НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)
“___” _____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: «Модель децентралізованої системи контролю доступу до ресурсів семантичного веб»

Виконав: студент 4 курсу, групи ФБ-51

Новоселецький Дмитро Геннадійович _____
(підпис)

Керівник _____к.т.н, доцент_____Коломицев М.В._____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)
«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Новоселецькому Дмитру Геннадійовичу

1. Тема роботи «Модель децентралізованої системи контролю доступу до ресурсів семантичного веб»

науковий керівник роботи _____к.т.н, доцент Коломицев М._____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____ » 2019 р. № _____

2. Термін подання студентом роботи 10 червня 2019 р.

3. Вихідні дані до роботи: опубліковані джерела за тематикою досліджень.

4. Зміст роботи: аналіз та синтез складних об'єктів, програмного забезпечення, пристосованих для роботи у грід-середовищі. Розробка методу виявлення порушень результатів логічних висновків.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація за дипломною роботою.

6. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Вибір тематики роботи	30.09.18	Виконано
2	Аналіз обраної тематики	21.10.18	Виконано
3	Постановка задачі	01.02.19	Виконано
4	Узгодження змісту з керівником	03.03.19	Виконано
5	Написання основної частини	30.05.19	Виконано
6	Узгодження основної частини з керівником	01.06.19	Виконано
7	Оформлення роботи	02.06.19	Виконано
8	Написання висновків	04.06.19	Виконано
9	Підготовка до доповіді	16.06.19	Виконано
10	Подання роботи на рецензію та до захисту	17.06.19	Виконано

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник роботи _____ _____
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Робота обсягом 81 сторінки містить 46 ілюстрацій та 12 літературних посилань.

Метою даної роботи є дослідження моделі децентралізованої системи контролю доступу до ресурсів семантичного веб.

Об’єктом роботи є модель контролю доступу до ресурсів семантичного веб.

Головними вимогами до системи є масштабованість, розширюваність предметної області та здатність працювати у реальному часі. Робота включає аналіз результатів практичного дослідження.

КОНТРОЛЬ ДОСТУПУ, СЕМАНТИЧНИЙ ВЕБ, RDF, БАЗА ЗНАНЬ

ABSTRACT

Diploma work consists of 81 pages contains 46 illustrations and 12 literary references.

The purpose of diploma work is to research the model of a decentralized access control system for semantic web resources.

The object of the research is the model of access control to semantic web resources.

The main requirements of the system are the scale, the expansion of substantive zone and the opportunity to work in real time. The work includes an analysis of the results of practical research.

ACCESS CONTROL, SEMANTIC WEB, RDF, KNOWLEDGE BASE

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Веб	всесвітня мережа інтернет
БД	База даних
СБД	система бази даних
СУР	система управління ресурсами
СМО	система мережевих обчислень
ПРГ	Пошук ресурсів в грід
N3	Notation 3, спосіб запису моделей RDF призначений для легкого сприйняття людиною
OWL	Web Ontology Language, мова опису онтологій для семантичного веб
RDF	Resource Description Framework, абстрактна модель представлення метаданих
SPARQL	SPARQL Protocol and RDF Query Language, мова запитів до RDF-даних
SWRL	Semantic Web Rule Language, мова опису правил для семантичного веб

ВСТУП

У зв’язку зі значною інтенсифікацією процесів розвитку, що відбуваються у терміносистемах більшості предметних галузей світової науки на сучасному етапі, терміносистеми предметних галузей піддаються певному узагальненню, набуваючи рис онтології предметної галузі. Відповідно до цього **актуальним завданням** в сучасних українських умовах є створення національної термінологічної системи, а також забезпечення її неперервного і оперативного вдосконалення й адаптації до світових стандартів, що зумовлюється необхідністю інтенсифікації міжнародних науково-технічних, виробничих і комерційних контактів. Таких властивостей національної терміносистеми можна досягти за умови, що сама термінологічна галузь задовольняє властивості мобільності, масштабованості та інтероперабельності.

Досягнути зазначених властивостей можливо лише за принципово нової інформаційно-лінгвістичної парадигми формування і функціонування терміносистеми відповідної предметної галузі, лінгво-технологічне ядро якої має складати віртуальна термінографічна лабораторія.

Очевидно, що така парадигма диктує й певні інформаційно-технологічні вимоги до функціонування віртуальних лабораторій зазначеного типу: онтолого-керованість та сервіс-орієнтованість з використанням грід- та хмарних технологій.

У роботі демонструються системи аналізу та синтезу складних об’єктів, програмного забезпечення, пристосованих для роботи у грід-середовищі. Основними функціями системи, що розроблятиметься є: виведення закономірностей, характерних для класів об’єктів, використання отримання закономірностей для вирішення задач, розвиток системи трансдисциплінарних знань у заданих предметних областях.

Мета роботи – дослідження моделі децентралізованої системи контролю доступу до ресурсів семантичного веб.

Об’єкт роботи - модель контролю доступу до ресурсів семантичного веб.

Предмет роботи – децентралізована система контролю доступу до ресурсів семантичного веб.

1 ПОНЯТТЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ ДО РЕСУРСІВ СЕМАНТИЧНОГО ВЕБ

1.1 Архітектура і організація ресурсів семантичного веб

Семантичний веб являє собою розширення існуючого вебу, суть якого полягає в тому, що даним надається чітко визначений сенс, що дозволяє чітко однозначно і повно сприймати такі дані людьми і комп'ютерами, що також сприяє вдосконаленню взаємодії людей з комп'ютерами. Таке розширення реалізується за рахунок розмітки вмісту вебу, його властивостей і відношень з використанням мов розмітки з чітко визначеною семантикою. Такі мови, звані мовами семантичного вебу, призначені для ідентифікації та подання ресурсів вебу та їх взаємозв'язків, семантичних правил, яким вони повинні задовольняти і, нарешті, здійснювати їх пошук. Далі наводиться перелік цих мов.

RDF, RDF-S. На нижньому рівні стека мов семантичного вебу розташована мова RDF (the Resource Description Framework). Вона призначена для подання інформаційних ресурсів у вигляді трійок: суб'єкт-предикат-об'єкт. Мова RDF Schema (RDF-S) призначена для опису властивостей RDF-ресурсів.

OWL/OWL2 (Web Ontology Language) - найбільш виразна мова семантичного вебу, яка призначена для більш глибокого опису семантики веб-ресурсів в термінах індивідів, класів і властивостей. У мові можна описувати нові класи на основі раніше визначених, визначати обмеження різного виду на класи і властивості, а також упорядковувати класи і властивості у вигляді таксономічних структур. Вона також має вбудований механізмом виведення.

SWRL (Semantic Web Rule Language), За допомогою аксіом OWL можна виразити безліч різних обмежень цілісності на структуру веб-ресурсів. Однак існують обмеження цілісності, що не підтримуються в OWL. У зв'язку з цим був запропонована мова опису правил SWRL, яка суттєво розширює ці можливості.

Description Logic (DL). Мова, яка базується на логіці першого порядку і яка є формальною основою OWL. По суті, є ціле сімейство мов DL. DL має чітко

визначену семантику. У мові можна виділити три складових: визначення структури, операції над структурними елементами і аксіоми.

Сервіси семантичного вебу, які також називаються семантичними веб-сервісами, це парадигма, яка інтегрує семантичні метадані, онтології, формальні засоби опису, а також інфраструктуру веб-сервісів. Семантичні веб-сервіси описуються як сервіси вебу але з використанням мови, що має чітко визначену формальну семантику. Використання семантичного вебу для опису сервісу надає можливість визначити семантику його інтерфейсу, яка, у зв'язку з наявністю її формальної специфікації, по-перше, може сприйматися комп'ютером і, по-друге, може використовуватися для взаємодії між сервісами. Зокрема, семантичні веб-сервіси використовують семантичний веб для опису:

функціонального призначення сервісів;

способів звернення до них;

правил звернення та взаємодії.

Таким чином, семантичний опис сервісів сприяє максимальної автоматизації завдань їх аналізу, пошуку, композиції, виклику, моніторингу, та управління.

Суть семантичного опису сервісу полягає в тому, що будується його онтологія. В основу структуризації онтології покладений той факт, що опис сервісу має надавати таку інформацію про нього (див. рис.1.3 нижче):

Що собою представляє сервіс? Яку послугу він надає клієнтам? Відповідь на це питання задається в «профілі сервісу».

Як використовувати сервіс? Відповідь на це питання задається в «моделі сервісу».

Як працювати з сервісом? Відповідь на це питання задається в «прив'язці сервісу». Деталі опису профілю, моделі і прив'язки сервісу можуть мінятися від сервісу до сервісу. Проте всі ці три складові повинні надавати обов'язкову інформацію відповідного типу, яка наводиться нижче.

Профіль сервісу містить інформацію, необхідну для того, щоб агент (людина або комп'ютер) міг знайти необхідний сервіс. Агент в процесі пошуку

на підставі профілю сервісу має обґрунтовано прийняти рішення, чи задовольняє даний сервіс пропонованим вимогам. Профіль сервісу включає опис того, що власне робить сервіс, які обмеження на застосовність сервісу, які якісні та кількісні характеристики сервісу, що собою являють вимоги, щоб успішно скористатися послугою, що надається сервісом.

Ця форма представлення сервісу також включає в себе опис того, що досягається за рахунок обслуговування, обмеження на застосовність сервісу, якість обслуговування і вимоги, які повинні задовольнятися, щоб скористатися послугою успішно.

Модель сервісу пояснює агенту, як скористатися сервісом і що він отримає в результаті його виконання шляхом семантичної деталізації наступних аспектів.

що собою являє вхідна інформація;

які повинні існувати обмеження, які сприятимуть успішному виконанню сервісу;

що отримується в результаті виконання сервісу;

які можливі післядії (зовнішні ефекти) в результаті виконання сервісу.

Прив'язка сервісу специфікує деталі, необхідні для того, щоб «доступитися» до сервісу. Зазвичай це протокол взаємодії, формати повідомлень, мова опису входів і виходів, номери портів. Крім того, прив'язка повинна специфікувати для кожного типу вхідного і вихідного параметра моделі сервісу однозначний спосіб обміну даними цих типів.

Під грід будемо розуміти систему, що здійснює інтеграцію, віртуалізацію і керування сервісами і ресурсами в розподіленому гетерогеному середовищі, що підтримує сукупність користувачів і/або віртуальних організацій для вирішення поставлених ними завдань [1]. Гріди надають інтегровану сукупність розподілених пулів ресурсів, які можуть динамічно розподілятися між додатками і/або сервісами, із забезпеченням необхідної ефективності, продуктивності, масштабованості. Гріди надають велику гнучкість у зв'язку з тим, що ресурси можуть динамічно перепризначатися відповідно до потреб організацій.

Існують наступні категорії грід:

Обчислювальні гріди. Гріди цієї категорії відносяться до таких систем, які націлені на надання додаткам великих обчислювальних потужностей за рахунок того, агрегована обчислювальна потужність багатьох машин виявляється вище, ніж окремих комп'ютерів. Обчислювальні гріди, у свою чергу, класифікуються на категорії:

Гріди з розподіленими суперкомп'ютерами використовуються для паралельного виконання обчислювально-складних додатків з метою скорочення часу їх роботи (наприклад, задачі гідродинаміки, прогноз погоди, молекулярне моделювання, моделювання живих організмів). У свою чергу гріди з високою пропускною здатністю збільшують продуктивність роботи тих додатків, які вимагають великих потоків даних.

Гріди даних. Грід даної категорії призначений для систем, які в процесі своєї роботи використовують дані великого обсягу. Ці дані розташовуються в репозиторіях типу баз даних, сховищ даних, електронних бібліотек, які розподілені по мережі грід. Для додатків, що працюють в обчислювальних грідах, також можуть знадобитися дані, але в цьому випадку кожне з них має реалізувати свою власну систему управління даними. Типовими додатками, що вимагають грід даних, є ті, які здійснюють інтелектуальний аналіз даних. Прикладами грід даних є CERN DataGrid [2] і Globus [3].

Гріди сервісів. Гріди цій категорії необхідні в тому випадку, коли ті чи інші послуги (сервіси) неможливо надати з використанням однієї машини. Ці гріди класифікуються на:

грід-сервіси на вимогу,

співпрацюючі грід-сервіси,

мультимедійні.

Сервіси на вимогу ініціюються клієнтом сервісу і вони працюють самостійно для виконання затребуваної послуги. Вони динамічно агрегують ресурси на час виконання сервісу і потім звільняють їх по завершенню роботи.

Співпрацюючі грід-сервіси взаємодіють з клієнтами та іншими сервісами для виконання поставленого завдання. Системи подібного роду здійснюють

взаємодію в реальному масштабі часу між людьми і додатками з використанням віртуального робочого простору.

Мультимедійні гріди надають інфраструктуру для працюючих в реальному масштабі часу мультимедійних грід-додатків.

Слід зазначити, що створення універсальної грід-системи, яка б об'єднувала ці категорії, є складним завданням.

Система мережевих обчислень (СМО) - це віртуальний комп'ютер, сформований з мережі гетерогенних комп'ютерів, які узгоджено спільно використовують свої локальні ресурси. Грід - це дуже велика швидка, СМО, що працює в середовищі Інтернет і що використовує комп'ютери, розподілені по різних організаційних та адміністративних доменах. Комп'ютери грід можуть групуватися по автономним адміністративним доменам.

Система управління ресурсами (СУР). Відповідно до даного на початку розділу 1 визначенню, в грідах однією з центральних компонент є СУР, яка функціонує спільно і узгоджено з іншими компонентами. Ресурсами в грід є обчислювальні засоби, дані, програми. Безліч послуг, пропонованих СУР, може змінюватися в залежності від передбачуваного призначення грід. Проте до основних функцій СУР належать такі:

опис ресурсів;

організація зберігання ресурсів;

організація доступу до ресурсів;

пошук ресурсів;

розподіл ресурсів серед його споживачів;

підтримка цілісності і безпеки ресурсів.

При функціонуванні СУР передбачається існування наступних компонент:

провайдер ресурсів - машина (людина, сервіс), яка надає ресурс;

споживач ресурсу - машина (людина, сервіс), яка запитує ресурс;

планувальник ресурсів - машина (сервіс), яка відповідає за розподіл ресурсів серед споживачів.

Грід-системи, як правило, мають комп'ютери, які виконують ці ролі.

У цьому розділі визначається множина понять, що мають відношення до СУР з встановленням їх таксономічної впорядкованості. Ця таксономія може використовуватися для подальшого визначення класів СУР. По суті тут робиться спроба визначення онтології СУР. Ця класифікація може використовуватися для подальшого порівняння існуючих СУР, наприклад, з точки зору їх можливостей щодо вирішення проблем масштабованості, надійності, продуктивності, тощо. Ця таксономія класифікує СУР по мережевій архітектурі комп'ютерів, моделям ресурсів, зберігання й пошуку ресурсів.

Ця таксономія класифікує мережеву організацію комп'ютерів в гріді. Згідно даної класифікації виділяються наступні моделі (див. рис. 1.4, 1.5): однорангова мережа, ієрархічна структура, клітинна організація.

Однорангова мережа. У даному випадку всі комп'ютери рівноправні. Будь-який комп'ютер в принципі може бути пов'язаний з іншим напряму, без звернення до якогось проміжного керуючого комп'ютера. Тим не менше, для кожного комп'ютера визначені його «сусіди» - тобто множина комп'ютерів, з якими він може взаємодіяти (хоча ця межа в принципі може збігатися з множиною всіх інших комп'ютерів гріда). Тут чітко простежується аналог з механізмом взаємодії веб-сервісів, який отримав назву «хореографія».

Ієрархічна структура. Ця організація комп'ютерів передбачає, що у будь-якого комп'ютера може бути тільки один його комп'ютер - предок і множина комп'ютерів - нащадків. Сам же комп'ютер є нащадком для свого предка і предком для своїх нащадків. У цій архітектурі комп'ютер може взаємодіяти тільки зі своїм предком і з нащадками. Предок, крім усього іншого, відповідає за управління (координацію) роботи всіх своїх нащадків. Дана організація взаємодії по суті є аналогом механізму взаємодії веб-сервісів, який отримав назву «оркестровки».

Клітинна організація. У цій архітектурі комп'ютери об'єднуються в множину клітин. Комп'ютери всередині клітини мають однорангову мережу. Серед комп'ютерів клітини виділяється один або більше, які формують так званий кордон клітини і які відповідають за взаємодію даної клітини з іншими.

Тільки граничні комп'ютери можуть взаємодіяти з іншими клітинами. Самі ж клітини в цілому, знову ж, можуть структуруватися або у вигляді однорангової мережі, або в ієрархію. Знову ж, клітинна організація комп'ютерів в гріді є аналогом підходу у веб- сервісах, коли будь-який додаток, що використовує при своїй роботі множину веб-сервісів, може бути оголошено як веб-сервіс. Слід зазначити, що ця структура також називається суперструктурованою одноранговою мережею.

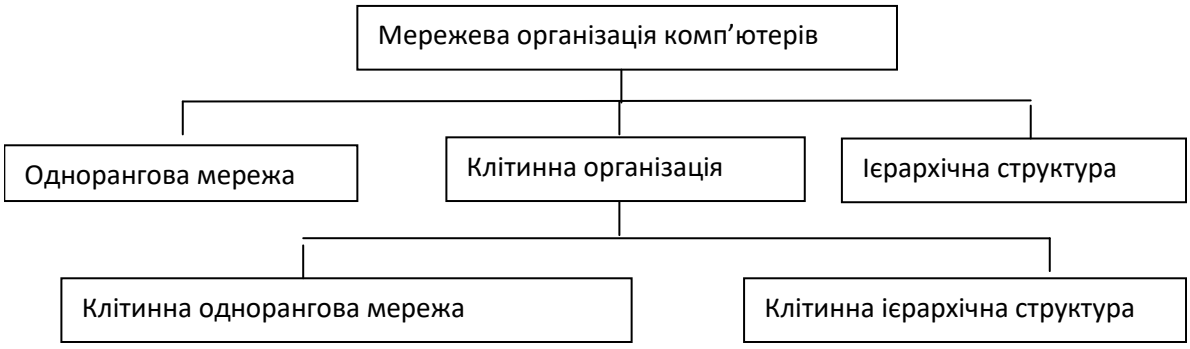
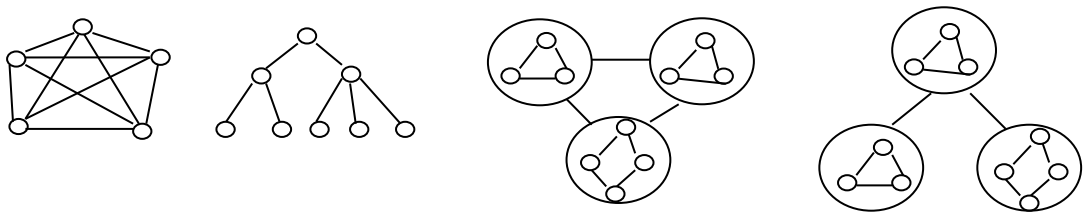


Рисунок 1.1 - Таксономія мережевої організації комп’ютерів в грідах



A B C D

Рисунок 1.2 - Мережева організація комп’ютерів в грідах: A – одноранговa мережа; B – ієрархічна структура; C – клітинна одноранговa мережа; D – клітинна ієрархічна структура

Модель ресурсів визначає, яким чином СУР описує і управляє ресурсами. На рис 1.3. наводиться таксономія моделі ресурсів. В якості основи побудови цієї таксономії узятий спосіб опису і взаємодії структурної та операційної складових моделей.

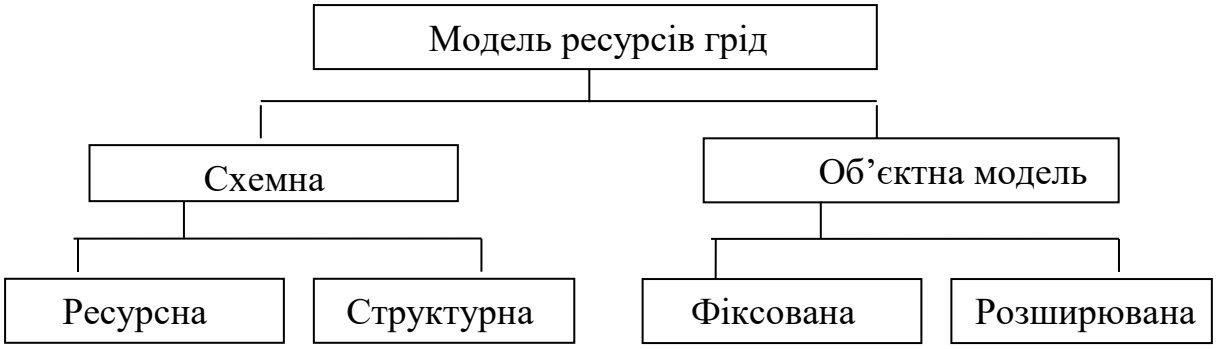


Рисунок 1.3 - Таксономія моделей ресурсів грід

У схемному підході структура та операції існують самостійно. У цьому випадку існує мова опису структури ресурсів і обмежень цілісності, що накладаються на структуру. Опис структури називається схемою структури ресурсу. У деяких випадках мова опису схеми інтегрується з мовою запитів для того, щоб можна було описувати з його допомогою структури, що виводяться з існуючих. Виділяється два різновиди схемного підходу на основі розширюваності моделі: фіксована і розширювана схема. Фіксований підхід передбачає, що мова опису не надає можливості з визначення нових типів структурних елементів. У розширюваному варіанті можна конструювати нові типи об'єктів і використовувати їх для опису структури ресурсу.

У об'єктному підході операції над ресурсами розглядаються як складова і невід'ємна частина опису ресурсу. Як і в схемному підході, тут також виділяються фіксований і розширюваний варіанти з тією ж семантикою. Реалізація розширюваної об'єктної моделі ресурсів грід є дуже складним завданням. В даний час використовуються в об'єктних моделях використовуються тільки фіксований варіант з наданням невеликої кількості примітивних-операцій над ресурсами.

Виділяється два способи зберігання ресурсів грід: мережеві каталоги та розподілені об'єкти. Різниця між цими підходами така ж, як і у схемному і об'єктному підходах опису моделі ресурсу. Мережеві каталоги припускають, що опис зберігання і маніпулювання збереженими ресурсами - це дві самостійні

незалежні складові. У підході з використанням розподілених об'єктів визначення зберігається об'єкта включає опис його структури і маніпулювання їм.

Важливою характеристикою грід є можливість мережевого пошуку необхідних ресурсів грід-додатками. На рис 1.4. наводиться таксономія пошуку ресурсів. Виділяється два підходи з організації пошуку: з використанням запитів і з використанням агентів.

Підхід, орієнтований на запит. Він передбачає, що пошук формується у вигляді запиту і що існує механізм інтерпретації запитів. Далі пошук за запитом може бути централізований і розподілений. Розподілений варіант передбачає організацію пошуку за запитом в багатьох вузлах грід. Централізований варіант передбачає існування єдиного вузла, в якому зберігається описова інформація про ресурси та їх пошук здійснюється саме там. Подальше ж використання ресурсу може проводитися за місцем його розташування. Другий варіант має свій аналог у вигляді реєстру сервісів в сервіс-орієнтованій архітектурі. У централізованому підході існує комп'ютер (сервіс), який реалізує функцію розподілу ресурсів. Розподіл проводиться не описів ресурсів (яке необхідно для знаходження ресурсів) а власне ресурсів, що надаються провайдерами і розташованих у вузлах провайдерів.

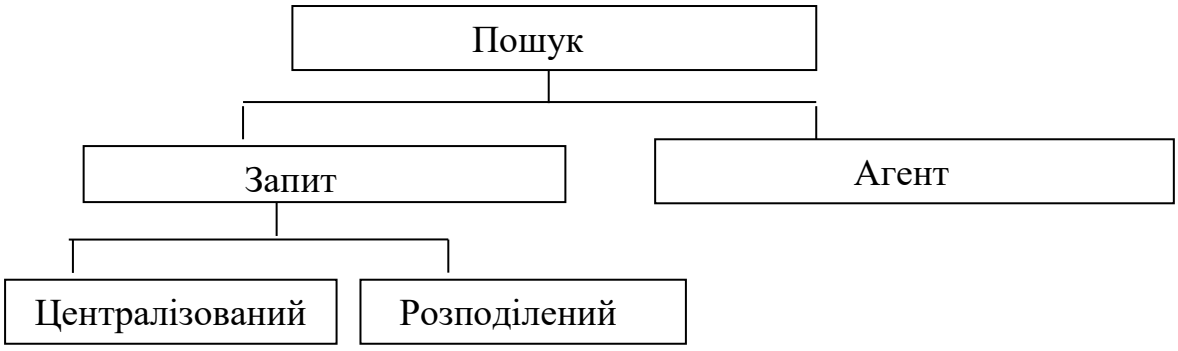


Рисунок 1.4 - Таксономія пошуку ресурсів в грід

У ієрархічному підході існує ієрархічно організована структура комп'ютерів (сервісів), які відповідають за розподіл «підконтрольних» їм ресурсів.

Агентно-орієнтований підхід. Він передбачає відсилання активних програмних фрагментів по машинах гріда, які інтерпретуються там локально відповідними механізмами. Агенти на місцях можуть моніторити ситуацію, пов'язану з ресурсами і потім відсилають інформацію про ресурси або періодично, або на вимогу. Таким чином, агенти можуть імітувати підхід пошуку, орієнтований на запит. Проте, суттєва відмінність цих двох підходів полягає в тому, що агенти можуть керувати як самим пошуковим запитом, так і процедурою пошуку на основі своєї власної логіки. Агентно-орієнтований підхід завжди є розподіленим.

У децентралізованому підході відсутні спеціально виділені сервіси з розподілу ресурсів. Провайдери і споживачі ресурсів самостійно вирішують проблему розміщення і розподілу ресурсів.

Питання пошуку ресурсів грід залежно від архітектури мережі обговорюються нижче.

Наведені вище класифікаційні схеми перш за все використовуються для опису існуючих СУР. Наприклад, в роботі [4], яка була взята за основу цієї класифікації, аналізується біля 20 СУР грід.

Одним з важливих складових СУР є пошук ресурсів грід.

Пошук ресурсів в грід (ПРГ) може бути визначений як процес пошуку і визначення місця розташування ресурсів серед множини адміністративних доменів на підставі опису необхідного ресурсу, сформульованого конкретним додатком. У результаті механізм пошуку формує і повертає список знайдених ресурсів з можливим зазначенням сервісів, що маніпулюють знайденими ресурсами. При пошуку використовуються механізми зіставлення. До механізму ПРГ висуваються такі вимоги:

повнота і релевантність результатів пошуку;

прийнятний час виконання пошуку;

здійснення пошуку у динамічному і масштабованому середовищі.

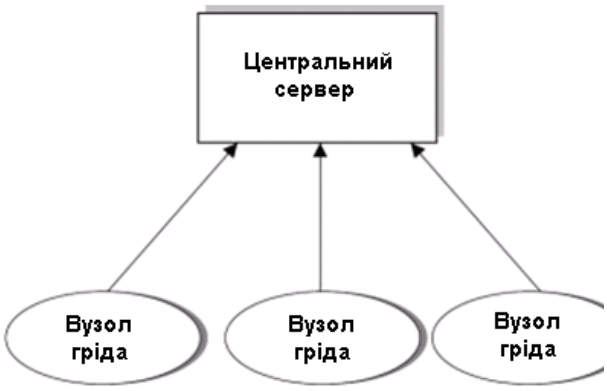
Ми вже привели два різновиди пошуку ресурсів в грід - орієнтовані на запит і на агента. У даному розділі пошук розглядається з іншого боку.

Розглядаються різні моделі і стратеги пошуку ресурсів з урахуванням мережевої архітектури технічних засобів.

До них відносяться централізована і ієрархічна моделі ПРГ, які широко використовуються в програмно-апаратному забезпеченні грід (Grid Middleware) вже багато років.

Централізована модель ПРГ (Рис. 1.6.) грунтується на концепції, що інформація про ресурси грід зберігається в єдиній централізованій базі даних. У зв'язку цим полегшується процедура ведення цієї інформації. Формулювання пошуку проводиться на єдиній мові запитів, яка може мати достатню повноту (наприклад, SQL). Однак, наявність єдиної точки зберігання і підтримки описів ресурсів стає вузьким місцем при їх пошуку при великих навантаженнях. Таким чином, централізована архітектура ПРГ не підходить для суттєво розподіленої архітектури грід. Зокрема, в цьому випадку ставляться під загрозу масштабованість і надійність в доповненні до зниження як ефективності використання ресурсів, так і їх доступності. Також знижується надійність і відмовостійкість.

Ієрархічна модель ПРГ (Рис. 1.7.) широко використовується в сучасних грід. Ідея організації ієрархічного пошуку полягає в тому, що використовується множина індексуючих серверів, які упорядковуються у вигляді ієрархічної структури.



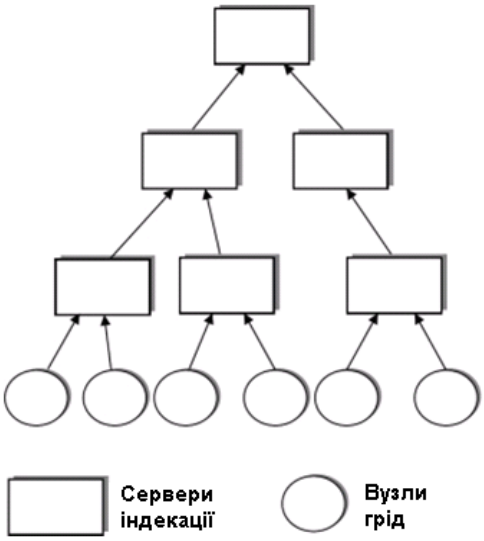


Рисунок 1.6 - Централізована модель ПРГ

Рисунок 1.7 - Ієрархічна модель ПРГ

Ця модель організації пошуку добре підходить для грід середніх розмірів, проте стає непрактичною для великих багатодоменних грід. У зв'язку з цим додається багато зусиль щодо вдосконалення цього підходу. Тим не менше, для великих грід виникають проблеми з продуктивністю, зростанням часу реакції, автономністю та інші. У зв'язку з цим пропонуються інші архітектури ПРГ, які описуються далі.

Однорангові мережі використовуються в грід для організації пошуку ресурсів. Є такі різновиди однорангової моделі ПРГ:

неструктурована однорангова модель ПРГ;

структурована однорангова модель ПРГ;

суперструктурована однорангова модель ПРГ.

Назви цих моделей пошуку базуються на назвах відповідних мережевих структур, на яких реалізується ПРГ.

Суть неструктурованої однорангової (P2P) моделі ПРГ полягає в тому, що описи ресурсів публікуються всім партнерам без використання будь-яких глобальних характеристик або наявності будь-яких глобальних структур. Суть пошуку полягає в тому, що вузол посилає запити всім своїм партнерам. Ті ж, у свою чергу, або безпосередньо відповідають на запит, якщо це в їхніх силах, або далі відсилають цей запит своїм партнерам і т.д.. Таке хвилеподібний поширення запиту гарантує знаходження необхідного ресурсу, якщо він існує в гріді. Схематичне представлення цього пошуку наведено на рис. 1.8.

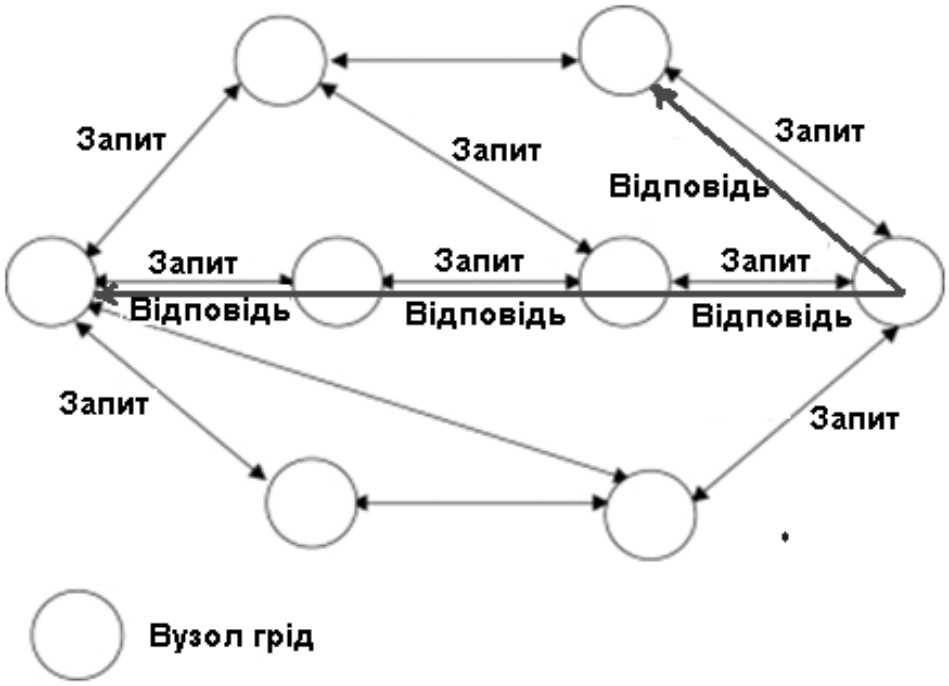


Рисунок 1.8 - Неструктурована однорангова модель ПРГ

Щоб знизити навантаження на трафік і щоб мінімізувати час відповіді пропонуються різні стратегії розповсюдження запитів, стислий виклад яких наводиться нижче.

Стратегія встановлення радіусу пошуку. Радіус - це лінійна послідовність партнерів, серед яких поширюється запит. Природно, що запит може хвилеподібно поширюватися серед всіх партнерів. Якщо загальний діаметр мережі грід відомий, то можна ефективно визначити оптимальний радіус пошуку. Це підхід вирішує проблему масштабованості і мінімізує трафік в мережі.

Стратегія випадкового обходу. Ця стратегія є популярною альтернативою хвилеподібного пошуку в P2P-мережах. Вузол мережі посилає N запитів своїм випадково вибраним сусідам. Кожен з таких N запитів називається випадковим обхідником. Цей обхідник аналогічним чином переміщується випадковим чином серед інших сусідніх вузлів і періодично консультується з тим вузлом, який ініціював запит, на предмет того, чи слід продовжувати обхід.

Стратегія ранжируваних маршрутів. Згідно цієї стратегії по запиту будується список сусідів вузла, який ініціював запит, впорядкованих згідно їх

ступеня відповідності запиту. Потім ця процедура виконується для всіх сусідів вихідного вузла. Таким чином будується безліч ранжируваних шляхів. В якості критерію відповідності вузла запитом може використовуватися різна статистична інформація. Ця стратегія може відноситися до класу «здатних до навчання» в тому сенсі, що в міру відпрацювання запитів ПРГ статистична інформація, що накопичується, може змінювати порядок ранжирування шляхів для одного і того ж запиту.

Підводячи підсумки короткому опису неструктурованої P2P-моделі ПРГ можна відзначити наступне. Вона збільшує масштабованість і відмовостійкість порівняно з централізованою та ієрархічною моделями. Далі, можна в деякій мірі знизити навантаження трафіку. Проте ефективність пошуку не є достатньо високою. Цей метод зручний для застосування багатоатрибутного, послідовного пошуку і пошуку за ключовими словами.

Ідея структурованої однорангової P2P-моделі ПРГ полягає в тому, що грід-вузли та ресурси відображаються в деякі ключі за допомогою функції хешування. І ці вузли/ресурси організовуються в жорстку структуру згідно їх ключів. Функції пошуку безпосередньо використовують таку структурованість вузлів для знаходження ресурсів. Одним із прикладів структурованості вузлів є хордова P2P-структура (рис. 1.9).

Структурована P2P-модель ПРГ задає жорстку структуру вузлів і використовує строгий механізм індексування ресурсів. Побудовані індекси використовуються при пошуку.

Ця модель добре підтримує масштабованість, пов'язану зі збільшенням розміру гріда, однак саму структуру підтримувати складно. Іншим аспектом функціонування структурованої P2P-моделі є підтримання збалансованої навантаження у великомасштабних грід. Крім того, тут доводиться мати справу зі складними пошуковими запитами, так як ресурси грід повинні публікуватися і відшукуватися з використанням багатоатрибутних та інтервальних значень.

Основна ідея цієї суперструктурованої однорангової моделі ПРГ полягає в наступному: великомасштабна грід може бути представлена у вигляді мережі

відносно невеликих грід з вузлами партнерами, що підтримують P2P архітектуру, і побудованими за принципом групування деяких адміністративних доменів, які об'єднують логічно взаємопов'язані організаційні структури. Це так звана клітинна організація. У кожній клітці виділяється один або більше вузлів, які виконують функції так званих супер-партнерів (супер-вузлів). Через супер-партнерів здійснюється зв'язок з іншими клітинами, зокрема для організації ПРГ. Ця архітектура представлена графічно на рис. 1.10.

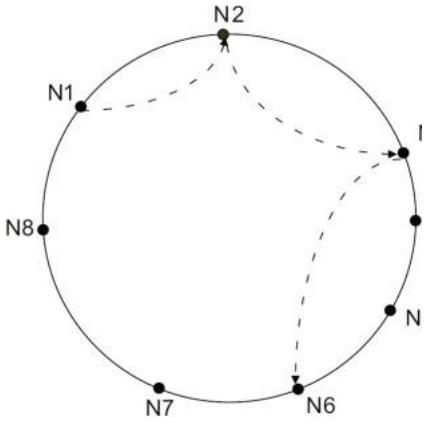


Рисунок 1.9 - Хордова P2P-структура

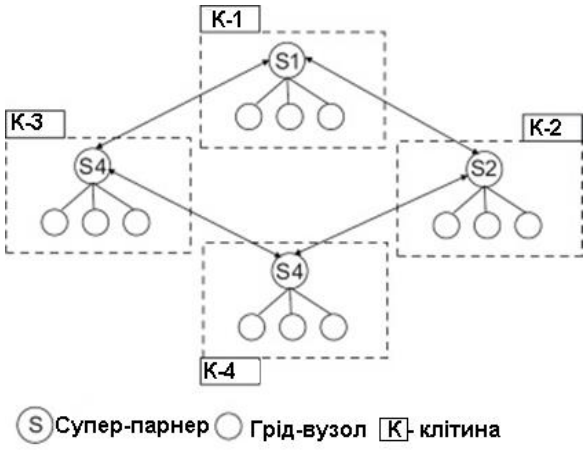


Рисунок 1.10 - Суперструктурована P2P-модель ПРГ

Ця модель по суті є дворівневою у зв'язку з введенням супер-партнерів, через які здійснюється зв'язок з іншими клітинами. Модель грід нижнього рівня може бути неструктурованою або структурованою. На цьому рівні можуть застосовуватися відповідні стратегії пошуку. Що стосується мережі клітин, то вона також може бути неструктурованою або структурованою.

Було запропоновано безліч алгоритмів пошуку в такій моделі [4]. Ця модель добре справляється з масштабуванням. Так як супер-вузол управляє своїми власними ресурсами і ресурсами своєї клітки, то проблема пошуку на рівні клітини вирішується ефективно. З іншого боку, супер-вузли стають вузьким місцем при обробці великої кількості запитів.

Суть цієї моделі полягає в наступному. Всі вузли грід об'єднуються в грід-спільноти за ознакою подібності ресурсів. Спільноти грід можуть мати будь-яку із запропонованих вище структур (централізовані, децентралізовані,

структуровані P2P і т.д.). Процедура пошуку розбивається на два етапи. Спочатку визначається співтовариство, яке може містити необхідний ресурс, а потім здійснюється пошук усередині співтовариства. Ключовою проблемою в цій архітектурі є механізм формування співтовариств. Пропонуються наступні стратегії створення співтовариств.

На основі категорії ресурсу. Ресурси однієї категорії об'єднуються один кластер. Категоріями ресурсів можуть бути, наприклад, дані, програми, обчислювальні засоби і т.д. Принцип категоризації і власне класи категорій залежать від функціонального призначення грід.

За відповідями на запити. В одну групу об'єднуються вузли, які раніше давали однакові або схожі відповіді на одні й ті ж запити.

1.2 Доступ до ресурсів семантичного веб

Дослідження і розробки в галузі семантичного вебу призвели до того, що ця технологія була сприйнята в грідах, що призвело до поняття семантичного гріда. Семантичний грід згідно з визначенням Де Рура [5] «є розвитком існуючого грід в напрямку, згідно з яким інформація та сервіси грід представляються з використанням чітко визначеного сенсу, що дозволяє істотно поліпшити взаємодію людей і машин у вирішенні поставлених завдань». Результати семантичного вебу насамперед застосовуються в грідах для вирішення проблеми управління всіма видами ресурсів, включаючи і їх пошук.

У грідах користувачі та програмні агенти повинні мати можливість знаходити, викликати, компонувати і відстежувати роботу вузлів гріда, які пропонують необхідні сервіси та ресурси. Опис цих ресурсів і сервісів за допомогою технології семантичного вебу призводять до розуміння їхнього змісту. Це допомагає планувальнику грід вирішувати задачу знаходження необхідних ресурсів, навіть якщо вони і не абсолютно точно відповідають тому, що відшукує агент.

У наступних двох розділах розглядаються використання онтологічного підходу для опису задач пошуку ресурсів і сервіс-керований підхід функціонування грід.

Робота грід полягає в скоординованому спільному використанні всіх видів ресурсів грід для вирішення різних обчислювальних завдань у динамічному, розподіленому і неоднорідному середовищі. Щоб успішно вирішувати поставлені завдання, слід мати можливість ефективно приймати оптимальні рішення щодо всіх ресурсів, які необхідні для цього (обчислювальні, програмні, інформаційні). Наприклад, щоб вибрати найкращий метод і програму, яка вирішує поставлене завдання згідно з вибраним методом, необхідно мати можливість якомога детальніше і повно описувати як самі методи, так і програми. Більше того, бажано мати також механізми виведення, які б дозволяли принципово нову інформацію про методи і програмах на основі тієї, яка є в їх описах. Для цього краще всього використовувати онтологічний похід до опису ресурсів грід (обчислювальних, програмних, інформаційних). Онтології в порівнянні з традиційними способами описи:

надають всім, хто буде брати участь у використанні грід, загальне і однозначне розуміння суті ресурсів і сервісів грід через використання єдиної системи понять і їх взаємовідношень;

надають міцну основу представлення смислу понять і їх зв'язків. Володіють достатньою гнучкістю щодо оперативної зміни наших знань про предметну область;

суттєво покращують якість і повноту інформації про ресурси грід;

дозволяють представити цю інформацію у формі, що читається і розуміється машиною;

дозволяють спільно використовувати інформацію про ресурси, що однозначно розуміються усіма агентами грід;

дозволяють повною мірою використовувати принцип повторного використання;

сприяють вирішенню проблеми інтероперабельності даних в грід, нарешті,

дозволяють підключати потужні механізми виведення, які надають гнучкі та ефективні способи міркувань про ресурси грід.

Онтології є фундаментальними будівельними конструкціями семантичного грід, Онтології дозволяють не тільки описувати самі ресурси, він їх розміщення, розподіл по мережі, способи зберігання і способи доступу. Нижче наводяться приклади варіантів використання онтологій в грідах.

У роботі [6] пропонується проводити семантичне розширення традиційних грід за рахунок онтолого-орієнтованого представлення метаданих грід і, тим самим, полегшити та вдосконалити завдання контекстного пошуку ресурсів, який є складовою частиною компоненти управління ресурсами.

У статті [7] описується архітектура GODIS (Grid Ontological Directory and Integration System) - онтологічної системи для спільного використання та виявлення ресурсів у великомасштабних грідах. У цій архітектурі кожен вузол має власну онтологію ресурсів, яка інтегрується в онтологію співтовариства (віртуальної організації) з тим, щоб розкрити семантику спільно використовуваних ресурсів.

У статті [8] обговорюється проблема використання онтологій в грідах в цілому і різні варіанти її використання.

У статті [**Error! Reference source not found.**] розглядається варіант використання онтологій при пошуку ресурсів, а саме при вирішенні завдання зіставлення онтології ресурсу з пошуковим запитом. Проблемі зіставлення онтологій запитів і ресурсів також присвячена стаття [10], в якій визначено і використовується поняття замкнутих співставлень, що настроюються..

У статті [11] описується онтолого-орієнтована модель публікації та знаходження ресурсів грід (Grid Resource Publication and Discovery-GRPD). Пропонується використання множини онтологічних реєстрів для кожного домена для управління відповідними ресурсами у віртуальній організації з метою отримання високоефективної системи GRPD. Опису ресурсів і запитів ресурсів проводиться з використання доменно-орієнтованих онтологій.

Далі коротко розкриваються питання семантизації раніше розглянутих моделей ПРГ.

Основна суть семантизації пошуку ресурсів в грід полягає у використанні онтологій ресурсів і на підставі цього підключення більше інтелектуальних механізмів пошуку. Використання онтологій дозволяє визначати семантичні моделі ресурсів а також семантичні моделі реєстрів ресурсів. Таким чином, при розподілі запитів по вузлах грід та їх виконанні використовується семантична інформація реєстрів. Далі описуються семантичні моделі та ПРГ.

Централізована модель ПРГ. У цій моделі онтології ресурсів (семантична інформація) всієї системи зберігається і індексується в центральному реєстрі ресурсів та запити ПРГ відсилаються і відпрацьовуються в цьому центральному семантичному реєстрі. Наприклад, в роботі [12] описується агентно-орієнтована система, що базується на централізованій архітектурі, в якій використовується спеціальний механізм зіставлення онтологій ресурсів у реєстрі із запитами клієнтів.

Ієрархічна модель ПРГ. В ієрархічній моделі сама ієрархічна структура стає онтологією, в якій задається спеціальне онтологічне відношення між вузлами ієрархії реєстрів. Зазвичай це стандартне родо-видове відношення (IsA) і кожен видовий вузол успадковує властивості (онтологію) родового вузла. Наприклад, в роботі на базі такої моделі пропонуються персоналізовані семантико-орієнтовані інформаційні сервіси грід (Personalized and Semantics - based Grid Information Services - PS-GIS), які виконують функцію управління інформацією для вирішення завдань публікації, пошуку і моніторингу сервісів.

P2P-модель ПРГ. У цій моделі кожен партнер веде семантичну інформацію (онтології) своїх локальних ресурсів, а також може підтримувати або звертатися до онтології своїх сусідів. Наприклад, в роботі на базі P2P-моделі пропонується наступний підхід до використання семантичних ресурсів в грід. У гріді відсутній спеціальний вузол з централізованою онтологій, що використовується для опису та пошуку ресурсів. Замість цього кожен P2P - вузол має свою власну, можливо неповну онтологію, яка може поповнюватися інформацією з мережі. Таке

динамічне формування онтологій дозволяє розширювати опис ресурсів навіть тією інформацією, яка не була надана їм їх провайдерами. У роботі вводиться поняття семантичного спільноти, яка, по суті, означає створення онтологій клітин суперструктурованої P2P моделі ПРГ. Спільноти створюються за принципом однорідності семантичних властивостей вузлів спільноти. Всі завдання ведення співтовариства вирішуються децентралізовано в самому співтоваристві.

Модель групової кластеризації. Кожна група має свій онтологічний реєстр. Всі ці реєстри об'єднуються в єдиний реєстр верхнього рівня. Запити ПРГ відсилаються до реєстру верхнього рівня і потім вони потрапляють у відповідний реєстр нижнього рівня. Наприклад, в роботі пропонується підходяща для даної моделі концепція мережі семантичних зв'язків (Semantic Link Network - SLN) для організації ресурсів у грідах знань.

У роботі наводиться порівняльний аналіз всіх перерахованих вище моделей ПРГ.

Ще однією складовою створення семантичних грід є використання сервіс-орієнтованого підходу до архітектури грід і використання в цій архітектурі семантичних сервісів. У наступних двох розділах наводяться звичайна, а також семантична сервіс-орієнтована архітектура грід.

Сервіс-орієнтована архітектура грід була ретельно опрацьована і запропонована в звіті OGSA - Open Grid Services Architecture, який був підготовлений в рамках ініціативи OGF - Open Grid Forum. Суть цієї пропозиції описується далі.

Згідно OGSA грід має трирівневу архітектуру (див. рис. 1.11).

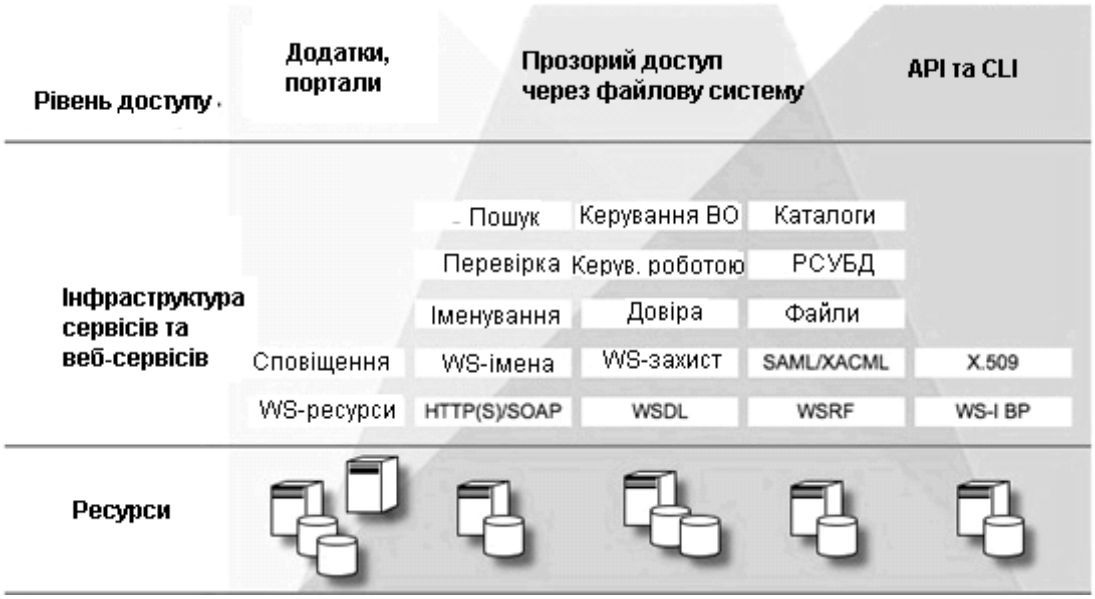


Рисунок 1.11 - Трирівнева сервіс-орієнтована архітектура грід

Нижній рівень - рівень ресурсів, які надає фізичні або віртуальні ресурси, які використовуються сервісами для реалізації своїх функцій. Ресурси можуть бути самими різними - обчислювальні (комп'ютерні системи), програмні або інформаційні (файли, бази даних, сховища даних), або навіть цілі віртуальні організації.

Середній рівень - рівень сервісів, який надає набір стандартизованих послуг по роботі з ресурсами грід. Це ядро OGSA.

Верхній рівень - рівень доступу, надає функціональні можливості по взаємодії користувачів з грід або через спеціалізовані програми, або через сайти, або веб -портали, або з використанням набору API, або за допомогою інтеграції з локальної операційною системою.

У OGSA робиться особливий наголос на сервіси. Всі ресурси грід представляються у вигляді сервісів, тільки через сервіси можна отримати доступ до ресурсів. Сервіс-орієнтований підхід у грідах дозволяє вирішувати проблему надання стандартних механізмів визначення інтерфейсів, вирішувати проблему локальної / віддаленої прозорості ресурсів, адаптувати систему грід до сервісів локальної ОС, уніфікувати семантику сервісів. Сервісна архітектура також спрощує віртуалізацію, яка необхідна, наприклад, для створення віртуальних організацій.

Специфіка сервісів в гріді. В рамках цієї ініціативи грід-сервіс визначається як веб-сервіс, який надає набір чітко визначених інтерфейсів і який задовольняє спеціальним вимогам, а саме:

наявність стандартних інтерфейсів;

динамічне створення і знищення сервісів;

можливість автоматичного пошуку сервісів;

управління життєвим циклом сервісів;

наявність механізму повідомлень;

наявність механізмі захисту;

підтримку інтероперабельності у динамічному гетерогенному середовищі.

У OGSA в рамках вимог динамічного створення та знищення сервісів сформульовано ще одна важлива вимога до сервісів - створення тимчасового екземпляра сервісу. Будь-яка діяльність може бути представлена у вигляді безлічі можливих станів і переходів. Кожне з станів може ініціювати динамічне створення екземпляра сервісу, який виконує цільову функцію цього стану. Коли система перейшла в новий стан, створений екземпляр сервісу вже не потрібен і він знищується.

Сервіси у великих складних розподілених обчислювальні середовищах повинні мати здатність незалежного оновлення. У зв'язку з цим в рамках механізмів управління життєвим циклом повинні надаватися можливості з ведення версій сервісів і підтримці сумісності не тільки серед версій одного сервісу, але і версіями різних сервісів. Більше того, ця система ведення версій повинна функціонувати без втручання в обчислювальний процес і роботу клієнтів гріда. Нарешті, слід підтримувати не лише версійність сервісів, а й оновлення відомостей про сервіси у відповідних реєстрах.

Архітектура OGSA представляє множину сервісів гріда, їх інтерфейси, семантику/поведінку і взаємодію між ними. Згідно OGSA грід містить такі групи сервісів:

сервіси інфраструктури;

сервіси управління обчисленнями;

сервіси даних;

сервіси управління ресурсами;

сервіси безпеки;

сервіси самокерування;

інформаційні сервіси.

Далі наводиться стислий опис цих сервісів.

Сервіси інфраструктури. Грід-сервіси створюються і функціонують не на порожньому місці. Вся грід-сервісна архітектура базується на веб-сервісній архітектурі. Тут повною мірою використовуються моделі, методи, мови, протоколи, сховища і, нарешті, архітектурні рішення веб-сервісів.

Завдання сервісів інфраструктури полягає в тому, щоб стати посередниками між веб-сервісами та іншими сервісами грід. Всі інші сервіси грід функціонують з використанням сервісів інфраструктури.

Сервіси управління обчисленнями. Сервіси цієї групи вирішують проблему забезпечення роботи будь-якої задачі (мається на увазі прикладної задачі), включаючи, але не обмежуючи:

Визначення місць можливого виконання завдання з урахуванням різних вимог і обмежень на її виконання (процесори, пам'ять, мережу, дані, наявність ліцензій, тощо.

Вибір конкретного місця виконання завдання. Використовуючи результати попереднього пункту, вибирається конкретне місце з урахуванням різних оптимізаційних алгоритмів.

Підготовка до виконання. Проведення всіх необхідних підготовчих заходів для запуску завдання (розгортання і конфігурація програм і бібліотек, резервування даних, резервування обчислювальних ресурсів і пам'яті, підготовка навколишнього середовища тощо).

Запуск на виконання. Коли все готове, завдання запускається на виконання з відпрацюванням можливо додаткових дій, пов'язаних з цим запуском, наприклад, реєстрація цього факту у відповідному журналі.

Управління виконанням. Управління обчисленням включає: відстеження процесу обчислень, відпрацювання збійних ситуацій, можливий перезапуск завдання в іншому місці, простановку контрольних точок для можливого відкоту, резервне копіювання і відновлення, збір статистики, і багато інших питань.

Завершення виконання завдання. Звільнення ресурсів, реєстрація завершення, можливі повідомлення, тощо.

Сервіси даних. Ці сервіси мають відношення до розміщення, зберігання, маніпулювання, доступу до даних та їх вибірці, а також до переміщення даних між різними ресурсами. Ця група може містити такі сервіси:

розміщення і зберігання даних;

маніпулювання даними;

організація доступу до даних;

пошук даних ;

резервне копіювання і відновлення даних;

тиражування даних ;

перетворення даних;

віртуалізація даних;

робота з метаданими ;

історія походження і використання даних.

При цьому сервіси цієї групи в цілому повинні забезпечувати масштабованість, ефективність, доступність даних.

Сервіси управління ресурсами. У OGSA пропонується три види управління, які передбачають використання ресурсів:

Управління самими фізичними і логічними ресурсами / сервісами;

Управління ресурсами на функціональному рівні;

Управління ресурсами на рівні інфраструктури (також званої грід-фабрикою).

На рис.1.12 наведено 3 рівня грід-архітектури, щодо яких розглядаються сервіси управління.

На рівні ресурсів управління ресурсами здійснюється з використанням їх власних інтерфейсів. Управління на цьому рівні включає моніторинг (наприклад, відстеження стану ресурсу), настройку і контроль (наприклад, установка необхідного стану ресурсу) і пошук. Ці ресурси управляються з використання їх описів (онтологій), які містять, наприклад, їх властивості, призначення, вхідні і вихідні дані, їх взаємозв'язок та ін.

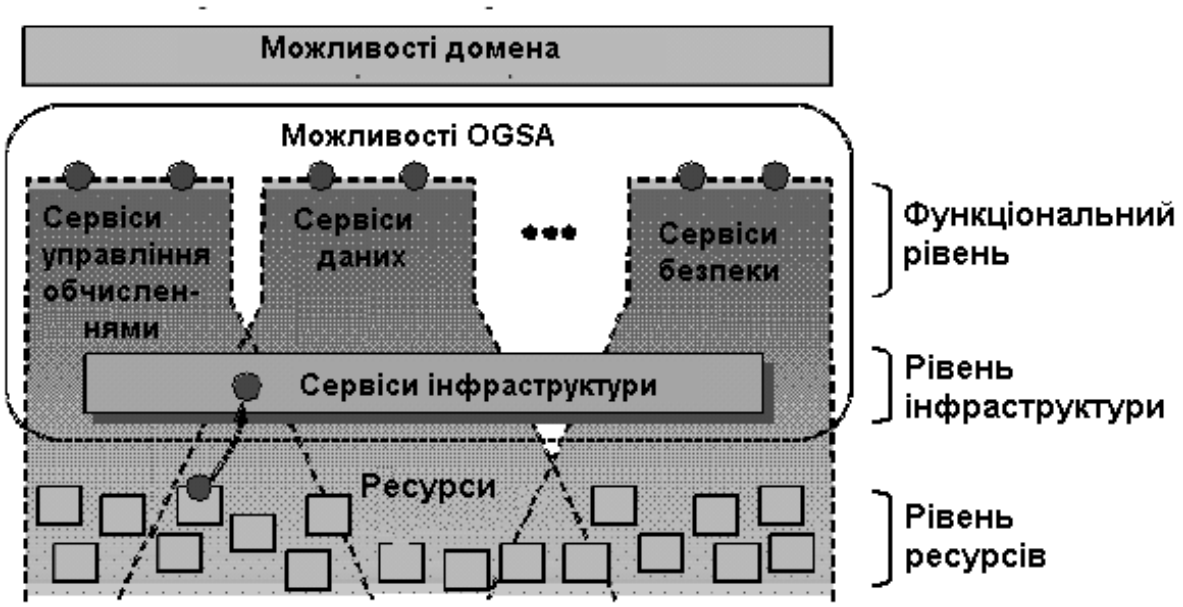


Рисунок 1.12 - Рівні управління в грід

На рівні інфраструктури визначається базовий набір операцій управління ресурсами, необхідних для вирішення завдань управління в середовищі грід.

На функціональному рівні надається два види інтерфейсів управління, що позначаються на рис 1.12 двома кружечками у верхній частині кожного з сервісів: функціональний інтерфейс управління і спеціальний інтерфейс управління кожним з ресурсів/сервісів.

Сервіси безпеки сприяють підтримці політики безпеки у віртуальній організації. Політика безпеки може включати: збереження і конфіденційність переданих повідомлень на всіх рівнях, аутентифікацію взаємодіючих індивідів, безпечної авторизації та аудиту, поділ повноважень, виявлення та недопущення незаконних втручань, перевірка виконання політики авторизації, перевірка

повноважень на виконання операцій, контроль доступу, міра довірливості до навколишнього середовищі, запобігання атак, тощо.

У зв'язку з цим пропонуються наступні можливості щодо забезпечення безпеки і відповідні їм сервіси:

Аутентифікація (перевірка справжності) - перевірка, чи є хтось або щось тим, за кого себе видає.

Відображення ідентифікацій. У різних системах існують механізми ідентифікації необхідних об'єктів, наприклад, ідентифікації користувачів через їхні паролі. Якщо система велика, то може виявитися, що в різних підсистемах один і той же об'єкт має різну ідентифікацію. Завдання цього сервісу - можливість встановлення відповідності між множиною таких ідентифікуючих систем.

Авторизація. Встановлення і подальша перевірка того, що той чи інший індивід має право виконувати ті чи інші дії по відношенню до того чи іншому у ресурсу.

Перетворення облікових даних. Облікові дані необхідні багатьом сервісам і, можливо, у специфічній для сервісів формі. Цей сервіс здійснює необхідне перетворення облікових даних.

Аудит і ведення журналу безпеки. Сервіс аудиту відповідальний за створення і збереження записів, що мають відношення до всіх подій, що відбуваються в системі, що мають відношення до політики безпеки. Журнал, при цьому ведеться, може бути проаналізований на предмет виявлення відхилень або спроб відхилень від політики безпеки.

Конфіденційність. Сервіс конфіденційності насамперед стосується політики керованої класифікації персональної інформації. За допомогою цього сервісу провайдери сервісів і споживачі сервісів можуть зберігати конфіденційну інформацію. Цей сервіс може використовуватися для вироблення і дотримання політики конфіденційності у віртуальній організації.

Сервіси самокерування. Самокерування передбачає, що всі системні компоненти, включаючи комп'ютери, мережі, програми, мають властивість

самоконфігурування, самовідновлення і самооптимізації. Одне з основних завдань самокерування - досягти максимального ступеня автономності. Сервіси самокерування відповідальні за встановлення та налаштування відповідних політик функціонування системи та подальшої зміни поведінки підпорядкованих їм сервісів згідно з аналізом стану системи. У свою чергу поведінка самих сервісів самокерування управляється тими правилами, які були закладені при їх реалізації, або які вони отримують в результаті взаємодії з іншими сервісами самокерування.

Виділяється три різновиди сервісів цього типу

Сервіси самоконфігурування - динамічно адаптують підпорядковані їм сервіси згідно зі змінами в системі.

Сервіси самовідновлення - виявляють некоректну поведінку підлеглих їм сервісів та ініціюють коригувальні дії.

Сервіси самооптимізації - виявляють неефективне функціонування підпорядкованих їм сервісів і вживають заходів щодо оптимізації їх роботи.

Ці три види сервісів самокерування є залежними один від одного і функціонують спільно для досягнення поставлених ним цілей. Вони спільно відстежують ситуацію, що складається в системі,, проводять аналіз і вживають необхідних заходів.

Інформаційні сервіси грід призначені для ефективного маніпулювання інформацією про додатки, ресурси та сервіси грід. Джерелами такої інформації можуть бути їхні постачальники, або вона може породжуватися в процесі функціонування системи. Існують наступні види сервісів, які призначені для виконання відповідних функцій з маніпулювання та надання інформації:

Пошук інформації. Надає можливість відшукувати необхідну інформацію про інформаційних ресурсах системи та функціонування системи.

Обмін повідомленнями. Здійснює обмін інформаційними повідомленнями між постачальниками і споживачами інформації.

Ведення журналів. Фіксується інформація про необхідні події, що відбуваються в системі.

Моніторинг. Відстеження стану інформаційних ресурсів.

Інформаційні сервіси повинні мати властивості безпеки, необхідної якості обслуговування, доступності, масштабованості, необхідної продуктивності.

Як вже було зазначено вище, суть семантизації сервісів в Інтернеті полягає в тому, щоб побудувати онтологію сервісів. У рамках семантичного вебу такі онтології існують і один з підходів, який підтримується в мові OWL-S, описаний в розділі «1.1.5. Сервіси семантичного веба». Далі у розділі описується архітектура, яка розширює OGSA для забезпечення явного оперування семантикою, і в якій визначаються сервіси знань з тим, щоб підтримувати спектр можливостей семантичних базових сервісів. У цій моделі, що отримала назву семантичної OGSA (Семантичний - OGSA - S-OGSA), визначаються модель, можливості та механізми семантичного гріда, які описані далі.

У спрощеному варіанті можна уявити, що S-OGSA - це OGSA з включеною єдиною додаткової групою сервісів Сервіси забезпечення семантики (рис. 1.13), які взаємодіють з сервісами всіх інших груп.

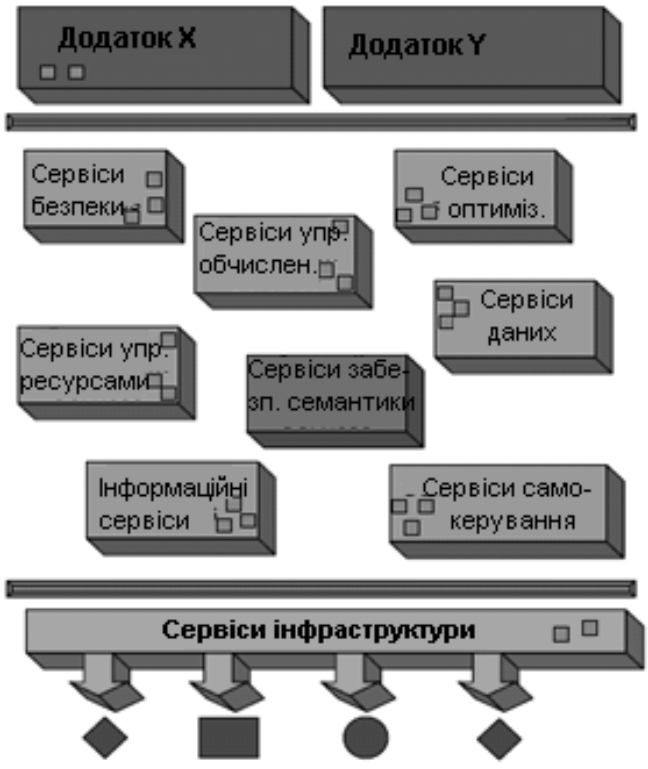


Рисунок 1.13 - Спрощена архітектура OGSA с єдиною групою сервісів S-OGSA

S-OGSA будується з дотриманням наступних шести принципів:

Простота архітектури. Архітектурні рішення повинні бути мінімально простими з тим, щоб мінімізувати вплив на все те, що вже є в грідах.

Розширюваність. Архітектурні рішення повинні припускати розширюваність і налаштованість.

Сумісність : Все, що пропонується в S-OGSA, є сумісним з OGSA. Сервіси S-OGSA є сервісами OGSA, знання і метадані є грід -ресурсами.

Різноманітність семантичних можливостей. У будь-який момент часу з кожним сервісом грід може бути пов'язано безліч семантичних можливостей. Сутності грід не є сутностями семантичного грід. Однак, семантичні характеристики можуть бути можливими для грід-ресурсів в будь-який момент часу.

Неоднорідність семантичного представлення. Будь-яке властивість ресурсу може мати безліч різних видів семантичного опису і кожне з таких описів може бути представлено у різній формі (текст, логічна формула, онтологія, правило).

Розвиток сервісів. Сервіси повинні мати природний шлях їх розвитку в бік інтелектуалізації з метою надання можливостей з маніпулювання.

Визначення семантичних ресурсів, які поставляються і споживаються сервісами, розширює загальноприйняту модель OGSA. У S-OGSA поняття семантики вводиться таким чином, що група сервісів, що забезпечують семантику, по суті, здійснює встановлення взаємозв'язку між об'єктами традиційного грід, і об'єктами того середовища, яка забезпечує подання знань (у нашому випадку це семантичний веб).

На рис. 1.14 дається графічне представлення цієї концепції, а далі описуються основні складові відповідної моделі.

Сутності грід. Будь-які сутності, які мають властивості ідентифікації в гріді, включаючи ресурси і сервіси. (щодо термінології в грідах див. [**Error! Reference source not found.**]).

Сутності знань. Спеціальний вид сутностей грід, які або представляють, або можуть оперувати з деякими знаннями. Прикладами сутностей знань є онтології, правила, бази знань, або навіть повнотекстові документи, всі з яких містять в собі знання. Сервісами знань є такі, які надають доступ або оперують такими ресурсами знань. Прикладами можуть бути механізми підтримки правил виводу, механізми підтримки автоматичних міркувань, тощо.

Семантичне зв'язування. Це сутності, які призначені для представлення асоціацій сутностей грід з сутностями знань. Існування такої асоціації дозволяє представити вихідну сутність грід у вигляді сутності семантичного грід. Проводячи аналогію з семантичним вебом, семантичне зв'язування дозволяє прив'язувати веб-ресурси до відповідних їм метаданих. Зазначимо, що сутності зв'язування також є грід-сутностями. І це диктується принципом сумісності S-OGSA і OGSA. Як випливає з позначень зв'язків на рис 1.16 кожна сутність семантичного зв'язування здійснює зв'язування однієї або більше сутності знань з однією або більше сутностей грід. У свою чергу сутності знань і сутності грід можуть існувати без будь-якого зв'язування один з одним (тобто вони можуть бути не пов'язаними з сутностями семантичного зв'язування).

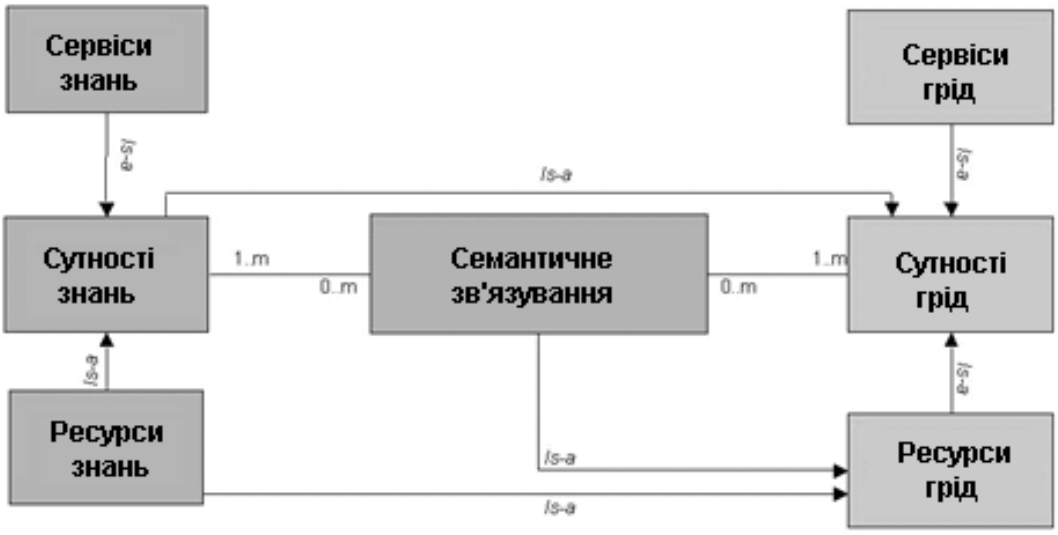


Рисунок 1.14 - Модель S-OGSA

Сутності семантичного грід. Це такі сутності, які або є суб'єктом семантичного зв'язування, або самі є сутностями семантичного зв'язування, або

є сутностями знань. По суті все, що наведено на рис. 1.16, є сутностями семантичного грід. Дотримуючись принципів різноманітності семантичного представлення, неоднорідності подання та розвитку семантики, сутності грід можуть бути одночасно пов'язані з нулем або більше сутностями знань різної форми і з різними можливостями, а також можуть набувати і позбавляться цих зв'язків динамічно в будь-який момент часу свого життєвого циклу.

Сервіси забезпечення семантики (рис.1.15.) здійснюють підтримку механізмів маніпулювання семантикою, надаючи можливості по створенню, зберіганню, оновленню, видалення та пошуку і доступу до знань різного виду. Семантика, що надається сервісами цієї категорії, може використовуватися як всередині самого грід, так і поза його (у зовнішніх додатках).

Сервіси забезпечення семантики, у свою чергу, розбиваються на такі дві категорії:

сервіси забезпечення знаннями та

сервіси забезпечення семантичного зв'язування.

Сервіси забезпечення знаннями включають:

Сервіси онтологій. Вони відповідають за створення, збереження і доступ до онтологій, які є концептуальними інформаційними моделями предметних областей.

Сервіси міркувань. Вони відповідають за підтримку механізмів виведення на онтологіях з метою або виведення нових знань з існуючих онтологій, або підтримки обмеження цілісності, заданих на онтологіях.

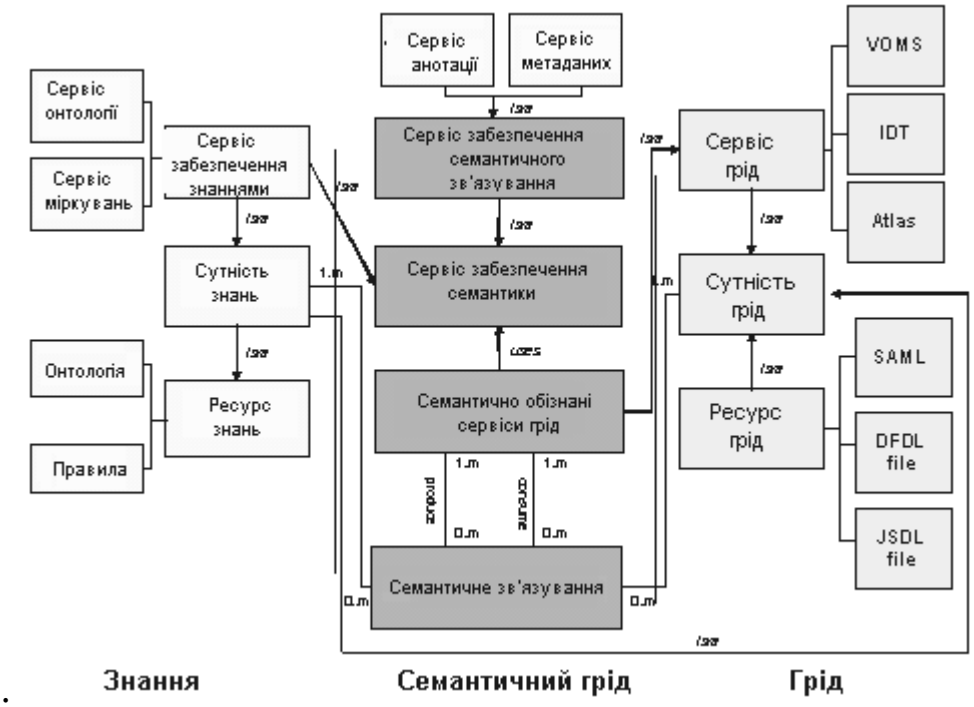


Рисунок 1.15 - Сервіси забезпечення семантики

Сервіси забезпечення семантичного зв'язування включають:

Сервіси метаданих. Вони відповідають за зберігання і доступ до сутностей семантичного зв'язування, і зазвичай вони представляються у вигляді екземплярів відповідної онтології. Існує тісний взаємозв'язок між сервісами метаданих і сервісами онтологій, так як дані, що запам'ятовуються сервісом метаданих, зазвичай базуються на концептуальних моделях, що представляються і запам'ятовуються сервісами онтологій. Сервіси метаданих також можуть використовувати сервіси онтологій для здійснення виводу на метаданих.

Сервіси анотування. Їх призначення - проводити інтелектуальний аналіз і в результаті створювати опис (метадані) інформаційних ресурсів різного виду (документи, бази даних, сховища даних, архіви, електронні бібліотеки, тощо). Концептуалізація предметних областей, побудова їх концептуальних моделей і представлення їх у вигляді онтологій є одноразовим завданням. У свою чергу процес аналізу даних і створення метаданих є постійним процесом, який необхідно постійно підтримувати при функціонуванні грід.

Семантично обізнані сервіси. Деякі з сервісів грід можуть використовувати технологію роботи зі знаннями для виконання своєї функціональності. Такі «просунуті» сервіси грід називаються Семантично обізнаними сервісами. У запропонованій архітектурі семантична обізнаність сервісу означає здатність ініціювати семантичне зв'язування і використовувати результати цього зв'язування (тобто отримані знання) у своїй роботі.

1.3 Постановка задачі дослідження

Для реалізації запропонованої системи забезпечення безпеки семантичних БД необхідно вирішити набір завдань, таких, як узгодження рівнів безпеки елементів онтології, визначення рівнів безпеки триплетів і результатів логічних висновків, виявлення порушень результатів логічних висновків і контроль отриманих результатів при виконанні запитів. В наступному розділі описується постановка цих задач.

Висновки до розділу 1

Семантичний веб являє собою розширення існуючого вебу, суть якого полягає в тому, що даним надається чітко визначений сенс, що дозволяє чітко однозначно і повно сприймати такі дані людьми і комп'ютерами, що також сприяє вдосконаленню взаємодії людей з комп'ютерами. Таке розширення реалізується за рахунок розмітки вмісту вебу, його властивостей і відношень з використанням мов розмітки з чітко визначеною семантикою. Такі мови, звані мовами семантичного вебу, призначені для ідентифікації та подання ресурсів вебу та їх взаємозв'язків, семантичних правил, яким вони повинні задовольняти і, нарешті, здійснювати їх пошук.

Сервіси семантичного вебу, які також називаються семантичними веб-сервісами, це парадигма, яка інтегрує семантичні метадані, онтології, формальні засоби опису, а також інфраструктуру веб-сервісів. Семантичні веб-сервіси описуються як сервіси вебу але з використанням мови, що має чітко визначену формальну семантику. Використання семантичного вебу для опису сервісу надає можливість визначити семантику його інтерфейсу, яка, у зв'язку з наявністю її формальної специфікації, по-перше, може сприйматися комп'ютером і, по-друге, може використовуватися для взаємодії між сервісами.

2 ПРОЕКТУВАННЯ ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ ДО РЕСУРСІВ СЕМАНТИЧНОГО ВЕБ

2.1 Вибір і обґрунтування оптимальності децентралізованої системи контролю доступу до ресурсів семантичного веб

Зареєстровані користувачі U, що мають рівень доступу sl_U і права доступу U_p, можуть відправляти SPARQL-запити q до семантичної БД DB_S для перегляду, отримання, додавання або зміни їх даних D і для виконання логічнихправил r для отримання результатів логічних висновків r_L.

Семантична БД вважається безпечної, якщо задовольняє наступним умовами:

користувач U може мати право доступу на перегляд даних D, якщо sl_U sl ;

користувач U має права на зміну, видалення і додавання даних D, якщо $sl_U \geq sl_D$ і U_p = {Перегляд, зміна, видалення, додавання};

користувач U може виконувати логічні правила r, якщо $sl_U \geq sl_r$;

користувач може отримати результати логічних висновків r_L, якщо $sl_U \geq sl_{rL}$.

На малюнку 2.1 показаний загальний процес забезпечення безпеки СБД.

Процес забезпечення безпеки СБД виконується наступним чином:

Якщо запит користувачів до СБД є прямим запитом, то модуль «управління доступом» виконує наступні дії:

перевіряє рівні доступу користувачів;

визначає рівні безпеки отриманих відповідей на запит;

дає результати користувачам відповідно до їх рівнями доступу.

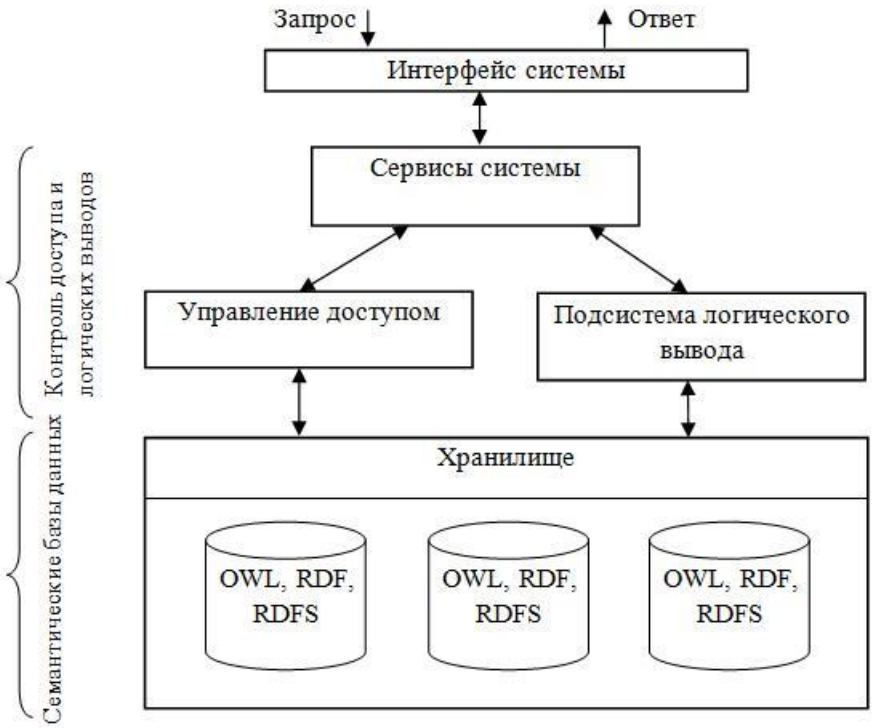


Рисунок 2.1 - Процес забезпечення безпеки роботи з семантичними БД

Якщо запит є логічним запитом, то модуль «підсистема виконання логічних висновків» виконує наступні дії:

перевіряє рівні доступу користувачів;

визначає можливість виконання логічних правил;

виконує логічні висновки;

виявить порушення результатів логічних висновків;

контролює результати логічних висновків;

дає результати користувачам відповідно до їх рівнями доступу.

2.2 Постановка задачі моделювання, обґрунтування припущень і розробку базової моделі, аналіз адекватності розроблених моделей

Для забезпечення безпеки СБД потрібно побудувати систему забезпечення безпеки роботи з семантичними БД (що позначається як SS), під якою розуміється система, що володіє двома можливостями: контролем доступу

користувачів до окремих елементам СБД і контролем результатів логічних висновків.

Пропонована архітектура системи SS розділена на 6 рівнів [15], відповідних різним етапам обробки запитів користувачів (малюнок 2.3):

Рисунок 2.3 - Пропонована структура системи забезпечення безпеки роботи з семантичними БД

Інтерфейс системи забезпечує взаємодію між користувачами і системою, за допомогою нього користувачі можуть надіслати запити до системи і отримати відповіді на них.

Відповіді можуть бути оформлені з використанням різних форматів даних, наприклад таких, як RDF / XML, Turtle або N3.

Рівень забезпечення безпеки - основна частина системи підтримки безпеки семантичних БД. На малюнку 2.3 показані типові функції даної частини, які

розробляються і досліджуються в даній роботі. Вони реалізовані у вигляді наступного набору модулів:

модуль узгодження рівнів безпеки елементів онтологій і індивідів метаданих;

модуль визначення рівнів безпеки триплетів - здійснює процес розрахунку рівнів безпеки всіх можливих триплетів в СБД;

модуль визначення рівнів безпеки результатів логічних виводів - визначає всі рівні безпеки триплетів, отриманих на основі відомих даних за допомогою використання логічних правил;

модуль уявлення семантичної БД у вигляді RDF- графів - визначає всі RDF- графи в СБД;

модуль виявлення порушень результатів логічних висновків - виділяє всі несанкціоновані отримані триплети при виконанні правил.

Сховище баз даних використовується для зберігання RDF- метаданих, онтологій предметних областей і логічних правил.

На основі запропонованої архітектури системи забезпечення безпеки роботи з семантичними БД підтримується виконання процесів, показаних на малюнку 2.4.

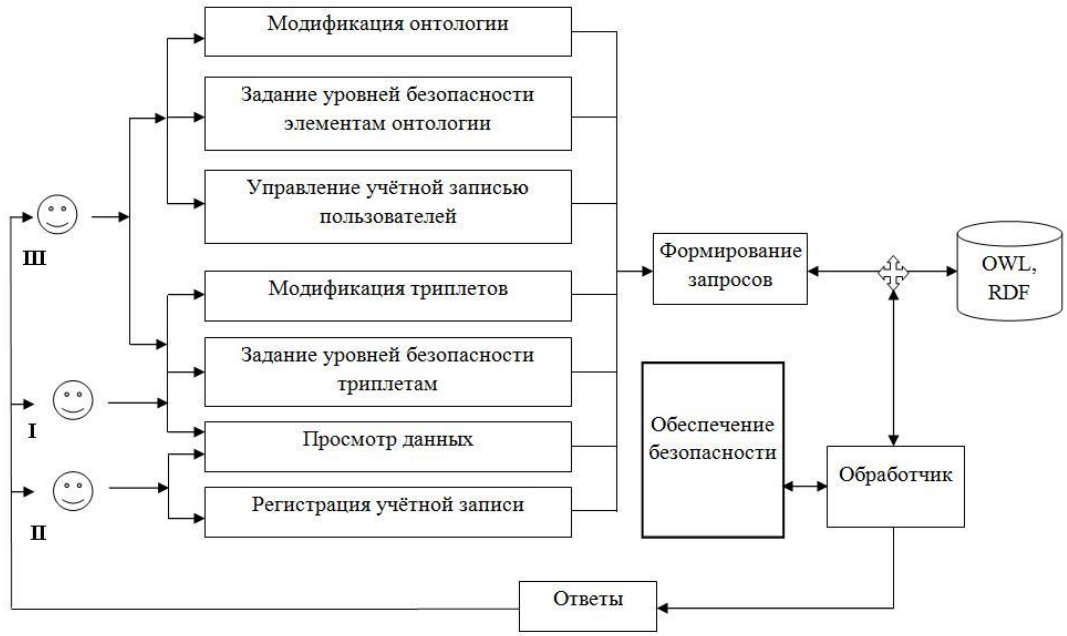


Рисунок 2.4 - Основні процеси роботи користувачів системи всі процеси роботи користувачів розділені на три групи:

Група I - користувачі: користувачі мають свої облікові записи та можуть виконувати функціональності, такі, як модифікація триплетів, завдання рівнів безпеки триплетів і перегляд даних. Залежно від обраної функціональності система формує відповідний запит, який від-спрямовується оброблювачу. оброблювач обробляє дані в СБД і викликає компонент «забезпечення безпеки »для підтримки безпеки даних при виконанні відповідної функціональності. результати виконання роботи видаються користувачам.

Група II - гості: користувачі низького статусу можуть виконувати функціонального перегляду даних і реєстрацію до системи.

Група III - адміністратори: користувачі високого статусу мають можливість виконувати всі функціональності системи, таких як перегляд, модифікація, завдання рівнів безпеки даними (триплети і елементам онтологогії, логічним правилам), управління облікового записом користувачів.

Пропонована система забезпечує безпеку семантичної БД в порядок виконання наступних дій [6] (рисунок 2.5):

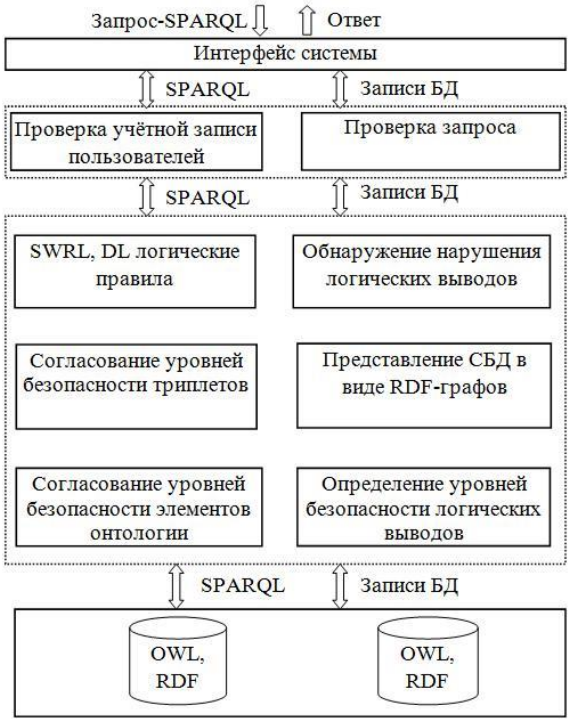


Рисунок 2.5 - Процес забезпечення безпеки СБД

При початковому запуску система виконує наступні дії: визначає рівні безпеки SL DB всіх елементів онтології O і метаданих M с допомогою модуля «узгодження рівнів безпеки елементів онтології»; визначає рівні безпеки SC DB всіх триплетів в СБД за допомогою модуля «узгодження рівнів безпеки триплетів»; визначає рівні безпеки SC L можливих результатів логічних висновків СБД за допомогою модуля «визначення рівнів безпеки результатів логічних висновків». Покриття безпеки триплетів SC DB і покриття безпеки можливих результатів логічних висновків SC DB зберігаються в СБД.

При кожному вході користувача U в систему за допомогою модуля «перевірка облікового запису користувачів »виконується перевірка наявності його облікового записі, рівня доступу sl U і прав доступу U p, інформація про яких зберігається в СБД. якщо перевірка закінчується успішно, то користувач може відправляти запити q до системи для виконання різних операцій над даними відповідно до його рівнями і правами доступу.

При відправці користувачем запиту q система виконує його перевірку за допомогою модуля «перевірка запиту».

Якщо запит q складений граматично неправильно, то q не виконується, користувач повинен сформувати інші запити до системи.

Якщо q є логічним запитом (q збігається з логічним правилом r) і якщо sl U ≥ sl r (рівень доступу користувачів не менше рівня безпеки логічного правила), то запит q може виконуватися для отримання результатів логічних висновків, інакше, якщо sl U <sl r, то запит q не виконується.

Якщо q є прямим запитом, то він може виконуватися. Система дає користувачу результати відповідно до політики безпеки:

При виконанні прямого запиту q система дає користувачу U відповіді A, яких рівні безпеки sl A не більш рівня доступу користувачів sl U (sl A ≤ sl U).

При виконанні логічного запиту система виконує виявлення порушень результатів логічних висновків за допомогою модуля «виявлення порушень результатів логічних висновків ». В результаті цього система дає користувачу дозволені відповіді.

2.2 Розробка алгоритму і методики проведення моделювання

У семантичних БД для забезпечення безпеки доступу до онтології, класам і властивостями задаються початкові рівні безпеки sl і ∈ MS. Множина рівнів безпеки всіх елементів онтології SL O може бути визначено як SL O = {SL C, SL P}, де SL C = {sl C 1, ..., sl Cm} - безліч всіх рівнів безпеки класів в онтологіях СБД, а SL P = {sl P 1, ..., sl Pn} - безліч рівнів безпеки властивостей онтологій СБД [110].

Початкові рівні безпеки елементів можуть бути неузгодженими, як показано на малюнку 2.6. Рівні безпеки елементів в показаної частини є неузгодженими в зв'язку з наступними проблемами:

Користувачі, що мають рівні доступу sl U = 1, можуть мати доступ до підкласу «Commentation_Reward» класу «Company», так як підклас «Commentation_Reward» має рівень безпеки sl = 1, sl U = sl. Але ці користувачі не можуть мати доступ до класу «Company», так як у нього рівень безпеки sl = 2> sl U = 1, отже, користувачам не можна мати доступ добудь-яким його подкласам або індивідам. З урахуванням цього, захищеність данної онтології порушена.

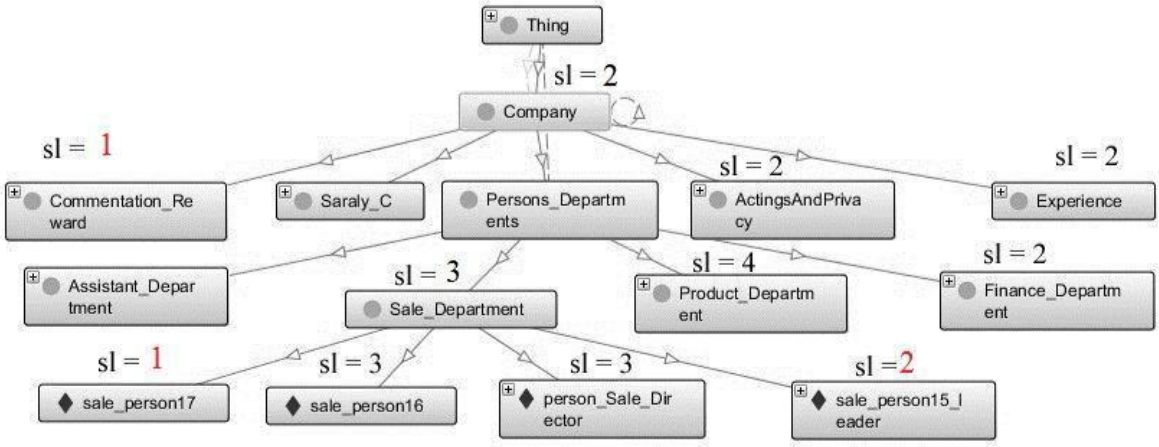


Рисунок 2.6 - Частина онтології з неузгодженими рівнями безпеки

Користувачі, що мають рівні доступу sl U = 2, можуть мати доступ до «sale_person17» і «sale_person15_leader» класу «sale_Department». Але з метою

безпеки СБД будь-які користувачі, які мають рівні доступу sl U <3, не можуть мати доступ до будь-яких індивідам в класі «sale_Department», так як його рівень безпеки sl = 3> sl U. З урахуванням цього, захищеність даної онтології порушена.

Для контролю доступу користувачів до елементам онтології необхідно узгодити рівні безпеки її елементів (класів, підкласів та індивідів). Узгодження рівнів безпеки елементів в онтології слід виконати на основі наступних умов (принцип 1).

Принцип 1. Узгодження рівнів безпеки елементів в СБД [11, 12].

Елемент А семантичних БД повинен мати рівень безпеки sl A.

Якщо елемент А включає в себе елемент В і належить елементу С, то між їх рівнями безпеки повинно виконуватися ставлення sl B ≥ sl A ≥ sl C, де sl A, sl B, і sl C - рівні безпеки елементів A, B і C відповідно.

Якщо між елементами А і В існують зв'язки, за допомогою яких користувач можуть дізнатися А через В (наприклад, властивості «isSuperiorOf» - бути руководієм і «hasSuperior» - має керівника в онтології), то ці елементи повинні мати однакові рівні безпеки, sl B = sl A.

На малюнку 2.7 показаний приклад узгодженості рівнів безпеки елементів частини онтології.

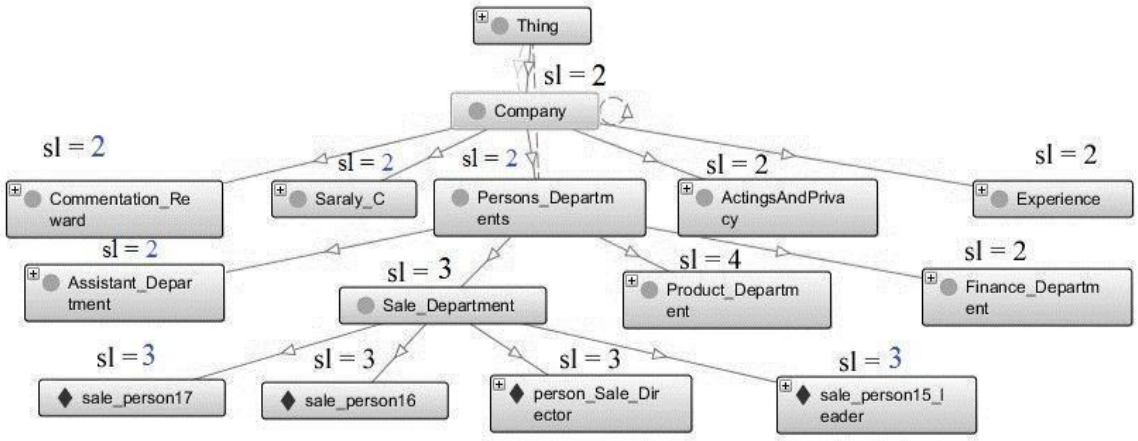


Рисунок 2.7 - Частина онтології з узгодженими рівнями безпеки

Визначення рівнів безпеки елементів онтологій

На основі принципу 1, з урахуванням зв'язків між класами (rdfs: subClassOf, owl: sameAs, owl: equivalentClass) і відносин між властивостями

(Rdfs: subPropertyOf, owl: inverseOf, owl: equivalentProperty, owl: InverseFunctional-Property) онтології пропонуються наступні принципи узгодження рівнів безпеки елементів в СБД [113].

Принцип 2. Узгодження рівнів безпеки класів онтології.

У онтологіях O немає класів з x, що не мають рівнів безпеки sl Cx.

Рівень безпеки sl Cx класу з x повинен бути більше або дорівнює рівню безпеки його суперкласу sl Csup, sl Cx ≥ sl Csup.

Якщо клас з x зв'язується з класом з y, які мають рівень безпеки sl y, допомогою відносин owl: sameAs, owl: equivalentClass, то sl Cx = MAX (sl Cx, sl Cy);

Принцип 3. Узгодження рівнів безпеки властивостей онтології.

У онтологіях O немає властивостей p x, що не мають рівнів безпеки sl Px.

Рівень безпеки sl Px властивості p x повинен бути більше або дорівнює рівню безпеки його супервластивості sl Psup, sl Px ≥ sl Psup.

Якщо p x зв'язується з іншими властивостями p y по ставленням rdfs: subPropertyOf, owl: inverseOf, owl: equivalentProperty, owl: Inverse-FunctionalProperty, то sl Px = MAX (sl Px, sl Py), де sl Py - рівень безпеки свійства p y.

Принцип 4. Узгодження рівнів безпеки індивідів.

У семантичних БД кожному індивіду i x заданий початковий рівень безпеки sl Ix.

Рівень безпеки sl Ix індивіда i x повинен бути більше або дорівнює рівню безпеки sl Cy класу, якому він належить, sl Ix ≥ sl Cy.

Визначення рівнів безпеки класів онтологій

В онтології O є безліч класів C = {з 1, ..., з m} і безліч заданих їм рівнів безпеки SL C = {sl C 1, ..., sl Cm}, де sl C 1, .. ., sl Cm - рівні безпеки класів c 1, ..., c m відповідно.

На основі принципу 2 рівень безпеки sl Cx класу з x ∈ З може бути визначений наступним чином:

якщо рівень безпеки sl Cx класу з x не заданий, то sl Cx = 0;

якщо з x зв'язується з з у ставленням owl: sameAs, owl: equivalentClass, то sl Cx MAX (sl Cx, sl Cy), де sl Cy - рівень безпеки класу с у;

якщо клас з x є підкласом класу з z, який має рівень безпеки sl Cz, то sl Cx = MAX (sl Cx, sl Cz).

Нижче показаний алгоритм визначення рівнів безпеки класів онтології на основі використання принципу 2.

Вхідними даними є: онтологія O, безліч класів C і відповідні їм рівні безпеки SL C. Вихідними даними алгоритму є рівень безпеки sl Cx класу з x.

Алгоритм 2.1

Крок 1: Початок алгоритму

= {З 1, ..., з m}, SL C = {sl C 1, ..., sl Cк}; с x ∈ C

Крок 2: Якщо LEVEL_CLASS (з x) = false, то CREATE_LEVEL (с x, sl Cx = 0); Інакше sl Cx = GETLEVEL (C x);

Крок 3: For i = 1 to n

Якщо GETRELATION_CLASS (с i, с x) = true

sl Ci = GETLEVEL (с i); sl Cx = MAX (sl Ci, sl Cx); i = i + 1; інакше i = i +1;

Крок 4: з z = GETSUBCLASS (з x);

sl Cz = GETLEVEL (с z);

sl Cx = MAX (sl Cx, sl Cz);

Крок 5: вивід sl Cx;

кінець алгоритму

Визначення рівнів безпеки властивостей онтологій

Спочатку в онтології O властивостями P = {p 1, ..., p n} задаються початкові рів-ні безпеки SL P = {sl P 1, ..., sl Pn}, де sl P 1, ..., sl Pn - відповідні рівні безпеки властивостей p 1, ..., p n.

На основі принципу 3 рівні безпеки sl Px властивості p x ∈ P узгоджуються між собою в такий спосіб:

якщо рівень безпеки sl Px властивості p x не заданий, то sl Px = 0;

якщо властивість p x пов'язано з властивістю p y одним з відносин owl: inverseOf, owl: equivalentProperty, owl: SymmetricProperty або owl: InverseFunctionalProperty, то sl Px = MAX (sl Px, sl Py), де sl Py - рівень безпеки властивості p y;

якщо властивість p x є підвластивістю властивості p z, який має рівень безпеки sl Pz, то sl Px = MAX (sl Px, sl Pz).

Нижче описаний алгоритм 2.2 визначення рівнів безпеки властивостей онтології на основі використання принципу 3.

Алгоритм 2.2

Крок 1: Початок алгоритму

O = <C, P, E, F, L, AC>;

= {P 1, ..., p n}, p x ∈ P, SL P = {sl P 1, ..., sl Pк};

Крок 2: Якщо LEVEL_PROPERTY (P x) = false, то CREATE_LEVEL (p x, sl Px =0);

Інакше sl Px = GETLEVEL (P x);

Крок 3: For i = 1 to m

Якщо GETRELATION_PROPERTY (p i, p x) = true sl Pi = GETLEVEL (p i);

sl Px = MAX (sl Pi, sl Px);

i = i +1;

інакше i = i +1;

Крок 4: P z = GETSUBPROPERTY (p x);

sl Pz = GETLEVEL (p z);

sl Px = MAX (sl Px, sl Pz);

Крок 5: вивід sl Px;

кінець алгоритму

Вхідними даними алгоритму є онтологія O, безліч властивостей P {P 1, ..., p n} і їх відповідні рівні безпеки SL P. вихідними даними алгоритму є рівень безпеки sl Px властивості p x.

Визначення рівнів безпеки індивідів

В семантичних БД DB S є безліч індивідів I DB = {i 1, ..., i k} і безліч їх початкових рівнів безпеки SL I = {sl I 1, ..., sl Ik}, де sl I 1, .. ., sl Iк -рівні безпеки індивіда i 1, ..., i k відповідно.

На основі принципу 4 рівень безпеки sl Ix індивіда i x ∈ i DB може бути визначений наступним чином:

Якщо початковий рівень безпеки sl Ix індивіда i x не заданий, то sl Ix = 0.

Якщо індивід i x включається в клас c y, що має рівень безпеки sl Cy,

якщо sl Ix <sl Cy, то sl Ix = sl Cy.

Нижче показаний алгоритм 2.3 визначення рівнів безпеки індивідів.

Алгоритм 2.3

Крок 1: Початок алгоритму

= {З 1, ..., з m}

= {I 1, ..., i h}, i x ∈ c x, SL I = {sl I 1, ..., sl Ih};

Крок 3: For j = 1 to h

Якщо LEVEL_INDIVIDUAL (i j) = false, то CREATE_LEVEL (i j, sl Ij = 0); Інакше sl Ij = GETLEVEL (I j);

c j = GET_CLASS_INDIVIDUAL (i j);

sl Cj = GETLEVEL (c j);

sl Ij = MAX (sl Ij, sl Cj);

j = j + 1;

Крок 3: вивід sl Ix;

кінець алгоритму

Вхідними даними є: C = {з 1, ..., з m} - безліч класів, SL C = {sl C 1, ..., sl Cm} - безліч рівнів безпеки класів, I = {i 1, .. ., i k} - безліч індивідів, SL I = {sl I 1, ..., sl Iк} - безліч рівнів безпеки індивідів. Вихідними даними є безліч індивідів і їх відповідних рівнів безпеки.

Алгоритм визначення покриття безпеки семантичних баз даних

За допомогою алгоритмів 2.1 - 2.3 можуть бути визначені значення рівнів безпеки елементів онтології: SL C = {sl З 1, ..., sl Cm} - безліч узгоджених рівнів

безпеки класів C = {з 1, ..., з m}; SL P = {sl P 1, ..., sl Pn} - безліч узгоджених рівнів безпеки властивостей P = {p 1, ..., p n}; SL I = {sl I 1, ..., sl Ih} - безліч узгоджених рівнів безпеки індивідів I = {i 1, ..., i h}. На ос-нове цих значень уже можуть бути визначені узгоджені рівні безпеки триплетів.

Визначення рівня безпеки триплета

У семантичних БД компоненти кожного триплета t = [s, p, o] (s - суб'єкт, - предикат, o - об'єкт) можуть мати різні рівні: sl s - рівень безпеки суб'єкта, sl p - рівень безпеки предиката, і sl o - рівень безпеки об'єкта. Тоді загальний рівень безпеки всього триплета sl t буде визначатися як максимальне значення рівнів безпеки його компонентів (sl s, sl p, sl o): sl t = MAX {sl s, sl p, sl o},

Визначення узгоджених рівнів безпеки триплетів

Нижче показаний алгоритм 2.4 для створення покриття безпеки СБД.

Алгоритм 2.4

Крок 1: Початок алгоритму

Онтологія O = {З 1, ..., з m}; SL C = {sl З 1, ..., sl Cm}; P = {p 1, ..., p h}; SL P = {Sl P 1, ..., sl Ph}; I = {i 1, ..., i q}; SL I = {Sl I 1, ..., sl Iq}; T M = {T i, ..., t n}; SL M {Sl t1, ..., sl tk};

Крок 2:

For j = 1 to n

якщо FUNCTION_MAPPTING (Pt j, t j) = true, то sl j = GET_LEVEL (pt j);

Якщо CHECK_LEVEL (t j) = true, то

sl 'j = GET_LEVEL (t j);

Якщо sl 'j > Sl j, то sc j = (T j, sl 'j); DETETE_LEVEL (t j, sl j);

ADD_ LEVEL (t j, sl j);

Інакше ADD_ LEVEL (T j, sl j);

Інакше ADD_ LEVEL (T j, sl j);

Інакше j = j +1;

End For;

Крок 3: Кінець алгоритму

Вхідними даними алгоритму є: SL C, SL P, SL I - безлічі згоди-ванних рівнів безпеки множин класів C, властивостей P і індивідів I відповідно; T M = {t i, ..., t n} - безліч всіх триплетів семантичних метаданих-них; SL M = {sl t1, ..., sl tk} - безліч початкових рівнів безпеки триплетів семантичних метаданих.

Вихідними даними алгоритму 2.4 є покриття безпеки S = {s 1, ..., s k}, де s j = (T i, sl j), sl j - узгоджений рівень безпеки триплета t j.

Алгоритм визначення покриття безпеки результатів логічних виводів

Нижче показаний алгоритм 2.5 визначення рівнів безпеки результатів логічних висновків.

Алгоритм 2.5

Шаг1: Початок алгоритму

= {R 1, ..., r n}, r i = b 1 ∧ ... ∧ b k

= {T 1, ..., t m}, SL M = {sl t 1, ..., sl tm}

Крок 2:

For i = 1 to n:

Якщо DIFINE_FUNTION (f: b 1, ..., b k → □ 1, ..., □ k, де f (b 1) = □ 1, ..., f (b k) = □ k) = true, то:

sl rLi = 0; □ = F (q);

For j = 1 to k

sl j = GET_LEVEL (b j);

sl rLi = MAX (sl rLi, sl i);

j = j + 1;

End for;

Якщо CHECK (□) = true, то sl 'Li = GET_LEVEL (□);

Якщо (sl i ' > sl rLi), то sl rLi = sl i ';

інакше;

інакше;

ADD (□, sl rLi);

i = i + 1;

End For;

Крок 3: кінець алгоритму

Алгоритм контролю виконання запитів до семантичним баз даних

Нижче показаний алгоритм 2.6 для контролю виконання прямих і логічних запитів.

Алгоритм 2.6

Крок 1: Початок алгоритму.

Крок 2: визначення коректності відправленого запиту відповідно до оформленням SPARQL-мови.

Крок 3: порівняння рівня доступу sl U користувача U з рівнем безпечності SPARQL-запиту sl q до моменту його виконання (табл. 3.1).

Якщо sl U ≥ sl q, то крок 4 виконується, інакше запит не може виконуватися.

Крок 4: визначення виду запиту.

Якщо q є логічним запитом (шаблон запиту q збігається з тілом логічного правила r):

якщо sl U ≥ sl r, то запит q може виконуватися для отримання результатів логічних висновків і виконується крок 5,

інакше sl U <sl r, тоді алгоритм закінчується.

Інакше q є запитом прямого доступу, то виконується крок 5.

Крок 5: запит виконується відповідно до політики безпеки систем.

Для виконання логічного запиту q:

визначається рівень безпеки елементів в СБД;

визначається рівень безпеки результатів логічних ви-дів;

виявляються порушення результатів логічних висновків;

дозволені відповіді надаються користувачеві.

Для виконання прямого запиту q:

визначається рівень безпеки елементів в СБД;

визначається рівень безпеки триплетів;

визначається, чи є q частиною логічного правила:

якщо q не є частиною логічного правила, то система видає користувачеві U відповіді A, у яких sl A ≤ sl U.

якщо q є частиною логічного правила, то:

визначається безліч розкритих триплетів T '1;

видається результат A = {T 1 / T '1}.

Крок 6: кінець алгоритму

Висновки до розділу 2

Для забезпечення безпеки СБД потрібно побудувати систему забезпечення безпеки роботи з семантичними БД (що позначається як SS), під якою розуміється система, що володіє двома можливостями: контролем доступу користувачів до окремих елементам СБД і контролем результатів геологічних висновків.

Пропонована архітектура системи SS розділена на 6 рівнів, відповідних різним етапам обробки запитів користувачів .

Інтерфейс системи забезпечує взаємодію між користувачами і системою, за допомогою нього користувачі можуть надіслати запити до системи і отримати відповіді на них.

Рівень забезпечення безпеки - основна частина системи підтримки безпеки семантичних БД.

3 РЕАЛІЗАЦІЯ ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ ДО РЕСУРСІВ СЕМАНТИЧНОГО ВЕБ

3.1 Розробка моделі децентралізованої системи контролю доступу до ресурсів семантичного веб

Раніше в розділі 2 для контролю доступу користувачів до триплетів і елементам семантичних БД був розроблений ряд алгоритмів: узгодження рівнів безпеки елементів онтологій і індивідів метаданих; визначення покриття безпеки семантичних БД; управління відповідями при виконанні запитів до СБД. На їх основі розроблено загальну структуру роботи програми контролю доступу користувачів до семантичним БД, яка показана на малюнку 3.1.

Дана програма складається з наступних рівнів:

Інтерфейс побудований для взаємодії між користувачем і системою, за допомогою нього користувач може вибирати різні функціональності системи для роботи з СБД і отримати відповіді на них.

Рівень представлення даних дає відповіді користувачам. Відповіді можуть бути оформлені з використанням різних форматів даних, наприклад таких, як RDF / XML, Turtle або N3.

Рівень підготовки даних здійснює перевірку інформації користувачів і коректність заданих запитів. Він складається з наступних модулів:

модуль перевірки облікового запису користувача, який використовується для визначення їх рівнів та права доступу;

модуль перевірки запиту, який визначає коректність створеного запита; кожен запит повинен бути правильно складений відповідно до синтаксисом використовуваної мови запитів, наприклад SPARQL;

Рисунок 3.1 - Загальна структура програмного забезпечення контролю доступу користувачів до СБД

Рівень сервісів системи - основні функціональності системи, які користувачі можуть виконувати, такі, як редагування триплетів, редагування онтології, завдання рівнів безпеки даних, відправлення запитів, управління обліковим записом користувачів.

Рівень контролю доступу користувачів до семантичним БД є основним компонентом програми, в якому містяться наступні модулі:

модуль визначення рівнів безпеки елементів онтології і індивідів метаданих, розроблений на основі алгоритмів 2.1 - 2.3.

модуль визначення рівнів безпеки триплетів СБД, розроблений на основі алгоритму 2.4 (визначення узгоджених рівнів безпеки елементів СБД).

Семантична БД використовується для зберігання семантичних даних, де зберігаються онтології і метадані. У даній роботі в якості RDF-сховища обрана система Virtuoso Universal Server [2].

Діаграма програми SecSWD

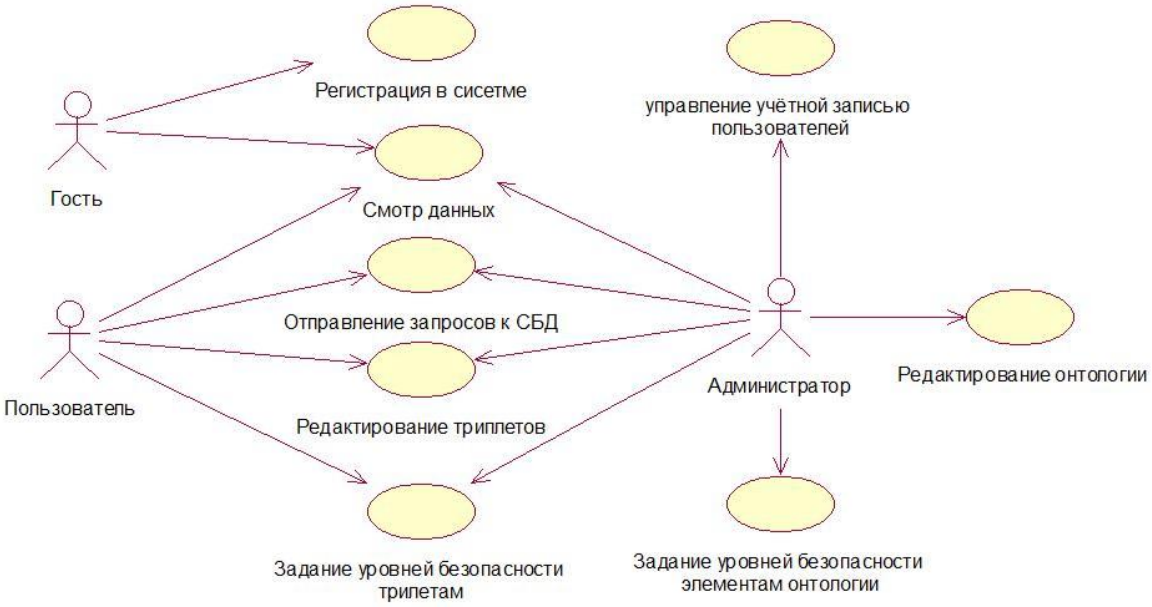


Рисунок 3.2 - Діаграма варіантів користувачів в програмах SecSWD і ContrLSD

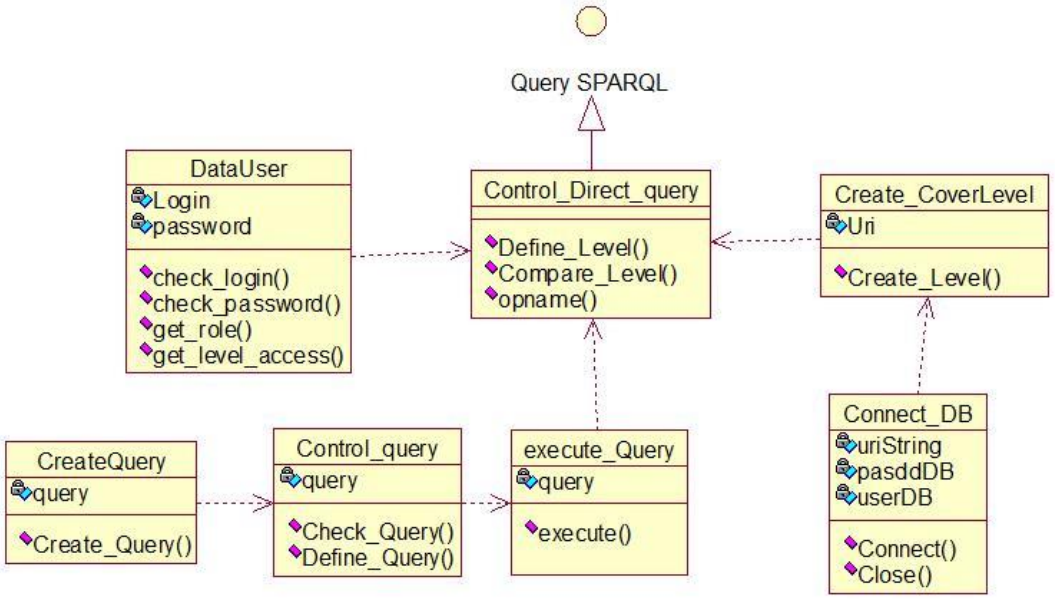


Рисунок 3.3 - Діаграма класів інтерфейсу Query SPARQL програми SecSWD

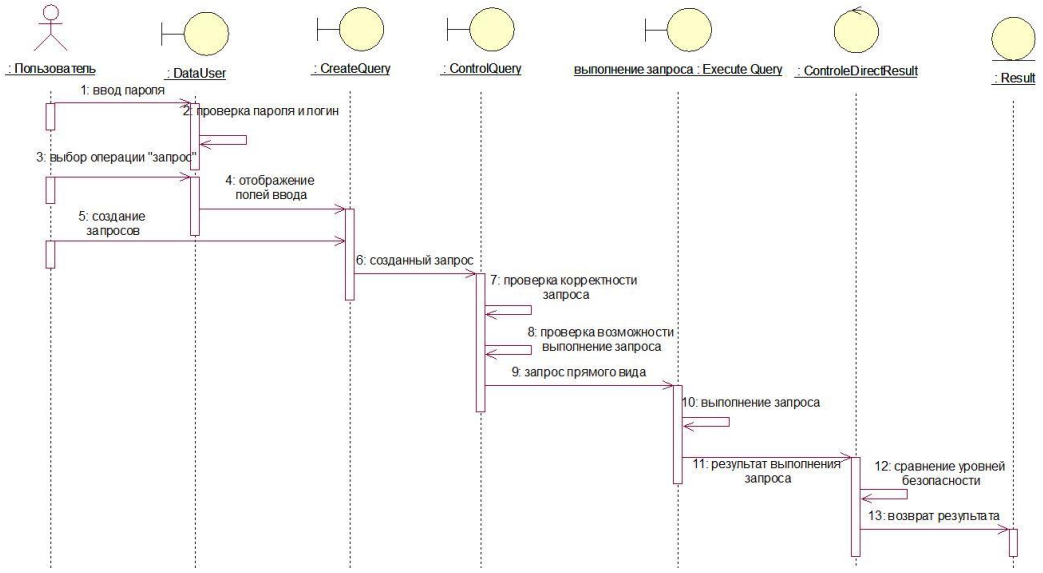


Рисунок 3.4 - Діаграма діяльності процесу контролю прямого запиту користувачів в програмі SecSWD

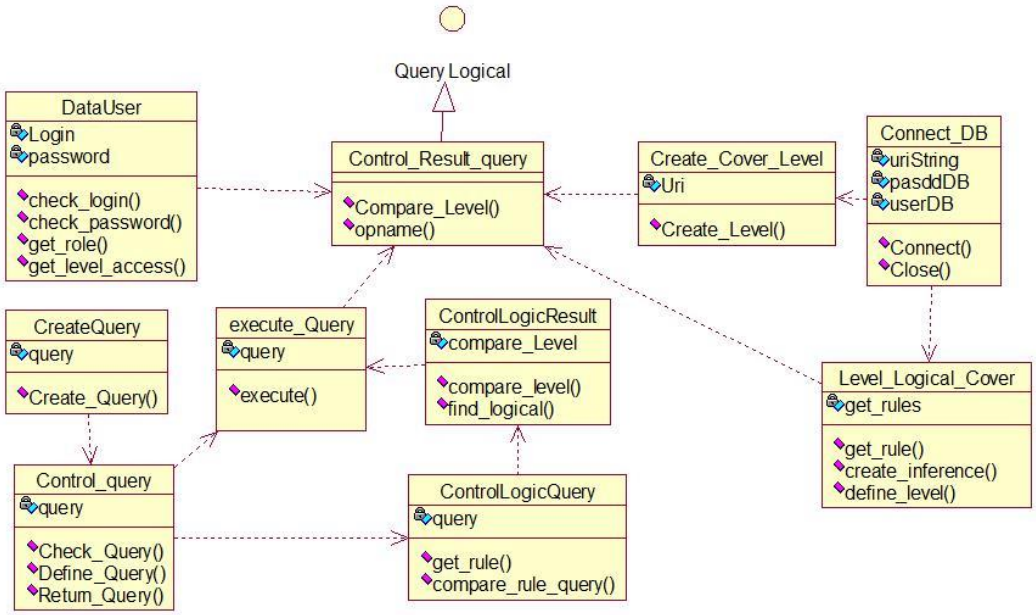


Рисунок 3.5 - Діаграма класів інтерфейсу Query Logical програми ContrLSD

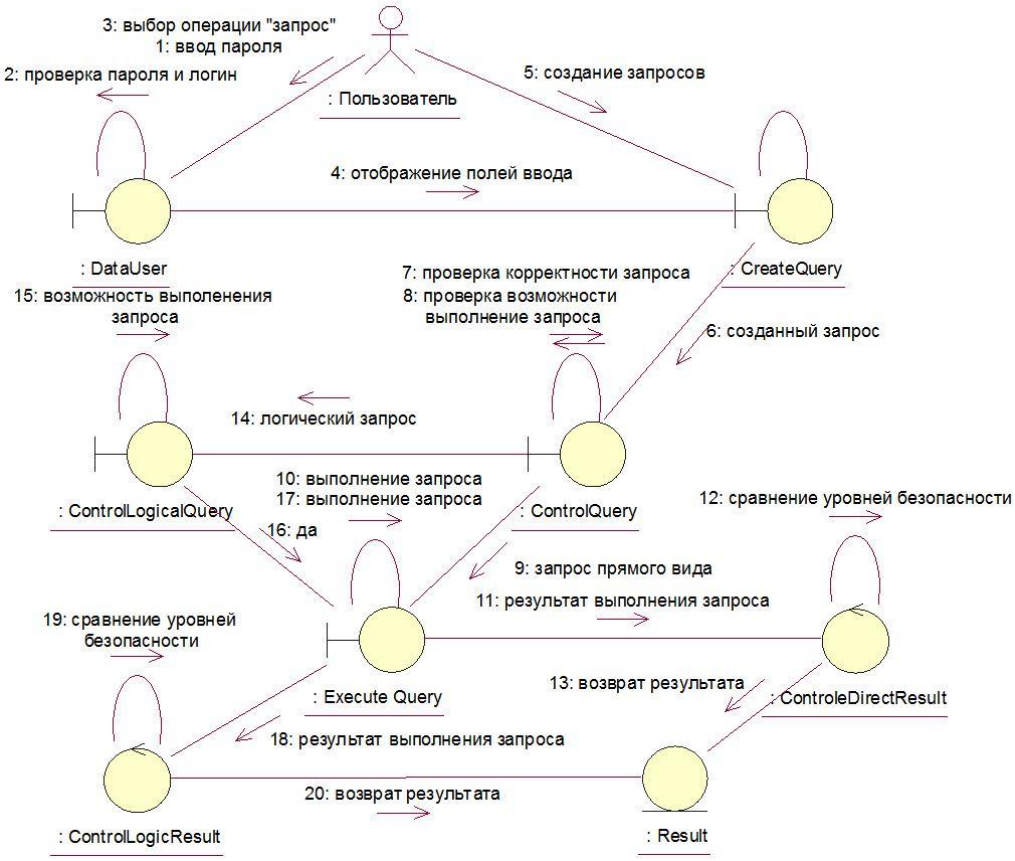


Рисунок 3.6 - Діаграма кооперації процесу контролю логічних запитів користувачів в програмі ContrLSD

Опис реалізації програми

Процес контролю доступу користувачів до елементам БД виконується наступним чином:

При кожному вході користувача в систему за допомогою модуля «перевірка облікового запису користувачів» виконується перевірка наявності його облікового записі, рівня доступу і прав доступу, інформація про яких зберігається в СБД.

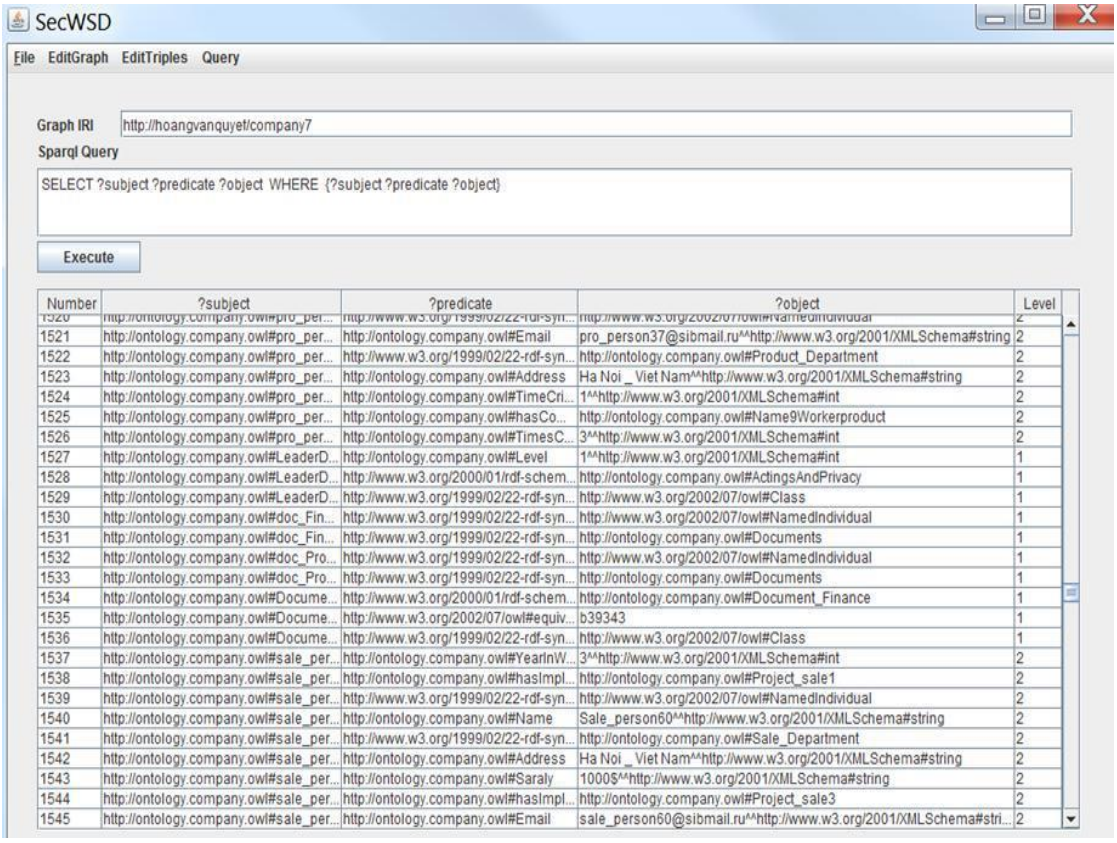
При відправці користувачем запитів система виконує їх перевірку з модуля «перевірка запиту».

Система визначає рівні безпеки всіх елементів онтологій в СБД.

Система дає користувачам відповіді на запит відповідно до його рівнями безпеки.

Дана програма гарантує, що користувачі виконають операції над даними семантичних БД і отримують результати відповідно до їх рівнями і правами доступу.

На малюнку 3.7 показаний графічний інтерфейс програми підтримки без-небезпеки роботи з семантичними базами знань SecWSD.



Number	?subject	?predicate	?object	Level
1521	http://ontology.company.owl#pro_per...	http://ontology.company.owl#Email	pro_person37@sibmail.ru^^http://www.w3.org/2001/XMLSchema#string	2
1522	http://ontology.company.owl#pro_per...	http://www.w3.org/1999/02/22-rdf-syn...	http://ontology.company.owl#Product_Department	2
1523	http://ontology.company.owl#pro_per...	http://ontology.company.owl#Address	Ha Noi _ Viet Nam^^http://www.w3.org/2001/XMLSchema#string	2
1524	http://ontology.company.owl#pro_per...	http://ontology.company.owl#TimeCri...	1^^http://www.w3.org/2001/XMLSchema#int	2
1525	http://ontology.company.owl#pro_per...	http://ontology.company.owl#hasCo...	http://ontology.company.owl#Name9Workerproduct	2
1526	http://ontology.company.owl#pro_per...	http://ontology.company.owl#TimesC...	3^^http://www.w3.org/2001/XMLSchema#int	2
1527	http://ontology.company.owl#LeaderD...	http://ontology.company.owl#Level	1^^http://www.w3.org/2001/XMLSchema#int	1
1528	http://ontology.company.owl#LeaderD...	http://www.w3.org/2000/01/rdf-schem...	http://ontology.company.owl#ActingsAndPrivacy	1
1529	http://ontology.company.owl#LeaderD...	http://www.w3.org/1999/02/22-rdf-syn...	http://www.w3.org/2002/07/owl#Class	1
1530	http://ontology.company.owl#doc_Fin...	http://www.w3.org/1999/02/22-rdf-syn...	http://www.w3.org/2002/07/owl#NamedIndividual	1
1531	http://ontology.company.owl#doc_Fin...	http://www.w3.org/1999/02/22-rdf-syn...	http://ontology.company.owl#Documents	1
1532	http://ontology.company.owl#doc_Pro...	http://www.w3.org/1999/02/22-rdf-syn...	http://www.w3.org/2002/07/owl#NamedIndividual	1
1533	http://ontology.company.owl#doc_Pro...	http://www.w3.org/1999/02/22-rdf-syn...	http://ontology.company.owl#Documents	1
1534	http://ontology.company.owl#Docume...	http://www.w3.org/2000/01/rdf-schem...	http://ontology.company.owl#Document_Finance	1
1535	http://ontology.company.owl#Docume...	http://www.w3.org/2002/07/owl#equiv...	b39343	1
1536	http://ontology.company.owl#Docume...	http://www.w3.org/1999/02/22-rdf-syn...	http://www.w3.org/2002/07/owl#Class	1
1537	http://ontology.company.owl#sale_per...	http://ontology.company.owl#YearInW...	3^^http://www.w3.org/2001/XMLSchema#int	2
1538	http://ontology.company.owl#sale_per...	http://ontology.company.owl#hasImpl...	http://ontology.company.owl#Project_sale1	2
1539	http://ontology.company.owl#sale_per...	http://www.w3.org/1999/02/22-rdf-syn...	http://www.w3.org/2002/07/owl#NamedIndividual	2
1540	http://ontology.company.owl#sale_per...	http://ontology.company.owl#Name	Sale_person60^^http://www.w3.org/2001/XMLSchema#string	2
1541	http://ontology.company.owl#sale_per...	http://www.w3.org/1999/02/22-rdf-syn...	http://ontology.company.owl#Sale_Department	2
1542	http://ontology.company.owl#sale_per...	http://ontology.company.owl#Address	Ha Noi _ Viet Nam^^http://www.w3.org/2001/XMLSchema#string	2
1543	http://ontology.company.owl#sale_per...	http://ontology.company.owl#Saraly	1000$^^http://www.w3.org/2001/XMLSchema#string	2
1544	http://ontology.company.owl#sale_per...	http://ontology.company.owl#hasImpl...	http://ontology.company.owl#Project_sale3	2
1545	http://ontology.company.owl#sale_per...	http://ontology.company.owl#Email	sale_person60@sibmail.ru^^http://www.w3.org/2001/XMLSchema#stri...	2

Рисунок 3.7 - Виконання запиту користувачів до СБД

Дана програма дозволяє контролювати доступ користувачів до ділових елементів даних. Користувачі можуть виконувати різні операції над даними відповідно їх прав і рівнями доступу.

Наприклад, користувач, який має рівень доступу рівний 2, отримує тільки триплети, у яких рівні безпеки не більше 2 (рисунок 3.7).

Користувач, який має права доступу на додавання даних може до-додавати дані в СБД (рисунок 3.8).

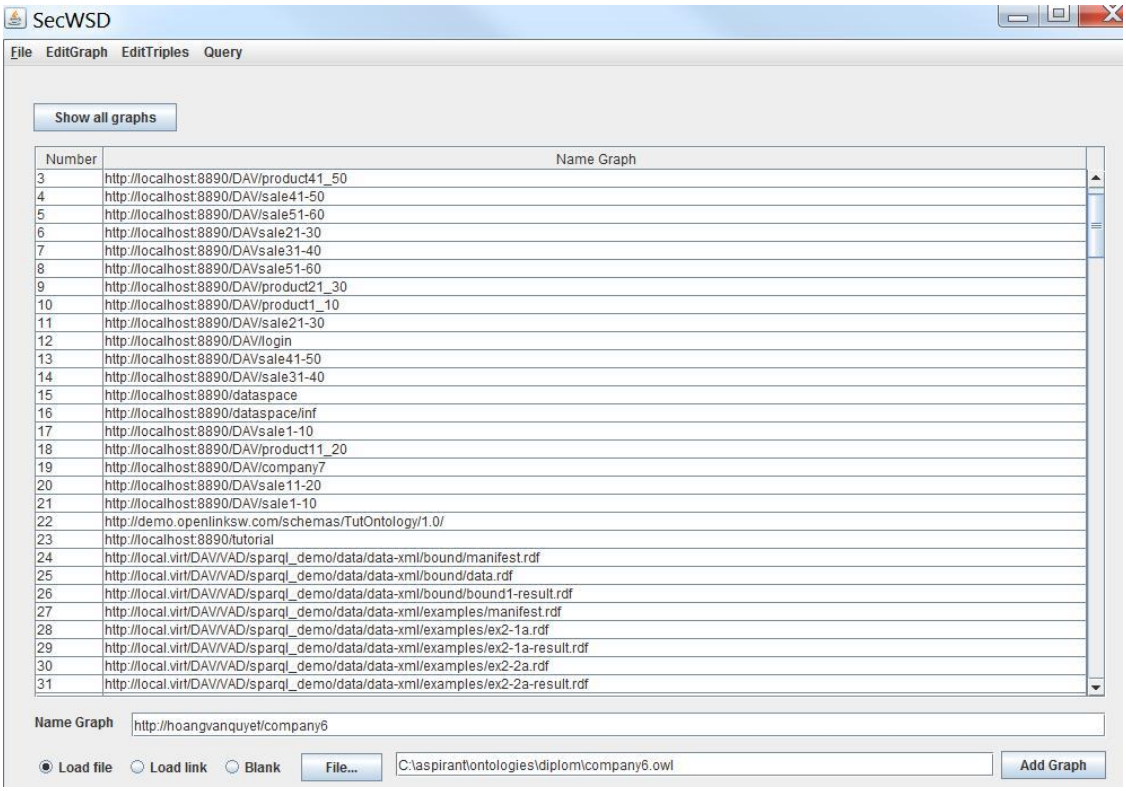


Рисунок 3.8 - Додавання RDF-даних в СБД

На малюнку 3.9 показана операція зміни RDF-триплетів в СБД.

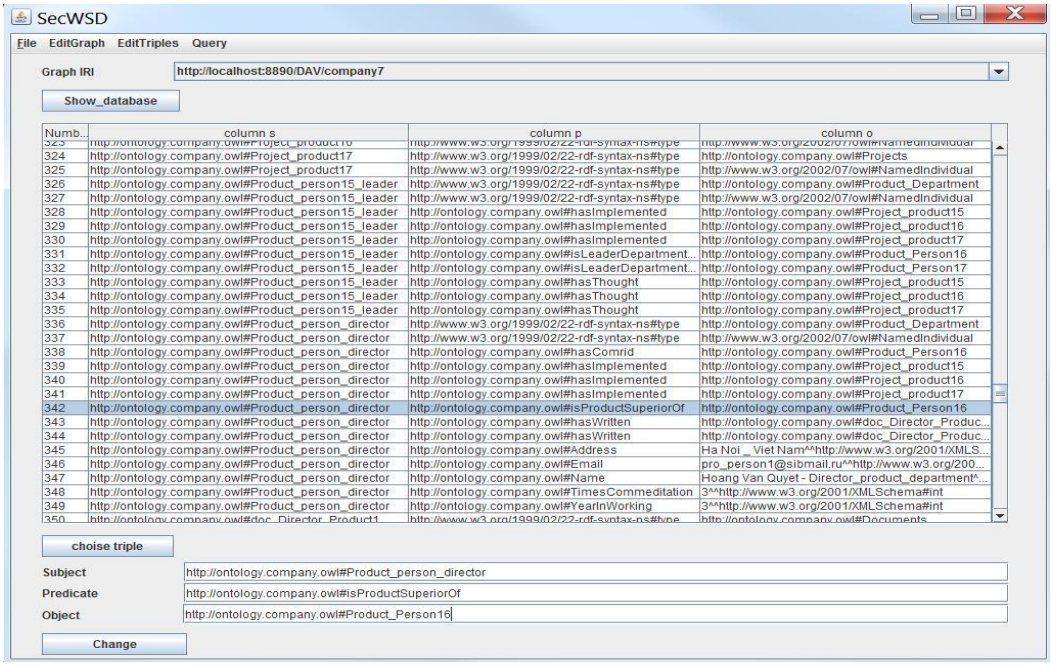


Рисунок 3.9 - Зміна RDF-триплетів в СБД

3.2. Експериментальні дослідження децентралізованої системи контролю доступу до ресурсів семантичного веб

В даному розділі наведені основні експерименти з дослідження ефективності розроблених методів і алгоритмів, таких, як визначення злагодисова рівнів безпеки елементів онтології; визначення рівнів без-небезпеки всіх триплетів і результатів логічних висновків семантичних БД; виявлення порушень результатів логічних висновків; контроль результатів, отриманих при виконанні запитів до СБД.

Всі експерименти були проведені на персональному комп'ютері, маю-щем наступну конфігурацію: процесор - AMD A8-4500M APU - 4 ядра - 1,90 GHz; оперативна пам'ять - 4 GB.

В якості тестових даних використовувалося безліч триплетів семантемческий БД компанії BAVIMILK. Загальна кількість понять склало 162, кількість відносин - 137, кількість триплетів - 0.23 мільйона. Приклад частини онтологічної моделі наведено на малюнку 3.10.

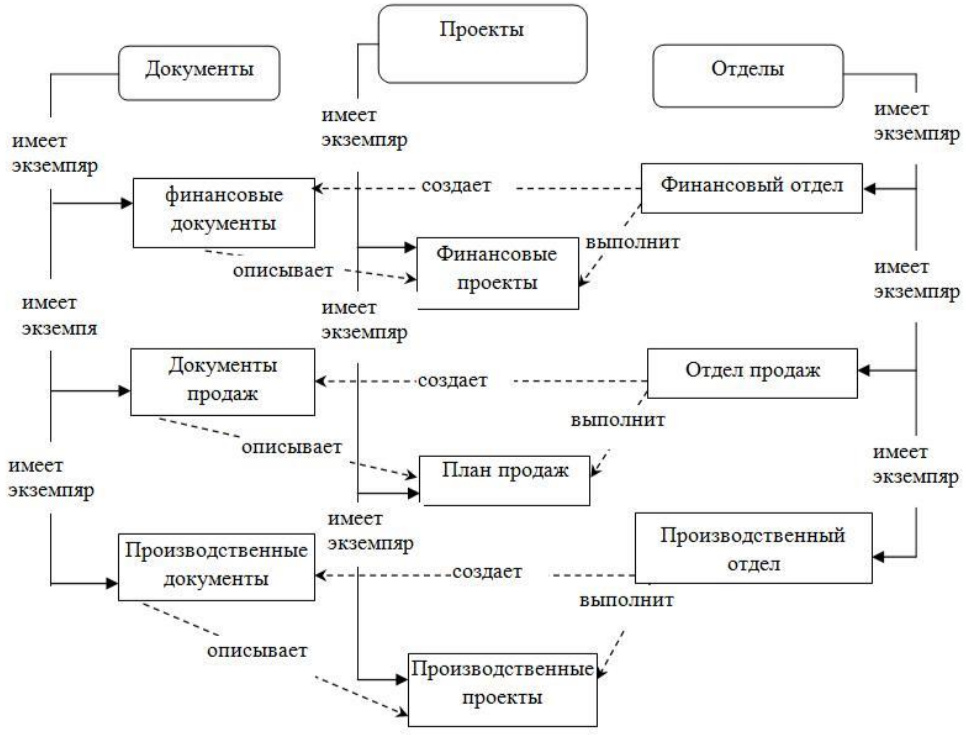


Рисунок 3.10 - Фрагмент використовуваної семантичної БД

Перше дослідження включає в себе 2 експерименту:

Перший експеримент наводиться з різними кількостями триплетів. Кількість понять онтології дорівнювало 80, а кількість триплетів змінювалося від 2000 до 16000.

Другий експеримент наводиться з різними кількостями класів. Кількість триплетів семантичної БД дорівнювало 10000, а кількість класів онтологій змінювалося від 20 до 160.

Залежності часу (мілісекунда - *мс)* визначення покриття безпеки семантичних БД від кількості триплетів і понять показані в діаграмах на малюнках 3.11 і 3.12.

Дані діаграми показують, що час визначення покриття безпеки семантичних БД мало залежить від кількості триплетів метаданих і сильно залежить від кількості класів онтологій.

Метою другого дослідження є порівняння ефективності алгоритму визначення покриття безпеки семантичних БД, розробленого в розділі 3.2, з іншим підходом, в якому кожному елементу БД заданий початковий рівень безпеки (підхід, використаний в моделі AC4RDF [84]).

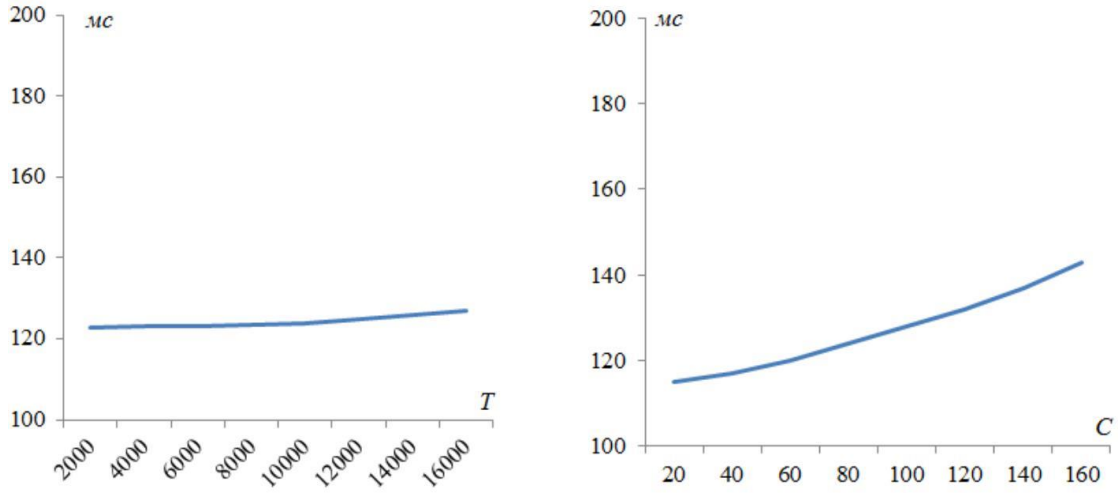


Рисунок 3.11 - Діаграми залежності часу від кількості триплетів

У таблиці 3.1 показані результати визначення обсягів (мегабайт - МБ) простору для зберігання триплетів в СБД в трьох випадках.

Таблиця 3.1 - Результати експерименту дослідження обсягів зберігання даних

Триплети	20000	40000	60000	8000	10000	12000	14000	16000
Обсяг 1 (МБ)	1.924	2.7	3.6	4.5	5.12	5.7	6.4	7.13
Обсяг 2 (МБ)	2.21	3.01	3.91	4.81	5.31	6.01	6.71	7.31
Обсяг 3 (МБ)	2.31	3.32	0.453	5.74	6.55	7.56	8.57	9.58

Діаграма залежності обсягу простору СБД від кількості триплетів показана на малюнку 3.13.

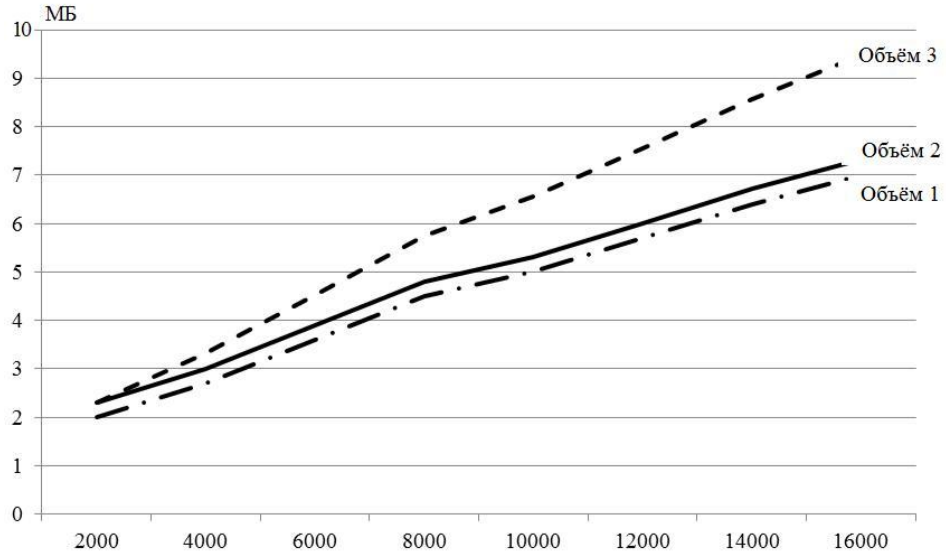


Рисунок 3.13 - Діаграма залежності обсягів простору зберігання даних від кількості триплетів

Дані результати показують, що відстань «Обсяг 2» і «Обсяг 1» обсягів простору СБД для зберігання даних не змінюється з рахунком кількості

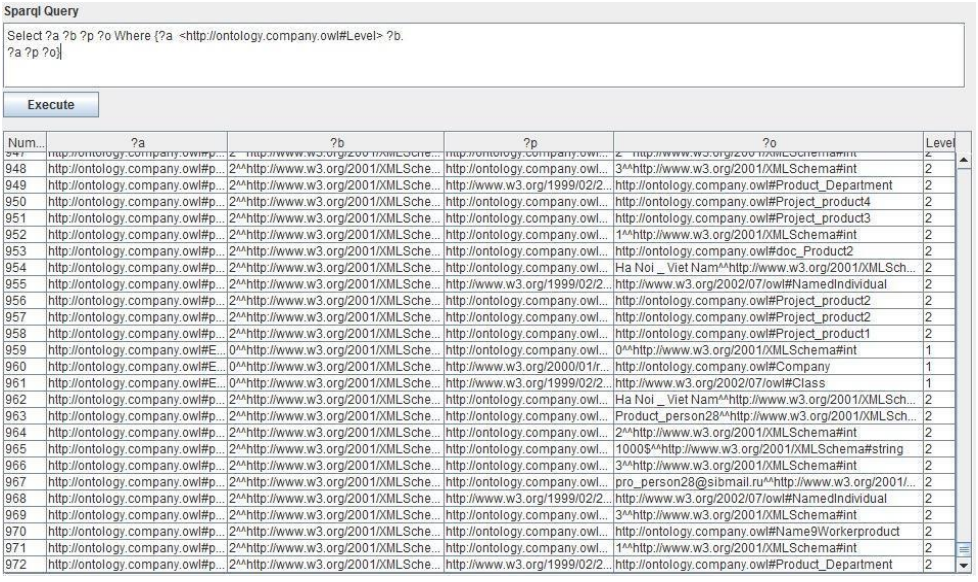
триплетів метаданих, а відстань «Обсяг 3» і «Обсяг 2» обсягів простору СБД збільшується з рахунком збільшення кількостей триплетів метаданих.

Це дозволяє робити висновок про те, що з використанням алгоритму, в розділі 2.3, для завдання і визначення рівнів безпеки всіх елементів і триплетів в СБД не потрібно великий обсяг простору СБД в порівнянні з підходом, в якому всім триплетів задані початкові рівні безпеки.

Контроль виконання прямих запитів користувачів до СБД

За допомогою алгоритму визначення покриття безпеки семантичних БД були визначені рівні безпеки всіх триплетів метаданих. Отримані значення рівнів безпеки використовувалися для виконання контролю результатів при виконанні прямого запиту користувачів до семантичної БД.

На малюнках 3.14 і 3.15 показані результати виконання прямого запиту до СБД користувачами, що мають різні рівні доступу $sl_1 = 2$ і $sl_2 = 4$.



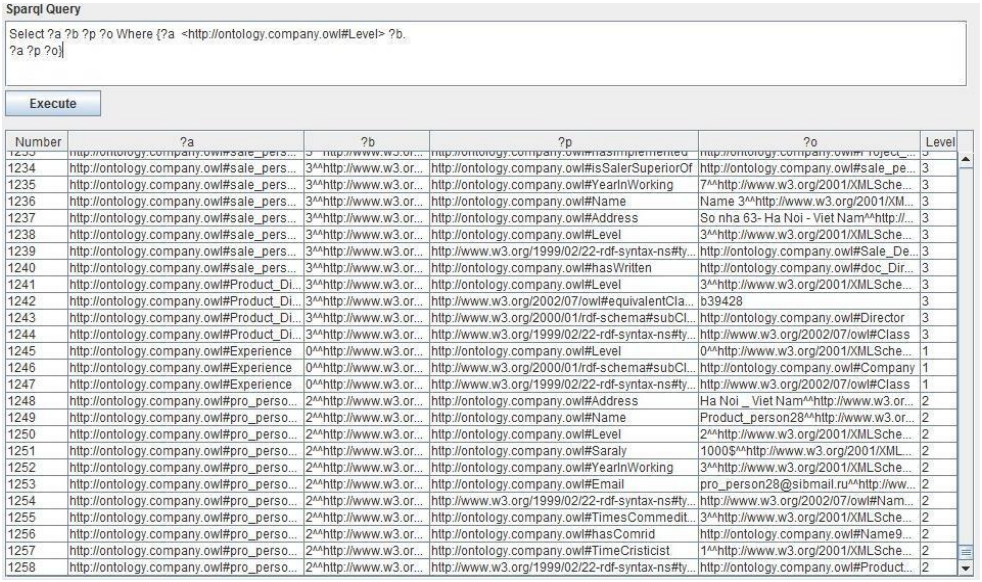
Num...	?a	?b	?p	?o	Level
948	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	3^^http://www.w3.org/2001/XMLSchema#int	2
949	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/1999/02/2...	http://ontology.company.owl#Product_Department	2
950	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Project_product4	2
951	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Project_product3	2
952	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	1^^http://www.w3.org/2001/XMLSchema#int	2
953	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#doc_Product2	2
954	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	Ha Noi _ Viet Nam^^http://www.w3.org/2001/XMLSch...	2
955	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/1999/02/2...	http://www.w3.org/2002/07/owl#NamedIndividual	2
956	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Project_product2	2
957	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Project_product2	2
958	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Project_product1	2
959	http://ontology.company.owl#E...	0^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	0^^http://www.w3.org/2001/XMLSchema#int	1
960	http://ontology.company.owl#E...	0^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/2000/01/r...	http://ontology.company.owl#Company	1
961	http://ontology.company.owl#E...	0^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/1999/02/2...	http://www.w3.org/2002/07/owl#Class	1
962	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	Ha Noi _ Viet Nam^^http://www.w3.org/2001/XMLSch...	2
963	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	Product_person28^^http://www.w3.org/2001/XMLSch...	2
964	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	2^^http://www.w3.org/2001/XMLSchema#int	2
965	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	1000$^^http://www.w3.org/2001/XMLSchema#string	2
966	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	3^^http://www.w3.org/2001/XMLSchema#int	2
967	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	pro_person28@sibmail.ru^^http://www.w3.org/2001/...	2
968	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/1999/02/2...	http://www.w3.org/2002/07/owl#NamedIndividual	2
969	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	3^^http://www.w3.org/2001/XMLSchema#int	2
970	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	http://ontology.company.owl#Name9Workerproduct	2
971	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://ontology.company.owl...	1^^http://www.w3.org/2001/XMLSchema#int	2
972	http://ontology.company.owl#p...	2^^http://www.w3.org/2001/XMLSche...	http://www.w3.org/1999/02/2...	http://ontology.company.owl#Product_Department	2

Рисунок 3.14 - Результат виконання прямого запиту користувачів, имечих рівень доступу $sl_1 = 2$

В результаті виконання запитів (рисунок 3.14) показано, що користувачі, що мають рівень доступу $sl_1 = 2$, отримують тільки результати, що мають рівні безпеки не більше 2.

В результаті виконання запитів (рисунок 3.15) показано, що користувачі, що мають рівень доступу $sl_2 = 4$, отримують тільки результати, що мають рівні безпеки не більше 4.

Таким чином, можна робити висновок про те, що алгоритми контролю виконання прямого запиту до семантичним БД гарантують, що користувачі використовують триплети відповідно до їх рівнями доступу.

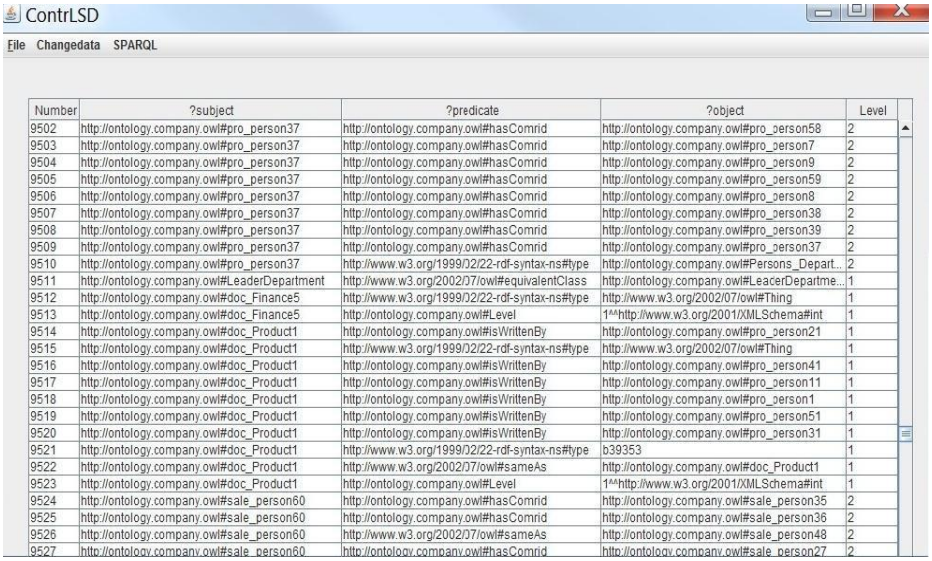


Number	?a	?b	?p	?o	Level
1234	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#isSalerSuperiorOf	http://ontology.company.owl#sale_pe...	3
1235	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#YearInWorking	7^^http://www.w3.org/2001/XMLSche...	3
1236	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#Name	Name 3^^http://www.w3.org/2001/XM...	3
1237	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#Address	So nha 63- Ha Noi - Viet Nam^^http://...	3
1238	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#Level	3^^http://www.w3.org/2001/XMLSche...	3
1239	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://www.w3.org/1999/02/22-rdf-syntax-ns#ty...	http://ontology.company.owl#Sale_De...	3
1240	http://ontology.company.owl#sale_pers...	3^^http://www.w3.or...	http://ontology.company.owl#hasWritten	http://ontology.company.owl#doc_Dir...	3
1241	http://ontology.company.owl#Product_Di...	3^^http://www.w3.or...	http://ontology.company.owl#Level	3^^http://www.w3.org/2001/XMLSche...	3
1242	http://ontology.company.owl#Product_Di...	3^^http://www.w3.or...	http://www.w3.org/2002/07/owl#equivalentCla...	b39428	3
1243	http://ontology.company.owl#Product_Di...	3^^http://www.w3.or...	http://www.w3.org/2000/01/rdf-schema#subCl...	http://ontology.company.owl#Director	3
1244	http://ontology.company.owl#Product_Di...	3^^http://www.w3.or...	http://www.w3.org/1999/02/22-rdf-syntax-ns#ty...	http://www.w3.org/2002/07/owl#Class	3
1245	http://ontology.company.owl#Experience	0^^http://www.w3.or...	http://ontology.company.owl#Level	0^^http://www.w3.org/2001/XMLSche...	1
1246	http://ontology.company.owl#Experience	0^^http://www.w3.or...	http://www.w3.org/2000/01/rdf-schema#subCl...	http://ontology.company.owl#Company	1
1247	http://ontology.company.owl#Experience	0^^http://www.w3.or...	http://www.w3.org/1999/02/22-rdf-syntax-ns#ty...	http://www.w3.org/2002/07/owl#Class	1
1248	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#Address	Ha Noi _ Viet Nam^^http://www.w3.or...	2
1249	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#Name	Product_person28^^http://www.w3.or...	2
1250	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#Level	2^^http://www.w3.org/2001/XMLSche...	2
1251	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#Saraly	1000$^^http://www.w3.org/2001/XML...	2
1252	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#YearInWorking	3^^http://www.w3.org/2001/XMLSche...	2
1253	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#Email	pro_person28@sibmail.ru^^http://ww...	2
1254	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://www.w3.org/1999/02/22-rdf-syntax-ns#ty...	http://www.w3.org/2002/07/owl#Nam...	2
1255	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#TimesCommedit...	3^^http://www.w3.org/2001/XMLSche...	2
1256	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#hasComrid	http://ontology.company.owl#Name9...	2
1257	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://ontology.company.owl#TimeCristicist	1^^http://www.w3.org/2001/XMLSche...	2
1258	http://ontology.company.owl#pro_perso...	2^^http://www.w3.or...	http://www.w3.org/1999/02/22-rdf-syntax-ns#ty...	http://ontology.company.owl#Product...	2

Рисунок 3.15 - Результат виконання запиту користувачів, що мають рівень доступу $sl_2 = 4$

Визначення рівнів безпеки результатів логічних висновків

Результат виконання алгоритму визначення покриття безпеки результатів логічних висновків в СБД показаний на малюнку 3.16. За допомогою данно-го алгоритму були визначені рівні безпеки всіх можливих результатів логічних висновків.



Number	?subject	?predicate	?object	Level
9502	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person58	2
9503	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person7	2
9504	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person9	2
9505	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person59	2
9506	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person8	2
9507	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person38	2
9508	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person39	2
9509	http://ontology.company.owl#pro_person37	http://ontology.company.owl#hasComrid	http://ontology.company.owl#pro_person37	2
9510	http://ontology.company.owl#pro_person37	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://ontology.company.owl#Persons_Depart...	2
9511	http://ontology.company.owl#LeaderDepartment	http://www.w3.org/2002/07/owl#equivalentClass	http://ontology.company.owl#LeaderDepartme...	1
9512	http://ontology.company.owl#doc_Finance5	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing	1
9513	http://ontology.company.owl#doc_Finance5	http://ontology.company.owl#Level	1^^http://www.w3.org/2001/XMLSchema#int	1
9514	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person21	1
9515	http://ontology.company.owl#doc_Product1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing	1
9516	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person41	1
9517	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person11	1
9518	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person1	1
9519	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person51	1
9520	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#isWrittenBy	http://ontology.company.owl#pro_person31	1
9521	http://ontology.company.owl#doc_Product1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	b39353	1
9522	http://ontology.company.owl#doc_Product1	http://www.w3.org/2002/07/owl#sameAs	http://ontology.company.owl#doc_Product1	1
9523	http://ontology.company.owl#doc_Product1	http://ontology.company.owl#Level	1^^http://www.w3.org/2001/XMLSchema#int	1
9524	http://ontology.company.owl#sale_person60	http://ontology.company.owl#hasComrid	http://ontology.company.owl#sale_person35	2
9525	http://ontology.company.owl#sale_person60	http://ontology.company.owl#hasComrid	http://ontology.company.owl#sale_person36	2
9526	http://ontology.company.owl#sale_person60	http://www.w3.org/2002/07/owl#sameAs	http://ontology.company.owl#sale_person48	2
9527	http://ontology.company.owl#sale_person60	http://ontology.company.owl#hasComrid	http://ontology.company.owl#sale_person27	2

Рисунок 3.16 - Покриття безпеки результатів логічних висновків в СБД

Для дослідження ефективності розробленого алгоритму були проведені експерименти з метою визначення залежності часу його виконання від кількості триплетів і понять в семантичної БД.

Діаграми залежності часу визначення покриття безпеки результатів логічних висновків від кількості триплетів *(T)*метаданих і кількості класів онтології *(C)* показані на малюнках 3.17 і 3.18.

На основі аналізу отриманих результатів можна зробити висновок про те, що час визначення рівнів безпеки результатів логічних висновків в семантичної БД значно залежить від кількості понять онтологій і не сильно від кількості триплетів метаданих.

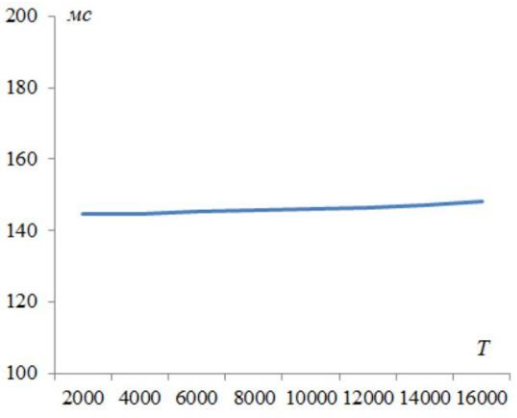


Рисунок 3.17 - Діаграма залежності

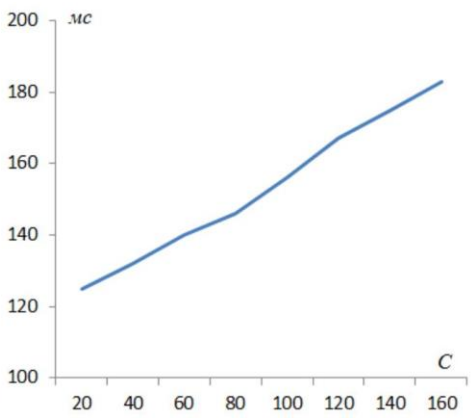


Рисунок 3.18 - Діаграма залежності

часу визначення покриття без пеки від кількості триплетів

часу визначення покриття без-пеки від кількості понять

Контроль виконання логічних запитів до семантичним БД

Для забезпечення безпеки інформації всіх індивідів в СБД, які належать класу *«виробничого директора»(Product_Director)* онтології, даного класу було поставлено рівень безпеки $sl_e = 3$. З урахуванням цього всі користувачі *U*, у яких рівні безпеки менше 3 $(sl_U < 3)$, не можуть мати можливості дізнатися інформацію будь-якого члена даного класу.

В семантичних БД зберігаються логічні правила, за допомогою яких користувачі можуть отримати нову інформацію. В якості таких правил використовувалися:

правило 1: «якщо об'єкт А керує якимось об'єктом Б, то А є об'єктом класу Product_Director ». Мовою SWRL дане правило може бути описано наступним чином: «AisProductSuperiorOf B → A owl: type Product_Director »;

правило 2: «Якщо об'єкт А керує якимось об'єктом В і якщо В є співробітником якогось об'єкта С, то слід, що А керує С ». Мовою SWRL дане правило може бути описано наступним чином: «A isProductSuperiorOf B, B iscomid C → A isProductSuperiorOf C».

На основі цих логічних правил, якщо користувачі хотіли дізнатися «хто є членом класу *Product_Director?»*, То він може відправляти один з двох наступних запитів:

Запит 1: «визначити всі об'єкти А, які обіймають керівні посади якимось об'єктом В », може бути описаний на мові SPARQL наступним чином

«SELECT? A? B WHERE {? A <http: //ontology.company.owl#isProductSuperiorOf> ? b.} ».

Запит 2: «визначити всі об'єкти А, які є елементами класу Product_Director », може бути описаний на мові SPARQL наступним: «SELECT? a WHERE {? a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http: //ontology.company.owl#Product_Director>} ».

Виконання запитів користувачами U_1, що мають рівень доступу $sl_U = 4$

Користувач U_1 може отримати результати логічних висновків, так як $sl_U > Sl_e$.

Якщо запити 1 і 2 виконані за допомогою алгоритмів контролю доступу до СБД, то на них користувачі отримують тільки такі відповіді:

Відповідь 1: http: //ontology.company.owl#Product_Person_director (рисунок 3.19).

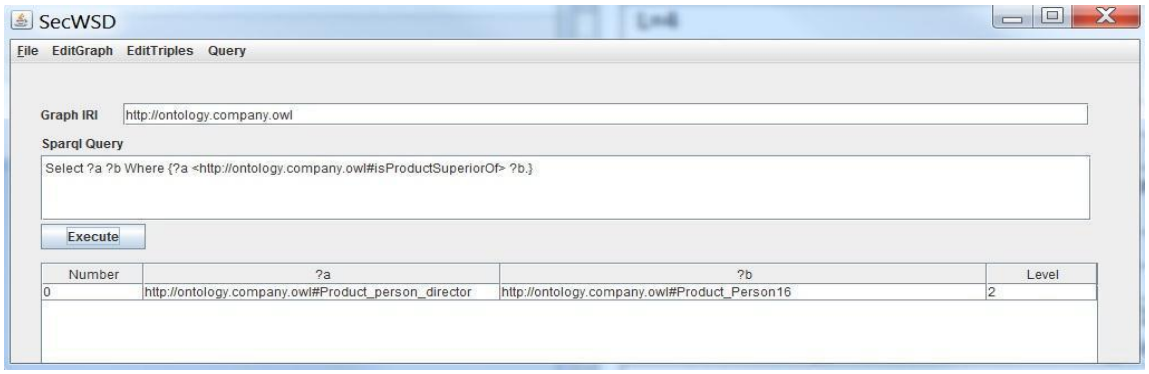


Рисунок 3.19 - Результат виконання запиту 1 за допомогою алгоритмів контролю доступу користувачів до СБД

Відповідь 2: порожня відповідь (рисунок 3.20).

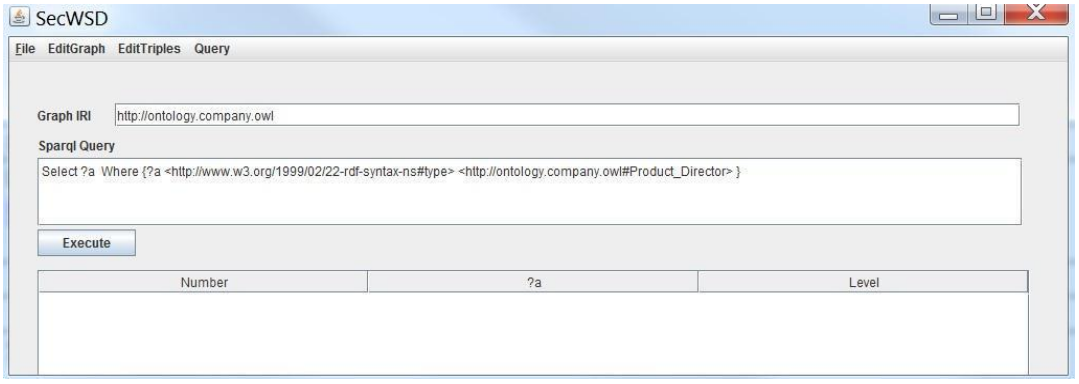


Рисунок 3.20 - Результат виконання запиту 2 за допомогою алгоритм ів контролю доступу користувачів до СБД

Якщо дані запити виконані за допомогою алгоритмів контролю логічних висновків в СБД, то на них користувачі отримують такі відповіді:

Відповідь 1: <http: //ontology.company.owl#Product_Person_director> (рис унок 3.21).

Відповідь 2: безліч триплетів, показане на малюнку 3.22. Користувачі мають право на доступ до всіх отриманих триплетів. Його користувачі не тільки получають інформацію про директора відділу, а також і про всіх членів відділу, якими даний директор керує.

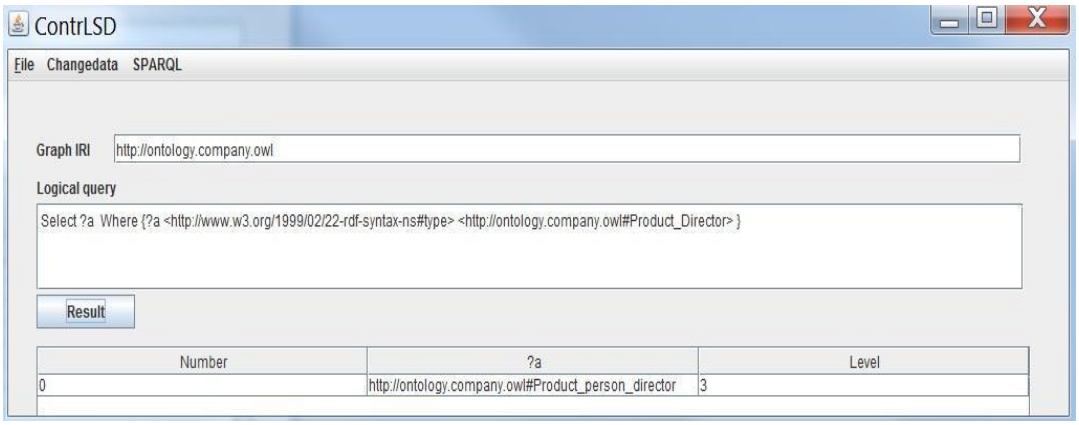


Рисунок 3.21 - Результат виконання запиту 1 за допомогою алгоритмів контролю результатів логічних висновків в СБД

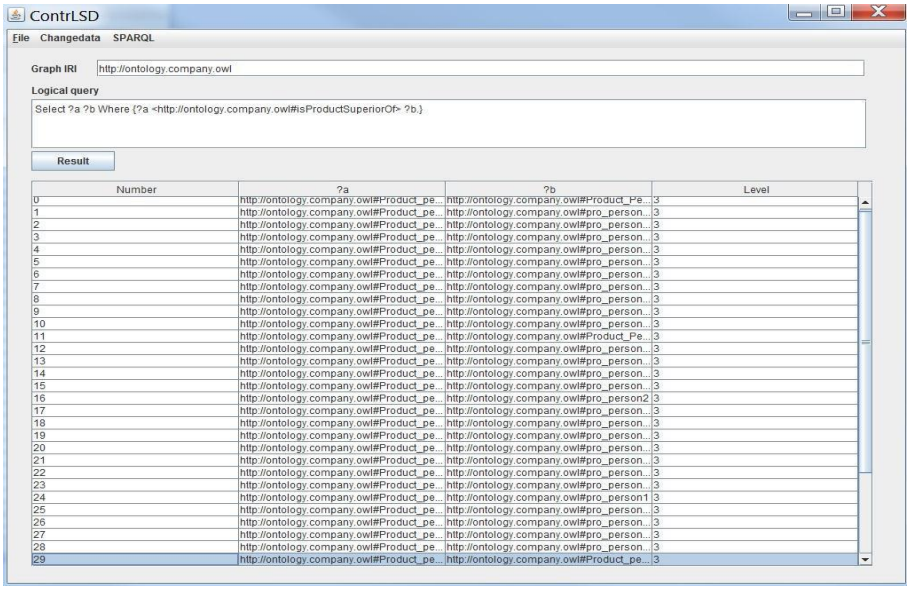


Рисунок 3.22 - Результат виконання запиту 2 за допомогою алгоритмів контролю результатів логічних висновків в СБД

Виконання запитів користувачами U_2, мають рівень доступ $sl_U = 2$

В даному експерименті розроблена програма повинна гарантувати, що користувачам U_2 можна отримати результати логічних висновків, так як sl_U sl_e.

Якщо запити 1 і 2 виконані за допомогою алгоритмів контролю доступу до СБД, то на них користувачі отримують тільки такі відповіді:

Відповідь

1: <http: //ontology.company.owl#Product_Person_director> (рисунок 3.23).

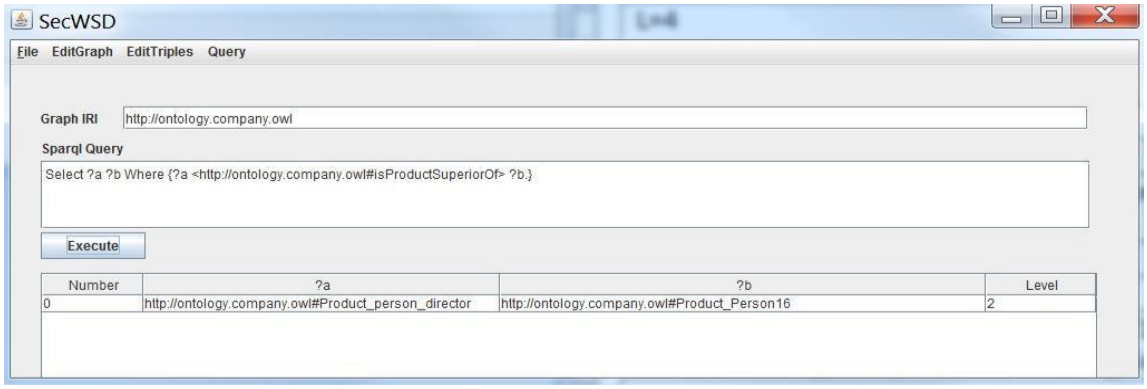


Рисунок 3.23 - Результат виконання запиту 1 за допомогою алгоритмів контролю доступу користувачів до СБД

Відповідь 2: порожній результат (рисунок 3.24).

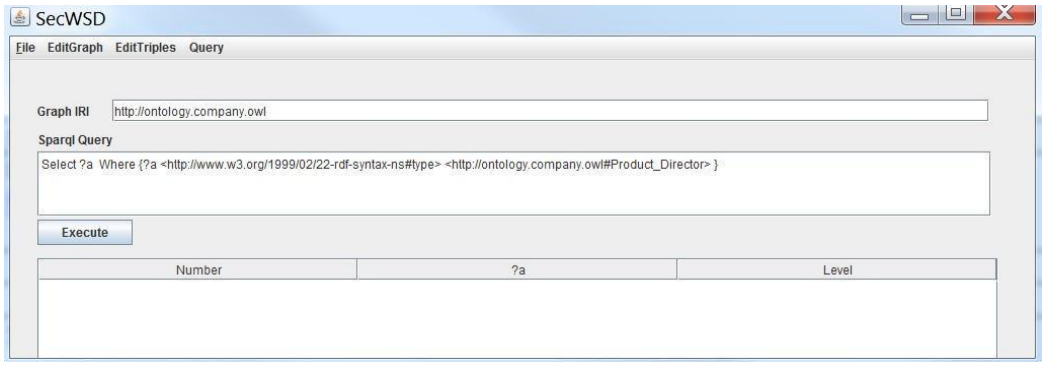


Рисунок 3.24 - Результат виконання запиту 2 за допомогою алгоритмів контролю доступу користувачів до СБД

З першого відповіді, користувач U_2 може дізнатися, що об'єкт *«Product_Person_director»* є директором відділу, отже, *безпека семантичної* БД *порушена.*

Якщо дані запити виконані за допомогою алгоритмів контролю результатів логічних висновків в СБД, то на них користувачі отримують такі відповіді:

Відповідь 1: порожній результат - не має права доступу (рисунок 3.25).

Відповідь 2: порожній результат - не має права доступу (рисунок 3.26).

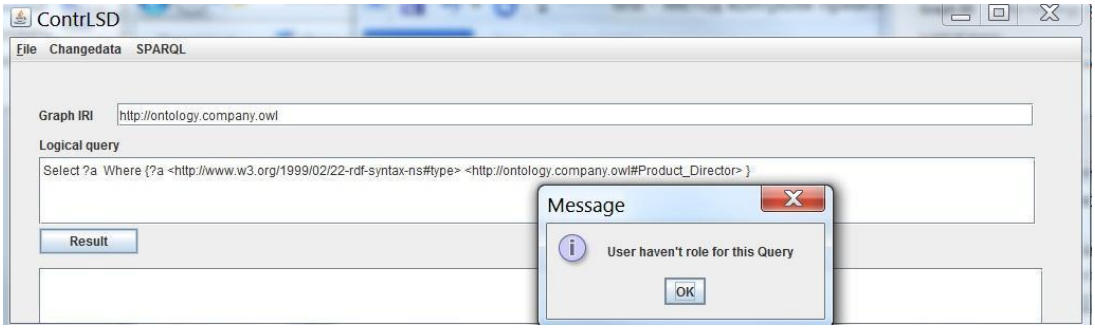


Рисунок 3.25 - Результат виконання запиту 1 за допомогою алгоритмів контролю результатів логічних висновків в СБД

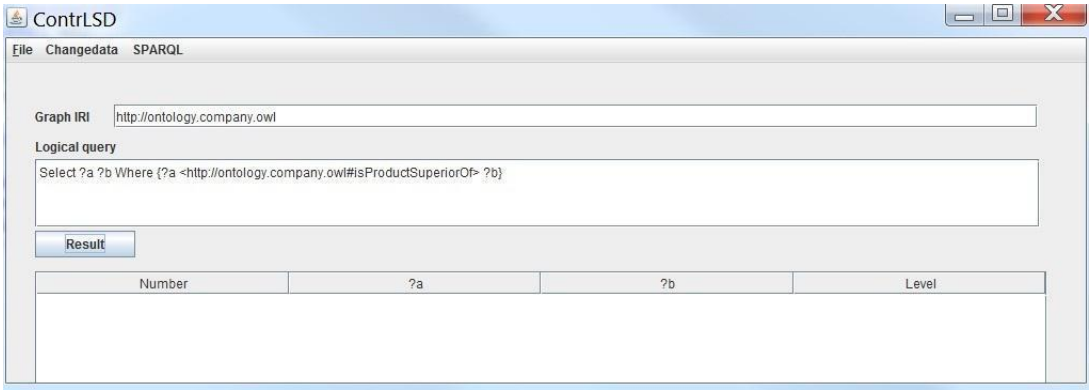


Рисунок 3.26 - Результат виконання запиту 2 за допомогою алгоритмів контролю результатів логічних висновків в СБД

Дані результати показують, що користувачі не мають права дізнатися інформацію про індивідів класу *Product_Director*, а це означає, що підтримується безпеку семантичної БД.

Розроблено метод виявлення порушень результатів логічних висновків алгоритм контролю результатів логічних запитів дозволяють користувачам отримати результати логічних висновків відповідно до їх рівнями доступу, що гарантують безпеку семантичної БД.

Висновки до розділу 3

В розділі 3 для контролю доступу користувачів до триплетів і елементам семантичних БД був розроблений ряд алгоритмів та їх проаналізовано програмну реалізацію: узгодження рівнів безпеки елементів онтологій і індивідів метаданих; визначення покриття безпеки семантичних БД; управління відповідями при виконанні запитів до СБД. На їх основі розроблено загальну структуру роботи програми контролю доступу користувачів до семантичним БД.

ВИСНОВКИ

У роботі надано стислий опис концептуальних положень, що відносяться до веб-сервісів, семантичних веб-сервісів, сервіс-орієнтованої архітектури, онтологій з точки зору семантичного вебу. Ці поняття викладаються згідно з пропозиціями комітету W3C.

Запропоновано підхід та деякі архітектурні рішення щодо опису, розміщення, інтеграції та пошуку ресурсів у семантичних грідах. Суть цього підходу полягає у наступному. Існує сильно пов’язана мережа вузлів. Ця мережа об’єднується по принципу спільно використовуваних ресурсів, наприклад, інформаційних ресурсів. Прикладом такої мережі може бути віртуальна організація. Таких мереж може бути багато, але вони мають так звані «слабі» зв’язки між собою. Таку структуру можна представити у вигляді слабо пов’язаної сукупності сильно пов’язаних підграфів. На рівні під графів більш ефективно вирішується задача управління ресурсами. Усі види ресурсів мають власні онтології. Онтології підграфів зберігаються у єдиному реєстрі. Що стосується глобального графу, от він також має онтології (які також розміщуються у реєстрах), але вони перш за все вирішують проблему співставлення онтологій підграфів для вирішення задачі інтеграції і спільної роботи усієї грід-структури. Запропоновані основні принципи, які повинні підтримуватись при створенні такої архітектури.У зв’язку з цим принципово важливою є задача є побудова онтологій усіх можливих ресурсів грід. У дипломній роботі наводяться спрощені онтології деяких видів ресурсів грід у вигляді таксономій які є основою для опису особливостей архітектурних рішень для побудови системи управління ресурсами.

Розроблено метод виявлення порушень результатів логічних висновків, алгоритм контролю результатів логічних запитів дозволяють користувачам отримати результати логічних висновків відповідно до їх рівнями доступу, що гарантують безпеку семантичної БД.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Kifer M. F-Logic: A higher-order language for reasoning about objects, inher-itance, and scheme // Proc. of the ACM SIGMOD international conference on manage-ment of data. - 2018. - P. 134-146.

2. Guinness D. DAML-ONT: An ontology language for the Semantic Web // Spinning the Semantic Web: Bringing the World Wide Web to its full potential. - Mas-sachusetts: MIT Press, 2013. - 479 p.

3. Hendler J. DAML + OIL: An ontology language for the Semantic Web // IEEE Intelligent Systems. - 2012. - V. 17. - №5. - P. 72-80.

4. A Survey of Semantic Web Technology in the Oil and Gas Industry // http://www.w3.org/2001/sw/sweo/public/ UseCas-es / Chevron /.

5. Тузовський А.Ф., Васильєв І. А. Структура системи управління знаннями // Праці міжнародного симпозіуму «Інформаційні та системні технології в індустрії, освіті та науці. - Караганда: Видавництво КарГТУ, 2013. - С. 286-288.

6. Програмна система «SemDL - система управління сховищем електронних ресурсів з використанням семантичних технологій» / Ле Хоай, А.Ф. Тузовський // Свідоцтво про державну реєстрацію програми для ЕОМ № 2013613266. М .: Федеральна служба з інтелектуальної власності- 2015.

7. Хоанг Ван Кує. Основні завдання організації безпеки для Семантичних веб / В.К. Хоанг, А.Ф. Тузовський // VII Міжнародна науково-практична конференція «Електронні засоби та системи управління». - Томськ, 2017. - С. 197-200.

8. Зегжда Д.П. Івашко А. М. Основи безпеки інформаційних систем / Д.П. Зегжда, А. М. Івашко. - М .: Гаряча Лінія - Телеком, 2016. - 452 с.

9. Ярочкин В.І. Інформаційна безпека. - М .: Академічний проект Фонд "Мир", 2013. - 638 с.

10. Казанцева С.Я. Правове забезпечення інформаційної безпеки. -М .: Академія, 2015. - 239 с.

11. Бєлов О.Б. Основи інформаційної безпеки. - М .: Гаряча лінія-Телеком, 2016. - 544 с.

12. Шаньгина В. Ф. Захист комп'ютерної інформації. Ефективні методи і засоби. - Москва: ДМК Пресс, 2010. - 544 с.