницю) впливають на Cost (Вартість/Витрати).

- Revenue (Виручка) та Cost (Вартість/Витрати) спільно визначають Profit (Прибуток).

Загалом, діаграма впливу надає чітке візуальне представлення ключових змінних, рішень та їхніх взаємозв'язків, що є основою для систематичного аналізу та прийняття обґрунтованих рішень щодо продукту з метою максимізації прибутку.

Діаграма впливу - це спосіб опису залежностей між алеаторними змінними та рішеннями. Її можна використовувати для візуалізації ймовірнісних залежностей в аналізі рішень та для визначення станів інформації, для яких можна припустити існування незалежностей. Вона походить від байєсівських мереж, але має високий ступінь анотації, що є ключовою вимогою для розуміння та повторного використання аналізу рішень. Діаграма впливу є мережею, що складається зі спрямованого графа, який має три типи вузлів:

- Вузол корисності або вузол значення (не більше одного), що представляє величину, яку необхідно максимізувати.

- Вузли випадковості (нуль або більше), що представляють випадкові змінні.

- Вузли рішень (нуль або більше), що представляють альтернативи, доступні особі, яка приймає рішення [10].

Діаграма в цілому кодує критерії. Оцінка діаграми для кожної альтернативи з метою отримання значень корисності для порівняння є аргументом у процесі прийняття рішення. Існують різні алгоритми для оцінки діаграм впливу. Зокрема, Шахтер розробив алгоритм, який може оцінювати будь-яку коректно сформовану діаграму впливу та визначати оптимальну політику для її рішення.

## 2.6. Архітектура інтероперабельної DSS системи на базі семантичних веб-технологій

Пропонується архітектура інтероперабельної DSS системи підтримки рішень, заснована на семантичних веб-технологіях, як показано на рисунку 2.2. Основні цілі цієї архітектури:

- а) забезпечити обмін даними та обчислювальними послугами на семантичному рівні через веб;
- б) максимізувати продуктивність та ефективність обміну даними та послугами;
- в) подолати проблеми дублювання та підтримки даних;
- г) спростити інтеграцію з іншими застосунками СПР.

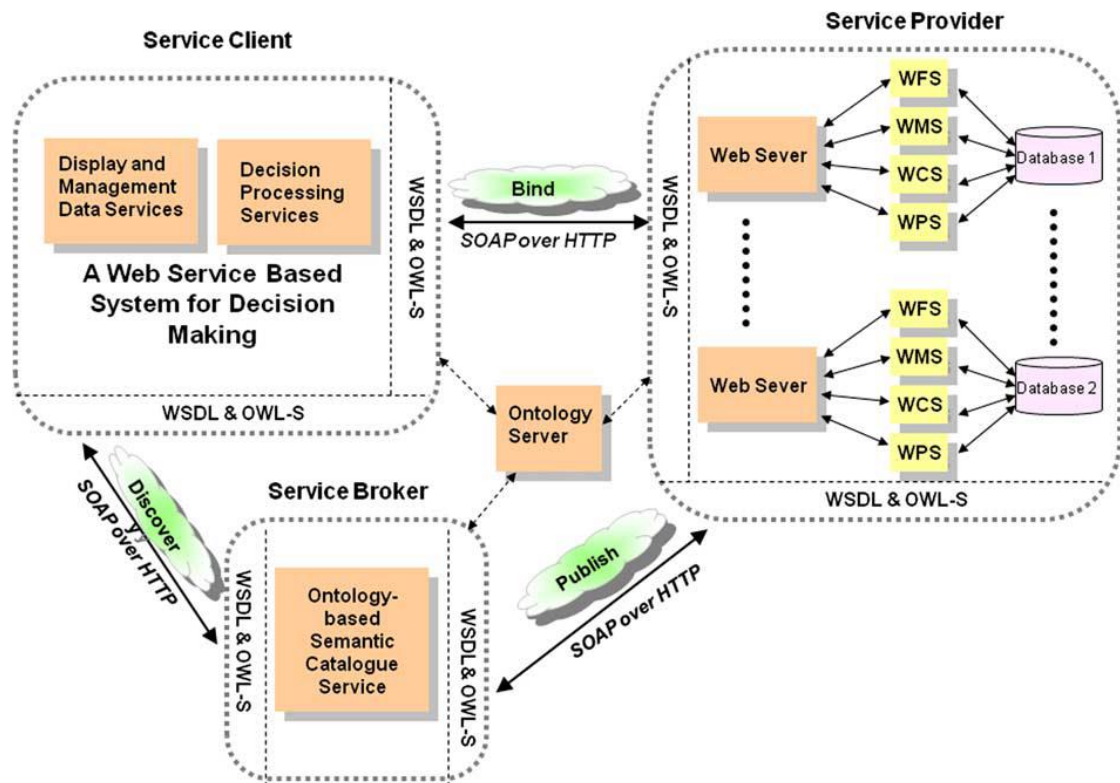


Рисунок 2.2 - Архітектура інтероперабельної DSS системи

Ця архітектура базується на незалежних семантичних веб-сервісах та сервіс-орієнтованій архітектурі (COA). Структура, заснована на COA,

складається з чотирьох елементів: постачальника послуг, брокера послуг, клієнта послуг та онтологічного сервера. Рисунок 2.3 ілюструє чотири компоненти запропонованої структури.

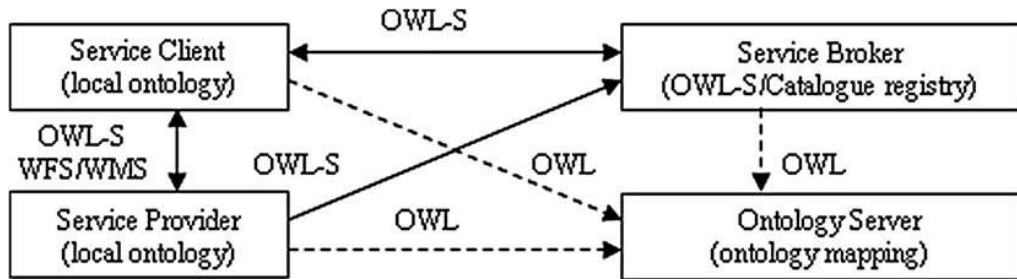


Рисунок 2.3 – Компоненти структури

Постачальник послуг надає гетерогенні обчислювальні та інформаційні сервіси з розрізнених джерел через стандартизовані веб-сервіси та сервіси даних. Сервіси обробки рішень та сервіси даних можуть бути об'єднані в ланцюжок для побудови специфічних сервісів підтримки рішень.

Клієнт послуг допомагає особам, які приймають рішення, відображати та керувати сервісами даних, а також отримувати доступ до сервісів обробки рішень для генерації та оцінки альтернативних рішень напівструктурованих проблем.

Брокер послуг надає реєстр доступних послуг. Брокер послуг використовує семантичні каталожні послуги (CS) на основі онтологій для реєстрації та управління сервісами даних та обробки, а також для того, щоб дозволити користувачам шукати ці сервіси на семантичному рівні.

Онтологічний сервер забезпечує семантичну інтероперабельність онтологій клієнтів та постачальників послуг. Веб-сервіси з'єднані за допомогою Web Service Description Language (WSDL) та OWL-S між постачальником послуг, брокером послуг та клієнтом послуг. Для комунікації між веб-сервісами через Інтернет використовується прив'язка Simple Object Access Protocol (SOAP) поверх HTTP.

На рисунку 2.4 проілюстровано процес пошуку еквівалентних класів та властивостей онтології. Спочатку ми об'єднуємо всі набори класів, об'єктних властивостей та властивостей даних як з локальних онтологій, так і з серверних. Потім ми ініціалізуємо розбиття класів, об'єктних властивостей та властивостей даних для формування еквівалентного розбиття. Після цього ми спочатку уточнюємо розбиття класів та об'єктних властивостей на основі їхньої структури. Якщо виявляється забагато еквівалентних класів та об'єктних властивостей, ми уточнюємо розбиття на основі їхніх назв. Процес уточнення продовжуватиметься, доки розбиття не стануть стабільними. Нарешті, ми отримуємо кінцеві стабільні розбиття, які не змінюватимуться і вважаються еквівалентними класами та властивостями онтології.

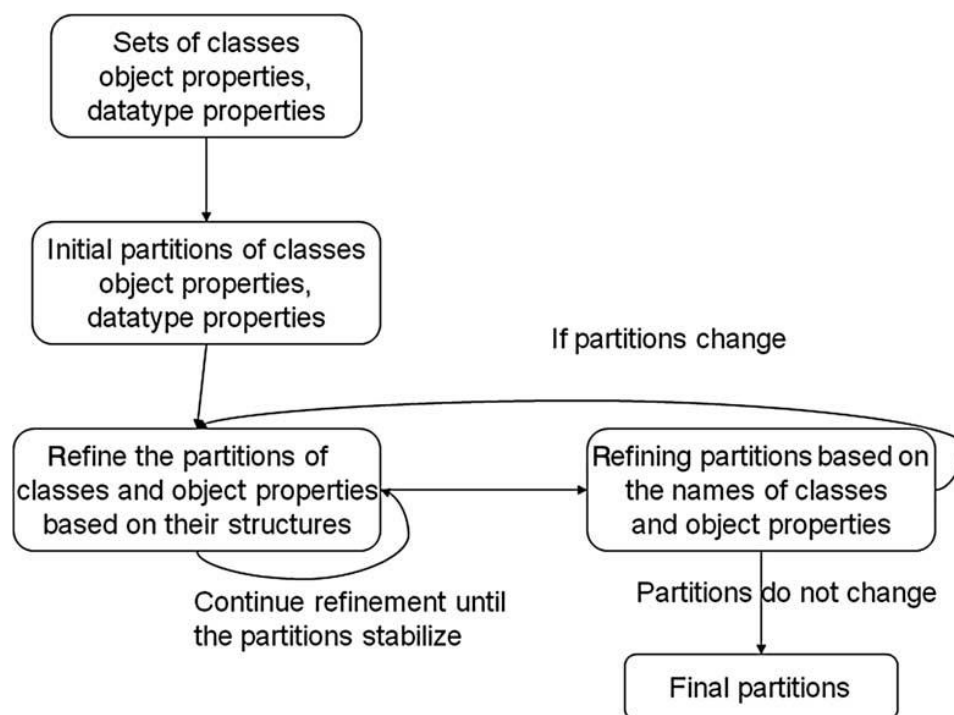


Рисунок 2.4 - Процес пошуку еквівалентних класів та властивостей онтології шляхом уточнення розбиття

Описаний вище алгоритм завжди завершується, оскільки кожна ітерація розбиває щонайменше один набір у розбиттях класів або об'єктних

властивостей. Ця проблема схожа на пошук структурно еквівалентних рекурсивних типів для об'єктно-орієнтованих мов програмування, і її часова складність була показана як  $N \log N$ , де  $N$  — це загальна кількість класів та властивостей.

## 2.7. Композиція веб-сервісів DSS системи

Для ефективного використання веб-сервісів користувачам необхідно динамічно аналізувати їхні функціональні можливості та оцінювати їхню застосовність до поточного завдання. Іноді може виникнути потреба динамічно зв'язувати компонентні сервіси або викликати їх у певній послідовності під час виконання, щоб задовольнити індивідуальні потреби. Динамічний характер доступності веб-сервісів, велика кількість альтернативних комбінацій вибору сервісів та вимоги реального часу до композиції сервісів роблять розробку інтегрованого фреймворку для просторово-орієнтованої системи підтримки рішень (ПОСПР) на базі веб-сервісів складним завданням.

У запропонованому фреймворку для полегшення динамічної композиції сервісів необхідні два кроки: по-перше, на основі запиту користувача, знайти можливі сервіси, спираючись на адекватні описи. Багата семантика, додана до описів веб-сервісів за допомогою OWL-S, уможливорює семантичну композицію шляхом зіставлення описів можливостей сервісів на семантичному рівні з вимогами. По-друге, на основі необхідних обмежень керування та потоку даних між сервісами, розробити план сервісів для ініціювання виконання сервісів у правильній послідовності. План сервісів може створюватися та оновлюватися динамічно і базується на централізованому брокері, який керує процесом композиції сервісів. План сервісів має визначати, які запити сервісів можуть бути реально оброблені, а які не можуть бути виконані. Коли один сервіс не може задовольнити запит

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

користувача, план сервісів взаємодіятиме та співпрацюватиме з іншими сервісами, щоб завершити композиційний сервіс для запиту. Таким чином, запропонована інтероперабельна ПОСПР на базі веб-сервісів демонструє такі системні характеристики, як самоорганізація, еволюція, масштабованість та адаптивність, хоча кожен з веб-сервісів сам по собі не призначений для цього.

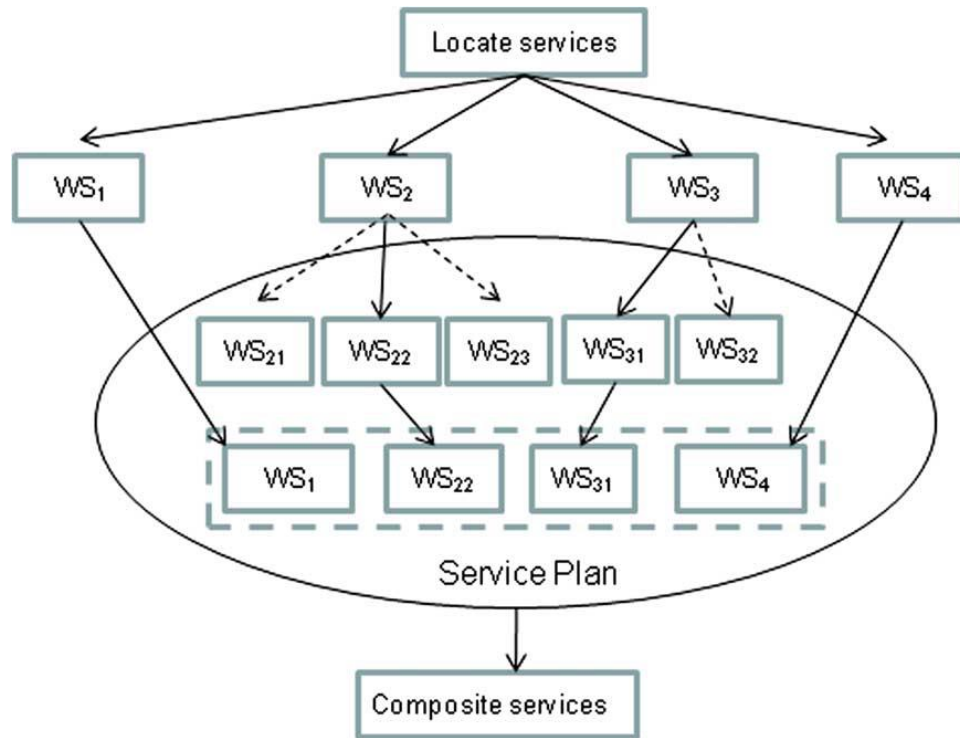


Рисунок 2.5 - Процес композиції веб-сервісів

Рисунок 2.5 ілюструє процес композиції веб-сервісів. Він ґрунтується на двох процесах: процесі виявлення сервісів та процесі композиції сервісів. Виявлення сервісів базується на семантичному описі сервісу і автоматично визначає сервіси даних та обчислювальні сервіси, які точно відповідатимуть потребам прийняття рішень. Якщо окремий сервіс даних та обчислювальний сервіс не можуть бути знайдені, ПОСПР шукатиме два або більше сервісів, які можна скомпонувати для синтезу необхідного сервісу. Це називається композицією сервісів. Алгоритм виявлення та композиції, запропонований у даній ПОСПР, ілюструється наступним чином:

Алгоритм: Виявлення та композиція сервісів

Вхідні дані:

- Веб-сервіси ПОСПР, які потребують даних просторових об'єктів та надають результати оцінки для підтримки просторових рішень.
- WFS-сервіси, які надають просторові об'єкти.
- Онтологічні сервіси, які надають онтологічні визначення для просторових об'єктів та критеріїв для ПОСПР.
- Сервісний запит Q щодо підтримки просторових рішень.

Вихідні дані: Комбінація WFS, онтологічних веб-сервісів та веб-сервісів ПОСПР.

Крок 1: Прийняти сервісний запит Q та звернутися до веб-сервісів онтологій, щоб знайти назви веб-сервісів ПОСПР, які можуть виконати запити щодо просторового рішення.

Крок 2: Запитати веб-сервіси ПОСПР, щоб знайти просторові об'єкти, необхідні сервісам для обчислення просторового рішення.

Крок 3: Знову звернутися до веб-сервісів онтологій, щоб знайти назви WFS-сервісів та їхніх просторових об'єктів, які необхідні веб-сервісам ПОСПР.

Крок 4: Зіставити онтологічні визначення для просторових об'єктів, необхідних ПОСПР та наданих WFS-сервісами.

Крок 5: Перетворити просторові об'єкти, надані WFS-сервісами, у формати, прийнятні для веб-сервісів ПОСПР.

Крок 6: Зробити запит до веб-сервісів ПОСПР з перетвореними просторовими об'єктами WFS для отримання результатів.

## 2.8. Аналіз рішень

Аналіз рішень (DA - Decision Analysis) включає стратегії, методи та інструменти, необхідні для ідентифікації, чіткого представлення та

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

формальної оцінки важливих аспектів ситуації прийняття рішення. Його метою є призначення рекомендованого курсу дій шляхом застосування аксіоми дії максимальної очікуваної корисності до коректно сформованого представлення рішення, а також перетворення формального представлення рішення та відповідної рекомендації на інсайти для особи, яка приймає рішення, та інших зацікавлених сторін.

Незважаючи на те, що обґрунтування дизайну часто документуються архітекторами програмного забезпечення, причини вибору певного дизайну та його переваги над іншими альтернативними рішеннями, як правило, не фіксуються належним чином [11].

Якщо аналіз рішення, включаючи розгляд альтернатив, їх переваг та недоліків, був проведений у минулому, вкрай важливо забезпечити його подальше використання у майбутніх подібних ситуаціях, щоб уникнути повторної роботи "з нуля". Ігнорування ефективних методів прийняття рішень або продовження використання неефективних може призвести до негативних наслідків.

## 2.9. Порядок виконання проекту

Наш підхід відрізняється тим, що ми пропонуємо анотувати не тільки аналіз рішення в цілому, але й його складові частини. Реалізація цієї онтології виконана на базі OWL (Web Ontology Language).

Реалізація дослідження здійснювалася згідно з наступним планом:

1. Ідентифікація проблем. Початковим етапом було виявлення ключових проблем, з якими стикаються користувачі при роботі з аналізом рішень, зокрема відсутність чіткого визначення документації аналізу рішень у моделях процесу життєвого циклу розробки.

2. Моделювання процесу. Наступним кроком було створення представлення моделі процесу аналізу рішень за допомогою Eclipse Process

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Framework (EPF). Це дозволило забезпечити легку інтеграцію з існуючим життєвим циклом розробки програмного забезпечення.

3. Проектування онтології. Після формування чіткого розуміння моделі процесу, було розпочато проектування онтології рішень. Ця онтологія слугує основою для анотування аналізу рішень, забезпечуючи його формалізоване представлення.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						46
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 3. ПОБУДОВА ОНТОЛОГІЙ DSS СИСТЕМИ НА ОСНОВІ КОНЦЕПЦІ СЕМАНТИЧНОЇ МЕРЕЖІ

### 3.1. Використання онтологічного редактора Protege

Protege є онтологічним редактором та фреймворком, розробленим у Стенфордському медичному інформаційному центрі. Він широко застосовується для створення та маніпулювання онтологіями, зокрема в біомедичній галузі. Protege підтримує збереження онтологій у різних форматах, включаючи CLIPS, RDF, XML, UML та, відповідно до поточних стандартів, OWL.

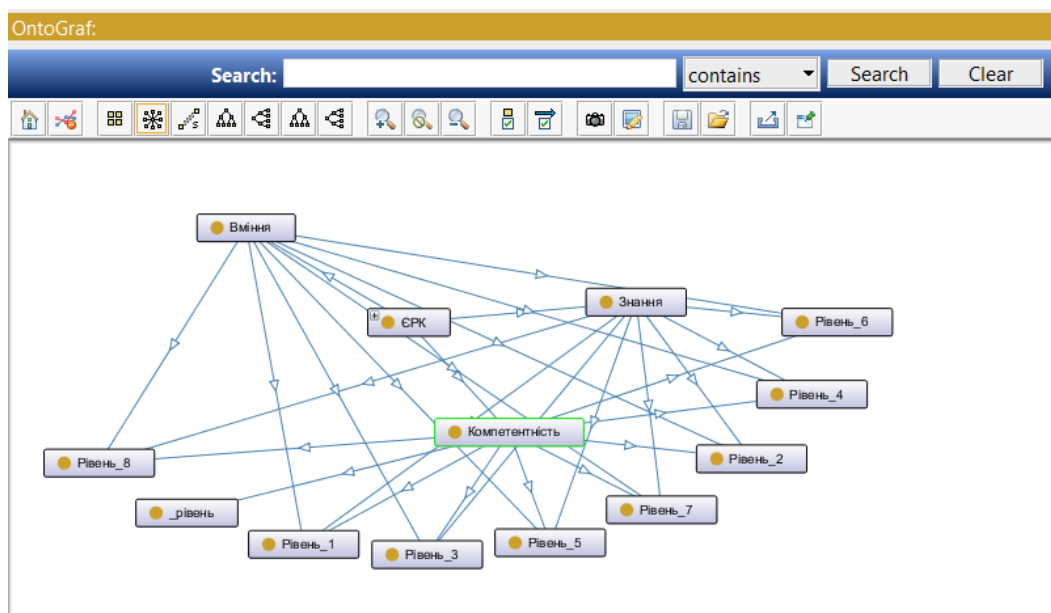


Рисунок 3.1 – Створення онтології засобами Protege

Використання OWL для реалізації онтології, незважаючи на складність та багатослівність його синтаксису, значно спрощується завдяки інструментам, таким як Protege. Він не тільки надає графічний інтерфейс користувача (GUI) для проектування онтології, але й автоматично генерує форми для створення індивідів (екземплярів онтології). Навігація в Protege здійснюється за допомогою восьми вкладок:

- Вкладка *Active Ontologies* використовується для управління анотаціями онтології, метриками, імпортованими онтологіями, загальними аксіомами, а також для рендерингу в форматах *RDF/XML*, *OWL/XML* та функціональному синтаксисі *OWL*.

- Вкладка *Entities* призначена для роботи з усіма типами сутностей, такими як класи, об'єктні властивості, властивості даних та індивіди. Друга частина цієї вкладки динамічно змінюється залежно від обраного типу сутності, при цьому для кожного з чотирьох основних типів сутностей також передбачені окремі вкладки.

### 3.2. Використання *Смар* для побудови діаграм

*СмарTools* — це програмне середовище, розроблене в Інституті знань людини та машини (ІНМС - *Institute for Human and Machine Cognition*), яке дозволяє візуалізувати моделі знань, зокрема розроблену онтологію, за допомогою концептуальних карт.

Його основне призначення — дозволити користувачам створювати, ділитися та керувати концептуальними картами. Концептуальні карти є графічними інструментами для організації та представлення знань. Вони складаються з вузлів (концептів), які представляють ідеї, та ліній, що зв'язують ці концепти, з текстовими мітками, що пояснюють взаємозв'язки між ними.

*СмарTools* надає інтуїтивно зрозумілий інтерфейс для:

- Візуалізації знань. Перетворення складної інформації у зрозумілі та структуровані діаграми.

- Організації інформації. Допомагає структурувати думки, ідеї та відносини між ними.

- Співпраці. Дозволяє користувачам спільно працювати над концептуальними картами.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

- Публікації та обміну. Можливість публікувати карти в інтернеті або обмінюватися ними в різних форматах.

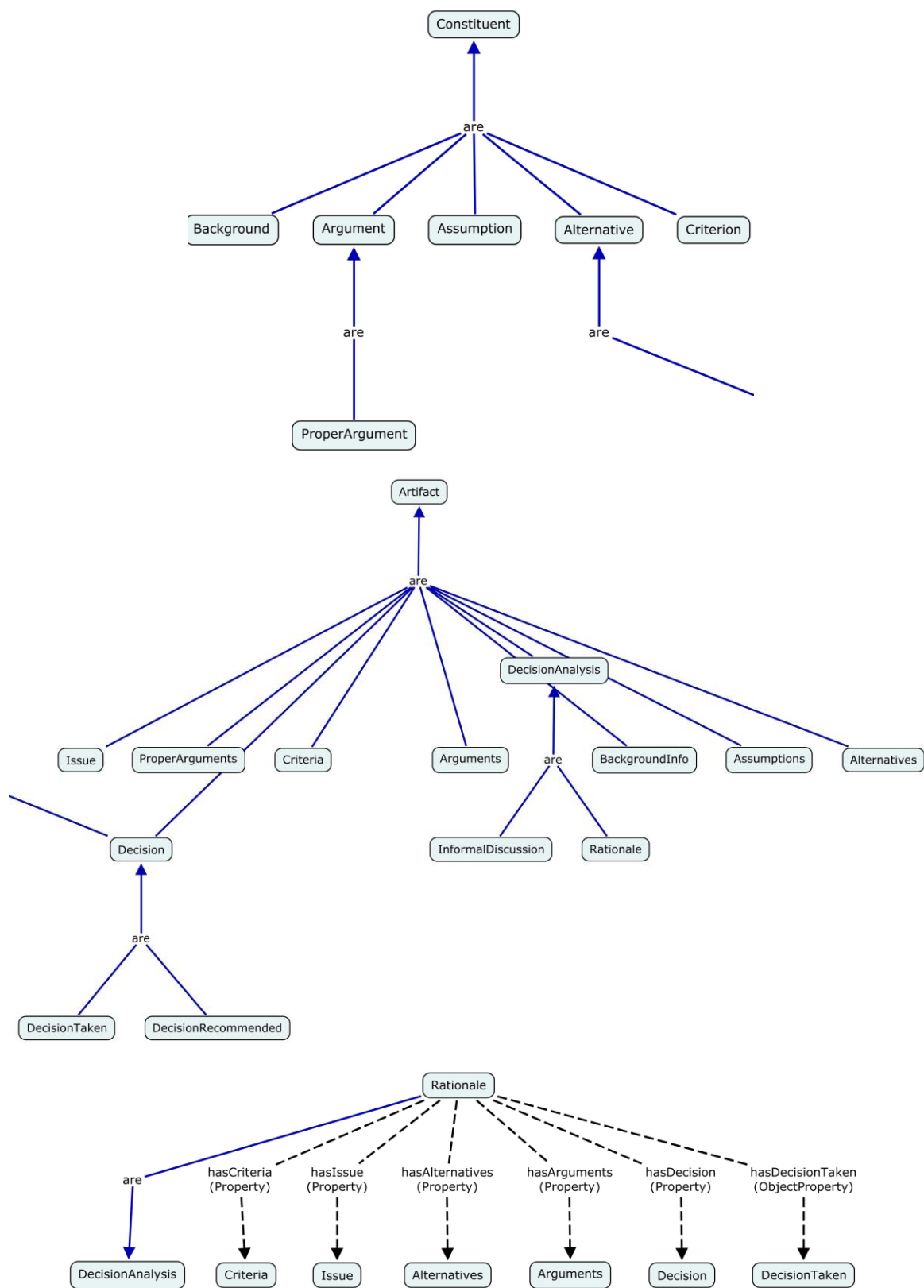


Рисунок 3.2 – Розроблена онтологія

Змн.	Арк.	№ докум.	Підпис	Дата

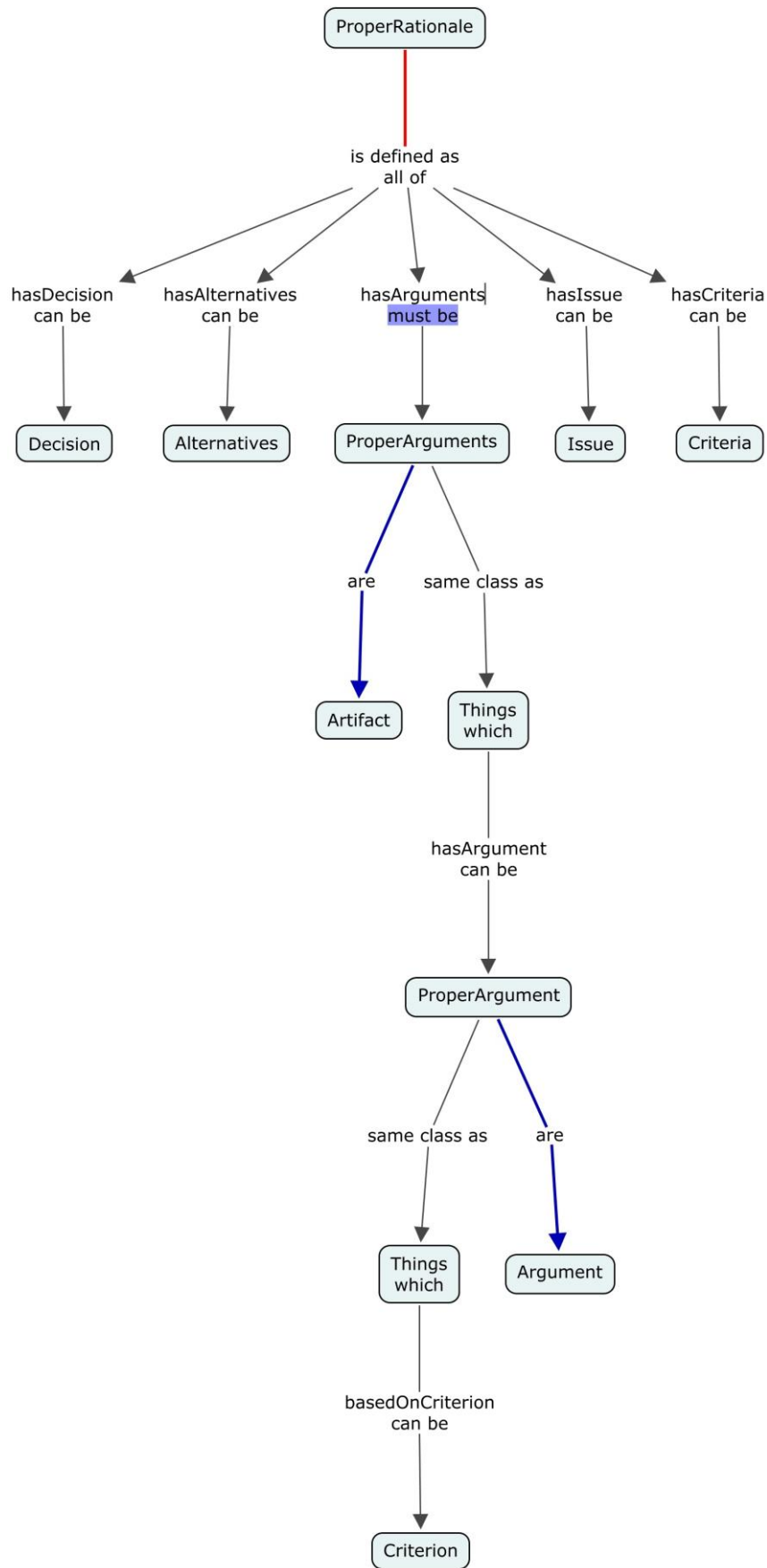


Рисунок 3.2 – Розроблена онтологія

Змн.	Арк.	№ докум.	Підпис	Дата

### 3.3. Методологія для формалізації та інтеграції аналізу рішень

Даний розділ присвячений опису методології, що використовується для формалізації та інтеграції аналізу рішень. Методологія ґрунтується на розробці онтології як фундаментального елемента для подальшого моделювання процесів аналізу рішень.

Опис аналізу рішення, включаючи його основні причини та обґрунтування, називається обґрунтуванням (Rationale). Ключові складові обґрунтування включають:

1. Питання: Проблема або задача, що потребує вирішення.
2. Аналіз: Включає всю відповідну контекстну інформацію та її взаємозв'язки.
  - а) Контекст: Релевантна інформація про середовище, в якому приймається рішення.
  - б) Припущення: Твердження, що приймаються як істинні для цілей аналізу, базуючись на контексті або поточній ситуації.
  - в) Аргумент: Твердження, що підтримує або заперечує одну або кілька альтернатив на основі певних критеріїв. У неформальному представленні аналіз складається з таких тверджень, тоді як у формалізованому вигляді аналіз виражається за допомогою діаграм або таблиць, що кількісно або якісно показують ступінь підтримки альтернатив критеріями.
3. Критерії: Вимоги або стандарти, необхідні для вибору оптимального варіанту серед усіх доступних альтернатив.
4. Альтернатива: Доступні варіанти або вибори, з яких може бути прийнято рішення. Формальний аналіз рішень може бути представлений за допомогою таких механізмів, як діаграми впливу, дерева рішень або таблиці.
5. Рішення/Пропозиція: Опис остаточного рішення або запропонованого курсу дій.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51



перегляд рішень та здійснення висновків), які не були розроблені в межах даного дослідження.

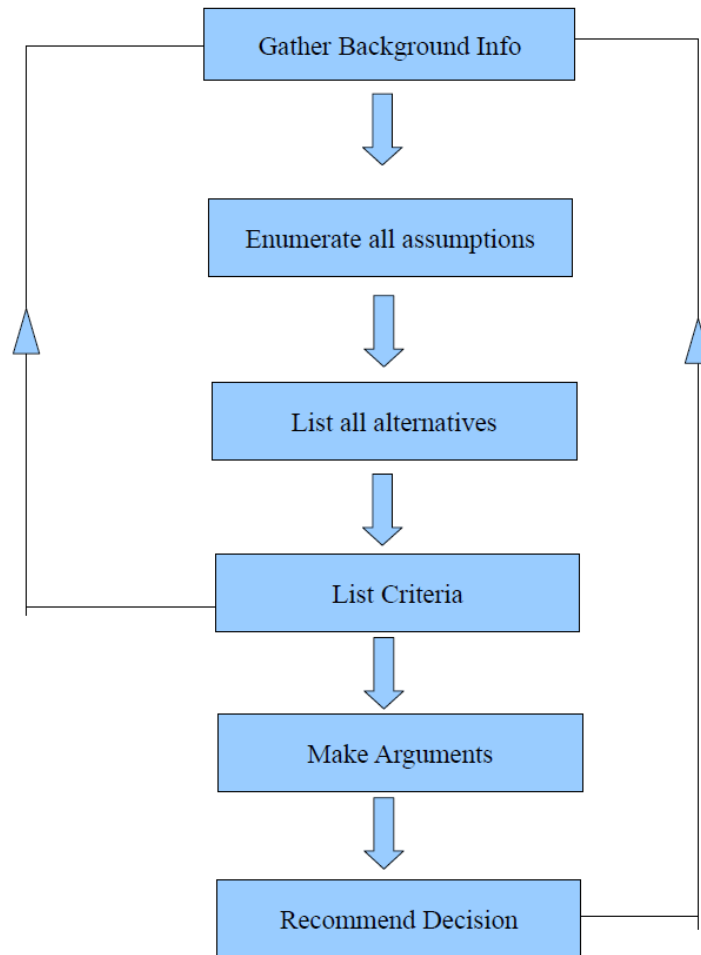


Рисунок 3.4 – Послідовний процес для структурованого підходу до прийняття рішень

Рисунок 3.4 відображає алгоритм (або послідовний процес) для структурованого підходу до прийняття рішень, що може бути застосований, наприклад, для оцінки ефективності нового продукту. Цей процес є ітеративним і складається з шести основних кроків:

1. Збір фонові інформації (Gather Background Info):

- Процес починається зі всебічного збору всієї релевантної контекстної та вихідної інформації, що стосується проблеми або рішення, яке необхідно прийняти.

2. Перерахувати всі припущення (Enumerate all assumptions):

- На основі зібраної фонові інформації ідентифікуються та перераховуються всі відповідні припущення, які можуть впливати на процес прийняття рішення.

3. Перерахувати всі альтернативи (List all alternatives):

- На цьому кроці визначаються та складається вичерпний перелік усіх можливих варіантів або виборів, доступних для прийняття рішення.

4. Перерахувати критерії (List Criteria):

- Визначаються та перераховуються критерії або вимоги, які будуть використовуватися для оцінки та порівняння перерахованих альтернатив.

5. Формування аргументів (Make Arguments):

- Для кожної альтернативи розробляються аргументи, які обґрунтовують її переваги або недоліки, як правило, на основі того, наскільки добре вона відповідає визначеним критеріям.

6. Рекомендувати рішення (Recommend Decision):

- На підставі сформованих аргументів та оцінки альтернатив за критеріями формується остаточне рішення або рекомендація.

Діаграма також ілюструє ітеративний характер цього алгоритму за допомогою зворотних зв'язків:

- Стрілка від кроку "Перерахувати критерії" до "Збір фонові інформації" вказує на те, що процес збору інформації, визначення припущень, альтернатив та критеріїв може бути ітеративним. У разі потреби перегляду критеріїв або альтернатив, може виникнути необхідність повернутися до початкового збору інформації або перегляду припущень.

- Стрілка від кроку "Рекомендувати рішення" також веде назад до "Збір фонові інформації", що означає, що весь цикл процесу (від збору інформації до рекомендації рішення) може бути повторений. Це підкреслює, що якщо рекомендоване рішення виявляється незадовільним або виникають нові дані,

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

весь процес може бути ініційований знову з уточненням вхідних даних або підходів.

Цей алгоритм представляє собою системний та структурований підхід до вирішення проблем та прийняття рішень, який наголошує на важливості ретельного збору даних, чіткого визначення варіантів та метрик оцінки, а також логічного обґрунтування.

### **3.4. Алгоритмічна реалізація логічного виведення (реазонатора) в DSS системі**

У контексті семантичного вебу та систем представлення знань, реазонер (reasoner), або логічний вивідник, — це програмна система або програмний модуль, призначений для автоматичного виведення нових логічних наслідків (імплікацій) із наявного набору фактів та аксіом у базі знань або онтології.

Основні функції реазонера включають:

- Перевірка на несуперечність (Consistency Checking) - визначення, чи містить онтологія або база знань логічні протиріччя, тобто, чи є вона логічно послідовною.

- Перевірка задовольняє мості / класифікованості (Satisfiability/Classifiability Checking) - встановлення, чи може певний концепт (клас) мати екземпляри без виникнення протиріч, або чи може екземпляр бути належним до певного класу.

- Перевірка субсумпції (Subsumption Checking) - визначення, чи є один клас підкласом іншого, виходячи з їхніх формальних визначень. Це дозволяє будувати ієрархії класів автоматично.

- Перевірка екземплярів (Instance Checking) - встановлення, чи є певний індивід (екземпляр) членом певного класу.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

- Відповіді на запити (Query Answering) - надання відповідей на логічні запити до бази знань, включаючи виведення неявно присутньої інформації.

Protege інтегрує декілька реазонерів, що дозволяють здійснювати логічні висновки на основі онтологій. Серед них, зокрема, виділяються Pellet та FaCT++.

Pellet є провідним повнофункціональним реазонером для OWL-DL, який характеризується широкою підтримкою міркувань з індивідами, включаючи номінали та кон'юнктивні запити. Він також надає функціонал для роботи з користувацькими типами даних та засобами налагодження онтологій. Pellet реалізує кілька розширень OWL-DL, серед яких формалізм комбінування для онтологій OWL-DL, немонотонний оператор та попередня підтримка гібридного міркування OWL/Rule.

### 3.4.1. Практичний сценарій

Для демонстрації застосовності запропонованої методології було обрано прикладний випадок, запозичений з роботи [10]. Цей приклад охоплює аналіз рішення демократичного кандидата у президенти щодо вибору напарника для участі у президентських виборах у США 1988 року.

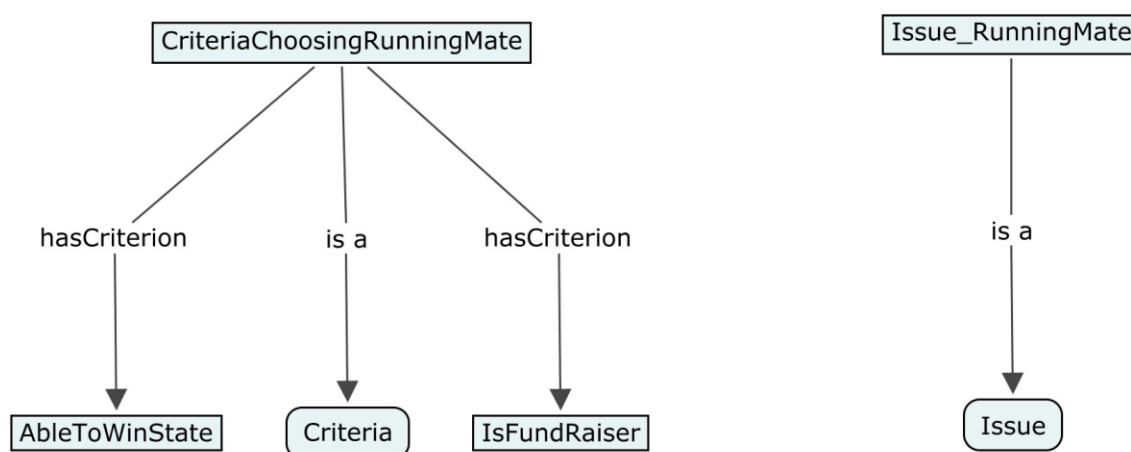


Рисунок 3.5 – Фрагмент онтології, що моделює ключові аспекти процесу прийняття рішення

Діаграма складається з двох окремих, але взаємопов'язаних графів, які ілюструють визначення критеріїв та формулювання проблеми.

1. Ліва частина діаграми - критерії вибору напарника (Criteria for Choosing Running Mate)

Цей граф відображає концепт "CriteriaChoosingRunningMate" (Критерії Вибору Напарника) та його взаємозв'язки з конкретними критеріями:

- "CriteriaChoosingRunningMate" є основним концептом, що представляє набір критеріїв.

Від цього концепту відходять три стрілки, що позначають зв'язки:

- "hasCriterion" вказує на конкретні критерії:

- "AbleToWinState" (здатність перемагати в штаті) - критерій, ймовірно, стосується здатності кандидата на посаду віце-президента допомогти виграти вибори в певних штатах.

- "IsFundRaiser" - критерій вказує на здатність кандидата ефективно залучати фінансові ресурси для кампанії.

- "is a" (є), що пов'язує "CriteriaChoosingRunningMate" з більш загальним концептом "Criteria", що показує, що "Критерії вибору напарника" є спеціалізованим типом загальних "Критеріїв".

2. Права частина діаграми - проблема вибору напарника (Issue of Running Mate)

Цей окремий граф ілюструє формулювання проблеми:

- "Issue\_RunningMate" - представляє конкретну проблему, пов'язану з вибором напарника.

Від цього концепту відходить стрілка "is a" (є), що пов'язує його з більш загальним концептом "Issue", що вказує на те, що "Проблема\_Напарник" є спеціалізованим типом загальної "Проблеми".

Ця схема є прикладом представлення знань в онтології, використовуючи концептуальні зв'язки. Вона демонструє, як специфічні критерії для прийняття рішення (вибір напарника) можуть бути визначені та

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

пов'язані з більш загальними категоріями критеріїв, а також як конкретна проблема може бути класифікована в рамках ширшої концепції проблеми. Таке формалізоване представлення дозволяє системі логічно обробляти та аналізувати інформацію, пов'язану з процесом прийняття рішення.

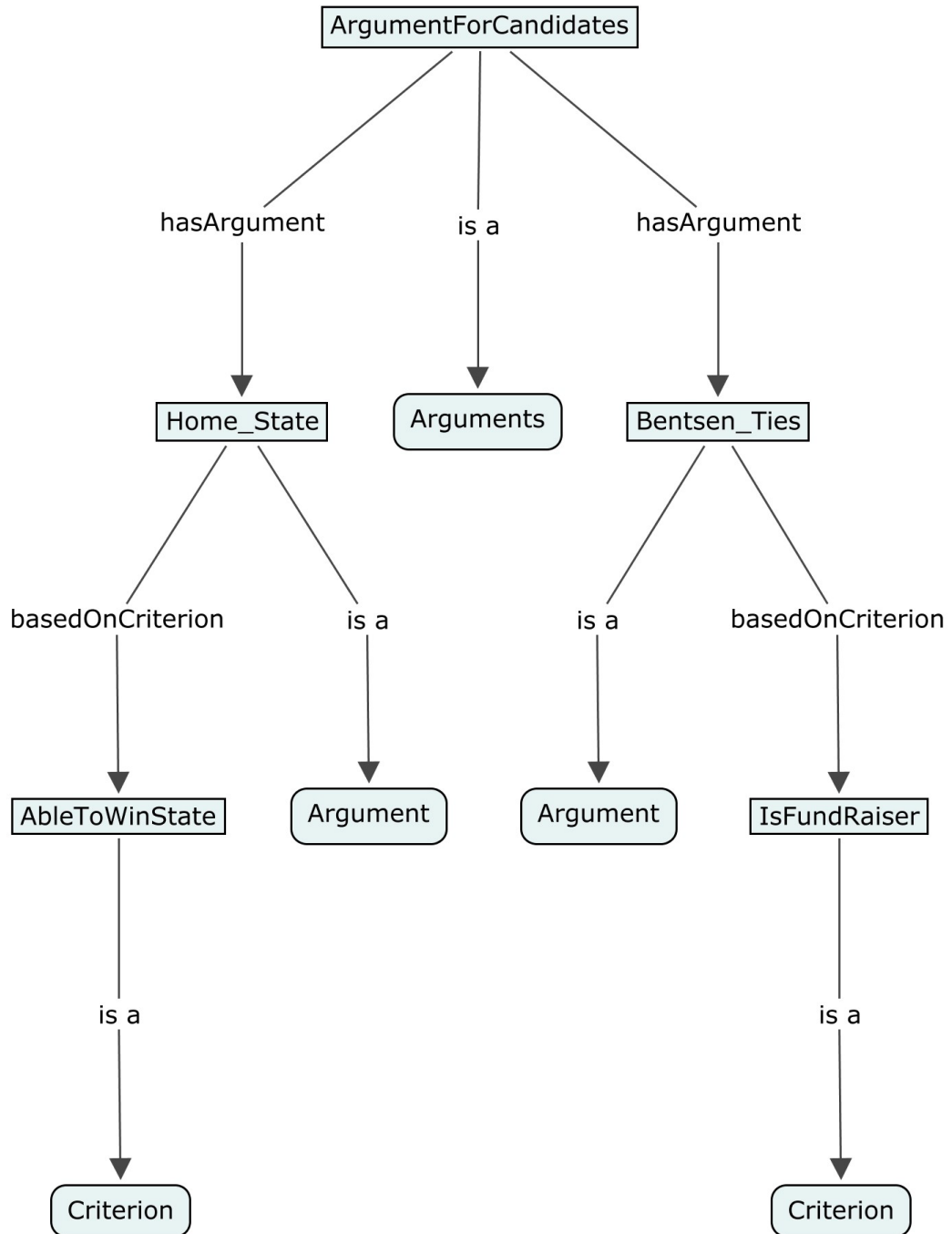


Рисунок 3.5 – Фрагмент онтології, що моделює ключові аспекти процесу прийняття рішення



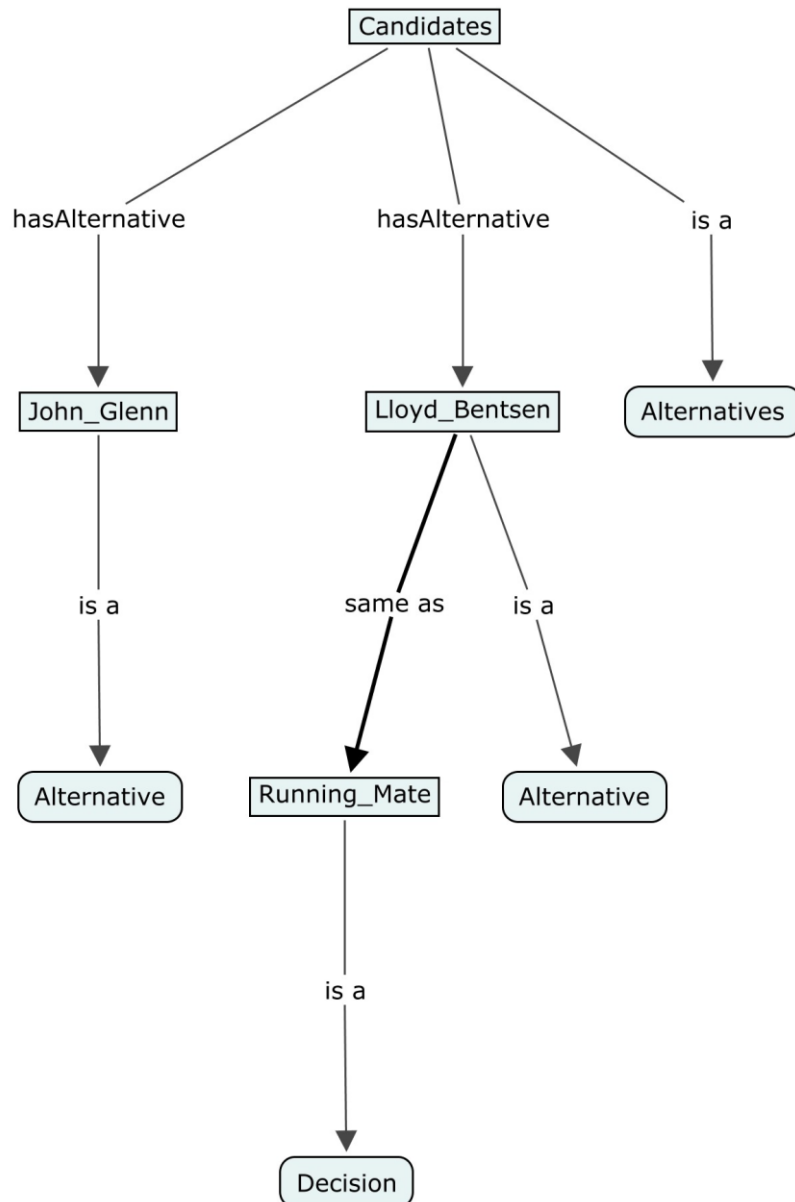


Рисунок 3.7 – Фрагмент онтології, що моделює ключові аспекти процесу прийняття рішення

Ми анотували цей аналіз за допомогою нашої онтології та виконували міркування за допомогою Pellet. Реазонер виводить, що екземпляр належить до класу ProperRationale, якщо він задовольняє всі обмеження, згадані в класі ProperRationale. Всі виведені об'єкти відображаються як жовті прямокутники.

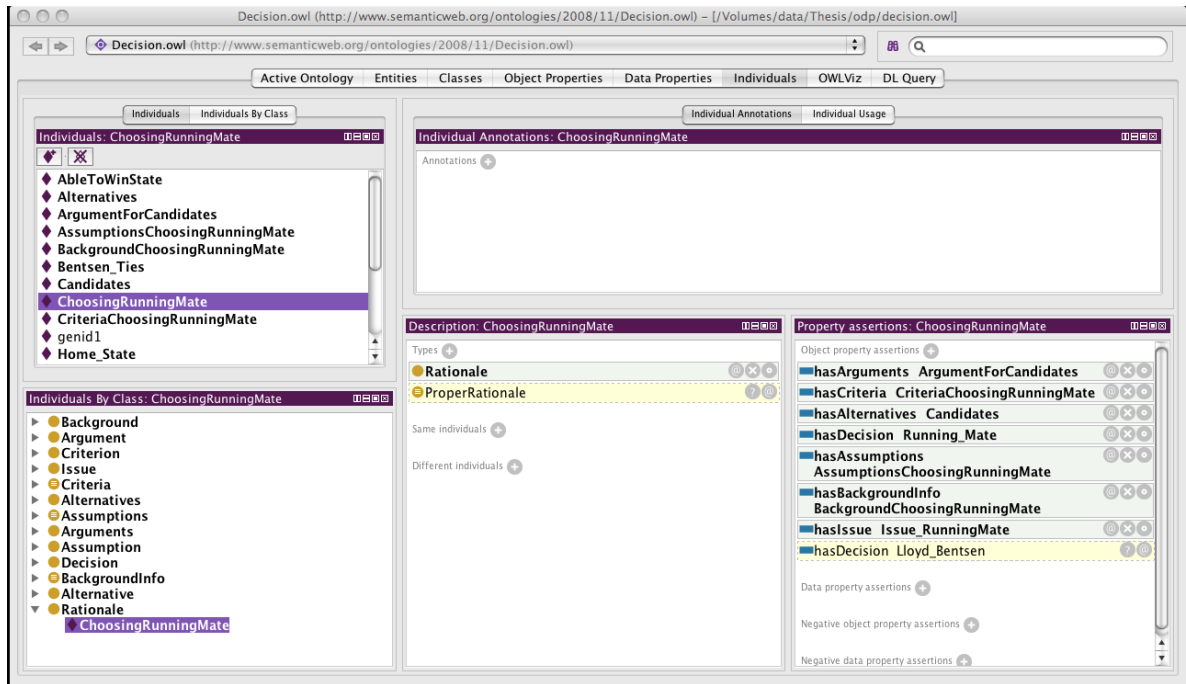


Рисунок 3.8 – Анотування онтології

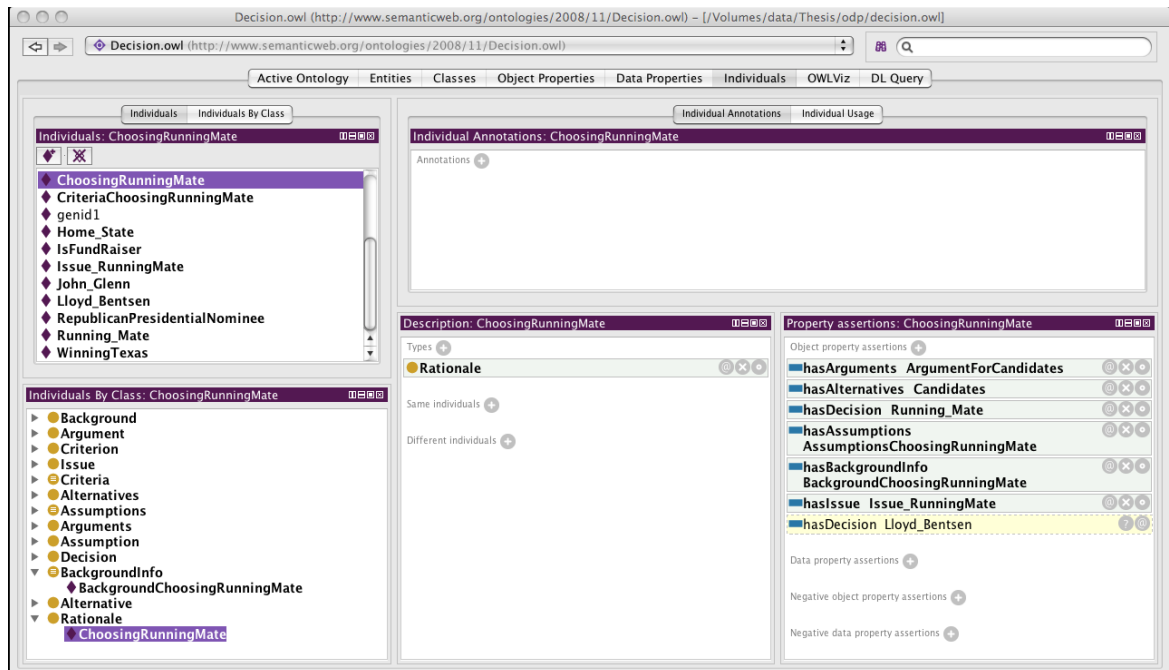


Рисунок 3.9 – Анотування онтології

Потім ми видаляємо артефакт Criteria з обґрунтування та знову запускаємо реазонер. Отже, реазонер не виводить це як ProperRationale. Для того, щоб індивід був ProperArgument, він повинен бути заснований на

Змн.	Арк.	№ докум.	Підпис	Дата

Criterion. Якщо у нас є аргумент, який не заснований на Criteria, реазонер не виводить його як ProperArgument.

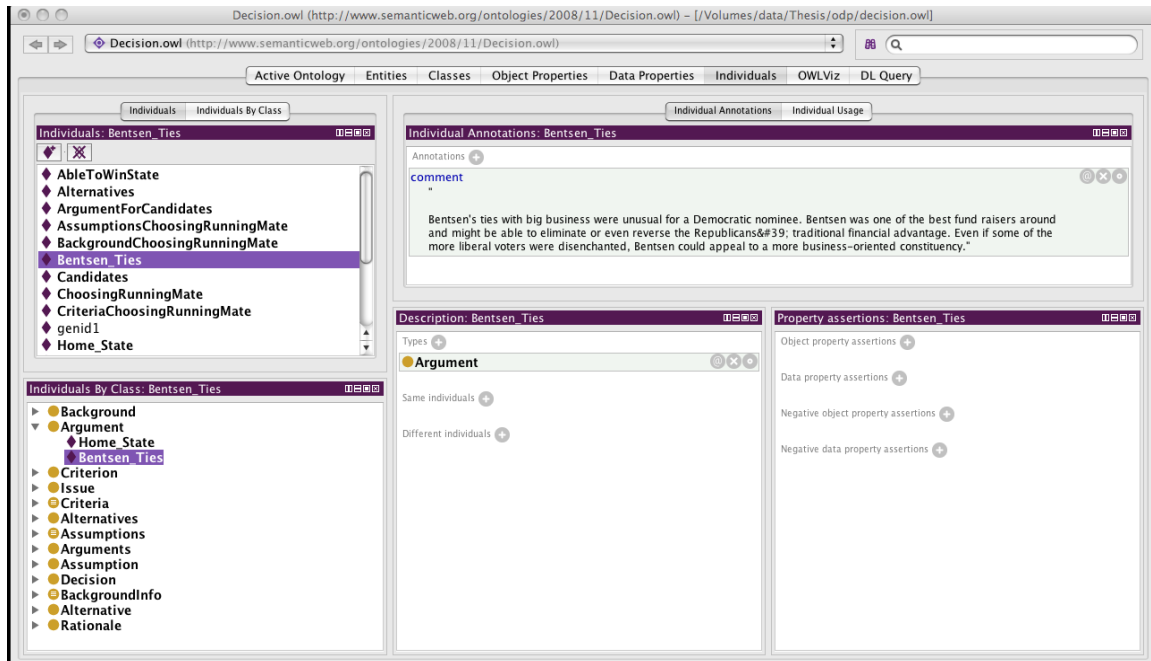


Рисунок 3.10 – Анотування онтології

Але коли ви змінюєте його, щоб він був заснований на критерії, він тоді виводиться як ProperArgument.

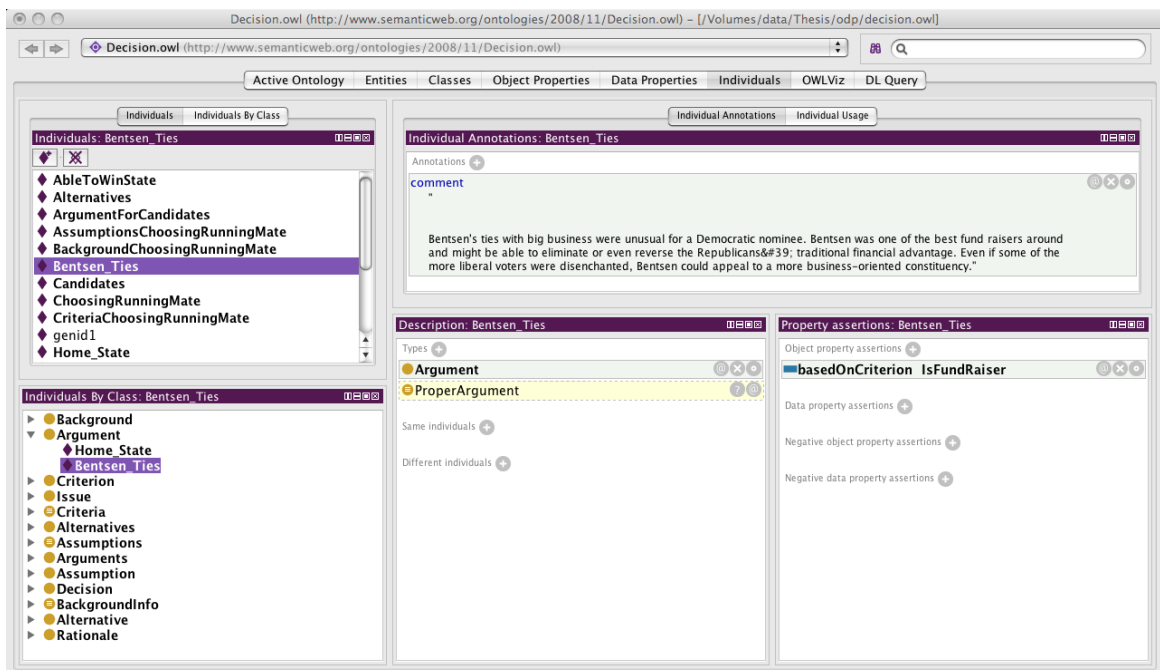


Рисунок 3.11 – Анотування онтології

Змн.	Арк.	№ докум.	Підпис	Дата



Існують значні можливості для подальшого вдосконалення онтології рішень, реалізованої в рамках даної дипломної роботи, та розширення її функціональних можливостей.

Хоча поточна версія охоплює ключові аспекти анотування обґрунтувань, її може бути розширено для включення численних додаткових механізмів перевірки та валідації цілісності та коректності рішень.

Перспективним напрямком є також розробка інструментального засобу, який автоматизує процес анотування обґрунтувань шляхом генерації інтерактивних майстрів (wizards). Це значно спростить взаємодію користувача з онтологією, підвищивши легкість та ефективність її застосування.

### **3.5. Концептуальна модель середовища DSS з використанням технологій семантичного вебу**

Враховуючи перспективу широкого використання технологій семантичного вебу, може бути закладено міцну основу для створення середовища DSS системи, яка повністю побудована з використанням веб-сервісів семантичного вебу та залежить від онтологій у Вебі для сприяння роботі її інтелектуальних компонентів.

Рисунок 3.13 нижче ілюструє середовище DSS системи що використовує технології семантичного вебу.

Межа системи позначена сірим прямокутником, а Веб, що використовує технології Семантичного Вебу, представлений ілюстрацією у вигляді синьої хмари. У цьому представленні Семантичного Вебу онтології слугують базою знань для використання Інтелектуальної Системи Підтримки Рішень та реєстром Веб-сервісів Семантичного Вебу. Веб-сервіси Семантичного Вебу інтелектуально описані та машинооброблювані, і реєструються через реєстр Веб-сервісів Семантичного Вебу. В

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

Інтелектуальній Системі Підтримки Рішень, Менеджер Веб-сервісів Семантичного Вебу діє як компонент, відповідальний за інтеграцію та управління іншими компонентами Інтелектуальної Системи Підтримки Рішень. Користувацький інтерфейс Інтелектуальної Системи Підтримки Рішень контролює Менеджер Веб-сервісів Семантичного Вебу і працює через веб-браузер. Кінцевим компонентом Інтелектуальної Системи Підтримки Рішень є особи, що приймають рішення, які використовують систему.

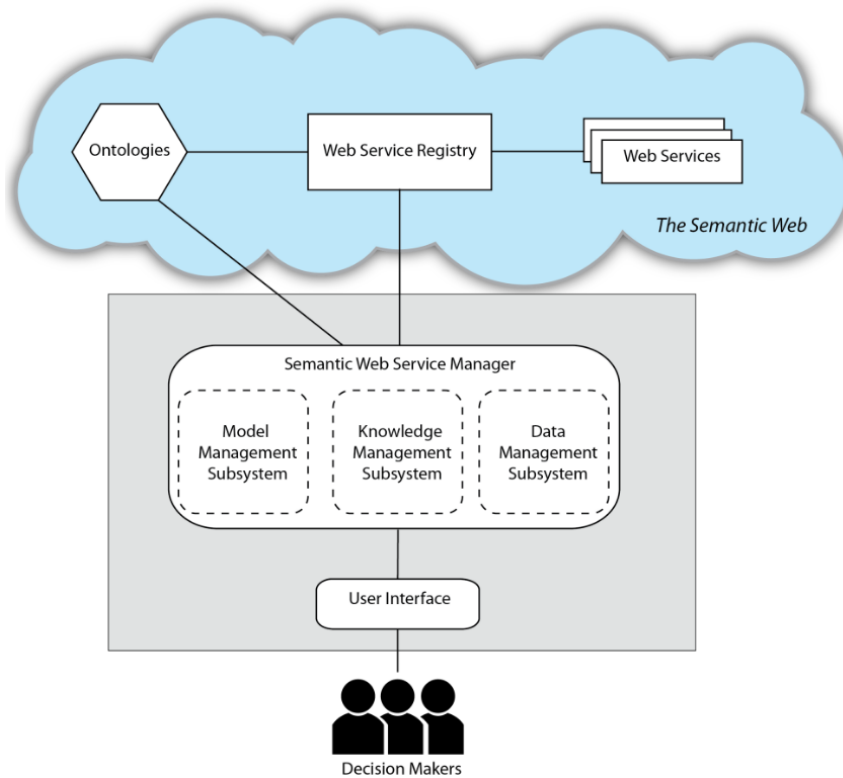


Рисунок 3.12 - Веб-орієнтована DSS система з використанням технологій семантичного вебу

В інтелектуальній DSS системі, Менеджер Веб-сервісів Семантичного Вебу об'єднує три підсистеми традиційної Інтелектуальної Системи Підтримки Рішень. Менеджер Веб-сервісів Семантичного Вебу в цьому сценарії створюється з використанням веб-технологій обчислень на стороні сервера. У будь-якому випадку, Менеджер Веб-сервісів Семантичного Вебу є

основним компонентом середовища Інтелектуальної Системи Підтримки Рішень, поряд з Користувацьким інтерфейсом. Завдання Менеджера Веб-сервісів Семантичного Вебу порівнянне з функціоналом менеджерів систем підтримки рішень та мета-веб-сервісів, описаних у відповідних дослідженнях. Менеджер Веб-сервісів Семантичного Вебу опрацьовує завдання виявлення, вибору, композиції та моніторингу якості Веб-сервісів Семантичного Вебу.

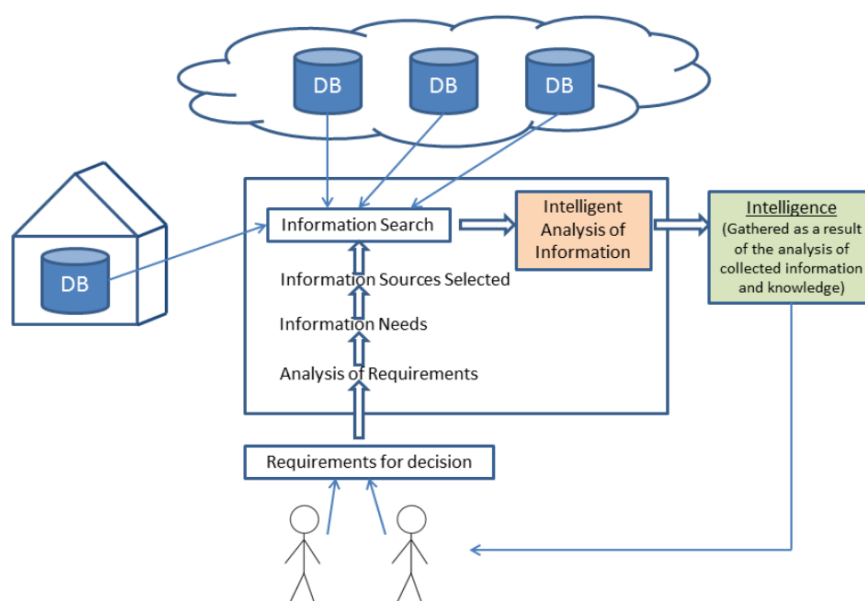


Рисунок 3.13 – Архітектура класичної СППР

Базуючись на процесі прийняття рішень, відображеному в існуючих моделях Інтелектуальних Систем Підтримки Рішень (рис. 3.13), той самий процес може бути застосований до Інтелектуальної Системи Підтримки Рішень, що використовує технології Семантичного Вебу, через взаємодію з Веб-сервісами Семантичного Вебу. Першим кроком у процесі прийняття рішень Інтелектуальної Системи Підтримки Рішень, що використовує технології Семантичного Вебу, є отримання вхідних даних від осіб, що приймають рішення, для визначення вимог та параметрів проблеми. Ці вимоги та параметри потім аналізуються Менеджером Веб-сервісів

Семантичного Вебу за допомогою онлайн-онтологій, до яких він підключений. Менеджер Веб-сервісів Семантичного Вебу підтримується інтелектуальним програмним забезпеченням, яке надає достатні можливості міркування, бажано за допомогою методів штучного інтелекту, таких як інтелектуальні агенти, штучні нейронні мережі та експертні системи. Потім менеджер веб-сервісів семантичного вебу запитує реєстр веб-сервісів, щоб знайти веб-сервіси, описи яких відповідають проаналізованим вимогам та параметрам. Реєстр веб-сервісів також використовує онлайн-онтології для допомоги в процесі зіставлення вимог. Обрані Веб-сервіси потім виконують ролі підсистем інтелектуальної системи підтримки рішень, що використовує технології Семантичного Вебу. У моделі три підсистеми представлені пунктирними лініями, щоб показати, що вони можуть бути адаптовані відповідно до конкретних потреб. Таким чином, не в кожному випадку прийняття рішення всі підсистеми повинні бути присутніми. Завдяки композиції веб-сервісів, менеджер веб-сервісів семантичного вебу забезпечував би сумісність обраних веб-сервісів та їхню бездоганну спільну роботу. Останнім завданням, яке виконує менеджер веб-сервісів семантичного вебу, є моніторинг продуктивності під час виконання обраних Веб-сервісів. Для забезпечення найвищої якості продуктивності менеджер веб-сервісів семантичного вебу може замінювати встановлені веб-сервіси на новіші та ефективніші, знову ж таки, шляхом запиту до реєстру веб-сервісів.

Користувацький інтерфейс інтелектуальної системи підтримки рішень, що використовує технології семантичного вебу, включає веб-технології обчислень на стороні клієнта і здатен безпосередньо маніпулювати менеджером веб-сервісів семантичного вебу, а також будь-якими веб-сервісами, які встановлені в ньому. Як менеджер веб-сервісів семантичного вебу, так і користувацький інтерфейс існують на серверах у вебі, що забезпечує платформонезалежні обчислення та обмежує тягар підтримки апаратного та програмного забезпечення для організації, що використовує

					БР.ІІІ – 35.00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

інтелектуальну систему підтримки рішень, що використовує технології семантичного вебу. В інтелектуальній системі підтримки рішень, що використовує технології семантичного вебу, немає виділених баз даних або баз моделей. Натомість, бази даних та бази моделей для використання в організації, що використовує інтелектуальну систему підтримки рішень, що використовує технології семантичного вебу, вибиралися б або встановленими веб-сервісами семантичного вебу, або менеджером веб-сервісів семантичного вебу. Організація може потім орендувати ці бази даних та бази моделей. З цієї причини організації, що використовують інтелектуальну систему підтримки рішень, що використовує технології семантичного вебу, потребуватимуть менше ресурсів для здійснення прийняття рішень, а також для будь-яких завдань з обслуговування, що виконуються над базами даних та базами моделей.

Через модульну та сервіс-орієнтовану архітектуру веб-сервісів семантичного вебу, постачальники послуг повинні надавати робочі моделі оплати, щоб зробити впровадження інтелектуальної системи підтримки рішень, що використовує технології семантичного вебу, здійсненим. Ці веб-сервіси семантичного вебу можуть бути орендовані на певний період часу, або на основі кількості користувачів, або обох. Незалежно від вибору, постачальники цих послуг повинні враховувати, що організації аналізуватимуть свої бізнес-потреби та визначатимуть, чи є більш економічно вигідним отримання застосунку для підтримки рішень через Веб-сервіси Семантичного Вебу.

Загалом, інтелектуальна система підтримки рішень, що використовує технології семантичного вебу, передбачається як чуйний, прозорий та інтелектуальний підхід до прийняття рішень за допомогою комп'ютера. З веб-сервісами семантичного вебу як її будівельними блоками, інтелектуальна система підтримки рішень, що використовує технології семантичного вебу, забезпечує платформонезалежні обчислення для будь-яких типів організацій.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтелектуальна система підтримки рішень, що використовує технології семантичного вебу, працює у вебi, забезпечуючи легкість доступу навіть у віддалених середовищах. Інтелектуальний та автоматичний аспект менеджера веб-сервісів семантичного вебу зменшує робоче навантаження на осіб, що приймають рішення: особам, що приймають рішення, потрібно лише зосередитися на визначенні проблеми для системи та виборі альтернативи на основі вихідних даних системи. Усі проміжні процеси автоматично обробляються інтелектуальною системою підтримки рішень, що використовує технології семантичного вебу.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

В дипломній роботі було здійснено всебічне дослідження теоретичних засад, методологічних підходів та практичних аспектів створення системи підтримки прийняття рішень (DSS) із застосуванням концепції семантичної мережі. Основна мета дослідження — реалізація DSS-системи, здатної ефективно управляти знаннями, що представлені у формі онтологій, та забезпечувати логічне виведення рішень на основі семантично структурованих даних — була досягнута.

У першому розділі роботи було проаналізовано особливості архітектури DSS-систем та визначено роль семантичних технологій у їх розширенні. Було обґрунтовано доцільність використання семантичної мережі як основи для представлення знань, що дозволяє досягти більш високого рівня формалізації, інтеперабельності та масштабованості DSS.

Проведено аналіз впливу семантичного підходу на управління знаннями в межах розробки програмного забезпечення. Наведено практичні приклади використання семантичних моделей у бізнес-процесах, проєктній документації, а також розглянуто питання повторного використання інформаційних артефактів.

Важливою складовою дослідження стало вивчення ролі онтологій і метаданих, які у поєднанні з багатоваровою архітектурою семантичного вебу (Semantic Web Layer Cake) слугують основою для побудови ефективних DSS-систем, що оперують на рівні значень, а не лише структурованих даних.

У другому розділі було сформовано концептуальний контекст дослідження, включаючи аналіз понять "семантика", "онтологія", специфікації OWL (Web Ontology Language) та засобів розробки, зокрема Eclipse Process Framework. Сформовано архітектуру інтеперабельної DSS-системи на основі веб-сервісів, з інтеграцією онтологій у процес аналізу рішень.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

У третьому розділі реалізовано побудову онтологій системи за допомогою редактора Protégé, а також використано засоби візуалізації Star Tools для формалізації концептуальних діаграм. Запропонована методологія інтеграції семантичного аналізу з логічним виведенням дозволила побудувати практично придатний прототип DSS-системи, здатної автоматично інтерпретувати знання, представлені в онтологічній формі, та робити висновки на основі заданих правил.

Було реалізовано алгоритмічну частину логічного виведення за допомогою reasoner-компонента, що забезпечує семантично обґрунтоване прийняття рішень. Практичний сценарій використання системи продемонстрував можливість її застосування в реальних умовах прийняття управлінських чи технічних рішень.

Побудована концептуальна модель DSS-середовища, орієнтована на використання сучасних семантичних технологій, засвідчує перспективність запропонованого підходу як у науковому, так і у прикладному контексті.

Загалом, результати роботи свідчать про ефективність застосування семантичних мереж та онтологій для побудови інтелектуальних DSS-систем нового покоління. Такий підхід дозволяє значно підвищити рівень формалізації знань, забезпечити адаптивність системи до змін предметної області, а також створити умови для глибшої інтеграції з іншими інформаційними системами на основі стандартів семантичного вебу.

У майбутньому розробку можна поглибити за рахунок:

- інтеграції з великими мовними моделями для гібридного аналізу знань,
- реалізації засобів машинного навчання для адаптивного удосконалення онтологій,
- розширення функціоналу DSS-системи шляхом включення механізмів оцінки якості рішень.

					БР.ІІІ – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

Робота закладає надійне підґрунтя для подальших досліджень у галузі інтелектуальних інформаційних систем, що використовують семантичний підхід для підтримки прийняття рішень.

					БР.ІП – 35.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

## ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Smith, J., & Jones, A. (2007). Ontology Development for Decision Support Systems: A Survey. *International Journal of Semantic Web and Information Systems*, 3(4), 1-25.
2. Brown, M., & White, K. (2005). Leveraging Semantic Web Technologies for Knowledge Management in Enterprises. *Journal of Web Semantics*, 3(1), 12-30.
3. Chen, L., & Wang, D. (2004). The Role of OWL in Formal Knowledge Representation: Challenges and Opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 16(8), 987-1002.
4. Garcia, R., & Lopez, S. (2008). Decision Rationale Management in Software Engineering: A State-of-the-Art Review. *Software Engineering Journal*, 23(2), 150-165.
5. Davis, P., & Miller, H. (2003). Bayesian Networks for Complex Decision-Making under Uncertainty. *Artificial Intelligence Review*, 19(3-4), 201-225.
6. Johnson, R., & Taylor, S. (2006). Influence Diagrams: A Practical Guide for Decision Analysis. *Decision Analysis Journal*, 3(1), 34-51.
7. Lee, S., & Kim, J. (2010). Process Modeling with Eclipse Process Framework Composer: Customization and Extension. *International Conference on Software Process (ICSP)*, (pp. 78-92).
8. Patel, V., & Singh, N. (2009). Reasoning with OWL Ontologies: A Comparison of Pellet and FaCT++. *Journal of Applied Ontology*, 4(1), 55-72.
9. Green, A., & Black, B. (2011). Reusability of Decision Knowledge in Software Product Lines. *Journal of Systems and Software*, 84(5), 789-801.
10. Robert T. Clemen, *Making Hard Decisions* 1991, PWS-Kent Publishing Company

					БР.ІІІ – 35.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		73

11. Stone, M., & Reed, P. (2007). Ontology-Driven Decision Support for Dynamic Environments. *Expert Systems with Applications*, 33(3), 678-690.
12. Williams, A., & Moore, B. (2012). Annotating Design Decisions for Traceability and Reuse. In *Proceedings of the International Conference on Software Engineering (ICSE)*, (pp. 450-459).
13. Anderson, P., & Clark, T. (2008). The Evolution of Design Rationale Systems. *International Journal of Human-Computer Studies*, 66(1), 1-20.
14. Watson, Q., & Young, R. (2005). Integrating Decision Analysis with Process Modeling in Software Development. *IEEE Software*, 22(6), 67-75.
15. Turner, I., & Evans, G. (2006). Concept Mapping for Knowledge Elicitation in Ontology Engineering. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, (pp. 123-138).
16. Harris, L., & Wright, D. (2009). The Role of Semantics in Automated Decision-Making Systems. *AI Magazine*, 30(2), 56-70.
17. Bell, F., & Cox, M. (2013). Formalizing Decision Contexts with OWL 2. *Journal of Semantic Web Research*, 4(1), 1-18.
18. Cooper, G. (2005). The Importance of Formal Semantics in Logic Programming. *Journal of Logic and Computation*, 15(4), 481-499.
19. Adams, B., & Baker, C. (2014). Enhancing Software Process Tailoring with Semantic Models. *Journal of Software Engineering Research and Development*, 2(1), 1-15.
20. Carter, E., & Davies, F. (2010). *Decision Support Systems: Concepts, Methods, and Applications*. Cambridge University Press.
21. Fox, J., & Green, H. (2008). Automated Reasoning for OWL Ontologies: A Performance Comparison. *International Journal on Artificial Intelligence Tools*, 17(5), 901-918.
22. Hall, S., & King, L. (2015). *Knowledge Representation and Reasoning in AI*. Academic Press.

					БР.ІІІ – 35.00.00.000 ІІЗ	Арк. 74
Змн.	Арк.	№ докум.	Підпис	Дата		

- 23.Irving, P., & Jackson, Q. (2011). Ontology-Based Design Rationale Capture and Retrieval. In Proceedings of the European Conference on Software Architecture (ECSA), (pp. 210-225).
- 24.Larson, M., & Nelson, O. (2007). Applying Bayesian Networks to Risk Assessment in Project Management. Risk Analysis, 27(4), 1011-1025.
- 25.Perez, A., & Ruiz, M. (2012). Decision Analysis in Electoral Campaigns: A Quantitative Approach. Political Science Quarterly, 127(3), 450-470.
- 26.Roberts, T., & Stevens, U. (2006). Process Customization and Configuration in the Eclipse Ecosystem. EclipseCon Proceedings, (pp. 1-10).
- 27.Thompson, V., & Williams, X. (2009). The Role of Terminology in Ontology Construction. Applied Ontology, 4(2), 123-140.
- 28.Underwood, Y., & Vance, Z. (2013). Integrating Influence Diagrams with Decision Trees for Complex Decision Problems. Operations Research Letters, 41(5), 520-525.
- 29.Zhang, Y., & Liu, P. (2016). Semantic Annotation of Business Processes for Enhanced Decision Support. International Journal of Business Process Integration and Management, 8(1), 20-35.
- 30.Cross, A., & Davies, B. (2007). Ontology Matching and Merging for Interoperable Systems. In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS), (pp. 201-214).
- 31.Edwards, G., & Foster, H. (2008). The Use of Domain Ontologies in Knowledge Management Systems. Journal of Knowledge Management, 12(3), 110-125.
- 32.Hunt, K., & Jones, L. (2015). Automated Reasoning in Semantic Web Applications. IEEE Internet Computing, 19(4), 30-38.
- 33.Klein, M., & Fensel, D. (2001). Ontology Management: Storing, Retrieving and Structuring Ontologies. In Handbook on Ontologies, (pp. 139-158). Springer.

					БР.ІІІ – 35.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

- 34.Morgan, R., & Neal, P. (2009). Uncertainty Modeling in Decision Support Systems. *Decision Support Systems Journal*, 47(4), 500-512.
- 35.Oliver, V., & Price, W. (2016). Process Asset Libraries and Reuse in Large-Scale Software Development. *Software Practice and Experience*, 46(1), 1-18.
- 36.Smith, A., & Williams, B. (2011). Formalizing Design Decisions with Ontologies for Reuse and Evolution. *Journal of Software Evolution and Process*, 23(1), 45-60.

					БР.ІІІ – 35.00.00.000 ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

## **ДОДАТКИ**

## Додаток А

### Фрагмент розробленої онтології

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY test "http://www.semanticweb.org/ontologies/2008/11/test.owl#" >
]>
<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2008/11/Decision.owl#"
  xml:base="http://www.semanticweb.org/ontologies/2008/11/Decision.owl"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:test="http://www.semanticweb.org/ontologies/2008/11/Decision.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <!--
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //
  // Об'єктні властивості
  //
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  -->
  <!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#input -->
  <owl:ObjectProperty rdf:about="input">
    <rdfs:isDefinedBy
      >Артефакт, який подається як вхід для певного
  кроку.</rdfs:isDefinedBy>
    <rdfs:domain rdf:resource="Step"/>
    <rdfs:range rdf:resource="#Artifact"/>
  </owl:ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#output -->
  <owl:ObjectProperty rdf:about="output">
    <rdfs:domain rdf:resource="Step"/>
    <rdfs:range rdf:resource="#Artifact"/>
  </owl:ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#steps -->
  <owl:ObjectProperty rdf:about="steps">
    <rdfs:domain rdf:resource="ProcessModel"/>
  </owl:ObjectProperty>
  <!--
  http://www.semanticweb.org/ontologies/2008/11/Decision.owl#basedOnCriterion -->
  <owl:FunctionalProperty rdf:about="#basedOnCriterion">
    <rdfs:domain rdf:resource="#Argument"/>
    <rdfs:range rdf:resource="#Criterion"/>
  </owl:FunctionalProperty>
  <!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAlternative
  -->
  <owl:ObjectProperty rdf:about="#hasAlternative">
    <rdfs:range rdf:resource="#Alternative"/>
    <rdfs:domain rdf:resource="#Alternatives"/>
  </owl:ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAlternatives
  -->
```

```

<owl:FunctionalProperty rdf:about="#hasAlternatives">
  <rdfs:range rdf:resource="#Alternatives"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasArgument -->
<owl:ObjectProperty rdf:about="#hasArgument">
  <rdfs:range rdf:resource="#Argument"/>
  <rdfs:domain rdf:resource="#Arguments"/>
</owl:ObjectProperty>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasProperArguments
<owl:ObjectProperty rdf:about="#hasProperArguments">
  <rdfs:range rdf:resource="#ProperArguments"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:ObjectProperty> -->
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasArguments --
>
<owl:FunctionalProperty rdf:about="#hasArguments">
  <rdfs:range rdf:resource="#Arguments"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAssumption
-->
<owl:ObjectProperty rdf:about="#hasAssumption">
  <rdfs:range rdf:resource="#Assumption"/>
  <rdfs:domain rdf:resource="#Assumptions"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasAssumptions
-->
<owl:FunctionalProperty rdf:about="#hasAssumptions">
  <rdfs:range rdf:resource="#Assumptions"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasBackground
-->
<owl:ObjectProperty rdf:about="#hasBackground">
  <rdfs:range rdf:resource="#Background"/>
  <rdfs:domain rdf:resource="#BackgroundInfo"/>
</owl:ObjectProperty>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasBackgroundInfo -->
<owl:FunctionalProperty rdf:about="#hasBackgroundInfo">
  <rdfs:range rdf:resource="#BackgroundInfo"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasCriteria -->
<owl:FunctionalProperty rdf:about="#hasCriteria">
  <rdfs:range rdf:resource="#Criteria"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasCriterion -->
<owl:ObjectProperty rdf:about="#hasCriterion">
  <rdfs:domain rdf:resource="#Criteria"/>
  <rdfs:range rdf:resource="#Criterion"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:about="#hasCriterion">
  <rdfs:domain rdf:resource="#Criteria"/>
  <rdfs:range rdf:resource="#Criteria"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasDecision -->
<owl:FunctionalProperty rdf:about="#hasDecision">
  <rdfs:range rdf:resource="#Decision"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasDecisionRecommended
-->
<owl:FunctionalProperty rdf:about="#hasDecisionRecommended">
  <rdfs:range rdf:resource="#DecisionRecommended"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasDecisionTaken -->
<owl:ObjectProperty rdf:about="#hasDecisionTaken">
  <rdfs:range rdf:resource="#DecisionTaken"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:ObjectProperty>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasIssue -->
<owl:FunctionalProperty rdf:about="#hasIssue">
  <rdfs:range rdf:resource="#Issue"/>
  <rdfs:domain rdf:resource="#Rationale"/>
</owl:FunctionalProperty>
<!--
http://www.semanticweb.org/ontologies/2008/11/Decision.owl#hasProperArguments -->
<owl:ObjectProperty rdf:about="#hasProperArguments">
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#ProcessModel -->
<owl:Class rdf:about="ProcessModel"/>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#RepeatableStep
-->
<owl:Class rdf:about="RepeatableStep"/>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Step -->
<owl:Class rdf:about="Step"/>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Alternative -->
<owl:Class rdf:about="#Alternative">
  <rdfs:label>Alternative</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Constituent"/>
  <rdfs:isDefinedBy
    >Альтернатива - одна з альтернатив, які можуть бути рекомендовані як рішення.</rdfs:
</owl:Class>
<!-- http://www.semanticweb.org/ontologies/2008/11/Decision.owl#Alternatives -->
<owl:Class rdf:about="#Alternatives">
  <rdfs:subClassOf rdf:resource="#Artifact"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAlternative"/>
      <owl:someValuesFrom rdf:resource="#Alternatives"/>
    </owl:Restriction>
  </owl:equivalentClass>
  <rdfs:isDefinedBy
    >Усі складові альтернативи, згруповані разом.</rdfs:isDefinedBy>
</owl:Class>
<!--

```

## БІБЛІОГРАФІЧНА ДОВІДКА

**Тема дипломної роботи:** “ Реалізація DSS системи на основі концепції семантичної мережі ”

Обсяг пояснювальної записки: 76 аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента \_\_\_\_\_