МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ЛЯХОЦЬКИЙ Олександр Сергійович

Програмний модуль для побудови семантичних порталів/ The software module for semantic models building

спеціальність: 8.05010302 - Інженерія програмного запезпечення
магістерська програма - Інженерія програмного забезпечення

Магістерська робота

Виконав студент групи ІПЗм-21
О. С. Ляхоцький

_____

Науковий керівник:
к.е.н., доцент ГОНЧАР Л.І.

_____

Магістерську роботу допущено
до захисту:

"___" _____ 20___ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

РЕЗЮМЕ

Дипломна робота містить 109 сторінок, 23 рисунків, список використаних джерел із 53 найменувань, 2 додатки.

Метою дослідження є побудова web-орієнтованого програмного комплексу для аналізу і зберігання інформаційних ресурсів, а також здійснення пошуку та фільтрації даних.

Об'єкт дослідження – семантичні портали.

Предмет дослідження – web-орієнтований програмний комплекс для побудови семантичних порталів

Одержані висновки та їх новизна: здійснено програмну реалізацію автоматичного рубрикування на основі семантичного аналізу вмісту інформаційних ресурсів

Ключові слова: програмний засіб, семантичний портал, рубрикатор, метаданні, веб - ресурс.

RESUME

Diploma contains 109 pages, 23 figures, list of references with 53 titles, 2 annexes.

The aim of the thesis is to build a web-based software for analysis and storage of information resources and the implementation of a Search and filter data.

Object of research is Semantic portals.

The subject of research - web-oriented software package for building semantic portals.

The resulting conclusions and innovation: Done heading automatic software implementation based on semantic analysis of the content of information resources.

Keywords: software tool, semantic web, Categories, metadata, web, resource.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,СКОРОЧЕНЬ І ТЕРМІНІВ

ЕБ – електрона бібліотека;

СКВІР - Система колективного використання інформаційних ресурсів;

Мережа – система зв'язку комп'ютерів або обчислювального обладнання.

Інтернет – всесвітня система об'єднаних комп'ютерних мереж для зберігання і передачі інформації;

СУБД – системи управління базами даних;

OWL(Web Ontology Language) – мова онтології для інтернета на основі XML/Web стандарта;

RDF(Resource Description Framework, RDF) — Розроблена консорціумом W3C технологія семантичної павутини, яка включає в себе середовище опису ресурсів;

XTM(XML Topic Maps) - дозволяє адаптувати стандарт topic map (ISO 13250) для спільного використання з XML і іншими стандартами Web. Мова XTM надає синтаксис опису topic maps, заснований на XML.

URI(Uniform Resource Identifier) — ідентифікатор ресурса..

SPARQL(Protocol And RDF Query Language) — нова мова запитів для швидкого доступу до даних RDF;

АЛОТ - автоматизована лінгвістична обробка текстів;

SEAL(SEmantic portAL) - основна ідея складається в побудові Intranet додатків, здатних надати інформацію для користувачів і агентів з обліком їхньої семантичної структури;

Інтранет (Intranet) - на відміну від Інтернету, це внутрішня приватна мережа організації;

SOAP(Simple Object Access Protocol*)* — простий протокол доступа до об’єктів, вплоть навіть до специфікації або іншими словами це протокол обміну структурованими повідомленнями в розподіленому вичислительному середовищі.

ВСТУП

На сьогоднішній день у всесвітній мережі Інтернет знаходиться 1 099 511 627 776 гігабайт або 1 зеттабайт даних. Більша частина цих даних є не структурованою і , мало того, ще й зберігається в різних форматах даних, що погіршує їх обробку на програмному рівні і значно уповільнює фільтрацію. Для вирішення цієї проблеми було створенно Semantic Web [1,2].

Semantic Web — це надбудова над сучасною Всесвітньою павутиною, яка покликана зробити інформацію, що розміщена в мережі, зрозумілішою для комп'ютерів.

На думку авторів технології Semantic Web, це може досягти за допомогою введення метаданих, які повинні супроводжувати будь-яку інформацію й розповідати про її походження, формат і багато іншого, що повинне радикальним способом полегшити пошук інформації в Web і її обробку.

Актуальність теми .

Ми знаємо, що майже вся інформація в Інтернеті зберігається в текстовій формі. Ні для кого не секрет також, що прогрес в обропці людської мови (англ. Natural Language Processing)розвивається дуже повільно. Комп'ютери не можуть сприймати і розуміти словесну інформацію, розміщену в Інтернеті, і в найближчому майбутньому, ймовірно, не зможуть. Тому розробка веб-порталів для побудови семантичних моделей є надзвичайно актуальною задачею. Вона дасть людям швидше знаходити і фільтрувати інформацію, без людської допомоги визначати куди завантажувати текстовий документ в яку рубрику, тему, що дасть як користувачам і програмам чітку структуру даних.

Зв'язок роботи з науковими програмами, планами, темами

Тема магістерської роботи відповідає напрямку наукових досліджень кафедри комп'ютерних наук Тернопільського національного економічного університету.

Мета і задачі дослідження

Метою наукового дослідження є побудова веб-семантичного порталу для того, щоб зробити інформацію, передану в Web, більш формалізованою й зручною для машинного сприйняття, для її ідентифікації та класифікації Для досягнення мети було поставлено такі задачі:

- здійснено аналіз існуючих підходів до розробки порталів знань на основі технологій Semantic Web.
- спроектовано оптимальну модель структури зберігання й доступу до інформаційних ресурсів;
- здійснено дослідження порталів знань і семантичних порталів;
- розроблено тематичний рубрикатор для СКВІР;
- здійснено програмну реалізацію автоматичного рубрикування на основі семантичного аналізу вмісту інформаційних ресурсів.
- здійснено тестування розробленого програмного продукту.

Об'єктом дослідження є семантичні портали.

Предметом дослідження є web-орієнтований програмний комплекс для побудови семантичних порталів

Методи досліджень базуються на використанні технологій Ruby on Rails, Microsoft Share Point, бібліотеки та методів інтелектуального аналізу даних DataMining..

Наукова новизна одержаних результатів

Дістав подальший розвиток семантичний аналіз вмісту інформаційних ресурсів для реалізації автоматичного рубрикування .

Вперше для структуризації та класифікації інформаційних ресурсів використовується підхід Topic Maps.

Практичне значення одержаних результатів

Здійснено реалізацію автоматичного рубрикування на основі семантичного аналізу вмісту інформаційних ресурсів, що дасть можливість використовувати семантичний веб-портал в межах організації, підприємства, електронних бібліотек та дозволить структуровано зберігати інформаційні ресурси по рубриках та темах.

Особистий внесок здобувача

Автором самостійно розроблена програмний засіб для побудови семантичних порталів.

Апробація результатів роботи.

Основні результати роботи представлено на VI Всеукраїнській школі-семінарі молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології», АСІТ'2016, яка відбулась 20-21 травня 2016, м.Тернопіль.

Публікаці.

Гончар Л.І., Ляхоцький О.С. Програмний засіб побудови семантичних порталів // Матеріали VI Всеукраїнської школи-семінару молодих вчених і студентів " Сучасні комп'ютерні інформаційні технології ", АСІТ'2016-Тернопіль : ТНЕУ, 2016. - с.110-111.

РОЗДІЛ І
ОГЛЯД ТЕХНОЛОГІЙ SEMANTIC WEB

1.1. Поняття Semantic Web

Semantic Web представляє собою мережу інформаційних вузлів, які пов’язанні один з одним таким чином, щоб накопичена інформація мала можливість легко оброблятися на стороні комп’ютера, тобто Основною задачею Semantic Web є зробити інформацію, передану в Web, більше формалізованою й зручною для машинного сприйняття, зокрема, для того щоб її можна було ідентифікувати й класифікувати. Цю технологію можна розглядати як ефективний засіб представлення даних в Всесвітній паутині інтернету або як глобальну пов’язану базу даних.

Автори Semantic Web технологій можуть цього досягнуто за рахунок впровадження метаданих, які повинні супроводжувати будь-яку інформацію і говорити про її походження, формат і інші необхідні дані, які мають полегшити і допомогти в пошуку необхідної інформації в Інтернеті та її обробці [9].

Ґрунтуючись на відкритих стандартах, технології Semantic Web дозволяють описувати й виділяти значеннєву інформацію (семантику) з довільних даних, зокрема змісту документів або коду додатків. Говорячи, що машина розуміє семантику документа, мається на увазі не тільки інтерпретація набору символів, що зустрічаються в документі, але й те, що машина розуміє зміст документа, тобто значення документа в цілому. Наступні технології є основними в складі Semantic Web:

- Глобальна схема імен (URI);
- Модель опису даних (RDF);
- Мова опису словників (RDFS);
- Засоби опису зв'язків між об'єктами даних (онтології, і мова

їхнього опису OWL);

- Метадані;
- SPARQL (англ. *Protocol And RDF Query Language*) — нова мова запитів для швидкого доступу до RDF даних. Використання простої мови і протоколу SPARQL, програма може аналізувати RDF-описи ресурсів і отримувати необхідні з всесвітньої мережі дані.

Ключовим елементом технології Semantic Web є унікальна система ідентифікації об'єктів. URI (Uniform Resource Identifier) - це ідентифікатор об'єкта (ресурсу) в глобальній мережі. Будь-який елемент схеми або семантичної моделі даних мережі повиннен мати свій власний унікальний адрес (URI). В даний час існує два типи ідентифікаторів.

Універсальний покажчик ресурсів (Uniform Resource Locator, сокр. URL) - це URI, що, крім ідентифікації ресурсу, указує на спосіб обігу з ресурсом шляхом опису способу доступу до нього або його положення в мережі.

Універсальне ім'я ресурсу (Uniform Resource Name, сокр. URN) - це URI, що ідентифікує ресурс за допомогою ім'я в певному просторі імен. Це дозволяє посилатися на ресурс без використання інформації про його розташування.

Другий базовий компонент Semantic Web - це модель даних Resource Description Framework (RDF), що дозволяє об'єднати інформацію з довільних джерел. Формат RDF найбільш корисний у забезпеченні спільного використання інформації, зміст якої може однаково інтерпретуватися різними програмними агентами. Специфіка моделі даних RDF складається тім, що ресурси й властивості ідентифікуються за допомогою глобальних ідентифікаторів (URI). RDF описує предметну область у термінах ресурсів, властивостей ресурсів і значень властивостей. RDF-Дані можна розцінювати як сукупність тверджень - суб'єкт, предикат і об'єкт твердження, і представляти у вигляді спрямованого графа, утвореного такими твердженнями.

Метадані – це дані про дані. На даний час багато розроблено схем описання метаданих, але серед всіх виділяють наступні:

Topic Maps (XMT) [47] - стандарт ISO (ISO / IEC 13250: 2003) для подання та обміну знаннями з точки зору пошуку інформації.

Text Encoding Initiative (TEI) [48] - міжнародний проект з розробки нормативів для розмітки (marking up) електронних текстів, таких як романи, п’єси, вірші; головним чином для підтримки досліджень в гуманітарній сфері.

Metadata Encoding and Transmission Standard (METS) [49] - стандарт кодування і передачі метаданих, був розроблений для задоволення потреби в стандартній структурі даних для опису складних цифрових бібліотечних обьектов.

Metadata Object Description Schema (MODS) [50] - схема метаданих опису об'єктів, яка була виведена з MARC 21, і призначена для перенесення відібраних даних з існуючих записів метаданих MARC 21 або для створення оригінальної запису опису ресурса.

Encoded Archival Description (EAD) [51] - закодоване архівне опис, було розроблено як спосіб розмітки даних, які містяться в пошукових засобах, для того, щоб вони знаходилися й показувалися в оперативному режіме.

Learning Object Metadata (LOM) [52] - стандарт IEEE 1484.12.1-2002 метаданих об'єктів навчального процесу для повторного використання ресурсів навчального характеру, таких як комп'ютерного та дистанційного навчання.

Базовими для Semantic Web в даний момент визнаються стандарти Dublin Core, FOAF, SIOC і DOAP [54] .FOAF (Friand-Of-A-Friend) [55 - 57] - це формат машинно-оброблюваних сторінок, що описують персональну інформацію про людей і їх діяльності (фотографії, календарі, блоги та інше) в форматі XML.SIOC (Semantically-Interlinked Online Communities) [58] - документи, що описують онлайн-спільноти. SIOC забезпечує взаємозв'язок таких засобів обговорення інформації, як блоги, форуми і поштові розсилки

між собой.

Description of a Project Description of a Project (DOAP) [59] - документи, що описують в мережі проекти з відкритим вихідним кодом.

Ці всі стандарти використовуються і на даний час, тільки зараз їх починають компанувати і в кожен з них слід додаляти щось своє, універсальне і унікальне, що дасть змогу краще описати і представити метадані і поєднати їх до власного проекту.

1.2.. Принципи побудови моделі RDF

Наступний рівень у піраміді технологій Semantic Web займає *RDF Schema* – мова опису словників RDF-Термінів. RDFS служить фундаментом для більше багатих мов опису онтологій предметної області, які дозволяють адаптувати до Web системи логіки й забезпечити семантичну обробку даних. Схема RDF представляє собою систему типів для Semantic Web і дозволяє визначити класи ресурсів і властивості як елементи словника, зокрема задати, які властивості з якими класами можуть бути використані.

Схема RDF була розроблена як проста модель типізації даних для RDF. RDF є мовою загального застосування для подання інформації в Інтернет. Дана специфікація описує як використовувати RDF для опису RDF-словників. Вона визначає базовий словник, призначений для цих цілей і прийняті угоди, які можуть бути використані при створенні додатків Semantic Web для підтримки більш складних словників RDF-описів. Мова опису словника RDF визначає класи і властивості, які можуть бути використані для опису інших класів і властивостей, а також виробляти деякі більш складні речі, такі, як створення діапазонів і областей для властивостей.

Базовий будівельний блок моделі даних RDF - твердження, що представляє собою трійку: ресурс, іменована властивість і його значення. У термінології RDF ці три частини твердження називаються відповідно: суб'єкт (subject), предикат (predicate) і об'єкт (object) [15]. Ресурсом у цьому випадку

називають усе, що описується засобами RDF. Це може бути звичайна Web-Сторінка або якась її частина, наприклад, окремий елемент HTML розмітки.

Також ресурсом може бути ціла колекція сторінок, наприклад, Web-Сайт. І, нарешті, як ресурс може виступати щось, що не є доступним безпосередньо через Інтернет.

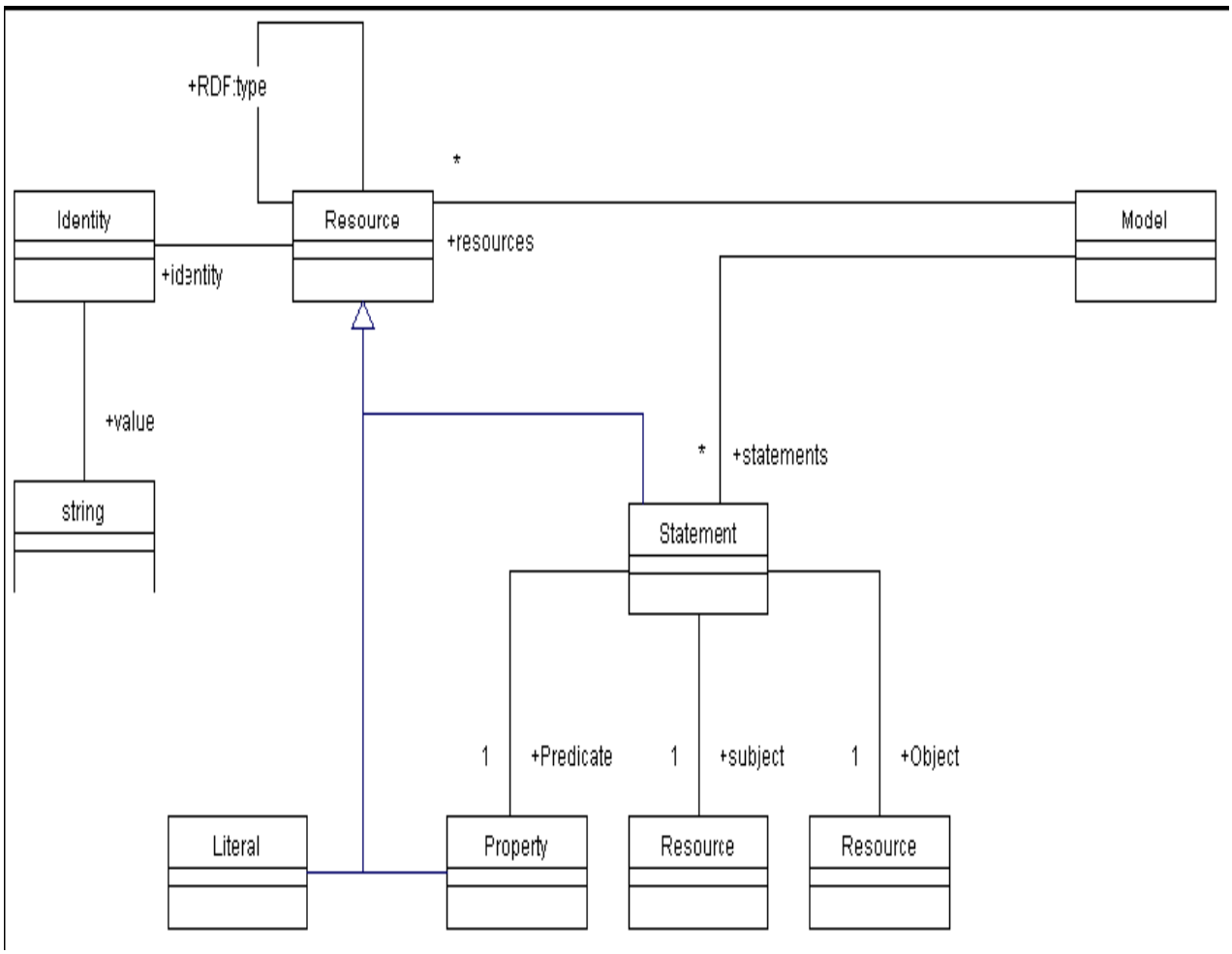

Рисунок 1.1 - Модель RDF

На рисунку 1.1 у термінах відповідних сутностей і зв'язків зображена загальна схема моделі RDF. Тут під властивістю (Property) варто розуміти якийсь аспект, характеристику, атрибут або відношення, що використовується для опису ресурсу. Кожна властивість має свій специфічний зміст, припустимі значення, тип ресурсів, до яких воно може бути застосовано, а також відносини з іншими властивостями. Для забезпечення унікальності імен властивості дотримуються концепції URI,

тобто властивість стає потенційним об'єктом для опису за допомогою RDF окремо від ресурсу, що характеризується й наявного значення.

Таким чином, кожна властивість в RDF саме є ресурсом і може мати свої власні атрибути. Цей факт перетворює модель даних з дерева, яким є XML-розмітка, в орієнтований граф. Вершинами цього графа є суб'єкти й об'єкти, а дугами - іменовані властивості. Оскільки властивість у свою чергу може бути суб'єктом деякого твердження, графи можуть бути як лінійними, так і вкладеними, наприклад, ми можемо виражати сумнів або згоду з яким-небудь твердженням або вказувати джерело одержання відомостей [13].

Одним із загальноприйнятих властивостей є «type», що ставиться до простору імен, що задається безпосередньо специфікацією RDF. Воно дозволяє вказати клас описуваного ресурсу. Це може бути автомобіль, людина, книга, а може бути деяка послідовність об'єктів (для вираження даного факту існує спеціальне значення «Seq», що також належить до простору імен RDF). Відповідно до специфікації [14], значення властивості може мати один із двох типів. Перший - це ресурс, що задається деяким URI. Другий тип - літерал - є деяке текстове значення характеристики. Втім, літерал може представляти собою значення будь-якого примітивного типу даних, присутнього в XML. Його текст також може містити в собі якусь розмітку, наприклад, XML, але відмінною рисою такої розмітки є те, що вона не обробляється RDF-процесором і сприймається як звичайний рядок.

Використання словників RDF Schema.

Модель даних сама по собі містить всього лише основу. Для того щоб опис знайшов якийсь зміст, необхідно скористатися словниками, які задаються за допомогою додаткової технології - RDF Schema, що відіграє для RDF таку ж роль, що й схема для XML.

Під словником ми будемо розуміти сукупність ресурсів, що використовуються для опису властивостей інших ресурсів; класів ресурсів, які можуть бути описані за допомогою заданих властивостей; і обмеження, що накладаються на їхні значення або набори припустимих значень. При

цьому класи можуть складатися у відношенні «підклас» і аналогічно властивості можуть бути зв'язані відношенням «подвластивість» [16].

Модель даних, побудована при використанні належних словників, пропонує осмислений опис ресурсів, але цього ще не досить для розуміння Web машинами. Подібно тому, як одна людина не має можливості передати знання іншому, якщо вони обоє вміють говорити на одній мові, але використовують для цього різну лексику, ціль не буде досягнута, поки не будуть розроблені єдині словники для опису якихось фактів, і програми не зможуть користуватися ними.

Реальне значення RDF неможливо оцінити, поки він використовується для внутрішніх цілей окремо взятого додатка. Користь від впровадження RDF буде тоді, коли він стане засобом між програмної взаємодії, обміну даними, коли машини одержать здатність комбінувати інформацію, отриману з різних джерел, тим самим, одержуючи якусь нову інформацію.

Чим більше додатків в Інтернеті зможуть працювати з даними, тим вище стане їхня цінність [15]. У той же час RDF прекрасно підходить і для подання самих даних, їхньої структури й зв'язків. Таким чином, при застосуванні спеціально розроблених RDF-Схем (як засіб опису онтології предметної області) технологія Semantic Web може бути використана для опису інформації, що ставиться до деяких розділів знань, зрозумілим для різних додатків Інтернету.

1.3. Онтологія й мова OWL

Онтології, в загальному вигляді визначаються як спільно використовувані формальні концепції конкретних предметних областей, вони дають загальне уявлення про поняття, інформацією з яких можуть обмінюватися люди і додатки. Вони дозволяють концептуалізувати домен фіксуванням сутностей (entities) і зв'язків в домені. Вказівка в яких зв'язках приймає участь сутність частково дозволяє зрозуміти і її значення (зміст),

оскільки це надає можливість бачити, де дана сутність входить у відносини з іншим доменом.

Відповідно до принципів Semantic Web, процес створення електронних документів розбивається на дві частини: створення власне документа, що містить деякі терміни, і створення його онтології. Онтологія може описуватися різними засобами й сьогодні існує кілька мов опису онтології, однак через те, що в будь-якій онтології визначаються терміни й задаються логічні зв'язки між ними, точна семантика описуваних термінів і зв'язків у різних мовах буде та сама.

Найбільш удалою мовою є Ontology Web Language (OWL), що описує онтологію у форматі XML. Онтологія OWL є послідовністю аксіом і фактів з додаванням посилань на інші онтології, які вважаються включеними в онтологію. Онтології OWL є Web-документами і на них можна посилатися. Онтології також мають не пов'язану з логікою компоненту (поки ще не визначену), що може бути використано для запису авторства, і інша не пов'язана з логікою інформація, асоційована з онтологією. OWL додає до RDF і RDFS ще більше можливостей для того, щоб описати властивості й класи: зокрема, відносини між класами, кардинальність, рівність, більше типів властивостей, характеристик властивостей, і перелічені класы, OWL представляє три підмножини, що мають різний ступінь деталізації [17].

OWL Lite призначений для користувачів або додатків, яким необхідна лише класифікаційна ієрархія сутностей і деякі прості умови погодженості сутностей.

OWL DL (Description Logic) розрахований на користувачів, яким необхідна максимальний ступінь виразних можливостей мови без втрати обчислювальної повноти (без втрати жодного із семантичних втілень - змістовних тлумачень виводів, отриманих формально-логічним шляхом) і можливості розв'язання (обчислення будуть закінчені за кінцевий час). Рівень OWL DL орієнтований на існуючі сьогодні системи опису знань і системи логічного програмування.

OWL Full розрахований на користувачів, яким необхідні максимально виразні можливості мови й воля вибору кінцевого формату RDF, але без яких-небудь гарантій обчислювальної повноти й можливості розв'язання. OWL Full дозволяє розширити зміст терміна, узятого з якого-небудь заданого словника, і додати його в онтологію.

Формальна семантика OWL описує, як одержати логічні висновки на основі онтологій, тобто одержати факти, які не представлені буквально, а випливають із семантики онтології. Ці висновки можуть базуватися на аналізі одного документа або множині документів, розподілених в Web. Останнє забезпечується можливістю онтологій бути зв'язаними, включаючи прямий імпорт інформації з інших онтологій. Щоб написати онтологію, що може однозначно інтерпретуватися й використовуватися програмними агентами, задіються синтаксис і формальна семантика OWL.

На практиці створення онтологій починається з ієрархії класів понять, що становлять предметну область [8]. Для того, що б поняття предметної області були наповнені певним суттєвим змістом, вони повинні характеризуватися конкретними наборами властивостей і складатися в певних зв'язках один з одним. Це завдання в мові OWL вирішують механізми властивостей і асоційованих з ними обмежень. Властивості поділяються на два види: властивості-характеристики й властивості-зв'язки.

Перші характеризують об'єкти (класи) і приймають як свої значення дані певних типів. Другі асоціюють об'єкти (класи) один з одним і відповідно приймають як свої значення об'єкти (класи).

На властивості накладаються обмеження двох типів: глобальні й локальні. До глобальних обмежень ставляться домени (класи, об'єкти яких можуть мати ці властивості) і діапазони (класи, об'єкти яких можуть виступати як значення цих властивостей). Локальні обмеження накладаються на властивості в рамках певного класу й можуть ще більше звужувати діапазони для властивостей у рамках цього класу, визначати потужність властивостей і їхні види.

1.4. Topic Maps як альтернатива RDF

Технології Topic Maps і RDF виникли досить давно й до деякого часу розвивалися незалежно один від одного. Вони підтримувалися, з одного боку, асоціацією ISO, а з іншого боку - консорціумом W3C. Незважаючи на те, що ці семантичні технології дуже схожі, вони споконвічно були призначені для різних цілей. Технології Topic Maps створювалися з метою забезпечення найбільш якісного пошуку по вмісту Web (зокрема індексації), але як ми вже з'ясували, методологія RDF надає, насамперед, можливість структурного опису ресурсів за допомогою метаданих і логічних зв'язків [11].

Синтаксис Topic Maps описується за допомогою мови XTM (XML Topic Maps), що дозволяє адаптувати стандарт topic map (ISO 13250) для спільного використання з XML і іншими стандартами Web [10]. Мова XTM надає синтаксис опису topic maps, заснований на XML.

Для того щоб більш наочно побачити розходження розглянутих технологій, розглянемо модель Topic Map, зображену на рис. 1.2. Відповідно до діаграми, зображеної на цьому рисунку, модель Topic Map описується за допомогою класів, властивостей і зв'язків між ними. Елемент TopicMap - основний у моделі, як властивості йому властиві набори тим і асоціацій. Елемент тема (Topic) є вираженням деякого поняття, ідеї або будь-якої іншої сутності.

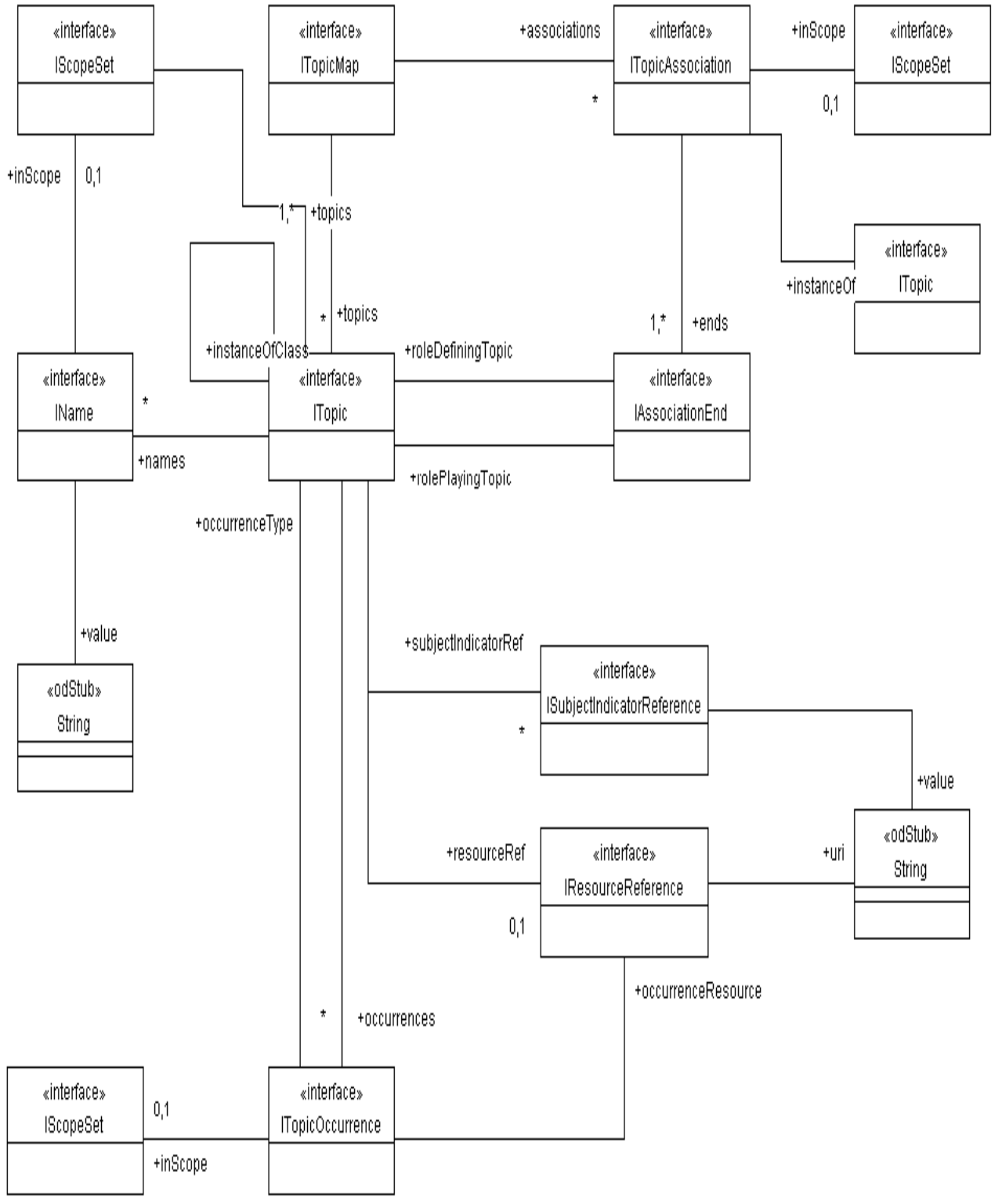

Рисунок 1.2- Модель Topic

Відповідно до синтаксису XTM, тема має такі властивості, як набір імен і входжень (occurrences), а також може брати участь у зв'язках з іншими темами [18]. Зв'язок між темами виражаються за допомогою елементів TopicAssociation, які задаються наборами тими областями - контекстами, у

яких діють зв'язку між ними. Елемент TopicOccurrence зв'язує ресурси з темами на основі визначення відповідних властивостей, при цьому посилання на ресурси задаються в такий же спосіб, як і в моделі RDF - за допомогою URI.

Відомо, що існують способи моделювання сутностей, що описуються за допомогою RDF на основі Topic Maps і навпаки [12]. Таким чином, існує можливість прямого відображення однієї моделі в іншу - вона заснована на зіставленні тверджень RDF зв'язкам Topic Maps. Тому що RDFS виражається через RDF, то можна стверджувати про можливість зіставлення словників RDFS з Topic Maps.

Аналогом для RDFS у технології Topic Maps є Ontopia Schema Language (OSL). На відміну від RDFS для мови опису онтологій OWL не існує аналога в технології Topic Maps, тому що OWL розширює можливості RDFS і не виражається через нього. Таким чином, можливість створення довільних онтологій (у значенні Semantic Web) на базі Topic Maps досить протирічна. Незважаючи на це, існують деякі окремі випадки, у яких на практиці вдається здійснити опис конкретної онтології за допомогою Topic Maps [11].

1.5. Особливості SEMANTIC Web в контексті електронної бібліотеки

Електронна бібліотека є складною інформаційною структурою. Саме поняття електронна бібліотека на даний момент конкретно не визначено. Ми зупинимося на такому підході, що електронна бібліотека це інформаційна система, що дозволяє надійно зберігати і ефективно використовувати різноманітні колекції електронних документів які є об’єднанні через мережу електронних текстів, документів, зображень, звуків, наукових даних та програмного забезпечення яке є ядром сьогоднішнього Інтернету, а в майбутньому через організацію доступу до електронних бібліотек буде утворюватися база знань людства. Згідного даного пояснення ми можемо

сказати, що Основним завданням ЕБ є інтеграція інформаційних ресурсів і ефективна навігація в них.

Цей підхід породжує так звану колективну пам’ять. Поняття колективної пам’яті саме для цифрових бібліотек виникло відносно недавно. Проте цей термін набув широкого поширення, зокрема комітет IEEE Technical Committee on Digital Libraries, трактує це поняття як сукупність електронних бібліотек, електронних музеїв, електронних архівів. Зазвичай основна інформація яка передавалася це була текстова інформація, про те на разі, передається інформації інших типів відео, звук, фотографії та ін. [4].

Такий підхід є інноваційним оскільки дає цілісний доступ до інформації будь-яким користувачам будь-де. Розвиток сучасних інформаційних технологій дає можливість реалізувати це. Оскільки сучасний науковий пошук пов’язаний з великою кількістю даних, тому важливо щоб цю інформацію могли використовувати всі науковці крім цього для наукових даних необхідним є прослід жування виникнення цих даних. Тим самим можна гарантувати їх достовірність, водночас зберігати унікальні екземпляри в бібліотеках, музеях та архівах.

Сьогодні в багатьох музеях та бібліотеках зберігається величезна кількість безцінної інформації, проблема в тому, що до неї є поки тільки фізичний доступ. Створення колективної пам’яті позитивно відобразиться на науці, такий підхід повинен стати серйозним поштовхом.

Відомо, що кожна наукова група збирає та поповнює свій інформаційний фонд, причому якщо кожна така група працює в одному напрямі то відповідно і інформаційні фонди будуть одного змісту, але можливо різної структури, що призводить до неефективної роботи, та різного тлумачення наукових понять.

Обмін інформацією дасть можливість аналізувати дані спостережень одночасно багатьом науковим групам навіть якщо вони будуть знаходитися на дуже великих відстанях. Таким чином утворюється спільний робочий простір, що дозволяє всім взаємовигідно працювати над проблемо.

Колективна пам’ять утворює так звані портали знань та контенту, що представляє собою мережу з розподіленими ресурсами.

Розвиток колективної пам’яті одночасно потребує розвитку інших напрямів.

•Зберігання. Система колективної пам’яті повинна та здатна зберігати великі об’єми інформації різнорідних форматів.

•Інтерфейс користувача. Один з найважливіших компонентів колективної пам’яті, який повинен представляти велику кількість сервісів, для взаємодії між користувачем та інформацією яку він шукає.

•Класифікація та індексація. Дає змогу групувати об’єкти. Однак виявлено, що на це сильно впливає індивідуальне сприйняття та великий обсяг інформації яку необхідно індексувати.

•Інформаційний пошук. В цій області існує багато методів пошуку, включаючи пошук мета даних та контенту. Визначити корисність результату пошуку може тільки сам користувач. Для покращення ефективності використовують додаткові метадані, які описують документ. Дослідники також зосереджуються на автоматизації створення і обслуговування параметрів користувача для використання їх в процесі пошуку.

•Адміністрування та збереження. Традиційні бібліотеки зберігають копію книги, музеї зберігають фізичний експонат. Система колективної пам’яті дозволяє зберігати декілька версій документа. Окрім того цифрова бібліотека може розмежовувати права доступу до авторських екземплярів тим самим зберігаючи авторське право. І всі перегляди будуть автоматично фіксуватися. Механізм захисту повинен бути надійним для виключення несанкціонованого доступу. Зміни технологій організації структури середовища зберігання інформації, доступу до неї та старіння засобів збереження становить серйозну проблему яка повинна також вирішуватися. Окрім розглянутих вище напрямків необхідно також збільшувати степінь деталізації.

Створення нових схем мета даних, зосередження на інформаційному

вмісту а не на інформаційних об’єктах [5]. Отже оперування такими обсягами інформації породжує певні проблеми. З погляду на вище сказане ці проблеми можна розділити на дві великі групи, перша група проблем пов’язана з технічними труднощами організації збереження інформації та доступу до неї, друга група проблем пов’язана з логічною організацію колективної пам’яті та забезпечення доступу до неї з подальшим аналізом змісту. Ця робота стосується вирішення проблем 2-гої групи.

Використання семантичних технологій в електронних бібліотеках було приділено увагу в багатьох Європейських проектах. В проекті SWHi [6] онтологія розроблена на основі електронної бібліотеки з точки зору, коли наші основні джерела даних в репозиторії описані метаданими. Це метадані відображається і зберігається в онтології, яка базується на онтології схеми. Крім того, буквені значення в метаданих, наприклад, заголовок, піддається аналізу в наслідок якого видобуваються імена сутностей, події та термінологія.

Для збагачення онтології, також видобувають нову зв’язану інформацію з обраних веб-документів. Пошук в цій системі реалізований у двох формах, простий та складний. Система використовує мову запитів RDF таку як SeRQL. Процесор генерування запитів SeRQL стикається, принаймні з двома проблеми. По-перше, він не знає, в якому класі чи властивості можуть бути знайдені слова. Щоб уникнути цю проблему, прикладне програмне забезпечення Semantic Web, таке як OpenAcademia [13] вимагає від користувачів вводити ключові слова у відповідне поле (автор, назва або рік) в її розширений пошуковий інтерфейс. По-друге, існують деякі обмеження в підстроках відповідності SeRQL при використанні символу загальності. Цю проблему можна вирішити за допомогою інформаційно-пошукових програм, таких як Lucene яка забезпечує потужний алгоритм, точного і ефективного пошуку. Окрім самого пошуку, важливим також є питання представлення результатів пошуку. Одним із напрямків є візуалізація пошуку. У Semantic Web, візуалізація стає все більш важливою.

Існують випадки складних взаємин між ресурсами, які не можуть бути представлені за допомогою простого списку. Крім того, як правило, відображається тільки невелику кількість результатів пошуку (в діапазоні 10-20 результатів на сторінці). До документів які знаходяться в хвості результату пошуку, швидше за все, ніколи не будуть зверталися.

Загальна архітектура системи SWHi показана на рис. 1.3.

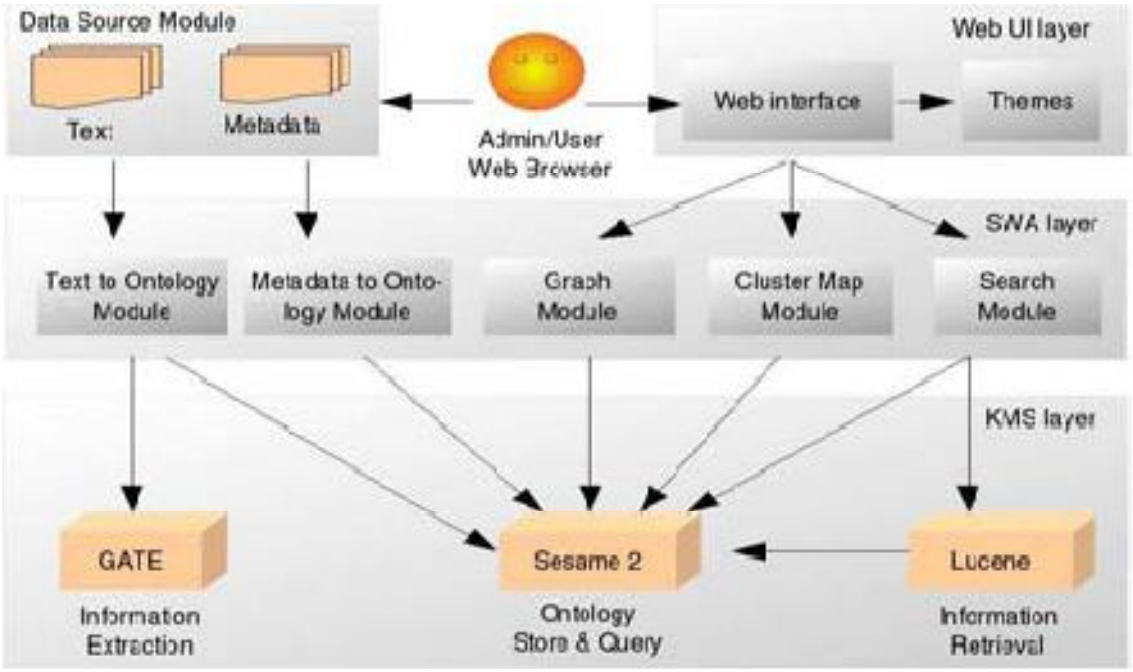

Рисунок 1.3 – Архітектура SWHi

Для розвитку SWHi онтології, повторно використовуються наявні онтологічні ресурси для структурування і збереження історичної інформації, а саме: PROTON базова онтологія, таксономія предметної класифікації NewsBank/Readex, Дублінське Ядро та та словник FOAF Vocabulary. Це в онтології зберігається з використанням Sesame2.

Проект eCulture є семантична пошукова система, яка дозволяє одночасно шукати в кількох колекціях установ культурної спадщини [7]. Це робиться шляхом перенесення цих колекцій в RDF шляхом зв’язування об’єктів колекцій як екземплярів класів через загальнодоступні словники, тим самим створюючи великий RDF граф.

Потім, в ході пошуку, цей граф трасується і деякі підграфи повертаються у вигляді результату.

Основним механізмом пошуку є використання Prolog. Більшість Semantic Web додатків реалізовано за зразком реляційних баз даних, де прикладна логіка має доступу до бази даних на основі SQL [8]. Існує ряд еквівалентів SQL для Semantic Web, таких як SeRQL [9] і рекомендація W3C SPARQL [17]. Обидва дозволяють виразити граф, що складається з низки обов'язкових і необов'язкових ребер та вузлів, з розширеними умовами літеральних значень. SeRQL відповідає вираженню графу на транзитивно закритому виразі використовуючи семантику RDFS.

Стандарт SPARQL не визначає, чи виконується логічний наслідок, який виконаний механізмом судження СУБД. Однак додатки eCulture, не використовують SeRQL або SPARQL. Замість цього, запити прикладної логіки виражаються як Prolog цілі на необроблених даних RDF та/або модулях судження RDFS/OWL. Проект IPISAR (Image Preservation, Information Systems, Access and Research) досліджує розповсюдження, вивчення і раціональне використання культурної спадщини, та спроби представити вирішення загальних проблем в цих областях в рамках Semantic Web (SW) [11].

У рамках магістерської роботи розроблений додаток "Pescador", який буде зберігати каталогізовані дані в трійках, які зберігатимуться в сховищах (чиї функції будуть такі ж, що в реляційної бази даних в традиційних системах). Додаток "Pescador", як показала практика, необхідно в подальшому вдосконалювати, зокрема забезпечення більш гнучких механізмів з подолання обмежень мови DSL, що створена в рамках IPISAR.

Для досягнення цієї мети ми пропонуємо семантичну компоненту архітектуру (SCA), тобто, адаптація компонентів архітектури відповідно до принципів SW, в якому дані, структура та правила прикладної логіки, тісно пов'язані між собою. SCA повинен координувати "компоненти", що підключаються, які б були б обгорнуті оболонкою яка б могла взаємодіяти з

наступними типами: схемами; обмеженнями; правилами виводу; онтологіями; визначення шляхів; програмних кодом; специфікацією виводу; інформацією про конфігурацію Abox; посиланнями до зовнішніх джерел даних.

Алгоритм вилучення інформації з графів часто потребує визначення шляхів між ресурсами. Ці шляхи в значній мірі відрізняються по довжині і складності, тому SCA повинна включати засоби визначення моделі шляху. Попередній огляд існуючих механізмів визначення шляхів показує, що SPARQLeR розширення SPARQL може бути кращим кандидатом для адаптації для SCA і Пескадор.

Проект EPOCH та AMA [13] представляють ЕБ як великий індекс, покликаний служити в якості довідника для пошуку і повернення цифрової інформації яка зберігається на веб-сайті в різних форматах і архівах.

Перша проблема полягає в тому, що кожний довідник має свою пошукову систему і використовує свою граматику метаданих для опису та індексації даних, зокрема, що вона ніколи не буде працювати на інших системах. Жодна з цих систем метаданих може проаналізувати всю інформацію на веб-сайті, якщо ми не будемо робити їх доступними через машину зрозумілій формі з використанням RDF.

Друга проблема стосується безпосередньо інформації: величезна кількість різноманітних форматів, що використовуються для індексування даних, є великою перешкодою на шляху до інтеграції, і повинні бути серйозно проаналізовані. Навіть якщо ми обмежуємо наші зусилля виключно для культурної спадщини, архіви (наприклад, бази даних музеїв і колекцій, археологічні розкопки звіти, доповіді та інші неструктуровані дані), ми змушені визнати, що інформація, також є гетерогенною.

Щоб створити єдиний концептуальний шар, семантична інформація повинна бути взята з бази даних, HTML-сторінок, описових текстів, метаданих і повинна буди представлена в стандартному форматі, з метою отримання концептуального змісту інформації створивши концептуальний

мапінг.

Як тільки концептуальний шар для даних і метаданих готовий, семантична інформація буде зберігатися в контейнері засновані на RDF і онтології.

Це реальним місцем інтеграції, де об'єднані основні відомості з різних електронних архів можуть бути переглянуті та в яких можливо здійснити пошук як в єдиній цілій електронній бібліотеці. RDF мова є надійною і достатньо гнучкою, щоб забезпечити сумісність і забезпечити загальну основу не тільки для цифрових бібліотек, але і з інших систем та послуг.

Мапінг є одним із самих важливих кроків при інтеграції даних. Для спрощення та доступності процесу мапінгу в проекті AMA було розроблене програмне забезпечення AMA Mapping Tool гнучкий інструмент який сприяє мапінгу різних археологічних та музейних колекції моделей даних (з різною структурою, а також неструктуровані дані, тобто текстовий опис) на загальний стандарт ґрунтується на CIDOC CRM- онтології.

Аналізуючи ці проекти можна стверджувати, що існують ряд проблем при створенні електронних бібліотек та їх інтеграції з використанням семантичних технологій. Наприклад така проблематика як створення підходів до семантичної анотації електронних об'єктів. Особливо це стосується семантичної анотації контенту електронних бібліотек у випадку, якщо контент представлений у різних форматах та у різних галузях знань людства. Одним із способів вирішення цієї проблеми може бути узагальнена формальна модель анотації. Іншою проблемою яка постає при оперуванні великої кількості гетерогенної інформації це забезпечення відповідних сервісів. Оскільки сервіси є специфічними для різних форматів документів та повинні враховувати особливості вимог користувачів для обробки цієї інформації.

Висновки до розділу I :

1. Охарактеризовано поняття Semantic Web.
2. Досліджено підходи і технології створення Semantic Web.
3. Описана еволюція розвитку SW
4. Розглянуто проблеми SW, як їх вирішують і які ще залишилися.
5. Досліджено алгоритм вилучення інформації

РОЗДІЛ II
МЕТОДИ ТА ТЕХНОЛОГІЇ ПОБУДОВИ СЕМАНТИЧНИХ ПОРТАЛІВ

2.1. Дослідження порталів знань і семантичних порталів

Поняття Web-Порталу зародилося в надрах мережі Інтернет і було спрямовано на створення зручного входу в «Всесвітню павутину». Майже відразу, в 1998 р., були початі спроби поширити плідні ідеї й технології Інтернет порталу на ґрунт корпоративних інформаційних систем. Веб-портали є точками входу для представлення інформації та обміну через Інтернет, використовуваної в спільних інтересах. Тому вони вимагають ефективної підтримки для спілкування і обміну інформацією. Сучасні веб-технології, використовувані для створення цих порталів створюють серйозні обмеження щодо об'єктів для пошуку, доступу, вилучення, інтерпретації та обробки для інформації.

Ці обмеження природно успадковується існуючих порталів, що перешкоджає комунікації і процес обміну інформацією між членами спільноти нформацією.
Застосування Semantic Web технологій має потенціал подолання цих обмежень і, отже, вони можуть бути використані для еволюції поточних веб-порталів для семантично розширених веб-порталів. Ця стаття являє стан додатки мистецтва семантичних веб-технологій в веб-порталів і поліпшень, досягнутих за рахунок використання таких технологій. Широка оцінка розділених схем і докладні критерії оцінки, розроблені для оцінки та порівняння існуючих Semantic Web-порталів.

Далі представлеийі підхід, що використовується в даній роботі для опису і оцінки Semantic Web-порталів. Наведемо схему оцінки, яка дозволяє провести загальний аналіз SW порталу. У контексті дослідження SW портал розуміється як веб-сайт, який надає інформацію і дозволяє обмінюватися

умовами для спільності інтересів, заснованої на використанні технологій Semantic Web. На рис. 2.1. показана схема, яка використовується для опису і SW оцінки порталів. В основному виділяються три шари: доступ до інформації - з точки зору користувача, обробка інформації - особливості порталу та заземлюючі технології.

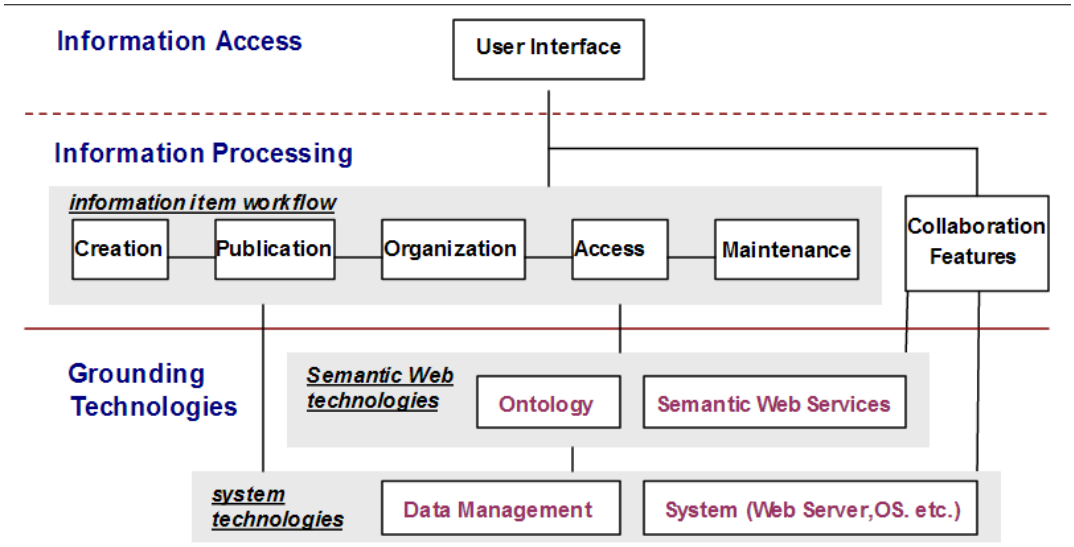

Рисунок 2.1 – шари Семантік веб поталів

Шар доступу до інформації включає в себе функції для користувача - системи - взаємодії, які об'єднані в оцінці юзабіліті, призначеного для користувача інтерфейсу і в оцінці порталу в якості веб - додатку. Шар обробки інформації охоплює можливості обробки інформації елементів із зображенням SW порталу, диференційовані процеси інформації елемента, зображені на рисунку 2.1. Оцінка особливостей обробки і методів реалізації перевіряються на кожному етапі. Крім того, функції для спільної роботи завжди включені. Вони розширюють обмін інформацією та комунікаційні засоби обслуговування SW порталу з точки зору користувацької взаємодії. Самий нижній шар схеми замикає технології, що дозволяють функціям(фючам) на верхніх рівнях звертатися до нижнього і того він називається шаром заземлення технологій.

Існують наступні типи порталів: інформаційні портали, портали для

спільної роботи, портали експертизи, портали знань (типи порталів перераховані в порядку зростання їхньої складності й функціональності) [1].

- Інформаційний портал зв'язує людей з інформацією.
- Портал для спільної роботи підтримує різні засоби взаємодії людей, засновані на комп'ютерних технологіях.
- Портал експертизи зв'язує людей один з одним на підставі їхнього досвіду, області експертизи й інтересів.
- Портал знань комбінує можливості перерахованих вище типів і забезпечує доставку персоніфікованої інформації з урахуванням конкретної роботи, що виконує кожний користувач у певний момент часу.

Під корпоративним порталом знань звичайно розуміють єдиний засіб доступу до корпоративної інформації, що дозволяє співробітникам взаємодіяти один з одним, зв'язувати інформацію з колективним розумінням, системою цінностей і досвідом. Наступні властивості порталу знань є типовими:

- сприяє збору, структуруванню й передачі інформації з різних внутрішніх і зовнішніх джерел і систем;
- дозволяє співробітникам взаємодіяти один з одним;
- забезпечує підтримку командної роботи з інформацією й знаннями;
- зв'язує інформацію з колективним розумінням, системою цінностей і досвідом, сприяє появі нових знань у процесі взаємодії людей, а також задоволеності співробітників від роботи в компанії.

При цьому портал знань повинен мати всі властивості звичайного порталу, такими як персонізація для кінцевих користувачів, організація клієнтського місця, розподіл ресурсів, відстеження виконання робіт, доступ до інформації з безлічі гетерогенних джерел, локалізація й виявлення потрібних людей. На даний момент існує безліч готових рішень таких компаній, як IBM, Microsoft і Oracle, здатних забезпечити побудову корпоративного порталу, що задовольняє цим властивостям. При цьому дуже

мала частина сучасних портальних рішень може бути використана в якості базової для побудови порталів знань, у таких випадках на допомогу приходять описані вище семантичні технології, зокрема, онтології й мову опису їхнього опису OWL.

Можливості застосування онтологій полягають у наступному.

На сьогоднішній день у рамках семантичних технологій найбільше активно досліджується й розвивається онтологічний підхід до подання знань предметної області, на підставі якого розробляються інтелектуальні інформаційні системи, і в тому числі портали.

Онтологія - модель подання знань якої-небудь предметної області у вигляді набору понять цієї предметної області й існуючих між ними відносин. Тобто онтологія представляє предметну область у вигляді деякої мережної структури, у якій семантика кожного поняття визначається через його відносини з іншими поняттями.

Онтологічна модель предметної області задає семантику понять, які використовуються для опису інформаційних об'єктів порталу. Такі описи називаються семантичними метаданими, які дозволяють:

- усунути лексичну багатозначність термінів, використовуваних для опису інформаційних об'єктів;
- визначати відповідність між різними інформаційними об'єктами, використовуючи онтологію.

Семантичні метадані можуть описувати об'єкт із різних точок зору: структури, контексту й змісту (контенту). Опис контенту найбільше важливо для тих інформаційних систем, які реалізують функції повнотекстової обробки інформації.

Для забезпечення семантичного узагальнення розділів інформації в корпоративних порталах необхідно визначити онтологію, що описує термінологію, зміст порталу, а також аксіоми, що задають правила використання цих термінів у контексті інших термінів. Ці аксіоми можуть описувати нові терміни, користуючись уже певними примітивними

термінами з поточної або якої-небудь іншої онтології [17].

Даний підхід дозволяє на основі наявних фактів із предметної області із цих визначень і аксіом виводити нові факти, які, у свою чергу, можуть бути отримані користувачем при такому інтелектуальному пошуку по порталі, але не могли бути отримані раніше при звичайному пошуку. Приведемо класичний приклад: якщо зареєстрований користувач порталу шукає деяку інформацію про комп'ютерні програми, і в його реєстраційних даних зазначено, що людина займається комп'ютерами, то при добре спроектованій онтології в процесі пошуку більший пріоритет будуть мати документи, що містять інформацію про комп'ютерні програми, а не про програми загалом.

Зрозуміло, ефективність запропонованої технології буде залежати від авторів, що публікують інформацію, тому що буде необхідно анотувати ресурси метаінформацією, записаної мовою онтології. У наступних двох розділах будуть показані приклади інших підходів до рішення даної проблеми.

Онтології досить корисні при систематизації даних на корпоративних порталах для індексації й зручного пошуку, незважаючи на те, що багато великих організацій мають власну таксономію для організації внутрішньої інформації, цього звичайно недостатньо. Проста класифікація сильно обмежує можливості пошуку й індексації, оскільки багато документів можуть підпадати під різні категорії, тому пошук за різними критеріями буде набагато ефективніше, ніж звичайний пошук по ключових словах. Крім того, типова проблема користувачів корпоративного порталу складається в неможливості користування єдиною термінологією.

Наприклад, фахівець із комп'ютерного відділу використовує для опису якої-небудь проблеми свій професійний жаргон, а фахівець із відділу продажів погано розуміє отриманий документ через те, що звик користуватися іншою термінологією. Зрозуміло, ця проблема вирішується іншими засобами, але майже завжди створює додатковий труднощі. У випадку наявності онтології досить створити кілька термінів для різних типів

користувачів, але з посиланнями на відповідні терміни з інших онтологій, щоб одні терміни могли транслюватися в інші автоматично.

На сучасний момент існує дуже невелика кількість технологій побудови семантичних порталів і, в основному, вони мають застосування лише в якій-небудь специфічній предметній області. З відомих робіт хотілося б виділити досвід проектування корпоративного порталу групи компаній «ИТЕРА» [2]. У даній роботі застосовується методологія проектування корпоративних систем на основі положень теорій послідовностей, категорій, обчислень і семантичних мереж. Однак, як такі семантичні технології в ній застосовуються не в чистому вигляді, зокрема, всі моделі даної системи описуються на основі XML, а не RDF або XTM.

2.2. Технології побудови семантичних порталів SEAL

У якості одного з найбільш вдалих прикладів апробації семантичних технологій при побудові порталів можна розглянути підхід SEAL (SEmantic portAL). Основна ідея SEAL складається в побудові Intranet додатків, здатних надати інформацію для користувачів і агентів з обліком їхньої семантичної структури [6]. Прикладом реалізації підходу SEAL є Intranet портал інституту AIFB (University of Karlsruhe).

Загальна архітектура й оточення цієї системи зображені на рис. 2.2.

Дані системи зберігаються в спеціалізованому сховищі знань (Knowledge warehouse), у якому реляційна база знань будується з урахуванням онтології. Основний механізм взаємодії зі сховищем здійснює система Ontobroker, що також забезпечує можливість описувати онтології, правила й факти компіляторам різних мов. Зовнішня частина системи доступна трьом типам агентів, які спілкуються із системою через Web-Сервер і відповідають трьом основним типам взаємодій.

Програмні агенти (software agents) можуть обробляти інформацію через Інтернет, для цього модуль RDF Generator надає факти з бази знань у форматі

RDF.

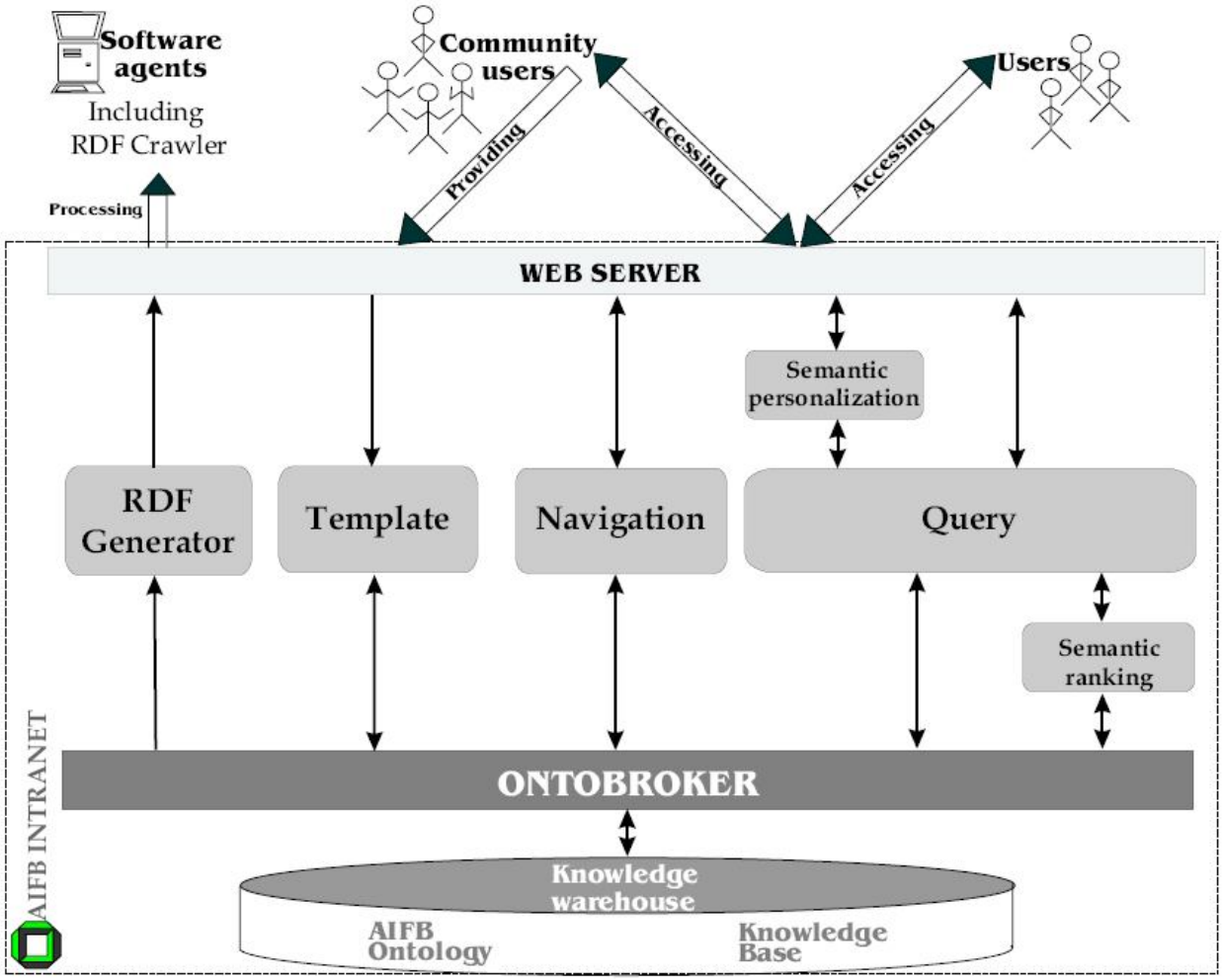

Рисунок 2.2 - Архітектура системи AIFB Intranet і її оточення

Звичайні користувачі (general users) мають можливість робити доступ до порталу двома способами: роблячи переходи по гіперпосиланнях, що зберігається в документах і роблячи пошук по запиті. Структура гіперпосилань надається укладачам сторінок порталу, але також може доповнюватися за допомогою модуля навігації (navigation module).

Модуль навігації використовує спеціалізовану машину логічного виводу посилань, що здатна генерувати структуру семантичних (зв'язаних за змістом) посилань. Крім того, користувач може особисто настроїти інтерфейс пошуку, використовуючи модуль семантичної персонализации (semantic personalization) а також засобу ранжирування результатів пошуку на основі

семантичної відповідності (semantic ranking).

Члени співтовариства крім доступу до даних системи можуть надавати системі нові дані, наприклад, інформацію про їхні області, що цікавлять, знань, персональні дані, публікації й інші дослідницькі роботи. Кожному типу даних, що поставляються, відповідає хоча б одне поняття онтології. При запиті відповідних частин онтології, модуль шаблонів (template module) автоматично генерує підходящі HTML-Форми для уведення даних. Члени співтовариства заповнюють ці форми, а модуль шаблонів зберігає нові дані в сховище знань.

Існують і інші, більше відомі приклади використання семантичних технологій, такі як Web-Портали Yahoo (www.yahoo.com) або Google (www.google.com).

На жаль, області застосування технологій Semantic Web залишаються досить вузькими, в основному обмежуючись спеціалізованими порталами знань наукових співтовариств і інститутів.

2.3 Формальна модель анотації

Для того, щоб показати переваги семантичного підходу до електронної бібліотеки на відміну від стандартної, необхідно визначити ті структурні елементи, які раніше не мали ніякої семантичної моделі, і ми пропонуємо цю модель, щоб побудувати. Основні два компонента електронна бібліотека контенту і набір програмного забезпечення для роботи з цим змістом.

По-перше, розглянемо зміст ЕБ.

Інформація, що міститься в електронних бібліотеках, описаних в термінах електронних об'єктів (Digital об'єкти - DO), які в більшості випадків складаютьсяз мультимедійного контенту і метаданих [15]. Через це і велику кількість даних DO, бажано зробити пошук і класифікацію за допомогою анотацій DO. Формальна модель, яка запропонована в роботі [16], передбачає

два підходи до розуміння анотації як анотації метаданих або анотації як змісту.

У першому випадку ми маємо справу з різними схемами метаданих (Dublin Core, MARC і т.д.). Вони використовуються для опису інформаційних ресурсів. Ці анотації в першу чергу спрямовані на користувача.

У другому випадку в якості абстрактних уявлень контенту, призначених для автоматизованої обробки машини. Ці анотації забезпечують семантику документа.

Семантична анотація - анотація написана формальною мовою з чітко визначеною семантикою і на основі онтологій. Насправді, ця анотація є формальною моделю DO, з можливістю машинної обробки.

Семантика змісту, в свою чергу, можуть бути засновані на зовнішніх пов'язаних онтологіях, які дозволять побудувати семантичну модель документа, де зв'язок між різними сегментами, визначений DO, і на основі семантичних зв'язків між структурними компонентами DO, які визначаються співвідношенням між логічними закінченими структурними компонентами.

Авторство документа (рис. 2.3) полягає у визначенні відносини між термінами, та зовнішніми отологіями а також полягає в визначенні зв'язків між логічного закінченими елементами з іншими DO. Крім того DO описана за допомогою метаданих.

Модель яка буде представляти цифровий об'єкт повинна відбивати реальний зміст об'єкта. Серед багатьох розвинених моделей, ми використовуємо модель, яка запропонована в [15], [16] з урахуванням змін і уточнень, які враховують використання онтологій. Розташування кожного цифрового об'єкта, а також тези ідентифікується унікальним ідентифікатором - посиланням (Link). Також з'єднує цифрові посилання на об'єкт і анотацію, і може відображати зв'язок між об'єктами.

Таким чином, існує два типи посилань: посилання анотацій (Annotate link) - показує відношення в середині цифрового об'єкта, який може бути документом і анотацію; Співвідношення посилання (Relate-to link) - визначає ставлення зовнішнього цифрового об'єкта до об'єкта анотування.

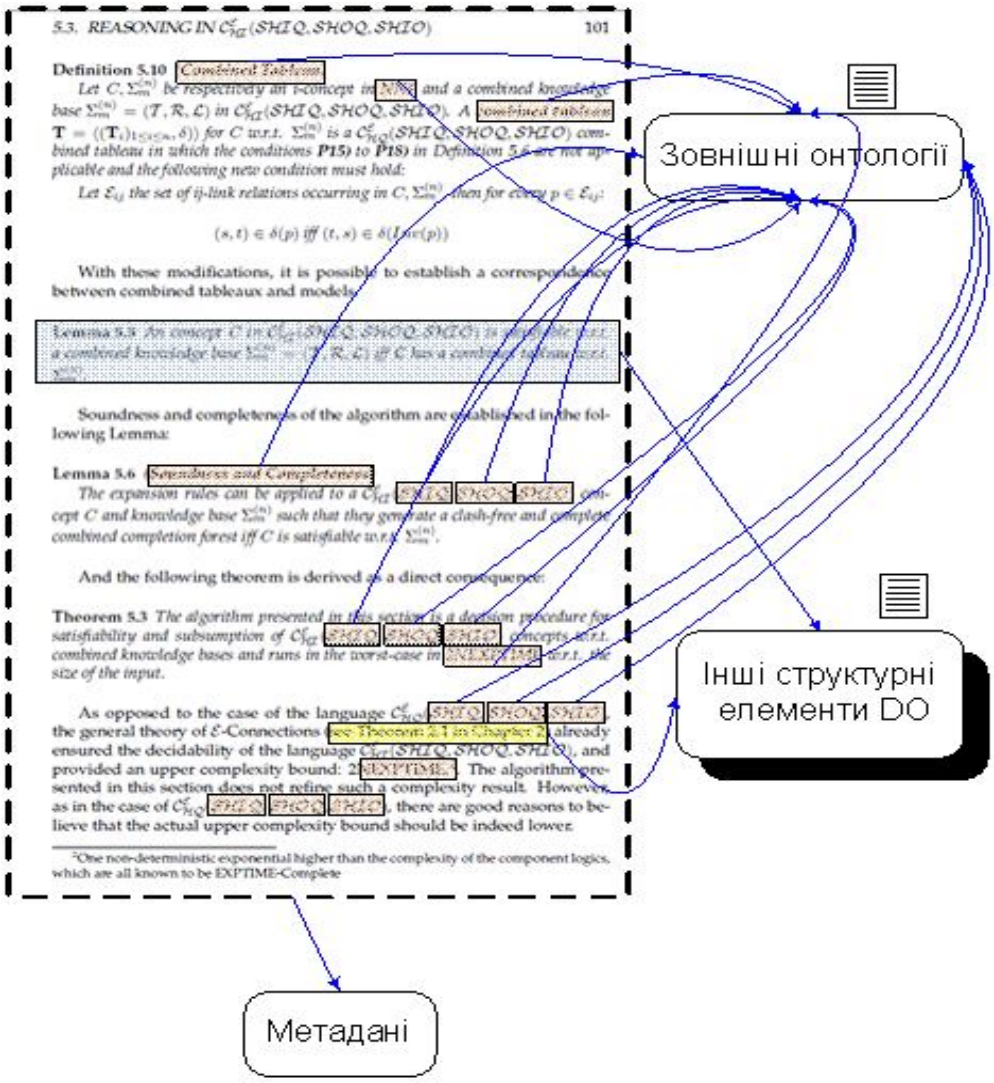

Рисунок 2.3 – Анотування документа

Нехай *LT является* множиною типів посилань, і в свою чергу *LT* складається з наступних типів посилань

LT = {*AnnotateLink*,*Relate - to Link*}.

Посилання задаються у вигляді ідентифікаторів.

Найвідомішими і популярнішими ідентифікаторами можуть виступати URI, OPENURL, DOI, Persistent URL (PURL), PURL-based Object Identifier. *H*

(*k*) - це інтервал ідентифікаторів для цифрових об’єктів у певний проміжок часу *k* .

В даній моделі цифровий об’єкт представлений у вигляді потоку *sm*.

Потік *sm* - це кінцева послідовність, де 1 2 3 (, ,....,) *n e = j j j* – алфавіт символів.

Якщо ми маємо sm потоку: якщо цей потік ми можемо вибрати безперервний сегмент SM st послідовність чисел а, Ь, так що, багато сегментів ми позначимо ST. Так якщо цифровий об'єкт DO має безліч унікальних ідентифікаторів HSM функції H потім відображає унікальний ідентифікатор для сегментів, що містяться в DO. Потік вимагає, щоб не було ніякої функції і властивості сюрєктивності і інєктивності. Кожен цифровий об'єкт може мати щонайменше один потік. Sm утворює безліч потоків, таким чином, тези можна розглядати як процес розширення онтології O.

Розглянемо найпростіший випадок, коли розширення шляху додавання нових екземплярів онтології O. Кожен клас онтології O будемо позначати як kl, набір класів через KL.

Анотація - це кортеж, де *a i h* – має власний унікальний ідентифікатор для анотації *a*, тобто ( ) *a h h = a*.

Aa інтервал з *n*-парних відношень анотації *a* і визначається він як добуток інтервалів *KL*, *SM, LT*, *ST* та *H*. Коли буде проводитися анотування веб-документів, то формальна модель буде змінена.

Нехай *A* буде множиною всіх анотацій *a*, а *D* - множина документів, які відповідають, *DO = D* U *A*, до того підмножина інтервалу *DO* позначимо як *do*, тобто анотацією для веб документів ми будемо називати помічений граф: де *DO = D* U *A є* вершинами графа і сторонами графа.

2.4. Сервісна архітектура БД

Як було сказано раніше, при об’єднанні електронних бібліотек

виникають проблеми з відмінностями в архітектурі ПЗ. Ми вважаємо, що використання сервіс-орієнтованої архітектури є ключовим елементом в досягненні сумісності і інтероперабельності. У цьому середовищі, складне програмне забезпечення може бути розроблене на основі послуг, які доступні і які не доступні на місцевому рівні, і в тому числі через Інтернет. У цьому контексті, для побудови електроних бібліотек ми пропонуємо використовувати сервіс-орієнтовану архітектуру Digital Library (Сервіс-орієнтована архітектура цифрова бібліотека - SODL) [17].

Це забезпечить зручний спосіб для досягнення будівництва колективної пам'яті. Для отримання вільного доступу до бібліотечних послуг, система повина бути реалізована на відкритій архітектурі. Для цього сервіси взаємодії повинні бути описані з використанням стандартних правил і умов, які будуть мати можливість вільно ввести новий пункт обслуговування без повторного перебудування системи. Для забезпечення динамічного настроювання електронної бібліотеки, необхідно створити механізми для виконання цієї функції.

Цей механізм введе семантику в середовище роботи служб Web. Такий підхід називається Semantic Web, і це не тільки семантичний опис послуг, а також семантичного опису інтернет-ресурсів (веб-сторінок, документів і т.д.).

Під даним сервісом ми розуміємо деяку службу, яка прагне задовольнити користувача. Послуги виконуються з використанням веб-сервісів. Веб-сервісом є програмний додаток. Основна відмінність між сервісом і веб-службою і є той же сервіс, який можна одночасно побудувати за допомогою різних веб-сервісів. У свою чергу, веб-служби можуть бути об'єднані один з одним, створюючи новий веб-сервіс, і даний підхід ностить назву композиція веб-сервісів.

Електронні послуги бібліотеки можуть бути класифіковані за різними критеріями. Зокрема, результат виконання запиту на обслуговування запитуючої сторони буде виводитися у вигляді структурованої електронної інформації об'єктів, що містяться в ЕБ. Тому, якщо ми будемо розглядати ЕБ

як замкнуте середовище, то ясно, що ЕБ зміст не було змінено, сервіси з такими властивостями буденосити назву сенсорні сервіси.

Поняття сенсорних послуг для ЕБ ми ввели тому, що ці послуги аналогічні відфільтрованої інформації датчика навколишнього середовища в дійсності, не зачіпаючи зміст навколишнього середовища. Відповідно до Semantic Web Sensor [18] Крім того, ми анотуємо результати послуг. Тому, на відміну від класичного будівельного середовища для семантичних веб-сервісів, де результат не анотується, ми очікуємо, що результами будуть анотації метаданих веб-служб. Характер метаданих повинна включати в себе режим історії, місце і час отримання результатів.

Для того, щоб створити електрону бібліотеку, засновану на сервіс-орієнтованому підході потрібно виділити послуги, за якими композиція буде реалізована [19]. Це відбувається тому, що робота композиції легше вирішується і, крім того, вимоги до електронної бібліотеки така, що в першу чергу необхідно вирішити задачу композиційну. Базові послуги виділяються в [20]. Нехай ми маємо запит *r*, в якому є задані початкові параметри *in r* і жаданні атрибути виходу - *out r*, але необхідно знайти веб-сервіс *w*, який би виконував *r* і при цьому задовільняв умови: *in in r* ⊇ *w out out r* ⊇ *w*.

Завдання знаходження служби, яка може самостійно виконати запит г називається дослідження веб-сервісів (Web Service Discovery- WSD).

При використанні однієї служби не може досягти запиту г, необхідно створити композицію з декількох веб-сервісів {w1, w2, w3, ..., шп} послідовним або паралельним способом, так що все Wi ∈ (w1, w2, i3, ..., шп), причому *I in w* може бути заснований на *I out w* та виконувалось відношення (1 ... *n*) *in out out out r* U *w* U U *w* ⊇ *r*.

Ця проблема називається композицією веб-сервісів (Web Service Composition - WSC).

Але більш складною зараз є проблема композиції веб-сервісів.

Під час роботи сервіс-орієнтованої електроного бібліотечного середовища, в якім веб-сервіси працюють постійно змінюється. Зміна

середовища, породжена двома основними факторами: по-перше проблемами, розподілених систем, генерації тимчасових затримок і ненадійний транспортний протокол, відсутність спільного використання пам'яті між частинами розподіленої системи, проблема відмови в доступі і паралельних запитів і проблем, пов'язаних з програмним забезпеченням несумісності, результатом поновлення розподіленої системи; по-друге, людський фактор в служби користувач може змінити вимоги користувачів, і, отже, якимось чином впливати на результат.

Звичайні користувачі цільової аудиторії не мають уявлення про архітектуру бібліотеки, і, отже, не можуть чітко визначити цілі, які повинні задовольняти сервіси.

Ми будемо розглядати композицію на основі незбираної-орієнтованої парадигми, яка заснована на початкових умовах і доступність послуг робить стверджуванної композиції. І тому, що веб-сервіси знаходяться в семантичному середовищі, ми вибираємо такі плани композиції, які можуть бути корисні для кінцевого користувача. Тобто на відміну від класичної постановки задачі, специфікація прагне знайти послуги, які можуть досягти цієї мети, виходять з того, що існуючий набір послуг може досягти деяких цілей які наперед невизначенні і які можуть бути обрані користувачем.

У такій складній меті системи перед композицією можна варіюватися. Ми також вважаємо, що знання навколишнього світу не є повним, і, отже, цілі є неповними. Для цих цілей HorstRittel конфлікт і Melvin Webber ввели поняття вікід(wicked) задачі.

Оскільки класичні методи планування виявилися неефективними. В умовах, коли цілі не будуть чітко визначені і в них це динамічний процес, план має інший сенс, а саме, план дій не є однозначним, то алгоритм для досягнення мети, планів, функцій може бути сформульований таким чином [21]: перевірка ресурсів ; початок координації процесів або допомага спростити процес координації на початку; встановлення відповідальності і ідентифікації; відслідковування прогресу і що більш важливо генерація

намірів; розпізнавання і управління ризиками; Підтримується 5 ч \ u1110 імпровізація і, найголовніше, формалізація права представництва (користувач) проблеми, з якою план намагається вирішити її. Таке поєднання функціональних властивостей призводить до того, що плани повинні додатково бути вивченні, прийнятті і репланованні.. Ця зміна планів відбувається під час виконання цих планів.

У даній статті наведені теоретичні та практичні основи створення ЕБ з використанням семантичних технологій.

Проте проблематика створення таких ЕБ потребує подальшого вивчення, зокрема в роботі не формалізовано поняття онтології. Водночас також не вказано і типи зв’язків які можуть бути між екземплярами та онтологією. Проте, вже зараз можна стверджувати, що необхідно буде вирішувати проблему вирівнювання між онтологіями, яка виникне внаслідок інтеграції двох або більше семантичних ЕБ.

2.5.Дослідження архітектури і функцій СКВІР

Система колективного використання інформаційних ресурсів (СКВІР) призначена, для автоматизації процесів діяльності, зокрема, обробки, завантаження й публікації вхідного обсягу контенту(інформації) і суть її полягає в тому, що користувачі які знаходяться в різних місцях, мають можливість одночасно працювати з сховищами даних, прикладними процесами, розташованими в взаємопов'язаних кінцевих системах.

Наступні властивості характеризують дані процеси:

- значні надходження документів із зовнішніх організацій;
- перевірка цілісності вхідної інформації;
- розподіл матеріалів по рубриках у процесі публікації;
- робота з обробкою даних або пошуком інформації;
- аналіз даних;

- регламентований доступ користувачів до значного обсягу затребуваної інформації.

Значну частину інформації, яка надходить, становлять об’єми документів, які наділені метаданими і мають внутрішню структуру. У метаданих зберігається не вся інформація, яка є в документі, лише певна частина, а решта зберігається у змісті самого документа. Структури тематичних ієрархій пакетів документів утримуються усередині додаткових Word або HTML документів і не придатні для подальшого ефективного використання.

Для дослідження інформації накопичується й використовується досить велика кількість інформаційних ресурсів, як первинних, отриманих від різних постачальників інформації, так і власних, що є результатом роботи підрозділів. Використовувані дані можна умовно розділити на наступні групи:

- статистичні – в основному це динамічні ряди числових даних за різними показниками;
- повнотекстові – являють собою електронні документи різних форматів;
- метадані – являються структурованим набором даних про дані.

Значні обсяги накопиченої інформації, її розмаїтість і складність, наявність у аналітичної обробки даних із застосуванням різних інформаційних систем, виконання функції інформаційного обслуговування великої кількості співробітників з розмежуванням доступу - ці й інші фактори роблять інформаційні ресурси складною структурою, що вимагає системного підходу для реалізації функції адміністрування інформаційних ресурсів.

У даній сфері досліджень і інформації довгий час не існувало засобів автоматизації процесів обробки вхідних пакетів документів, які змогли б полегшити процедури формування їх метаданих і уможливити генерацію підходящих структурних описів. Появі подібних засобів автоматизації

перешкоджали експлуатовані Web-Технології, не придатні для ефективної роботи з метаданими й тематичними рубрикаторами.

Ефективне використання інформаційних ресурсів було утруднено через безліч недоліків існуючого Web-Сайту, у тому числі проблем, що утрудняють процеси обробки й публікації інформації. Основними із цих проблем були:

- відсутність загальної пошукової системи: повнотекстовий пошук по документах, опублікованим у наявному порталі, був неможливий, що посилається інформація не була проіндексована;
- ручна публікація: документи публікувалися шляхом розміщення файлів на Web-Сервері, що ускладнювало адміністрування інформаційних ресурсів і керування доступом;
- рубрикатори: створені вручну у вигляді HTML-Файлів і не були представлені у вигляді декларативного опису, що ускладнювало рубрикування документів і редагування самих рубрикаторів;
- децентралізоване керування доступом - необхідність знати, хто підтримує конкретний інформаційний ресурс;
- слабка масштабованість і гнучкість - була відсутня можливості швидкого редагування структури відображення інформації;
- неоднозначний інтерфейс доступу до інформації, що заплутувало користувачів при спробах знайти потрібний ресурс.

Програмне рішення було покликане замінити існуючі Web-Сайти і вирішити зазначені вище проблеми. Як базова технологія для розроблювальної системи по міркуваннях, було обране рішення Microsoft SharePoint Portal Server, так як серед своїх аналогів дане рішення вже давно на ринку і воно вважається сильним рішенням, звісно гугл системи знаходяться на іншому рівні. Порівнюючи з аналогом голандської компанії O2Spaces і від IBM, то дане рішення вважається дешевшим, в порівнянні з голандським рішенням і більш розвиненим і гнучким в цих операціях, від IBM “Lotus Quickr” є дуже потужним рішенням, але ціна його використання висока.

Microsoft SharePoint дасть змогу нам ряд наступних можливостей, для реалізації нашого рішення:

- набір веб-засобів для організації одночасної роботи користувачів і системи;
- функціональність для створення веб пропозицій;
- модуль пошуку даних(інформації) в документах, які будуть в нас зберігатися і оброблятися;
- модуль створення форм для заповнення інформації;
- функціональність для проведення бізнес-аналізу

Основне призначення СКВІР - здійснення інформаційного обслуговування, що полягає в наданні інформаційних послуг, таких як формування інформаційних ресурсів на підставі вихідних даних, категоризація й надання доступу користувачам до цих ресурсів. Функціонування системи СКВІР забезпечує наступні види діяльності:

- публікація інформаційних ресурсів у кластер, у тому числі ресурсів, що надійшли з інших організацій;
- використання й пошук інформаційних ресурсів, що надійшли з інших організацій;
- використання, відновлення, обговорення й пошук інформаційних ресурсів;
- пошук фактографічної інформації в рамках вказаних інформаційних систем.

У більшості випадків користувачі системи розміщають свої документи в сховище через Web-Інтерфейс, задаючи при цьому поля метаданих. Але крім публікації робочих документів працівникам необхідно розміщати інформацію (наприклад, форми звітності, добірки економічних статей і доповідей) із зовнішніх організацій. Ця інформація може надходити з різною періодичністю й у різних форматах (у вигляді архівів з файлами певного формату; у вигляді набору зв'язаних HTML-Документів). Процес її

завантаження автоматизує шляхом завдання конфігурацій завантаження - дескрипторів завантаження. Перетворення й публікація цих пакетів документів з полями метаданих виконується відповідними компонентами системи під керівництвом оператора завантаження.

Користувачі системи можуть звертатися до будь-яких інформаційних ресурсів за допомогою пошуку або безпосередньо по місцю розташування ресурсу шляхом навігації за структурою порталу. Пошук здійснюється на основі інформації про назву ресурсу або полів його метаданих, які можуть приймати числові й строкові значення, а також бути датою. Необхідні для роботи користувача документи можна переносити в зручне для розташування місце й рубрикувати відповідно до їх призначення.

При володінні відповідними правами користувач системи засобами Microsoft Office або через веб-інтерфейс системи може створити або змінити документ. Обговорення документів або інших інформаційних об'єктів системи відбувається через веб-інтерфейс на вузлах, у яких розташовані обговорювані об'єкти. Будь-який документ може мати кілька версій у сховище для забезпечення спільного редагування декількома користувачами. Існуючі в сховище інформаційні ресурси можна рубрикувати за допомогою підтримки ієрархії областей і створення на посилань ці ресурси (входжень), а також ведення спеціалізованого тематичного рубрикатора.

Вся робоча інформація структурується відповідно до загальних міркувань по її рубрикуванні в рамках даної системи адміністраторами й самими користувачами системи. Пошук необхідної інформації виробляється у відповідних областях, вузлах або бібліотеках, що містять шукані інформаційні ресурси. Звуження області пошуку можна робити шляхом навігації по структурі інформаційних ресурсів. Так само можливий пошук по всім усьому сховищі, у тому числі з урахуванням визначення додаткових умов на значення полів метаданих.

Автоматизована система СКВІР є досить складною й виконує безліч функцій. Така функціональність забезпечується за рахунок розбивки системи

на відповідні функціональні блоки або підсистеми. Зв'язку між цими підсистемами влаштовані таким чином, щоб надати кінцевому користувачеві повний набір можливостей системи через єдиний інтерфейс. В особливих випадках є необхідність у забезпеченні окремих користувачів доступом до спеціалізованих функціональних модулів системи.

На рисунку 2.5 зображена схема інформаційної моделі СКВІР.

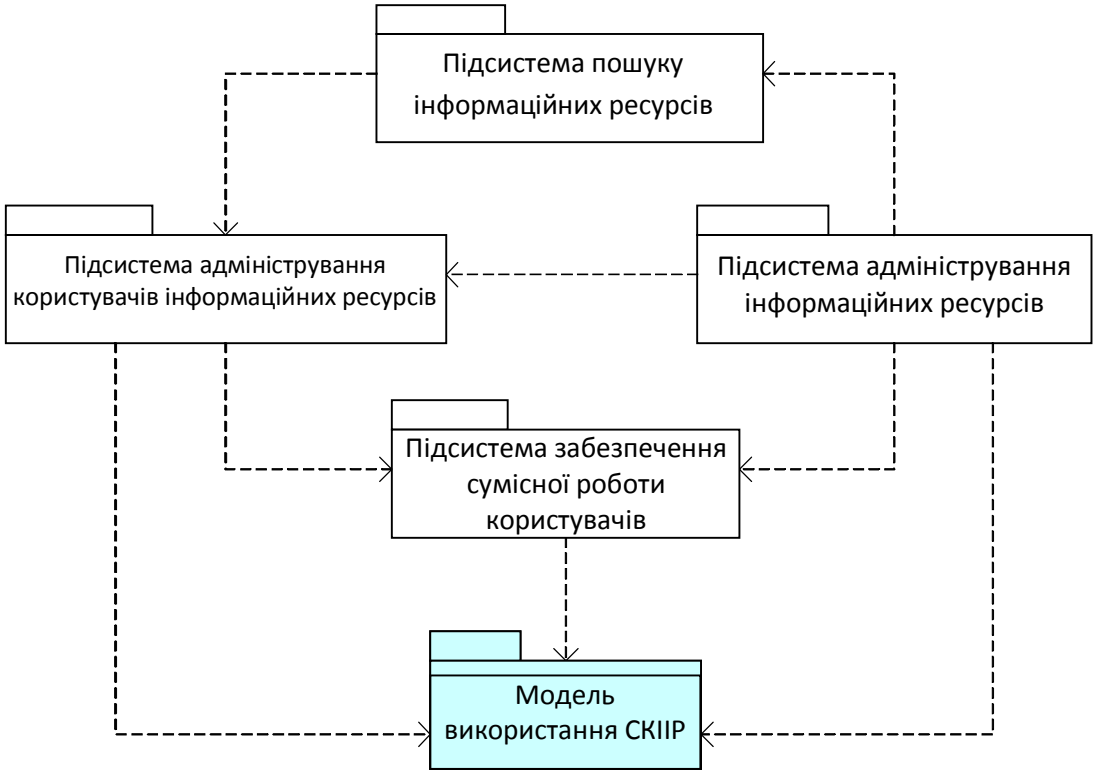

Рисунок 2.5 - Схема інформаційної моделі СКВІР

Перевагою впроваджуваної технології порталу є забезпечення інтегрованого подання різних типів інформації, наявність єдиної пошукової системи, розвинених засобів підтримки колективної роботи. Портал надає можливість пошуку документів по всіх джерелах, або окремим колекціям. Підтримується ранжирування результатів пошуку за різними критеріями: вузлах, авторах, датах, областях.

Технологія порталу надає ряд додаткових елементів користувальницького сервісу: можливість гнучкого керування вмістом порталу, зокрема, створення власних добірок матеріалів, одержання

відомостей про відповідальні за підтримку ресурсу й ін. Послідовне застосування, що розширюється, технології SharePoint спричиняє ріст числа ресурсів, доступних через портал.

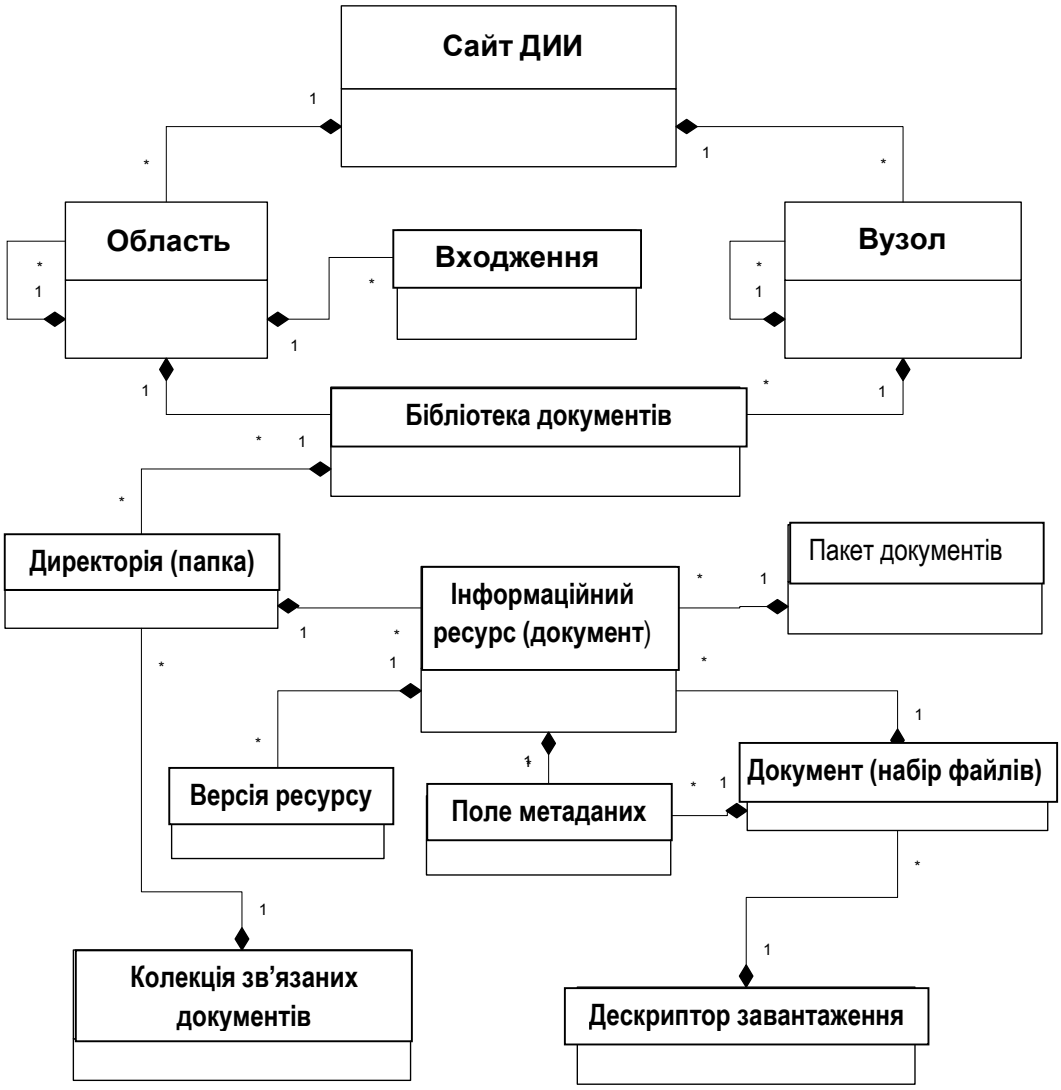

Рисунок 2.6 - Схема зв'язків між основними об'єктами системи

Склад інформаційного забезпечення системи СКВІР відповідає принципам рубрикування вхідної й використовуваної інформації. Як видно із цього опису, схема зв'язків між інформаційними об'єктами системи СКВІР носить ієрархічний характер, тобто більшість зв'язків має тип «від цілого до частини».

Схема цих зв'язків між основними інформаційними об'єктами СКВІР зображена на рис. 2.6..

Вибір такої організації інформаційного забезпечення уможливлює

структуроване зберігання інформаційних ресурсів. Крім того, вона сприяє спільній роботі декількох користувачів СКВІР - підтримці функції ведення версій документів, розмежуванню прав користувачів і т.д.

Наявність таких інформаційних об'єктів, як подання й структури метаданих, робить навігацію, пошук і перегляд умісту в системі більше зручним.

Побудова порталу на основі технологій SharePoint вирішувало більшу частину завдань по організації доступу до інформаційних ресурсів, зокрема, дане рішення дозволяло організувати загальну пошукову систему, надавало централізоване керування доступом і мало досить зручні інтерфейси доступу до інформаційних ресурсів. Однак, дані технології не вирішували основне завдання подання інформаційних ресурсів у єдиному структурованому виді.

Проблема ефективного використання інформаційних ресурсів, що накопичуються в системі, вирішувалася за допомогою таких додаткових засобів, тобто:

- зведений тематичний рубрикатор інформаційних ресурсів;
- тематичні добірки посилань для різних категорій користувачів.

Таким чином, основним і найбільш складним завданням при розробці системи СКВІР було завдання побудови системи ведення й керування рубрикатором інформаційних ресурсів. Завдання рубрикації інформаційних ресурсів (розбивки потоку ресурсів на тематичні добірки) вирішувалися за допомогою методики, описаної в наступному підрозділі.

2.6. Методика побудови тематичного рубрика тора СКВІР

Під рубрикатором у даній роботі розуміється класифікаційна таблиця ієрархічної класифікації, що містить повний перелік включених у систему класів і зв'язків між ними й призначена для систематизації інформаційних ресурсів а також для пошуку в них. Предметна рубрика – елемент

інформаційно-пошукової мови, що представляє собою коротке формулювання теми природною мовою.

Постановка завдання. На основі аналізу інформаційних процесів, описаних у попередньому розділі були зроблені висновки про необхідність розробки тематичного рубрикатора, що задовольняє наступним вимогам:

- багатомірність - наявність декількох предметних областей, у кожній з яких може бути визначене власне дерево тематичних рубрик.
- багатокористувальницьке керування рубриками, зв'язками між ними й інформаційними ресурсами.
- різні типи зв'язків між тематичними рубриками.
- наявність типізованих метаданих у рубрик і інформаційних ресурсів.
- наявність ключових слів у тематичних рубрик, які характеризують конкретну тему в рамках множини сусідніх тем.
- можливості автоматичної рубрикації вхідних інформаційних ресурсів на основі опорних множин інформаційних ресурсів.

При складанні тематичного рубрикатора необхідно виходити з наступних правил:

- кожний документ предметної області повинен мати відповідну предметну рубрику;
- не повинно бути рубрик, яким відповідає відносно мала кількість документів;
- рубрики по можливості повинні бути чітко відділені одна від одної. Для близьких по змісту рубрик краще мати короткі коментарі, у яких випадках проставляти одну з рубрик, у яких випадках обидві рубрики.

Рішення завдання побудови рубрикатора могло б ґрунтуватися на ключових особливостях портального рішення на базі технологій SharePoint, зокрема його ієрархічній структурі. Дана ієрархія формується на основі зв'язків спадкування між тематичними областями. Тематичні області є

розділами порталу, які забезпечують подання сховища інформаційних ресурсів у вигляді структури єдиного дерева. У вузлах цього дерева перебувають тематичні області, що представляють собою Web-Сайти - набори сторінок, бібліотек документів і списків, призначених для зберігання інформаційних ресурсів - документів і елементів списку, що включають метадані.

Для кожної теми може бути заданий набір вхідної інформації (посилань на ресурси, які можуть перебувати як на порталі, так і поза нього). Структура SharePoint має деревоподібний вигляд і не дозволяє будувати повноцінний тематичний рубрикатор. Для його побудови було потрібно використання додаткових семантичних технологій, що не входять до складу SharePoint - технологій Semantic Web. Очевидними перешкодами для використання даного підходу служили наступні обмеження, що накладаються реалізацією порталу на основі технологій SharePoint:

- відсутність зв'язків між тематичними областями;
- неможливість побудови незалежних ієрархій рубрик;
- відсутність метаданих у тематичних областей;
- неможливість установлення зв'язків між тематичними рубриками.

Корпоративні портальні системи, що працюють у локальному середовищі підприємства або організації, відрізняються від Internet порталів тим, що не вимагають зовнішнього доступу з боку довільних користувачів або агентів. Більше того, у нашім випадку тематичний рубрикатор будувався по кінцевій безлічі конкретних предметних областей і не мав потребу в особливих перевагах онтологічного підходу. Таким чином, розумним рішенням став підхід, заснований на використанні технології Topic Maps для побудови схеми рубрикатора й інформаційної бази порталу.

Реалізація підходу. На основі описаної моделі були спроектовані структури таблиць і зв'язків елементів рубрикатора. Основні частини цієї моделі зображені в додатку . Варто відзначити, що отримана в підсумку модель не була альтернативою для наявної моделі тим SharePoint. Вона

доповнювала її особливими типами зв'язків, завдяки яким дерево областей і ресурсів стало графом, у вершинах якого розташовані елементи моделі Topic Maps, а дугами є відповідні семантичні зв'язки. При цьому сама внутрішня структура сховища порталу не змінювалася.

Для керування й роботи з темами рубрикатора були створені спеціальні засоби адміністрування, що дозволяють легко оперувати з елементами рубрикатора - змінювати й додавати теми, наповнювати їхніми ресурсами й т.д. Як механізм зіставлення структури рубрикатора й областей порталу використовується служба реплікації змін, завдяки якій всі виправлення у змісті рубрикатора моментально проносяться в структуру областей і входжень порталу.

Одним з основних типів зв'язків, застосовуваних у СКВІР, є зв'язок, що базується на посиланні. Такого типу зв'язки використовуються для порівняння інформаційних об'єктів (таких як вузли, бібліотеки й документи) елементам списків інших типів, наприклад, входження в область можуть посилатися на документи загальної бібліотеки, при цьому їхній список відображається користувачеві СКВІР залежно від аудиторії, у яку входить даний користувач. Елементи рубрикатора (зокрема, рубрики) також використовують такі зв'язки для порівняння з інформаційними ресурсами СКВІР. Основним способом такого зіставлення є вказівка веб-адреси ресурсу (URI), однозначно ідентифікуюча шуканий ресурс.

Теми рубрикатора – основні інформаційні об'єкти в рубрикаторі інформаційних ресурсів, які є елементарними одиницями ієрархії рубрикатора. Кожна з тем несе в собі строго певне змістовне навантаження й відповідає певному поняттю, що має актуальність у контексті даного рубрикатора. У тематичному рубрикаторі інформаційні ресурси розташовуються в прив'язці до даних тим, причому розташування інформаційного ресурсу в межах якої-небудь теми означає прив'язку до даного ресурсу певної вузько специфікованої семантики, що несе в собі тема.

Входження інформаційного ресурсу – під даним об'єктом розуміється

фіксація факту зв'язку між інформаційним ресурсом і темою рубрикатора. Входження може бути виражене у вигляді визначення в ресурсі поняття, що відповідає темі, згадування теми в ресурсі й інших видах відносин;

Типи входжень – типи, що відповідають певним видам відносин між темою й ресурсом, наприклад, згадування, визначення, непряме відношення й інші.

Типи тим – теми, які по своїй суті відповідають поняттям тих або інших груп тим. Типи тим призначені для визначення набору характеристик тим, які повинні бути виражені однозначно для кожної теми, тому кожній темі може відповідати тільки один тип теми.

Зв'язок між темами – фіксація факту зв'язку певним відношенням двох або декількох тим. Кожний зв'язок заснований на певному типі зв'язку.

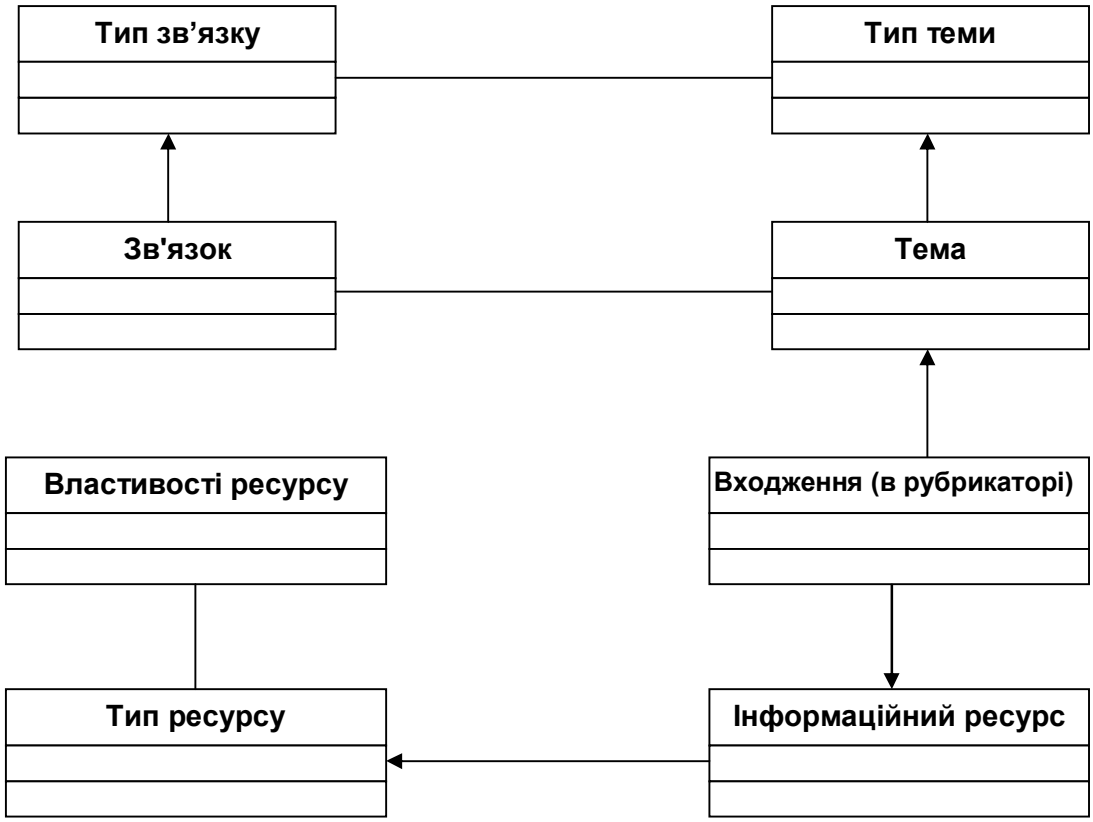

Рисунок 2.7 - Схема зв’язків між об’єктами

Типи зв'язків між темами – відносини, побудовані на темах. Дані відносини будуються як упорядкований і фіксований набір типів тим, які

можуть брати участь у зв'язку. Типи зв'язків визначають можливість утворення зв'язків того або іншого роду між темами.

Тип входження інформаційних ресурсів – вид відносин між інформаційним ресурсом і темою, що визначає характер можливого входження інформаційного ресурсу в тему. Кожне входження інформаційного ресурсу в тему може бути типізовано не більш ніж одним типом входжень.

Типи ресурсів – вид інформаційних ресурсів. Типи ресурсів призначені для можливості формування профілів ресурсів - набору типів властивостей ресурсів, які можуть бути визначені для ресурсів одного типу.

Властивості ресурсів – визначення властивостей в інформаційних ресурсів означає присвоювання певних метаданных ресурсу, причому в кожного ресурсу може бути визначене властивість, що входить у набір властивостей типу даного ресурсу.

Зв'язки між основними інформаційними об'єктами рубрикатора зображено на рис. 2.7.

Таким чином, рубрикація інформаційних ресурсів, що зберігається в СКВІР, досягається двома способами:

- по її розташуванню в дереві інформаційних об'єктів СКВІР;
- згідно рубрикатору, що містить інформацію про приналежність даного ресурсу до тієї або іншої рубрики.

У першому випадку послідовність ідентифікаторів (назв і URL-Адрес) вкладених інформаційних об'єктів, останнім з яких є шуканий інформаційний ресурс, задає послідовний набір вкладених рубрик у які входить даний ресурс.

Всі матеріали, що надходять із зовнішніх організацій, класифікуються відповідно до шляхів завантаження й значеннями полів метаданих, описаних в дескрипторі завантаження.

У справжній моделі дескриптора передбачається, що класифікація ведеться по наступних п'яти рівнях:

- Web-Вузол порталу - джерело інформації;
- бібліотека документів - Звітна форма (збірник звітів);
- папка в бібліотеці - Період звітності;
- документ у папці - Звіт деякого типу (у деякому розрізі);
- поля метаданих документа - Основні характеристики звіту.

Тут проведений приклад логічної відповідності між інформаційними об'єктами СКВІР і вступників. У процесі автоматизованого завантаження досягаються всі ці рівні рубрикатора.

У процесі експлуатації системи СКВІР з'ясувалося, що формальні способи завдання правил рубрикації вступників пакетів документів не завжди достатні, тому що існують випадки, при яких приналежність документа до якій або рубриці не описується в метаданих або семантичній структурі вхідного пакета. Для таких випадків потрібні додаткові дії по наповненню відповідних рубрик. Це наводить на думку про можливість використання засобів автоматичної рубрикації.

2.7. Проектувння системи

Після огляду семантичних порталів, методів їх реалізації, технологій детального аналізу їх функціонування та визначення основних переваг і недоліків кожного з них перейдемо до етапу проектування програмного продукту. Тут буде містись чіткий та повний опис розроблюваного продукту.

Для створення програмного продукту було вибрано мову програмування Ruby і фреймворк Ruby on Rails по наступних причинах:

- Мова має високу ефективність розробки програм і увібрала в себе найкращі риси Perl, Java, Python, Smalltalk, Eiffel, Ada і Lisp.
- Багатоплатформова реалізація інтерпретатора мови Ruby поширюється як вільне програмне забезпечення.

- Перенесена на багато платформ. Мова розроблялася на GNU/Linux, але працює на багатьох версіях Unix, DOS, Microsoft Windows (частково, Win32), MacOS, BeOS, OS/2 і т. д.
- Має каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних.
- Інтеграція з готовими Windows рішеннями.

Після того як був вибраний фреймворк для написання програмного продукту, було вибране рішення, яке допоможе створити свого роду корпоративну мережу для організації і надати її користувачам доступ до цих даних. У цьому випадку було вибрано Microsoft Share Point(про його можливості було описано вище) і бібліотеку DataMining для отримання аналізу вхідної інформації, яка разом з Microsoft Share Point дасть змогу визначити і вказати, куди і до якої теми віднести даний файл.

Наступним етапом є побудова діаграми варіантів використання, яка зображена на рисунках 2.8 і 2.9. На даних діаграмах зображені актори і функції, які вони можуть виконувати в web-системі.

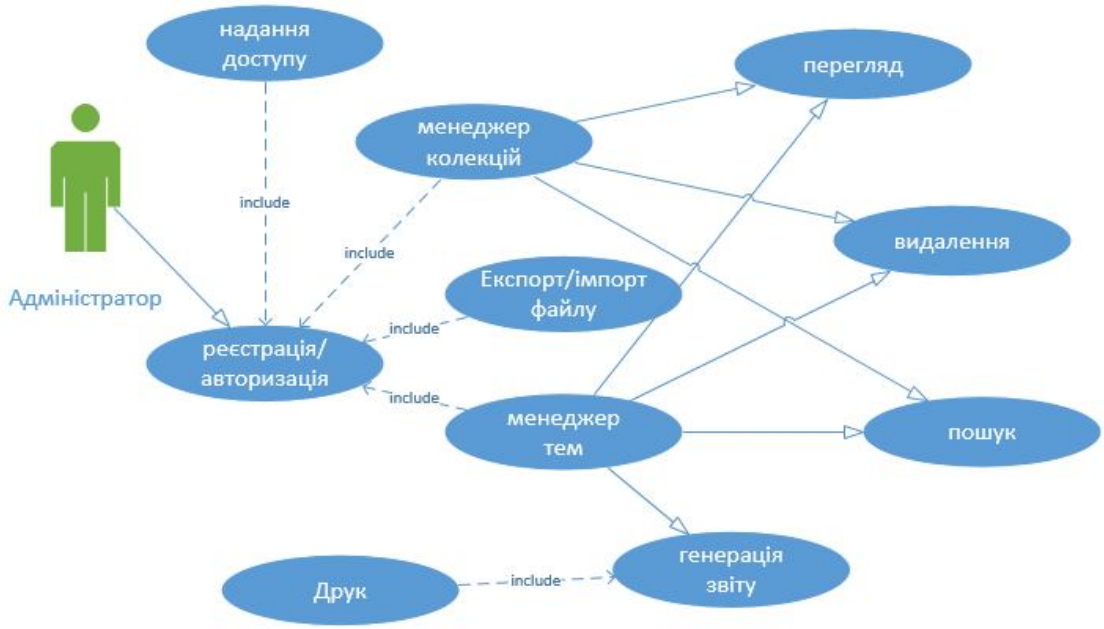

Рисунок 2.8 – Діаграми варіантів використання для адміністратора

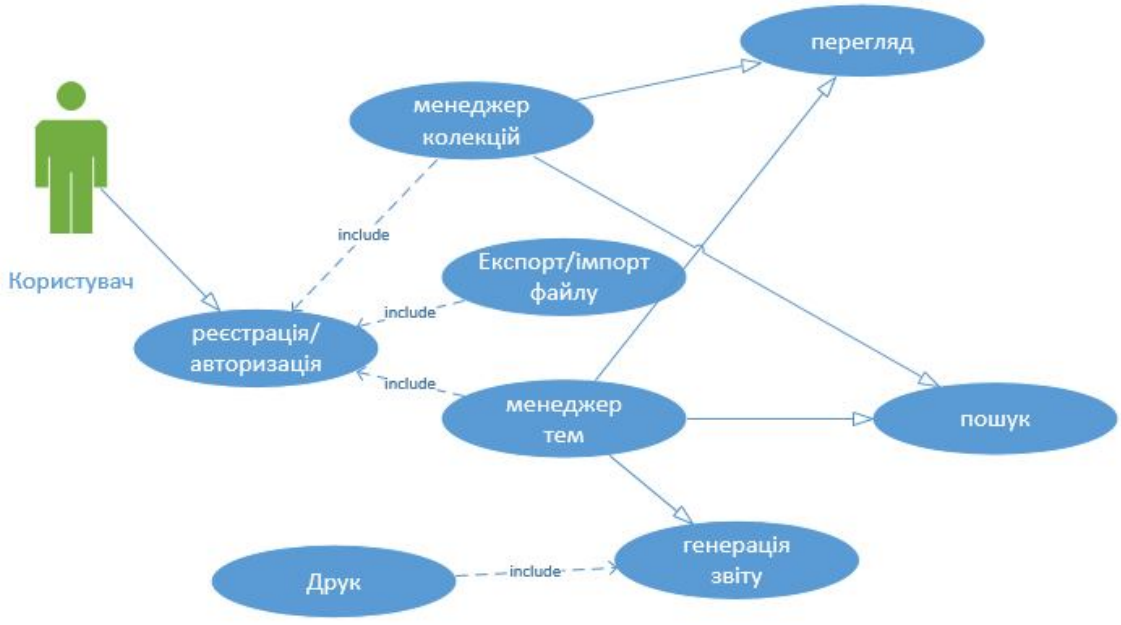

Рисунок 2.9 – Діаграми варіантів використання для користувача

Для відображення основної бізнес-логіки розроблюваної системи побудованно UML - діаграму класів, яка зображена на рисунку 2.10. Діаграма класів відображає класи спроектованої системи і відношення між ними.

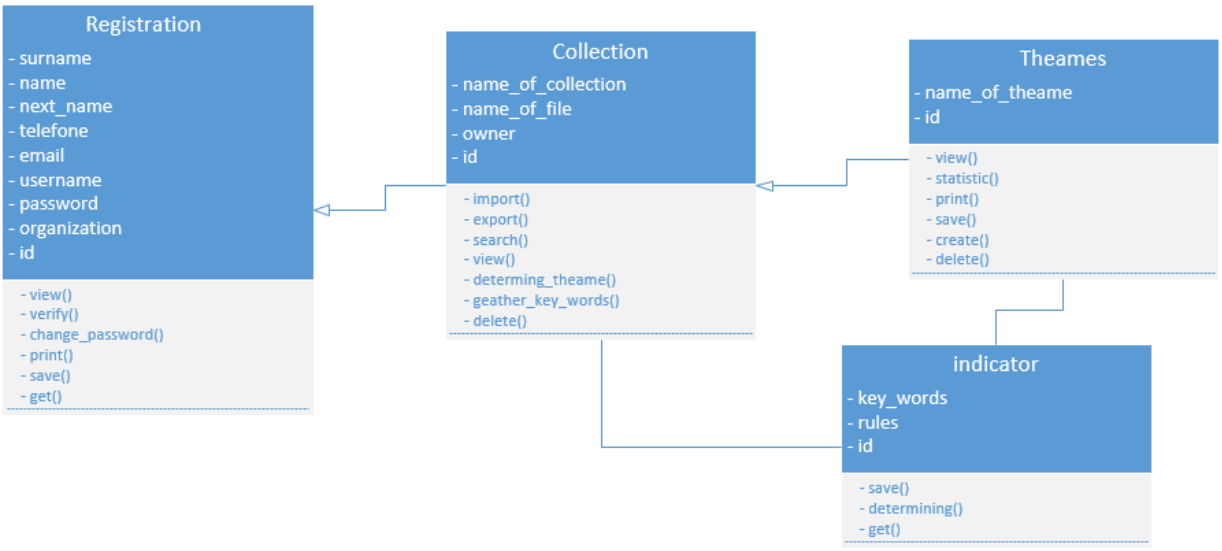

Рисунок 2.10 - Структура діаграми класів

Згідно діаграми класів в системі ключовими є чотири класи, в яких є свої методи. В класі «індикатор» організовано свого роду фільтр, в який записуються ключові слова, вирази, правила пошуку, для визначення в яку тему відносити завантажений файл і по яким критеріям і правилам опріділяти його приналежність до тієї чи іншої теми. Клас «тема» і «Колекція» в свою чергу тісно взаємодіють з класом «індикатор», для отримання його можливостей. Клас «тема» дає можливість створити тему, видалити і переглянути існуючі теми і їх вміст, тобто файли, які належать до даної теми і звісно можливість пошуку інформації по вкладенній інформації.

«Колекція» містить назву колекції, власника цього файлу. У класі «реєстрація» описаний сам процес реєстрації. Реєстрація є обов'язковою, вона потрібна для авторизації і для входу в систеу, адже користувач або адміністратор, який є не зареєстрований, не може зайти в систему.

При реєстрації обов'язковим моментом є заповнення всіх полів. Якщо якесь поле було не заповнено, то реєстрація не пройде успішно і буде видана помилка, що не всі поля заповненні. Після того як користувач заповнив всі поля з'являється повідомлення про успішну реєстрацію.

Щоб авторизуватися користувачу лише потрібно перейти на головну сторінку сайту і у вікні авторизація ввести ім’я користувача і пароль. Після того користувач авторизований, і він отримає доступ до можливостей системи.

За допомогою даної діаграми буде розроблено програмний продукт, структура і методи якого повинні бути в тому чи іншому класі для реалізації системи. Також так як ми використовуємо фреймворк Ruby on Rails і при написанні потрібно строго дотримуватися MVC патерну, то діаграма класів буде слугувати свого роду планом створення семантичного порталу.

Висновки до розділу II:

1. Розглянуто технології побудови семантичних порталів.
2. З'ясованно як відбувається анотування документа і її формальна модель.
3. Описана сервісна архітектура ЕБ
4. Запроектовано поведінку і струтуру системи

РОЗДІЛ ІІІ

ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕМАТИЧНОГО РУБРИКАТОРА В СИСТЕМІ КОЛЕКТИВНОГО ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ

3.1. Обгрунтування вибору середовища і мови програмування

Сама програмна реалізація СКВІР являє собою сукупність взаємопов’язаних модулів або систем, які отримують дані на вході, потім обробляють їх і в кінці аналізують і записують їх систему. Кінцеві користувачі при взаємодії з даними системами працюють тільки з UI, яка надає можливість керувати і проводити операції CRUD з інформаційними ресурсами які є і які потраплять в систему. Під час виконання дій над інформаційними ресурсами або об’єктами, то система починає тісно взаємодіяти з функціями Microsoft SharePoint Portal Server. Microsoft SharePoint Portal Server у свою чергу, взаємодіє із базою даних Microsoft SQL Server, де буде зберігатися вся вхідна інформація, тобто наші дані.

Програмне рішення робочих ділянок для користувачів і адміністраторів СКВІР вміщують в себе наступні компоненти:

- Модуль, який надає можливість керувати сховищем з інформаційними ресурсами. В свою чергу модуль складається з засобів для навігації по ієрархічній структурі веб системи. В даних системах доступ зосереджений до основних інформаційних об'єктів, інформаційних ресурсів системи;
- Модуль, для завантаження вхідних файлів(інформаційних ресурсів) – згідно назви буде призначений для зберігання вхідних файлів і автоматичного завантаження колекцій, що потрапляють до системи із різних джерел. Також запис пов'язаних з файлами або колекціями метаданих;

- модуль рубрикації вхідної інформації - він дозволяє структурувати або розподіляти вхідні інформаційні ресурси по темах, рубриках. Також модуль підтримує тематичний рубрика тор, тобто зв'язує теми відношеннями і управляє групами властивостей вхідних даних, які є в системі;
- модуль для організації публікації RSS - відновлень, що виконує функцію настроювання публікації RSS-Відновлень у системі СКВІР;
- модуль керування доступом – надає можливість автоматизовано керувати груповим(між організаціями) доступом до інформації;
- Модуль для пошуку інструментів – дає можливість користувачам шукати потрібну інформацію в системі;
- Модуль дял публікації фактографічних ресурсів - він дозволить відобразити в контенті веб-вузлів табличне і графічне представлення даних;

Серверне реалізація програмної системи, що розташована на серверах системи СКВІР складається з наступних компонентів:

- Microsoft SharePoint Portal Server – рішення яке дає змогу створити портал, сервер веб-сайтів, що забезпечить роботу з інформаційними ресурсами, в нашому випадку з файлами, колекціями і надасть можливість водночас зберігання метаданих і відношень цих даних до тих чи інших ресурсів. Також ключову роль дане рішення відіграє в функціонуванні веб-сайту, розмежуванні доступу до самої системи і контенту який є всередині. Тобто дане рішення допоможе з керуванням користувацькою аудиторію і дасть змогу розділити вхідну інформацію і область її видимості між організацями;
- база даних яка використовується для рубрикатора інформаційних ресурсів розміщена на сервері Microsoft SQL Server і включає в себе набір методів, які якраз допоможуть розмежувати або керувати доступом до даних системи СКВІР. Системи СКВІР взаємодіють з іншими підсистемами, і офіційні процедури для автоматичної обробки, додавання і зміни даних,

які контролюють цілісності в ході операцій, проведених за допомогою програмного забезпечення, сервер служби синхронізації даних між сервером і інформації рубрикатора ресурсів SharePoint Portal, призначених для синхронізації баз даних SharePoint Portal Server і рубрикатора періодично виконує дії рубрикатора призначені для підтримки даних рубрикатора і Microsoft SharePoint Portal Server на сьогоднішній день є в актуальному відносно один одного стану.

- Сервер служби реплікації бібліотеки SharePoint - призначений для реплікації регулярних колекцій статичних ресурсів.
- Пошукова система, яка включає в себе серверні продукти Microsoft Search і RCO для BackOffice.

На рисунку 3.1 зображена схема для взаємодії модулів системи СКВІР, вона зображена за допомогою UML діаграми компонентів. Взаємодія між модулями зображена штрих-пунктирними лініями

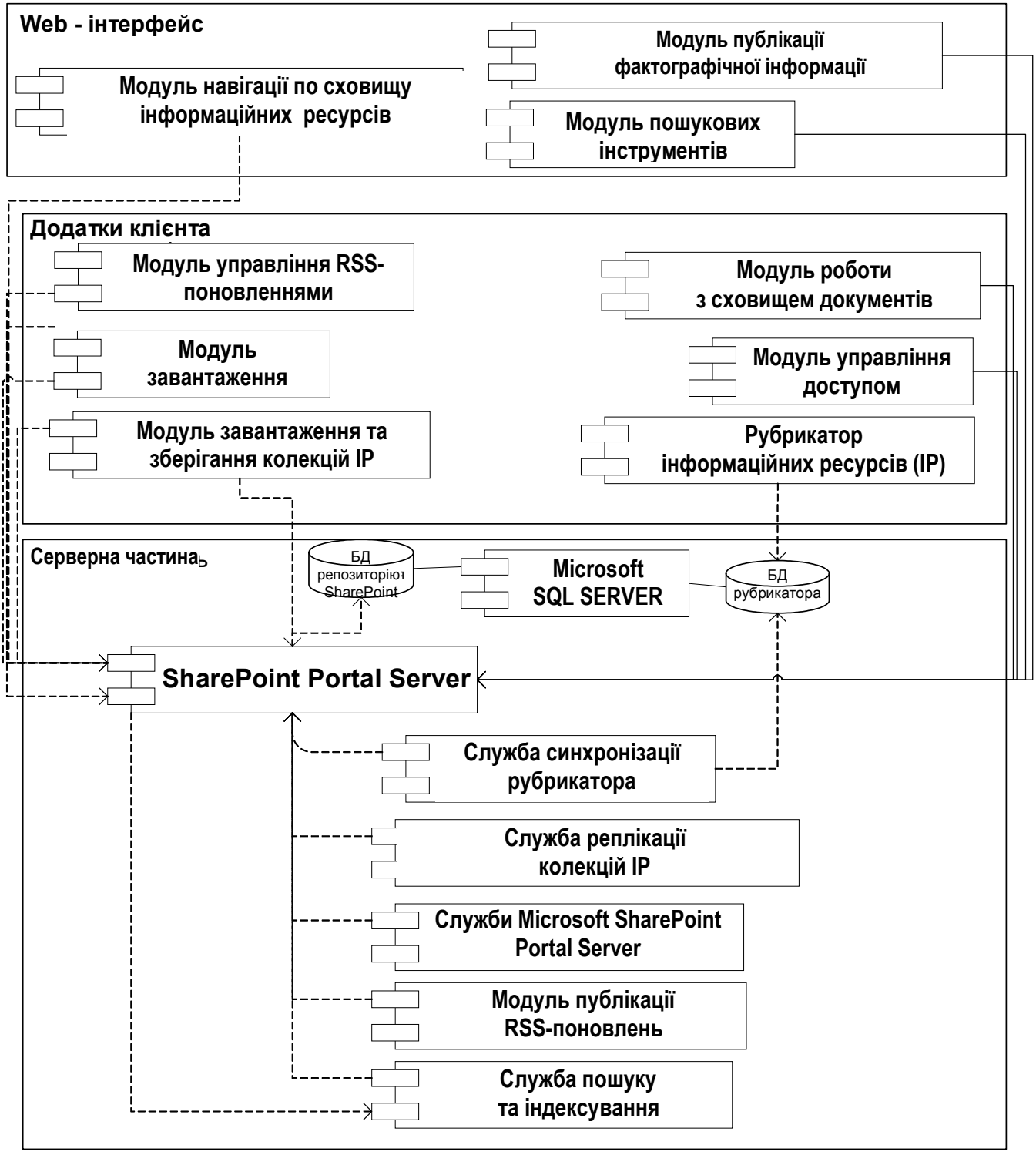

Рисунок 3.1 - Схема структури програмного забезпечення

Модуль для рубрикації інформаційних ресурсів(файлів), звертається до бази даних рубрикатора, розташованого на СУБД Microsoft SQL Server. Саме звернення реалізоване на мові Ruby, а також до серверної частини SharePoint Portal Server, використовуючи протокол SOAP. Код підключення до бази

даних має наступний вигляд:

```
require "mysql"

@db_host  = "localhost"
@db_user  = "root"
@db_pass  = "root"
@db_name = "alldata"

db = Mysql::Client.new(:host => @db_host, :username => @db_user, :password => @db_pass, :database => @db_name)
```

Модуль завантаження інформаційних ресурсів, модуль завантаження документів модуль і робота з сховищем файлів, документів, використання протоколів WebDAV і SOAP, звертаючись до SharePoint Portal Server, в результаті чого здійснюється додавання та відновлення даних в базі даних.

Синхронізація рубрикатора служби і SharePoint Portal Server відноситься до API, SharePoint Portal Server, відстеження змін об'єктів і реєструє відповідні зміни в рубрикаторі бази даних, розташованої на сервері баз даних під керуванням Microsoft SQL Server.

Служба реплікації відповідає за відновлення колекцій інформаційного контенту ресурсів SharePoint Portal Server, в результаті чого інструмент SharePoint Portal Server для вставки об'єктів в базі даних.

Служба реплікації відповідає на події відновлення вмісту в колекціях інформаційних ресурсів SharePoint Portal Server, викликаючи при цьому функції в SharePoint Portal Server по вставці об'єктів у БД.

При розробці програмних систем СКВІР використовувалися сучасні методи проектування і розробки програмного забезпечення, яке має широке використання в даний час і розвиток. Система реалізована з використанням багаторівневої архітектури з використанням різних інструментів і технологій. Для створення програмних модулів СКВІР використовувався високорівневий фреймворк для веб-розробки Ruby On Rails. Ruby On Rails розробники стверджують, що їхня технологія на багато разів більш продуктивніша для

веб-розробки і забезпечує опосередкований доступ до всіх функцій операційної системи, файлової системи, бази даних, а також взаємодія між різними компонентами СКВІР. На основі цієї технології реалізовані серверне програмне забезпечення, а також робочі місця користувачів і адміністраторів. Модель розробки базується на архітектурі Модель-Представлення-Контролер (Model-View-Controller, MVC) і дана технологія перенесена на багато платформ. Вона була розроблена на Linux, але працює на багатьох версіях Unix, DOS, Microsoft Windows (в тому числі, Win32), Mac OS, BeOS, OS / 2 і т. далі., що іншими словами означає не привязаність до ОС і працювати на різних інвайроментах.

Для веб-сервера був використаний Nginx з модулем Phusion Passenger, Ruby on Rails і використання в якості бази даних PostgreSQL. Це друга база даних для CRUD операцій і аналізу на стороні нашого розробленогоя сервера, що вони не проводилися на стороні SharePoint і її бази даних. Сам Nginx являє собою HTTP-сервер і зворотний проксі-сервер, поштовий проксі і TCP / UDP проксі для загальних цілей. У Nginx процеси одночасно обслуговують безліч з'єднань мультиплексії системного виклику в їх операційній системі select, epoll (Linux) і kqueue (FreeBSD), що дана розробка дозволить вирішити проблему з великою кількістю клієнтів і одночасної роботи, розробленої в нашій системі.

Автоматична рубрикація на основі методу SVM виглядає наступним чином:

Серед методів, використовуваних авто рубрикування є два основні підходи:

• рубрикування на основі машинного навчання на прикладі;

• рубрикування, засноване на знаннях, тобто опис категорій експертів.

Автоматичне рубрикування на основі машинного навчання розглядається як поняття, до якого ми повинні навчитися, рубрики. Машине навчання засноване на прикладах текстів, які раніше були оброблені вручну.

Для забезпечення автоматизованого інформаційного рубрикування ресурсів, що надходять на портал було вирішено, згідно всіх переваг і автоматизації, використати засоби автоматичної класифікації документів, вбудованих в SharePoint Portal Server. Можливість використання цих інструментів в рамках системи стало предметом подальших досліджень, так як якість категоризації сильно впливає на ефективність розподілення вхідної інформації на категорії.

Цей метод заснований на тому, що інформаційні ресурси, приписані до певної категорії є позитивними прикладами для даної категорії, а також документи знайдені тільки в категоріях, які залишаються - негативними. Перед використанням методу SVM необхідно заповнити деякі категорії, характерними для них ключовими словами, фразами і після цього зробити індексацію вмісту для інформаційних ресурсів порталу.

Метод SVM будує статистичні моделі, засновані на категоріях опорних множин набору інформаційних ресурсів. Статистична модель побудована на основі входження ключових слів у зміст або метаданих інформаційних ресурсів, з додатковою кількістю ключових слів, пов'язаних з ключовими словами тем, рубрик. Автоматична рубрикація може бути зроблено після проходження процесу навчання на основі еталонних наборів.

Під час процесу рубрикування створюється список нових інформаційних ресурсів і їх входжень в тематичний рубрикатор, з тим же ресурсом можуть бути віднесені до різних категорій. Експерт несе відповідальність за водіння рубрика тором. Він повинен обробляти кожну модуль і це є результатом автоматичного категоризації і атрибуції в разі неправильної категорії інформаційних ресурсів, щоб автоматичний рубрика тор міг відхилити цей запис. Розробка алгоритмів для автоматичного семантичного аналізу тексту (AsemAT) посилюється особливо в зв'язку зі створенням мережі загального користування.

Розпізнавання тексту. Коли OCR призначена для завершення будівництва синтаксичної структури пропозицій, адекватної семантичної

структури. Іншими словами - це переклад природної мови, формальної семантичної мови, який здатний працювати з комп'ютером. У чистому вигляді ця можливість може бути корисна у всіх завданнях, що вимагають розпізнавання тексту або мови. Найпоширенішим завданням є голосове управління і OCR. Використавши повний аналізатор тексту ми маємо можливість значно підвищити якість розпізнавання. Це особливо стосується розпізнавання мови, де все ще існують значні труднощі в розрізненні слова, слів, які звучать подібно. Механізм правильного вибіру змісту альтернатив на основі змісту здатний істотно (в кілька разів), щоб зменшити неоднозначності лексичних одиниць.

Пошук документів. Вихідною підставою для обшуку, як правило, великої кількості неструктурованих або слабо структурованої інформації в текстах на природній мові. Масив текстів попередньо індексується. Індекс сигналізує відповідність між основними одиницями і пошуку документів, які їх містять. У найпростішому випадку, ці одиниці є слова (або форми слова). У більш простому випадку може бути тема тексту (документа), фрагменти фраз або цілих фраз або пропозицій. Крім того, можна виконувати пошук документів, "аналогічно набору". Якісний пошук по темі або документу, що визначає ступінь подібності до даного документу, то потрібно потрібно мати можливість визначити тему документа. Індекси можуть бути побудовані автоматично або вручну. Автоматично побудований, як правило, тільки індекс, заснований на словах (в дуже обмеженій формі, заснований на визначенні предмета тексту).

Синтез текстів. У вузькому сенсі цього слова в процесі синтезу означало побудову фраз і пропозицій на природній мові записів для формальної мови. До згенерованих фраз може бути висунуті або не висунуті вимоги на стилістичну правильність, але вони ні в якому разі не включають в себе семантичні і граматичні помилки. Такод сам це процес - повний синтез тексту, є як не менш - трудомістким завданням.

Перевірте правильності текстів. Так як автоматичний семантичний аналіз надає повний аналіз пропозицій, для нього, щоб перевірити граматичної правильності аналізованих повідомлень, таких як, наприклад, визначення узгодженості опорного слова, відмінкову форму дял заявки, кореляція і угоду між підметом і присудком.

Будівництво тезаурусов. Створення інформаційно-пошукових тезаурусів, словники термінів і раніше залишається вкрай важкою і трудомісткою роботоботою, яка потребувала ступіння автоматизації, яка є дуже низькою. Насправді, всі тезауруси створені вручну. Автоматично може бути лише перевірена тільки погодженність накопичених визначень. Альтернативою може стати підхід при створенні визначень для існуючого тексту з наступними описами (енциклопедії, підручники, довідники), а потім, при необхідності, скорегувати процес діалогу з експертом. При такому підході, потрібно мати можливість провести детальний аналіз семантики тексту.

Автоматичне реферування та анотування. Суть анотації (абстрагування) тексту, полягає в тому щоб сформувати короткий опис основних тем тексту. Є два різних підходи до анотування. У першому випадку виявлення невеликого числа речень в тексті якіо представляють собою той основний текст. Крім того, часто виділяють ключові слова. В іншому випадку, той, основний текст виявляють зміст як основну тему текст, а потім вже цей зміст виражають новим тестом, реченням.. Другий варіант є у більшості випадків набагато кращим, але він і набагато складніший. Всі сучасні системи анотування / реферування скеровуються на перший варіант.

Класифікація та рубрикація документів, визначення теми документів. Незважаючи на гадану простоту, проблема категоризації і визначення предмета документів дуже важко реалізувувана. На основі тільки ключових слів або синтаксичної структури простих фраз задовільно вирішити дану проблему не можна. Фрагментований підхід використовує загальні семантичні класи, але він також принципово нічого не змінить. Існуючі

системи забезпечують точність класифікації (і, отже, визначають предмети) в порівнянні з людською оцінкою, без заздалегідь визначених класів - близько 60%, використовуючи зумовлені класи і установки текстів на цю тему - до 95% .Механізм автоматично визначає тему текстів, є основним предметом нашого огляду, перш за все, включає в себе визначений обмежений список тем, категорій, які можуть включати в себе конкретні тексти. Цілком можливо, мінімалістичний підхід вичерпується, для списків тих чи інших категорій, які не перевищують десяти, наприклад, обмежений список основних стилів: художнього, наукового, публіцистичної, розмовної, конфесійних та багато іншого.

По-друге – максималістичний підхід - є найбільш комплексним підходом, який передбачає використання надзвичайно широких (десять тисяч) товарних позицій, які повинні охоплювати всі сфери людського знання. Наприклад, в основі цієї класифікації може бути каталог одного з пошукових сайтів (www.yahoo.com, www.meta.ua, www.yandex.ru і т.д.). В системі «Смислове Дзеркало» (www.ashmanov.com) використовуються 2500 рубрик. Інформаційна система університету «Росія» (uisrussia.msu.ru) в якості одного з варіантів з використанням "Класифікації правових актів Російської Федерації" (приблизно 1200 тем, 4 рівня ієрархії). Третій підхід можна назвати проблемно-орієнтованим – в ньому введиться наступна класифікація рубрик, які відповідають конкретному завданні, що вони в основному будуть вирішуватися на рівні системи. Зокрема, ці рубрики, які використовують контекстну рекламу: в залежності від. рекламованих продуктів або географічне покриття мережі продажу створює обмежений набір тем (кілька сотень), до яких може бути зарахований аналізований текст, щоб продемонструвати прив'язаність до предмету реклами. Прикладами таких систем є RORER (www.rorer.ru, 82 суб'єктів). Основою категоризації, крім того, може служити бібліотечний каталог, наприклад, дослідницької служби конгресу Бібліотеки конгресу США (Legislative Indexing Vocabulary, LIV, 80 рубрик). Після визначення бажаної глибини рубрикування для кожного з

категорії, що визначають набір ключових слів, чия присутність в тексті може вказувати на його приналежність до певної рубрики.

Пізніше при визначенні теми конкретної сторінки в Інтернеті пропонують наступний алгоритм:

1. Система завантажує призначену для аналізу сторінки;

2. Відкриття коду сторінки - написано на мові HTML або будь-якою іншою мовою набір офіційний команд для браузеру, який містить теги - макет сторінки із зазначенням назви, ключового слова, основний текст, колір тексту, напівжирний або курсивний стиль шрифту і багато іншого.

3. На сторінці коду виділимо службові слова і написи, навігацію, допоміжні пристрої, зовнішні силки (тобто, що введе до іншого мережевого адресу, який знаходиться поза сайтом проаналізованої сторінки);

4. Сформувати загальний список слів з кодової сторінки, в тому числі в інномовному написанні (mp3, GPRS), які не є частиною офіційних команд.

5. З цього списку, викинути всі слова з 1 або 2 літер і всі цифри, тому що вони не можуть нести інформацію, яка допоможе визначити тему тексту.

6. Із загальної лексики двох стоп-слова (числівники, займенники, прийменники, сполучники, частки). Список стоп-слів закритий, ви можете встановити невеликий словник, щоб включити в його внутрішню структуру, створених. Причина видалення є можливість використовувати їх в текстах будь-якої теми..

Відповідним є текст для визначення теми теусту є тільки іменники, дієслова і прикметники. Це є ключові слова тексту. Непотрібні слід розглядати як прислівники, так як їх смислові навантаження не мають тематичного вектора, більшість з них може бути використана з однаковою частотою в текстах будь-якої рубрики;

7. Визначення частоти ключових слів;

8. Список ключових слів виконуємо дематизацію тексту - вони встановлені в первісному вигляді – з допомогою словника дематизації, який включає тільки слова з рубрикатором тем, інші залишаються в їх первинному

вигляді. Після того, як дематизацію словоформи слова будуть об'єднані в лексему, і їх частоти підсумовуванні;

9. У результаті буде сформований список ключових слів. Воно будуть визначатися загальним числом слів (F) і найдовшим словом (L);

10. Для кожного ключового слова тексту визначається число входжень (частота, e), довжина (в листах, L), місце тексту (p, наприклад, якщо слово в назві, то p = 4, список ключових слів, сторінка = 3, внутрішнє посилання p = 2, загальний текст p = 1), наявність тегів форматування поруч зі словом (наприклад, якщо слова відформатовані в напівжирний, курсив, напівжирний або виділенні, вказуючи її відносно вище ролі в тексті T = 2, а то й T = 1);

11. Існує рубрикатор теми (описано вище), в якому кожна тема містить ключові слова різної ваги слів в темі W (це також тільки іменники, дієслова, прикметники, наприклад, тема "Cars" слова кермо, спідометр, водій з слова в темі = 3 Вт, слова скло, гума, колеса, двигун –слова в темі = 2 Вт, вікна, дверні ручки - слова в темі = 1 Вт;

12. .Кожна з ключових слів тексту перевіряється на наявність в категорії. Якщо це станеться,то оновлюється теми рейтинг (індекс R, який вказує кількість потоків), в якій знайдене слово: теми X. Теми X слова в тексті промови в темі $R = R + W \times W$

13. У разі негативного результату оновлюється позатематиний рейтинг (міра кількості слів в тексті, які не ввійшли в жодну з тим, в подальшому аналізі поповнять словники окремих категорій і встановлюють вагу окремих тем по всьому тексту) позатематичні слова в тексті R = R + W;

14. Після обробки всіх ключових слів визначається вага кожної з тем (W), які будуть представлені у відсотках.

15. Відповідно до найбільшої ваги, визначається як ключ. Інші теми кваліфікувати як асоційовані.

Перспектива застосування запропонованого алгоритму автоматичного визначення тематики тексту (АВТТ) охоплює насамперед галузі інформаційного пошуку та контекстної реклами.

Сучасні інформаційні пошукові системи не обмежені певними користувачем словами. Нещодавно постулюється перехід від пошуку по словах до пошуку об'єктів [3], який передбачає розширення меж пошуку користувача, визначаючи тему його запиту та відповідних документів до рубрики. В результаті, пошукова система може знайти документи, де немає жодного визначеного користувачем слова, але повністю задовольнити його прохання. Інший підхід, запропонований в роботі [9] і буде розширювати пошук шляхом додавання слів, введених користувачем, адміністратором асоціативних зв'язків, тобто виявилення, в системі синонімів гіпероні мів і так далі. В обох випадках цей механізм повинен бути використаний AVTT, в тому числі на українській мові, якщо пошукова система підтримує індексацію і правильно індексує українську мову в документі.

Механізм може бути використаний для суто лінгвістичних проблем: поповнення словникові бази даних, термінологічних баз даних, забезпечуючи пошуку інформації та експертних систем.

Промисловість контекстної реклама включає в себе досить потужні системи, які забезпечують пошук конкретних рубрик для рекламованих продуктів. Механізм AVTT використовується при визначенні теми відкритої сторінки користувача відповідно до якого буде відображатися тематичний банер, який відповідає темі тексту і географічному розташуванні користувача.

Таким чином, перспективи практичної реалізації запропонованого алгоритму AVTT для української мови є дуже широким, і реалізація цього плану полягає в подальших дослідженнях.

Результати для автоматичної рубрикації будуть наступні:.

Для оцінки ефективності системи автоматичної рубрикації використовуються такі функції, як повнта і точність. Повнота - відношення R / Q, де R - кількість текстів правильно приналежних до деякої рубрики, а Q - загальна кількість текстів, які повинні бути віднесені до цієї категорії. Точність - відношення R / L, де R - кількість текстів правильно

класифікованих системою до певної категорії, а L - загальна кількість текстів, віднесена до системи цієї колонки.

Було встановлено, що для більшості категорій повнота коливалася від 70% до 90%, а точність в діапазоні від 20% до 40%. Таким чином, потрібні значні витрати на робочу силу в перевірці і відсіювання результатів автоматичної обробки.

Серед обставин, які ускладнюють проблему за допомогою машинного навчання для автоматичного рубрикування текстів можуть бути визначені наступним чином:

• Масив попередньо оброблених ресурсів не є і не може бути створений вручну протягом короткого часу.

• Масив обробляється ресурсами, але ресурси рубрикування користувачів, тобто люди, які не мають чіткого уявлення вмісту кожної категорії.

• Масив ресурсів, але рубрикування проведене не послідовно, тобто, наприклад, можуть бути групи аналогічних документів, що згадуються різними наборами категорій.

• Навчання проводиться на збір ресурсів і рубрикування на інший колекції.

Проблеми, пов'язані з автоматичним рубрикуванням повязанні з наступними причинами:

• Для автоматичної класифікування вам потрібно якимось чином створити категорію зображення як деякого виразу за допомогою слів і (або) термінів фактичного тексту. Це може бути зроблено на основі експертного опису або методів машинного навчання, для рубрикованих колекцій, рубрик.

• Автоматична обробка спеціальних текстів може бути дуже серйозною проблемою аналізу мовного матеріалу, використання контексту вживання слова або слів, які вимагають участі великого знання мови і предметної області, що дуже важко описати в програмному забезпеченні операційної системи автоматичного рубрикування.

Система призначена для автоматичного пошуку і повнотекстової інформаційно-аналітичної обробки (Online джерело, електронна пошта, корпоративні сховища даних і т.д.). Система дозволяє визначити джерела інформації і створити тематичні фільтри для створення персоналізованих структурованих колекцій інформації, який відповідає потребам користувача. Незалежність системи інформаційних джерел мови дозволяє йому контролювати джерела з різних регіонів світу, надаючи окремим клієнтам інформацію, що стосується їх, зацікавлених в певних областях. Збереження тексту в текст повнотекстової бази даних забезпечує доступ до інформації, яка серверному ресурсу через деякий час може бути вже не доступна (поновлення сайту, видалення з пам'яті і т.д.).

У бібліотеках по всьому світу щодня є мільйони кілобайт інформації, яка є джерелом для вирішення завдань. Але інформаційна насиченність інформації та постійність поновлення цієї інформації в Інтернеті і створюють значні труднощі в пошуку інформації. Час який витрачається в наш час на пошук та первинну обробку інформаційних потоків з використанням популярних пошукових систем, є порівнянням з традиційними методами для пошуку даних. Це вимагає використання нових інформаційних технологій, які підтримують систему для поліпшення виконання пошуку.

За допомогою системи можна значно підвищити ефективність збору інформації мережевих джерел, пошуку інформації відповідно до індивідуальних потреб користувачів і тематичних матеріалів для аналізу формування.

Функціональний список системи показана на рисунку 3.2.

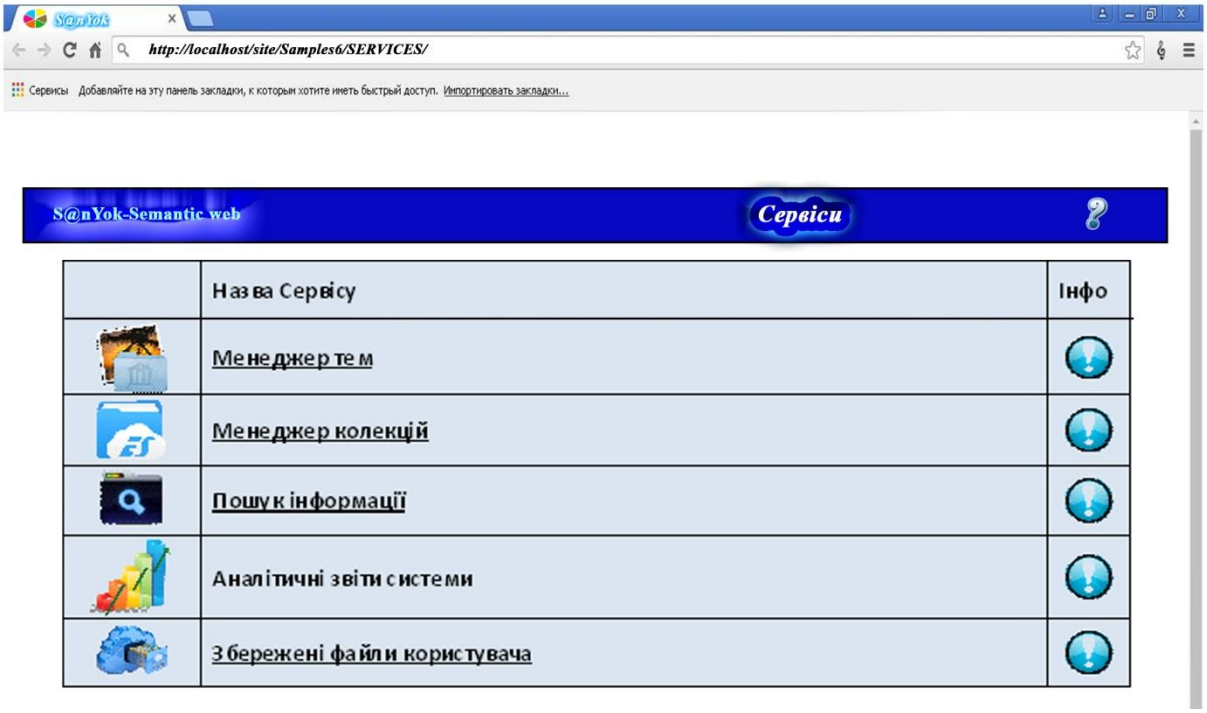

Рисунок 3.2- Функціонал системи Semantic web

Сервіс "Менедження тем" призначений для визначення системного набору тем, які використовуються для вирішення завдань пошуку і аналізу текстової інформації. При використанні сервісу "Менеджер тем" (див. рисунок.3.1) користувач отримує вікно з горизонтальним меню у вигляді тексту та піктограм. Горизонтальне меню складається з наступних елементів: Вихід теми (формування теми, Перегляд файлів), експорт / імпорт (експорт імпорт тем), вікна (по вертикалі, по горизонталі, шари, Каскад, Закрити всі) Допомога (Зміст, Про контент теми) сервіс "Створення теми" призначений для визначення системи приналежності відповідних тем структурованих у вигляді дерева (рис. 3.6). По кожній темі в системі визначені індикатори, тобто набір логічних виразів пошуку об'єктів, який відноситься до тексту належить до обраної теми. Елементами логічних виразів є слова і фрази (використовуючи спеціальні символи *?, А ~~) пов'язані логічні оператори AND, OR, NOT та спеціальні оператори визначають приналежність одночасної присутності термінів.

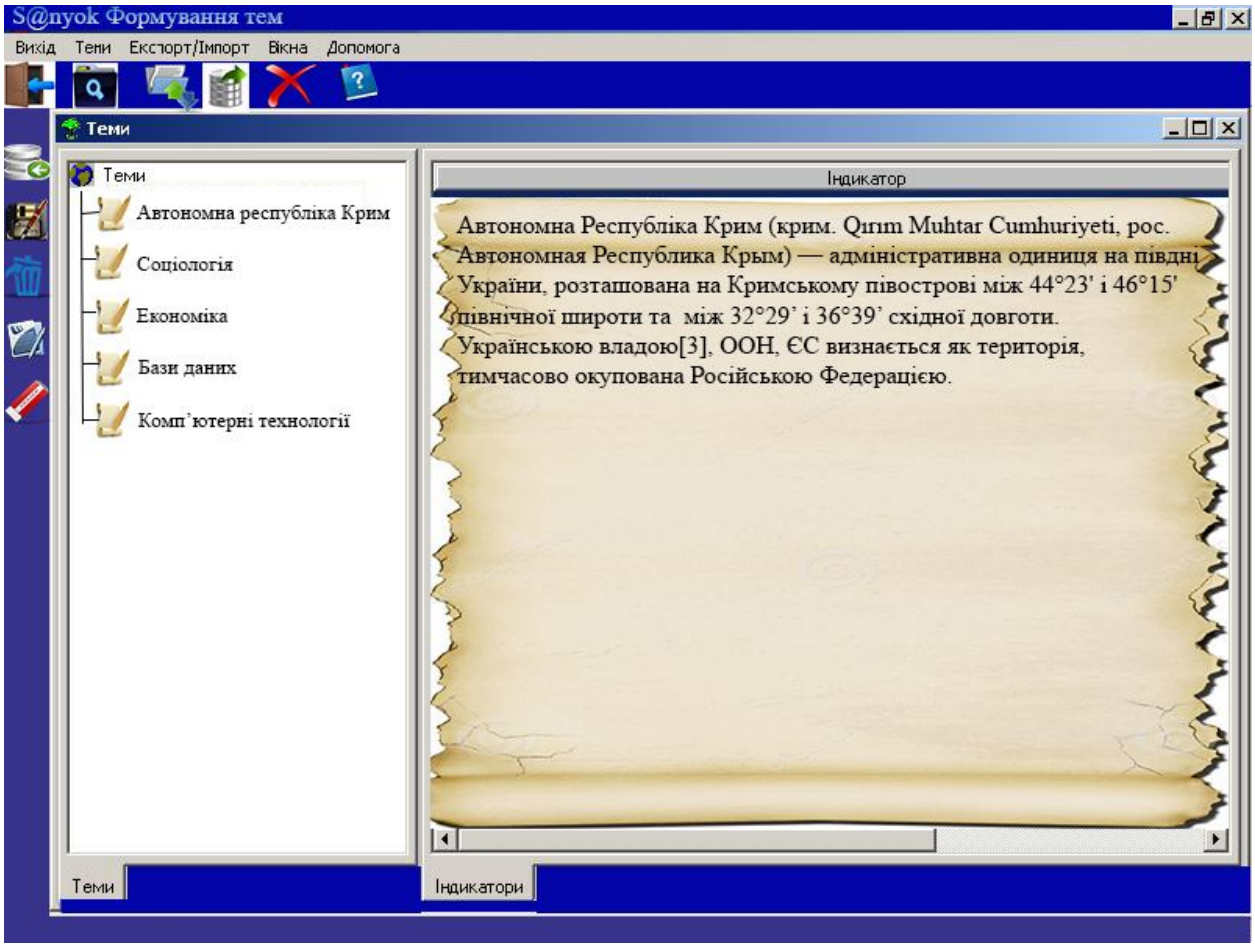

Рисунок 3.2 - Теми. Формування тем

В меню дерева тем вискакує меню: Оновити, Розгорнути все, Згорнути все, Додати, Редагувати, Видалити, реалізація якого повністю визначає функціональніможливості роботи з деревами і списками.

У списку індикаторів тем (права сторона вікна) при натисканні на праву кнопку "миші" вискакує наступне меню: Додати, Редагувати, Видалити, Перегляд файлів.

При виборі "Додати" користувач отримує вікно, показане на рисунку 3.3.

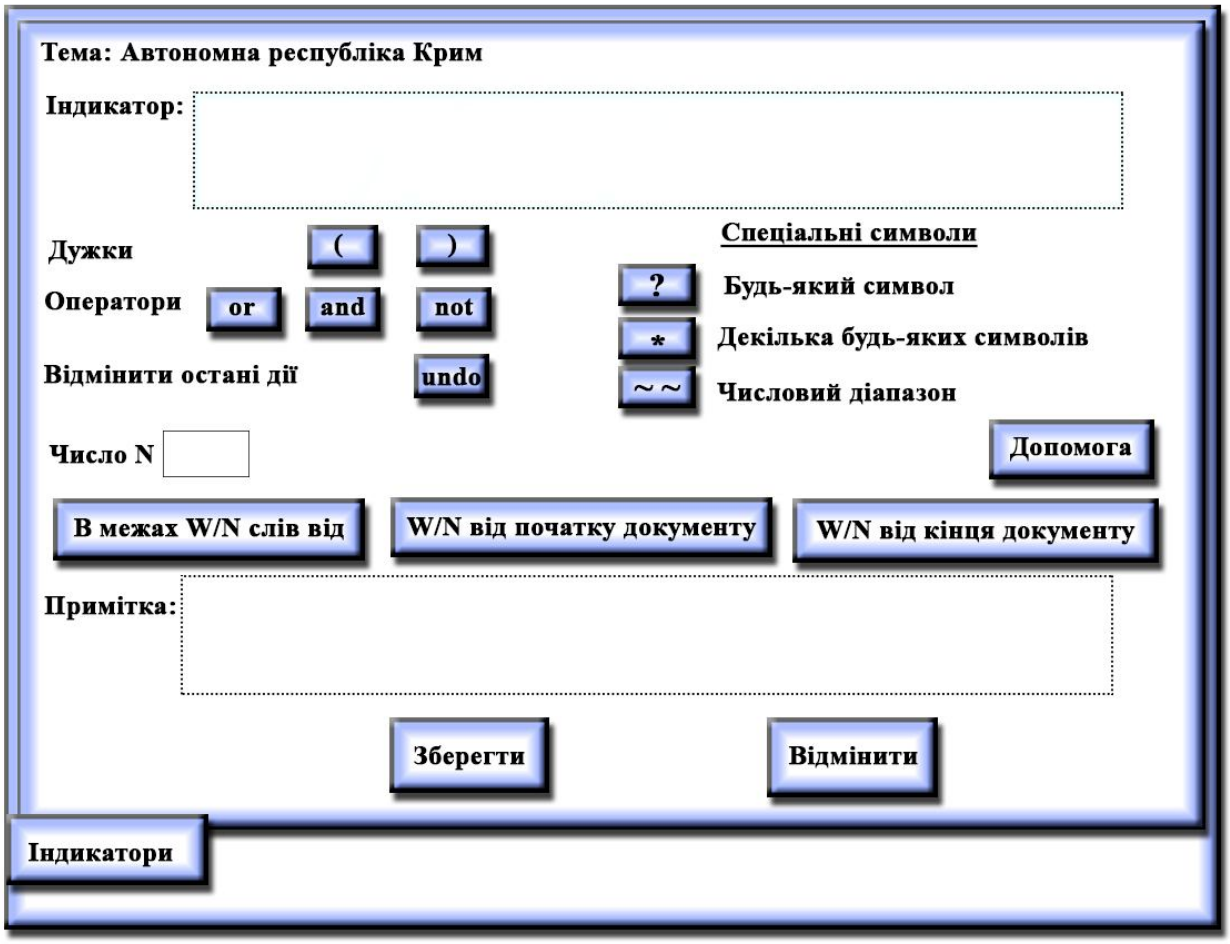

Рисунок 3.3 - Добавити індикатор

За допомогою допоміжних кнопок логічних операторів, спеціальні символи і елементи для визначення терміна пошуку околиці в формуванні індикатора для прискорення і полегшення формування індикатора. При натиску на кнопку, ставиться відповідний символ в тестове поле індикатора і воно якраз формує рубрикатор. Значення за замовчуванням N встановлено значення 3. Число N використовується при натисканні на кнопки «В межах W/N слів від», «W/N від початку документу» та «W/N від кінця документу». "Примітка" вікно використовується для зберігання довільної або допоміжної інформації, пов'язаної з індикатором.

У міру того як система реалізована для зміни є функція "Змінити", а для видалення - "Знищити" вказаний ідекатор.

В системі є реалізована можливість для перегляду файлів, які завантаженні в систему і відповідають умовам, викладеним у вікні пошуку (цей показник).

Шаблон пошуку, використовуваний в системі є логічним виразом, в якому пошукові терміни (слова і фрази), пов'язані логічними операторами:

Таблиця 3.1

Логічні оператори

AND	вживається для визначення необхідності одночасної наявності в ресурсі слів ,виразів, які пов’язує цей оператор
OR	використовується для визначення необхідності наявності в тексті одного з виразів, які пов’язує цей оператор
NOT	використовується для визначення необхідності відсутності в тексті виразу, який визначений за цим оператором. Якщо NOT - не перший в пошуковому рядку, то необхідно використовувати AND або OR з оператором NOT (наприклад, «Ріпка AND NOT бабка»).

Пошукові терміни можуть включати наступні спеціальні символи:

Таблиця 3.2

Спеціальні символи

?	Відповідає довільному символу
*	Відповідає довільній кількості довільних символів
~~	Числовой диапазон

Окрім того, слова, вирази або речення можуть бути пов’язані між собою спеціальними операторами, для визначення рубрики і одночасної наявності термінів. Терміни пов’язуються наступним чином.

Таблиця 3.3

Зв'язок термінів

<термін1> w/<n> <термін2>	<термін1> повинен зустрітись в околиці з n слів від <термін2> (по замовчуванню n = 5)
<термін1> w/<n> xfirstword	<термін1> повинен зустрітись в перших n словах (по замовчуванню n = 5)
<термін1> w/<n> xlastword	<термін1> повинен зустрітись в останніх n словах (по замовчуванню n = 5)

Слід зазначити, що в формуванні логічного виразу , також є дужки "(" і ")".

Сервіс "Пошук інформації" призначений для пошуку потрібної інформації користувачеві відповідно до потреб, проблем, які потрібно вирішити. Це базова система обслуговування, використовує певні теми, структури системи і структури колекцій для індексованих текстових файлів. Сервіси "Менеджер тем" і "Менеджер колекцій" використовуються для створення умов для найбільш ефективного і зручного пошуку текстової інформації, яка відповідає умовам користувача.

При виборі сервісу "Пошук інформації" в списку послуг (див. рисунок 3.5) отримати вікно "Пошук інформації" (рисунок 3.4).

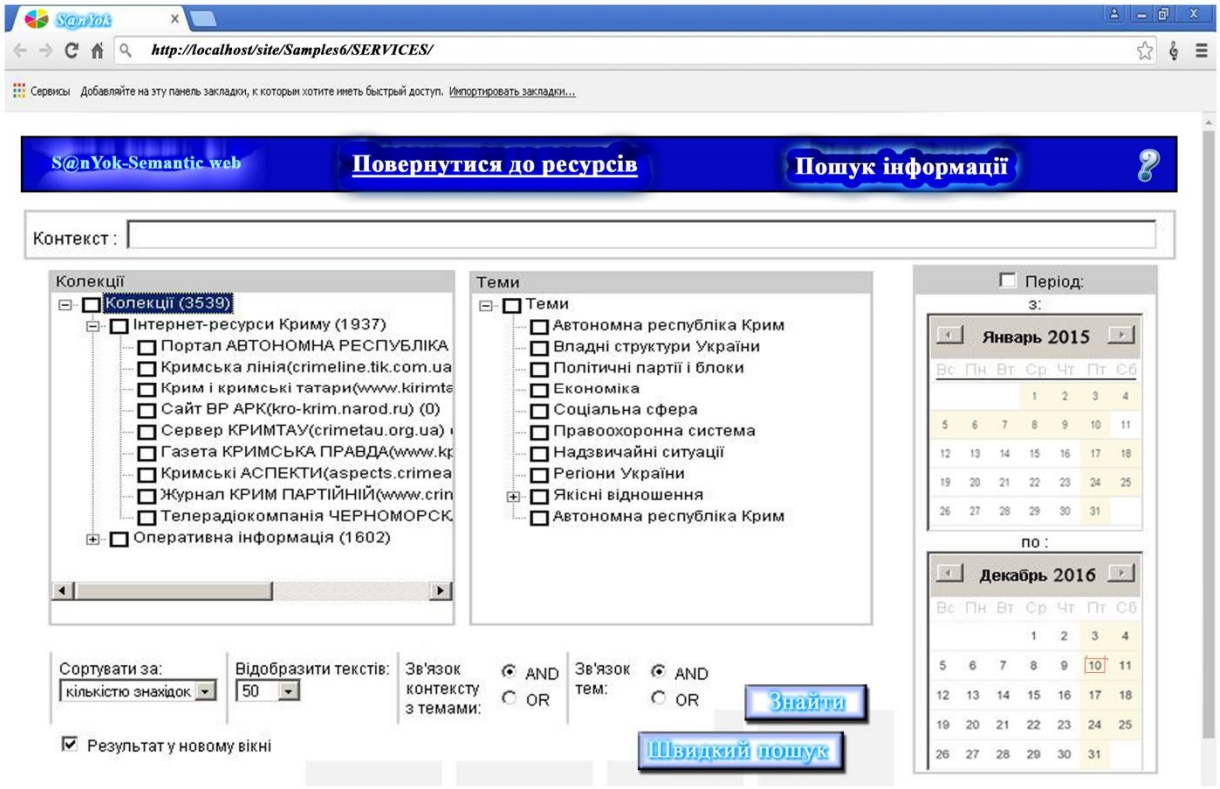

Рисунок 3.4 - Формування умов пошуку інформації

Вікно містить наступні елементи:

- ✓ поле введення "Контекст" (заповнення за бажанням);
- ✓ вибір з двох зон "Колекції" і "Теми";
- ✓ календарі, щоб визначити період часу, який включає в себе системну дату створення файлу;
- ✓ радіо-кнопки для вибору логічних операторів для зв'язку між індекаторами тем і звязку між показниками контексту і тим;
- ✓ поле вибору максимальної кількісті текстів, які з'являються в результатах пошуку (за замовчуванням - 45);
- ✓ вибір правильного впорядкування результуючого списку слів (за замовчуванням - "кількість відкриттів");
- ✓ нявність функціоналу "Знайти" і "Швидкий пошук";

Для пошуку не забудьте зазначити великий перелік колекцій (по що найменшій мврв, однієї колекції) і безліч предметів (по крайній мірі, один

суб'єкт під час відсутності контексту). Зверніть увагу, що вибір теми або колекції включає в себе вибір певного рівня підлеглих елементів.

При необхідності, пошук повинен розглянути і враховувати дату створення текстового файлу, користувач "включає" пошук пункт "Період" (заповнивши чекбокс) і календарях визначає потрібні дати. Зверніть увагу, що вибір колекцій в календарях встановлюється за допомогою часового інтервалу, який задовольнить всі вибрані користувачем колекції. Інформація в межах засначеного часу, яка включає в себе всі вибрані колекції, надана під колекцією дерев. Ви можете зменшити інтервал в календарі через певні проміжки часу відповідно до ваших потреб.

При формуванні умов пошуку за умовчанням встановлено, щоб відзначити елемент "Результат в новому вікні." Це дозволяє отримати результат кожного наступного пошуку в новому вікні і зберегти доступ до результатів попередніх пошуків. Якщо видалити мітку, то результат пошуку буде сформований в вікні попередніх результатів і знищить попередній пошук.

Для економії часу пошуку і відображення текстів у формуванні пошукових термінів, ви можете вибрати "Швидкий пошук" і отримати результат разом з темами в дужках буде відображається кількість файлів з текстами актуальних тем в обраних колекцій.

Кнопка "Знайти" в вікні, щоб ініціювати формування пошукових термінів в колекціях пошуку тексту, обраних відповідно до визначених умов (за умови дотримання, часу, контексту і т.д.). При формуванні логічного виразу для пошукової системи використовуються визначені в формуванних умовах пошуку логічні оператори, щоб об'єднати контекст і індикатори визначених тем.

При відсутності текстів, що задовольняють умовам пошуку, видається повідомлення, а в іншому разі надається вікно результатів, аналогічне зображеному на рисунку 3.5.

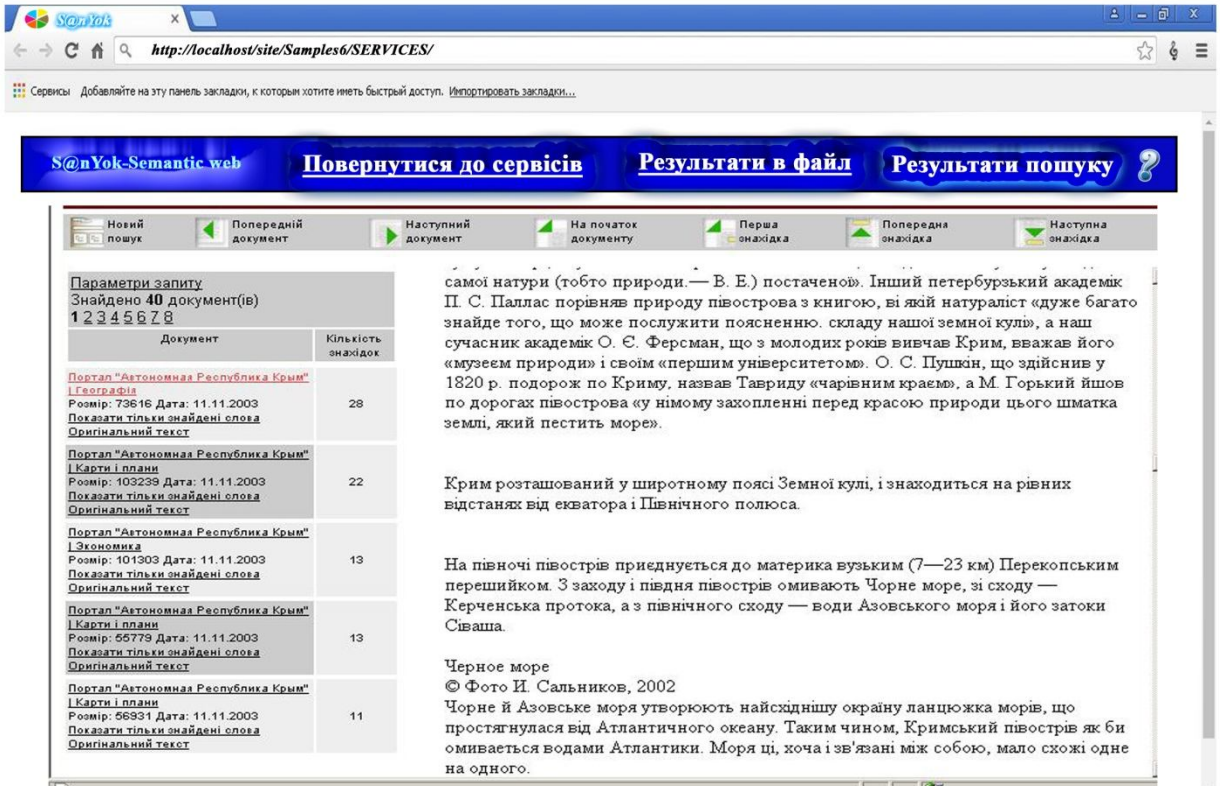

Рисунок 3.5 - Результат пошуку

Ліва сторона вікна "Результати" містить інформацію про кількість знайдених документів, а також перелік параметрів запиту отриманих документів (обмеження кількісті визначається в розділі "Пошук інформації" номером, відповідним за це). При натисканні на посилання "Параметри пошуку" ми отримаємо параметри запиту.

Посилання, що містяться в елементах результатів пошуку, дають можливість переглянути поточний текст, крім форми, яка забезпечує систему і навіть в своєму первісному вигляді і форму пошуку тільки пошуклвих слів (індикатори, певні теми).

Система забезпечує текст з виділеними пошуковими елементами (контекст індикатори тем). Кнопки на панелі у верхній частині вікна дозволяють перегортати сторінки тексту для першої знайденної інформації, попередньо знайденноїнаступно знайденної і повернутися на початок документа.

Крім того, кнопки на панелі у верхній частині екрану, дозволяють повернутися в вікно, щоб знов сформувати новий пошуковий запит ("Новий пошук") і перейти до списку документів ("документ Попередній", "Наступний документ") для їх перегляду. В даний час документ стає поточним при натиску на його границі в списку результатів.

Методи і технічні засоби, що підтримують автоматизовану побудову і розширюють тематичні онтології. Сформована онтологія може використовуватися при роботі з декількома джерелами інформації.

3.2 Розгортання програмного продукту

Для запуску системи програмного продукту потрібна будь-яка операційна система або то Windows або Linux. Саме на цих ОС було перевірено коректність роботи, а також потрібен браузер новіших версій, не старого покоління, тому що будуть не коректно відображатися UI-елементи на сторінці, це особливо стосується побудови і відображення графіків. Відносно апаратних вимог, то процесор має бути покоління Intel core I3-420 2.1 GHz і вище, оперативної пам’яті 4 -6 Гбайт, відео карта на 1 GB та якомога більше вільного місця на жорсткому диску. Але покращивши всі ці характеристики ми пришвидшемо роботу нашого сервісу, так як всі операції будуть швидше виконуватися в рази, і основною з них являється аналіз документу і запис його в відповідну рубрику.

3.3 Інструкція користувача

Встановлення:

При запуску системи потрібний сервер, на якому має бути встановлений і запещений nginx, він схожий на apache тільки в рази краще і новітніше на теперішній час. Сам сервер в мому випадку находиться

локально в вагранті, що дає змогу звертатися до даної системи як до сервера. Сам процес інсталювання даної системи займає 30 хв, доки не встановляться автоматично всі залежності для коректної роботи системи і не налаштуються всі конфігурації. Після того як сервер підніметься ми зможемо до нього звернутися по адресу localhost/site/. Також всі файли мають міститися в одній папці інакше система буде працювати не коректно. В разі доопрацювання або покращення сервісу потрібно встановлювати з'єднання з базою даних MSSQL Server, і встановити також середовище програмування Ruby on Rails і Angular, для зручності розробки.

Аналіз помилок:

При виникненні помилки з виходом з ситеми, потрібно почистити кукі в браузері. Дана помилка виникає, так як для процесу авторизації була взята модель ADFS, через яку було реалізовано реєстрацію і авторизацію. Active Directory (ADFS) спрощують доступ до систем і додатків за допомогою механізму перевірки достовірності на основі заявок (CBA). ADFS підтримує технології єдиного входу (SSO), що допомагають IT-організаціям у спільній роботі, не обмеженої межами організації. Дане рішення ідеально підходить для нашої системи, но є помилка, коли ми виходимо з системи, то наші дані зберігаються в кукі і зайшовши назад в систему ми зайдемо на аккаунт з якого вийшли, а рішення її очистка кукі з усією інформацію яка там збергається, тобто не саме краще рішення і проблема продовжує своє існування, до подальшого вирішення.

Базові функції системи:

- ✓ Реєстрація користувачів в системі;
- ✓ Авторизації користувачів в системі і надання їм відповідних прав доступу в системі.
- ✓ Експорт/імпорт файлів ;
- ✓ Проведення CRUD операцій для тем і колекцій;
- ✓ Налаштування рубрикатора;

- ✓ Виведення результатів у вигляду звіту і графіка, на якому відображенна статистика;
- ✓ Здійснення пошуку інформації яка є файлах;
- ✓ Перегляд інформації по рубриках;
- ✓ Друк результатів звіту і статистики ;
- ✓ Блокування та розблокування користувача;
- ✓ Змінна пароля користувача в системі;

Для початку роботи в сервісі потрібно бути зареєстрованим і мати доступ, тобто дозвіл до даних відповідної організації. Отримання доступа до системи, організації відбувається тільки через адміністратора(ів), які будуть обслуговувати систему.

3.4.Тестування

Невід'ємною частиною будь-якої реалізації програмного продукту є процес тестування або так званої перевірки роботи програми згідно вимог, які були зазначенні при початку реалізації продукту. Під час перевірки програмного продукту зазвичай виявляються дефекти системи, при яких система може не працювати, блокуватися або працювати не вірно. В разі виявлення дефекту або помилки, розробник має виправити її.

Тестування за своєю специфікою поділяється на два види: ручне тестування та автоматизоване. Під час ручного тестування всі операції виконуються вручну, тобто самі перевіряємо і оцінюємо, як працює та чи інша функція[13]. Автоматизоване тестування дає змогу перевірити код реалізації функцій, за допомогою якого маємо можливість протестувати фунціонал, і перевірити, чи повертається потрібний результат. Кожен з цих видів містить в собі вже свої види тестування.

Під час розробки даної системи було проведено:

- ✓ автоматизоване тестування за допомогою програмного засобу Selenium і Capybara ;
- ✓ ручне тестування, під час виконання якого перевірили систему за допомогою функціонального тестування.

Спочатку було проведено автоматизоване тестування за допомогою Selenium, перевірили чи працює система вірно, виконавши ту чи іншу операцію.

Приклад тесту перевірку запису файла в відповідну рубрику, чи система працює коректно:

```
require 'test_helper'

class DataSaverControllerTest < ActionView::TestCase
  test '#save_data' do
    params = {
      file_name: 'file_name',
      theame_of_file: theame_of_file ',
      theame: theame,
      text: 'text'
    }
    response = 'test.txt'
    mock_yaml_load_file(response)
    theame_of_file = DataSaverController.send_file
    assert_equal "Product Feedback by #{params[:file_name]} at #{params[:
theame_of_file]}", theame.subject
    assert_equal [response], theame.to
    assert_equal response, theame.tag
    assert_equal 'text/html; charset=UTF-8', theame.content_type
    body = response.body.encoded
    assert_includes body, params[:file_name]
    assert_includes body, params[:theame_of_file]
    assert_includes body, params[:theame]
    assert_includes body, params[:text]
  end

  private
```

```
  def mock_yaml_load_file(response)
    feedback_config_mock = { 'ctheame' => response }
YAML.stubs(:load_file).with('config/feedback.yml').returns(feedback_config_mock)
  end
end
```

Наступним видом тестування було «Тестування безпеки», яке відноситься до ручного тестування. Під час проведення даного виду тестування було перевірено авторизацію, яка згідно ідентифікаторів надає можливість кожному користувачеві увійти в систему під певними правами. Так як в системі є три типи ідентифікатора, то було проведено авторизацію з кожним з них для перегляду, чи надаються кожному певні права доступу до системи.

Висновки до розділу ІІІ:

1. Побудовано архітектуру і функції СВІР.
2. Розроблено тематичний рубрикатор СВІР.
3. Здійсненно програмну реалізацію програмного продукту.
4. Проведено тестування програмного продукту і написано авто тести на фунціонал.

ВИСНОВКИ

У результаті виконання магістерської роботи розроблено web-орієнтований програмний модуль для побудови семантичних порталів.

У даній науковій роботі вирішені наступні завдання:

- здійснено аналіз існуючих підходів до розробки порталів знань на основі технологій Semantic Web. Встановлено, що для структуризації й класифікації інформаційних ресурсів необхідно було використати підхід Topic Maps;
- спроектовано оптимальну модель структури зберігання й доступу до інформаційних ресурсів. Дана модель була застосована при розробці системи СКВІР;
- здійснено дослідження порталів знань і семантичних порталів;
- розроблено тематичний рубрикатор для СКВІР;
- здійснено програмну реалізацію автоматичного рубрикування на основі семантичного аналізу вмісту інформаційних ресурсів. Даний підхід базується на певних способах подання знань про предметну область і текстової інформації. Він забезпечує значно кращу якість автоматичної класифікації ресурсів за рахунок того, що семантичний аналіз змісту ресурсу забезпечує більш достовірну оцінку приналежності ресурсу до тієї або іншої тематичної рубрики;
- здійснено тестування розробленого програмного продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1.*W3C* Semantic Web Activity. – http://www.w3.org/2001/sw/Activity

2. *SemanticWeb* organization. – http://www.semanticWeb.org/

3. *Getting* into RDF "Semantic Web using N3", Tim Berners-Lee – http://www.w3.org/2000/10/swap/Primer.html

4. *Web* Architecture: Describing and Exchanging Data", Berners-Lee, Connolly, Swick, W3C Note 7 June 1999. – http://www.w3.org/1999/04/WebData

5. *Metadata* Architecture, W3C Design Issues. – http://www.w3.org/DesignIssues/Metadata

6. *RDF* and Metadata, Tim Bray, June 09, 1998. – http://www.xml.com/xml/pub/98/06/rdf.html

7. *The Power* of Metadata, book chapter by Rael Dornfest, Dan Brickley. – http://www.openp2p.com/pub/a/p2p/2001/01/18/metadata.html

8. *Web* Metadata: A Matter of Semantics by Ora Lassila, IEEE Internet Computing, July-August 1998. – ttp://computer.org/internet/ic1998/w4030abs.htm

9. *W3C*, The Semantic Web Home Page. – http://w3.org/sw/

10. *AgentWeb*, resource guide and newsfeed covering Agent-related technologies. – http://agents.umbc.edu/

11. *A Model-Theoretic* Semantics for DAML+OIL, W3C Note 18 December 2001. – http://www.w3.org/TR/daml+oil-model

12. *An Axiomatic* Semantics for RDF, RDF-S, and DAML+OIL, W3C Note 18 December 2001. – http://www.w3.org/TR/daml+oil-axioms

13. *DAML+OIL* (March 2001) Reference Description, W3C Note 18 December 2001. – http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218

14. *XML* Schema, RDF Schema & DAML Comparison. – http://www.isi.edu/expect/Web/semanticWeb/comparison.html

15. *W3C* Web Ontology. – http://www.w3.org/2001/sw/WebOnt/

16. *Requirements* for a Web Ontology Language, W3C Working Draft. – http://www.w3.org/TR/Webont-req/

17. *SemanticWeb*: роль XML и RDF/ С. Декер, С. Мельник, Ф. ван Хермелен, Д. Фенсел, М. Клейн, Д. Брукстра, М. Эрдманн, Я. Хоррокс // Открытые системы. 2001 — № 9. – http://www.osp.ru/os/2001/09/041.htm.

18. *Distributed* XML: the role played by XML in the next-generation Web, Edd Dumbill. – http://www.xml.com/pub/2000/09/06/distributed.html

19. *XML* and the Web, by Tim Berners-Lee, XML World 2000, Boston 2000/09/06. – http://www.w3.org/2000/Talks/0906-xmlWeb-tbl/

20. *An Introduction* to the Resource Description Framework by Eric Miller, D-Lib Magazine, May 1998. – http://www.dlib.org/dlib/may98/miller/05miller.html

21. *Putting* RDF to Work, Edd Dumbill. – http://www.xml.com/pub/2000/08/09/rdfdb/index.html

22. *RDF* tutorial, Pierre-Antoine Champin (for developers). – http://www710.univ-lyon1.fr/~champin/rdf-tutorial/

23. *W3C* Web Service`s Home Page. – http://www.w3.org/2002/ws/

24.*Web* Services Architecture, W3C Working Draft 14 November 2002. – http://www.w3.org/TR/ws-arch/

25.*Web* Services Architecture Requirements, W3C Working Draft 14 November 2002. – http://www.w3.org/TR/wsa-reqs

26.*Web* Services Architecture Usage Scenarios, W3C Working Draft 30 July 2002. – http://www.w3.org/TR/ws-arch-scenarios/

27. *Web* Services Description Requirements, W3C Working Draft 28 October 2002. – http://www.w3.org/TR/ws-desc-reqs/

28. *Web* Services Glossary, W3C Working Draft 14 November 2002. – http://www.w3.org/TR/ws-gloss/

29. *Лифшиц* Ю., Семантический Веб, лекция, 2006. – http://logic.pdmi.ras.ru/˜yura/internet.html

30. *The Semantic* Web. By Tim Berners-Lee, James Hendler and Ora Lassila. Scientific American, May 17, 2001. –

http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

31.*The Semantic* Web Roadmap, Tim Berners-Lee, 1998. – http://www.w3.org/DesignIssues/Semantic.html

State of The SemanticWeb, Ivan Herman, Stavanger, Norway, 2007.

33. *Semantic* Web for Developers. – http://logicerror.com/semanticWeb-Webdev

34. *Extensible* Markup Language (XML) 1.0, W3C Recommendation 10.02.1998. – http://www.w3.org/TR/1998/REC-xml-19980210

35. *RDF/XML* Syntax Specification (Revised), W3C Working Draft 25 March 2002. – http://www.w3.org/TR/rdf-syntax-grammar/

36. *RDF* Model Theory, W3C Working Draft 29 April 2002. – http://www.w3.org/TR/rdf-mt/

37. *RDF* Semantics, W3C Working Draft 23 January 2003. – http://www.w3.org/TR/2003/WD-rdf-mt-20030123/

38.*RDF* Primer, W3C Working Draft 11 November 2002. – http://www.w3.org/TR/rdf-primer/

39.*RDF* Test Cases, W3C Working Draft 12 November 2002. – http://www.w3.org/TR/rdf-testcases

40.*RDF* Tutorial, W3C. – http://www.w3.org/TR/rdf-tuturial

41.*Resource* Description Framework (RDF): Concepts and Abstract Data Model, W3C Working Draft 29 August 2002. – http://www.w3.org/TR/rdf-concepts/

42.*Resource* Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999. – http://www.w3.org/TR/REC-rdf-syntax/

43.*Using* RDF to model multimedia content – slide «Relation with MPEG-7». – http://www.w3.org/Architecture/1998/06/Workshop/paper29/slides/slide13-0.html

44. *RDF* syntax, W3C Recommendation. – http://www.w3.org/TR/PR-rdf-syntax

45. *RDF* Schema, W3C Working Draft. – http://www.w3.org/TR/PR-rdf-schema

46.*RDF* Vocabulary Description Language 1.0: RDF Schema, W3C Working Draft 23 January 2003. – http://www.w3.org/TR/2003/WD-rdf-schema-20030123/

47.*Topic* Maps (XMT). – http://www.topicmaps.org/

48.*Text* Encoding Initiative. – http://www.tei-c.org/

49.*Metadata* Encoding and Transmission Standard. – http://www.loc.gov/standards/mets/

50. *Metadata* Object Description Schema (MODS). – http://www.loc.gov/standards/mods

51.*Encoded* Archival Description (EAD). – http://www.loc.gov/ead

52.*Learning* Object Metadata (LOM). – http://www.ltsc.ieee.org/wg12/

53. *Online* Information Exchange (ONIX). – http:// www.editeur.org/onix.html

http://cyberleninka.ru/article/n/strukturno-vidovoy-analiz-kraevedcheskih-elektronnyh-kollektsiy-bibliotek