Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____комп'ютерних наук_____
(повна назва)

Кафедра _____Програмної інженерії_____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____другий (магістерський)_____
(рівень вищої освіти)

Дослідження онтологічних моделей тексту для аналізу технічної документації
(тема)

Виконав: студент 2 курсу, групи ІПЗм-18-2__
спеціальності _121- Інженерія програмного забезпечення_____
(код і повна назва спеціальності)
освітньо-наукової програми _Інженерія програмного_забезпечення_ _____
(повна назва освітньої програми)

_____Вавілов П.О._____
(прізвище, ініціали)

Керівник _____проф. Лєсна Н.С._____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____ З.В.Дудар

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Освітньо-наукова програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Вавілову Павлу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження онтологічних моделей тексту для аналізу технічної документації

затверджена наказом по університету від « » 2020 р № __

2. Термін подання студентом роботи до екзаменаційної комісії «11» травня 2020 р.

3. Вихідні дані до роботи *Алгоритми обробки текстової інформації, алгоритми захисту даних, методи розробки онтологій, пояснювальна записка. Використовувати ОС Windows, середовище об'єктно-орієнтованого проектування.*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі і постановка задачі, методи пошуку корисних даних, опис об'єктних моделей, використовувані. методи та алгоритми, архітектура програмної системи, опис розробленої програмної системи. , результати тестування програмної системи*

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Лєсна Н.С.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	25 березня 2020 р.	
2.	Огляд існуючих методів	31 березня 2020 р.	
3.	Методи онтологій та обробки текстів	15 квітня 2020 р.	
4.	Підготовка пояснювальної записки	20 квітня 2020 р.	
5.	Спецчастина	28 квітня 2020 р.	
6.	Підготовка презентації та доповіді	03 травня 2020 р.	
7.	Попередній захист	05 травня 2020 р.	
8.	Нормоконтроль, рецензування	07 травня 2020 р.	
9.	Занесення диплома в електронний архів	08 травня 2020 р.	
10.	Допуск до захисту у зав. кафедри	10 травня 2020 р.	

Дата видачі завдання _ «____»_ _____2020 р.

Студент _____
(підпис)

Керівник роботи _____ __проф. Лєсна Н.С._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 106 с., 9 табл., 43 рис., 3 дод., 27 джерел.

ОНТОЛОГІЯ, НОРМАТИВНИЙ ПРОФІЛЬ, ОНТОЛОГІЧНА МОДЕЛЬ, ОНТОЛОГІЧНИЙ ЗАПИТ, ЕКСПЕРТНА СИСТЕМА, НЕЧІТКА АТРИБУТНА ГРАМАТИКА.

Об'єкт дослідження – процес семантичного аналізу тексту технічної документації при сертифікації програмних систем.

Мета – синтезувати онтологічну модель тексту для автоматизації семантичного аналізу технічної документації.

Метод дослідження – математичний апарат теорії автоматів, нечітких множин, методи структурної та прикладної лінгвістики, методи представлення знань в штучному інтелекті.

В результаті наведено основні методи побудови онтологічних моделей, які використовувалися при побудові даної моделі. Спроектована онтологічна модель тексту технічної документації та її програмна модель.

ONTOLOGY, NORMATIVE PROFILE, ONTOLOGICAL MODEL, ONTOLOGICAL INQUIRY, EXPERT SYSTEM, FUZZY ATTRIBUTE GRAMMER.

The object of study is the process of semantic analysis of the text of technical documentation during certification of software systems.

The purpose is to synthesize an ontological model of text to automate the semantic analysis of technical documentation,

As a result, the basic methods of ontological model construction were used, which were used in the construction of this model. The ontological model of the text of the technical documentation and its program model is designed.

ЗМІСТ

ВСТУП

Останнім часом спостерігається тенденція до збільшення числа проектів, що використовують онтології як формальної основи для розробки структури інформаційних ресурсів, систем пошуку. Зокрема, онтології знаходять широке застосування в масштабних системах, де експлуатаційні властивості програмного забезпечення (ПЗ) в значній мірі визначаються якістю ПЗ.

У зв'язку з цим при розробці онтологій особливої важливості набуває організація ефективного оцінювання ПЗ, проведеного сертифікаційними центрами, до функцій яких входить, в тому числі і залучення кваліфікованих фахівців, здатних швидко і достовірно оцінити ПЗ. При цьому якість самої оцінки визначається кваліфікацією експертів і наявних в їх розпорядженні інформаційних ресурсів – бази нормативних документів.

Таким чином, для створення експертної нормативної бази документів, може бути використаний онтологічний підхід.

Застосування такого підходу для оцінки якості ПЗ [1] передбачає вирішення низки завдань, серед яких слід виділити:

– формування нормативного профілю (НП) – гармонізованої з міжнародними та національними стандартами сукупності вимог, що пред'являються до даного проекту або групі проектів [2]. НП можуть бути знову розробляються державні або галузеві стандарти, нормативно-методичні документи підприємств і загальне вимоги специфікацій ПЗ;

– реінжиніринг процесу проектування ПЗ і його оцінка на основі НП;

– динамічний аналіз ПЗ і інтервальний аналіз виконуваного модуля.

У свою чергу онтології володіють власними засобами обробки (логічного висновку), відповідними засобами семантичної обробки інформації, які можуть бути покладені в основу експертної системи, яка забезпечить підтримку експерта при роботі з нормативною базою. Така система дозволить, шляхом накопичення та узагальнення досвіду оцінок в базі знань експертних систем, знизити

суб'єктивність і підвищити ефективність прийнятих рішень за рахунок врахування великої кількості факторів, що визначають властивості аналізованого проекту [7].

Так, завдяки онтології, при зверненні до пошукової системи користувач зможе отримувати у відповідь ресурси, семантично релевантні запиту, завдяки цьому онтології набули широкого поширення в рішенні проблем подання знань та інженерії знань, інформаційного пошуку і т.д. [5].

Варто відзначити, що онтологічні системи є інтелектуальними засобами для пошуку, новими методами подання та обробки знань і запитів [6]. Вони здатні точно і ефективно описувати семантику даних для деякої предметної області і вирішувати проблему несумісності і протилежності понять.

З огляду на зазначені обставини, при розробці онтологій особливої важливості набуває організація ефективного оцінювання ПЗ, проведеного сертифікаційними центрами, до функцій яких входить, в тому числі і залучення кваліфікованих фахівців, здатних швидко і достовірно оцінити ПЗ.

Проектування ПЗ – це процес складання опису, його перетворення, визначення архітектури, компонентів, інтерфейсів, інших характеристик системи і кінцевого складу програмного продукту, результатом якого є проектне рішення, яке задовольняє заданим вимогам. Проектування являє собою трудомісткий процес, що вимагає від користувача глибокого знання предметної області та навичок в проектуванні.

У зв'язку з цим вельми актуальна тема магістерської роботи, а саме: використання онтологій для формування нормативного профілю НП, при сертифікації програмних систем (ПС), це дає можливість отримання моделі декларативних знань у вигляді класів зі ставленням ієрархії між ними, що допускає повторне використання.

Наразі проектування включає розробку вимог або технічної документації (ТД), розробку системи або технічного проекту, програмування або робоче проектування, пробну експлуатацію, супровід і поліпшення. Необхідно враховувати взаємозалежність всіх основних частин процесу проектування від інструментарію, технологій і організації робіт.

Найбільш відомі з комерційних програмних продуктів, що використовуються для проектування програмного забезпечення, призначені для візуалізації проміжних і кінцевих результатів процесу проектування. Деякі з них дозволяють повністю автоматизувати останні етапи проектування: генерація коду, створення звітної і супроводжуючої документації тощо При цьому завдання автоматизації семантичного аналізу технічної документації – залишається відкритою.

Це пов'язано з надзвичайною складністю проблеми синтезу та аналізу семантики технічного тексту, для вирішення якої необхідно використовувати симбіоз методів штучного інтелекту, прикладної лінгвістики, психології і т.п. [8].

Таким чином, актуальною є задача автоматизації процесу сертифікації ПС аудитором, головним аспектом якої є автоматизація семантичного аналізу технічної документації.

Основна мета полягає у вирішенні завдань, пов'язаних з розробкою технологій формування нормативного профілю для сертифікації програмних систем і інтелектуальної підтримки прийняття рішень аудитором сертифікаційного центру при реалізації основного завдання експертизи програмного забезпечення – формування нормативного профілю.

Розроблена методика семантичного аналізу тексту ТД шляхом використання онтологічного підходу та онтологічна модель аналізу тексту для автоматизації семантичного аналізу технічної документації, що дозволяє спростити процес сертифікації ПЗ

В результаті розробки і впровадження пропонованої моделі підвищується якість проектування за рахунок автоматизації рутинної праці людини.

1 ОГЛЯД І АНАЛІЗ ПРОБЛЕМ, ПОВ'ЯЗАНИХ З ВИКОРИСТАННЯМ ОНТОЛОГІЧНИХ МОДЕЛЕЙ

1.1 Загальна характеристика предметної області

В останні роки почали розвиватися методи автоматичної побудови онтології, що базуються на лінгвістичному аналізі тексту [2]. Розвиток цих методів стало можливим в силу наявності значного досвіду, накопиченого в області природно-мовної обробки тексту. Звичайно, поява таких методів дозволить значно прискорити процес створення онтології.

Зазвичай методи автоматичного вилучення знань з тексту розробляються в основному як програми традиційного типу, засновані на жорстких алгоритмах. В інтелектуальних завданнях повинна існувати можливість на кожному кроці оцінювати ситуацію і робити відповідні дії. Таку можливість забезпечують правила.

У зв'язку з цим в роботі методи лінгвістичного аналізу, включаючи і методи здобування знань, мають декларативне подання до вигляді систем продукційних правил. Застосування продукційних правил забезпечує наступні переваги: простоту і модульність, зручність модифікації, ясність, прозорість, можливість поступового нарощування, високий ступінь спільності правил обробки даних. Такий підхід до подання методів визначив технологію їх створення.

Розвиток індустрії систем електронного документообігу, що супроводжується зростанням масивів оброблюваних повнотекстових документів, вимагає нових засобів організації доступу до інформації, багато з яких слід віднести до розряду систем штучного інтелекту – систем обробки знань.

Одним з ефективних підходів до виявлення та обробці сенсу текстових документів є використання онтологій.

Онтологія – це інформаційна структура, що припускає повторне використання і містить комплекс понять, від найзагальніших до найбільш конкретних, що охоплює повний спектр об'єктів і відносин, включаючи події і

процеси, а також значення (атрибутів і відносин), що визначаються, якщо необхідно, в часі і просторі . Онтологія визначає терміни, використовувані для опису і представлення знань тієї чи іншої предметної області [3]. Онтологічна модель знань нормативної бази програмної інженерії – модель предметної області, яка використовує всі доступні засоби представлення знань, релевантні для проведення аналізу галузі формування нормативного профілю.

База, що образує профіль (СТ) – множина, що складається з одного або декількох стандартів, що служить основою для формування нормативних профілів.

Технічна документація (ТД) – сукупність кількох нормативних документів, що використовуються при проектуванні, сертифікації.

Онтології включають доступні для комп'ютерної обробки визначення основних понять предметної області і зв'язки між ними [4].

Автоматичне вилучення знань з монологічних текстів з метою побудови онтології передбачає не тільки виявлення термінів, але і витяг знань про терміни. Це означає, що для опису семантичної структури термінології необхідно розпізнати в тексті як терміни, так і семантичні відносини між термінами. Але перш ніж розглядати типи семантичних відносин, необхідно вибрати підхід, на основі якого слід виконувати аналіз пропозиції монологічного тексту.

Профілювання в загальному випадку включає формування нормативного профілю – НП (при розробці ПЗ або стандартів для нього) і верифікацію НП (при експертизі) [2, 3].

В ході формування НП повинен бути проведений синтез (на основі національної та міжнародної нормативної бази, що включає стандарти ДСТУ, ЕС38, IEEE, IAEA ін.) бази за допомогою методів семантичного аналізу ТД [1].

Процес формування НП повинен включати в себе основні принципи стандартизації [5]:

– доцільність розробки НП визначається шляхом аналізу його технічної документації;

– пріоритетним напрямком при формуванні НП є безпека об'єкта стандартизації для людини і навколишнього середовища;

– НП не повинні бути технічним бар'єром. Для цього необхідно враховувати міжнародні стандарти, правила, норми міжнародних організацій і стандарти інших країн;

– розробка НП повинна бути також проаналізовано експертом;

– при розробці НП повинні дотримуватися: норми законодавства, правила в області державного нагляду і контролю; взаємопов'язаність об'єктів стандартизації з іншими об'єктами стандартизації і т.д .;

– обов'язкові вимоги до НП повинні бути перевірені і придатні для цілей сертифікації відповідності;

– стандарти, що входять до НП, що застосовуються при сертифікації ПП, не повинні дублювати один одного.

Процес формування НП включає наступні етапи [6]:

– відбір і систематизація нормативних документів, формування на їх основі загальної бази, що образує профіль;

– розробка на основі загальної бази номенклатури приватних баз, виходячи з особливостей розробки і застосування систем з інтенсивним використанням ПЗ, формування множини приватних баз, що образують профіль;

– формування глосаріїв базових термінів для кожної приватної бази;

– розробка за допомогою глосаріїв базових термінів множин приватних баз, що образують профіль стандартів-глосаріїв для загальних баз, що образують профіль, формування профілів термінів;

– створення на основі профілів термінів нормативних профілів для кожної приватної бази.

Реалізація перерахованих етапів має ряд проблем, які потрібно вирішити для створення експертної системи, побудованої за допомогою онтологічного підходу [7]. Дані проблеми пов'язані з множиною профілеобразуючих організацій за критеріями, враховуючи їх досвід, міжнародний статус, обсяг і номенклатуру випущених стандартів і т.д., формуванням і ранжування стандартів з яких

складається СТ, відбір стандартів з ПЗБ, відповідних технічної документації, які і ляжуть в основу формування нормативного профілю на ПП [8].

Також однією з проблем є побудова ієрархії стандартів, тобто перехід від міжнародних, до національних, галузевих стандартів і т.д., і організація пошуку по ним стандартів підходять під технічну документацію.

Перераховані вище проблеми відносяться до розряду таких, що складно формалізувати, комплексне їх рішення утруднено через відсутність ефективних аналітичних методів [2].

Виходячи з цього, доцільно в процесі формування та верифікації НП використовувати інтелектуальні методи.

Разом з тим, ряд приватних завдань може бути успішно вирішений з використанням аналітичних методів і традиційної обробки даних.

До цих завдань відносяться, в першу чергу вибір найбільш представницьких нормативних документів, формування таблиць вимог з однієї груп для цих документів і т.д. [9].

Таким чином, комп'ютерна підтримка експерта при формуванні нормативного профілю, може бути організована на основі інтегрованої фреймової системи, яка поєднувала в собі традиційні методи і засоби обробки даних з можливостями знання орієнтованих методів, побудованих із застосуванням онтологічного підходу.

Під НП розуміється сукупність кількох нормативних документів з чітко визначеними і гармонізованими підмножинами обов'язкових і факультативних можливостей, призначених для реалізації заданих функцій. Експертна система формування НП складається з наступних аспектів характерних для систем штучного інтелекту: виявлення знань, подання знань, маніпулювання знаннями, способу реалізації виведення на знаннях з бази [2], яка являє собою множина, що складається з одного або декількох стандартів, що є основою для формування нормативних профілів. У свою чергу профіль – це один або кілька базових стандартів, включаючи міжнародні функціональні стандарти [10].

При вирішенні поставленого завдання виділяються кілька фаз, які необхідно пройти для досягнення поставленої мети рис. 1.1, починаючи від аналізу вихідних даних, підготовки їх до машинної обробці, порівняно ТД з ПЗБ і отримання результатів, які повинен перевірити експерт сертифікаційної організації.

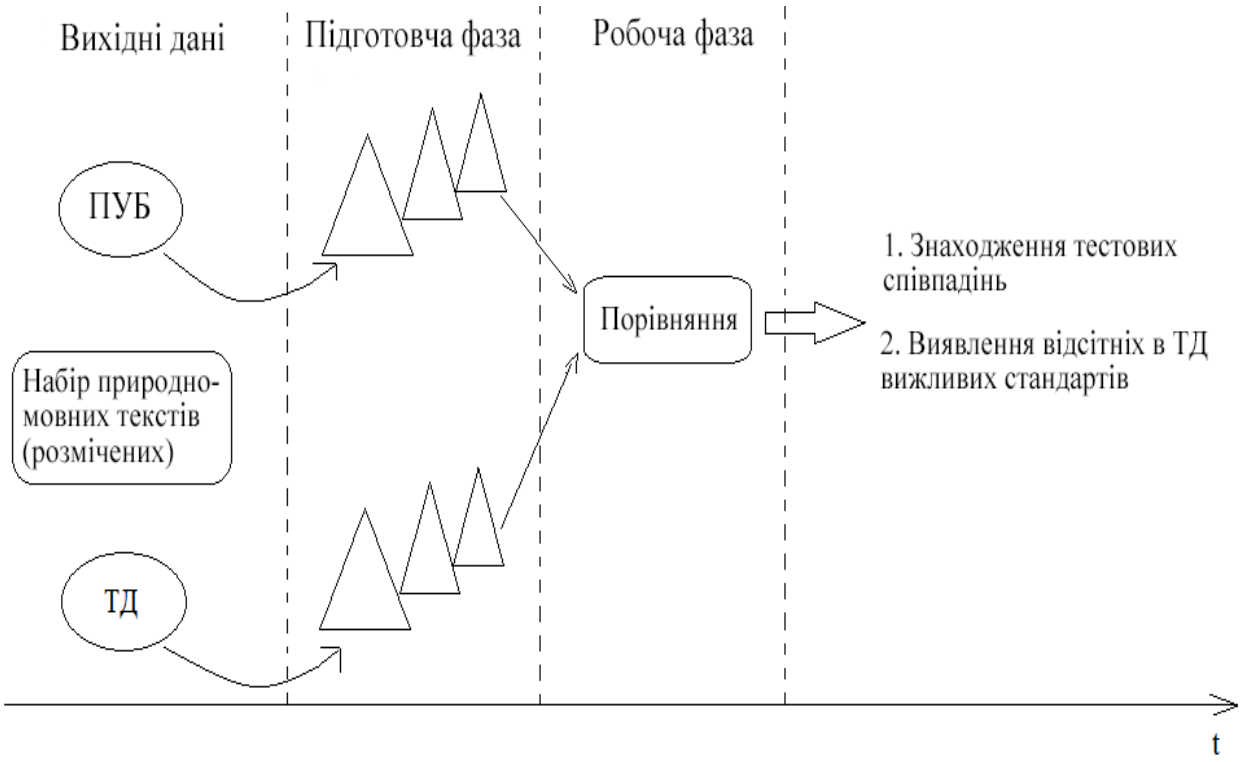

Рисунок 1.1 – Фази проходження сертифікації

Виявлення знань ґрунтується на комплексних рішеннях, традиційних, аналітичних, інтелектуальних методах, а отримана інформація може надаватися у вигляді фреймів [11], оскільки СТ має чітко виражену ієрархічну структуру. На різних рівнях надається інформація СТ і НП стереотипна [1], так як і НП має ієрархічну структуру, представлену у вигляді витягів з стандартів підходять під технічну документацію на ПП. Механізм активізації фреймів, приєднаних процедур і різних слотів представляється у вигляді мета-правил.

1.2 Сучасні CASE-засоби

Онтології – більше ніж просто складний підхід до опису та класифікації інформації. Вони можуть використовуватися для підтримки функціонування і зростання нового виду цифрових бібліотек, реалізованих як розподілені інтелектуальні системи.

Під онтологією можна розуміти [5]:

- надійний семантичний базис у визначенні змісту;
- загальну логічну теорію, яка складається зі словника і набору тверджень на деякій мові логіки;
- основу для комунікації між людьми і комп'ютерними агентами.

Онтології дозволяють представити нові поняття так, що вони стають придатними для машинної обробки. За допомогою онтології можна "побудувати зв'язку" між новими поняттями, з якими система ще не зустрічалася, і описами вже відомих класів, відносин, властивостей і об'єктів реального світу.

Компоненти, з яких складаються онтології, залежать від парадигми уявлення. Але практично всі моделі онтологій в тій чи іншій мірі містять концепти (поняття, класи, сутності, категорії), властивості концептів (слоти, атрибути, ролі), відносини між концептами (зв'язку, залежності, функції) та додаткові обмеження (визначаються аксіомами).

Термін екземпляр використовуються для представлення елементів в предметної області, тобто елемента даного концепту. Онтологія разом з множиною окремих екземплярів становить базу знань.

Сучасні CASE-засоби охоплюють велику область підтримки численних технологій проектування ІС: від простих засобів аналізу і документування до повномасштабних засобів автоматизації, що показують весь життєвий цикл ПЗ [6]. Повний комплекс CASE-засобів, що забезпечує підтримку життєвого циклу ПЗ, містить наступні компоненти:

– репозиторій, що є основою CASE-засобу. Він повинен забезпечувати зберігання версій проекту і його окремих компонентів, синхронізацію надходження інформації від різних розробників при груповий розробці, контроль метаданих на повноту і несуперечність;

– графічні засоби аналізу і проектування, що забезпечують створення і редагування ієрархічно пов'язаних діаграм (потоків даних, «сутність-зв'язок» та ін.) утворюють моделі ІС;

– кошти конфігураційного управління;

– кошти документування;

– кошти тестування;

– засоби управління проектом;

– кошти реінжинірингу.

Всі сучасні CASE-засоби можуть бути класифіковані в основному за типами та категоріями.

Класифікація за типами відбиває функціональну орієнтацію CASE-засобів на ті чи інші процеси життя ТД.

Класифікація за категоріями визначає ступінь інтегрування по виконуваних функцій і включає окремі локальні засоби, вирішальні невеликі автономні завдання, набір частково інтегрованих засобів, що охоплюють більшість етапів життєвого циклу ІС і повністю інтегровані засоби, що підтримують весь ЖЦ ІС і пов'язані спільним репозиторієм [7].

Класифікація за типами переважно збігається з компонентним складом CASE-засобів і включає наступні основні типи:

– кошти аналізу (Upper CASE), призначені для побудови і аналізу моделей предметної області (Design / IDEF, BFwin):

– засоби аналізу і проектування (Middle CASE), що підтримують найбільш поширені методології проектування й які використовуються для створення проектних специфікацій (Vantage Team Builder, Designer / 2000, Silverrun, PRO-IV,

CASE. аналітик). Виходом таких засобів є специфікації компонентів і інтерфейсів системи, архітектури системи, алгоритмів і структур даних;

– засоби проектування баз даних, що забезпечують моделювання даних і генерацію схем баз даних (як правило, на мові SQL) для найбільш поширених СУБД, до них відносяться ERwin, S-Designor і DataBase Designer (ORACLE), Засоби проектування баз даних є також у складі CASE -Засобів Vantage Team Builder, Designer / 2000, Silverrun і PRO-IV;

– засоби розробки додатків. До них відносяться засоби 4GL (Uniface, JAM, PowerBuilder, Developer / 2000, New Era, SQLWindows, Delphi і ін.) І генератори кодів, що входять до складу Vantage Team Builder, PRO-IV і частково – в Silverrun;

– засоби реінжинірингу, що забезпечують аналіз програмних кодів і схем баз даних і формування на їх основі різних моделей і проектних специфікацій – Засоби аналізу схем БД і формування ERD входять до складу Vantage Team Builder, PRO-IV, Silverrun, Designer / 2000, ERwin і S- Designor. У сфері аналізу програмних кодів найбільшого поширення отримують об'єктно-орієнтовані CASE-засоби, що забезпечують реінжиніринг програм на мові С ++ (Rational Rose, Object Team).

Допоміжні типи включають: кошти планування та управління проектом (SE Companion, Microsoft Project та ін.); кошти конфігураційного управління (PVCS, SCCS і ін.); засоби тестування (Quality Works і ін.).

На сьогоднішній день Російський ринок програмного забезпечення має такими розвиненими CASE-засобами: Vantage Team Builder (Westmount I-CASE); Designer; Silverrun; ERwin + BPwin; S-Designor; CASE Аналітик; Rational Rose. [9].

Побудова онтологій – складний і такий, що займає багато часу, процес. Щоб полегшити його, в середині 90-х років почали створюватися перші середовища для процесу розробки онтологій. Вони забезпечили інтерфейси, які дозволили виконувати створення концепції, реалізацію, перевірку несуперечності і документування. За останні роки число інструментів онтологій різко зросла.

Інженерію онтологій можна визначити як сукупність дій, що стосуються:

– процесу розробки онтологій;

– життєвого циклу онтологій;

– методів і методологій побудови онтологій;

– набору інструментів і мов для їх побудови і підтримки.

Наразі 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.д.

Розглянемо найбільш відомі інструменти інженерії онтологій, URL і основні характеристики.

Система Ontolingua [11] 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 OKBC (доступ до онтології Ontolingua по протоколу OKBC) і Chimaera (аналіз, об'єднання, інтегрування онтологій). Всі додатки, крім сервера OKBC, реалізовані на основі форм HTML. Система подання знань реалізована на Lisp. Сервер Ontolingua також надає архів онтологій, що включає велику кількість онтологій різних предметних областей, 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 [12] – локальна, вільно поширювана Java-програма, розроблена групою медичної інформатики Стенфордського університету. Програма призначена для побудови (створення, редагування і перегляду) онтологій прикладної області. Її початкова мета – допомогти розробникам програмного забезпечення в створенні і підтримці явних моделей предметної області 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. Інструмент має графічний інтерфейс, зручний для використання недосвідченими користувачами, забезпечений довідками та прикладами.

Protégé заснований на фреймовій моделі представлення знання OKBC (Open Knowledge Base Connectivity) [13] і забезпечений рядом плагінів [4], що дозволяє його адаптувати для редагування моделей, що зберігаються в різних форматах (стандартний текстовий, в базі даних JDBC, UML, мов XML , XOL, SHOE, RDF і RDFS, DAML + OIL, OWL).

OntoEdit [14] 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. Наразі OntoEdit підтримує мови представлення: FLogic, включаючи машину виведення, OIL, розширення RDFS і внутрішню, засновану на XML, створення серій 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; расширяемую структуру за допомогою плагінів, а також дуже хорошу документацію. 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 [15] – 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 повну середу розробки – не підтримує розробка онтологій великого масштабу, міграція та інтеграція онтологій, контроль версій і т.д. 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 [16]. 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. Забезпечуються 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 символьному 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 реляційної 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 (EER – Extended Entity-Relationship Model) доповнивши її обмеженнями багатовимірної агрегації і проміжних схем.

1.3 Аналіз систем обробки природної мови

Комп'ютерне моделювання мовної діяльності людини є однією з базових проблем в області побудови інтелектуальних систем, що має багату історію і можливо, настільки ж далекою від свого повного вирішення, як і на початку досліджень.

Завдання автоматизованої обробки текстів природною мовою (ПМ-текстів), була вперше сформульована в 1960 – 70-х р.р. З тих пір було зроблено множина різних спроб її вирішення, проте широкого поширення такі системи поки не отримали, як правило, через невисоку якість розпізнавання, жорстких вимог до синтаксису «природної мови», а також великих витрат машинних ресурсів, необхідних для їх роботи . У всіх системах машинного аналізу тексту використовується обмежений ПМ, оскільки повної і суворої формальної моделі ні для одного ПМ поки не створено [16].

ПМ-системи постійно розвиваються, що обумовлено з одного боку, розвитком теоретичних засобів опису ПМ, а з іншого – прогресом технологій програмування [17].

Існує велика кількість систем, що обробляють ПМ. Можна виділити кілька критеріїв класифікації таких систем. Часто використовується об'єднана класифікація, при якій ПМ-системи діляться на наступні категорії: інтелектуальні

питання-відповідь системи, системи спілкування з базами даних, діалогові системи вирішення завдань і системи обробки зв'язкових текстів. У даній класифікації виділяється аспект мовного взаємодії з певною категорією програмних систем [12]-[13].

Спочатку питально-відповідні системи стали розроблятися як реакція на погану якість кодування запитів при пошуку інформації в інформаційно-пошукових системах.

Другий клас систем виник у зв'язку з появою баз даних з метою забезпечення доступу до інформації широкого класу непідготовлених користувачів. До систем цього класу відносяться PARNAX, TEAM, IRUS і ін.

Діалогові системи вирішення завдань, на відміну від систем попереднього типу, грають в комунікації активну роль, оскільки їх завдання полягає в тому, щоб отримати рішення проблеми на основі тих знань, які представлені в ній самій, і тієї інформації, яку можна отримати від користувача.

Системи обробка зв'язкових текстів досить різноманітні за структурою. Їх спільною рисою можна вважати широке використання технологій представлення знань.

Функції систем такого роду полягають в розумінні тексту і відповідях на питання про його зміст, теми використовуються для пошуку тексту з потрібною користувачеві інформацією, його реферування, пошуку додаткової інформації, пов'язаної з ним і т.д. До цих систем відносяться TACC, KERNEL, RESEARCHER, TAILOR, FAUSTUS і ін. [18].

Найбільш відомі в даний час розробки в області аналізу тексту здійснюють семантичний аналіз на рівні пропозиції або слова. Важливим класом ПМ-систем є системи перевірки орфографії [18].

Ще одним напрямком обробки текстів на природних мовах є системи автоматичного перекладу. Вони також використовують індексацію словника за словами і використовують принципи семантичного аналізу та додаткову інформацію про характерні оборотах [19].

Найбільш відомі системи обробки тексту: Alex, InBASE, AURA, InDoc – НДІ ІІ; GalaktikaZGGM – Галактика; Link Grammar Parser -Davy Temperley; Гарант-Парк-Интернет.

Спільною рисою цих систем можна вважати широке використання технологій представлення знань. Функції систем такого роду полягають в розумінні тексту і відповідях на питання про його зміст. Розуміння розглядається не як універсальна категорія, а як процес добування інформації з тексту, який визначається конкретним комунікативним наміром. Іншими словами, текст «прочитується» тільки з установкою на те, що саме потенційний користувач захоче дізнатися про нього. Тим самим і системи обробки зв'язкових текстів виявляються аж ніяк не універсальними, а проблемно-орієнтованим і [20].

Найбільш відомі в даний час розробки в області аналізу тексту здійснюють семантичний аналіз на рівні пропозиції або слова.

За результатами порівняльного аналізу існуючих систем, що використовують семантичний аналіз тексту (Galaktika-ZOOM, Link Grammar Parser, AURA, Алхімік, Робоче Місце Лінгвіста, AuthorlT і ін,), не було знайдено системи, яка аналізувала текст ТД. Тому першочерговим завданням є онтологічна модель аналізу тексту з метою побудови якісно нової моделі із застосуванням онтологічної концепції.

Прикладна морфологія, що є складовою частиною комп'ютерної лінгвістики, традиційно вважається в ній найбільш дослідженою областю, в завдання якої входить розробка морфологічних процесорів – систем автоматичного морфологічного аналізу та синтезу слів, а також систем лематизації – відомості словоформ до словниковим словами та автоматизація морфологічних досліджень, що передбачає проведення ряду лінгвістичних робіт за допомогою ЕОМ, для вирішення основного завдання – розробки морфологічних процесорів: використання баз знань для зберігання морфологічних словників, проведення типологічних досліджень, моделювання морфологічних явищ і т.ін.

Метою і результатом морфологічного аналізу є визначення морфологічних характеристик слова і його основна словоформа, основою морфологічні аналізаторів мови, що працюють без будь-яких словників, є принцип аналогії. Робляться спроби модифікацій цього підходу, які передбачають виключення словників для цілей морфологічного аналізу або використання їх в мінімальному обсязі .

В ПМ-системах морфологічний етап є початковим (або кінцевим в разі завдання синтезу) етапом лінгвістичного аналізу. Наслідком нерозв'язності проблеми морфологічної багатозначності на даному етапі є множинність виходів системи.

Таким чином, є досить великий набір методів і систем морфологічного аналізу, призначених для різноманітних задач.

Синтаксис розглядає правила побудови і окремі різновиди словосполучень і пропозицій. Результатом такого аналізу є граф, вузлами якого виступають слова пропозиції; при цьому, якщо дна слова пов'язані якимось чином, то відповідні їм вершини графа пов'язані дугою з певною забарвленням.

Можливі забарвлення дуг залежать від мови, на якому написано пропозицію, а також від обраної (X) способу представлення синтаксичної структури пропозиції.

До проблеми комп'ютерного синтаксичного аналізу існують два підходи: формально-граматичний і ймовірнісно-статистичний.

Перший спрямований на створення складних систем правил, які дозволяли б у кожному конкретному випадку приймати рішення на користь тієї чи іншої синтаксичної структури; другий – на збір статистики виникнення різних структур в схожому контексті, на основі якої і приймається рішення про вибір варіанта структури.

У ряді випадків імовірнісний підхід виявляється практичніше, оскільки не завжди потрібне точне знання синтаксичної структури. Однак формально-граматичний підхід може забезпечити більш високу точність аналізатора і не залежить від коректності матеріалу.

Для роботи з цими граматиками використовують різні методи розпізнавання і розбору тексту. Вони діляться два основних види: детерміновані і недетерміновані. У детермінованих методах на кожному кроці алгоритму існує тільки один варіант інтерпретації обробленої частини ланцюжка. Такі методи, при підстановці ланцюжка замість нетермінала, з усіх можливих виразів вибирають тільки одне. Це дозволяє виключити багатозначні прочитання вхідних символів і значно зменшує загальний час роботи алгоритму і його вимоги до пам'яті. Алгоритми цієї групи є самим швидкими, їх найгірший час роботи обмежено.

Семантика (від грец. Semantikos – позначає) – розділ мовознавства і логіки, в якому досліджуються проблеми, пов'язані зі змістом, значенням і інтерпретацією знаків і знакових виразів. У широкому сенсі семантика, поряд з сіптактікою і прагматикою, є частиною семіотики – комплексу філософських і наукових теорій, предметом яких є властивості знакових систем: природних мов, штучних мов науки (в тому числі частково формалізованих мов природничо-наукових теорій, логічних і математичних обчислень), різних систем знакової комунікації в людському суспільстві, тваринний світ і в технічних інформаційних системах [3, 4].

Семантика – є діяльність, яка полягає в роз'ясненні сенсу людських висловлювань. Її мета полягає в тому, щоб виявити структуру думки, приховану за зовнішньою формою мови.

В мінімум уявлень про семантики входить представлення, про те, що семантика є компонентом повного лінгвістичного опису, мислимого в вигляді моделі, яка вміє:

– будувати правильні пропозиції природної мови за значеннями або витягувати значення з заданих пропозицій;

– перефразувати ці пропозиції;

– оцінювати їх з точки зору семантичної зв'язності і виконувати ряд інших завдань.

Головним засобом вирішення всіх цих завдань визнається спеціальний семантичний мова для запису змісту висловлювання, а також словники та

правила, за допомогою яких встановлюється відповідність між переводять один одного пропозиціями природного і семантичного мов.

Ядро семантичних досліджень становлять розробки семантики ПМ і логічної семантики. Семантика ПМ вивчається конкретними методами в лінгвістиці, зокрема математичної.

На рівні формальної системи центральним семантичним поняттям є інтерпретація, тобто відображення формалізмов системи на деяку область реальних або ідеальних об'єктів, в деяку змістовну теорію або її частина.

У семантиці досліджуються несуперечливість і повнота таких систем за допомогою різних семантичних моделей; основну роль при цьому відіграють визначення поняття істинності.

На цей момент розроблений ряд лінгвістичних теорій, реалізованих в системах обробки природної мови, таких так трансформаційні граматики, генеративний лексикон, модель «Сенс-Текст», I1PSG, функціональні граматики, системи семантичних зразків.

В даний час жодна з теорій не може претендувати на повноту опису ПМ-феноменів, хоча найбільш просунуті лінгвістичні теорії (такі як модель «Сенс-Текст» і трансформаційні граматики Хомського) досягли задовільних теоретичних результатів. Однак подібні теорії використовують пропозицію як структурну одиницю і мають слабкі засоби представлення зв'язного тексту [22].

Найбільш популярними в даний час в комп'ютерній лінгвістиці є сучасна модель Хомського GB і теорії HPSG і SFG.

1.4 Постановка задач дослідження

У результаті розгляду можливості представленої онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації з використанням онтологічного підходу слід зробити наступні ВИСНОВКИ:

Використання онтологічних моделей і інформаційних технологій дозволяє уявити множину її об'єктів і відносин між ними в явному вигляді.

Структурованість бази знань, насиченість її термінами і дефініціями дозволяють використовувати найбільш продуктивні – текстологічні методи видобування знань, які створюють передумови побудови автоматизованих процедур обробки текстів.

Висока трудомісткість побудови, необхідність наявності певних навичок створення і масовість онтологій для представлення знань в процесі сертифікації вимагають розробки спеціальних методів і алгоритмів обробки текстових документів.

Найбільш трудомісткими процедурами в побудові онтологій є побудова словників термінів, що описують предметну область і визначення зв'язків між ними, ця обставина призводить до необхідності в розробці спеціальних методів.

В сучасних інформаційних технологіях важливе місце відводиться інструментальним засобам, системам розробки і супроводу ПЗ. Ці технології і середовища утворюють CASE-системи. Широко відомі CASE-системи, такі як BPWin, ERWin, OOWIn фірми Logic Works, Design / IDEF, CASE-Аналітик, Silverrun, Rational Rose, Vantage Team Builder, S-Designor і ін., Дозволяють практичний повністю автоматизувати процес проектування програмного забезпечення. Однак, як показав аналіз, дані системи автоматизують кінцеві етапи проектування програмного забезпечення, такі як створення звітної і супроводжуючої документації, генерація коду і т.д. Етап сертифікації – аналіз тексту технічної документації виконується аналітиком, тобто задача автоматизації даного етапу залишається відкритою.

Проведений аналіз систем автоматизації показує, що важливе значення в процесі прийому передачі ПЗ мають ресурси сертифікації проектів ПЗ.

Побудова бази здійснюється аналітиком на основі тексту технічного завдання та існуючих стандартів є зараз не автоматизованим етапом, тому що більшу складність викликає автоматизований семантичний аналіз природної мови.

Системи обробки зв'язкових текстів досить різноманітні за структурою: інтелектуальні питання-відповідь системи, системи спілкування з базами даних, діалогові системи вирішення завдань, системи обробки зв'язкових текстів та ін. Їх спільною рисою можна вважати широке використання технологій представлення знань. Функції систем такого роду полягають в розумінні тексту і відповідях на питання про його зміст. Тим самим і системи обробки зв'язкових текстів виявляються аж ніяк не універсальними, а проблемно-орієнтованими.

Найбільш відомі в даний час розробки в області аналізу тексту здійснюють семантичний аналіз на рівні пропозиції або слова.

Проведений аналіз існуючих моделей природно-мовної обробки текстів (трансформаційні граматики, генеративний лексикон, «СмыслаТекст», HPSG, функціональні граматики, системи семантичних зразків, атрибутних граматики, SFG і ін.) дозволяє стверджувати, що обробка тексту зводиться до методів синтаксичного аналізу, а проблема семантичного аналізу не вирішена.

Найбільш відомі в даний час розробки в області аналізу тексту здійснюють семантичний аналіз на рівні пропозиції або слова

Автоматичне вилучення знань з монологічних текстів з метою побудови онтології передбачає не тільки виявлення термінів, а й і здобуття знань про терміни.

Це означає, що для опису семантичної структури термінології необхідно розпізнати в тексті, як терміни, так і семантичні відносини між термінами.

Але перш ніж розглядати типи семантичних відносин, необхідно вибрати підхід, на основі якого слід виконувати аналіз пропозиції монологічного тексту.

Пропонована методика містить формалізми, необхідні для уявлення семантики ТД до ПЗ на етапі сертифікації. Відповідно до запропонованої методики система розглядається як чорний ящик, а вимоги до неї вимоги представляються у вигляді специфікації функцій і визначення потоків вхідних і вихідних впливів.

Для реалізації онтологічної моделі тексту, в рамках магістерської роботи розроблена семантична модель тексту ТД, що включає вимоги, сформульовані у вигляді документа на обмеженій природній мові.

Семантична модель аналізу тексту ТД містить розроблену розширену нечітку атрибутну граматику над онтологічної структурою формального документа "Технічна документація" на обмеженій природній мові, яка дозволяє найбільш повно відобразити вміст технічної документації.

Модель програмного забезпечення являє собою ієрархію діаграм потоків даних: загальна структура системи із зазначенням се вхідних і вихідних потоків, функції, виконувані системою із зазначенням їх вхідних і вихідних потоків.

Основна мета полягає у вирішенні завдань, пов'язаних з розробкою технологій формування нормативного профілю для сертифікації програмних систем і інтелектуальної підтримки прийняття рішень експертом при реалізації завдання експорту програмного забезпечення – формування нормативного профілю.

Завдання семантичного аналізу тексту, що пов'язане з формуванням нормативного профілю, є актуальним і передбачає, що пошук буде здійснюватися за допомогою онтологічної моделі тексту. Користувач вводить запит, який піддається семантичному аналізу, розширюється за рахунок побудованої граматики, потім перетворюється в ключові слова і вирушає пошуковій машині. Пошукова машина повертає знайдені документи, вони також піддаються лінгвістичному розбору і формуються семантичні образи документів. Образи документів порівнюються з образом запиту, робиться висновок про релевантності кожного з документів і результати аналізу (документи, які були визнані релевантними) надаються користувачеві.

2 АЛГОРИТМИ ОНТОЛОГІЧНОЇ МОДЕЛІ ТЕКСТУ ДЛЯ АВТОМАТИЗАЦІЇ СЕМАНТИЧНОГО АНАЛІЗУ ДОКУМЕНТАЦІЇ

2.1 Застосування моделей нечіткої атрибутної граматики

Схема лінгвістичного аналізу [16], наведена на рисунку 2.1. Як видно з рисунка центральне місце при такій моделі пошуку інформації займають онтології.

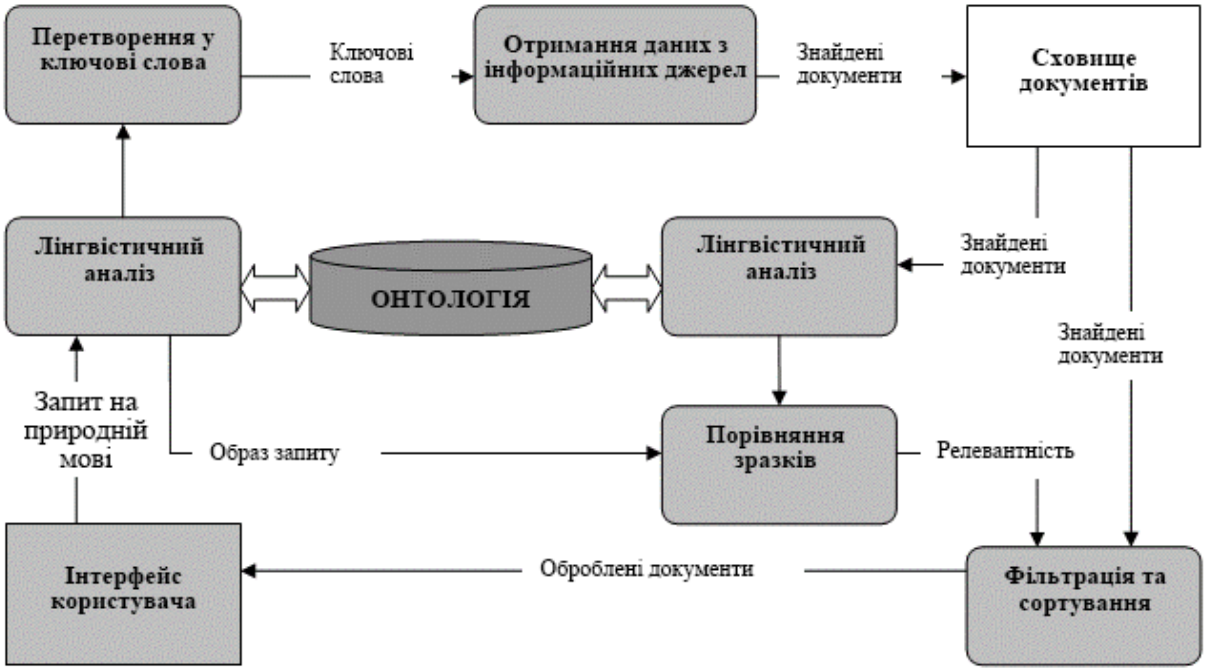

Рисунок 2.1 – Діаграма потоків даних онтологічної моделі

Однак, процес створення онтологій складний процес. Інформаційні онтології складаються з екземплярів, понять, атрибутів і відносин між ними.

Загальновідомо, що розробка і аналіз технічної документації вимагає від осіб, що займаються сертифікацією ПЗ, семантичної обробкою великого обсягу технічного тексту, глибокого знання предметної області та навичок в сертифікації. Трудомісткість процесу аналізу тексту призводить до необхідності його автоматизації. Однак надзвичайна складність проблеми синтезу та аналізу

семантики технічного тексту, для вирішення якої необхідно використовувати симбіоз методів штучного інтелекту, прикладної лінгвістики, онтологічних моделей, психології тощо, призводить до того, що вона до цих пір не вирішена.

Для семантичного аналізу тексту ТД необхідне створення уніфікованої структури тексту ТД, тобто створення такої граматики і таких онтологічних моделей, яка дозволить найбільш повно відобразити вміст ТД.

Проблема полягає в тому, що частина компонентів ТД містить інформацію, яка за своїм характером є нечіткою, що обумовлено варіантністю і рухливістю меж мовної норми і статистичними характером окремих видів інформації. Неточність інформації, що міститься в компонентах ТД, відноситься до семантичного і предметно-залежному рівню ТД і обумовлена складністю процесу формалізації описуваних явищ [19]. Рекомендації по проведенню такої формалізації складаються у вигляді описів на ПМ, що апелюють до мовної інтуїції людини, і можуть трактуватися по-різному різними фахівцями.

Практично непереборної причиною неповноти лінгвістичної інформації є відкритість і постійний розвиток ПМ: поява нових мовних одиниць, зміна властивостей існуючих одиниць та правил їх сполучуваності. Така динаміка особливо помітна в підмовою нових предметних областей з слабкою термінологією.

Іншою причиною неповноти лінгвістичної інформації є наявність величезного числа нюансів і мовних особливостей окремих носіїв мови, описати і формалізувати які на сьогоднішній день не представляється можливим.

Завданням дослідження є автоматизована обробка тексту документа на основі онтологічної моделі.

На даному етапі розвитку методів аналізу тексту не представляється можливим аналізувати природний мову без будь-яких обмежень, тому необхідно виявити особливості існуючої практики написання ТД і сформулювати додаткові вимоги.

При проведенні досліджень було розглянуто множину різних варіантів ТД і детально вивчені стандарти.

В результаті було відзначено, що [15] ТД є документ, написаний технічною мовою. ТД володіє наступними стильовими особливостями:

– текст не містить образних виразів, оціночних прикметників, майже позбавлений говірок, природна полісемічність мови зводиться до мінімуму використанням заздалегідь визначених термінів;

– текст містить наступні граматичні конструкції: граматична основа з рядом доповнень (домінуюча конструкція), причетні і дієприслівникові обороти. З точки зору російської мови причетні і дієприслівникові обороти еквівалентні окремих пропозицій, де підмет запозичується з основного пропозиції.

Найбільш зручним і поширеним способом опису функцій системи, а також її вхідних та вихідних даних є пронумеровані пропозиції українською мовою, причому однієї функції або елементу даних відповідає одне речення.

Як правило, для елементів даних вказано тип, число елементів даних і назва.

Для функцій часто вказується тип: «Основна», «Додаткова» або «Допоміжна».

Функція системи описується згідно з концепцією «чорного ящика», тобто в опис включають її входи і виходи, не торкаючись способу реалізації функції і її внутрішніх процесів.

Текстовий рядок, що описує функцію, як правило, містить назву дії, об'єкт, над яким воно виконується і джерело дії.

Дуже часто входом є джерело дії, а виходом – об'єкт, рідше і входом і виходом функції є об'єкт, тоді джерело відсутнє (за рахунок збігу джерела і об'єкта).

На основі розглянутих особливостей ТД сформульовані вимоги до методики аналізу: можливість подання неточної інформації і гнучкість – процес обробки ТД повинен здійснюватися з мінімально можливими втратами.

Нечітка атрибутна граматика ТД є формалізм, що описує структуру ТД, і є допоміжним елементом, призначеним для перетворення даних з «сирого тексту» ТД до виду списку пропозицій обмеженого природної мови і подальшого його розбору.

Фактично граматика ТД використовується для розбиття вихідного тексту документа на розділи і обробки найбільш важливих з них для нашої задачі. Для цього потрібно чітке дотримання структури документа. ТД є структурований текст, що складається з послідовності заздалегідь заданих розділів.

Сформульована ТД на природній мові містить семантично значиму інформацію, яку і слід формалізувати і представити у вигляді інваріантної до ПМ моделі.

Щоб зробити ПМ інваріантним до перестановок слів, відмінкові зміни словоформ передбачається використання методів морфологічного аналізу шляхом побудови морфологічного фільтра, який на основі аналізу відмінків іменників у реченні ПМ виділяє елементи. Наприклад: Іменник в називному відмінку є об'єктом, але в разі, якщо таких іменників два і знахідний і називний відмінок нероздільні – перший зустрінуте іменник – суб'єкт, друге – об'єкт.

Важливо відзначити, що в рамках пропонованої методики не розглядаються питання контролю повноти і несуперечності тексту ТД.

У структурі технічної документації виділені ключові слова, на підставі яких визначається приналежність розділу до певного нетерміналу, так як назви розділів можуть бути нефіксованим і відрізнятися в формулюваннях.

Запропонована методика аналізу тексту технічного завдання рис. 2.2 містить формалізми, необхідні для подання семантики вимог до програмного забезпечення на ранніх етапах проектування. Відповідно до запропонованої методики система розглядається як чорний ящик, а вимоги до неї вимоги представляються у вигляді специфікації функцій і визначення потоків вхідних і вихідних впливів.

Методика аналізу тексту ТД включає в себе обробку тексту технічної документації, створення онтологічної моделі тексту.

Для реалізації першого етапу необхідна семантична модель тексту ТД, сформульована у вигляді документа на обмеженій природній мові; другого етапу – онтологічна модель у вигляді опису вимог на графічній мові у вигляді діаграм.

Семантична модель тексту ТД містить розроблену розширену нечітку атрибутного граматику [18] над онтологічної структурою формального документа (рисунок 2.2).

Рисунок 2.2 – Методика аналізу тексту технічної документації

Розширена нечітка атрибутна граматика, необхідна для автоматизованого аналізу тексту технічного завдання, визначена у вигляді [18]:

$$AG = <N, T, P, S, B, F, A, R (A)>,$$

де N – кінцеве множина нетермінальних символів ;

T – непересічне з N множина термінальних символів;

P – кінцеве множина правил;

S – виділений символ з N, званий початковим символом;

B – множина лінгвістичних змінних $\beta_{k,i}$, відповідно термінальним символам T (змінна *i* на *k*-рівні);

F – множина функцій приналежності $f_{k,i}$, що визначають ступінь приналежності лінгвістичних змінних $\beta_{k,i}$;

A – кінцева множина атрибутів, A = AsinUAsem, де Asin – синтаксичні атрибути, Asem – семантичні атрибути;

R (A) – кінцева множина семантичних дій.

Лінгвістичні змінні з множини B = $\{\beta_{k,}i\}_{k,\ i}$ використовуються для аналізу тексту технічного завдання та описуються наступним кортежем:

$$\beta_{k,\ i} = <\beta, T(\beta), U, G, M>,$$

де β – назва лінгвістичної змінної (найменування і область застосування, підстава для розробки, призначення розробки, технічні вимоги до програмного виробу, стадії і етапи розробки і т.д.);

T (β) – мовні вирази, для лінгвістичних змінних верхнього рівня вони є лінгвістичними змінними, відповідними терміналів правій частині правила, а для лінгвістичних змінних нижнього рівня – нечіткими змінними, тобто виразами природної мови;

U – універсум, T (β) ∈ U;

G – правила морфологічного і синтаксичного опису мовних виразів, які визначають синтаксичні атрибути Asin.

Мова представлення висловів складається з констант і правил їх послідовного застосування. На морфологічному рівні константами є грамеми (рд – родовий відмінок, мн – множина). На синтаксичному – назви відносин і груп (підло – відношення між підметом і присудком, ПГ – прийменникова група). Для кожного слова вхідного тексту видається множина морфологічних інтерпретацій такого вигляду: лема; морфологічна частина мови; набір загальних граммем; множина наборів грамем.

2.2 Синтаксичні правила утворення тексту

Використовуються такі синтаксичні правила утворення тексту:

- дієприслівникових, причетний, вступний оборот;
- необособлене узгоджене визначення в препозиції;
- кількісна група; послідовність чисел упереміш зі знаками пунктуації;

– фрагмент з особистої формою дієслова, з коротким причастям, з коротким прикметником, з предикативу, з інфінітивом, з тире, з порівняльним прикметником;

– іменник + числовий ідентифікатор;

– правила для побудови ПІБ;

– слова ступеня з групою прикметника або дієприкметника;

– однорідні прикметники, прислівники, інфінітиви, прикметники вищого ступеня;

– групи дати, часових відрізків;

– аналітична форма вищого ступеня дод. або прислівники;

– наріччя та дієслово;

– одне або кілька прикметників, узгоджених за родом, числом і падежу з іменником і інші.

Семантичне правило *M* для лінгвістичних змінних індукується морфологічними і синтаксичними правилами, так як сенс терму в *T* частково визначається його синтаксичним деревом, і семантичними атрибутами Asem.

Методи представлення зв'язків між правилами транслюються на мову нечіткої математики. При цьому зв'язку представляються нечіткими відносинами, предикатами і правилами, а послідовність перетворень цих відносин – як процес нечіткого виведення.

Лінгвістичні змінні верхнього рівня є складовими, тобто включають лінгвістичні змінні нижнього рівня. Завдяки цьому можна побудувати дерево лінгвістичних змінних і встановити залежність між ними.

Функції приналежності з множин $F = \{f_{k,i}\}_{k,i}$ лінгвістичних змінних $\{\beta_{k,i}\}_{k,i}$,необхідні для побудови нечіткого виведення. Зокрема, кожного правила граматики з множини P ставиться у відповідність функція приналежності $f_{k,i}$. Ця двоїста система підстановок використовується для обчислення сенсу лінгвістичної змінної.

Синтаксичні атрибути Asin, використовувані в граматиці [9]:

– 'Назва' – текст являє собою найменування розділу;

– 'Вміст' – текст являє собою вміст розділу;

– 'Клауза' – клауз;

– 'Клауза тире' – фрагмент з тире;

– Група ГЕНІТ_ІГ '– назівний група, пов'язана родовим відмінком і ін.

Семантичні атрибути, використовувані в граматиці, містять назву атрибута Asem і семантичні дію R (A) [59]:

– "онтологічна СИСТЕМА = Створення" – створюється онтологічна модель;

– "Слот НАЗВА СИСТЕМИ = Присвоєння" – значення присвоюється слоту НАЗВА СИСТЕМИ;

– "Онтогогический ПЗТІК ДАНИХ = Створення" – створюється онтологічний ПЗТІК ДАНИХ;

– "Слот ВХІД = Присвоєння", "Слот ВИХІД = Присвоєння" – значення присвоюється слотів ВХІД, ВИХІД;

– "Слот КІЛЬКІСТЬ ДАНИХ = Присвоєння" – значення присвоюється слоту КІЛЬКІСТЬ ДАНИХ;

– "Слот ТИП ДАНИХ = Присвоєння" – значення присвоюється слоту ТИП ДАНИХ;

– "Слот НАЗВА ПЗТОКА ДАНИХ = Присвоєння" – значення присвоюється слоту НАЗВА ПЗТОКА ДАНИХ;

– "Слот НАЗВА ДІЇ = Присвоєння" – значення присвоюється слоту НАЗВА ДІЇ;

– "Слот Об'єкт = Присвоєння" – значення присвоюється слоту Об'єкт;

– "Слот ОБМЕЖЕННЯ НА ФУНКЦІЮ = Присвоєння – значення присвоюється слоту ОБМЕЖЕННЯ НА ФУНКЦІЮ і ін.

Фрагмент граматики представлений в таблиці 2.1.

Таблиця 2.1 – Фрагмент розробленої розширеної нечіткої атрибутної граматики для онтологічної моделі ТД

1	<Списоквходів>	: =	<Назвасписку входів> :: 'Назва';<Опис входу> :: 'Вміст'<Список входів>
1,1	<Назва списку входів>	: =	Текст на ЕЯ, що містить слова "вхідні дані" :: 'Клауза неопред'
1,2	<Опис входу>	: =	Текст, який містить "входи" або "вхідні дані" :: 'Клауза' <Вхід> :: "Онтологічний ПЗТІК дАНИХ = Створення", "Слот ВХІД = Присвоєння"
12,1	<Вхід>	: =	[<Числоодиницьданих>] :: "Слот КІЛЬКІСТЬ дАНИХ = Присвоєння" [<Типданих>] :: "Слот ТИП ДАНИХ = Присвоєння" <Назвапотокуданих>:: "Слот НАЗВА ПЗТОКА ДАНИХ = Присвоєння"
121,1	<Число одиниць даних>	: =	ε | «Список» | «Колекція» | «Файл» | «Дерево» | «Мережа» | «Масив»
121,2	<Тип даних>	: =	«структура» | «число» | «об'єкт» | «значення» | «дата» | «час»
2	<Список функцій>	: =	<Назва списку функцій > :: 'Назва' <Опис функції> :: "ФУНКЦІЯ = Створення" ;<Списокфункцій> | ε
2,1	<Назва списку функцій>	:: =	Текст на ПМ, що містить слова "функції" або "функціональні характеристики" :: 'Клауза неопред'
2,2	<Опис функції>	:: =	<Назвафункції> :: 'Назва', "Слот НАЗВА ФУНКЦІЇ = Присвоєння" <Списоквходів> <Список виходів>.
22,1	<Назва функції>	:: =	[<Суб'єкт>] <Дія>:: "Слот НАЗВА ДІЇ = Присвоєння"<Об'єкт>:: "Слот Об'єкт = Присвоєння" <Обмеження>:: "Слот ОБМЕЖЕННЯ НА ФУНКЦІЮ = Присвоєння "[<Джерело>]
221,2	<Дія>	:: =	слово зі списку операцій

Семантичний аналіз тексту проводиться на основі розробленої граматики тексту ТД [9]:

– кожна лінгвістична змінна ТД піддається розбору, в результаті чого виходить лінгвістичне дерево, кінцевими вершинами якого є нечіткі змінні;

– нечітким змінним на кінцевих вершинах дерева призначається їх зміст і потім за допомогою системи правил Р і відповідних функцій приналежності $f_{k,i}$ обчислюється сенс лінгвістичної змінної, відповідної лівій частині правила.

Продукційні правила Р верхнього рівня служать для розбору розділів верхнього рівня. Правила для розбору розділів складаються з двох частин: перша частина служить для розбору назви розділу; друга частина служить для розбору текстового вмісту розділу.

Для деякої лінгвістичної змінної $\beta_{k,\,i}$ значення функції приналежності:

$$\mu_{k,\,i} = f_{k,\,i}(\mu_{k\,+}1,1,\mu_{k\,+}1,2,\ldots,\ \mu_{k\,+\,1,}n),$$

де конкретне значення $\mu_{k,\,i}$ – ступінь приналежності лінгвістичної змінної $\beta_{k,\,i.}$

Спочатку припускають, що все лінгвістичні змінні нижнього рівня несуть однаковий внесок у значення функції приналежності, тому можна говорити, що функція приналежності лінгвістичної змінної $\beta_{k,i}$:

$$f_{k,i}(\{\mu_{k+1,j}\}) = q_{k+1,i} * \sum_{j=1}^{n} \mu_{k+1,\,j}$$

де $\mu_{k\,+\,1,\,j}$ – ступінь приналежності лінгвістичної змінної $\beta_{k\,+\,1,}$j;

$q_{k\,+\,1,\,j} = 1 / n$ – внесок ступенів належності в значення функції. На нижньому рівні k функції приналежності визначені.

Обчислена $\mu_{k,\,i}$ порівнюється з $\mu_{l,}$ що є граничним значенням ступеня приналежності. Якщо $\mu_{k,i}>\mu_l$ в правилах вказані синтаксичні або семантичні атрибути, в які поміщається текст з відповідною лінгвістичної змінної.

Після цього дерево урізують так, щоб обчислені лінгвістичні змінні виявилися кінцевими вершинами залишився піддерева.

Цей процес повторюється до тих пір, поки не буде вирахувано сенс лінгвістичної змінної, відповідної корені вихідного дерева. Основне призначення

описаної процедури полягає в тому, щоб зв'язати зміст лінгвістичної змінної зі змістом складових її нечітких змінних за допомогою граматики.

2.3 Функціональні моделі аналізу тексту технічної документації

Побудовано функціональні моделі та графічні нотації IDEF0, що стосуються вищевикладеного, так як відмінною рисою IDEF0 є її акцент на підпорядкованість об'єктів. В IDEF0 розглядаються логічні відносини між роботами, а не їх тимчасова послідовність.

Для семантичного аналізу необхідна попередня обробка тексту ТД, яка представлена на ПМ, а також використовуються методики та алгоритми лексичного аналізу. Так як іде праця з ПМ-текстом, то для його попередньої обробки потрібний тезаурус. Після попередньої обробки тексту наступний етап – його семантичний аналіз, де застосовано нечітку атрибутну граматику, після описаних операцій на виході отримано онтологічну модель ТД.

Описаний процес представлений на рисунку 2.3.

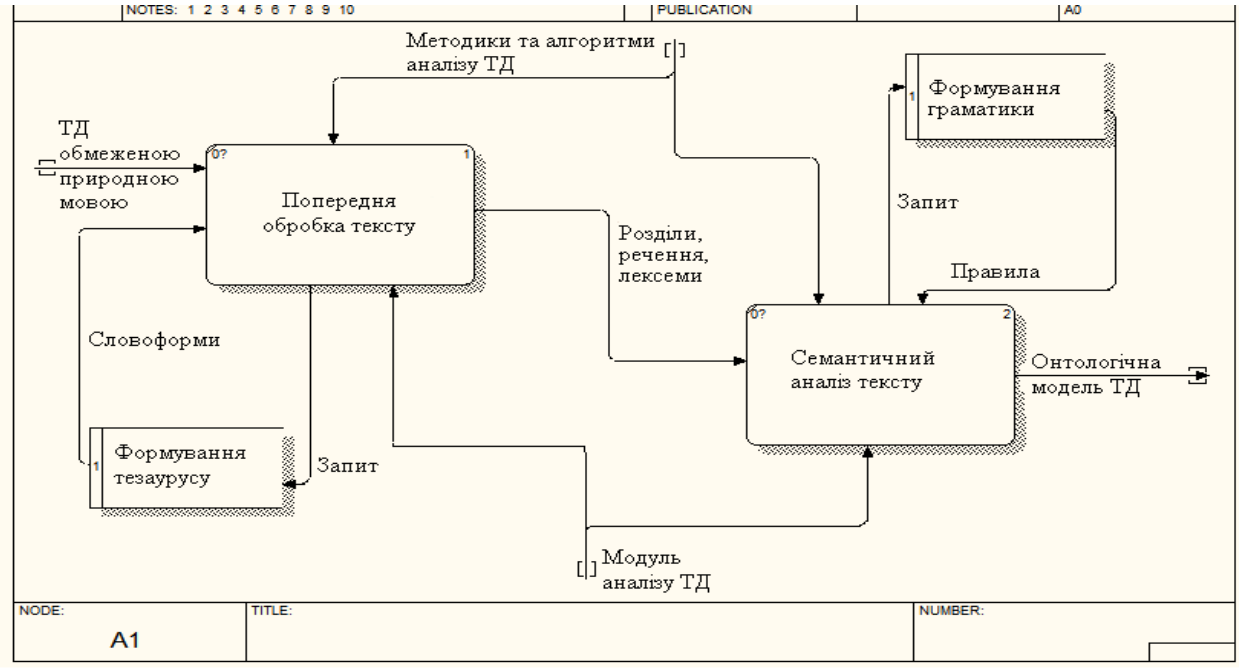

Рисунок 2.3 – Діаграма аналізу тексту ТД

Діаграма процесів автоматизації семантичного аналізу тексту ТД вказана на рисунку 2.4 (нотація IDEF0), він включає в себе (на вході) ТД природною мовою, допоміжними елементами є: методики і алгоритми, а також автоматизована система семантичного аналізу, після чого на виході отримуємо становить ядро онтологічної моделі.

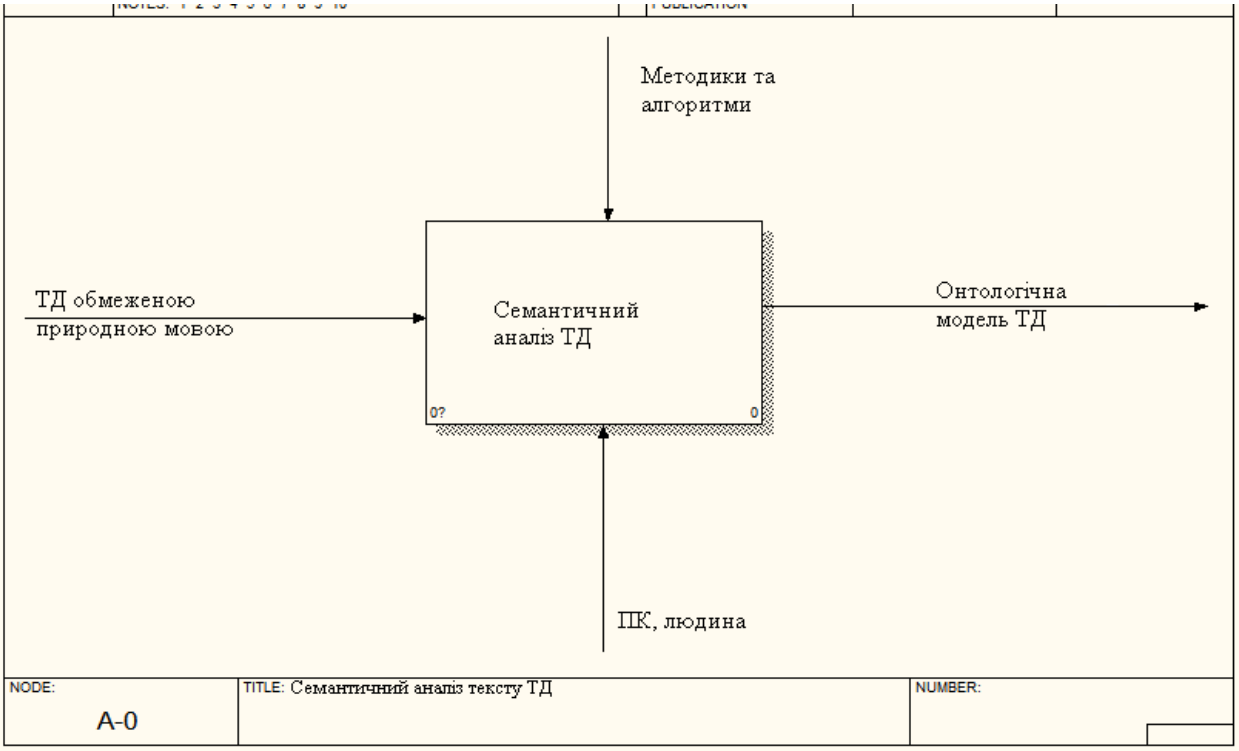

Рисунок 2.4- Діаграма семантичного аналізу тексту ТД

Для побудови онтологічної моделі необхідно вирішити задачу аналізу ТД. На вхід подається ТД на природній мові, а керуючими механізмами є методики і алгоритми аналізу ТД, підсистема формалізації ТД рисунку 2.5, модуль аналізу ТД, і методики та алгоритми формалізації ТД.

Онтології можуть бути використані у всіх областях, орієнтованих на застосування методів штучного інтелекту. У многоагентних системах онтології застосовуються як засіб, що надають загальний словник, на підставі яких здійснюється взаємодія агентів, які використовують загальну інтерпретацію знань.

Ефективне застосування онтологій в області обробки тексту, забезпечується за рахунок операцій, що дозволяють спільне і повторне використання раніше

розроблених онтологій. Послідовність операцій орієнтованих на таке використання онтологій, становить процес управління онтологіями.

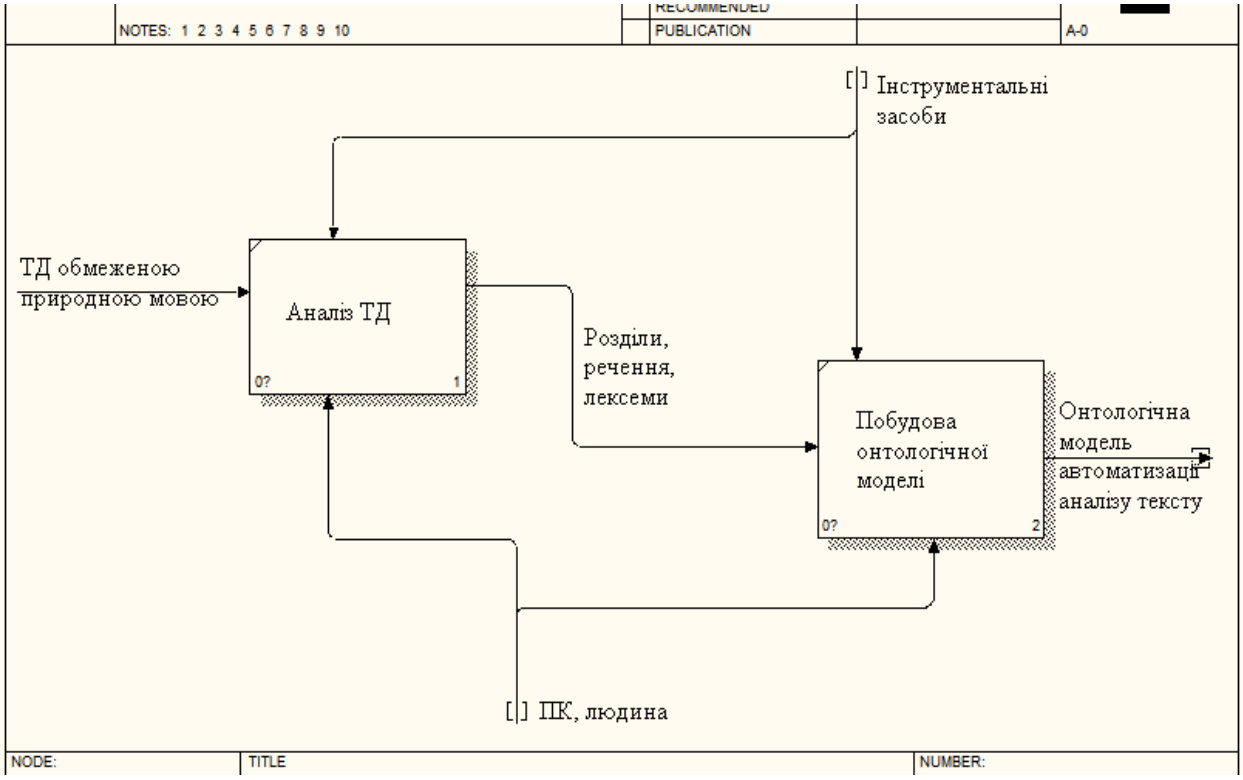

Рисунок 2.5 – Діаграма формального аналізу тексту ТД

При описі знань уявлення використовуються для позначення множин правил і об'єктів тексту, що мають загальні ознаки, але які не є типовими.

3 ОПИС ОНТОЛОГІЧНОГО ІНЖИНІРИНГУ

3.1 Формування семантичного опису математичної залежності

У проектуванні онтології прийнято виділяти два напрямки: уявлення онтології як формальної системи, заснованої на математично точних аксіомах; і розробка онтології для комп'ютерної обробки текстів. При цьому в другому напрямку на сьогодні розроблено більшу кількість онтологічних моделей, які описують різні предметні області і завдання.

Онтологічна модель тексту для автоматизації семантичного аналізу тексту ТД формально представляє математичну модель відносин, які існують в онтологіях.

Відзначимо, що незалежно від мети створення онтологічної моделі, вона повинна володіти наступним мінімальним набором властивостей: прозорість, зв'язність, розширюваність і незалежність від синтаксису.

Прозорість онтологічної моделі має на увазі, що вона повинна ефективно передавати мається на увазі значення певного терміну. Тобто вона повинна гранично чітко розділяти поняття, які беруть участь в математичних залежностях, і точно визначати їх значення.

Можливості підключення онтологічної моделі тексту говорить про те, що вона повинна дозволяти робити висновки, які узгоджуються з вихідними визначеннями понять. Все що визначаються аксіоми в онтологічної моделі тексту повинні бути логічно узгоджені між собою. Таким чином, для включення в онтологічну моделі тексту семантичної анотації, яка описує математичну залежність між об'єктами, такі об'єкти повинні бути пов'язані в онтології деякими смисловими відносинами.

Можливість розширення – властивість онтологічної моделі тексту, яке говорить про те, що дана модель повинна бути розроблена з можливістю використання розділяється і поповнюється словника термінів. Тобто онтологія не повинна оперувати вузькими поняттями, так як в подальшому використанні

можливе внесення в неї нових знань, які, в свою чергу, можуть бути використані і для створення опису раніше не врахованих або нових залежностей між її об'єктами.

Незалежність від синтаксису в онтології передбачає, що концептуалізація повинна бути специфікована на рівні знання максимально незалежно від уявлення понять предметної області на рівні символів.

Також слід зазначити, що природа об'єктів, що беруть участь у визначенні математичної залежності, може мати кілька різних уявлень. При цьому онтологічна модель тексту має інваріантне уявлення опису даної предметної області. Це властивість онтологічної моделі може, як ускладнювати, так і спрощувати, в залежності від контексту виникнення, формування опису математичної залежності. Таким чином, онтологічна модель тексту може мати таку структуру понять і властивостей, при якій математична залежність може існувати як на рівні групи пов'язаних об'єктів, так і на рівні властивостей (в тому числі, властивостей одного об'єкта).

При розробці семантичної анотації математичної залежності немає необхідності використовувати онтологію математичних функцій, операцій і операторів. Семантична анотація повинна бути записана в термінах такої онтологічної моделі, яка описує поняття самої залежності, а не її математичну структуру.

Таким чином, створювана онтологія описує знання про терміни, які використовуються у визначенні математичного виразу, а відповідна йому семантична анотація вказує лише на спосіб реалізації математичної залежності, описаної в онтологічної моделі тексту, представлені на рисунку 3.1. Виходячи з замкнутості контуру відносин між математичною залежністю, відповідної онтологічної моделі тексту та її понять і семантичної анотацією даної залежності, які слід, що процес створення онтологічної моделі тексту – ітеративний.

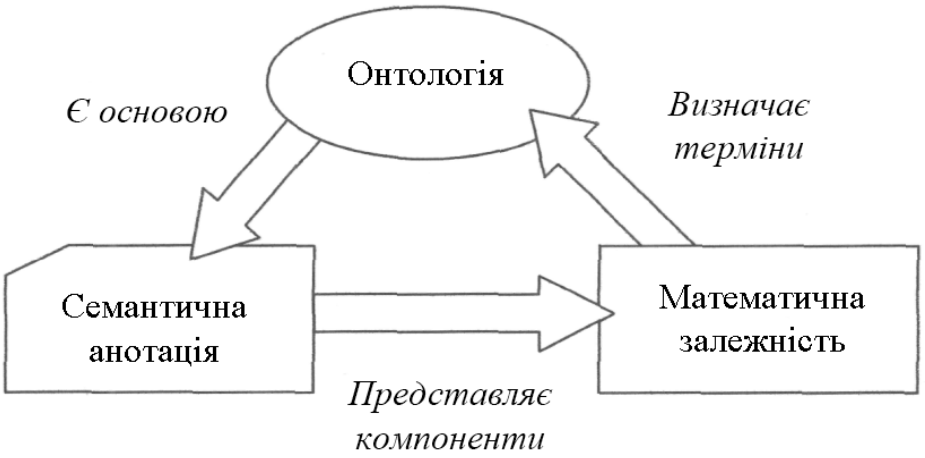

Рисунок 3.1 – Відносини між математичною залежністю і її семантичної анотацією

Таким чином, описаний похід до створення семантичної анотації математичної залежності відповідає загальним принципам онтологічного інжинірингу та фактично є процесом вивчення та опису об'єктів і їх взаємозв'язків в контексті онтологічної моделі тексту.

3.2 Розробка онтологічної моделі семантичного аналізу тексту

Для побудови онтологічної моделі тексту семантичного аналізу ТД використовуємо – стандарт онтологічного дослідження складних систем.

За допомогою методології IDEF5, онтологічна модель тексту може бути описана за допомогою певного словника термінів і правил, на підставі яких можуть бути сформовані достовірні твердження про стан розглянутої системи в деякий момент часу. IDEF5 – даний метод дозволяє розробити онтологічну модель, яка включає в себе ряд термінів галузі знань, правил, як терміни можуть комбінуватися, створюючи при цьому коректні ситуації в галузі знань і більш злагоджено висновки, які використовуються в онтологічної моделі.

Для опису стану моделі розроблено діаграму стану об'єкта, представлену на рис. 3.2.

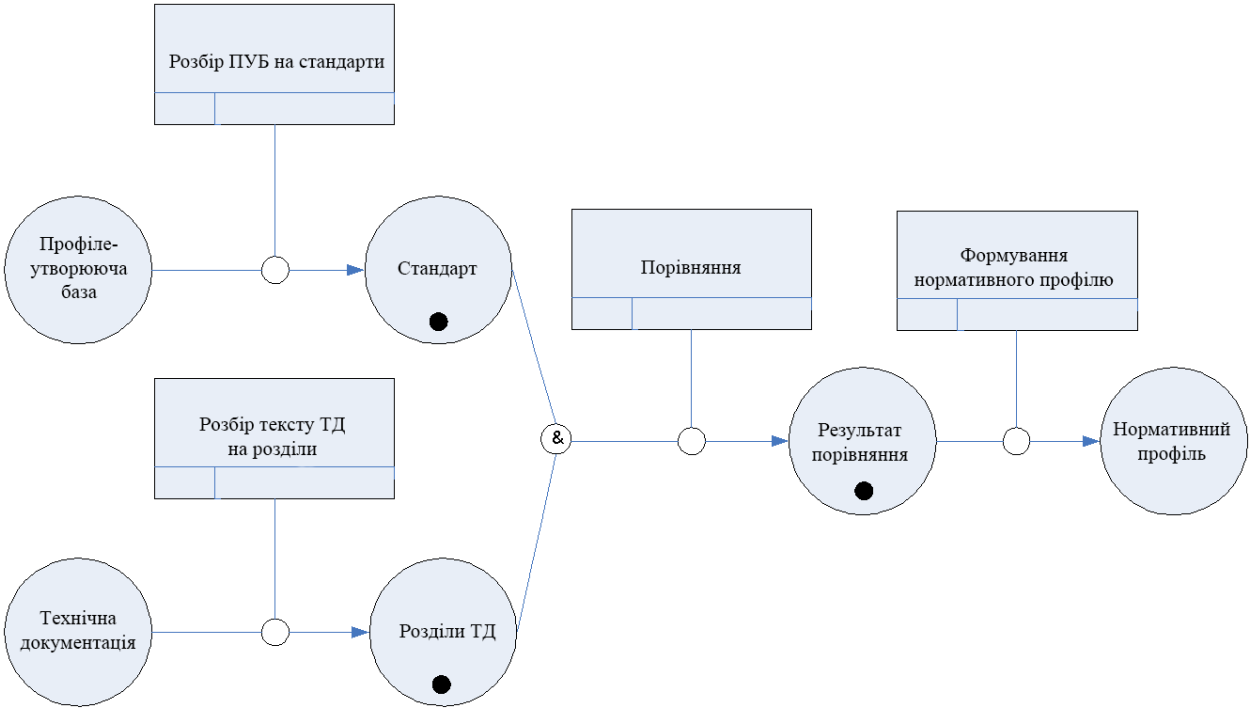

Рисунок 3.2 – Діаграма стану онтологічної моделі семантичного аналізу тексту технічної документації, стандарт IDEF5

Діаграма стану об'єкта онтологічної моделі семантичного аналізу тексту ТД, дозволяє документувати процес формування НП з точки зору зміни стану об'єкта.

У тому, що відбувається процесі можуть відбутися два типи зміни об'єкта: об'єкт «клас СТ», (як і «клас ТД»), може поміняти свій стан на елемент класу «стандарт», («Розділи ТД») або може з елементів класів, після порівняння , може бути сформований клас «нормативний профіль».

Між цими двома видами змін, по суті, не існує принципової різниці: об'єкти, що відносяться до певного класу K, в початковому стані протягом процесу можуть просто перейти до його дочірньому або просто родинному класу. Наприклад, отриманий в процесі розбору стандарт, вже відноситься не до класу «СТ», а до його дочірньому класу «класифікаційної групи стандартів».

Однак при формальному описі процесу, щоб уникнути плутанини, доцільно розділяти обидва види змін, і для такого поділу використовується позначення такого вигляду: "клас: стан". Наприклад, СТ буде описуватися наступним чином:

"СТ: категорія стандарту: клас стандарту: класифікаційна група стандартів", ТД – "ТД: розділи ТД" і так далі.

Таким чином, діаграми стану в IDEF5 наочно представляють зміни стану або класу об'єкта протягом усього процесу.

3.3 Розробка алгоритму автоматичної побудови онтологій

Інтелектуальні системи на основі онтологій показали на практиці свою ефективність, але побудова онтології вимагає експертних знань в досліджуваній предметній області і займає суттєвий обсяг часу.

Проблемою автоматизації побудови онтологій, автоматизації аналізу тексту займаються багато вчених.

Припущено, що онтології представляються у вигляді орграфа G = (V, E), де множина вершин V представляє множина предметних областей, а множина ребер E – бінарне відношення між цими предметними областями [18]. З кожним таким орграфом G = (V, E) потрібно асоціювати кінцевий частковий детермінований автомат без виходів:

$$A = (V, X = V, f, S, F),$$

де: V – множина станів, яке також служить вхідним алфавітом даного автомата;

S – підмножина початкових станів;

F – підмножина заключних станів (яке, зокрема, може бути порожнім).

Функція переходів даного автомата визначається наступним чином:

$$f(u, v) = v,$$

тоді і тільки тоді, коли (u, v) E і не визначено в інших випадках.

Автомат для наведеної онтології наведено на рисунку 3.3.

Подання онтологій у вигляді кінцевого автомата без виходів дозволяє ввести операції на онтологіях. Операції на автоматах означають операції на регулярних мовах, які акцептуються цими автоматами [12].

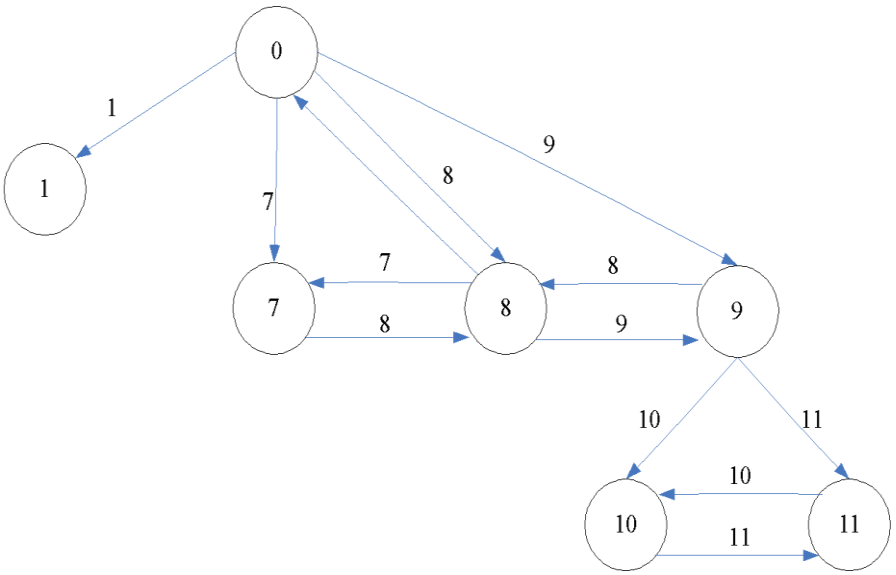

Рисунок 3.3 – Кінцевий автомат для онтологічної моделі тексту

Дане множина операцій (в разі потреби) можна розширювати в двох напрямках. Одним з таких напрямків є розширення операціями на графах (введення і видалення вершини і ребра, з'єднання графів, ізоморфного з'єднання декартового добутку і т.ін.). Іншим напрямком є алгебра відносин. Оскільки кожна онтологія є поданням деякої сукупності відносин, то можна вводити операції реляційної алгебри.

Завданням є створення онтологічної моделі тексту для автоматизації семантичного аналізу ТД при сертифікації ПЗ, яка проводиться аудитором.

Обробка тексту документа, який створюється людиною на природній мові з дотриманням зразкової структури, вимоги до якої викладені в стандартах.

На даному етапі розвитку аналізу тексту не представляється можливим аналізувати природний мову без будь-яких обмежень. Тому необхідно виявити особливості існуючої практики написання технічної документації на основі його створення з бази нормативного профілю, щоб вона могла бути проаналізована за допомогою пропонованої нечіткої атрибутної граматики тексту ТД і яка входить в онтологічну модель тексту

При проведенні передпроектних досліджень було розглянуто множина різних варіантів ТД і детально вивчені стандарти.

ТД є документ, написаний технічною мовою. ТД володіє наступними стильовими особливостями, а саме, текст не містить образних виразів, оціночних прикметників, з використанням заздалегідь визначених термінів. Текст містить наступні граматичні конструкції: граматична основа з рядом доповнень (домінуюча конструкція), причетні і дієприслівникові обороти.

Найбільш зручним і поширеним способом опису функцій системи, а також її вхідних та вихідних даних є пронумеровані пропозиції, причому однієї функції або елементу даних відповідає одне речення.

Як правило, для елементів даних вказано тип, число елементів даних і назва. Для функцій часто вказується тип: «Основна», «Додаткова» або «Допоміжна». Текстовий рядок, що описує функцію, як правило, містить назву дії, об'єкт, над яким воно виконується і джерело дії.

Для того щоб користувач системи міг без детального вивчення граматики готувати реальні ТД для обробки в системі, необхідно сформулювати рекомендації до написання ключових розділів.

Вимоги до структури пункту списку випливають з фрагмента граматики мови ТД «опис елемента даних». У специфікації елементів даних системи послідовно вказуються спочатку всі входи, а потім всі виходи, причому список входів і список виходів мають свою внутрішню нумерацію, а також ключові слова «Список входів» і «Список виходів» спочатку кожного зі списків. Опис входів і виходів рекомендовано виконувати за однією схемою: спочатку вказується число одиниць даних, далі вказується тип даних, потім назва елемента даних. Тип даних може бути як одним з основних типів: ціле, логічне, речовий, символьне, так і похідним: клас, структура, об'єднання і т.д.

Список функцій необхідно виконати у вигляді нумерованого списку. Опис кожної функції необхідно починати з нового рядка і з її порядкового номера.

Рекомендується в функції вказувати її входи і виходи у вигляді посилань на структури, описані в пункті «специфікація входів і виходів» (не допускається вставляти описі входу або виходу безпосередньо в специфікацію функції). Також необхідно вказувати конкретну дію, яке виконує функцію, її об'єкт і джерело.

Найбільший рівнем підпорядкування стандарти діляться на такі структурні елементи:

– елементи передньої частини:
– титульний аркуш;
– передмову;
– зміст;
– вступ;
– елементи основної частини:
– назва;
– сфера використання;
– нормативні посилання;
– терміни та визначення понять;
– значки та скорочення;
– вимоги до об'єкта стандартизації;
– доповнення;
– бібліографічні дані.

В результаті досліджень встановлено, що широке поширення набули підходи, засновані на статистичному аналізі тексту на природній мові.

На якість побудови онтології впливає попередня підготовка тексту, зокрема, особливості колекції стандартів.

Будова і властивості онтологічної моделі можуть бути ефективно досліджені і задокументовані за допомогою таких засобів: класів онтологічної моделі, які використовуються при описі характеристик об'єктів і процесів, що мають відношення до даної моделі, і класифікації логічних взаємозв'язків між цими класами.

Набір цих коштів, по суті, і є онтологічною моделлю тексту, а стандарт IDEF5 надає структуровану методологію, за допомогою якої можна наочно і ефективно розробляти, підтримувати і вивчати цю онтологію.

Висунуто вимоги до онтологічної моделі тексту для автоматизації семантичного аналізу тексту технічної документації: її структурі, наповненню і функціональному призначенню.

Також виділені основні принципи створення формального опису семантичного аналізу тексту технічної документації, як зовнішнього по відношенню до онтологічної бази знань інформаційного об'єкта.

3.4 Алгоритмічне забезпечення побудови онтологічної моделі тексту

Вхідний інформацією для алгоритму є текст технічного завдання, вихідна інформація – діаграми потоків даних.

Аналіз ТД складається з двох блоків: попередня обробка тексту ТД і семантичний аналізатор. Блок попередньої обробки тексту з вихідного тексту ТД генерує список лексем обмеженого природної мови технічної документації. Семантичний аналізатор приймає список лексем і обробляє його згідно граматиці ТД, генеруючи онтологічну структуру у вигляді бази знань.

Попередня обробка тексту призначена для того, щоб розділити вихідний текст на природній мові на окремі лексеми, ця операція виконується в три етапи: поділ на розділи, пропозиції і окремі лексеми.

Після першого етапу обробляється не весь текст ТД, а його частини, представлені по розділах. По ходу роботи лексичного аналізатора текст ТД дробиться спочатку на все більш дрібні розділи, потім на окремі пропозиції (зі збереженням структури розділів) і лексеми із зазначенням приналежності до пропозицій [13].

Вхід: текст ТД на ПМ як послідовність символів Вихід: дерево розділів у вигляді таблиці. Розділом називається фрагмент тексту між двома заголовками будь-якого рівня, від першого до третього (це пов'язано з тим, що заголовки

четвертого і подальших рівнів за українськими стандартами на текстові документи не нумеруються).

Таблиця розділів має зв'язок сама з собою типу «петля» по полях «код розділу» і «код батьківського розділу». Таблиця розділів пов'язана по полю «код батьківського розділу» ставленням типу «багато до одного» рекурсивно з таблицею розділів (таблиця 3.2), що використовується для зберігання і графічного відображення дерева розділів (рисунок 3.4).

Таблиця 3.2 – Структура таблиці дерева розділів

Поле	Тип	Примітка
Код розділу	Число	Унікальний числовий ідентифікатор
Код батьківського розділу	Число	Число, рівне коду розділу, в який безпосередньо входить даний розділ. У батьківського розділу це поле містить «-1». За коректно заповненою таблиці розділів завжди можна відновити структуру вихідного тексту
Номер розділу	Число	Число, рівне номеру розділу для збереження порядку розділів в структурі документа
Назва	Рядок	Ім'я розділу (унікальність не потрібно)
Текст	Рядок	Вміст розділу

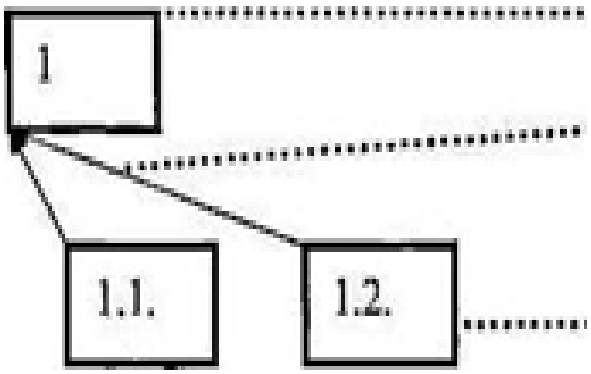

Рисунок 3.4 – Приклад роботи з деревом розділів – зв'язок по полю «код батьківського розділу»

Алгоритм працює наступним чином: спочатку весь текст ТД ділиться на розділи першого рівня («1.», «2.», «3.» і Т.Д.,), потім розділяється кожен розділ і так далі до третього рівня.

Поділ технічної документації на розділи здійснюється за допомогою кінцевого автомата, описаного на рисунку 3.5.

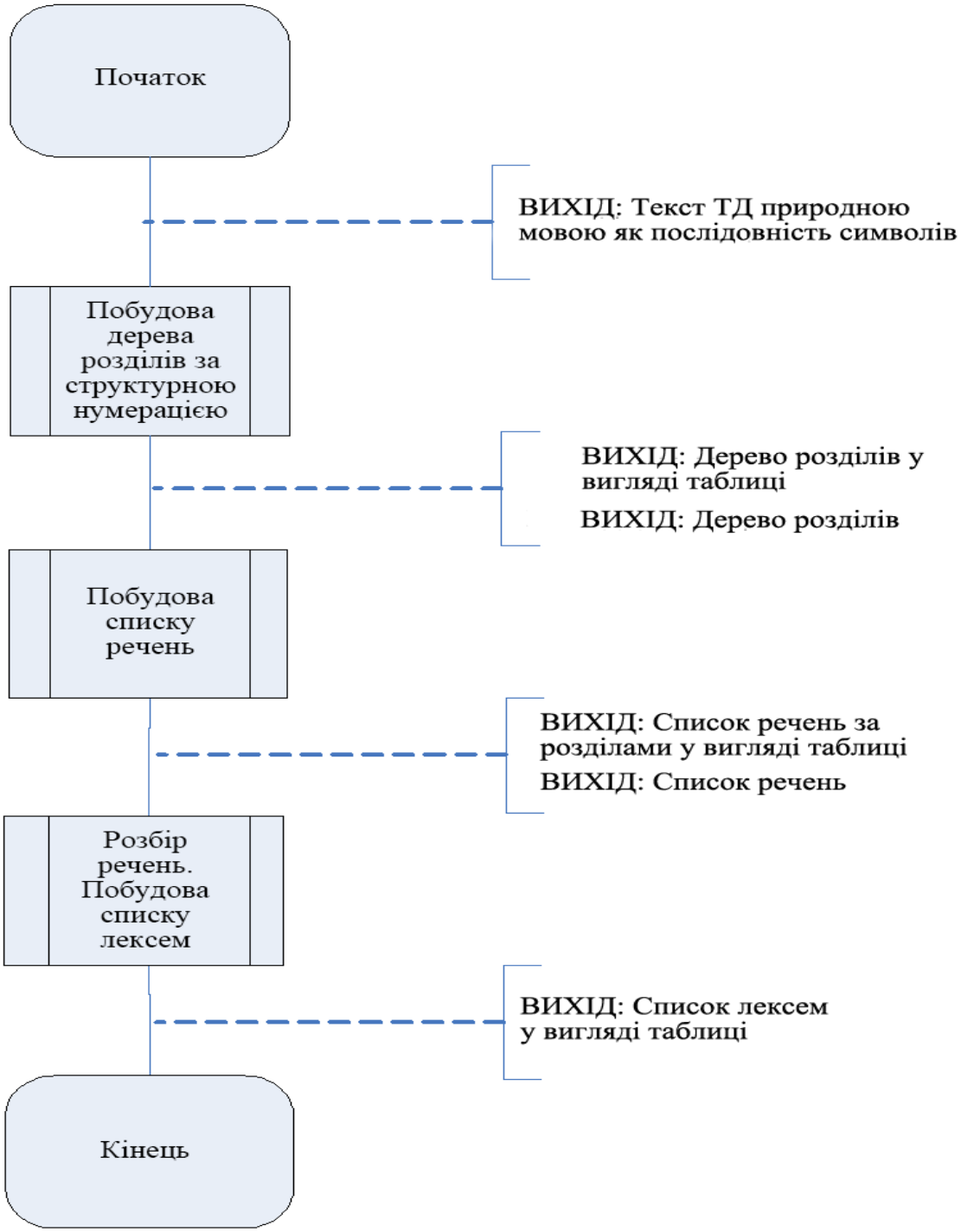

Рисунок 3.5 – Алгоритм попередньої обробки тексту

Вхідний символ кінцевого автомата: $З_1$ – порожній простір, $С_2$ – пробіл, $С_3$ – новий рядок, $С_4$ – кінець тексту, $С_5$ – '1' ... '9', $С_6$ -'П ', $С_0$ – будь-який інший символ.

Проміжні стану автомата: $а_1$ – початок розбору номера розділу, $а_2$ – послідовність символів – текст, $а_3$ – послідовність символів – нумерація, $а_4$ – початок розбору назви розділу, $а_5$ – послідовність символів – назва розділу, $а_6$ – початок розбору тексту розділу або додатку, $а_7$ – послідовність символів продовження тексту розділу або додатку, $а_8$ – початок розбору назви програми, $а_9$ – послідовність символів – назва програми, $а_0$ – кінець ТД.

Розбір тексту ТД починається з розбору розділів першого рівня і розбору додатків. Після формування таблиці першого рівня починається аналіз тексту кожного з розділів першого рівня.

Так як в тексті розділу першого рівня не може бути додатків, розбір додатків не проводиться. Після формування таблиці розділів другого рівня аналізується текст кожного розділу другого рівня і формується таблиця розділів третього рівня. В кінці роботи всі таблиці об'єднуються в автомат розбору тексту ТД на розділи представлений на рисунку 3.6.

При попаданні автомата в стан, в якому відбувається зміна таблиці розділів, в таблицю розділів вносяться зміни відповідно до попереднього проміжного стану. Наприклад, якщо послідовність символів – назва розділу, то в таблиці розділів заповнюється поле «Назва розділу».

Розбір технічної документації починається з розбору номера розділу. Якщо воно починається не з нумерації, а з тексту, то виводиться повідомлення про невідповідність технічного завдання ДСТУ. З цієї причини в автоматі двічі використовується блок розбору номера розділу. Потім створюється розділ і додається номер у відповідне поле таблиці розділів.

Після збереження номера проводиться розбір назви розділу до початку нового рядка і тексту розділу. В подальшому переходить в кінцевий стан, якщо зустрічається кінець тексту. Якщо зустрівся новий рядок, то перехід в стан

пропуску порожнього простору. Якщо ж в тексті є цифри, то перевіряється, чи не є це номером нового розділу.

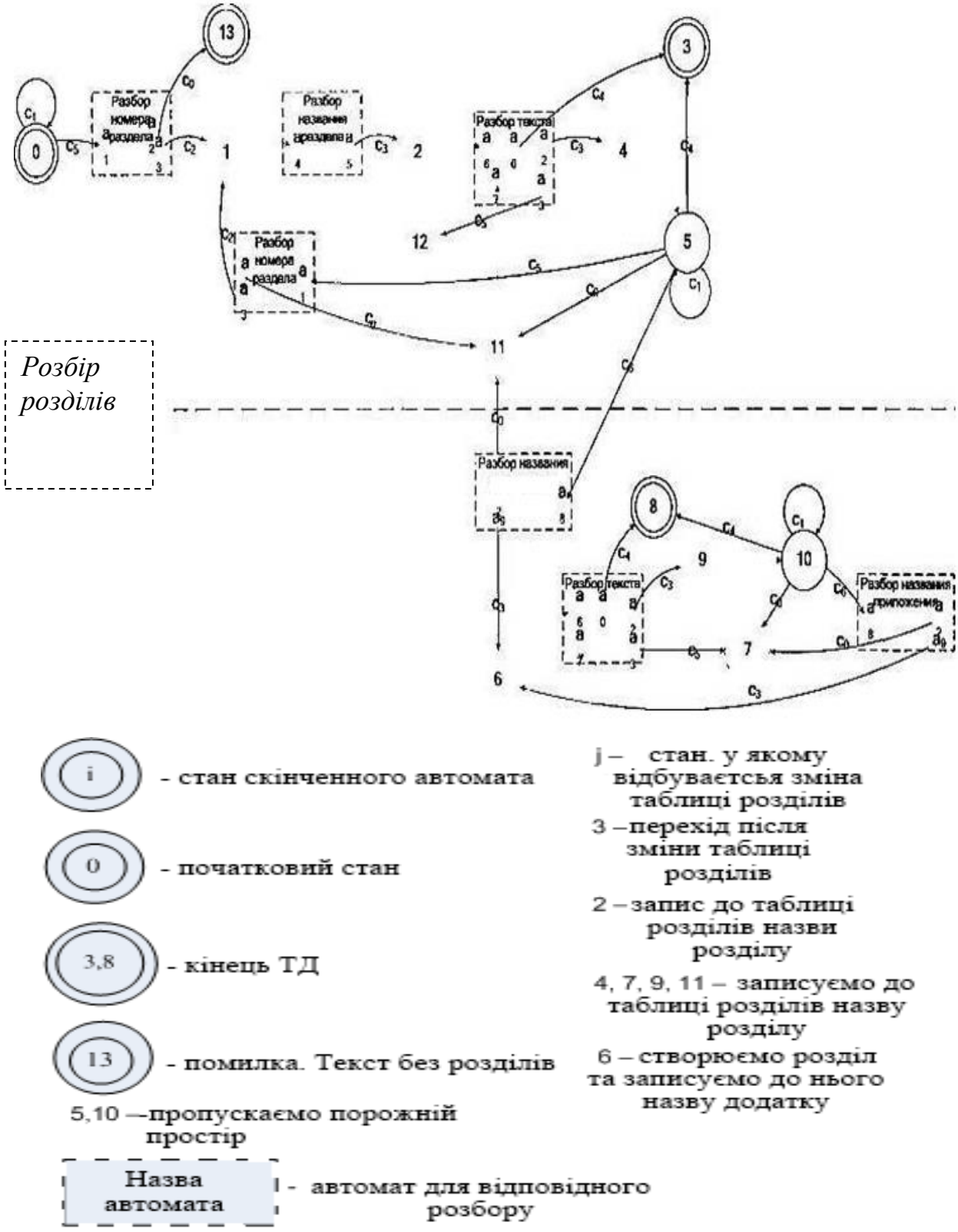

Рисунок 3.6 – Автомат розбору тексту ТД на розділи

Так як відповідно до ДСТУ ТД має «Додатки», то переходимо до розбору додатки при зустрічі символу "П". Розбір тексту додатка проводиться аналогічно розбору тексту розділу.

Формують таблицю лексем, що містить поля: код лексеми – унікальний числовий ідентифікатор; код пропозиції – число, рівне коду пропозиції, в якому знаходиться лексема: номер лексеми – число, рівне номеру лексеми в реченні; текст – текст лексеми.

Розроблено кінцевий автомат для розбору на лексеми рисунку 3.7.

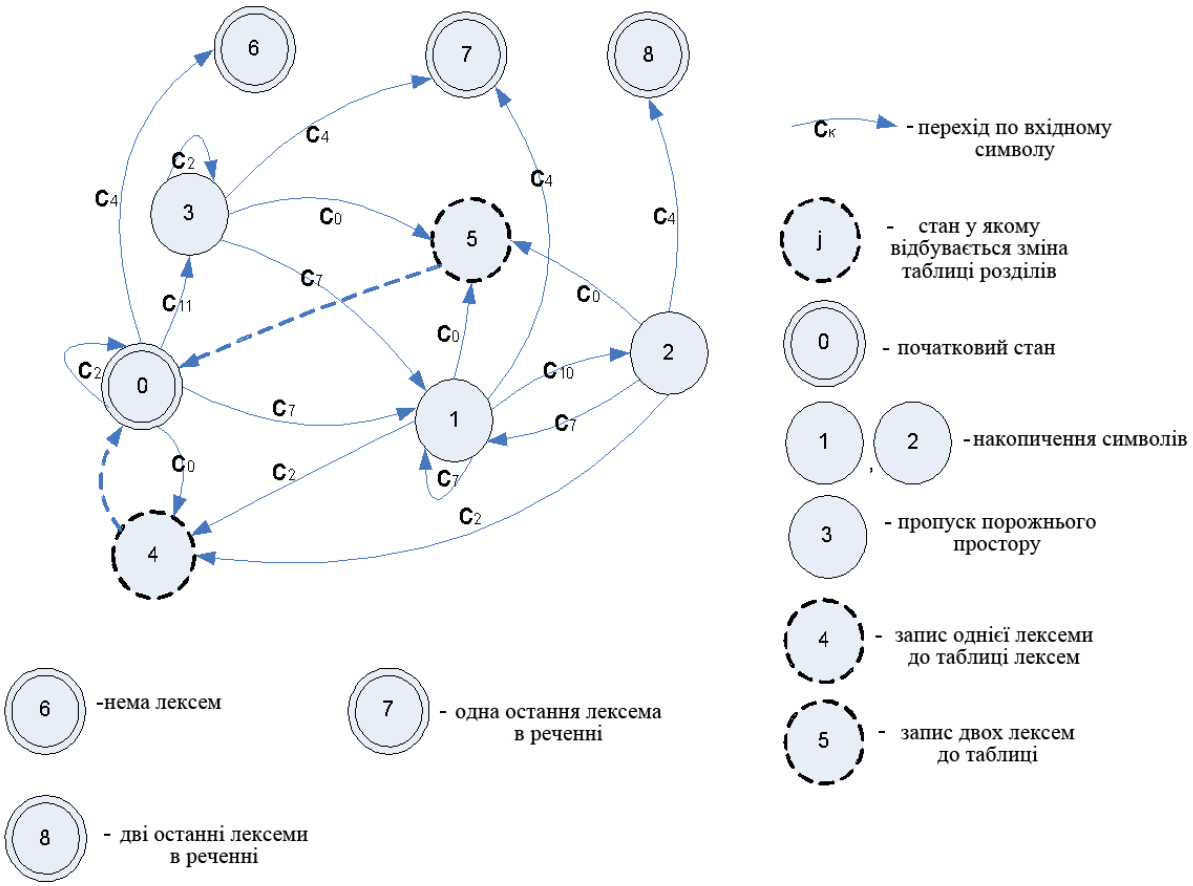

Рисунок 3.7 – Кінцевий автомат розбору тексту технічної документації на розділи

Вхідний символ кінцевого автомата: $З_1$ – порожній простір, $С_2$ – пробіл, $С_3$ – новий рядок, $С_4$ – кінець тексту, $С_5$ – '1' ... '9', $С_6$ -'П ', $С_0$ – будь-який інший символ, С 11'-'.

На вхід автомата подається кожне речення з таблиці пропозицій. Проводиться накопичення символів до точки, пробілу або кінця пропозиції і

збереження в таблицю лексем. Розбір проводиться до тих пір, поки не зустрінеться кінець пропозиції.

Для адекватної роботи алгоритму необхідно аналізувати текст технічної документації, написаної відповідно до ДСТУ.

Процес розробки онтології складається з стадії побудови і стадії підтримки. Кожен з таких етапів включає певний набір операцій.

Операції над онтологіями діляться на сім груп:

– операції по редагуванню;

– алгебра онтологій, в яку входить нечітка атрибутного граматика тексту ТД;

– операції по інтеграції онтологій;

– операції з агрегування і декомпозиції онтологій;

– операції по перетворенню;

– операції в порівнянні;

– перевірка і оцінка операцій, що виконуються інтелектуальними агентами.

Основна операція онтологічної моделі тексту – операція по формуванню зрізу. Це складна операція, що складається з операції вибірки, з'єднання, відсікання і перевірки на внутрішню узгодженість.

Після формування зрізу між описом запиту і сформованим зрізом встановлюються відповідності з правилами, описаним в розділі 2.

Пошук – операція, що виконує пошук в бібліотеці онтологічного додатки, що містить терміни онтологічного запиту, а також ТД і СТ.

Для цього спочатку терміни запиту відображаються в словнику бібліотеки, визначеному як множина термів

$$V = \{\text{Term}\},\ \text{Term} = (\text{ID}, \text{KeywordSet}, \text{Description}),$$

де: ID – ідентифікатор терма (відповідає ідентифікатору поняття або терміна, використовуваного в онтологічному додатку);

KeywordSet = {KeywordEntry} – множина ключових слів або синонімів, що відносяться до даного терму, отриманого після розкладання на розділи,

шляхом застосування нечіткої атрибутного граматики для автоматизації аналізу тексту;

KeywordEntry = (KeyWordID, KeyWord) – пара, що складається з ідентифікатора ключового слова і самого слова; Description – трактування значення терміна в описуваної області інтересів.

В бібліотеці проводиться пошук класів по ID, отриманого після відображення терміна запиту в словник бібліотеки.

Приклад описуваної операції для онтології A_1, A_2,..., A_m показаний на рис. 3.5. У розглянутому прикладі терміни запиту відобразилися в нумерацію розділів і слова словника бібліотеки, які відповідають іменам класів про1, $про_2$,$про_3$ онтологічної моделі.

Елементи «розділи» ТД і СТ, знайдені в результаті виконання операції і порівняння, на рисунку затінені.

Вибірка – операція по формуванню контекстів в онтологічного додатки, які увійшли в множина $\{A,\}^{m}_{=}1$.

На першому етапі виконання даної операції у всіх онтологічних додатках з множини $\{A, -\}$ $^{тм}_{=\ 1}$ позначаються класи, знайдені в результаті виконання операції *«пошук»*. Потім позначаються класи і атрибути, пов'язані обмеженнями зі знайденими класами і поміченими атрибутами рисунку 3.8.

У разі організації класів тільки у вигляді таксономії, класи і належні їм атрибути позначаються вниз до самого нижнього рівня і вгору до класу, з якого починається спадкування атрибутів.

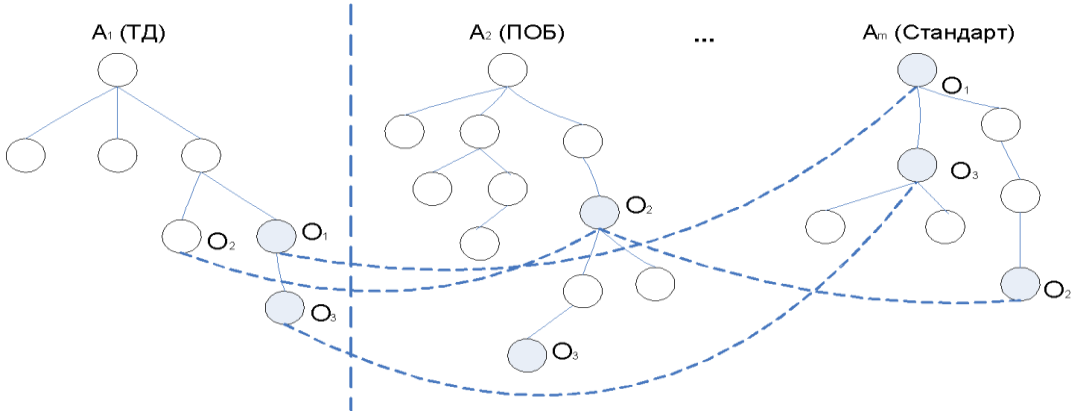

Рисунок 3.8 – Операція пошуку

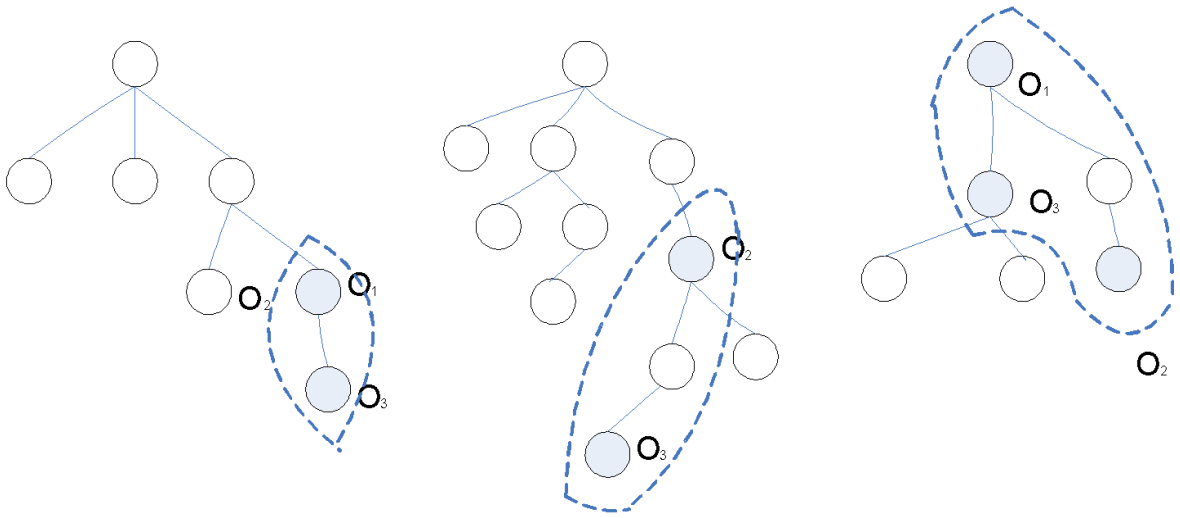

Рисунок 3.9 – Операція вибірки

Результат операції з'єднання онтології показаний на рисунку 3.10.

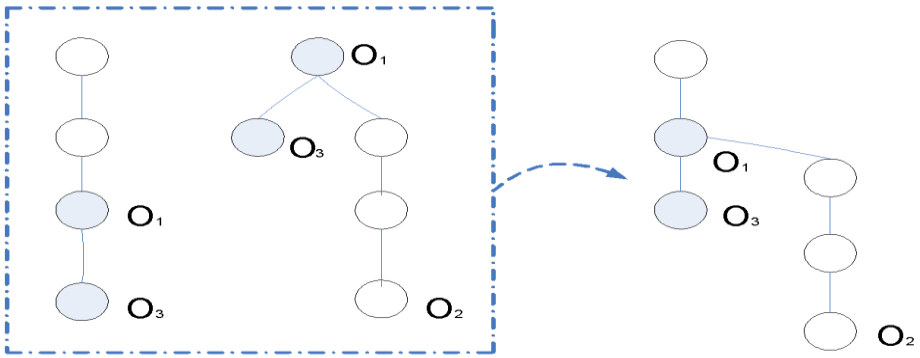

Рисунок 3.10 – Операція з'єднання

З'єднання – операція, що об'єднує в одну онтологію витягнуті контексти, що входять в множина $\{A,\}^{m}_{=}$ 1. Розділи зі схожими іменами об'єднуються в один клас «нормативний профіль» з включенням належних їм атрибутів і зв'язують їх обмежень. Якщо домени атрибутів мають різні, вони приводяться до єдиного типу. Домени, що описують різні діапазони значень однакових атрибутів, розширюються таким чином, щоб діапазон атрибута, в який об'єднуються кілька атрибутів, включав в себе діапазони значень цих атрибутів. Функціональні обмеження і обмеження сумісності класів перевіряються експертом на узгодженість з вже наявними обмеженнями даного виду. Якщо протиріч не виявлено, то розділи об'єднуються в клас «нормативний профіль». В іншому випадку узгодженням цих видів обмежень займається експерт.

4 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Реалізація онтологічної моделі тексту

Онтологічна база знань – це модель знань по завданої предметної області, яка містить і семантично описані об'єкти цієї предметної області. Передбачається, що база знань реалізована засобами стандарту Semantic Web – за допомогою Protege. Для реалізації, онтологія тексту для автоматизації аналізу тексту при формуванні нормативного профілю інтегрується в модель *Ф* математичної залежності в онтологічну базу знань, пропонується розширити моделі семантичного опису ресурсів і представлення логічних правил SWRL метамодель на основі синтаксису Protege, який складається з наступних компонентів (Додаток Б):

– клас `Dependence`, який реалізує модель інтеграції *Ф* і визначає набір математичних залежностей і логічних зв'язків, який є підкласом класу `swrl: Imp`. Згідно з моделлю SWRL в ньому визначені предикати `swrl.body` і `swrl: head`, які відповідають правилам *R* в моделі інтеграції *Ф* і являють собою множин *H* і *B.* Список атомів в `swrl.body` представляє логічну формулу, яка визначає екземпляри заданої онтології *Q,* властивості яких можуть бути зв'язані за допомогою семантичних анотацій математичних залежностей *F.* Атоми, записані в `swrl.head`, завдають зіставлення властивостей примірників онтології *Q,* які були представлені в *H,* з параметрами семантичних анотацій математичних залежностей, представлених в *F;*

– властивість `express`, для класу `Dependence`, яке представляє множина семантичних анотацій математичних залежностей *F*, що входять до відповідної модель *Ф;*

– властивість `implicate`, діапазон значень якого визначено класом `Dependence`, яке описує частину системи обмежень *E* (друга частина обмежень реалізована за допомогою `owl: Restriction` моделі OWL) в застосуванні до

існуючих класів `owl: Class` і об'єктним властивостям `owl.ObjectProperty` заданої онтології *Q*. Властивість призначене для фіксації в рамках моделі *Ф* маються на увазі логічних або математичних відносин у відповідних елементів вихідної онтології *Q*.

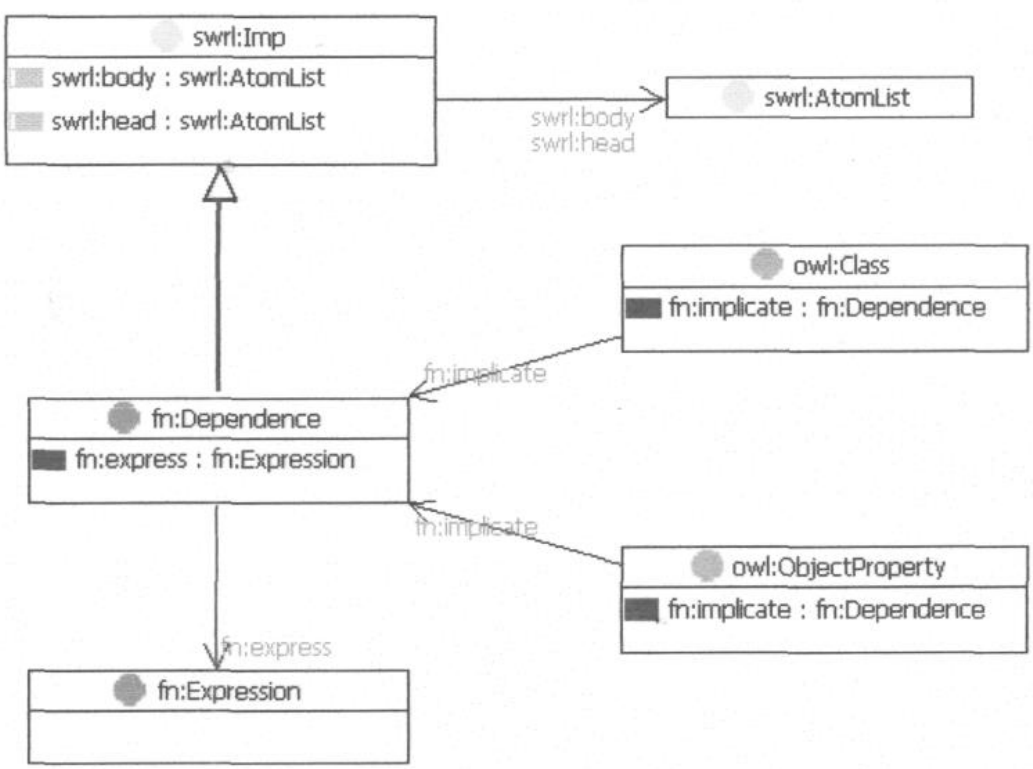

Рисунок. 4.1 – Модель інтеграції семантичної анотації математичної залежності в онтологію в термінах мови OWL

Виконання інтеграції математичної залежності в онтологію за допомогою розглянутої семантичної моделі *Ф* передбачає виконання таких дій.

По-перше, для визначення математичних залежностей, які застосовні до певних класів або відносинам цільової онтології, необхідно створити в ній екземпляр класу `Dependence`. Далі, в створеному екземплярі, необхідно задати значення наступних властивостей:

– `swrl: body` записати логічні формули, які будуть визначати правила вибору примірників заданої онтології, властивості яких повинні бути зв'язані за допомогою математичної залежності;

– в swrl: head визначити змінні, які використовуються в анотації математичної формули, що виражає задану математичну залежність, і пов'язати їх з відповідним предикатам заданої онтології з вибраними екземплярами;

– задати значення властивості express у вигляді посилань на семантичні анотації математичних залежностей, які застосовні в рамках створюваної моделі інтеграції для даної онтології.

По-друге, в описі класу або відносини заданої онтології необхідно визначити значення властивості implicate, вказавши його на новостворений екземпляр моделі інтеграції.

Результатом реалізації моделі інтеграції є деякий сполучний ресурс і його характеристики, які об'єднують між собою поняття і властивості двох початкових онтології. При цьому модифікація цільової онтології шляхом додавання в неї описів обмежень, відповідно до запропонованого методу інтеграції, має на увазі зв'язування утвореної ресурсу, тобто екземпляра класу Dependence, з онтологією предметної області.

Використання правил SWRL і специфікації обмежень онтології owl: Restriction надає можливість формально, ґрунтуючись на логіці предикатів, включити розроблену модель інтеграції математичних виразів, які реалізовані відповідними сервісами, в онтологію предметної області. При роботі з онтологією така модель може бути інтерпретована як людиною, так і програмними засобами при реалізації логічного висновку в результуючої онтології.

4.2 Опис функціональних вимог

Функціональні вимоги до ПЗ, визначають його основні функції. Функціональні вимоги описують функції, які має виконувати розробляється ПЗ.

В ході роботи над проектом функціональні вимоги були сформульовані наступні вимоги:

– поділ ТД на розділи;

– поділ СТ на категорії стандартів;

– поділ категорії стандартів на класи стандартів;

– поділ класів стандартів на класіфікаційні групи стандартів;

– поділ класіфікаційних груп стандартів на стандарти;

– поділ стандартів на розділи;

– порівняння розділів ТД і розділів стандартів, по наповненню сенсом, на сумісність;

– формування НП з розділів стандартів після їх порівняння з розділами ТД;

– збереження НП.

4.3 Опис бази онтологічної моделі

Онтологічна модель тексту для автоматизації семантичного аналізу технічної документації – нова організаційна форма семантичного аналізу ТД, яка складається з нечіткою атрибутного граматики, технічної документації та ряду стандартів, що представляють собою профілеобразуючу базу (ПЗБ) передмісний на рис. 4.2.

Основною ідеєю завдання побудови моделі є отримання її допустимої конфігурації, що задовольняє заданим вимогам при частково відомої функціональної структури на рівні компонент. Значення структури системи включає компоненти системи, їх параметри і обмеження на них.

Як приклад запиту розглянуто терміни запиту, на підставі яких будується загальний запит.

Опис структури запиту представлено на рисунку 4.2. Стрілки на рисунку позначають таксономічне відношення «бути екземпляром», суцільні лінії – зв'язки типу «частина – ціле», пунктирні – асоціативні відносини. Згідно

таксонометрічним принципам спадкування атрибутів, атрибут «критерії» визначається в класі «компонент», успадковується його підкласами.

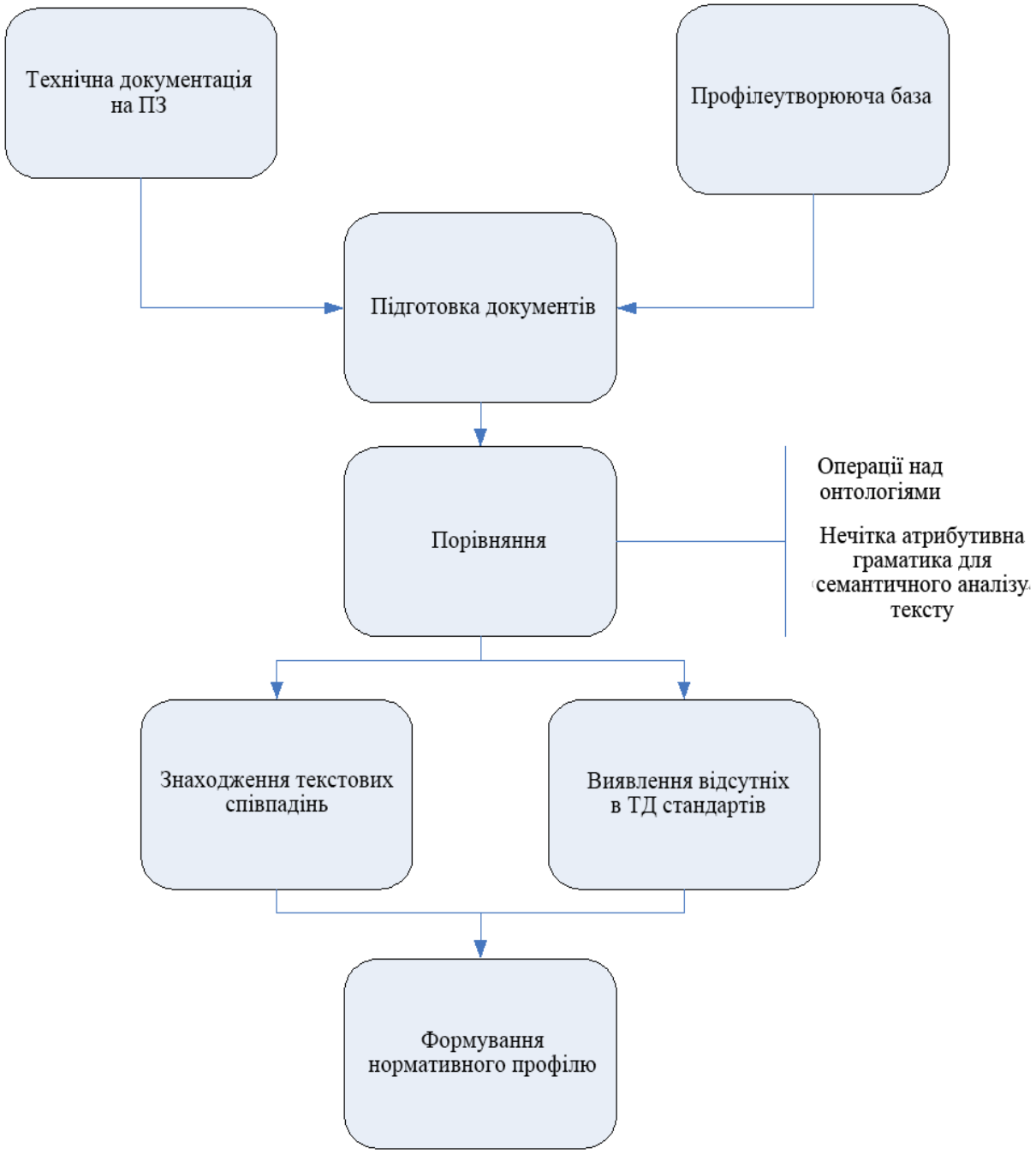

Рисунок 4.2 – Загальна схема онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації

Передбачається, що онтологічна модель тексту для автоматизації семантичного аналізу ТД, буде включена в бібліотеку онтологій (рисунок 4.3).

Рисунок 4.3 – Структура запиту

Коли нарешті з'явиться термінів запиту в використовуваний словник терміни відповідні їм в словнику потрапили в онтологічну модель предметної області «автоматизації семантичного аналізу технічної документації» рис. 4.4.

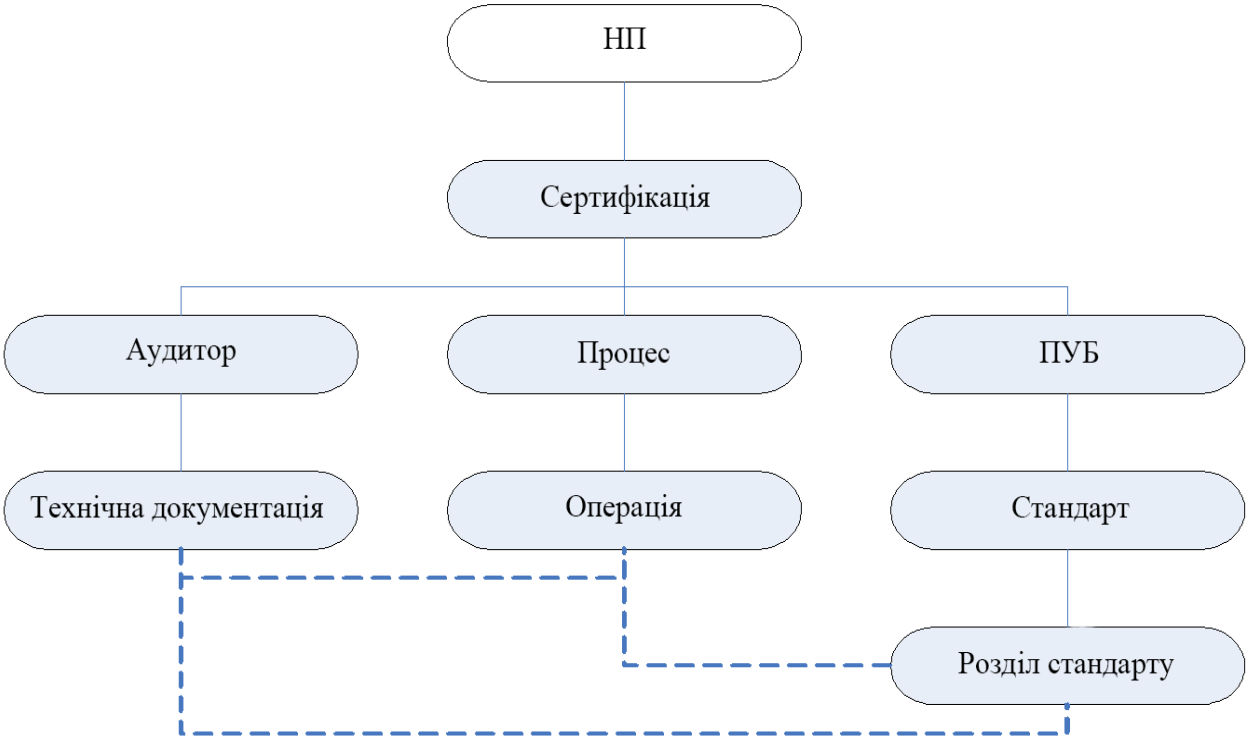

Рисунок 4.4 – Онтологічна модель тексту
для автоматизації семантичного аналізу технічної документації

Для даної онтологічної моделі необхідно виділити онтологію проблемної області «Управління», що має асоціативні зв'язки з онтологією завдань і методів, рис. 4.5.

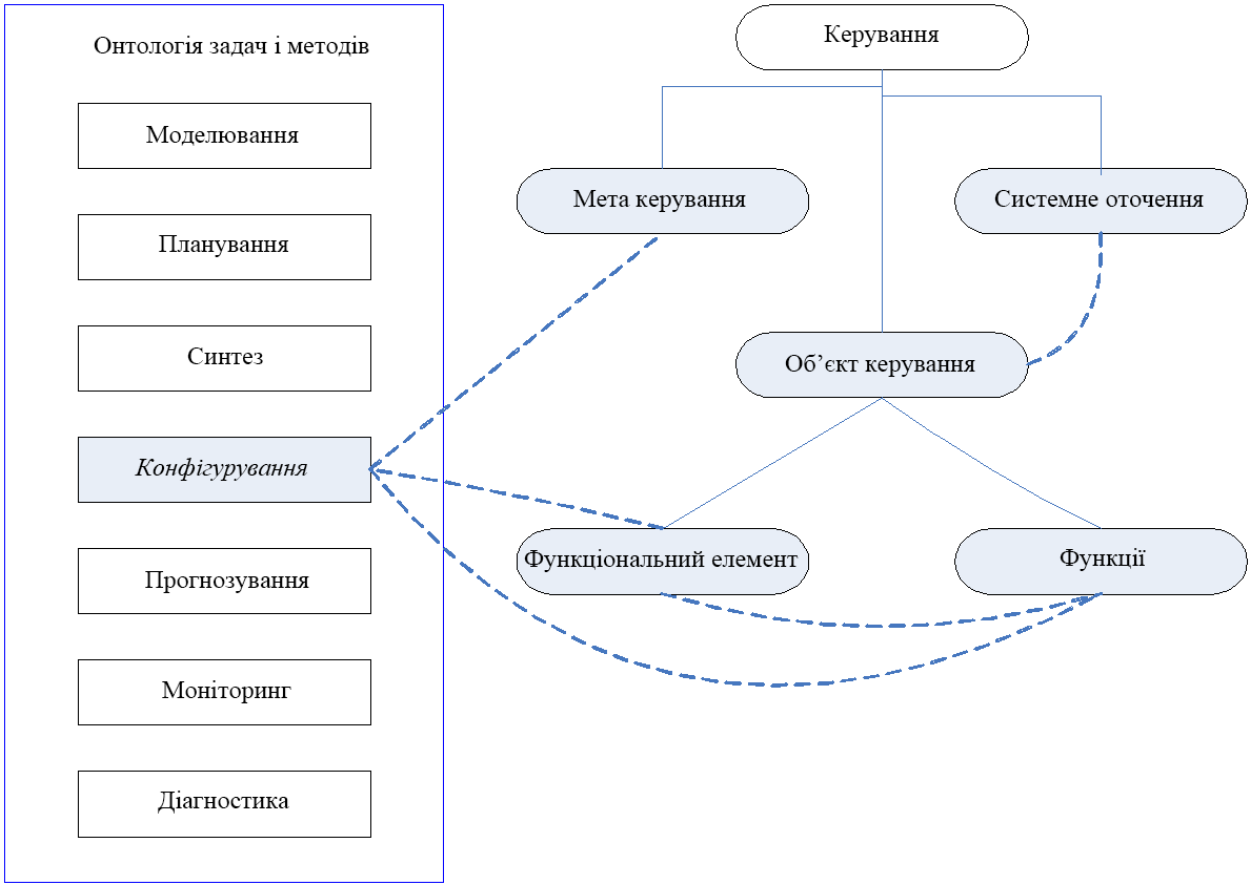

Рисунок 4.5 – Онтологія проблемної області «Управління»

Терміни, відповідні термінам запиту, на рисунку виділені курсивом. Типи завдань, включених в онтологію завдань і методів, узгоджуються з типологією завдань.

Для визначення нормативного профілю, виходячи з вимог ТД і СТ, необхідно використовувати описану в другому розділі нечітку атрибутну граматику. Текст ТД і стандарти, що входять в СТ, необхідно розкласти на розділи і провести порівняння. Глибина вибірки при формуванні зрізу, задається на основі експертної оцінки.

На підставі наявних асоціативних зв'язків між онтологією «Управління» і завданням «Конфігурація» класи онтології «Управління» і класи онтологічної моделі тексту, знаходяться в табл. 4.1 в однакових рядках, були охарактеризовані як ідентичні.

Таблиця 4.1 – Ідентичність класів

Клас онтології «Управління»	Клас онтології «Сертифікація»	Клас онтологічної моделі тексту
Об'єкт управління Функціональний елемент Функція Мета управління	Сертифікація Аудитор Операція СТ	Сертифікація Аудитор Операція СТ

Фактично, це класи онтології «Сертифікація», якими замінено класи онтології «Управління» в контексті

Між класами онтології «Сертифікація», які будуть використані при вирішенні завдань «конфігурація», і цим завданням в онтології задач і методів задаються асоціативні зв'язки.

У графі «Клас онтологічної моделі тексту» (див. табл. 4.1) представлені класи, під іменами яких були об'єднані ідентичні класи. Операція з'єднання виконує з'єднання контекстів (розділів стандартів після їх виділення), з урахуванням таблиці ідентичності класів. Результат операції – зріз онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації, який після виконання операції перевірки на внутрішню узгодженість стає новою онтологічної моделлю тексту (рисунок 4.6).

Дана онтологічна модель включається в бібліотеку онтологій.

Онтологія запиту, вийшла після операції установки відповідностей, представлена на рисунку 4.7. Штрих-пунктирні двонаправлені стрілки показують встановлені відповідності.

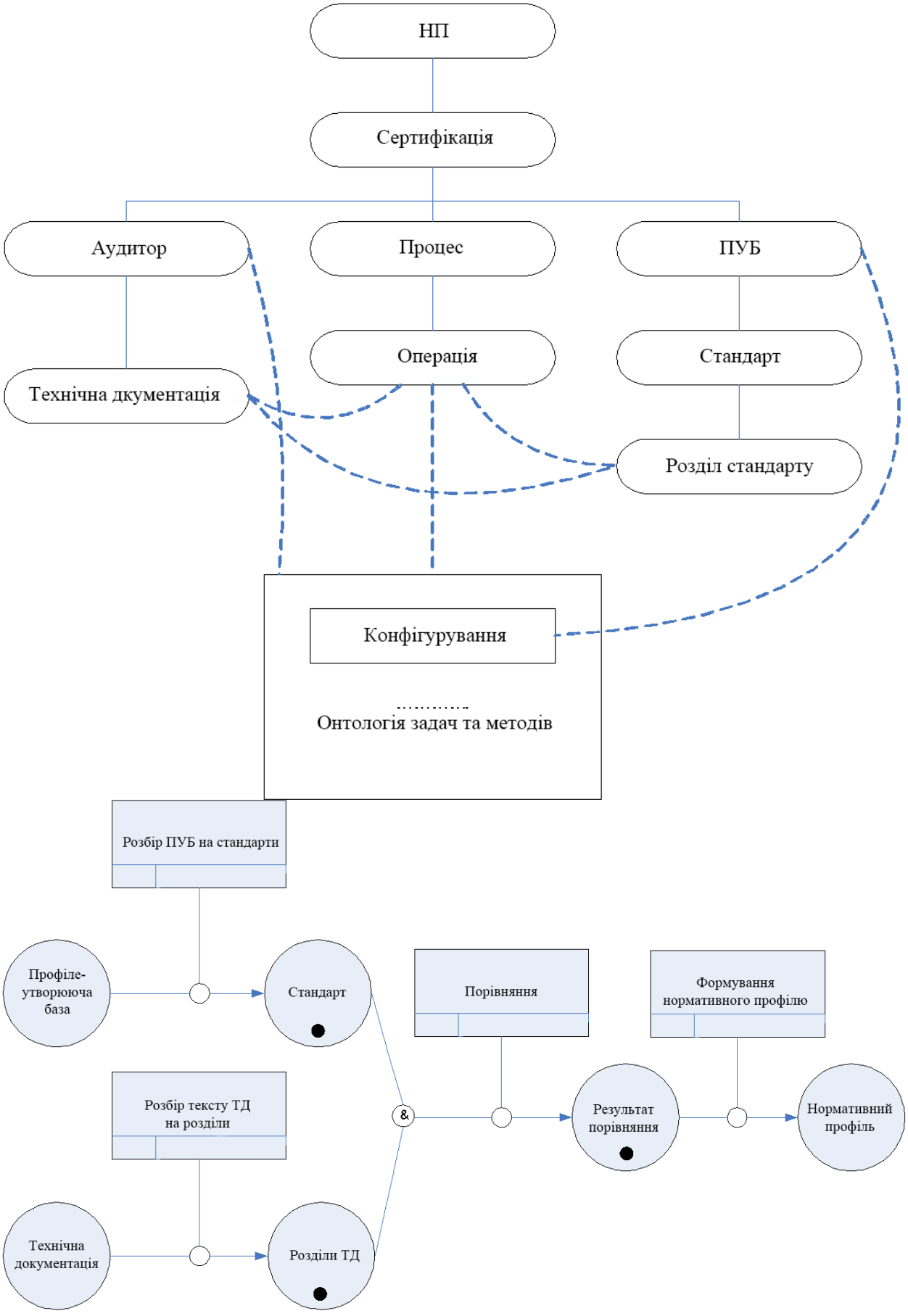

Рисунок 4.6 – Зріз онтологічної моделі тексту

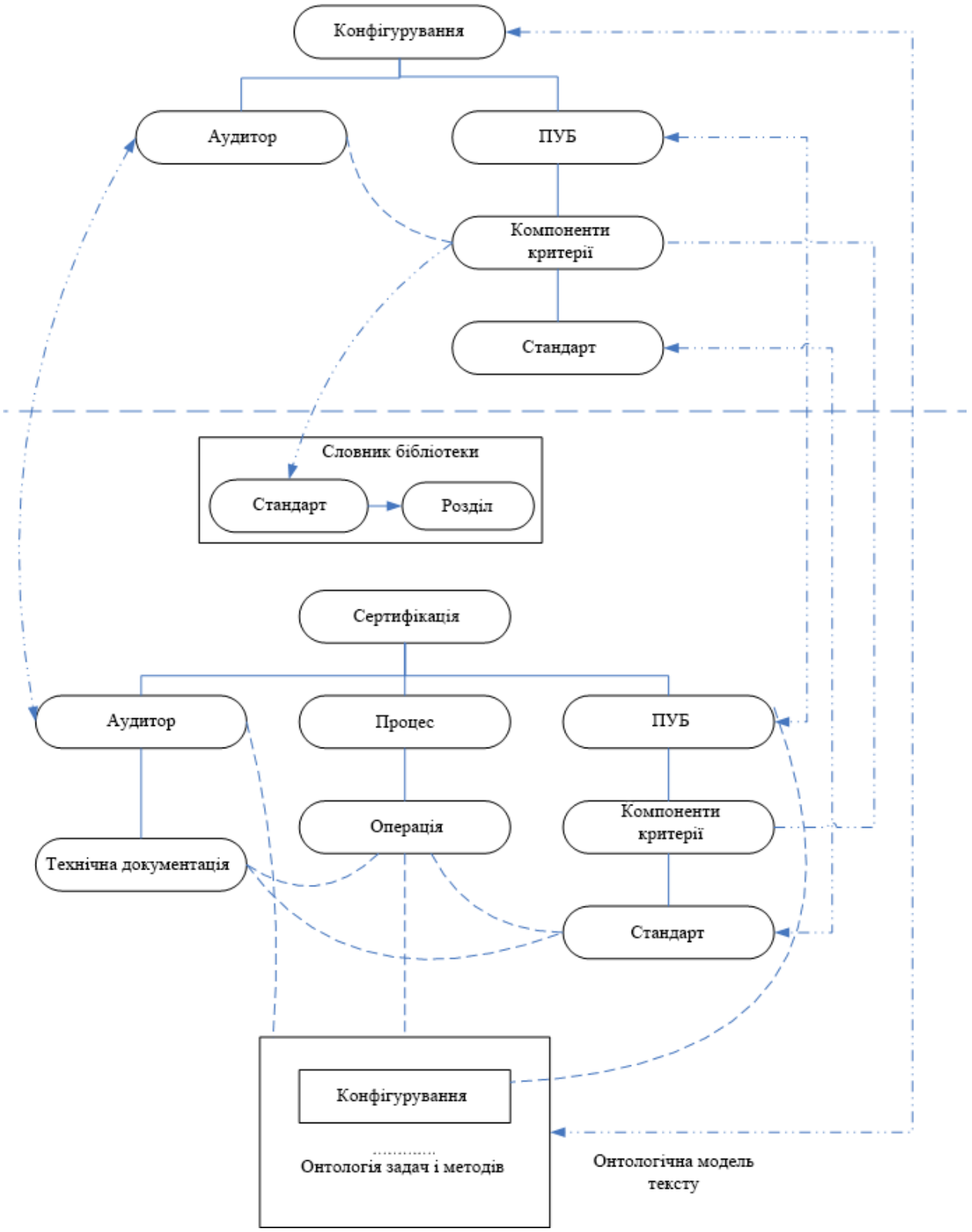

Рисунок 4.7 – Онтологія запиту

Для джерел знань, знайдених в результаті виконання операції пошуку джерел знань, будується онтологія джерела знань. Приклад структури джерела знань наведено на рисунку 4.8.

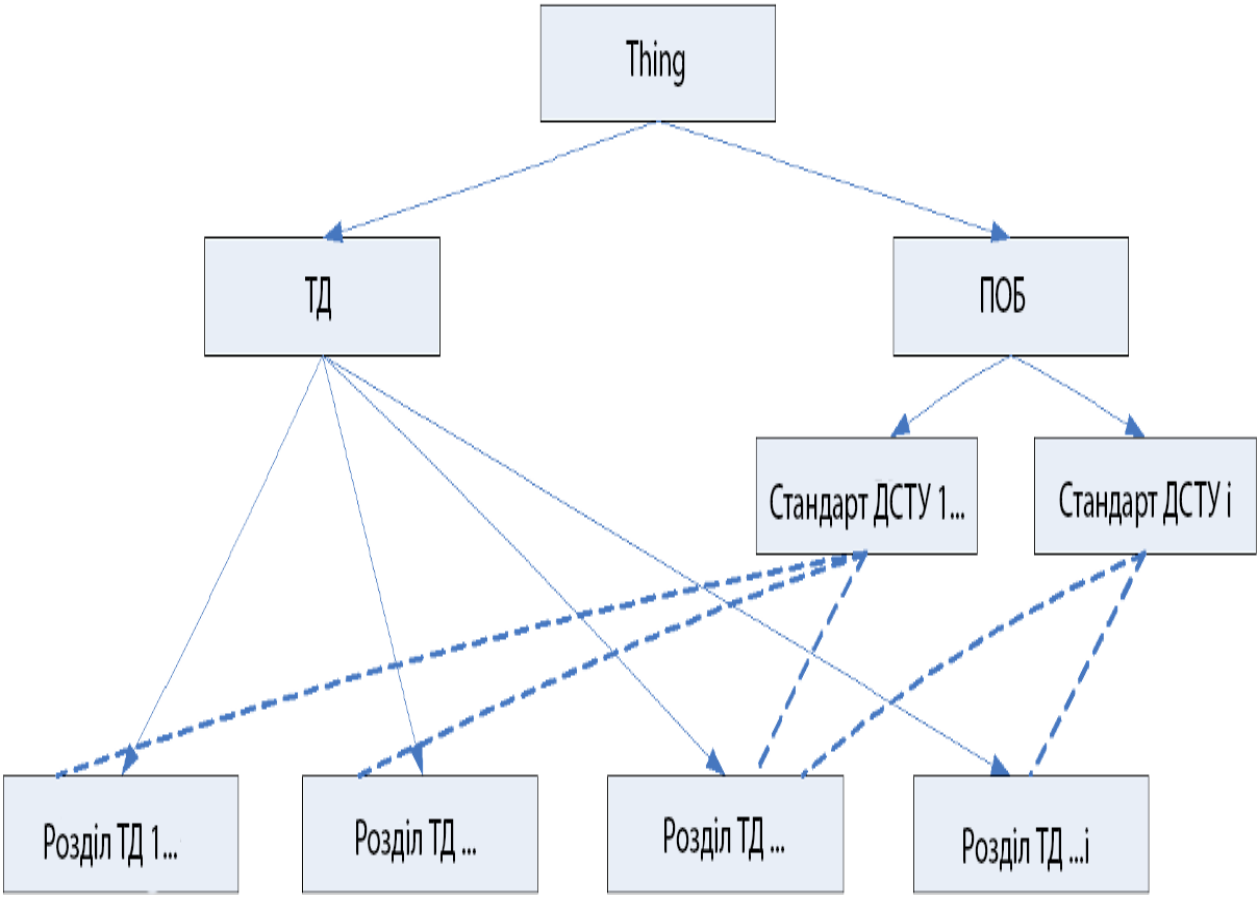

Рисунок 4.8 – Структура джерела знань (таксономія)

У побудованій онтологічної моделі джерела знань клас «Технічна документація» розглянутого джерела відповідає однойменним класу онтологічної моделі тексту. В результаті для класів «Технічна документація» та «Компонент» була отримана більш детальна класифікація, і онтологічна модель в частині, що відноситься до зазначених класів, була розширена, як це показано на рисунку 4.9.

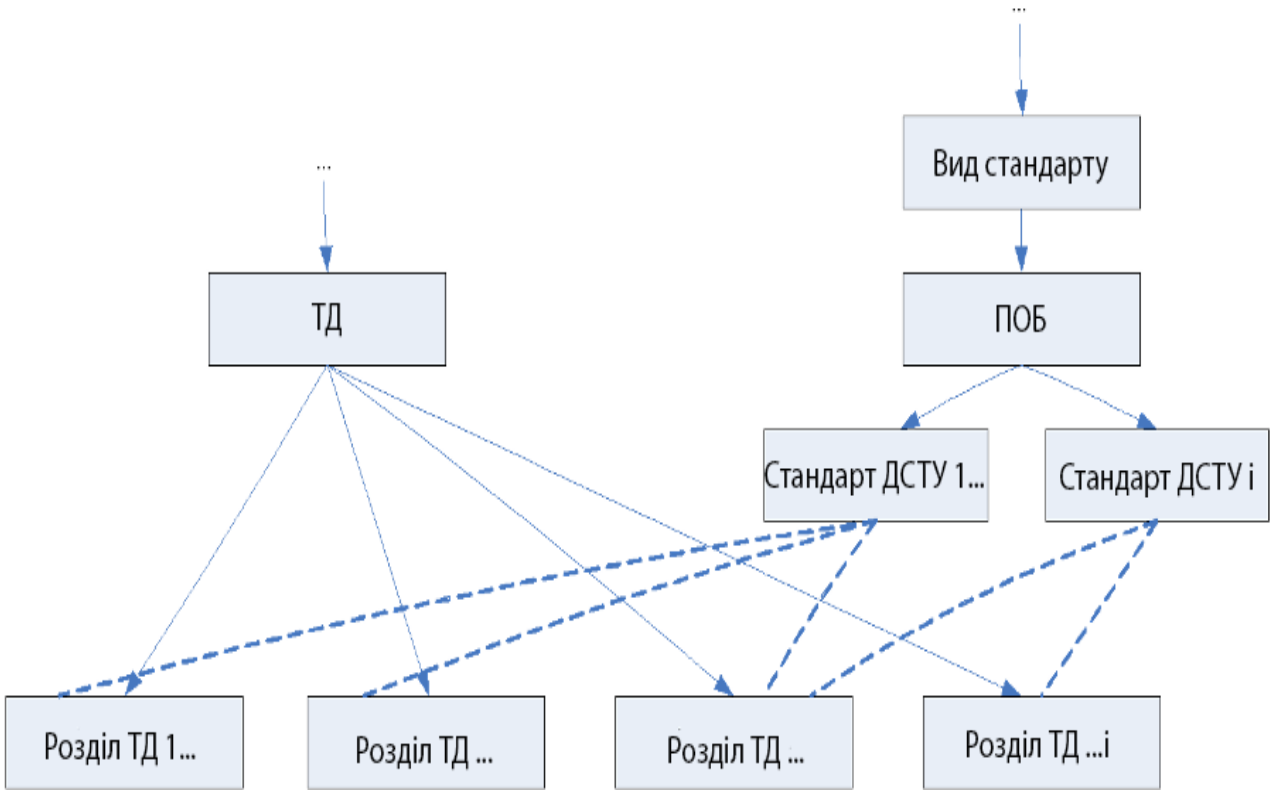

Рисунок 4.9 – Розширення онтологічної моделі

Для класу «Технічна документація» додані підкласи, відповідні розділах ТД. До класу «СТ» доданий підклас, що відповідає стандарту ДСТУ

4.4 Вибір середовища розробки

Protégé – це вільний, відкритий редактор онтологій і фреймворк для побудови баз знань. Платформа Protégé підтримує два основних способи моделювання онтологій за допомогою редакторів Protégé-Frames і Protégé-OWL. Онтології, побудовані в Protégé, можуть бути експортовані в множина форматів, включаючи RDF (RDFSchema), OWL і XMLSchema. 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.

Платформа Protégé підтримує моделювання за допомогою онтологій веб-клієнт або desktopclient. Протеже онтології можуть бути розроблені в різних форматах, включаючи OWL, RDF (S) і XML Schema. Protégé заснована на мові програмування Java, є розширюваною, і підтримує "plug-and-play" середовища, що робить її гнучкою основи для швидкого прототипування і розробки додатків.

Онтологічна модель, описує абстрактні об'єкти. Опис такого роду предметних областей, хоча і має кілька точок зору, не представляє великої складності, тому що розглянуті об'єкти вже формально описані деякою систем ой позначень. У випадку ж з об'єктами (речами) реального світу завдання коректних описів – виділення властивостей і стосунків, є більш складне завдання, тому що багато їх характеристики в розумінні людини є апріорними, тобто людина не замислюється про характер їх походження та інших властивостях. Онтологія предметної області «Формування нормативного профілю» представлена на рисунку 4.10.

В описі предметної області задано, що розглядається процес формування НП при сертифікації ПП. Отже, для формування НП, можна скористатися нечіткими атрібутними граматиками і операціями над онтологіями. Існуючі засоби онтологічного моделювання, засновані на стандартах OWL, не дозволяють автоматично отримати значення такого роду параметрів. Результат генерації коду онтологічної моделі тексту в Protege представлений в Додаток А.

Для вирішення поставленого завдання використовано розроблену онтологічну модель, яка містить в собі знання про те, які об'єкти належать ТД і СТ, як вони між собою співвідносяться і які мають взаємозалежні характеристики.

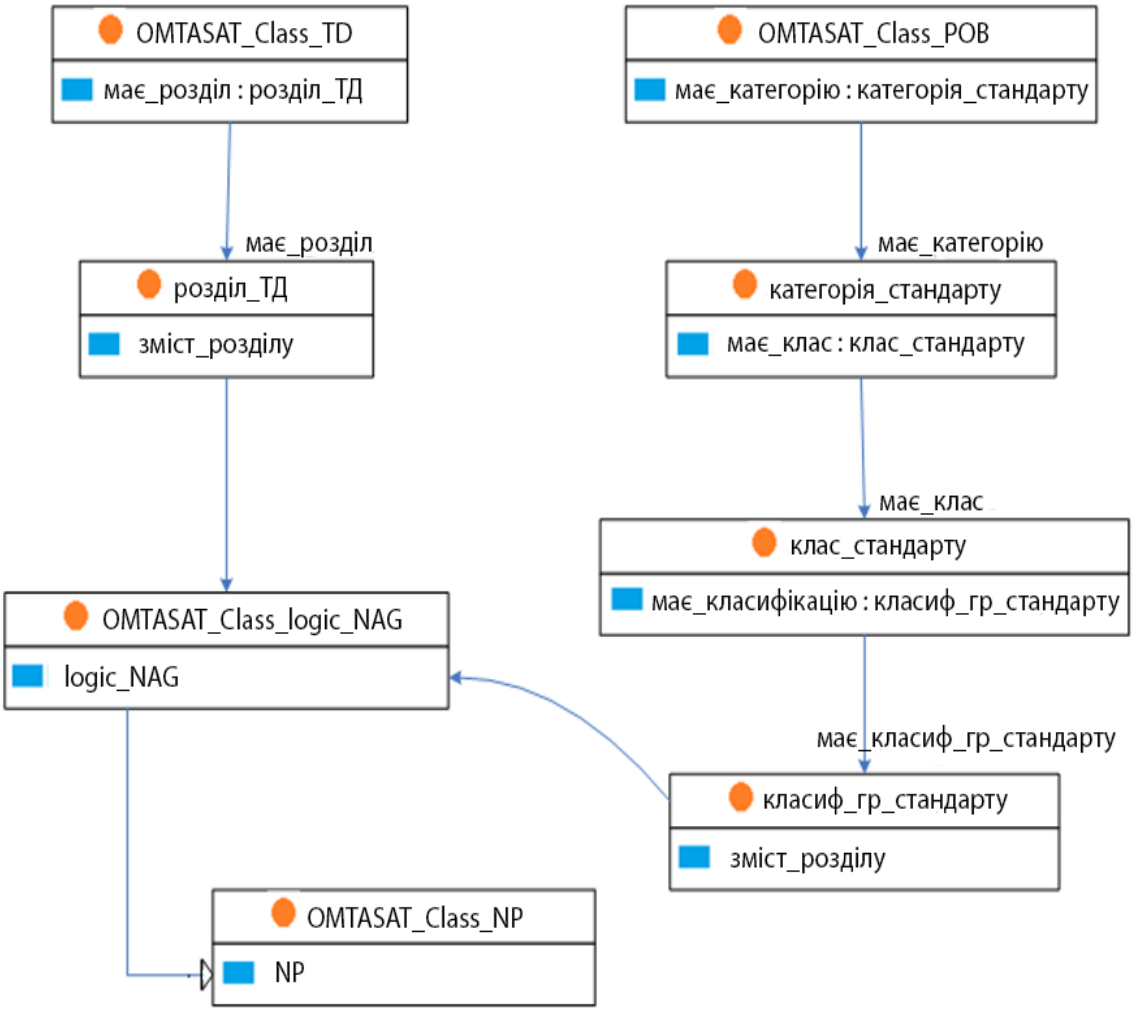

Рисунок 4.10 – Схема базових класів онтології

В результаті розробленої онтологічної моделі тексту стає можливим формування НП, шляхом семантичного аналізу тексту ТД і стандартів СТ, які описані за заданими правилами.

Для тестового прикладу ТД і СТ були взяті з державних стандартів «Єдина система програмної документації».

5 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Значно скорочується час сертифікації, що витрачається на формування НП до ТД, за допомогою використання онтологічної моделі тексту. Підвищення ефективності полягає в значному скороченні часу аналізу тексту ТД (від 36 до 57% залежно від обсягу ТД).

Таким чином, розділі розглянуто алгоритмічне забезпечення аналізу тексту ТД також алгоритми попередньої обробки тексту, семантичного аналізу тексту.

Представлений автомат розбору тексту ТД на розділи. Опис операції над онтологіями.

Та як алгоритм попередньої обробки тексту ТД ґрунтується на роботі кінцевих автоматів, то його недоліком є громіздкість опису і фіксована структура, яка задає обмеження на вхідний текст відповідно до ГОСТу.

Приклад завантаження ТД представлений на рисунках 5.1 та 5.2.

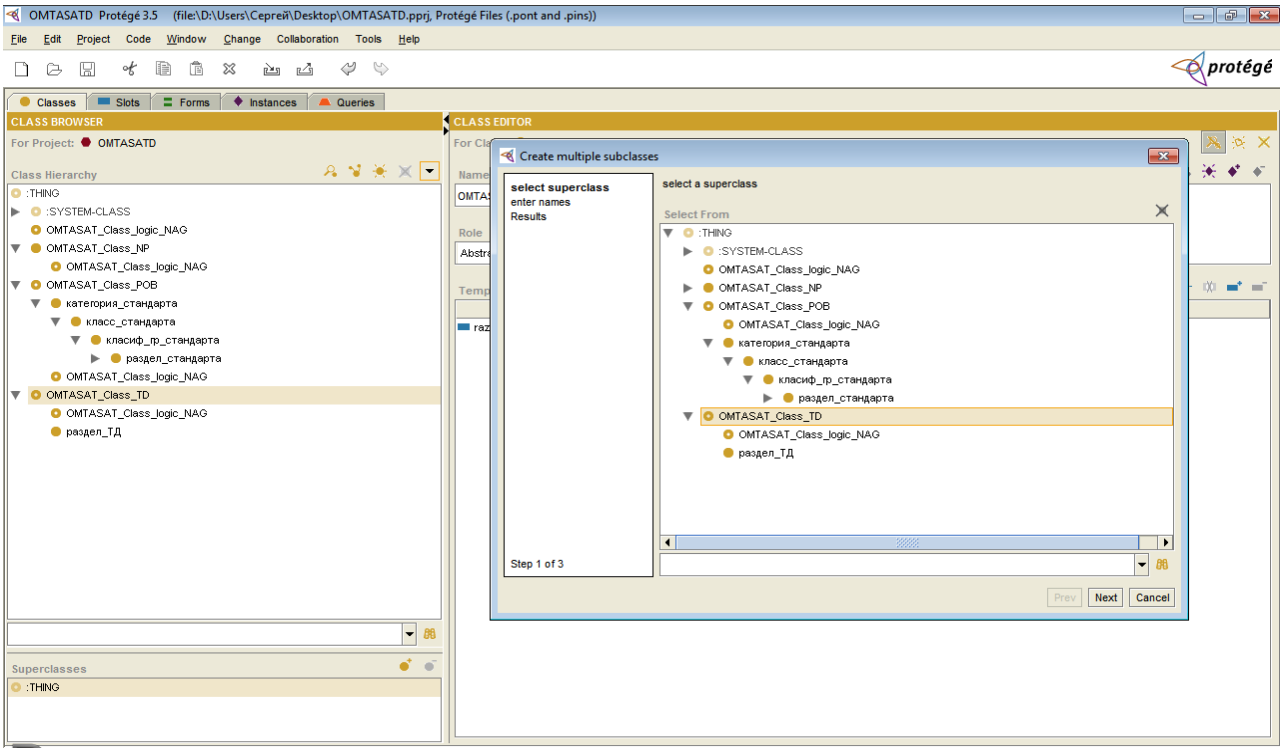

Рисунок 5.1 – Демонстрація онтологічної моделі тексту в середовищі Protégé

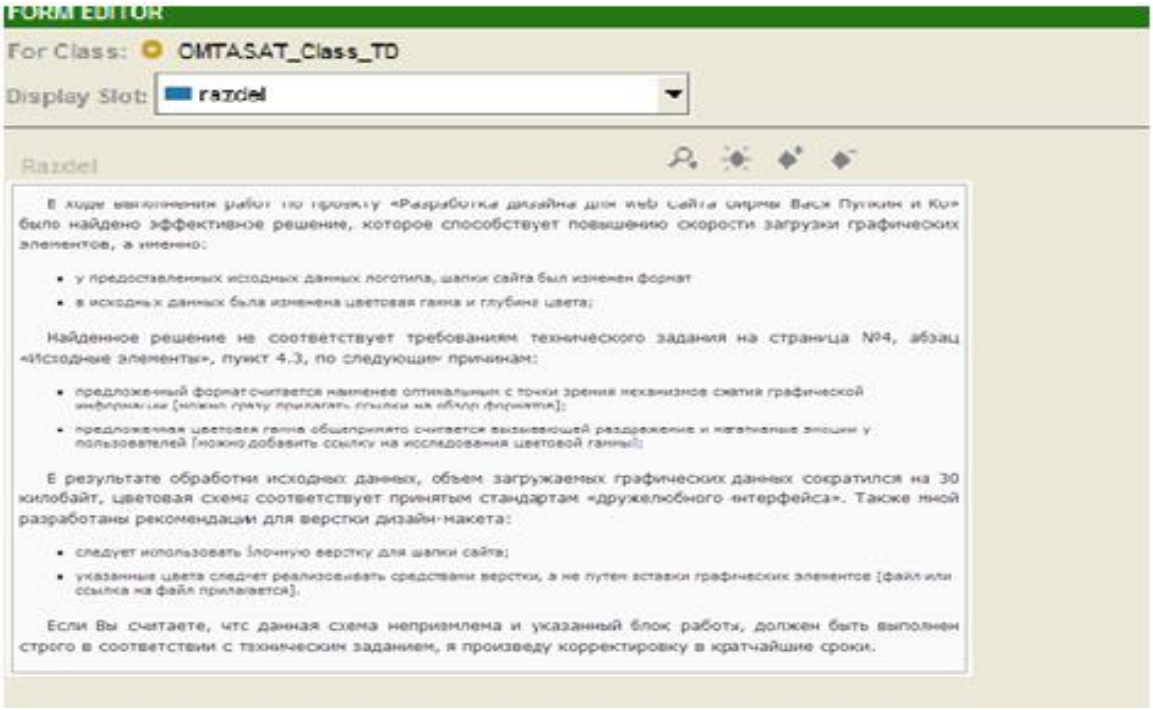

Рисунок 5.2 – Приклад розділу ТД

Подання стандартів входять до СТ, як ТД ідентичні, представлена на рисунку 5.3.

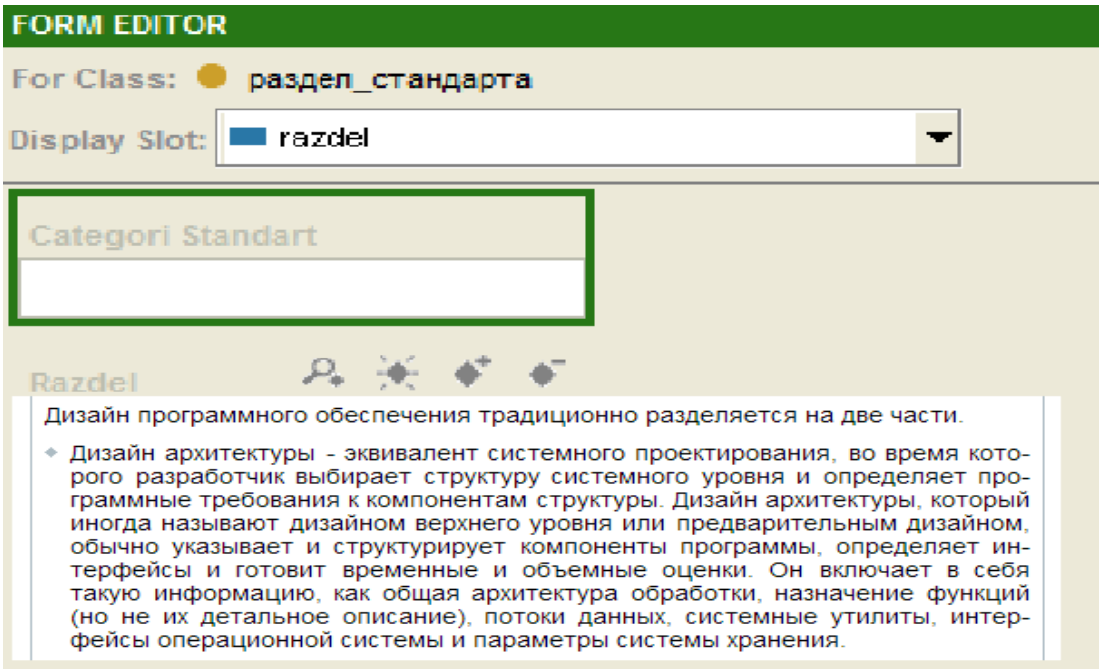

Рисунок 5.3 – Вид розділу стандарту СТ

Після проведення аналізу ТД і стандартів входять до СТ на розділи, і порівняння їх між собою, формується нормативний профіль на сертифікацію ПЗ (рисунок 5.4).

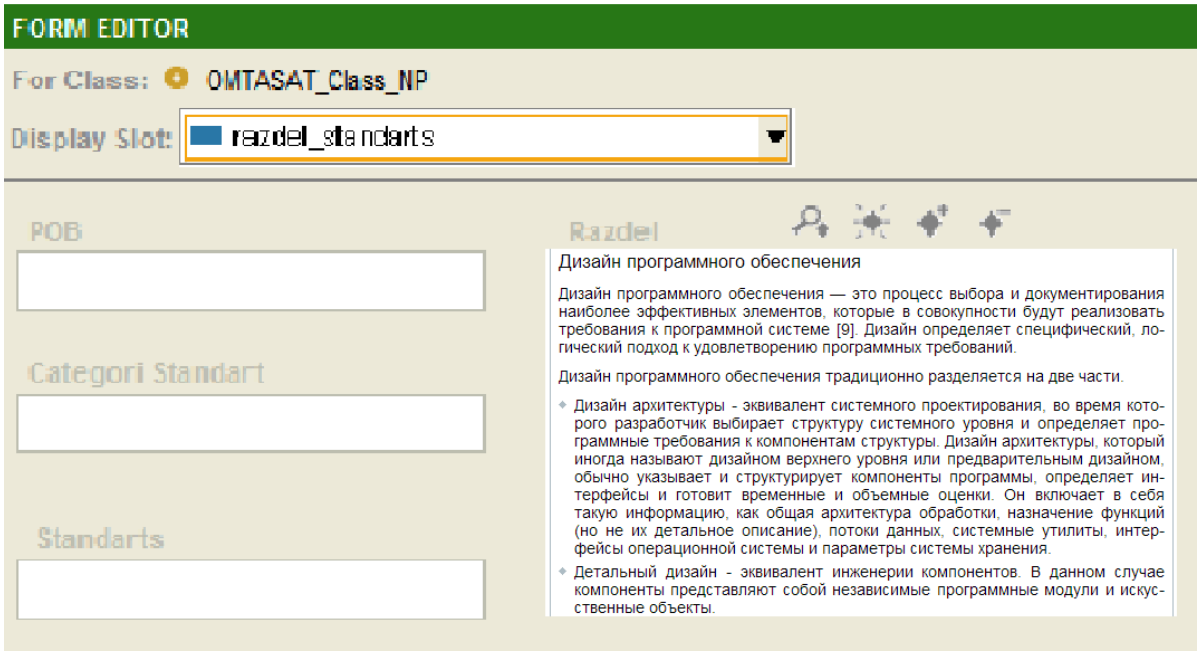

Рисунок 5.4 – Формування НП з розділу стандарту СТ

У цьому розділі показані результати тестування продуктивності розробленої онтологічної моделі тексту, що використовується для автоматизації семантичного аналізу технічної документації, при сертифікації ПЗ і формування до німу НП.

На основі експериментальних даних була виявлена залежність витрат часу аудитора сертифікаційного центру, від кількості обробленої їм ТД. Розглядаються витрати часу на аналіз ТД.

Показчики ефективності застосування онтологічної моделі тексту для семантичного аналізу тексту ТД представлені на рисунку 5.5.

Алгоритм семантичного аналізу має високу обчислювальну складність, тому аналіз проводиться повільно. Даний недолік можна виправити за рахунок подальшої оптимізації алгоритму.

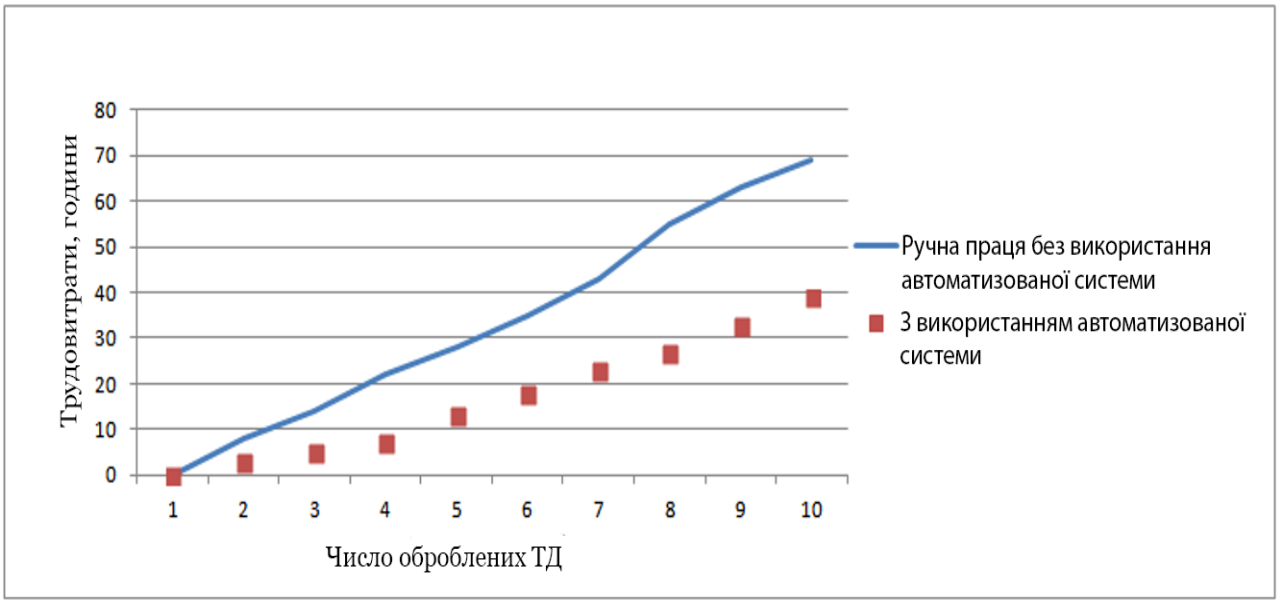

Рисунок 5.5 – Час, що витрачається на аналіз ТД

Спроектована автоматизована система семантичного аналізу тексту ТД, розроблена загальна архітектура, функціональна структура автоматизованої системи, представлена структура вихідних файлів, показані діаграми класів.

В ході роботи, був проведений аналіз онтологічної моделі тексту для автоматизації семантичного аналізу ТД. Результатом аналізу є виявлення і структурування таких аспектів системи:

– призначення онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації;

– засоби і принципи реалізації інформаційної системи;

– механізми автоматизованого формування НП на сертифікується ПЗ.

Проведено детальний аналіз онтологічної моделі тексту, формування НП, підсистеми виконання обчислень, яка реалізована в онтологічної моделі тексту.

Розглянуто середовище моделювання онтологічних структур – Protégé, як засіб розробки, що дозволяє розширювати себе для вирішення більш широкого кола завдань.

Розроблено програмами формування НП при сертифікації ПЗ, за рахунок якої реалізована можливість виробляти обчислення в онтологічної моделі

предметної області відповідно до математичними залежностями, які описані і інтегровані в онтологію з використанням запропонованої моделі та методів. Дана реалізація забезпечила можливість перевіряти несуперечність і відповідність стандартів СТ і ТД на підставі взаємозв'язку заданих характеристик.

На прикладі онтології проілюстровані можливості обробки онтологічної моделі тексту в середовищі Protégé.

Значно скорочується час сертифікації, що витрачається на формування НП до ТД, за допомогою використання онтологічної моделі тексту. Підвищення ефективності полягає в значному скороченні часу аналізу тексту ТД (від 36 до 57% залежно від обсягу ТД).

ВИСНОВКИ

В результаті розгляду можливості представленої онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації з використанням онтологічного підходу слід зробити наступні висновки:

− використання онтологічних моделей в освітній технологій дозволяє уявити множина її об'єктів і відносин між ними в явному вигляді.

− структурованість бази знань, насиченість її термінами і дефініціями дозволяють використовувати найбільш продуктивні – текстологічні методи видобування знань, які створюють передумови побудови автоматизованих процедур обробки текстів.

− висока трудомісткість побудови, необхідність наявності певних навичок створення і масовість онтологій для представлення знань в процесі сертифікації вимагають розробки спеціальних методів і алгоритмів обробки текстових документів.

Проведено аналіз існуючих проблем, пов'язаних із застосуванням онтологічних моделей тексту при автоматизації процесу семантичного аналізу технічної документації

Проведено дослідження ефективності застосування онтологічної моделі тексту для автоматизації семантичного аналізу технічної документації, а також застосування нечіткої атрибутного граматики.

Проведений аналіз систем автоматизації показує, що важливе значення в процесі прийому передачі ПЗ мають кошти сертифікації проектів ПЗ.

Побудова бази здійснюється аналітиком на основі тексту технічного завдання та існуючих стандартів є зараз не автоматизованим етапом, тому що більшу складність викликає автоматизований семантичний аналіз природної мови.

Методика аналізу тексту ТД має на увазі аналіз тексту ТД, оформленого відповідно до ДСТУ на ПЗ. Аналіз тексту при сертифікації ПЗ є ітеративним

процесом і вимагає участі людини для знаходження помилок. Ми не можемо говорити про повної автоматизації аналізу тексту ТД, а тільки про автоматизацію рутинної праці людини по вилученню корисної інформації зі стандартного документа.

Будова і властивості онтологічної моделі можуть бути ефективно досліджені і задокументовані за допомогою таких засобів: класів онтологічної моделі, які використовуються при описі характеристик об'єктів і процесів, що мають відношення до даної моделі, і класифікації логічних взаємозв'язків між цими класами. Набір цих коштів, по суті, і є онтологічною моделлю тексту, а стандарт IDEF5 надає структуровану методологію, за допомогою якої можна наочно і ефективно розробляти, підтримувати і вивчати цю онтологію.

Висунуто вимоги до онтологічної моделі тексту для автоматизації семантичного аналізу тексту технічної документації: її структурі, наповненню і функціональному призначенню.

Також виділені основні принципи створення формального опису семантичного аналізу тексту технічної документації, як зовнішнього по відношенню до онтологічної бази знань інформаційного об'єкта.

Розроблено алгоритмічне забезпечення аналізу тексту ТД, а також алгоритми попередньої обробки тексту, семантичного аналізу тексту. Представлено автомат розбору тексту ТД на розділи. Та як алгоритм попередньої обробки тексту ТД завдання ґрунтується на роботі кінцевих автоматів, то його недоліком є громіздкість опису і фіксована структура, яка задає обмеження на вхідний текст відповідно до ДСТУ.

Спроектована автоматизована система семантичного аналізу тексту ТД, розроблена загальна архітектура, функціональна структура автоматизованої системи, представлена структура вихідних файлів, показані діаграми класів.

Розглянуто середу моделювання онтологічних структур – Protégé, як засіб розробки, що дозволяє розширювати себе для вирішення більш широкого кола завдань.

Проілюстровані можливості побудови онтологічної моделі тексту в середовищі Protégé.

Побудовано графік аналізу ефективності використання автоматизованої системи, що значно скорочується час сертифікації, що витрачається на формування НП до ТД, за допомогою використання онтологічної моделі тексту. Підвищення ефективності полягає в значному скороченні часу аналізу тексту ТД (від 36 до 57% залежно від обсягу ТД).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шостак И.В., Бутенко Ю.И. Знание-ориентированные методы формирования нормативных профилей к системам критического применения на основе онтологий / И.В. Шостак, М.А. Бутенко Ю.И. // – Радіоелектронні і комп'ютерні системи.- 2010г. – С.104-107.

2. Шостак И.В. Текущее состояние базы знаний в динамических экспертных системах управления сложными объектами / Шостак И.В. // Радиоэлектроника и информатика. 2000.-№3.-С.68-71.

3. Нормативная база программной инженерии в разработке систем с интенсивным использованием программного обеспечения: учеб. пособ. / Б.М. Конарев, Л.Ф. Пудовкина, И.Б. Сироджа, О.Е. Федорович. Харьков: Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», 2001.-162с.

4. Харченко В.С. Нормирование и оценка безопасности информационных и управляющих АЭС: регулирующие требования к программному обеспечению / В.С. Харченко, М.А. Ястребенецкий, В.Н. Васильченко // Ядерная и радиационная безопасность. – 2002.-№1.-С.18-33.

5. Орлова, ЮА. Автоматизация начальных этапов проектирования программного обеспечения / ЮА.Орлова, Н.Н.Догадин //Интеллектуальные системы (AIS'06). Интеллектуальные САПР (CAD-2006): тр. Междунар. н.-техн. конф., 3-10 сент. 2016 - Т.2.- С.528-529.

6. Ермаков, А.Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста // Диалог: материалы ежегод. Междунар. конф,, Бекасово, 4-8 июня 2008 г. – М.: РГГУ, 2008. – Вып. 7 (14): Компьютерная лингвистика и интеллектуальные технологии.- С. 154-159.

7. Загоруйко, Н.Г. Система ONTOGRID для автоматизации процессов построения онтологии предметных областей // Автометрия. – , 2005. – Т.41. – № 5. -С. 13-25.

8. Загоруйко, Н. Г. Таксономия и распознавание в L-пространствах с использованием концептов Обнаружение эмпирических закономерностей. – Новосибирск: [б.и.], 1999. – Вып. 166: Вычислительные системы. – С.24-34

9. Lynn, S. Automatic Generation of Ontologies from Canonicalized Web Tables / S. Lynn, D.W. Embley. –URL: http://www.deg.byu.edu/papers/mogo.pdf.

10. Farquhar A., Fikes R., Rice J. The Ontolingua server: A tool for collaborative ontology construction // International Journal of Human-Computer Studies, 46(6), pages 707–728, 1997.

11. Musen, M. Domain Ontologies in Software Engineering: Use of Prot?g? with the EON Architecture // Methods of Inform. in Medicine, pages 540-550,1998.

12. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. V. Chaudhri, A. Farquhar, R. Fikes P. Karp J. Rice // Fifteenth National Conf. on Artificial Intelligence. AAAIPres/The MIT Press, Madison, P.600-607, 2018.

13. Creating Semantic Web Contents with Prot?g?-2000. N. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M. Musen // IEEE Intelligent Systems, March/April pages 60-71, 2019.

14. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2012.

15. Bechhofer S.,Horrocks I., Goble C., Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web // Joint German/Austrian conf. on Artificial Intelligence (KI’01). Lecture Notes in Artificial Intelligence LNAI 2174, Springer-Verlag, Berlin, pages.396-408, 2001.

16. Domingue J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web // Proc. of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, 1998.

17. Motta E. Reusable Components for Knowledge Modelling // Ph.D. Thesis. The Open University, 1997.

18. MacGregor R. Inside the LOOM classifier // SIGART bulletin, Vol.3, No.2, pages 70-76, 1991.

19. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++/ Г. Буч.- 2-е изд.- М.: Издательство Бином, СПб: Невский диалект, 1998.- 560 с.

20. Попов, Э.В. Общение с ЭВМ на естественном языке/ Э.В. Попов,- М,: Наука, 1982,- 360 с.

21. Поспелов, Д.А. Логико-лингвистические модели в системах управления/ Д.А. Поспелов.- М.: Энерго, 1981.- 231 с.

22. Кулагина, О.С, О проблемах автоматической обработки текстов на естественном языке/ О.С. Кулагина// Интеллектуальные системы.- 2006,- Т.1, вып. 1-4.-С. 109-116.

23. Преображенский, А.Б. Состояние развития систем естественно-языкового общения/ А.Б. Преображенский.- М.: Радио и связь, 1990.- 395 с.

24. Заболеева-Зотова А.В. Атрибутная грамматика формального документа "Техническое задание"// Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной техники и информатики в технических системах": Межвузовский сборник научных статей. – Волгоград: ВолгГТУ, 2008. – Вып.4, № 2. – С.39-43.

25. Лєсна Н.С., Фоменко О.А. Дослідження методів і моделей асинхронного оновлення даних для підвищення продуктивності web-систем /Наука онлайн: міжнародний електронний науковий журнал. - 2019. - №1..-С.6-10

26. Клир, Дж. Системология. Автоматизация решения системных задач: пер. с англ,/ Дж, Клир.- М.: Радио и связь, 1990.- 544 с.

27. Заболеева-Зотова, А.В. Подсистема предварительной обработки текста технического задания / Сб. науч. тр. В 17 т. Т. 3. Интеллектуальные системы и технологии /- М., 2013.-С.162-163.