

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

«Моделювання прикладних предметних галузей за допомогою онтологій»

(тема)

Виконав: студент 2 курсу, групи СШМ-18-3
Бех А.А.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)

Керівник професор кафедри ШІ Рябова Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

Філатов В.О.

(підпис)

(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
 (повна назва)
 Кафедра _____ Штучного інтелекту _____
 (повна назва)
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 122 Комп'ютерні науки _____
 (код і повна назва)
 Тип програми _____ освітньо-наукова _____
 (освітньо-професійна або освітньо-наукова)
 Освітня програма _____ Системи штучного інтелекту _____
 (повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2020 р.

ЗАВДАННЯ**НА МАГІСТЕРСЬКУ АТЕСТАЦІЙНУ РОБОТУ**

студентові _____ Бех Артему Антоновичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи Моделювання прикладних предметних галузей за допомогою онтологій
затверджена наказом по університету від "30" березня 2020 р. №480Ст
2. Термін здачі студентом закінченої роботи 8 травня 2020 року _____
3. Вихідні дані до роботи специфікація мови OWL, Protégé 5.0.8
4. Зміст пояснювальної записки (перелік питань, що їх належить розробити) _____
Аналіз предметної галузі та постановка задачі, дослідження системи оподаткування України, обґрунтування вибору технології онтологічного моделювання, розробка онтології прикладної предметної галузі
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових рисунків) _____
Рисунок 1 - Структура податкового законодавства України, рисунок 2 - Список класів онтології, рисунок 3 - Приклад розроблених підкласів для однієї сфери, рисунок 4 - Приклад створених властивостей онтології, рисунок 5 - Приклад розроблених властивостей, рисунок 6 - Приклад створених примірників, рисунок 7 - Приклад умов виконання примірника, рисунок 8 - Список створених запитів, рисунок 9 - Загальний вид вкладки Queries Tab, рисунок 10 - Кнопка Retrieve Query на панелі управління, рисунок 11 - Список критеріїв і полів на прикладі одного із запитів, рисунок 12 - Додаткове вікно Select Instance, рисунок 13 - Вибір параметра для критерію, рисунок 14 - Приклад заповненого поля для визначення розміру податку, рисунок 15 - Висновок

результату _____

6. Консультанти з роботи із зазначенням розділів проекту, що їх стосуються

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		(підпис)	(дата)
Основний розділ	проф. Рябова Н.В.		

7. Дата видачі завдання _____

Керівник роботи _____ проф. кафедри ІІІ Рябова Н.В.
(підпис) (посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання _____
(підпис студента)

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів магістерської атестаційної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання		Виконано
2	Аналіз предметної галузі та постановка задачі		Виконано
3	Теоретичні дослідження з предметної галузі		Виконано
4	Розробка онтології		Виконано
5	Розробка запитів до онтології		Виконано
6	Підготовка пояснювальної записки		Виконано
7	Попередній захист атестаційної роботи		Виконано
8	Захист атестаційної роботи		

Студент _____
(підпис)

Керівник роботи _____ (підпис)

РЕФЕРАТ

Записка пояснювальна: 75 с., 19 рисунків, 3 таблиць, 14 джерел

ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ, ОНТОЛОГІЯ, СИСТЕМА ОПОДАТКУВАННЯ УКРАЇНИ, ПОДАТКОВИЙ КОДЕКС УКРАЇНИ, ЗАПИТ, РОЗРОБКА

Об'єктом дослідження є система оподаткування України.

Предметом дослідження є онтологічне моделювання.

Метою атестаційної роботи є розробка онтології прикладної предметної галузі на прикладі системи оподаткування України.

Методи дослідження – системний аналіз, об'єктно-орієнтований підхід, онтологічні моделі та методи управління організацією.

Розроблено онтологію, яка має можливість обробляти запити користувача, для знаходження розміру податку в різних сферах діяльності, використовуючи дані, які вводить користувач.

ABSTRACT

Thesis: 75 pages, 19 pictures, 3 tables, 14 sources.

ONTOLOGICAL MODELING, ONTOLOGY, TAXATION UKRAINE,
UKRAINE TAX CODE, INVITE, DEVELOPMENT

The object of study is the tax system of Ukraine.

The subject of research is ontological modeling.

The aim of the thesis is the development of applied ontology of subject areas, for example, the tax system of Ukraine.

Research methods - system analysis, object-oriented approach, ontological models and organization management methods.

Ontology was developed that has the ability to handle user inquiries, to find the amount of tax in different areas, using data that you enter.

ЗМІСТ

<u>ВСТУП</u>	8
<u>1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ</u>	10
<u>1.1 Онтологічне моделювання</u>	10
<u>1.1.1 Проблеми термінології онтологічного моделювання</u>	18
<u>1.2 Процес розробки онтології</u>	19
<u>1.3 Постановка задачі</u>	20
<u>2. ВИВЧЕННЯ СИСТЕМИ ОПОДАТКУВАННЯ УКРАЇНИ. ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ</u>	21
<u>2.1 Податкове законодавство України</u>	21
<u>2.2 Податковий кодекс України</u>	23
<u>2.2.1 Структура податкового Кодексу України</u>	24
<u>2.3 Онтологічний підхід</u>	25
<u>2.3.1 Переваги онтології</u>	30
<u>2.4 Мови уявлення онтологій</u>	31
<u>2.4.1 Мова опису Owl</u>	33
<u>2.5 Структура Owl онтології</u>	35
<u>2.5.1 Класи</u>	35
<u>2.5.2 Властивості</u>	36
<u>2.5.3 Індивіди</u>	37
<u>3. СТВОРЕННЯ ОНТОЛОГІЇ ПРИКЛАДНОЇ ПРЕДМЕТНОЇ ГАЛУЗІ</u>	39
<u>3.1 Інструменти побудови онтологій</u>	39
<u>3.2 Інструменти для відображення і об'єднання онтологій</u>	44
<u>3.3 Інструменти анотування на основі онтологій</u>	47
<u>3.4 Порівняльний аналіз інструментів</u>	48
<u>3.5 Редактор онтології</u>	51

<u>3.6 Основні завдання розробки онтології</u>	51
<u>3.7 Визначення критеріїв відповідних розміром податку</u>	52
<u>3.8 Розробка онтології податкового кодексу України</u>	54
<u>3.8.1 Створення властивостей онтології</u>	56
<u>3.8.2 Створення примірників онтології</u>	57
<u>3.9 Створення запитів</u>	60
<u>ВИСНОВКИ</u>	68
<u>ПЕРЕЛІК ПОСИЛАНЬ</u>	69
<u>Додаток А</u>	71

ВСТУП

Онтологія – це формальний опис результатів концептуального моделювання предметної галузі, представлена у формі, яка сприймається людиною і комп'ютерною системою [1].

Онтологія складається з примітивів представлення знань предметної галузі (визначень основних понять, таких як імена індивідуумів, класи, функції та інші сутності), а також різного роду семантичних зв'язків, підтримуваних між ними.

Онтологію предметної галузі можна розглядати як словник термінів, специфічних для даної предметної галузі, разом з сукупністю аксіом, які забезпечують інтерпретацію і правильне використання цих термінів. Онтологічне уявлення знань використовується для семантичної інтеграції інформаційних ресурсів, адекватної інтерпретації змісту текстових документів і пошукових запитів, поданих на природній мові.

У даній роботі онтологія предметної галузі розробляється з використанням стандарту мови опису онтології - OWL.

Тема використання онтологічного підходу в сучасному світі є досить актуальною, так як даний підхід дозволяє використовувати ряд переваг побудови даної системи.

Онтологічний підхід дозволяє представити природно-мовний текст в такому вигляді, що він стає придатним для автоматизованої обробки. Потреба в онтологіях пов'язана з неякісною автоматичною обробкою природно-мовних текстів існуючими засобами [2]. Тому необхідно мати детальний опис проблемної галузі з безліччю логічних зв'язків, які показують відносини між термінами. Накопичення цінної інформації про функціонування складних систем. Такий аналіз починається зі складання словника термінів, а також створення системи точних визначень цих термінів [3]. Документуються основні логічні взаємозв'язки між

відповідними термінами і поняттями. Результатом цього аналізу є словник термінів і їх точні визначення, взаємозв'язки між ними. Формування цілісного погляду на предметну галузь, можливість відновити відсутні логічні зв'язки у всій їхній повноті. Матеріал, представлений в єдиній формі, набагато краще сприймається і відтворюється.

Онтологічний інжиніринг – ядро концепції “керування знаннями”, яке виникло в середині 90-х років минулого століття в великих корпораціях, де проблеми оброблення інформації набули особливої гостроти і стали критичними. Стало очевидним, що основним є оброблення знань, накопичених фахівцями компанії, оскільки саме знання забезпечують перевагу перед конкурентами.

Фактично менеджмент знань – це модне гасло в управлінні, яке займається проблемою організації інформації в компанії, адже часто інформації в компаніях накопичено навіть більше, ніж вони здатні оперативно оброблювати. Різні організації намагаються вирішувати це питання по-своєму, але кожна компанія прагне збільшити ефективність оброблення інформації. [5]

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Онтологічне моделювання

Масштабність застосування інформаційних технологій істотно змінила не тільки структура процесів, але і правила ведення бізнесу. Чим вище динаміка і складність бізнесу, тим більш розвиненими повинні бути і інформаційні системи.

Основна мета методу онтологічного моделювання даних полягає в тому, щоб забезпечити структурну методичку, підтримувану автоматизованими інструментальними засобами, які фахівець з проблемної галузі може ефективно розробляти і, тим самим, підтримувати корисні і точні онтології [4].

Аналіз предметної галузі становить особливий вид наукової діяльності, в результаті якої будується інтерпретаційна модель предметних знань. Деякі ідеї по розробці методології проектування онтології беруть свій початок в літературі з об'єктно-орієнтованого підходу (ООП), який з'явився як технологія програмування великих програмних продуктів.

Онтологічний підхід до моделювання предметної галузі має такі основні переваги:

а) дозволяє створювати системи, в яких знання, накопичені всередині організації, стають доступними для більшості користувачів;

б) представляє користувачеві цілісний, системний, однаковий погляд на предметну галузь дозволяє відновити відсутні логічні зв'язки предметної галузі у створеній моделі дає можливість генерувати нові знання.

Традиційно джерелами знань, які застосовувались в ранніх експертних системах, було їх вилучення знань за допомогою експерта. При цьому застосовувалися як пасивні методи (спостереження, аналіз протоколів роботи, лекційні матеріали), так і активні (анкетування, інтерв'ю, експертні

ігри і т.д.). Однак робота з експертами складна, дорога, трудомістка і вимагає значного часу. У зв'язку з цим найбільший інтерес представляють методи автоматизованого вилучення знань з потоків даних, що відображають реальну роботу фахівців в будь-якій предметній галузі і дозволяють узагальнювати і формалізувати їх досвід.

Специфіка таких потоків даних полягає в наступному:

- а) неоднорідність даних (інформація на природній мові, мультимедійні дані);
- б) великий обсяг даних, постійно зростаючий;
- в) інструментальні засоби аналізу даних, повинні бути доступні користувачеві, яка не є професіоналом в галузі інформаційних технологій;
- г) незалежність методів вилучення знань від розробника і доступність всім користувачам (наприклад, в рамках корпоративної системи).

Одним з напрямків автоматизованого вилучення знань (неявним чином присутніх в оброблюваній інформації), що базується на інтелектуальному аналізі даних, є Data Mining.

Розглянемо більш детально використання онтології для опису предметної галузі.

Поняття онтологія широко використовується в філософії, позначаючи частину метафізики - вчення про усьому суцільному, про його найбільш загальних філософських категоріях, таких як буття, субстанція, причина, дія, явище. При цьому онтологія як наука претендувала на повне пояснення причин всіх явищ. В інформаційних технологіях під онтологією розуміється детальний опис деякої предметної галузі, яке використовується для формального і декларативного визначення її концептуалізації.

Важливою стороною онтології є уявлення понять предметної галузі у вигляді, придатними для машинної обробки.

У центрі більшості онтологій знаходяться класи, які описують поняття предметної галузі.

На формальному рівні онтологія - система, що складається з наборів понять і тверджень про ці поняття, на основі яких можна будувати класи, об'єкти, відносини, функції та теорії.

Практично всі моделі онтології містять базові визначення:

- а) концепти (поняття, класи);
- б) властивості концептів (атрибути, ролі);
- в) відносини між концептами (залежно, функції);
- г) додаткові обмеження, які визначаються аксіомами.

Концептом може бути опис завдання, функції, дії, стратегії, процесу міркування і т. інш.

Онтологічні системи будуються на основі наступних принципів:

- а) формалізації, (опису елементів предметної галузі в єдиних, чітко визначених зразках (термінах, моделях і ін.);
- б) обмеженості кількості базових термінів (сутностей), на основі яких конструюються всі інші поняття;
- в) внутрішньої повноти і логічної несуперечності.

На відміну від звичайного словника для онтологічної системи характерно внутрішню єдність, логічний взаємозв'язок і несуперечливість використовуваних понять.

Побудова онтологій базується на аналізі, в основі якого лежить опис системи (наприклад, підприємства) в термінах сутностей, відносин між ними і перетворення сутностей, яке виконується в процесі вирішення певної задачі.

Основною характерною рисою цього підходу є, зокрема, розділення реального світу на складові і класи об'єктів і визначення їх онтологій, або ж сукупності фундаментальних властивостей, які визначають їх зміни і поведінку. Підходи і методології проектування і реалізації онтологій базуються на наступних принципах:

а) ясність - онтологія повинна ефективно передавати сенс введених термінів, її визначення повинні бути об'єктивні, а для їх об'єктивізації повинен використовуватися чітко фіксований формалізм;

б) узгодженість - все визначення повинні бути логічно несуперечливі, а ті твердження, які виведені в онтології, не повинні суперечити її аксіом;

в) розширюваність - необхідно проектувати онтологію так, щоб її словники термінів можна було розширювати без ревізії вже існуючих понять;

г) мінімум впливу кодування - концептуалізація онтології повинна бути специфікована на рівні уявлення, а не символного кодування;

д) мінімум онтологічних зобов'язань - онтологія повинна містити тільки найбільш істотні припущення про модельованій предметній галузі, щоб залишати свободу розширення і спеціалізації.

Онтологічний інжиніринг заснований на глибокому структурному аналізі предметної галузі, що включає наступні етапи:

а) виділення концептів - базових понять даної предметної галузі;

б) визначення «висоти дерева онтологій» - кількість рівнів абстракції;

в) розподіл концептів за рівнями;

г) побудова зв'язків між концептами - визначення відносин і взаємодій базових понять;

д) консультації з різними фахівцями (експертами) для виключення протиріч і неточностей.

Методологія побудови онтології включає наступні основні етапи:

а) позначення цілей і сфери застосування створюваної онтології;

б) фіксування знань про предметну галузь, тобто. е. визначення основних понять і їх взаємовідносин в обраній предметній галузі; створення точних несуперечливих визначень для кожного основного поняття і відносини; визначення термінів, які пов'язані з цими термінами і відносинами;

- в) кодування, тобто поділ сукупності основних термінів, які використовуються в онтології, на окремі класи понять;
- г) вибір або розробку спеціальної мови для подання онтології;
- д) безпосередньо завдання фіксованою концептуалізації на обраною мовою представлення знань.

Розглянемо теоретичні поняття, у вигляді формальної моделі онтології
 Онтологія складається з термінів (понять), їх визначень і атрибутів, а також пов'язаних з ними аксіом і правил виведення.

Формальна модель онтології $O = \langle T, R, F \rangle$ - це впорядкована трійка множин, де:

- T - терміни прикладної галузі, яку описує онтологія O;
- R - відносини між термінами заданої предметної галузі;
- F - функції інтерпретації, задані на термінах і/або відносинах онтології O.

У світі загалом і наукових дослідженнях зокрема існує багато факторів, що приводять до появи загальних вимог до інтелектуальних систем. До них належать: збільшення масштабу завдань, які ставлять у сфері інформаційних систем; зростання кількості сторін, які створюють і використовують інформацію; збільшення об'єму інформації та частоти її змін; розподіленість інформації. Ця система повинна бути гнучкою, масштабованою і відкритою.

Під відкритістю розуміється побудова інтелектуальної інформаційної системи на основі світових інформаційних стандартів. Масштабованість – це можливість побудови (зокрема поетапного) системи з готових або стандартних компонентів. Це дає змогу легко змінювати масштаби кола задач, розв'язуваних системою, змінювати і поетапно нарощувати систему. Гнучкість означає здатність системи до налаштування під інформаційні потреби користувача, можливість перенесення з одного середовища експлуатації в інше та незалежність від конкретних компонентів і програм, залучених у систему.

Для розроблення інтелектуальних систем, які б задовольняли цим вимогам необхідні стандарти. Їх завдання полягає в тому, щоб надати можливість користувачам і програмам спілкуватися між собою і один з одним, не обмежуючись конкретною предметною галуззю. Сьогодні кілька загальноприйнятих відкритих Інтернет-стандартів становлять базу для побудови таких систем. До стандартів, що задовольняють вищеописані вимоги, належать:

- інтернет-стандарти, розроблені консорціумами IETF, WWW та WS-I (наприклад, XML);
- Веб-служби як метод забезпечення обміну інформацією між різними системами різних розробників;
- єдиний підхід до організації метаданих (Дублінське ядро, стандарт RDF).

Вже розроблені та програмно підтримані десятки стандартів в галузі обміну інформацією. Серед інших можна виділити фундаментальні стандарти: InfoSet, Namespace, XML.

InfoSet – (XML Information set), специфікація цього стандарту визначає набір абстрактних інформаційних елементів, що використовують у правильно побудованих XML-документах.

Namespace – (Namespaces in XML), простір імен. Використовується для усунення неоднозначності в іменах елементів і атрибутів у процесі розподіленого розроблення.

XML – (Extensible markup language), мова, що описує клас об'єктів XML document, а також частково роботу комп'ютерних програм, які опрацьовують об'єкти з даними, що реалізують цей клас. XML стала стандартом відображення структурованих електронних документів для більшості сучасних програмних засобів, зокрема загальноновживаних пакетів Microsoft Office та OpenOffice.org.

Найістотніша заслуга XML – відображення даних однією мовою. Універсальний синтаксис привів до появи низки важливих XML-технологій. Це мови XSL і XPath, призначені для роботи з деревоподібною структурою документів; XML Schema – стандарт опису конкретних мов розмітки з використанням синтаксису XML; це самостійна специфікація, яка підтримує суворішу типізацію даних, надає ширші можливості для визначення типів даних елементів та їх атрибутів; XLink і XPointer – засоби зв'язування розподілених блоків інформації в загальний документ.

Окрім фундаментальних стандартів відображення даних, розглянемо й інші. Необхідне для уніфікації відображення даних та знань сімейство стандартів вже розроблене і широко використовується. Зокрема, формат RDF (Resource Description Framework – інструмент опису ресурсів) є різновидом формату XML. Він призначений для формування електронного інформаційного середовища між джерелами та споживачами інформації, об'єднаними в єдину мережу. Формат RDF – набір інструментів для роботи з метаданими, який забезпечує єдине (стандартизоване) середовище взаємодії програм, які обмінюються інформацією у Web зрозумілою мовою. RDF робить наголос на легкість автоматизованого опрацювання Web-ресурсів. Загалом, RDF забезпечує основу для елементарних інструментів авторизації, пошуку і редагування даних, що сприяє трансформації Web в апаратно-опрацьовуване сховище інформації. У лютому 2004 р. прийнято низку специфікацій до нього, які значно полегшують опис ресурсів. Серед них важливо виділити такі: Concepts and abstract syntax; RDF Semantics; RDF Primer; RDF Vocabulary description language 1.0: RDF schema; RDF/XML Syntax specification (revised); RDF Test cases.[6]

Як стандарт відображення онтологій в Інтернеті консорціумом W3C (<http://www.w3.org>) пропонують мову DAML (DARPA markup language), що є надбудовою над іншими мовами: RDF та RDFS, які, своєю чергою, є розширеннями XML. DAML основана на фреймах, в яких концептуалізація

задається у форматі об'єктів та властивостей. Ряд DAML-орієнтованих (або підтримуваних) інструментів доволі широкий. Серед них можна виділити: редактори онтологій, візуалізатори у вигляді графів, браузері, транслятори на інші мови, машини виведення тощо.

Зараз перспективною є ініціатива консорціуму W3C під назвою Семантичний Web (Semantic Web), метою якого є інтелектуалізація всієї мережі Інтернет. У межах проекту Semantic Web, спрямованого на аналіз семантики інформаційних ресурсів, здійснюється робота групи Web Ontology, оскільки саме онтологічний підхід є основою для подання знань про різні предметні галузі. Нещодавно ця група опублікувала першу версію робочого проекту, в якому викладені основні вимоги до нової мови подання онтологій – OWL (Ontology Web Language).

Закономірно виникає запитання: навіщо ще одна мова опису даних.

Українською мовою аббревіатуру OWL можна перекласти як “мова опису онтологій у Web”. Під онтологією розуміють сукупність термінів і понять, що використовують у певній галузі знань або діяльності (наприклад, машинобудуванні, біології, матеріалознавстві тощо). Онтологія також містить формальні, а отже, зрозумілі для комп'ютерів описи ключових понять і взаємозв'язків між ними. Зараз широко використовується мова XML та деякі інші механізми дають змогу забезпечити гнучкість процесу обміну даними між різними програмами, сервісами, базами даних тощо. Проте, вони ефективні тільки в тих випадках, коли контрагенти обміну використовують для відображення інформації однакові системи змістових координат. Засоби опису й опрацювання онтологій OWL дають змогу “порозумітися” найрізноманітнішим прикладним програмам, основне завдання яких полягає не в поданні інформації для людини, а безпосередньо в опрацюванні вмісту інформаційних ресурсів. Для створення цієї мови використаний і узагальнений досвід попередніх розроблень подібних мов. Прототипом для розроблення OWL є мова опису онтологій DAML+OIL (друга версія мови в

межах проекту DAML), розроблена Агентством перспективних досліджень Міністерства оборони США DARPA. У цій версії мови використані результати проекту OIL (Ontology Inference Layer або Ontology Interchange Language), який виконувався в межах програми Information Society Technologies за підтримки Європейського Союзу. Створена зусиллями консорціуму W3C мова OWL є мовою для визначення структурованих, підтримуваних у Web-онтологіях даних, які забезпечують багатшу інтеграцію, ніж попередні мови, що не були орієнтовані на Web і, зокрема, на Семантичний Web (рис. 7.3).

Стандарт OWL визначає три рівні мови або три підмови, що відрізняються рівнем своїх виразних можливостей: OWL Lite, OWL DL і OWL Full, які детально описані в наступних параграфах.

Важливою якістю всіх описаних стандартів є їх відкритість і незалежність від конкретних сфер застосування і розділів знань.[5]

1.1.1 Проблеми термінології онтологічного моделювання

Предметна галузь онтологічного моделювання розвивається на перетині ряду галузей знання, що мають власну багату історію розвитку і наукових досліджень. Онтологічне моделювання базується на знаннях з галузей філософії, лінгвістики, логіки, концептуального моделювання. Термінологія цих галузей використовується в онтологічному моделюванні. Однак існує також і тенденція застосовувати термінологію, специфічну для онтологічного моделювання, до різних галузей знань. Зокрема, в концептуальному моделюванні баз даних і проектуванні інформаційних систем зміна термінології на онтологічну багато в чому продиктовано зручністю застосування мови уявлення онтологій OWL (Web Ontology Language) [9] в якості мови визначення даних. В галузі подання знань використання термінології онтологічного моделювання пояснюється

застосуванням тієї ж мови, тому що вона заснована на дескриптивних логіках і зручна в завданні класифікації фактів. У лінгвістиці онтологія фігурує, в першу чергу, як словник лінгвістичних термів. В результаті, терміни, історично властиві перерахованим областям, нерідко підміняються термінами з галузі онтологічного моделювання. І назад, з термінами галузі онтологічного моделювання зв'язується семантика, притаманна цим областям. При цьому збільшуються різночитання в семантиці термінів онтологічного моделювання в різних спільнотах. На термінологію в цілому в галузі онтологічного моделювання також дуже впливають приватні термінологічні рішення, прийняті в конкретних засобах моделювання, в тому числі [10-11]

В цілому, термінологію в галузі онтологічного моделювання досі важко назвати усталеною. Зокрема, україномовна термінологія, крім усього іншого, залежить від англійської як міжнародної, і різночитання в ній виникають в результаті різних перекладів термінів з англійської мови на українську.

1.2 Процес розробки онтології

Розробка онтологій як ієрархічної структури понять (концептів) відрізняється від проектування об'єктів як класів і відносин в об'єктно-орієнтованому програмуванні. Останній зосереджується головним чином на методах опису класів - програміст приймає проектні рішення, засновані на операційних властивості класу, тоді як розробник онтології приймає ці рішення, ґрунтуючись на структурних властивостях класу. Дані факти дозволяють сформулювати основні принципи, притаманні онтології [12-13]:

а) дохідливість, ясність. Терміни і поняття онтології повинні відображати реальну дійсність. Їх символічні позначення (знаки) повинні формуватися на основі загальноприйнятих правил загальноприйняті смисли реальних об'єктів. У свою чергу, ці смисли витягуються із загальноприйнятих

визначень термінів (понять), зафіксованих в тлумачних словниках, різних глосаріях;

б) обґрунтованість, зв'язність. Формування початкового набору понять онтології і їх додаток має бути обґрунтованим, визначеним, в першу чергу, вимогами передбачуваної сукупності вирішуваних завдань. Логічні аксіоми початкового набору понять повинні бути повинні суперечити одна одній. Для цього повинен бути передбачений механізм логічного висновку, який, в тому числі, перевіряє на несуперечливість додаються аксіоми і виводяться в онтології затвердження;

в) розширяємість. Ядром онтології є спочатку введені поняття і описують їх аксіоми. В онтології повинен бути передбачений механізм розширення спільно використовуваних словників понять без порушення цілісності системи.

1.3 Постановка задачі

В ході розробки атестаційної роботи необхідно розглянути такі питання:

а) проаналізувати існуючу систему оподаткування України;

б) визначити найбільш затребувані простими користувачами частини податкового кодексу України;

в) за результатами аналізу предметної галузі визначити розміри податків в різних сферах оподаткування України, визначити умови для визначення розміру рівня податку в даних сферах;

г) використати онтологічне моделювання для розробки онтології предметної галузі.

2. ВИВЧЕННЯ СИСТЕМИ ОПОДАТКУВАННЯ УКРАЇНИ. ОБГРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ

2.1 Податкове законодавство України

Проблеми оподаткування завжди були актуальними і цікавили людство з давніх давен. Відомі історики і філософи по-різному визначали податки в своїх трудах: як дозволена форму грабежу Фома Аквінський; Ш. Монтеск'є вважав, що багато мудрості і розуму потрібно, щоб визначити частину, яку у поданих забирають, і частину, що лишають ним; Адам Сміт, якого по праву вважають одним з основоположників теорії оподаткування наголошував, що податки для тих, хто їх виплачує є ознакою свободи, а не рабства.

Сьогодні податки і система оподаткування є не тільки джерелом бюджетних надходжень, а це також найважливіші структурні елементи ринкової економіки. Зрозуміло, що без формування раціональної податкової системи, яка не буде пригнічувати підприємницьку діяльність і дозволить проводити ефективну бюджетну політику, неможливі прогресивні ефективні перетворення економіки України.

Сукупність загальнодержавних і місцевих податків та зборів, що стягуються в установленому податковим кодексом порядку, становить податкову систему України.

На теперішній час українська система оподаткування досить недосконала. Це створює велику кількість питань і труднощів. Через суперечності податкового законодавства України навіть юристи з великим досвідом і податкові консультанти не можуть відповісти на деякі питання оподаткування без детального вивчення законодавчої бази і численних роз'яснень окремих питань офіційними органами, практики податкового

арбітражу. До того ж з прийняттям податкового кодексу складності збільшилися.

Відповідальність за правильність обчислень, своєчасність сплати податків та інших платежів, що є обов'язковими, та дотримання законодавства про оподаткування покладається на платників податків. У разі приховування або заниження платниками суми податку або іншого обов'язкового платежу з платника стягується сума донарахованого податку або інших обов'язкових платежів і штраф у двократному розмірі тієї ж суми, а в разі повторного порушення протягом року після встановленого порушення - штраф у п'ятикратному розмірі.

За неподання або несвоєчасне подання декларації, розрахунків та інших документів податковим адміністраціям, а також неподання або несвоєчасне подання платіжних доручень на сплату податку установам банків - 10% належних до сплати сум податків (платежів). Санкції застосовуються до платників податків, які допустили помилки в своїх розрахунках або деклараціях, незалежно від того, зроблено це через незнання, халатності або з умислом.

Податкове законодавство України складається з [5]:

- а) Конституції України;
- б) податкового кодексу;
- в) митного кодексу України, інших законів з питань митної справи в частині регулювання правовідносин, що виникають у зв'язку з оподаткуванням митом операцій з переміщення товарів через митний кордон України;
- г) міжнародних договорів, згода на обов'язковість яких надана Верховною Радою (ВР) України і які регулюють питання оподаткування;
- д) нормативно-правових актів, прийнятих на підставі цього Кодексу та законів з питань митної справи;

е) рішень ВР, органів місцевого самоврядування з питань місцевих податків і зборів, прийнятих за правилами, встановлених цим Кодексом.



Рисунок 2.1 - Структура податкового законодавства України

У атестаційній роботі розглядається прикладна предметна галузь оподаткування України, таким чином не розглядатиметься вся структура податкового законодавства України, а лише її частина, у вигляді певних розділів податкового кодексу України.

2.2 Податковий кодекс України

Податковий кодекс України регулює відносини, що виникають у сфері справляння податків і зборів, зокрема визначає вичерпний перелік податків та зборів, що справляються в Україні, та порядок їх адміністрування, платників податків та зборів, їх права та обов'язки, компетенцію

контролюючих органів, повноваження і обов'язки їх посадових осіб під час здійснення податкового контролю, а також відповідальність за порушення податкового законодавства [6].

2.2.1 Структура податкового Кодексу України

Податковий кодекс України має наступну структуру:

- загальні положення;
- адміністрування податків, зборів, платежів;
- податок на прибуток підприємств;
- податок на доходи фізичних осіб;
- податок на додану вартість;
- акцизний податок;
- збір за першу реєстрацію транспортного засобу;
- екологічний податок;
- рентна плата за транспортування нафти і нафтопродуктів по магістральними нафтопроводами та нафтопродуктопроводами, транзитне транспортування трубопроводами природного газу та аміаку територією України;
- рентна плата за нафту, природний газ і газовий конденсат, що видобуваються в Україні;
- плата за користування надрами;
- місцеві податки і збори;
- плата за землю;
- спеціальні податкові режими;
- збір за користування радіочастотним ресурсом України;
- збір за спеціальне використання води;
- збір за спеціальне використання лісових ресурсів;

- особливості оподаткування платників податків в умовах дії угоди про розподіл продукції;
- посадові особи контролюючих органів і їх правовий і соціальний захист;
- податкова міліція;
- прикінцеві положення;
- перехідні положення.

2.3 Онтологічний підхід

При формуванні поля знань основним елементом є сам процес отримання знань, в якому виконується перенесення компетентності експертів на інженерів. Цей процес в літературі з експертних систем (ЕС) має декілька назв: формування, одержання, придбання, виявлення, здобування знань. У спеціальній літературі переважно використовуються два терміни: elicitation (виявлення, витягання, встановлення) та acquisition (придбання) [14].

Термін «придбання» трактують широко – в цьому випадку він об'єднує весь процес передавання знань від експерта до бази знань експертної системи, або вузько – як спосіб автоматизованої побудови бази знань, використовуючи діалог експерта зі спеціальною програмою (структура поля знань заздалегідь закладається у програмне застосування). В обох випадках термін «придбання» не стосується структуризації знань предметної галузі. Цей процес описується терміном «видобування». Кожному типу задач відповідають свої методи придбання або видобування знань.

Видобування знань (knowledge mining) проєдставляє собою процедуру взаємодії експерта з джерелом знань, у результаті чого підвищується ефективність процесу прийняття рішень фахівцями [15].

Сьогодні більша кількість розробників експертних систем відзначає, що процес видобування знань залишається “вузьким” місцем у побудові

складних експертних систем, до того ж їм доводиться самотійно розробляти методи видобування знань, що призводить до наступних проблем [16]:

- організаційні питання;
- неправильно обраний метод видобування знань, що не збігається зі структурою знань в обраній галузі;
- неадекватна модель або мова подання знань.

Крім цього:

- невміння налагодити контакт з експертами предметної галузі, їхня незацікавленість;
- термінологічні конфлікти;
- відсутність цілісної системи знань, в результаті виконується витягання тільки “фрагментів”;
- спрощення уявлень експерта щодо предметної галузі.

Процес видобування знань є досить тривалим і трудомістким. В цьому процесі інженерові, який володіє спеціальними знаннями з психології, системного аналізу, математичної логіки необхідно відтворити модель предметної галузі, якою користуються експерти для ухвалення рішень. Частіше розроблювачі-початківці експертних систем для спрощення цієї процедури намагаються підмінити інженера зі знань самим експертом. З багатьох причин це небажано.

По-перше, велика частина знань експерта – це результат численних нашарувань ступенів досвіду. Знаючи, що з A виводиться B , експерт не усвідомлює, що ланцюжок його міркувань був набагато довший, наприклад $A \rightarrow D \rightarrow C \rightarrow B$ або $A \rightarrow Q \rightarrow R \rightarrow B$.

По-друге, мислення діалогічне. І тому діалог інженера зі знань і експерта – найприродніша форма вивчення досвіду експерта, у якому зберігаються знання, частина яких має невербальний характер, тобто виражені не у формі слів, а, наприклад, у формі наочних образів. Саме в

процесі пояснення експерта інженерові зі знань розмиті асоціативні образи перетворюються у вербальні знання.

По-третє, експертів важче створити модель предметної галузі внаслідок глибини й обсягу інформації, якою він володіє. Крім цього, в ситуаційному керуванні виявлено, що об'єкти реального світу зв'язані більш ніж 200 типами різноманітних відношень (тимчасові, просторові, причинно-наслідкові, типу "частина–ціле" і т.інше). Ці відношення і зв'язки предметної галузі утворюють складну систему, з якої виділити головну структуру іноді доступно тільки аналітику, що до того ж володіє системною методологією.

Термін "придбання" залишається автоматизованим систем прямого спілкування з експертом, які безпосередньо здобувають вже готові фрагменти знань відповідно до структур, закладених розроблювачами систем. Більшість таких інструментальних засобів спеціально орієнтовано на конкретні інтелектуальні системи із жорстко позначеною предметною галуззю й моделлю подання знань, тобто вони не універсальні.

Придбання знань (knowledge acquisition) – процес наповнення бази знань експертом з використанням спеціалізованих програмних засобів.

Наприклад, система TEIRESIAS, що є батьківською для всіх інструментаріїв придбання знань, і призначена для наповнення бази знань системи MYCIN або її дочірніх галузей, побудованих на "оболонці" EMYCIN в галузі медичної діагностики з використанням продукційної моделі подання знань. Три покоління та основні тенденції системи придбання знань описуються нижче.

Термін формування знань традиційно закріпився за перспективною галуззю інженерії знань, що активно розвивається і займається створенням методів, моделей і алгоритмів навчання. Формування знань об'єднує індуктивні моделі формування знань і автоматичного породження гіпотез, наприклад, ДСМ-метод на основі навчальних вибирань, навчання за аналогією та інші методи. Ці моделі дають змогу виявити причинно-

наслідкові залежності в базах даних з неповною інформацією, що містять структуровані числові й символічні об'єкти, у тому числі в умовах неповноти інформації.

Формування знань (machine learning) – процес аналізу даних і виявлення прихованих закономірностей з використанням спеціального математичного апарату програмних засобів [17].

Традиційно до завдань формування знань або машинного навчання належать задачі індуктивного виведення й синтезу з додатковою інформацією, прогнозування, ідентифікації та синтезу функцій, розшифровування мов. У широкому сенсі до навчання на прикладах можна зарахувати і методи навчання розпізнавання образів [9].

Індуктивне виведення правил із фактів застосоване також у системах AQ, AQUINAS, KSSI, INSTIL і деяких інших.

Найдослідженішими серед методів машинного навчання є методи розпізнавання образів, зокрема алгебраїчний підхід, в якому передбачається збагачення вихідних евристичних алгоритмів з використанням алгебраїчних операцій і побудови сімейства алгоритмів, що гарантує одержання коректного алгоритму для виконання досліджуваного класу задач. Це алгоритм, що правильно класифікує кінцеву вибірку по всім класам. Однак застосування методів формування знань не є промисловою технологією розроблення баз знань [5].

Для того, щоб описані методи стали елементами технології інтелектуальних систем, необхідно виконати наступні етапи:

- забезпечити механізм сполучення незалежно створених баз даних, що мають різні схеми, з базами знань інтелектуальних систем;
- встановити відповідність між набором полів бази даних і безліччю елементів декларативного компонента бази знань;

– виконати перетворення результату роботи алгоритму навчання у спосіб подання, підтримуваний програмними засобами інтелектуальної системи.

Крім перерахованих, існують також й інші стратегії одержання знань. Наприклад, при навчанні на прикладах (case-based reasoning), коли джерелом знань є множина прикладів предметної галузі. Навчання на основі прикладів або прецедентів передбачає налаштування алгоритму розпізнавання на завдання за допомогою надання прикладів, класифікація яких відома.

Навчання на прикладах пов'язане з машинним навчанням, але відмінність у тому, що результат навчання повинен бути інтерпретований у деякій моделі, в якій вже містяться факти та закономірності предметної галузі, і перетворений у спосіб подання, що допускає використання результату навчання в базі знань для моделювання міркувань, для роботи механізму пояснення, тобто робить результат навчання елементом відповідної технології. Наприклад, у системі INDUCE породжується несуперечливий опис деякого класу об'єктів з великою кількістю прикладів і контрприкладів заданого класу. Як мова подання використовується мова багатозначної логіки першого порядку.

Є ще два методи машинного навчання – data mining і knowledge discovery. Обидва підходи ґрунтуються на пошуку закономірностей та на аналізу даних.

Тааким чином, можна виділити три основні стратегії проведення стадії одержання знань для розробки інтелектуальних систем [18].

1. З використанням комп'ютера з відповідним програмним інструментарієм, інакше це придбання знань.

2. З використанням програм навчання за наявності репрезентативної представницької вибірки прикладів прийняття рішень у предметній галузі та відповідних пакетів прикладних програм, інакше це формування знань.

3. Без використання обчислювальної техніки безпосереднім контактом інженера зі знань і джерела знань (експерт, спеціальна література або інші джерела), інакше це видобування знань.

У повсякденному житті існує безліч прикладних предметних галузей, в яких звичайний користувач не є фахівцем. Завданням атестаційної роботи є використання онтологічного підходу, для створення такої моделі, яка була б зручна та корисна всім користувачам.

До переваг такого підходу можна віднести зручність уявлення і створення моделі, сучасний підхід для вирішення різних прикладних задач, можливість розширення моделі без її повного зміни. Дана технологія дозволяє створити єдину модель предметної галузі, якою зможе користуватися широке коло користувачів.

На прикладі прикладної предметної галузі системи оподаткування України потрібно побудувати онтологію, з метою полегшення вивчення і знаходження необхідних знань в даній сфері.

Податковий кодекс України містить достатньо великий обсяг текстової інформації, яка постійно змінюється і доповнюється. Користувачеві, який не розуміється на тонокошах цього документу, буде досить важко використовувати його і шукати необхідну інформацію в обмежені терміни. Для вирішення даної задачі необхідно створити онтологію даної предметної галузі. Створена онтологія дозволить новим користувачам швидше знаходити необхідний їм матеріал з величезної кількості інформації, що значно спростить роботу. Також онтологія є легко розширюваної структурою, таким чином, з появою змін або доповнень в оригінальному тексті, це без зусиль можна додати в онтологію.

2.3.1 Переваги онтології

Онтології забезпечують загальне розуміння в конкретних областях, які можуть бути передані між людьми і прикладним системам.

Онтології можуть бути використані для:

- а) виділення загального сенсу структури інформації;
- б) включення повторного використання вже існуючих знань про предметну галузь, а не створення нових;
- в) діяння припущень в предметній галузі однозначними;
- г) перевірки знань про предметну галузь.

2.4 Мови уявлення онтологій

Для реалізації різних онтологій розробляються мови їх подання, тому ключовим елементом в проектуванні є вибір відповідної мови специфікації онтологій (Ontology Specification Language). Призначення таких мов - дати можливість вказувати додаткову машинно-інтерпретуєму семантику ресурсів, зробити машинне подання даних більш схожим на стан речей в реальному світі, істотно підвищити виразні можливості концептуального моделювання слабо структурованих Web-даних.

Існують традиційні мови специфікації онтологій: Ontolingua, CycL, мови, засновані на дескриптивних логіках (такі як LOOM), мови, засновані на фреймах (OKBC, OCML, Flogic). Пізніші мови засновані на Web-стандартах (XOL, SHOE, UPML). Спеціально для обміну онтологіями через Web були створені RDF (S), DAML, OIL, OWL, які будуть розглянуті далі.

В цілому, відмінність між традиційними мовами і Web-мовами специфікації онтології полягає в виразних можливостях опису предметної галузі і деяких можливостях механізму логічного висновку для цих мов.

Сьогодні деякі з таких мов набули неабиякої популярності і широко застосовуються (зокрема для опису інформаційних ресурсів і сервісів Інтернет).

Мова RDF. В рамках проекту семантичної інтерпретації інформаційних ресурсів Інтернет (Semantic Web) був запропонований стандарт опису метаданих про документ Resource Description Framework, який використовує XML-синтаксис.

RDF використовує базову модель даних «об'єкт - атрибут - значення» і здатний зіграти роль універсальної мови опису семантики ресурсів і взаємозв'язків між ними. Ресурси описуються в вигляді орієнтованого розміченого графа - кожен ресурс може мати властивості, які в свою чергу також можуть бути ресурсами або їх колекціями. Всі словники RDF використовують базову структуру, що описує класи ресурсів і типи зв'язків між ними. Це дозволяє використовувати різноманітні децентралізовані словники, створені для машинної обробки за різними принципами і методами. Важливою особливістю стандарту є розширюваність: можна задати структуру опису джерела, використовуючи і розширюючи такі вбудовані поняття RDF-схем, як класи, властивості, типи, колекції. Модель схеми RDF включає наслідування класів і властивостей.

RDF вже отримав підтримку багатьох провідних виробників ПЗ. Розроблено ряд програмних продуктів, що дозволяють створювати RDF-описи для різного роду систем. Передбачаються можливості інтеграції існуючих сховищ інформації в загальну базу семантичного опису та інтеграції концепції RDF-бази з форматом MPEG. RDF Schema - стандарт, запропонований з ініціативи W3C для подання онтологічних знань. Він специфікує безліч всіляких допустимих схем даних. Моделі предметних галузей описуються за допомогою ресурсів, властивостей і їх значень. RDFS надає хороші базові можливості для опису словників типів предметних галузей. Одне з обмежень - неможливість за допомогою RDFS висловити аксіоматичні знання, т. Е. Задати аксіоми і правила виводу, побудовані на них.

DAML + OIL - семантична мова розмітки Web-ресурсів, що розширює стандарти RDF і RDF Schema за рахунок більш повних примітивів моделювання. Остання версія DAML + OIL забезпечує багатий набір конструкцій для створення онтології і розмітки інформації таким чином, щоб їх могла читати і розуміти машина.

Першими варіантами щодо опису онтології на базі RDFS були DARPA DAML-ONT (DARPA Agent Markup Language) і European Commission OIL (Ontology Inference Layer). Ці стандарти специфікації і обміну онтологіями були розроблені для підтримки процесу обміну знаннями та інтеграції знань. На базі цих пропозицій і виникло спільне рішення DAML + OIL. Онтологія DAML + OIL складається з: заголовків (headers); елементів класів (class elements); елементів властивостей (property elements); примірників (instances).

OWL (Web Ontology Language) - мова представлення онтологій, що розширює можливості XML, RDF, RDF Schema і DAML + OIL. Цей проект передбачає створення потужного механізму семантичного аналізу. Планується, що в ньому будуть усунені обмеження конструкцій DAML + OIL.

Онтології OWL - це послідовності аксіом і фактів, а також посилань на інші онтології. Вони містять компоненту для запису авторства та іншої докладної інформації, є документами Web, на них можна посилатися через URI.

Онтології грають важливу роль в організації обробки знань на основі Web, а також для їх спільного використання. Онтології, які визначаються як спільно використовуємі формальні концепції конкретних предметних галузей, дають загальне уявлення про розділи, інформацією про які можуть обмінюватися і люди, і додатки. Онтології відрізняються від XML-схем тим, що це подання знань, а не формати повідомлень (більшість Web-стандартів складаються з комбінації форматів повідомлень і специфікацій протоколів).

2.4.1 Мова опису Owl

OWL (Ontology Web Language) - це мова, що базується на напрямку Semantic Web, службовець для подання web-онтологій предметних галузей, схвалений консорціумом W3C.

Під онтологією розуміється деякий набір термінів предметної галузі та зв'язків між цими термінами.

OWL надає три підмножини, що мають різну ступінь деталізації [7]:

а) OWL Lite призначене для користувачів або додатків, які потребують лише класифікаційну ієрархію сутностей і деякі прості умови узгодженості сутностей;

б) OWL DL (Description Logic) розраховане на користувачів, яким необхідна максимальна ступінь виразу можливостей мови без втрати обчислювальної повноти, без втрати жодного з семантичних втілень - змістовних тлумачень висновків, отриманих формально-логічним шляхом і можливості розв'язання, що означає що обчислення, будуть закінчені за кінцевий час. Рівень OWL DL орієнтований на існуючі сьогодні системи опису знань і системи логічного програмування;

в) OWL Full розраховане на користувачів, яким необхідні максимальні виразні можливості мови і свобода вибору кінцевого формату Resource Description Framework (RDF), але без будь-яких гарантій обчислювальної повноти і можливості розв'язання. OWL Full дозволяє розширити зміст терміну, взятого з будь-якого заданого словника, і додати його в онтологію.

В якості свого синтаксису OWL використовує мову XML. Основними елементами мови є властивості, класи і обмеження. Ці елементи дозволяють реалізувати уявлення про світ, як про безліч сутностей (об'єктів), що характеризуються деяким набором властивостей. Ці сутності складаються між собою в певних відносинах і об'єднуються за певними ознаками (властивостями і обмеженням) в групи (класи).

У мові OWL властивості поділяються на два види: властивості-характеристики (`DatatypeProperty`) і властивості-зв'язку (`ObjectProperty`). Перші характеризують об'єкти (класи) і приймають як своїх значень дані певних типів. Другі асоціює об'єкти (класи) один з одним і відповідно приймають в якості своїх значень об'єкти (класи).

Крім того, на властивості можуть накладатися обмеження. Обмеження поділяються на два види: глобальні та локальні. До глобальних обмежень належать домени (`domain`) (класи, об'єкти яких можуть володіти цими властивостями) і діапазони (`range`) (класи, об'єкти яких можуть виступати в якості значень цих властивостей). Локальні обмеження накладаються на властивості в рамках певного класу і можуть ще більше звужувати діапазони для властивостей в рамках цього класу, визначати потужність властивостей і їх види.

Також мова OWL має механізми опису версій онтології і механізми агрегування даних, що містяться в онтологіях.

2.5 Структура Owl онтології

Будь-яка онтологія має заголовок і тіло. У заголовку міститься інформація про саму онтологію, що імпортується онтологією. За заголовком слідує тіло онтології, що містить опис класів, властивостей і примірників.

2.5.1 Класи

В OWL введений новий термін клас (`owl: Class`). Необхідність цього пояснюється тим, що не всі класи діалектів DL і Lite є `rdfs` класами (в цьому випадку `owl: Class` є підкласом `rdfs: Class`). У діалекті Full, подібних обмежень немає, і `owl: Class` фактично є синонімом `rdfs: Class`.

Для організації класів в таксономії використовується властивість `rdfs: subclassOf`.

Особливе місце займають два взаємодоповнюючих класи Thing і Nothing. Перший з них є надкласом будь-якого класу OWL, другий - підкласом будь-якого класу OWL. Примірник будь-якого класу OWL входить в екстенціонал класу Thing. Екстенціонал класу Nothing є порожньою безліччю.

OWL клас може бути описаний наступними способами:

- а) ідентифікатором класу (URI);
- б) перерахуванням всіх екземплярів класу;
- в) обмеженням властивості;
- г) перетином 2 і більше визначень класів;
- д) об'єднанням 2 і більше визначень класів;
- е) доповненням визначення класу.

Тільки перший спосіб визначає іменований клас OWL. Все решта визначають анонімний клас через обмеження його екстенціонала. Спосіб 2 явно перераховує екземпляри класу, спосіб 3 обмежує екстенціонал тільки тими екземплярами, які задовольняють даній властивості. Способи 4, 5 і 6 використовують логічні операції (AND, OR і NOT) над екстенціоналу відповідних класів, щоб визначити екстенціонал нового класу.

Описи класу формують будівельні блоки для визначення класів через аксіоми. Найпростіша аксіома визначає іменований клас представляється у вигляді `<owl: Class rdf: ID = "Human" />`. Все що постулює ця аксіома - існування класу з ім'ям Human.

В OWL визначені ще 3 конструкції, комбінуючи які можна визначати більш складні аксіоми класів.

`rdfs: subClassOf` говорить про те, що екстенціонал одного класу (підклас) повністю входить в екстенціонал іншого (надклас).

`owl: equivalentClass` говорить про те, що екстенціоналу двох класів збігаються.

`owl: disjointWith` говорить про те, що екстенціоналу двох класів не

перетинаються. Іноді кажуть, що таким чином визначаються диз'юнктивні класи.

2.5.2 Властивості

В OWL виділяють дві категорії властивостей: властивості-об'єкти (або об'єктні властивості) і властивості-значення. Перші пов'язують між собою індивіди (екземпляри класів). Другі пов'язують індивіди зі значеннями даних. Обидва класи властивостей є підкласами класу `rdf: Property`.

Для визначення нових властивостей як примірників `owl: ObjectProperty` або `owl: DatatypeProperty` використовуються аксіоми властивостей.

Крім того, OWL підтримує такі конструкції для побудови аксіом властивостей:

а) конструкції RDF Schema: `rdfs: subPropertyOf` (визначає подвойство даного властивості), `rdfs: domain` (визначає домен властивості), `rdfs: range` (визначає діапазонсвойства);

б) відносини між властивостями: `owl: equivalentProperty` (визначає еквівалентну властивість) і `owl: inverseOf` (визначає зворотне властивість);

в) обмеження глобальної кардинальності: `owl: FunctionalProperty` (визначає однозначне властивість - однозначне відображення домена властивості на діапазон) і `owl: InverseFunctionalProperty` (назад функціональне властивість, тобто визначає, що властивість зворотне даним властивості є однозначним);

г) логічні показники якості: `owl: SymmetricProperty` (визначає властивість як симетричне) і `owl: TransitiveProperty` (визначає Транзитивне властивість).

2.5.3 Індивіди

Індивіди визначаються за допомогою аксіом індивідів (т.зв. фактів). Розглянемо два види фактів:

а) факти про членство індивідів в класах і про значення властивостей індивідів;

б) факти про ідентичність / різних індивідів.

Аксіоми другого виду необхідні для судження про ідентичність індивідів. Справа в тому, що в OWL не робиться ніяких припущень про відмінність (і вже тим більше про збіг) двох індивідів мають різні ідентифікатори URI.

Такі переконання виражаються аксіомами ідентичності за допомогою наступних конструкцій:

а) owl: sameAs постулює, що два посилання URI посилаються на один і той же індивід;

б) owl: differentFrom постулює, що два посилання URI посилаються на різні індивіди;

в) owl: AllDifferent надає засіб для визначення списку попарно різних індивідів.

3. СТВОРЕННЯ ОНТОЛОГІЇ ПРИКЛАДНОЇ ПРЕДМЕТНОЇ ГАЛУЗІ

3.1 Інструменти побудови онтологій

Побудова онтологій - складний процес, що займає багато часу. Щоб полегшити його, в середині 90-х років почали створюватися перші середовища для процесу розробки онтологій. Вони забезпечили інтерфейси, які дозволили виконувати концептуалізацію, реалізацію, перевірку несуперечності і документування. За останні роки число інструментів онтологій різко зросло (сайт консорціуму W3C, наприклад, надає список більш ніж 50 інструментів редагування).

В даний час для створення і підтримки онтологій існує цілий ряд інструментів, які крім загальних функцій редагування і перегляду виконують підтримку документування онтологій, імпорт і експорт онтологій різних форматів і мов, підтримку графічного редагування, управління бібліотеками онтологій і т.ін.

Розглянемо найбільш відомі інструменти інженерії онтологій.

Система Ontolingua була розроблена в KSL (Knowledge Systems Laboratory) Стенфордського університету і стала першим інструментом інженерії онтологій. Вона складається з сервера і мови подання знань.

Сервер Ontolingua організований у вигляді набору онтологій, що відносяться до Web-додатків, які надбудовуються над системою подання знань Ontolingua. Редактор онтологій - найбільш важливий додаток сервера Ontolingua є Web-додатком на основі форм HTML. Крім редактора онтологій Сервер Ontolingua включає мережевий додаток Webster (отримання визначень концептів), сервер ОКВС (доступ до онтології Ontolingua по протоколу ОКВС) і Chimaera (аналіз, об'єднання, інтегрування онтологій). Всі додатки, крім сервера ОКВС, реалізовані на основі форм HTML. Система подання знань реалізована на Lisp.

Сервер Ontolingua також надає архів онтологій, що включає велику кількість онтологій різних предметних галузей, що дозволяє створювати онтології з уже існуючих. Сервер підтримує спільну розробку онтології декількома користувачами, для чого використовуються поняття користувачів і груп. Система включає графічний браузер, що дозволяє переглянути ієрархію концептів, включаючи екземпляри. Ontolingua забезпечує використання принципу множинного спадкоємства і багатий набір примітивів. Збережені на сервері онтології можуть бути перетворені в різні формати для використання іншими додатками, а також імпортовані з ряду мов в мову Ontolingua.

Protégé [8] - локальна, вільно поширювана Java-програма, розроблена групою медичної інформатики Стенфордського університету (перша версія - 1987, остання Protégé-2.1.1 - червень 2004). Програма призначена для побудови (створення, редагування і перегляду) онтологій прикладної галузі. Її початкова мета - допомогти розробникам програмного забезпечення в створенні і підтримці явних моделей предметної галузі та включення цих моделей безпосередньо в програмний код. Protégé включає редактор онтологій, що дозволяє проектувати онтології розвертаючи ієрархічну структуру абстрактних чи конкретних класів і слотів. Структура онтології зроблена аналогічно ієрархічній структурі каталогу. На основі сформованої онтології, Protégé може генерувати форми отримання знань для введення примірників класів і підкласів.

Protégé заснований на фреймовій моделі представлення знання ОКВС (Open Knowledge Base Connectivity) і забезпечений рядом плагінів, що дозволяє його адаптувати для редагування моделей збережених в різних форматах (стандартний текстовий, в базі даних JDBC, UML, мов XML, XOL, SHOE, RDF і RDFS, DAML + OIL, OWL).

OntoEdit спочатку був розроблений в інституті AIFB (Institute of Applied Informatics and Formal Description Methods) Університету Karlsruhe

(зараз комерціалізувати Ontoprise GmbH) виконує перевірку, перегляд, кодування і модифікацію онтологій. В даний час OntoEdit підтримує мови представлення: FLogic, включаючи машину виведення, OIL, розширення RDFS і внутрішню, засновану на XML, серіалізацію моделі онтології використовуючи OXML - мова представлення знань OntoEdit (OntoEdit "s XML-based Ontology representation Language). До переваг інструменту можна віднести зручність використання; розробку онтології під керівництвом методології і за допомогою процесу логічного висновку; розробку аксіом; розширяемую структуру за допомогою плагінів, а також дуже хорошу документацію.

Так само як і Protégé, OntoEdit - автономне Java-додаток, яке можна локально встановити на комп'ютері, але його коди закриті. Архітектура OntoEdit подібна Protégé.

Існує дві версії OntoEdit: вільно поширювана OntoEdit Free (обмежена 50 концептами, 50 відносинами і 50 екземплярами) і ліцензована OntoEdit Professional (немає обмежень на розмір). Природно, що OntoEdit Professional має більш широкий набір функцій і можливостей (наприклад, машину виведення, графічний інструмент запитів, більше модулів експорту та імпорту, графічний редактор правил, підтримка баз даних JDBC і т.д.).

OilEd - автономний графічний редактор онтологій, розроблений в Манчестерському університеті в рамках європейського IST проекту On-To-Knowledge. Інструмент заснований на мові OIL (зараз адаптований для DAML + OIL, в перспективі - OWL), який поєднує в собі фреймової структуру і виразність дескриптивної логіки (Description Logics) з сервісами міркування. Що дозволило забезпечити зрозумілий і інтуїтивний стиль інтерфейсу користувача і переваги підтримки міркування (виявлення логічно суперечливих класів і прихованих відносин підкласу).

З недоліків можна виділити відсутність підтримки примірників. Існуюча версія не забезпечує повне середовище розробки - не підтримує

розробку онтологій великого масштабу, міграцію та інтеграцію онтологій, контроль версій і т.ін. OilEd можна розглядати як "NotePad" редакторів онтологій, що пропонує достатню функціональність, щоб дозволити користувачам будувати онтології і продемонструвати, як можна використовувати механізм міркування FaCT для перевірки онтології на несуперечливість.

Останнім часом спостерігається зростання популярності редактора OilEd. Він використовується як для навчання, так і для дослідження. Інструмент вільно поширюється по загальнодоступній ліцензії GPL.

WebOnto розроблений для Tadzebao - інструменту дослідження онтологій і призначений для підтримки спільного перегляду, створення і редагування онтологій. Його цілі - простота використання, надання коштів масштабування для побудови великих онтологій.

Для моделювання онтологій WebOnto використовує мову OCML (Operational Conceptual Modeling Language). У WebOnto користувач може створювати структури, включаючи класи з множинним спадкуванням, що можна виконувати графічно. Всі слоти успадковуються коректно. Інструмент перевіряє нововведені дані контролем цілісності коду OCML.

Інструмент має ряд корисних особливостей: збереження структурних діаграм, роздільний перегляд відносин, класів, правил і т.д. Інші можливості включають спільну роботу декількох користувачів над онтологією, використання діаграм, функцій передачі і прийому і ін.

OntoSaurus є Web-браузером для баз знань LOOM. Він складається з двох основних модулів: сервера онтологій і Web-браузера для редагування і перегляду онтологій LOOM за допомогою HTML-форм, забезпечуючи для них графічний інтерфейс. OntoSaurus також надає обмежені кошти редагування, але його основна функція - перегляд онтологій. Але для побудови складних онтологій потрібно розуміти мову LOOM. Більшість користувачів будують онтологію на мові LOOM в іншому редакторі, а потім

для перегляду і редагування імпортують його в OntoSaurus. У OntoSaurus реалізовані всі можливості мови LOOM. Забезпечується автоматичний контроль сумісності, дедуктивна підтримка міркування і деякі інші функції.

Конструктор онтологій ODE (Ontological Design Environment), який взаємодіє з користувачами на концептуальному рівні на відміну від інструментів, подібно OntoSaurus, спілкуються на символному рівні. Мотивом для ODE послужило те, що людям простіше формулювати онтології на концептуальному рівні. ODE забезпечує користувачів набором таблиць для заповнення (концептів, атрибутів, відносин) і автоматично генерує для них код в LOOM, Ontolingua і FLogic. ODE становить частину методології повного життєвого циклу побудови онтології згідно Methontology. Інструмент отримав свій подальший розвиток в WebODE, який інтегрує всі сервіси ODE в одну архітектуру, зберігає свої онтології в реляційній базі даних, забезпечує додаткові сервіси (машину виведення, побудова аксіом, збір онтологій, генерацію каталогів).

KADS22 - інструмент підтримки проектування моделей знань згідно з методологією CommonKADS. Онтології складають частину таких моделей знань (інша частина - моделі виведення). Моделі CommonKADS визначені в CML (Conceptual Modeling Language). KADS22 - інтерактивний графічний інтерфейс для CML з наступними функціональними можливостями: синтаксичний аналіз файлів CML, друк, перегляд гіпертексту, пошук, генерація глосарію і генерація HTML.

Подальший розвиток в рамках проекту DWQ (Data Warehouse Quality) веде до інструменту i.com, інструментального засобу підтримки концептуальній стадії проекту інтегрованих інформаційних систем. i.com використовує розширену модель даних сутностей-зв'язків, доповнивши її обмеженнями багатовимірної агрегації і проміжних схем. Інструмент i.com повністю інтегрований з потужним сервером міркування на основі DL. i.com

служить головним чином для інтелектуального концептуального моделювання.

3.2 Інструменти для відображення і об'єднання онтологій

Інструменти об'єднання онтологій допомагають користувачам знайти схожість і відмінність між вихідними онтологіями і створюють результуючу онтологію, яка містить елементи вихідних онтологій [9]. Для досягнення цієї мети вони автоматично визначають відповідності між концептами в початкових онтологіях або забезпечують середовище, де користувач може легко знайти і визначити ці відповідності. Ці інструменти відомі як інструменти відображення, вирівнювання і об'єднання онтологій, так як вони виконують подібні операції для процесів відображення, вирівнювання і об'єднання.

Відображення онтології полягає в знаходженні семантичних зв'язків подібних елементів з різних онтологіях. Вирівнювання онтологій полягає в тому, щоб встановити різні види відповідності між двома онтологіями, а потім повторно зберегти вихідні онтології і таким чином, в подальшому використовувати інформацію один одного. Об'єднання онтологій - генерація однієї узгодженої онтології з двох вихідних.

Дослідники різних галузей інформатики працюють над автоматичним або підтримуваним інструментально об'єднанням онтологій (або ієрархії класів, або об'єктно-орієнтованих схем, або схем баз даних - певна термінологія змінюється в залежності від галузі застосування). Однак і автоматичне об'єднання онтологій, і створення інструментальних засобів, які б управляли користувачем в цьому процесі, знаходяться на ранніх стадіях розвитку. У цьому розділі представлений короткий огляд деяких з існуючих підходів.

Інструментальні засоби, які мають справу з перебуванням відповідності між онтологіями, класифікуються:

- а) для об'єднання двох онтологій з метою створення однієї нової (PROMPT, Chimaera, OntoMerge);
- б) для визначення функції перетворення з однієї онтології в іншу (OntoMorph);
- в) для визначення відображення між концептами в двох онтологіях, знаходячи пари відповідних концептів (наприклад, OBSERVER, FCA-Merge);
- г) для визначення правил відображення для зв'язку тільки релевантних частин вихідних онтологій (ONION).

Розглянемо тепер вищезгадані засоби більш докладно.

PROMPT - доповнення до системи Protégé, реалізоване у вигляді плагіна, служить для об'єднання і угруповання онтологій. При об'єднанні двох онтологій PROMPT створює список запропонованих операцій. Операція може складатися, наприклад, з об'єднання двох термінів або копіювання термінів в нову онтологію. Користувач може виконати операцію, вибираючи одну із запропонованих або визначаючи безпосередньо операцію. PROMPT виконує обрану операцію і додаткові зміни, викликані цією операцією. Потім список запропонованих операцій модифікується і створюється список конфліктів і можливих рішень цих конфліктів. Це повторюється до тих пір, поки не буде готова нова онтологія.

Chimaera - інтерактивний інструмент для об'єднання, заснований на редакторі онтологій Ontolingua. Chimaera дозволяє користувачеві об'єднувати онтології, розроблені в різних формалізмах. Користувач може запитувати аналіз або керівництво від Chimaera в будь-який момент протягом процесу об'єднання, і інструмент направить його на ті місця в онтології, де потрібно його втручання. У своїх пропозиціях Chimaera головним чином покладається на те, з якої онтології прибутку концепти, ґрунтуючись на їхніх іменах. Chimaera залишає рішення про те, що робити користувачеві і не робить

ніяких пропозицій самостійно. Єдине таксономическое ставлення, яке розглядає Chimaera - відношення підклас-суперклас. Chimaera найближчий до PROMPT. Однак оскільки він використовує в своєму аналізі тільки ієрархію класу, він пропускає багато з відповідностей, які знаходить PROMPT. Ці відповідності включають пропозиції щодо об'єднання слотів з подібними іменами, які відносяться до об'єднаних класів, об'єднання доменів слотів, які були об'єднані і т. ін.

У OntoMerge об'єднана онтологія є об'єднання двох вихідних онтологій і набору аксіом сполучення. Перший крок в процесі об'єднання в OntoMerge полягає в трансляції обох онтологій до загального синтаксичному поданням на розробленому авторами мовою. Потім інженер онтології визначає аксіоми сполуки, що містять терміни з обох онтологій. Процес трансляції примірників виглядає наступним чином: всі примірники у початкових онтологіях, розглядаються як знаходяться в об'єднаній онтології. Потім на основі інструкцій в початкових онтологіях і аксіом сполучення машина виведення зробить висновок, таким чином, створюючи нові дані в об'єднаній онтології. OntoMerge надає інструменти для трансляції даних-екземплярів в об'єднану онтологію.

OntoMorph визначає набір операторів перетворення, які можна застосувати до онтології. Потім людина-експерт використовує початковий список пар і вихідних онтологій для визначення набору операторів, які повинні застосуватися до початкових онтологій для усунення відмінностей між ними, і OntoMorph застосовує ці оператори. Таким чином, сукупність операцій може виконуватися за один крок. Однак, людина-експерт не отримує ніякого керівництва за винятком початкового списку пар.

Система OBSERVER застосовує DL для відповіді на запити, використовуючи кілька онтологій і інформацію про відображення між ними. Спочатку користувачі визначають набір межонтологічних відносин. Система допомагає впоратися з цим завданням, знаходячи синоніми в

початкових онтологіях. Визначивши відображення, користувачі можуть формулювати запити в термінах DL за допомогою власної онтології. Потім OBSERVER використовує інформацію відображення для формулювання запитів до початкових онтологій. OBSERVER значною мірою покладається на той факт, що опису в онтологіях і запитах є змістовними.

FCA-Merge - метод для порівняння онтологій, які мають набір загальних примірників або набір загальних документів, анотуються за допомогою концептів вихідних онтологій. Грунтуючись на цій інформації, FCA-Merge використовує математичні методи з Formal Concept Analysis для того щоб зробити решітку концептів, що зв'язує концепти вихідних онтологій. Алгоритм пропонує відношення еквівалентності і підклас-суперклас. Потім інженер онтології може аналізувати результат і використовувати його як керівництво для створення об'єднаної онтології. Однак припущення, що дві поєднувані онтології використовують загальний набір екземплярів або мають набір документів, в якому кожен документ анотується термінами обох джерел занадто жорстке і на практиці така ситуація відбувається рідко. В якості альтернативи,

Система ONION (ONtology composItION) заснована на алгебрі онтологій. Тому, вона надає інструменти для визначення правил артикуляції (з'єднання) між онтологіями. Правила артикуляції зазвичай враховують тільки релевантні частини вихідних онтологій. Для того щоб запропонувати з'єднання, ONION використовує і лексичні методи, і методи на основі графів. Метод знаходження лексичного подібності між іменами концептів використовує словники і методи семантичної індексації, засновані на місцезнаходження групи слів в тексті.

3.3 Інструменти анотування на основі онтологій

Найважливішою передумовою реалізації цілей семантичного Web є можливість анотувати Web-ресурси семантичною інформацією. У зв'язку з цим в останні роки інструменти інженерії онтологій еволюціонують у бік розробки інструментів анотування на основі онтологій.

Інструмент анотації MnM забезпечує підтримку автоматичного і напівавтоматичного розмітки Web-сторінок семантичним змістом. MnM інтегрує Web-браузер і редактор онтологій і забезпечує відкриті інтерфейси зв'язку з серверами онтологій і інструментами вилучення інформації. MnM можна розглядати в якості одного з перших прикладів наступного покоління редакторів онтологій, на основі Web, що орієнтуються на семантичну розмітку і забезпечують механізм повномасштабної автоматичної розмітки Web-сторінок.

За допомогою SHOE's Knowledge Annotator користувач може також описувати зміст Web-сторінок. Інструмент має інтерфейс, який відображає екземпляри, онтології і затвердження (зібрані документи). Також забезпечується перевірка цілісності. SHOE's Knowledge Annotator дозволяє користувачам виконувати розмітку сторінок в SHOE, під керуванням онтологій доступних локально або через URL. Ці розмічені сторінки можуть бути проаналізовані інструментальними засобами, що знають мову SHOE, типу SHOE Search. Аннотуємою Web-сторінки можуть бути також проаналізовані іншим інструментом по імені Expos'e, а зміст буде збережено в репозитарії. Це SHOE-знання потім зберігається в базі знань Parka.

Інструмент Metabrowser також частково вирішує проблему аннотування Web-ресурсів. Він може працювати, наприклад, на базі онтології Дублінського ядра (Дублінський ядро можна розглядати як найпростішу онтологію) і пропонувати ряд можливостей для автоматичного створення та перегляду метаданих. Metabrowser (включаючи вільно поширювану версію), відображає метадані Web-сторінки разом із самою Web-сторінкою.

3.4 Порівняльний аналіз інструментів

Інструменти побудови онтологій можна розділити на два типи: розроблені для редагування онтологій певною мовою онтологій і інтегровані нарощувані інструментальні сайти (Web-додатки, на основі форм HTML і / або Java-апплетів), більшість з яких не залежить від мови уявлення.

Слід підкреслити, що більшість з розглянутих інструментальних засобів розробляються університетськими дослідницькими групами, які надають відкритий код, або пропонують вільний доступ до функцій. Однак найбільш перспективні з них передаються комерційним компаніям (наприклад, *OntoEdit Professional* - ліцензований продукт).

Інструменти *OntoEdit*, *WebODE* і *KADS22* дають підтримку методологій побудови онтологій, відповідно *On-To-Knowledge*, *Methontology* і *CommonKADS*, що не заважає їм використовуватися в інших методиках або взагалі без них.

Торкаючись технічного аспекту, а саме архітектури програмного забезпечення (локальна, клієнт-серверна, n-рівнева), розширюваності, мов програмування на яких реалізовані інструменти, способів зберігання онтологій (в файлах або базах даних), необхідно відзначити наступне. Більш ранні інструменти *Ontolingua*, *OntoSaurus* і *WebOnto* мають клієнт-серверну архітектуру. *Protégé*, *OntoEdit* і *OilEd* мають 3-х рівневу архітектуру, де існує чіткий поділ між зберіганням онтологій, модулями бізнес-логіки, логіки додатків і додатками інтерфейсу користувача. Ці інструменти мають більші можливості по нарощуванню (наприклад, за допомогою плагінів). Більшість інструментів зберігає свої онтології в текстових файлах, що обмежує розмір онтологій. Тільки *Protégé* і *WebODE* можуть зберігати свої онтології в базах даних і таким чином управляти великими онтологіями. Нарешті, більшість інструментів реалізовано на Java.

Вище вже говорилося про те, що моделі знання інструментів визначають компоненти, які повинні використовуватися при побудові онтології. Більшість інструментів представляє онтології, комбінуючи фрейми і логіку першого порядку. Однак це ще не означає, що вони можуть представляти одні й ті ж компоненти з однією і тією ж кількістю інформації. Тільки два з перерахованих інструментів, OilEd і OntoSaurus, засновані на дескриптивній логіці (DL).

Далі розглянемо деякі властивості редакторів онтологій. Інтерфейс користувача редакторів онтологій може бути Web-додатком, на основі форм HTML (Ontolingua, OntoSaurus і WebODE) і / або Java-апплетів (WebOnto) або локальним додатком (Protégé, OntoEdit, OilEd).

Всі редактори онтологій за винятком OilEd, Ontolingua і OntoSaurus забезпечують графічні засоби редагування і перегляду онтологій, де класи зазвичай представлені вузлами на графах, а відносини - дугами між ними. Додатково до цих графічних функцій, OilEd, OntoEdit Professional, Protégé і WebODE надають деяку підтримку в написанні формальних аксіом і складних виразів.

OntoEdit, Ontolingua, OntoSaurus, WebODE і WebOnto підтримують спільну розробку онтологій, надаючи окремим користувачам або групам користувачів дозвіл на доступ і написання різних наборів онтологій. Різноманітність інструментів для відображення, вирівнювання і об'єднання онтологій робить складним їх безпосереднє порівняння. Фактично, коли розробник повинен вирішити питання, який інструмент є найбільш гідною кандидатурою, все буде залежати від конкретного завдання. Наприклад, якщо поєднувані онтології спільно використовують набір екземплярів, то краще за всіх може працювати FCA-Merge. Якщо онтології мають екземпляри, але разом їх не використовують, і багато значення слотів містять текст, кращим вибором може стати Glue. Якщо тільки частини онтологій повинні бути відображені, можна було б вибрати інструмент Onion. Якщо онтології мають

дуже обмежену структуру, а концепти мають докладні визначення природною мовою (одному), інструментальні засоби ISI / USC можуть забезпечувати кращі відповіді. Якщо екземпляри взагалі не доступні, і онтології містять багато відносин між концептами, краще за всіх може працювати Prompt.

3.5 Редактор онтології

Для створення онтології було вирішено використовувати програмне забезпечення Protégé версії 3.4.8, так як даний продукт є зрозумілим і легким у використанні для користувачів, а також надається по безкоштовній ліцензії.

Protégé - це вільний, відкритий редактор онтологій і фреймворк для побудови баз знань [11].

Онтології, побудовані в Protégé, можуть бути експортовані в безліч форматів, включаючи RDF (RDF Schema), OWL і XML Schema.

Protégé має відкриту, легко розширювану архітектуру за рахунок підтримки модулів розширення функціональності.

Protégé підтримується значним співтовариством, що складається з розробників і вчених, урядових і корпоративних користувачів, що використовують його для вирішення завдань, пов'язаних зі знаннями, в таких різноманітних галузях, як біомедицина, збір знань і корпоративне моделювання.

Редактор Protégé - OWL дозволяє користувачам будувати онтології для семантичної павутини, зокрема на OWL. OWL - онтологія може включати описи класів, властивостей і їх примірників. Даючи таку онтологію, формальна семантика OWL визначає як отримувати логічні слідства, тобто факти, які не присутні безпосередньо в онтології, але можуть бути виведені з існуючих за допомогою семантики. Ці висновки можуть бути засновані на

одному документі або на безлічі розподілених документів, які були об'єднані з використанням певних механізмів OWL.

3.6 Основні завдання розробки онтології

У сучасному світі кожна людина так чи інакше стикається з питанням сплати податків, а зі зростаючою тенденцією побудови свого власного бізнесу, багатьом доводиться заглиблюватися в систему оподаткування набагато сильніше, ніж сплата стандартних податків.

Система оподаткування України, зокрема, податковий кодекс України є об'ємним документом з величезною кількістю інформації. Простому користувачеві, який не спеціалізується на даній сфері діяльності практично не надається можливим розібратися у всіх тонкощах системи, і отримати відповідь на його запитання. Зрозуміло, основним питанням є питання розміру податку, який доведеться сплатити в тому чи іншому випадку. Практично кожна людина стикається в житті з такими питаннями:

- а) отримання спадщини;
- б) покупка транспортного засобу;
- в) купівля нерухомого майна;
- г) відкриття власного бізнесу;
- д) сплата місцевих податків і т.ін.

Даний список є неповним, проте саме ці питання найбільш гостро постають в сучасному світі.

Розроблюєма онтологія повинна містити в собі розділи, відповідні податковим кодексом України, також необхідно визначити критерії, за якими визначається розмір податку, так як в кожному з розділів ці критерії будуть відрізнятися.

Завданням розроблюємої онтології є обчислення розміру податку для користувача, після вибору необхідного розділу, і введення значень для необхідних критеріїв.

3.7 Визначення критеріїв відповідних розміром податку

Для кожного з розділів податкового кодексу, який передбачає певний податок, існують певні критерії, в залежності від яких визначається розмір податку. Використовуючи різні значення для цих критеріїв на вході, користувач буде отримувати різне значення податку на виході.

У перерахованих нижче сферах розробляється онтологія дозволяє користувачеві дізнатися розмір передбачуваного податку, також представлені критерії для визначення розміру податку:

а) розмір збору за першу реєстрацію транспортного засобу:

- 1) тип транспорту;
- 2) тип двигуна;
- 3) обсяг циліндрів для легкових і вантажних автомобілів;
- 4) потужність двигуна для судна;
- 5) довжина корпусу для судна без двигуна;
- 6) спеціальний транспорт;

б) система єдиного податку (дозволяє визначити групу єдиного податку, до якої відноситься користувач):

- 1) вид платника податків;
- 2) рівень доходу;

в) податок на доходи фізичних осіб:

- 1) вид доходу;

г) розмір акцизного податку:

- 1) вид ставки податку;
- 2) опис товару;

- 3) тип продукції;
- д) розмір екологічного податку:
 - 1) тип забруднюючої речовини;
- е) рентна плата за транспортування нафти і нафтопродуктов:
 - 1) тип продукту, що транспортується;
- ж) розмір податку за виконання радіочастотного діапазону:
 - 1) вид радіозв'язку;
- з) розмір збору за спеціальне використання води:
 - 1) тип води;
- і) податок на нерухоме майно відмінне від земельної ділянки:
 - 1) вид платника податків;
 - 2) вид нерухомого майна.

Як видно з перерахованих вище критеріїв, деякі з них повторюються для різних сфер діяльності, однак більшою мірою критерії унікальні.

Виходячи з даних, що існують в податковому кодексі України, в розроблюваній онтології необхідно заповнити всі існуючі варіанти критеріїв. Таким чином, користувачеві необхідно буде лише обрати відповідний для його випадку варіант і отримати відповідь. Даний варіант системи є найбільш простим, так як звичайному користувачеві немає необхідності отримувати нові знання або володіти спеціальними навичками для використання програми, а досить лише виконати введення своїх параметрів для отримання відповіді.

3.8 Розробка онтології податкового кодексу України

Для початку представляємо структуру податкового кодексу у вигляді класів онтології, які в подальшому ми будемо використовувати. Отримана структура представлена на рисунку 3.1.

Таким чином, кожен клас відповідає певній сфері податкового кодексу, однак для більш зручного використання створимо підкласи, які будуть містити в собі параметри у вигляді критеріїв, а також розміри податків для даних сфер.

На рисунку 3.2 представлений приклад одного з класів з підкласами.

На рисунку 3.2 видно, що в розрахунку розміру збору за першу реєстрацію транспортного засобу беруться в розрахунок такі критерії як довжина корпусу для судна без двигуна, потужність двигуна для судна, об'єм циліндрів для легкових і вантажних автомобілів, спеціальний транспорт, тип двигуна, а також тип транспорту. Кожен з цих класів містить в собі екземпляри, що відповідають за ту чи іншу значення критерію.

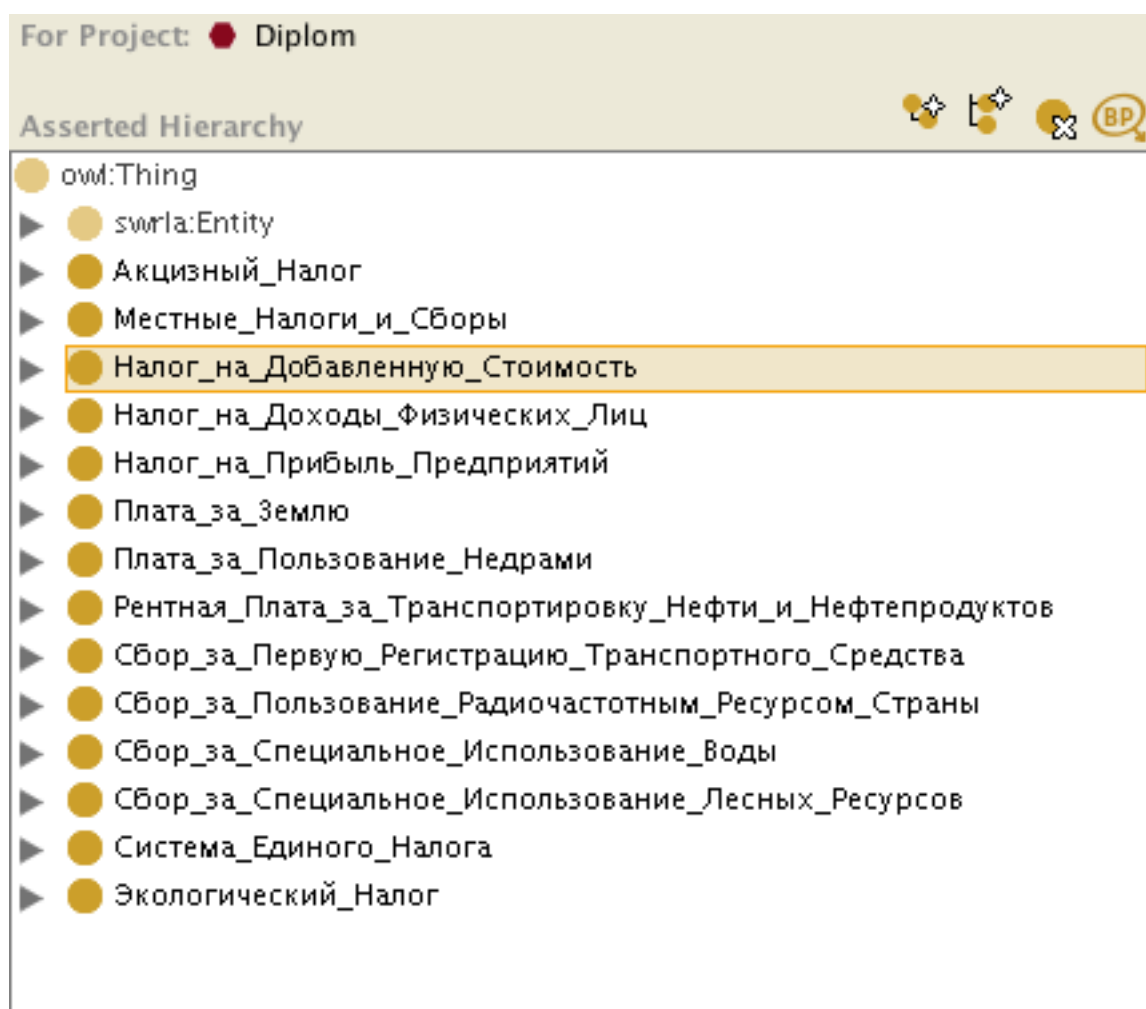


Рисунок 3.1 - Список класів онтології

- ▼ ● Сбор_за_Первую_Регистрацию_Транспортного_Средства
 - ▼ ● Параметры_Транспортного_Средства
 - Длина_Корпуса_Судна_без_Двигателя
 - Мощность_Двигателя_для_судна
 - Объем_Цилиндров_для_Легковых_и_Грузовых_Автомобилей
 - Специальный_Траспорт
 - Тип_Двигателя
 - Тип_Транспорта
 - Размер_Сбора_за_Регистрацию

Рисунок 3.2 - Приклад розроблених підкласів для однієї сфери

Також створено підклас, який зберігає в собі різні варіанти розміру збору за першу реєстрацію транспортного засобу, у вигляді примірників класу. Таким чином, виходячи із сукупності значень критеріїв, які введе користувач, буде обраний відповідний йому розмір збору.

Після розробки, існуюча онтологія містить в собі 73 класу, які передають структуру податкового кодексу України.

3.8.1 Створення властивостей онтології

Для зіставлення примірників онтології з класами створимо властивості онтології. Властивості будуть використовуватися для створення умов при визначенні розміру податку в тій чи іншій сфері. Таким чином, онтологія буде містити властивості для кожного з критеріїв.

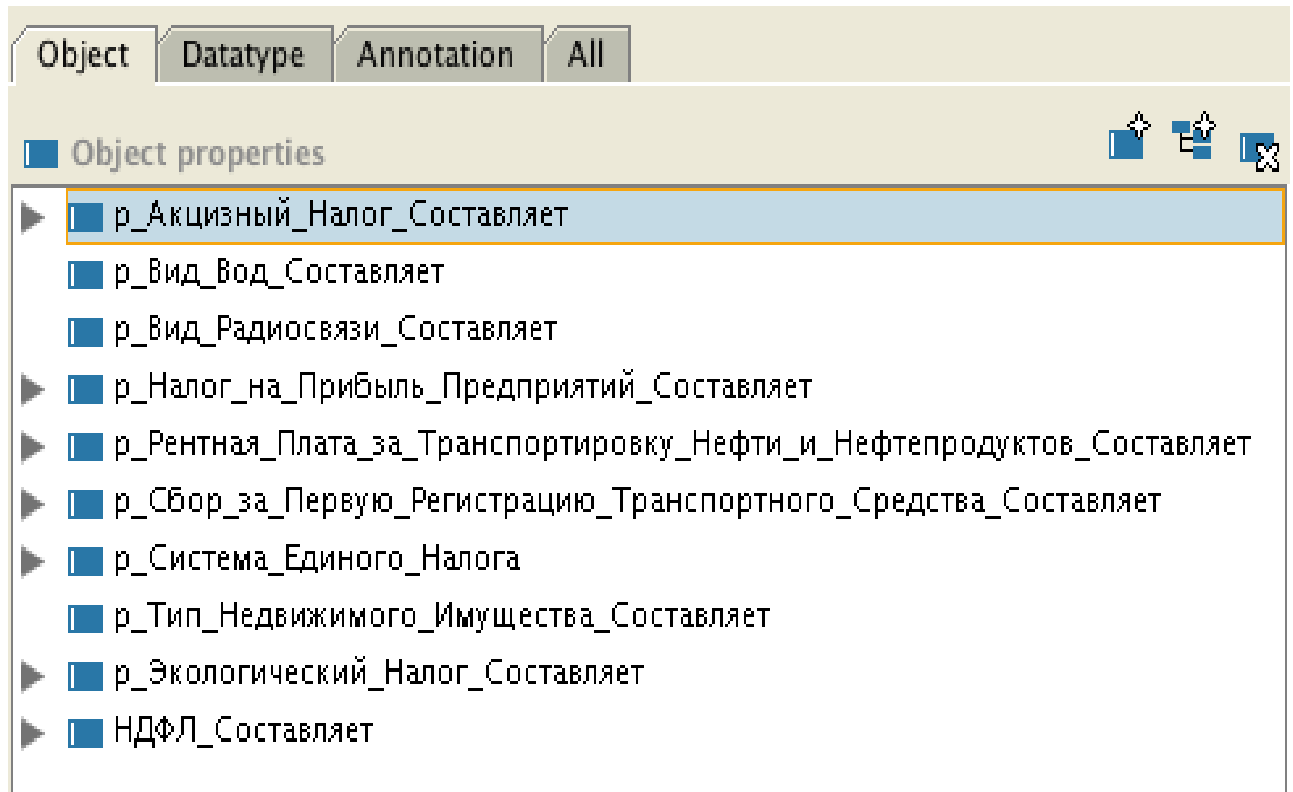


Рисунок 3.3 - Приклад створених властивостей онтології

За аналогією з класами, для зручності створюємо підвластивості, відповідні підкласам даної сфери. Приклад таких властивостей на прикладі збору за першу реєстрацію транспортного засобу представлений нижче:

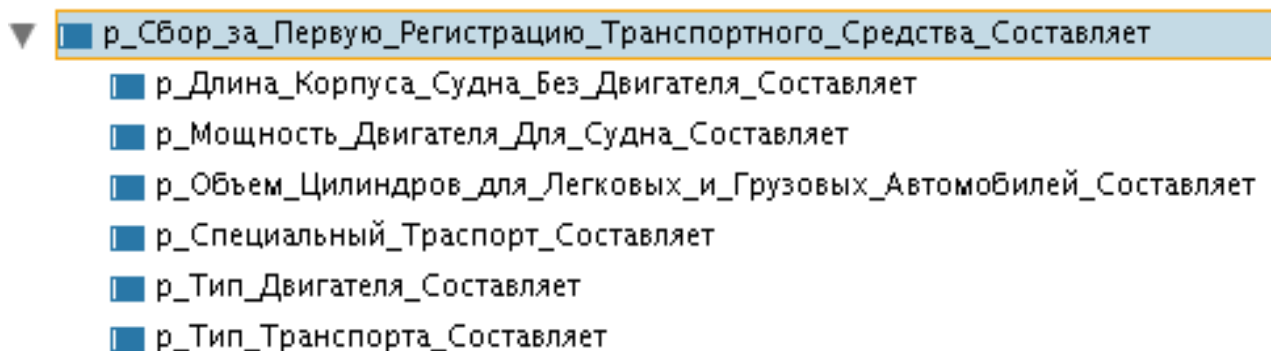


Рисунок 3.4 - Приклад розроблених подсвойств

3.8.2 Створення примірників онтології

Основне завдання онтології виконується за рахунок створених примірників. Примірники онтології умовно розділені на екземпляри критеріїв, і екземпляри податків.

Для кожного критерію в кожній сфері необхідно створити екземпляри. Кожен екземпляр буде відповідати одному існуючому варіанту, який користувач в подальшому може вибрати для заповнення критерію. Так, наприклад, для критерію тип транспорту необхідно створити такі екземпляри:

- а) легковий автомобіль;
- б) вантажний автомобіль;
- в) мотоцикл;
- г) судно;
- д) спеціальний транспорт.

При заповненні даних для визначення розміру збору, в графі тип транспорту користувачеві необхідно буде вибрати один із запропонованих варіантів. Такі ж дії потрібно повторити і для інших критеріїв даної сфери діяльності.

Крім примірників які відповідають критеріям, створені екземпляри, що відповідають розміру податку в тій чи іншій сфері. Саме дані екземпляри отримує користувач, після вибору сфери і введення необхідних критеріїв. Приклад створених примірників для обчислення розміру збору за першу реєстрацію транспортного засобу представлений нижче:

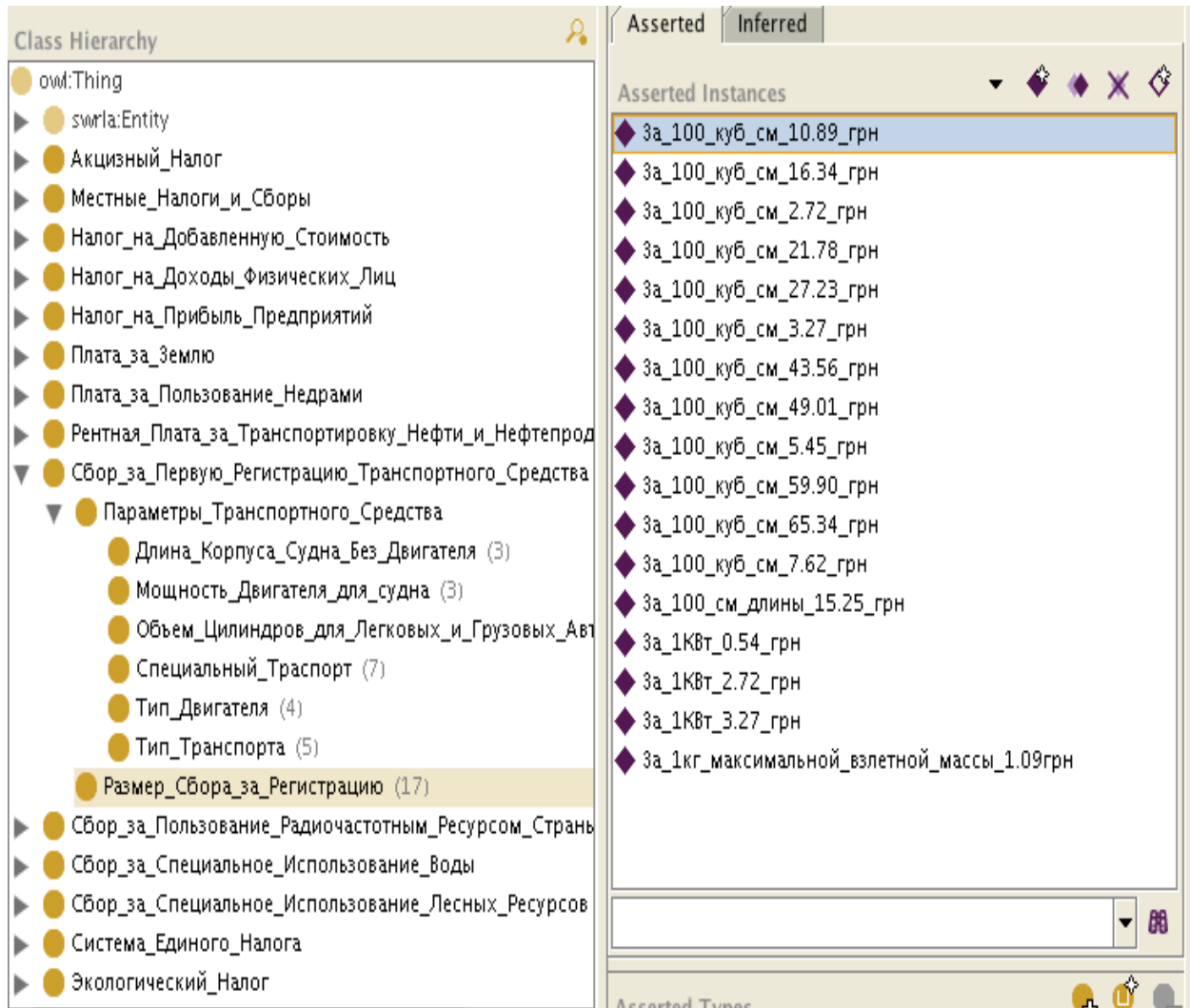


Рисунок 3.5 - Приклад створених примірників

Кожен з примірників зберігає умови свого виконання. Таким чином, кожен екземпляр може виконуватися, тільки при певному наборі виконаних умов.

The screenshot displays a software interface with two main panels. The left panel, titled 'Asserted Instances', contains a list of 20 items, each with a diamond icon and a text label. The item '3a_100_куб_см_21.78_грн' is highlighted in blue. The right panel shows a table with three columns: 'Property', 'Value', and 'Lang'. Below the table, there are three rows of property-value pairs, each with a diamond icon and a text label. The first row shows 'р_Длина_Корпуса_Су' with value 'Не_требуется'. The second row shows 'р_Объем_Цилиндров' with value 'от_8201_до_15000_куб_см'. The third row shows 'р_Тип_Двигателя_Со' with value 'Обычный'. The fourth row shows 'р_Мощность_Двигател' with value 'Не_Требуется'. The fifth row shows 'р_Специальный_Трас' with value 'Нет'. The sixth row shows 'р_Тип_Транспорта_С' with value 'Грузовой_автомобиль'.

Property	Value	Lang
rdfs:comment		
р_Длина_Корпуса_Су	Не_требуется	
р_Объем_Цилиндров	от_8201_до_15000_куб_см	
р_Тип_Двигателя_Со	Обычный	
р_Мощность_Двигател	Не_Требуется	
р_Специальный_Трас	Нет	
р_Тип_Транспорта_С	Грузовой_автомобиль	

Рисунок 3.6 - Пример условий выполнения примерника

Таким чином, можна визначити, що за умови, що типом транспорту є вантажний автомобіль зі звичайним типом двигуна, з об'ємом циліндрів від 8201 до 15000 куб.см., при цьому значення для спеціального транспорту, потужності двигуна для суден, і довжини корпусу для судів нам не потрібні, так як ми маємо справу з вантажним автомобілем, збір за першу реєстрацію транспортного засобу становитиме 21,78 гривень за кожні 100 куб.см. об'єму циліндрів.

За аналогічним принципом визначаються і інші розміри зборів і податків, описаних в розробленій онтології.

3.9 Створення запитів

Для можливості використання онтології звичайним користувачем потрібно створити макет запитів, які дозволять знаходити рішення задачі.

Для створення запитів використовуємо спеціальну вкладку Query Tab в Protégé.

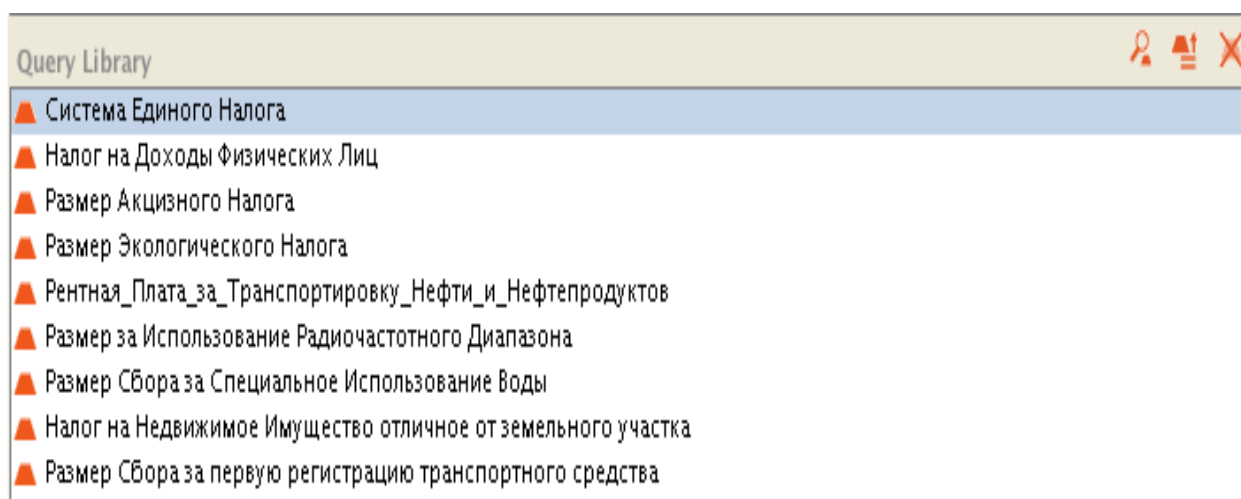


Рисунок 3.7 - Список створених запитів

Кожен запит дозволяє користувачеві дізнатися розмір податку або збору, в тій чи іншій сфері отримання прибутку.

Для використання онтології звичайним користувачем, і введення параметрів для розрахунку розміру податку, користувачеві необхідно відкрити розроблену онтологію за допомогою Protégé 3.4.8 і перейти на вкладку Queries Tab.

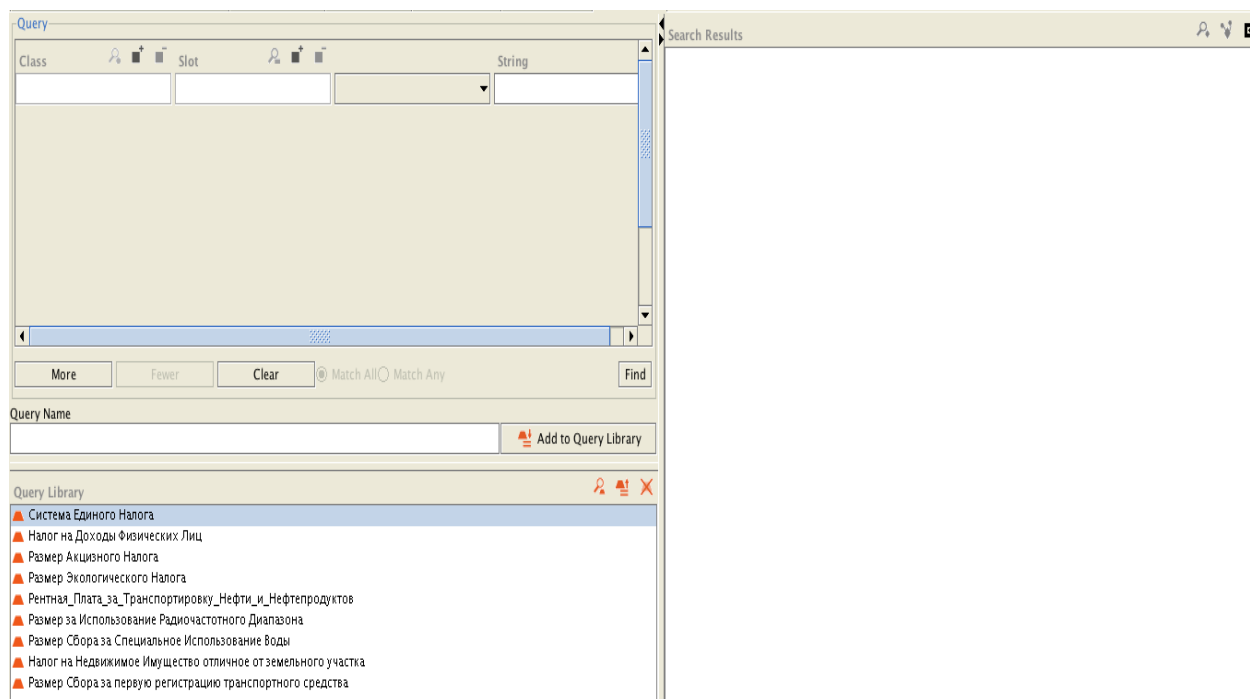


Рисунок 3.8 - Загальний вигляд вкладки Queries Tab

Після переходу на необхідну вкладку в програмі користувачеві необхідно вибрати сферу, розмір податку в якій цікавить користувача. Доступний список представлений в нижньому лівому кутку. Після вибору цікавить сфери необхідно один раз натиснути лівою кнопкою миші по рядку із запитом, відповідним даній сфері, і натиснути кнопку Retrieve Query

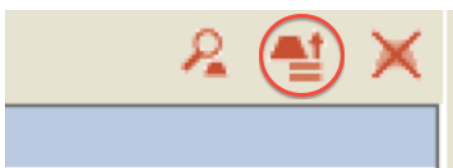


Рисунок 3.9 - Кнопка Retrieve Query на панелі управління

Після вибору запиту і натискання кнопки, з верхнього вікна з'явиться список критеріїв, а також поля для вибору значень для даних критеріїв.

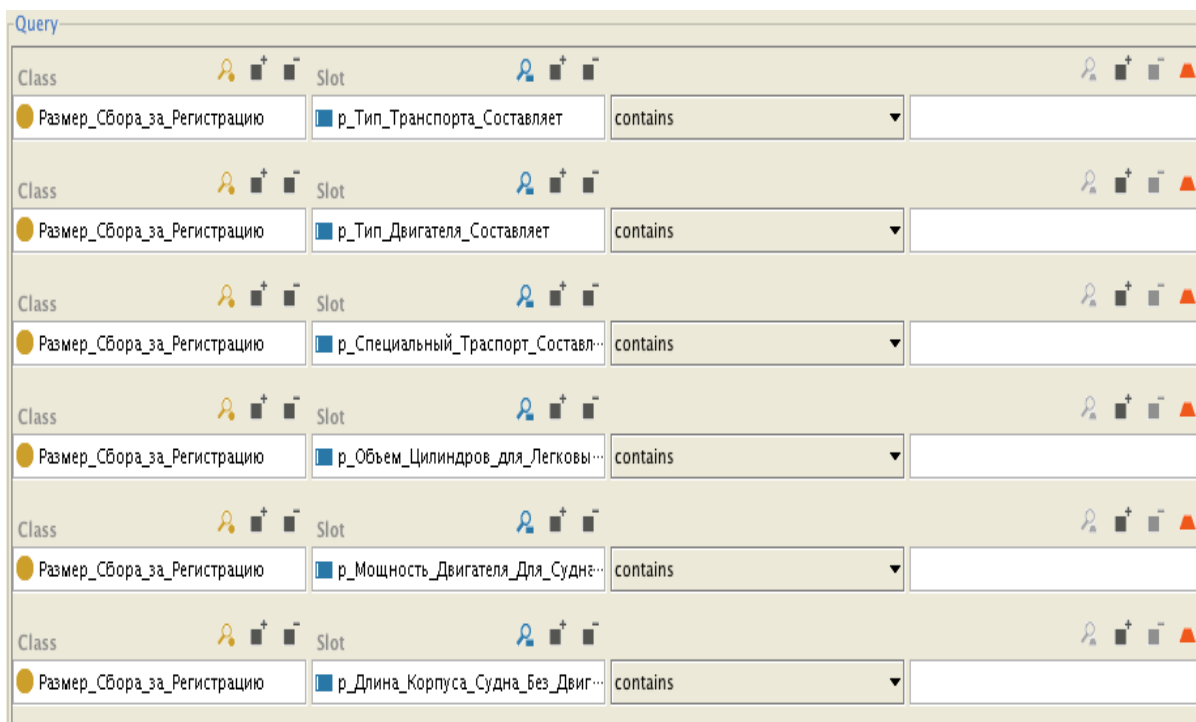


Рисунок 3.10 - Список критеріїв і полів на прикладі одного із запитів

У полях Slots представлені назви критеріїв, необхідних для заповнення. У правій частині вікна знаходяться порожні поля, для їх заповнення необхідно над кожним з них натиснути кнопку Select Instance і вибрати відповідні значення для відповідного критерію.

Після натискання кнопки для вибору критерію «Тип Транспорту», з'являється додаткове вікно, щоб вибрати налаштування.

У відповідному обраній сфері розділу, користувачеві необхідно вибрати необхідний критерій, і натиснути кнопку ОК.

Після вибору значення для критерію, вибране значення з'явиться в полі. За аналогією потрібно заповнити значення для всіх критеріїв.

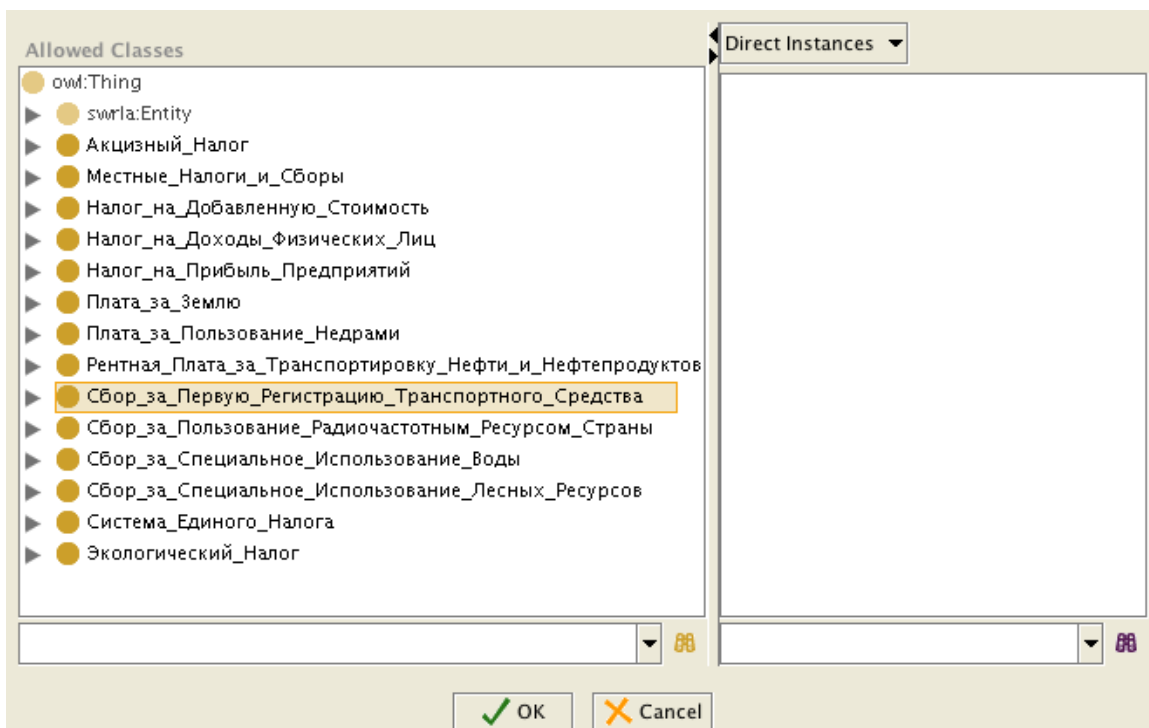


Рисунок 3.11 - Додаткове вікно Select Instance

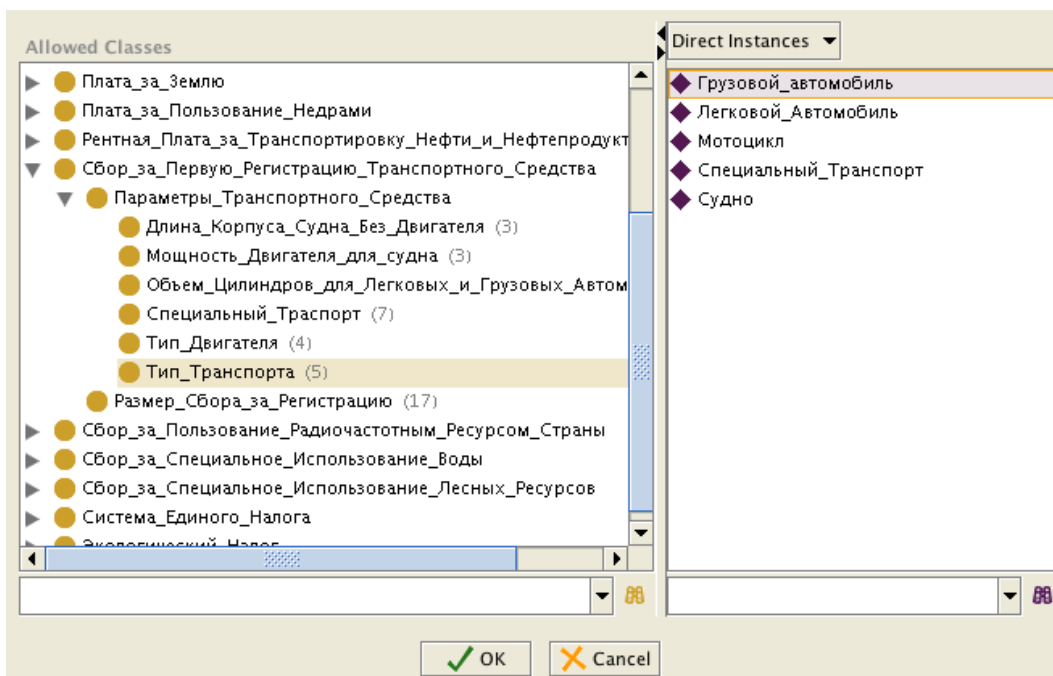


Рисунок 3.12 - Вибір параметра для критерію

Class	Slot	Relationship	Value
Размер_Сбора_за_Регистрац...	р_Тип_Транспорта_Составл...	contains	Грузовой_автомобиль
Размер_Сбора_за_Регистрац...	р_Тип_Двигателя_Составляет	contains	Обычный
Размер_Сбора_за_Регистрац...	р_Специальный_Траспорт_С...	contains	Нет
Размер_Сбора_за_Регистрац...	р_Объем_Цилиндров_для_Ле...	contains	от_8201_до_15000_куб_см
Размер_Сбора_за_Регистрац...	р_Мощность_Двигателя_Для...	contains	Не_Требуется
Размер_Сбора_за_Регистрац...	р_Длина_Корпуса_Судна_Бе...	contains	Не_требуется

More Fewer Clear Match All Match Any Find

Рисунок 3.13 - Приклад заповнених полів для обчислення розміру податку

Для правильного знаходження розміру податку необхідно заповнювати всі наявні критерії. Деякі поля можуть залишитися порожніми, проте в такому випадку програма видасть варіанти, відповідні тільки до введених параметрів.

Після введення всіх значень необхідно натиснути кнопку Find в нижній правій частині вікна і програма видасть розмір податку, відповідно до введених параметрами.

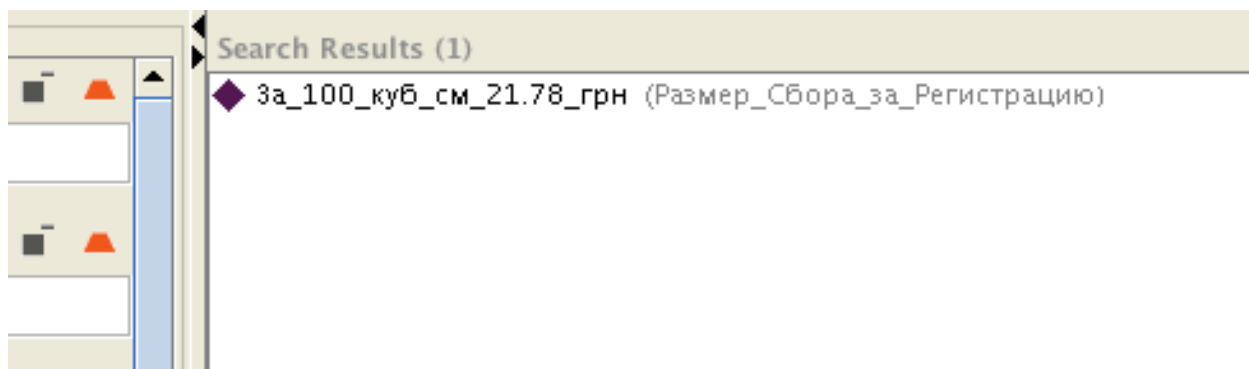


Рисунок 3.14 - Висновок результату

Після отримання результату, користувач має можливість замінити необхідні параметри критеріїв, і після натискання кнопки Find, отримати нову відповідь. Так само, є можливість вибрати інший запит, для цього варто повторити весь алгоритм дій з самого початку.

Таким чином, в магістерській роботі досліджена галузь онтологічного моделювання, розроблена онтологія прикладної предметної галузі на прикладі системи оподаткування України. До розробленої онтології за допомогою стандартних засобів і інструментів розробки створені запити для подальшого використання. Провівши дослідження в сфері оподаткування України, можна з упевненістю сказати, що основний елемент даної системи у вигляді податкового кодексу України, є складною структурою, практично недоступною для використання простими користувачами, без спеціальних можливостей або знань.

Розроблена онтологія дозволяє простим користувачам не тільки використовувати знання галузі, в якій вони не є фахівцями, але і знаходити відповіді на питання, з якими рано чи пізно доводиться стикатися кожній людині.

Так само розроблена онтологія має позитивні фінансові переваги, так як для її використання необхідний продукт, який надається з безоплатної ліцензії, при цьому користувач може безоплатно дізнатись відповідь, коли в

іншому випадку йому необхідно звертатися за консультацією к приватним особам або фірмам, чиї послуги надаються на фінансовій основі.

ВИСНОВКИ

Дослідивши сучасну систему оподаткування України, можна з упевненістю сказати, що основний елемент даної системи у вигляді податкового кодексу України є складною структурою, практично недоступною для використання простими користувачами без спеціальних можливостей або знань. У той же час, у кожної людини в житті настає такий момент, коли дані знання необхідні і доводиться користуватися додатковими послугами юристів або агенцій. Зрозуміло, що не кожен може собі це дозволити. В такому випадку розроблена онтологія допоможе дізнатися необхідний розмір податку в найбільш затребуваних сферах діяльності, при цьому заощадивши кошти.

Основними важливими галузями в житті людей є такі сфери як покупка транспортних засобів, отримання і покупка нерухомого майна, виграш призів, отримання спадщини тощо. Розроблена онтологія дозволяє користувачеві дізнатися необхідний розмір податку для сплати в різних сферах діяльності не вдаючись до послуг сторонніх осіб.

Так само розроблена онтологія має переваги, так як для її використання необхідний продукт, який надається з безоплатної ліцензії, при цьому користувач не платить за користування онтологією.

В ході атестаційної роботи визначені розміри різних податків для різних сфер діяльності, а також при різних умовах. У онтологію закладені всі критерії для правильного визначення розміру податку. Користувач повинен лише зробити введення необхідних даних для всіх параметрів і застосування підбере відповідну даним параметрам відповідь.

ПЕРЕЛІК ПОСИЛАНЬ

1. <http://www.aiportal.ru/articles/other/ontology.html> - Онтологія (дата звернення 15.04.2020).
2. <http://sergeiseleznev.blogspot.com/2010/07/1.html> - Онтології в корпоративних системах частина (дата звернення 13.04.2020).
3. Boose J.H. Knowledge Acquisition Tools, Methods, and Mediating Representations // In Motoda H., Mizoguchi R., Boose J., Gaines B. (Eds.) Knowledge Acquisition for Knowledge-Based Systems. IOS Press, Ohinsha Ltd. – Tokyo, 1990.
4. Анисимов А.В. Система обработки текстов на естественном языке /А.В. Анисимов, А.А. Марченко // Научно-теоретический журнал “Искусственный интеллект”, ИПШ “Наука і освіта”. – 2002. – Вип. 4 – С. 157–163..
5. Литвин В.В. Технології менеджменту знань. – Львів.: Львівська політехніка, 2010, 261 с.
6. Пасічник В.В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко. – К.: ВНУ “ПИТЕР”, 2006. – 460с
7. Гаврилова Т.А. Базы знаний интеллектуальных систем/ Т.А. Гаврилова, В.Ф. Хорошевский. –СПб.: Питер, 2001. – 384 с.
8. <http://www.w3.org/TR/2004/REC-owl-guide-20040210> - OWL Web Ontology Language guide W3C recommendation (дата звернення 15.04.2020).
9. http://profmeter.com.ua/communication/learning/course/course21/lesson280/?LESSON_PATH=470.559.280 - Структура податкового законодавства України (дата звернення 15.04.2020).
10. http://download.yandex.ru/class/solovyev/lecture06_2.pdf - Мови уявлення онтологій (дата звернення 10.04.2020).

11. <http://www.interface.ru/home.asp?artId=33246> - Огляд інструментів інженерії онтологій (дата звернення 08.04.2020).
12. O. Zolotukhin, M. Kudryavtseva. Authentication Method in Contactless Payment Systems International Scientific and Practical Conference «Problems of Infocommunications. Science and Technology», 9–12 October, 2018, Kharkiv, Ukraine, pp. 397-400.
13. Джим Арло, Айла Нейштадт UML 2 і Уніфікований процес. Практичний об'єктно-орієнтований аналіз і проектування. - СПб .: Символ-Плюс, 2007. - 624 с.
14. V. Filatov, A. Yerokhin, O. Zolotukhin, M. Kudryavtseva Personalized Adaptation of Learning Environments. CAOL*2019, DSMOLE*2019, UM*2019, 4-6 September, 2019, Bulgaria, pp. 584-588
15. Леоненков А.В. Об'єктно-орієнтований аналіз та проектування з використанням UML і IBM Rational Rose. - М .: Біном. Лабораторія знань, 2006. - 320 с.
16. V. Semenets, V. Filatov, O. Zolotukhin Synthesis of Semantic Model of Subject Area at Integration of Relational Databases CAOL*2019, DSMOLE*2019, UM*2019, 4-6 September, 2019, Bulgaria, pp. 598-602
17. MULTILAYER ADAPTIVE FUZZY PROBABILISTIC NEURAL NETWORK IN CLASSIFICATION PROBLEMS OF TEXT DOCUMENTS. Автор: Bodyanskiy, E. V.; Ryabova, N. V.; Zolotukhin, O. V. RADIO ELECTRONICS COMPUTER SCIENCE CONTROL Том: 1 Стр.: 39-45
Опубликовано: 2015
18. Пол Кімел UML. Основи візуального аналізу і проектування. М .: НТ Пресс, 2008. - 272 с.