

Створення та використання семантичних Wiki-ресурсів

Короткий зміст

Вступ	7
1. Технологія Wiki та її властивості.....	10
2. Україномовні Wiki-ресурси та енциклопедії.....	22
3. Семантизація Wiki-ресурсів.....	35
4. Створення та менеджмент Wiki-ресурсів у MediaWiki	65
5. Обробка семантики у Semantic MediaWiki	101
6. Побудова запитів до Wiki-ресурсів.....	110
7. Пошук у Wiki-ресурсах.....	125
8. Інтеграція семантичних Wiki-ресурсів з онтологіями Semantic Web. 134	
9. Використання Wiki-ресурсів в системах семантичного пошуку.....	142
10. Приклади використання Wiki-ресурсів у практичних застосунках. 151	
Додаток 1. Основні етапи побудови семантичного Wiki-ресурсу.....	160
Додаток 2. Глосарій.....	161
Додаток 3. Основні терміни і поняття.....	162
Додаток 4. Перелік скорочень	165
ЛІТЕРАТУРА.....	166

Зміст

Вступ	7
1. Технологія Wiki та її властивості.....	10
1.1. Wiki та концепції відкритих джерел	10
1.2. Основні елементи Wiki.....	11
1.3. Історія створення Wiki	13
1.4. Характерні риси Wiki-ресурсів.....	15
1.5. Технологічні платформа створення Wiki-ресурсів	16
1.6. Вандалізм у Wiki-середовищі	19
1.7. Теоретичний базис Wiki.....	20
1.8. Еволюція Wiki-ресурсів	21
2. Україномовні Wiki-ресурси та енциклопедії	22
2.1. Українська Вікіпедія.....	22
2.2. Українські Wiki-ресурси	26
2.3. Велика українська енциклопедія.....	28
3. Семантизація Wiki-ресурсів	35
3.1. Технології Semantic Web як засіб обробки розподілених знань	35
Онтології предметних областей	39
Мови подання комп'ютерних онтологій	41
3.2. Створення онтологій предметних областей за допомогою Protégé.....	44
Подання онтологій в Protégé.....	44
Основні етапи створення онтології.....	46
Створення класів.....	47
Створення властивостей об'єктів.....	48
Створення властивостей даних	52
Створення екземплярів.....	53
Візуалізація онтологій в Protégé.....	55
Побудова запитів до онтологій	56
3.3. Напрямки семантизації Wiki.....	59
3.4. Реалізації семантичних Wiki.....	60
4. Створення та менеджмент Wiki-ресурсів у MediaWiki	65
4.1. Базові можливості системи MediaWiki.....	65
Реєстрація в системі	65
Вхід в систему для зареєстрованих користувачів	67
Пошук сторінок в Wiki-ресурсу	67

Створення нової сторінки	67
Редагування сторінки	69
Обговорення сторінки	70
Вилучення сторінки	71
Історія сторінки	71
Скасування редагування	72
4.2. Розширені можливості системи	74
Створення заголовків	74
Посилання на інформаційні ресурси	75
Посилання на інші сторінки Wiki-ресурсу	76
Посилання на зовнішні інформаційні ресурси	76
Категоризація сторінок	78
Робота з шаблонами	80
Створення шаблону	80
Приклад застосування шаблону	84
Параметри шаблону	85
Керування включенням тексту шаблону до сторінок	86
Організація шаблонів	87
Копіювання шаблонів з одного Wiki-проекту до іншого	87
Зображення	88
Завантаження зображень	89
Розміщення зображення в тексті	91
Галереї	93
Форматування зображень	95
Таблиці	97
5. Обробка семантики у Semantic MediaWiki	101
5.1. Цілі та задачі Semantic MediaWiki	101
5.2. Семантичні властивості Semantic MediaWiki	103
5.3. Плагін семантичних форм Semantic Forms	104
Призначення плагіну Semantic Forms	104
Основні елементи плагіну Semantic Forms	105
Створення семантичних властивостей	106
Створення форм шаблонів	107
6. Побудова запитів до Wiki-ресурсів	110
6.1. Побудова запитів на основі функції ask	110
6.2. Кон'юнкція умов запитів	112
6.3. Використання шаблонів і компараторів у запитах	113
6.4. Об'єднання результатів запитів: диз'юнкція	115
6.5. Умови запитів щодо окремих сторінок	115
6.6. Підзапити і ланцюжки властивостей	116

Зміст

6.7. Використання шаблонів і змінних у запитах	117
6.8. Сортування результатів запиту	118
6.9. Посилання на результати семантичного пошуку	120
6.10. Приклади застосування запитів.....	120
7. Пошук у Wiki-ресурсах.....	125
7.1. Типи пошуку у Wiki-ресурсах	125
7.2. Використання типів інформаційних об'єктів для пошуку у Wiki-ресурсах	126
7.3. Використання онтологій для побудови типів інформаційних об'єктів.....	131
8. Інтеграція семантичних Wiki-ресурсів з онтологіями Semantic Web. 134	
8.1. Побудова онтологій засобами Semantic MediaWiki	134
8.2. Відповідності між елементами онтології та семантичного Wiki-ресурсу.....	138
9. Використання Wiki-ресурсів в системах семантичного пошуку.....	142
9.1. Основні напрямки розвитку інформаційного пошуку	142
9.2. Семантичний пошук у Web на основі онтологій.....	144
9.3. Використання тезаурусів у семантичному пошуку.....	147
9.4. Wiki-ресурси як джерело інформації для побудови тезаурусів та онтологій в системі МАПС	148
10. Приклади використання Wiki-ресурсів у практичних застосунках. 151	
10.1. Використання Semantic MediaWiki для побудови семантичного порталу	151
10.2. Визначення компетентності спеціалістів	153
10.3. Забезпечення прозорості рамок кваліфікацій	154
Додаток 1. Основні етапи побудови семантичного Wiki-ресурсу.....	160
Додаток 2. Глосарій.....	161
Додаток 3. Основні терміни і поняття.....	162
Додаток 4. Перелік скорочень.....	165
ЛІТЕРАТУРА.....	166

Вступ

На даний час світовим співтовариством вже усвідомлений головний напрямок у боротьбі з інформаційним вибухом – перехід від збереження й обробки даних до накопичення й обробки знань. Стрімке поширення Web – всесвітнього сховища, що відкриває доступ до величезної кількості різних інформаційних ресурсів (IP), незалежно від їх географічної і національної локалізації та засобів подання – викликало потребу у нових, більш ефективних методах представлення, аналізу та використання розподілених знань, орієнтованих на функціонування у відкритому інформаційному середовищі.

Через складність отримання знань важливим стає питання забезпечення їх інтеперабельності та повторного використання. Перспективним підходом до вирішення цих проблем є використання знань для семантичної розмітки наявної інформації.

Слід відмітити, що здобувати знання значно простіше не з природномовних або мультимедійних IP, а з тих, які вже структуровані та мають якусь семантичну розмітку. В той же час доцільно орієнтуватися на розповсюджені та зрозумілі для користувачів форми подання інформації. Обом цим умовам задовольняють семантичні Wiki-ресурси. Семантичні Wiki-ресурси досить легко інтегрувати із різноманітними інтелектуальними інформаційними системами (ІС). Wiki усе частіше сприймається як новий тип колаборативної технології, що може вплинути на управління знаннями, а також підтримувати створення й спільне використання знань.

Тому все більш актуальними стають питання створення та підтримки різноманітних довідників та енциклопедій, що базуються на технології Wiki (одним з найбільш відомих прикладів яких є Вікіпедія), та напрямки семантизації цих ресурсів. Крім того, виникає необхідність в методах і засобах, які дозволяють пов'язувати ці ресурси зі вже існуючими базами знань (БЗ). Через те, що значна частина сучасних ІС, призначених для роботи у Web, орієнтовані на використання онтологій та пов'язаних з ними розробок Semantic Web, доцільно розробляти засоби, що дозволяють інтегрувати Wiki-ресурси та онтології, що можуть надалі застосовуватися в якості баз знань інших розподілених застосунків.

Дане видання орієнтовано на розробників різноманітних семантичних Wiki-ресурсів різної спрямованості та обсягу. Його можуть використовувати як інженери зі знань, так і експерти предметних областей, що займаються розробкою довідників та енциклопедичних видань. Для попереднього ознайомлення з тематикою Wiki рекомендується послідовність розділів 1, 2, 4, 10. Для інженерів зі знань, що мають досвід роботи з онтологіями, але не використовували їх у розробці Wiki-ресурсів, доцільно використовувати послідовність розділів 1, 4, 5, 6, 8. Досвідченим розробникам Wiki-ресурсів, які потребують семантичного розширення цієї технології, доцільно

скористатися послідовністю розділів 3, 5, 6, 7, 8. Якщо потрібно інтегрувати Wiki-ресурс із інтелектуальним Web-застосунком, то цьому може допомогти послідовність розділів 8, 9 та 10.

Розділ 1 описує основні властивості Wiki-технології, її концепцію та базові елементи інформаційних ресурсів, що можуть бути побудовані на її основі. Цей розділ дозволяє визначити доцільність застосування Wiki для створення застосовних систем певного призначення.

Розділ 2 містить приклади широко відомих україномовних Wiki-ресурсів, що є прикладами застосування цієї технології.

У розділі 3 розглянуто сучасні напрямки інтелектуалізації Web на основі засобів менеджменту знань та їх зв'язок з семантизацією Wiki-ресурсів. Розглядаються основні напрямки проекту Semantic Web, пов'язані зі створенням та використанням онтологій. Значна увага приділяється роботі з редактором онтологій Protégé.

Розділ 4 присвячено створенню Wiki-ресурсів у середовищі MediaWiki. Ці рекомендації орієнтовані на тих користувачів, що ще не мають достатнього досвіду у створінні та редагуванні Wiki-сторінок, але потребують у використанні як базових, так і розширених можливостей цієї технології – посилань на інші ресурси, категорій та шаблонів.

У розділі 5 розглядається семантичне розширення MediaWiki, що забезпечує створення та обробку семантичних властивостей Wiki сторінок. Це дозволяє створювати семантичні Wiki-ресурси таким чином, щоб наявні в них знання можна булоб здобувати та обробляти автоматично іншими інтелектуальними застосунками.

Розділ 6 присвячено побудові запитів до Wiki-ресурсів. В таких запитах використовуються як категорії, так і семантичні властивості окремих сторінок. Застосування запитів забезпечує інтеграцію знань, що містяться у Wiki ресурсі, та значно полегшують сприйняття цих відомостей кінцевими користувачами. На основі цього розділу користувачі можуть навчитися самостійно будувати семантичні запити та вбудовувати їх у текст Wiki-сторінок.

У розділі 7 розглянуто засоби, що надає Wiki-технологія для пошуку відомостей, потрібних користувачам. Проаналізовано, які типи пошуку доцільно використовувати для різних завдань, та як саме застосовувати для цього елементи Wiki.

Розділ 8 розглядає інтеграцію семантичних Wiki-ресурсів з онтологіями Semantic Web. Наводяться методи співставлення елементів онтологій з елементами семантичного Wiki-ресурсу. Це має забезпечити як побудову онтології за множиною Wiki-сторінок, так і використання онтологій предметних областей для створення системи категорій та семантичних властивостей Wiki-ресурсу.

У розділі 9 описується, як саме онтології, побудовані за Wiki-ресурсом, можуть бути використані для персоніфікації семантичного пошуку у Web. Описано приклад інформаційно-пошукової системи, що реалізує ці функції.

У розділі 10 наведено приклади використання Wiki-ресурсів у практичних застосунках, що має продемонструвати інтеграцію Wiki-технології із різноманітними сучасними інформаційними системами, орієнтованими на функціонування у відкритому середовищі Web.

У додатку 1 наводяться основні етапи побудови семантичного Wiki-ресурсу, додаток 2 – це глосарій, який спрощує навігацію у даному виданні, додаток 3 містить перелік основних термінів, пов'язаних з семантичними Wiki-ресурсами, а у додатку 4 запропоновано перелік скорочень, що використовуються у тексті.

Список літератури містить посилання на 62 джерела українською, англійською та російською мовами. Ці ресурси дозволять читачам більш глибоко ознайомитися з різними питаннями, пов'язаними із семантичними Wiki-технологіями, що залишилися поза розглядом у даному виданні.

1. Технологія Wiki та її властивості

1.1. Wiki та концепції відкритих джерел

Сьогодні забезпечити на практиці спільне, вільне й відкрите використання інформації досить важко. Потрібний новий спеціалізований підхід, який би забезпечив баланс між відкритістю інформації й безконтрольністю її поширення. Колективне збирання й аналіз інформації «Інтелект Відкритих Джерел» (Open Source Intelligence – OS-INT) [17] є одним з напрямків концепції відкритих джерел Open Source [16], який зі збільшенням кількості користувачів Web і зростанням їхньої кваліфікації стає усе більш значущим джерелом знань.

До Open Source Intelligence відносяться соціальні мережі, Wiki-ресурси, блоги тощо, які створюють та поповнюють мільйони користувачів Web. Інформація в таких джерелах має певну структурованість, але вона досить часто буває суперечливою й неповною.

OS-INT базується на таких принципах спільної праці, як експертна оцінка, практика взаємного оцінювання експертами власної роботи, превалювання незалежної репутації над адміністративним контролем, вільний обіг продуктів і гнучкі рівні залежності й відповідальності. Оцінки тих спеціалістів, що мають високу репутацію, мають більшу вагу. Знання здобуваються в процесі уточнень і досягнення консенсусу.

OS-INT складається з великої кількості незалежних проектів, які мають різну історію, різну технічну й соціальну стратегію для реалізації принципів спільного використання відкритих джерел.

В OS-INT є вагоме теоретичне й технологічне підґрунтя. Багато з базових Інтернет-технологій були створені для сприяння вільному й доступному обміну інформацією між експертами, у процесі якого інформація не тільки ефективно поширюється, а й спільно оцінюється. Найпростіші платформи OS-INT – поштові списки розсилання (e-mail lists) – з'явилися в середині 70-х рр. минулого століття. У 80-ті дошки FidoNet та Usenet надавали користувачам деякі можливості OS-INT, але вони були досить складними у використанні. У 90-х рр. багато з цих платформ було трансформовано у зв'язку з появою World Wide Web.

Open Source Intelligence – ще досить молода галузь, що потребує рішення таких проблем, як необхідність масштабування та економічної підтримки. Порівняно з традиційними моделями виробництва й видавництва, проекти OS-INT здебільшого беруть участь за межами традиційної товарно-грошової економіки. Вкладники проектів загалом

не мотивовані безпосередньою фінансовою вигодою. Однак не всі ресурси можуть бути забезпечені безкоштовно, тому повинні бути винайдені нові моделі фінансування таких проектів. Ймовірно, що проекти OS-INT, з економічного погляду, будуть розвиватися в гібрид, який передбачатиме прямі доходи (наприклад, передплата, реклама), добровільні пожертвування й волонтерські зусилля. Співвідношення цих різних елементів буде змінюватися від проекту до проекту.

Сьогодні одним з найпоширеніших напрямів OS-INT є технологія Wiki [35]. Саме ця технологія нині активно використовується для розробки систем менеджменту знань у різних сферах, приміром, при створенні Вікіпедії [36]. Програмне забезпечення й контент, що міститься на Wiki-сторінках, найчастіше використовують ліцензію вільного поширення документації – GNU Free Documentation License (GFDL), що гарантує сумісність ліцензії з наявною сукупністю текстів GFDL. Це дає можливість мати один Wiki зі сторінками, що мають різну ліцензію.

1.2. Основні елементи Wiki

Wiki-середовище (від «Wiki Wiki», що гавайською мовою означає «швидко») – це гіпертекстове середовище (зазвичай – Web-сайт), яке дозволяє користувачам відносно легко створювати й модифікувати його контент. Контент такого сайту, який складається з окремих Wiki-сторінок та зв'язків між ними, називають *Wiki-ресурсом*. Його створення та редагування є колективним процесом. Кожен користувач, який бачить помилку чи недолік, може негайно її виправити чи додати інформацію, якої бракує. Оскільки процес перегляду й уточнення є публічним і безперервним, то немає принципової різниці між попередніми й фінальними версіями інформації. Використання принципу експертного перегляду відкритих джерел дає змогу Wiki-ресурсу поступово зростати як за кількістю статей, так і за їх глибиною та якістю внаслідок колективного внеску тих користувачів, що є експертами в окремих питаннях.

Під *Wiki-технологією* зазвичай розуміють таку технологію побудови Wiki-ресурсу, яка дає змогу відвідувачам брати участь у редагуванні його вмісту – виправляти помилки, додавати нові матеріали, не використовуючи спеціальні програми, не реєструючись на сайті й не вивчаючи мову HTML [35].

Кожна окрема сторінка Wiki-ресурсу називається **Wiki-сторінка** (*Wiki page*). Створення й редагування Wiki-сторінок не потребує від користувачів знання мови HTML: механізм Wiki надає можливість написання документів за допомогою простої мови розмітки й Web-браузеру. Всі Wiki-сторінки мають унікальні назви, на які можна посилатися з інших статей, і вміст.

Інформація, представлена у Wiki, має нелінійну навігаційну структуру: кожна сторінка, зазвичай, містить велику кількість гіперпосилань на інші сторінки.

У 2006 році термін Wiki добавлено до онлайнного Оксфордського Словника Англійської мови (OxfordEnglish Dictionary, OED). Він означає модель сайтів, контент яких може змінювати сам користувач.

Скориставшись спеціальним посиланням або кнопкою, відвідувач Wiki-сайту може відредагувати текст сторінки, яку він розглядає, безпосередньо всередині свого web-браузера й зберегти змінений варіант на сервері. Усі виправлення статей на Wiki-сайтах зберігаються в базі даних. Кожен користувач будь-коли може зробити запит щодо будь-якого попереднього варіанта статті чи проглянути різницю між будь-якими двома минулими варіантами статей, що дає змогу легко й швидко відстежувати випадки вандалізму.

Формат Wiki-сторінок (Wiki-text) – це спрощена мова розмітки, що використовується для того, щоб виділити в тексті різні структурні й візуальні елементи або вказати на них. Кожна Wiki-система має власний стиль і синтаксис залежно від реалізації. У багатьох реалізаціях Wiki гіперпосилання показується таким, яким воно є насправді, на відміну від HTML, де невидиме гіперпосилання може мати довільний видимий текст або зображення.

Для створення та підтримки Wiki-ресурсу необхідно спеціальне програмне забезпечення (ПЗ) – *Wiki-двигун*, який забезпечує роботу відповідного Web-сайту. Двигун Wiki-сайту робить запис змін так, щоб у будь-який час сторінку можна було б повернути до кожного з її попередніх станів. Wiki спроектовані так, щоб помилку було легше виправити, а не щоб її було важче зробити.

Wiki-система може також містити різні інструментальні засоби для простого контролювання стану Wiki, що постійно змінюється, а також для обговорювання й розв'язання проблем, що виникають через різні погляди на зміст Wiki-сторінок. Контент Wiki може містити неправдиву й некоректну інформацію, що вводиться користувачами навмисно або випадково. Більшість Wiki відкрита для широкої публіки навіть без необхідності реєстрування. Значна частина редагувань може бути зроблена в режимі реального часу й миттєво з'являтися на сторінці. Більшість Wiki-двигунів реалізує хронологію ревізій, що показує попередні версії сторінки, та пошук розбіжностей між версіями сторінки, яка висвічує рядки, що є різними в них.

Деякі останні версії Wiki-двигунів використовують різні методи: вони дають змогу редагувати методом WYSIWYG (що видно, те і є), що реалізується за допомогою JavaScript або ActiveX, які перетворюють графічно введені команди форматування, наприклад, шрифти «жирний»

(«грубий») і «курсив» на відповідні теги HTML. У таких реалізаціях HTML-розмітка Wiki-сторінки генерується й постачається на сервер автоматично, таким чином, користувач не бачить жодних технічних деталей. Споживачі, які не мають необхідного плагіну, можуть редагувати сторінку, набираючи безпосередньо код HTML. Останні Wiki-двигуни генерують синтаксис Wiki замість HTML.

Хоча протягом років стандартом де-факто був синтаксис першого WikiWikiWeb, нині команди форматування змінюються залежно від Wiki-двигуна. Простий Wiki дає змогу здійснювати тільки основне форматування тексту, тоді як більш складні системи забезпечують підтримку таблиць, зображень, формул і навіть інтерактивні елементи типу голосувань та ігор. Загального стандарту для розмітки Wiki на сьогодні немає.

Більшість Wiki забезпечує пошук за заголовками, а деякі – повнотекстовий пошук (full-text search) з усієї бази. Сфера пошуку залежить від використовуваної Wiki-двигуном бази даних, реалізації індексованого доступу до бази даних, що необхідно для високошвидкісного пошуку на великих Wiki. Для пошуку в множині Wiki створений пошуковий механізм MetaWiki, який здійснює пошук за ключовими словами (keyword-based).

1.3. Історія створення Wiki

Спочатку Wiki-системи являли собою просто сайти, сторінки яких міг редагувати будь-хто безпосередньо з Web. У сучасному варіанті – це система для збирання й структурування інформації, що характеризується такими ознаками:

- багато авторів, якими, зазвичай, можуть бути всі користувачі Wiki-ресурсу.
- багатокористувацький режим роботи – усе редагування здійснюється через web-інтерфейс, є центральний сервер (чи кластер), що зберігає весь масив даних.
- можливість багаторазово редагувати текст за допомогою самого Wiki-середовища без застосування спеціалізованого ПЗ з боку користувача.
- поява змін одразу після їх внесення.
- поділ інформації на однозначно ідентифіковані сторінки, кожна з яких має власну назву.
- нескладна мова розмітки, що дає змогу легко відокремити контент від оформлення.
- облік змін (версій) тексту й можливість повернення до попередньої версії.

Wiki-ресурси являють собою перспективний шлях розвитку приватних і публічних баз знань. Оригінальна система Wiki була винайдена в 1995 р. У. Канінгемом [13]. для Web-вузла Pattern Languages Community, щоб спростити спільне створення й документування програмного забезпечення. Всесвітньо відомий приклад застосування технології Wiki – це Вікіпедія, найбільша з безкоштовних онлайн-енциклопедій. Простота й зручність у підтримці колективної творчості, притаманні технології Wiki, дали змогу засновникам енциклопедичного проекту Nupedia – Д. Уельсу й Л. Сенгері використати її як основу електронної енциклопедії. Унаслідок цього в січні 2001 року було створено найпопулярнішу Інтернет-енциклопедію Вікіпедія ("Wikipedia"). Спочатку вона працювала на базі програмного забезпечення UseMod, але потім перейшла на власну відкриту базу кодів, яку тепер використовують і багато інших Wiki.

WIKIPEDIA



Рис.1.1. Версії Вікіпедії різними мовами.

Сьогодні найбільшим Wiki-ресурсом є англійська версія Вікіпедії [36], яка нараховує сьогодні 1,5 млн. статей 100 мовами. Для порівняння – Encyclopedia Britannica 2007 року містить близько 100 тис. статей. Невдоволення повільністю створення й виходу паперових енциклопедій було очевидним уже давно, але створення оперативних енциклопедій потребує багато людей і є економічно невиправданим. У Вікіпедії платою авторам стають не гроші, а соціальний статус. Особливість цієї енциклопедії в тому, що всі відвідувачі сайту wikipedia.org. можуть її редагувати й доповнювати власними статтями.

За коректністю наведеної в енциклопедії інформації стежать тисячі осіб, тому окремі вандали не можуть зашкодити репутації порталу, і його наповнення залишається на досить високому рівні.

Деякі інші великі Wiki – це WikiWikiWeb, Wikitravel, World66. Також є WikiNodes – вузлові Wiki-сторінки, які містять описи пов'язаних за темою Wiki. Інші цікаві проекти, що базуються на технології Wiki – словники Wiktionary, колекції книг Wikibooks, цитат Wikiquotes, документів Wikiource, новин Wikinews і матеріалів для самоосвіти Wikiversity.

Крім великих Wiki-ресурсів, існує велика кількість приватних й вузько спеціалізованих Wiki-спільнот – приміром, у межах організації, де вони використовуються для спільної роботи та ведення внутрішньої документації, або територіальних об'єктів (населених пунктів, районів). Такі Wiki зростають і розвиваються повільніше за загальнопоширені, такі, як Wikipedia. Чим більше користувачів має Wiki-спільнота, тим вище темп її зростання й удосконалення.

На початку використання Wiki поширювався на підприємствах як спільне програмне забезпечення для комунікації в рамках проектів, intranet і обміну документацією. У грудні 2002 р. Socialtext випустив перше комерційне рішення Wiki з відкритим кодом. Після цього відкритий код програмного забезпечення Wiki швидко поширювався, легко завантажувався й встановлювався. Сьогодні для деяких компаній Wiki стало єдиним спільним програмним забезпеченням і повністю замінило статичний intranet.

1.4. Характерні риси Wiki-ресурсів

Використання технології Wiki забезпечує:

- іменування та ідентифікацію – ідентифікатор Wiki-сторінки забезпечує доступ до неї;
- використання шаблонів для подання структурованих даних;
- автоматичне створення різноманітних класифікацій через механізм категорій;
- цілісність посилань – користувач Wiki може відстежити усі посилання з поточної сторінки та всі посилання на неї з інших сторінок Wiki-ресурсу.

Основні принципи розробки Wiki-ресурсів, які регулюють розробку Wiki-застосувань [35] :

- *Відкритість* (Open): якщо користувач вважає, що Wiki-сторінка неповна або погано організована, він може редагувати її так, як він вважає за потрібне;
- *Поетапність* (Incremental): сторінки можуть містити посилання на інші сторінки, зокрема й на сторінки, які ще не створено;

- *Органічність* (Organic): структура й контент Wiki-ресурсу відкриті для редагування й еволюції.
- *Простота* (Mundane): невелика кількість текстових конвенцій має забезпечити доступ до найбільш корисної (але обмеженої) розмітки сторінки.
- *Універсальність* (Universal): механізми редагування й організації Wiki-сторінок такі самі, як і ті, що використовуються для їх написання, тому будь-який автор автоматично стає редактором і організатором.
- *Відкритість* (Overt): відформатована і опублікована інформація містить дані, що потрібні для її відтворення (наприклад, розміщення на сторінці.)
- *Уніфікація* (Unified): імена сторінок отримуються з плаского простору (flat space), тому жодного додаткового контексту не потрібно, щоб інтерпретувати їх.
- *Точність* (Precise): сторінки отримують імена з достатньою точністю, щоб уникнути конфліктів між більшістю імен, як правило, шляхом формування іменників.
- *Толерантність* (Tolerant): інтерпретована (навіть якщо небажана) поведінка має переваги порівняно з повідомленням про помилки.
- *Спостережність* (Observable): за активністю в межах Wiki-ресурсу може спостерігати будь-який відвідувач сайту.
- *Конвергентність* (Convergent): дублювання може видалятися шляхом пошуку й за посиланнями на аналогічний або пов'язаний контент.

Wiki мають багато спільних рис і системами керування контентом (content management systems – CMS), але між ними існують важливі відмінності. Порівнюючи CMS з технологією Wiki, необхідно зазначити такі базові особливості:

- назва Wiki-сторінки одночасно є гіперпосиланням;
- Wiki-сторінки можуть створюватися або редагуватися в будь-який час кожним користувачем з відповідними повноваженнями (з деякими обмеженнями для захищених статей);
- Wiki-сторінки доступні для редагування у web-браузері;
- з кожної Wiki-сторінки можна отримати доступ до перегляду й редагування хронології /версій текстових варіантів сторінки, що підтримує пошук розбіжностей і видобування попередніх версій;
- з кожної Wiki-сторінки можна отримати доступ до сторінки обговорення, що стосується цієї статті.

1.5. Технологічні платформа створення Wiki-ресурсів

Платформа Wiki реалізувала одну з базових концепцій Т.Бернерса Лі – автора проекту Semantic Web, надавши користувачам можливість бачити відправний код і вільно редагувати контент сторінок, які вони

бачать. Зміни в контенті, внесені користувачами, набувають чинності відразу, без будь-якої перевірки й перегляду навіть автором чи модератором, але обов'язково підтримуються функції, що дають змогу користувачам переглядати зміни й, за необхідності, повертати сторінки до попередніх версій.

На сьогодні про Wiki потрібно говорити саме як про технологію, а не про програмний продукт чи систему, оскільки є багато реалізацій Wiki. Загальною рисою всіх реалізацій є швидкість уведення контенту й простота керування ним. Багато редакторів контенту Wiki мають функції WYSIWYG, а замість досить складної HTML-розмітки використовують спрощену Wiki-розмітку, яка, на жаль, не зовсім збігається в різних реалізаціях. Крім оформлення тексту, редактори Wiki надають засоби введення математичних формул, символів інших кодувань, а також інші корисні функції. Уведення контенту прискорюється також за рахунок автоматизації деяких задач і використання шаблонів, що дають змогу швидко перетворити вид документа.

Найбільш поширеними є серверні реалізації системи Wiki. Це означає, що функції редагування, відображення й керування забезпечуються на сервері за допомогою програмного забезпечення Wiki, яке відтворює зміст сторінки у форматі HTML для відображення у Web-браузері.

З боку клієнта Wiki-системи необхідно тільки, щоб сервер «обслуговував» Wiki за допомогою протоколу HTTP. У цьому типі Wiki-систем усе виконання складається з необхідності конвертування первинного Wiki-тексту в екранне відформатоване зображення сторінки у web-браузері клієнта. Аналогічно, інструментальні засоби редагування та інших функціональних можливостей містяться в програмі браузера.

Wiki-система, що реалізується з боку клієнта, відповідає HTML-коду, у якому сторінка містить команди для виконання, які інтерпретуються web-браузером. Такі системи можуть бути не на багато більшими, ніж код плагіну для традиційних браузерів.

Популярність технологій Wiki сприяла появі великої кількості реалізацій практично для всіх можливих платформ і конфігурацій ПЗ. Крім того, різні реалізації Wiki неоднаково розширюють основні можливості технології, додаючи, наприклад, убудовані графічні редактори чи сервіси WAP.

У Таблиці 1. представлено найпоширеніші Wiki-системи. Більш детально типи та приклади семантичних Wiki-систем розглянуто в [42].

Таблиця 1. Найпоширеніші Wiki-системи

Назва	Посилання	Платформа /Мова
AwkiAwki	http://awkiawki.bogosoftware.com/	Awk
CitiWiki	http://wiki.cs.cityu.edu.hk/citiwiki	PHP
CLiki	http://www.cliki.net/	CommonLisp
DidiWiki	http://didiwiki.org/	C
DotNetWiki	http://www.dotnetwiki.org/	C#
EddiesWiki	http://c2.com/cgi-bin/wiki?EddiesWiki	C++
FlexWiki	http://www.flexwiki.com/	C#
FreeWiki	http://sourceforge.net/projects/freewiki/	Java
JavaWiki	http://c2.com/cgi-bin/wiki?JavaWiki	Java
JspWiki	http://www.jspwiki.org	Java
MediaWiki	http://www.mediawiki.org/	PHP
NoodleWiki	http://flangy.com/dev/asp/noodle/	ASP
OpenWiki	http://www.OpenWiki.com	ASP
PmWiki	http://www.pmichaud.com/pmwiki	PHP
TiddlyWiki	http://www.tiddlywiki.com	JavaScript
WackoWiki	http://wackowiki.org/WackoWiki	PHP
WebMacro Wiki	http://www.webmacro.org/WebMacro Wiki	Java
WikiAsp		ASP
WikiCpp	http://wikicpp.sourceforge.net/	C++
WikicWeb	http://c2.com/cgi-bin/wiki?WikicWeb	C

На сьогодні є багато різноманітних реалізацій Wiki для різних платформ.

TikiWiki має всі необхідні функції, реалізується за допомогою PHP і використовує СКБД MySQL. TikiWiki підтримує блоги, передачу даних мобільними пристроями й управління голосом. За допомогою спеціального Java-скрипту можна створювати прості векторні малюнки. Система підтримує поштові списки розсилання, публікацію новин у форматі RSS, містить чат. Рядові користувачі системи володіють значним особистим простором, що дає їм змогу не тільки вибирати персональні налаштування для перегляду сторінок, а й надавати особистий блокнот для записів і календар для нагадувань.

Систему **MediaWiki** використовує проект Вікіпедія. MediaWiki побудована на основі PHP і MySQL. MediaWiki відрізняється якісним та зручним редактором контенту (рис. 1.2).

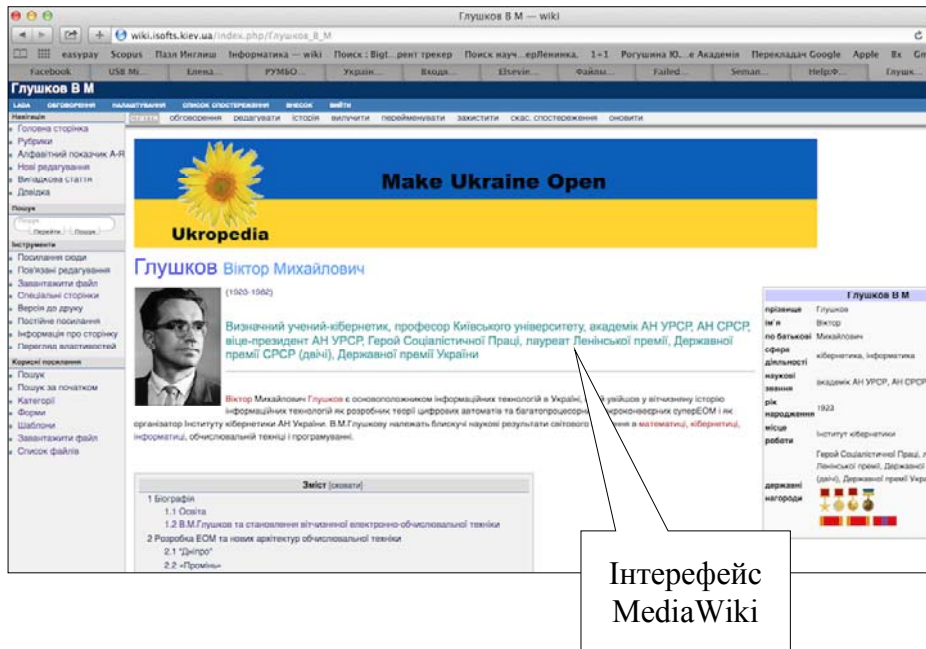


Рис. 1.2. Інтерфейс сторінки на базі MediaWiki

Реалізація Wiki **UseModWiki** написана на Pearl. UseModWiki не використовує бази даних, а тому може бути розміщена на серверах, що не пропонують таку послугу. Цікавою функцією UseModWiki є наявність «під-сторінок», що можуть функціонувати як самостійні Wiki.

FlexWiki написана на C# і призначена для роботи в середовищі Ms Windows +NET. FlexWiki може обійтися без баз даних, а може використовувати SQL Server чи інші СКБД, доступні через MSDE. Цікавий інструмент FlexWiki – FlexWikiPad – редактор контенту, що підтримує підсвічування синтаксису й автоматичне завершення введення.

SushiWiki також реалізована на C# і орієнтована на SQL Server 2000 та інші СКБД, доступні через MSDE. Якщо в інших редакторах контенту Wiki можливості WYSIWYG обмежені чи необхідні спеціальні модулі для їх під'єднання, то редактор SushiWiki дає можливість редагувати сторінку, не відриваючись від її зовнішнього вигляду.

1.6. Вандалізм у Wiki-середовищі

Відкрита філософія більшості Wiki – давати змогу будь-якому користувачеві редагувати вміст сторінок. Але наміри таких редакторів не завжди добрі. Більшість публічних Wiki уникає обов'язкових реєстраційних процедур, але багато Wiki-двигунів (приміром,

MediaWiki) мають певні методи для обмеження доступу до написання тексту. Приміром, можна забороняти редагування певним індивідуальним користувачам, блокуючи конкретні IP-адреси або імена користувачів, які відомі своїми шкідливими діями. У разі надзвичайних ситуацій деякі Wiki мають можливість змінювати режим баз даних так, що вони стають доступними тільки для читання.

Загальноживаний спосіб захисту від «вандалів» у Wiki-середовищі – дозволити їм зіпсувати стільки сторінок, скільки вони бажають, знаючи, що ці виправлення легко скасувати. Але якщо безсумнівну шкоду, завдану «вандалом», можна швидко й легко виправити, то більш проблематичними є “дрібні”, менш помітні помилки, наприклад, зміни в датах, прізвищах та назвах.

Багато Wiki мають змогу захистити певні сторінки від редагування: такі сторінки можуть редагувати тільки адміністратори, які здатні встановлювати й знімати такий захист. Але така практика суперечить основній філософії Wiki, а тому її, як правило, уникають. Наприклад, англійська Вікіпедія водночас має близько тридцяти захищених сторінок з понад трьохсот тисяч.

1.7. Теоретичний базис Wiki

Незважаючи на простоту використання, технологія Wiki ґрунтується на серйозному теоретичному базисі й використовує здобутки з інших галузей знань. Термін «наука Wiki» (Wiki Science) означає набір відомостей, поглядів і рекомендацій щодо створення й розвитку Wiki-застосовань. Наука Wiki використовує результати, отримані в рамках наступних наукових напрямків:

- теорія хаосу (Chaos theory);
- теорія емерджентної поведінки (Emergent behaviour);
- штучне життя (Artificial life);
- когнітивна нейробіологія (Cognitive neuroscience);
- загальне управління якістю (Total Quality Management).

Технологія Wiki може застосовуватися як основа для групової взаємодії й менеджменту знань [35].

Технології Wiki є надзвичайно ефективним інструментом, що спонукує активно працювати з інформацією замість того, щоб пасивно її сприймати. Користувачі можуть не тільки читати й аналізувати статті, а й брати участь у їх написанні. Тому ці технології часто застосовуються в освітніх проектах. На базі Wiki розробляється цікава наукова теорія процесу здобуття знань, що базується на теорії «емерджентної поведінки» (Emergent behaviour). Ця нова теорія освіти припускає, що використання Wiki спрощує здобуття знань індивідуумами. Процес здобуття знань розглядається як шлях між двома точками – точкою, що

характеризує знання, які індивідуум уже має, і точкою нових знань. Зазвичай, перехід з однієї точки до іншої, нової здійснюється з докладанням великих зусиль і траєкторія переходу має форму кривої. Скорочення шляху між цими точками відбувається завдяки тому, що Wiki має певну структуру, складену за певною логікою, а групова творчість поступово формує найчіткіші й найзрозуміліші вирази й образи, найбільш прийнятні для засвоєння. Отже, траєкторія переходу між точками набуває вигляду прямої, відстань між точками зменшується, а кількість необхідних зусиль зменшується.

1.8. Еволюція Wiki-ресурсів

«Ефект брунькування» (budding effect) – термін, що використовується для опису того процесу, коли один Wiki-ресурс породжує інший (за аналогією до рослин). Великий батьківський Wiki-ресурс може «народити» кілька менших дочірніх, створюючи своєрідне дерево.

Wiki може брунькуватися як повільно, так і швидко. Бруньки можуть виникати як результат зіткнення на батьківському Wiki або ж можуть поступово формуватися з членів батьківського Wiki, що вважають себе належними й до батька, й до дитини. Є багато технологій, що підтримують процес брунькування, починаючи з близького гіперпосилання й завершуючи сторінкою кластера з простору найменувань (name spaces).

Еволюція статей Wiki – це процес, коли стаття Wiki розвивається, починаючи з кореня (*stub*) або спеціально виділеного місця для майбутньої статті з невеликою кількістю інформації. Такий корінь створюється тим користувачем Wiki, який хоче ідентифікувати потребу в Wiki-сторінці, але не має часу або знань для створення її повної версії. З часом користувачі додають, розширюють цю статтю, поки вона не стане актуальною. Якщо корінь оригіналу не створити, то й інші користувачі не будуть зацікавлені у написанні кінцевої версії статті. Це є співробітництвом серед нестачі, де невелике початкове зусилля змушує інших виконувати певні дії й створювати кінцевий продукт, цікавий для всіх, навіть якщо кожен з учасників розробки не має всіх необхідних складників.

2. Україномовні Wiki-ресурси та енциклопедії

2.1. Українська Вікіпедія

Українська Вікіпедія – це україномовний розділ Вікіпедії – багатомовного Інтернет-проекту зі створення Wiki-енциклопедії, яку може редагувати кожний охочий користувач Web. На сьогодні кількість статей української Вікіпедії становить 572 676 (станом на 04. 06. 2015 р., 13:27). За цим показником вона посідає 16-е місце серед усіх мовних розділів, 11-е місце серед європейських вікіпедій і 3-є місце серед вікіпедій слов'янськими мовами. Доступ до цієї сторінки можна отримати, набравши в пошуковому запиті до Google «Українська Вікіпедія».

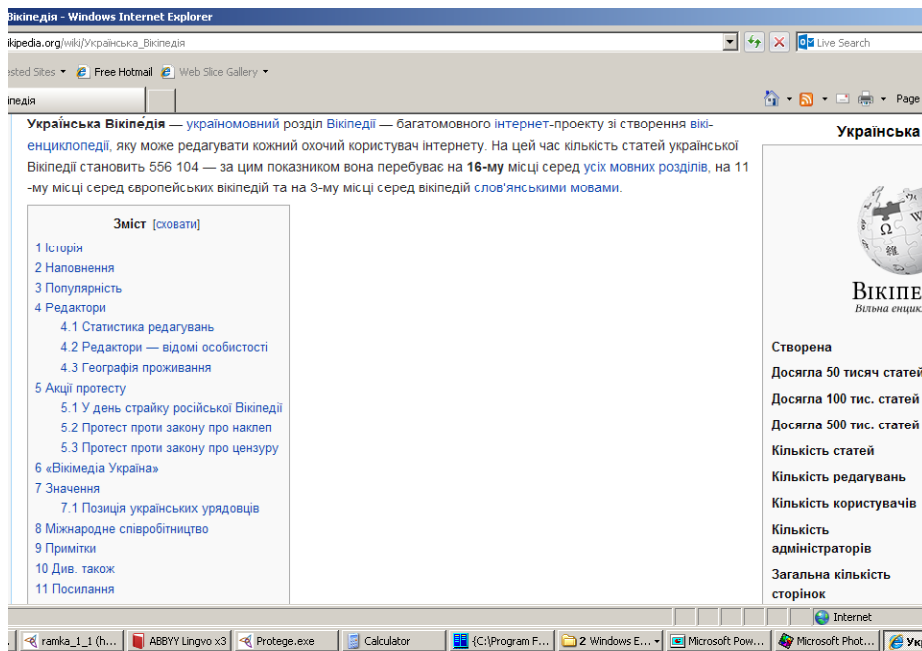


Рис.2.1. Головна сторінка української Вікіпедії

Ця стаття Вікіпедії містить відомості про історію, контент та інструментарій українського сегмента Вікіпедії.

Перші зафіксовані спроби створення статті в українській Вікіпедії датуються 26 січня 2004. Перша стаття – «Атом», створена 30 січня 2004 року користувачем 61.125.212.32 (пізніше зареєстрований як Yuri koval) з Японії.

Українська Вікіпедія (uk.wikipedia.org/wiki) – україномовний розділ Вікіпедії – багатомовного Інтернет-проекту зі створення Wiki-енциклопедії.

На даний кількість статей української Вікіпедії становить 707 551 – за цим показником вона перебуває на 16-му місці серед усіх мовних розділів, на

11-му місці серед європейських вікіпедій та на 3-му місці серед вікіпедій слов'янськими мовами.

Українська Вікіпедія станом на 19 липня 2017 року має 365 719 зареєстрованих користувачів, 39 адміністраторів [37]. Загальна кількість сторінок в українській Вікіпедії – 2 142 627, редагувань – 20 917 403, завантажених файлів – 87 903. Глибина (рівень розвитку мовного розділу) української Вікіпедії середня і становить 40.2 пунктів.

Станом на 19 липня 2017 року за кількістю статей Українська Вікіпедія посідає 16-те місце серед мовних розділів (між Вікіпедіями китайською та перською мовами) та третє місце серед слов'янських вікіпедій, маючи понад 700 тисяч статей. Для порівняння, англійська Вікіпедія налічує понад 5 млн. статей, німецька – понад 2 млн. статей, нідерландська, французька, італійська, іспанська, польська, в'єтнамська і японська – понад 1 млн. статей, португальська і китайська – понад 900 тисяч. За загальним розміром Українська Вікіпедія знаходиться на 12-му місці у світі.

Українська Вікіпедія посідає 5 місце у списку розділів Вікіпедії за розміром 1000 статей, які повинні бути в кожній Вікіпедії. Також Українська Вікіпедія займає 9 місце у списку розділів Вікіпедії за розміром обов'язкових 10000 статей.

Якість енциклопедії в цілому за суто статистичним аналізом, що враховує лише частку статей серед загальної кількості сторінок та середню кількість редагувань на статтю, можна оцінити лише приблизно.

Станом на 8 березня 2015 критерій «глибини» (англ. Depth), що враховує середню кількість редагувань на статтю та частку саме статей серед усіх сторінок Вікіпедії для українського розділу, дорівнює 40. За цим показником вона випереджає інші великі Вікіпедії (які мають більш як 100 тис. статей), зокрема норвезьку (36), каталонську (31), шведську (5), литовську (23), есперанто (17), польську (25), словацьку (19), нідерландську (13). Серед великих Вікіпедій найбільші показники глибини мають: Вікіпедія англійською (983), сербохорватською (791), івритом – 261, турецькою – 221, арабською – 188, перською – 196, французькою – 218.

Українська Вікіпедія містить багато статей, які у первісному вигляді було скопійовано з Енциклопедії українознавства, Української радянської енциклопедії, довідника з історії України, ГЕС, МГЕ (разом – понад 15 тис.), урядових та інших сторінок. У той же час, деякі з цих статей були пізніше перероблені та розвинуті редакторами Вікіпедії, які також створили чи переробили чимало статей у повністю самостійні роботи. Українська Вікіпедія також містить найбільшу кількість статей про населені пункти України.

Найкращими статтями української Вікіпедії є вибрані статті й вибрані списки, обрані користувачами як ті, що відповідають високому рівню якості, викладу матеріалу, досконалості, нейтральності, об'єктивності та відповідності до наукових джерел.

Хронологія подій в українській Вікіпедії:

- 26 лютого 2004 – перша реклама на українському форумі Альянсу «Майдан» (maidan.org.ua).
- 4 квітня 2004 було офіційно активовано українську частину Вікіпедії. Кількість статей на той момент – 1000.
- 16 листопада 2004 на Головній сторінці поставлено першу вибрану статтю та вперше запроваджено блок новин
- 10 жовтня 2007 було зроблено мільйонне редагування.
- 17 лютого 2008 зареєстровано 10 000-ого користувача: Purbo T.
- 28 березня 2008 було написано 100 000-ну статтю, на честь чого було розповсюджено прес-реліз.
- 7 квітня 2010 було написано 200 000-ну статтю, з нагоди чого було організовано прес-конференцію і розповсюджено прес-реліз.
- 12 лютого 2011 року за кількістю статей українська Вікіпедія наздогнала фінську.
- 7 липня 2011 було написано 300 000-ну статтю, з нагоди чого було розповсюджено прес-реліз
- 31 серпня 2011 українська Вікіпедія за кількістю статей перевершила Норвезьку, посівши 14 місце серед всіх розділів Вікіпедії.
- 26 жовтня 2011 загальна кількість сторінок в українській Вікіпедії перевищила 1 мільйон.
- 19 грудня 2011 число активних користувачів уперше перетнуло позначку 2000.
- 21 лютого 2012 українська Вікіпедія за кількістю статей перевершила каталонську, посівши 13 місце серед всіх розділів Вікіпедії.
- 14 березня 2012 року Українська Вікіпедія за кількістю статей поступилася В'єтнамській і опустилася на 14-е місце у світі.
- 22 липня 2012 кількість редагувань досягла 10 000 000.
- 20 вересня 2012 було написано 400 000-ну статтю.
- 27 квітня 2014 року було проведено Вікіфлешмоб, завдяки цьому 27 квітня було створено аж 940 статей – а це значить, що цього дня українці створили статей більше, ніж в англійській вікіпедії. Вплив цього заходу був такий сильний, що українська Вікіпедія ввійшла до десятки вікіпедій, що найшвидше розвиваються.
- 12 травня 2014 року була написана 500 000-на стаття.
- 13 вересня 2014 року кількість редагувань досягла 15 000 000.
- 13 листопада 2015 було написано 600 000-ну статтю.
- 4 червня 2017 було написано 700 000-ну статтю.

Вікіпедія базується на наступних принципах, які дозволяють їй залишатися енциклопедією, що більш детально розглянуті в [62]:

- редагуючи Вікіпедію, слід додавати виключно ту інформацію, яка міститься у достовірних джерелах, а не у пропагандистських або рекламних матеріалах або в оригінальних дослідженнях;

- слід дотримуватися нейтральної точки зору, що означає висвітлення різних підходів та думок, які існують щодо суперечливих тем, з обов'язковим посиланням на використані джерела;
- Матеріали Вікіпедії є вільними для використання, а тому статті не є чияюсь власністю і кожен має право їх змінювати та використовувати (тому у Вікіпедії діє заборона на використання матеріалів, які порушують авторське право);
- у Вікіпедії діють правила ввічливої поведінки, спрямовані на досягнення консенсусу щодо суперечливих питань, які забороняють погрози та образи.

Для того, аби написати гарну енциклопедичну статтю, слід використовувати факти, твердження, ідеї, теорії, точки зору, гіпотези та аргументи, опубліковані в авторитетних джерелах.

Верифікованість є ключовою умовою для повноти та надійності Вікіпедії та однією з її базових політик, тому дописувачі повинні використовувати та покладатися тільки на джерела, що варті довіри – так, щоб можна було перевірити наведені у статті факти за джерелом. Аналогом верифікації є примітки (посилання на літературні джерела) в наукових статтях. У створенні або доповненні статей для Вікіпедії потрібно базуватися на відкритих авторитетних незалежних джерелах, які можна легко перевірити. Авторитетність окремих друкованих та електронних видань є відкритим питанням. Також не бажано використовувати посилання на архівні джерела, оскільки перевірити їх достовірність важче, аніж видань, доступних у бібліотеках. Інформація, яка стосується добре вивчених та загальновідомих фактів, зазвичай не потребує додаткової верифікації, проте існує низка випадків, коли постає необхідність перевірки певних тверджень у статтях:

- існує конфлікт інтересів різних редакторів Вікіпедії щодо статті;
- у тексті статті знайдені фактичні помилки, а тому слід перевірити достовірність інформації в цілому;
- предмет статті належить до галузі, де можливі часті помилки або існують різні точки зору щодо нього;
- певне твердження є неймовірним чи сумнівним, або ключовим чи занадто розмитим, що потребує додаткового підтвердження авторитетними джерелами.

Розмір статей в Вікіпедії значним чином визначається специфікою того поняття, яке вона описує, але існує кілька загальних правил, яких за можливістю слід дотримуватися. Оптимальним розміром для статті є розмір у 30–50 кБ, якщо стаття більша, то можливе розділення її на кілька статей або винесення її головних розділів в окремі статті (якщо це доцільно з обсягу розділу).

Статті у Вікіпедії поділяють на стаби (до 10 кБ), середні статті (10–30 кБ) та великі статті (більше 30 кБ). Вважається, що текст з 6–10 тисяч слів є оптимальним для сприйняття читачем.

Створюючи нову статтю, слід хоча б мінімально розкрити у ній суть її предмета (це мінімум 2–3 кБ). Не варто також створювати у статті надто

дрібні за тематичним розкриттям розділи з одного–двох речень, а краще об'єднувати їх в один загальний (приміром, у статті про організацію не варто створювати підрозділи, присвячені різним хронологічним періодам її роботи, а краще все написати в одному розділі – «Історія»). Якщо з часом обсяг інформації збільшиться, тоді слід подумати про створення кількох підрозділів.

При цьому слід зважити на те, що розділи, які складаються з кількох абзаців, не слід переносити в окрему статтю, а лише ті, які значно розрослися та є цілісним і відокремленим текстом та висвітлюють одну тему або один з аспектів головної статті, не мають проблем з формулюванням та самостійним значенням теми.

У Вікіпедії слід поважати авторське право, яке поширюється на всі письмові, фото-, відео- та візуальні твори (у більшості країн світу терміном на 70 років після смерті їх автора).

Слід відмітити, що у Вікіпедії забороняється дослівне цитування фрагментів тексту понад 1–2 речення без посилань на авторство тексту. Пишучи статтю, Ви маєте переказати своїми словами зміст джерел.

Посилання на джерела є однією з вимог у Вікіпедії, оскільки це дає можливість підтвердити значимість предмета статті та перевірити достовірність інформації за наведеним джерелом. Крім того, посилання на джерела необхідні для подальшого ознайомлення з темою; перевірки неоднозначних тез, особливо у випадках вандалізму; мінімізації суперечок між редакторами Вікіпедії; підвищення загального рівня довіри до Вікіпедії та уникнення звинувачень у плагіаті та крадіжках інтелектуальної власності. Посилання на джерела використовуються у трьох розділах статті: Примітки, Джерела (та або Бібліографія), а також Посилання. Оформлення посилань на джерела необхідно робити згідно з правилами бібліографічного опису.

В українській Вікіпедії бажано посилатися на джерела українською мовою. За умови відсутності таких джерел або їх недостатньої авторитетності чи повноті, доцільно посилатися на джерела іншими мовами, вказуючи в кінці бібліографічного опису джерела (книги, статті з журналу або Інтернет-сторінки) мову, якою воно написано за допомогою шаблонів вказівки мови для посилань.

2.2. Українські Wiki-ресурси

Крім Вікіпедії, на сьогодні створено та активно використовується ціла низка інших корисних україномовних Wiki-проектів.

Вікісловник (<http://uk.wiktionary.org>) – проект фонду Вікімедіа, в якому подаються граматичні властивості, тлумачення та переклади слів різними мовами. Із 30 тисяч слів українського Вікісловника більше 23 тисяч – це українські слова, решта – слова іноземними мовами, зокрема англійською (близько 2300), японською (1900), російською (850), німецькою (500), білоруською (370), грецькою (320) та багатьма іншими, в цих статтях подається переклад слів українською. Український Вікісловник посідає 40-е місце серед 170 Вікісловників різними мовами світу.

Вікіпідручник (<http://uk.wikibooks.org>) – це вільна Web-колекція матеріалів, де кожен відвідувач може внести власний вклад в написання навчальної літератури. Цей ресурс базується на Wiki-технології, а представлені на ньому матеріали можна безкоштовно читати, копіювати, видавати та змінювати їх зміст. Розміщення інформації Вікіпідручнику здійснюється на умовах ліцензії GNU Free Documentation License. На головній сторінці Вікіпідручнику є 11 «книжкових полиць»: Мови, Історія, Масові комунікації, Пошук істини / Філософія, Математика, Комп'ютери, Програмування, Енергетика, Правознавство, Різне, Кулінарія.

ВікіЦитати (<http://uk.wikiquote.org>) – це проект, де зібрано цитати, афоризми, приказки тощо. Україномовна версія проекту стартувала 30 листопада 2005 року. Цей Wiki-ресурс містить цитати з фільмів, літературних творів, прислів'я та приказки народів світу, латинські вирази, вислови відомих особистостей з різних галузей.

Вікіджерела (<http://uk.wikisource.org>) – україномовний розділ мережевої бібліотеки Wikisource. Він містить довідники, документи, закони, історичні твори, наукові статті, пісні, публіцистику, художню літературу, тобто будь-які тексти та мультимедійні об'єкти (приміром, ілюстрації), що можуть бути корисні україномовній спільноті. Вікіджерела від Вікіпідручника відрізняється тим, що в останньому міститься тільки навчальна література (підручники, посібники тощо), а контент Вікіджерел значно ширший і може включати літературні твори будь-яких жанрів та тексти програм, але у Вікіджерелах заборонено оприлюднювати тексти, захищені авторським правом.

Віківиди (<http://species.wikimedia.org>) – проект Фонду Вікімедіа, задуманий як відкритий і вільний каталог біологічних видів, що містить класифікацію тварин, рослин, грибів, бактерій, археїв, найпростіших і решти форм життя.

Вікіновини (<http://uk.wikinews.org>) – сайт новин, які може писати кожний. Вікіновини містять два головних типи статей: загальні (синтетична стаття) та оригінальний репортаж (новини з перших рук, написані учасниками ВікіНовин). Статті новин відсортовані за регіонами (Україна – Європа – Азія – Північна Америка – Південна Америка, Африка – Австралія – Північний та південний полюси) та категоріями (Політика – Економіка – Культура – Спорт, Суспільство – Події – IT).

Вікісховище (<http://commons.wikimedia.org>) – це збірка медіа-файлів, яку може доповнювати кожний користувач. На головній сторінці Вікісховища представлено посилання на вибрані зображення (зображення і діаграми, які були визнані більшістю користувачів проекту найбільш значущими), зображення дня, медіа-файл дня. Шукати, переглядати чи додавати свої медіа-файли у Вікісховищі можна відповідно до змісту: за тематикою (Природа, Суспільство і Культура, Наука), за типом (зображення, звуки, відео), за авторами (архітектори, композитори, скульптори, фотографи, художники), за типом ліцензії, за походженням (з енциклопедій, з журналів, опубліковано авторами).

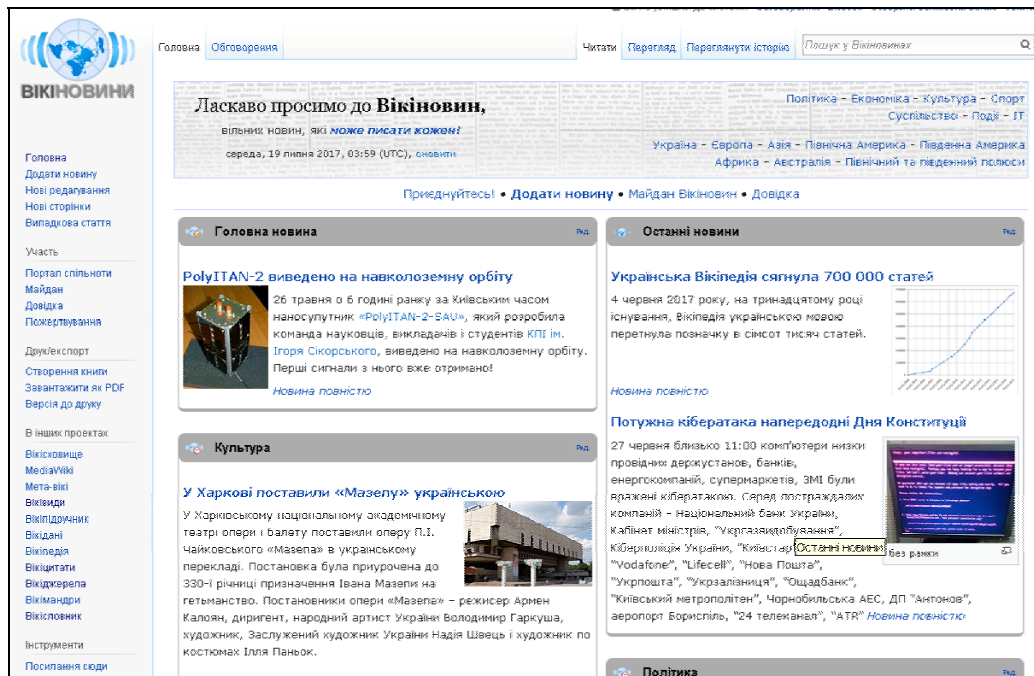


Рис.2.2. Головна сторінка сайту “Вікіновини”

Rodovid (uk.rodovid.org) – це відкритий wiki-проект, створений для того, щоб кожен міг побудувати власне генеалогічне дерево. Родовід – це єдине дерево, що подається в горизонтальному вигляді і яке можна вести одночасно кількома мовами. Це некомерційний, багатомовний, відкритий wiki-проект, створений у 2005 році в Києві українськими ентузіастами. Rodovid.org може працювати з глобальною базою 20-ма мовами. Усі мовні версії сайту використовують одну базу даних, тому записи, що додаються для однієї мовної версії, будуть з'являтися і для інших. Це стало можливим завдяки тому, що родинні дерева представлені передусім фактами: датами, іменами тощо. Таким чином, перекладатися повинні текстові дані (ім'я, вік, дата народження тощо), що може бути зроблено автоматично.

Wikiosvita (http://wikiosvita.tneu.edu.ua) – портал вільного розповсюдження освітніх знань (рефератів, навчальних робіт, досліджень), наукового дослідження), а також відстоювання власної позиції.

2.3. Велика українська енциклопедія

Зараз в Україні ведеться робота зі створення «Великої української енциклопедії» (ВУЕ) – багатомовного енциклопедичного видання універсального характеру, публікація якого запланована на 2013-2026 роки. У зв'язку з цим проводяться відповідні теоретичні розробки принципів, що забезпечують найбільш ефективні способи створення і використання таких ресурсів. Електронна версія Великої української енциклопедії (e-ВУЕ) – портална версія «Великої української енциклопедії», інтернет-ресурс, що працює за технологією Вікі, а саме Semantic MediaWiki. Це універсальна мультимедійна науково-популярна енциклопедія. Вона містить більше інформації порівняно з паперовою версією – вона доповнюється мультимедійними ресурсами, сторінками-інтеграторами, системою

семантичної навігації тощо. До переваг електронної енциклопедії можна віднести також компактність зберігання; відсутність обмежень на обсяг статей; більш швидкий та зручний пошук матеріалів; можливість постійного оновлення та редагування інформації; використання гіперпосилань для динамічного зв'язку з іншими об'єктами.

Енциклопедичні статті е-ВУЕ інтегровані в єдиний інформаційний ресурс шляхом прагматичного розподілу матеріалу в статтях та продуманою системою гіперпосилань: інформація у е-ВУЕ структурована за принципом «одне гасло — одна стаття», тож на свій конкретний інформаційний запит читач отримує чітку відповідь. Якщо представлено гасло семантично широке, а описуване ним явище має внутрішню структуру, то стаття про відповідне гасло є корінною, а через систему гіперпосилань із нею пов'язані інші, тематично вужчі і конкретніші статті, присвячені пов'язаним поняттям, підтипам та прикладам.

Більшість статей е-ВУЕ — авторські, написані провідними фахівцями України — співробітниками наукових та освітніх установ, експертами галузей народного господарства. Наявність у матеріалу висококваліфікованого автора є запорукою перевіреності й достовірності інформації.

Універсальні енциклопедичні видання, що виходять у різних країнах світу, прийнято оцінювати за різними кількісними і якісними параметрами. Найважливішими з кількісних вимірів вважаються *загальний обсяг* — кількість гасел (статей), та *глибина* — середня кількість знаків у статті.

Статті можуть бути пов'язані між собою зв'язками різного типу та семантики (саме використання Semantic MediaWiki дозволяє явно визначати семантику таких зв'язків). Наявність зв'язків дозволяють безпосередньо переходити від одних статей до інших, здійснюючи навігацію між ними в прямому і зворотному напрямку з метою отримання необхідної інформації. Більш того, можуть існувати посилання на зовнішні інформаційні ресурси.

Енциклопедичні статті містять визначення терміна або поняття, яке відображає його наукові класифікаційні ознаки. Стаття в залежності від обсягу енциклопедії та її завдань може бути обмежена лише однією дефініцією або включати також інші додаткові відомості про поняття. Статті портальної версії ВУЕ належать до трьох типів:

- 1) статті-огляди;
- 2) статті-довідки;
- 3) статті-тлумачення.

Статті зазначених типів мають різний характер та обсяг.

Статті-огляди — це найбільш докладні, присвячені широким темам: країнам, галузям знань і господарювання, найважливішим історичним подіям, найбільш значущим особам тощо. Вичерпна й структурована інформація у них супроводжується історичними довідками, статистичними відомостями, біографічними даними.

Статті-довідки висвітлюють конкретні об'єкти, приміром, наукову теорію, історичну подію або організацію. Характеристики цих об'єктів можуть також містити біографічні, географічні, статистичні дані.

Статті-тлумачення – це найкоротші статті, які містять лише визначення відповідного терміну.

Більшість статей перших двох типів містять списки літератури, а також супроводжуються нетекстовими матеріалами: фотографіями, рисунками, малюнками, кресленнями, картами, відео- чи аудіозаписами тощо.

Планується що більшість статей е-ВУЕ буде мати обсяг від 2000 до 6000 знаків, щоб досить повно представити основні аспекти описуваного об'єкта.

Рубрикація статті має повторювати її типову схему, розроблену для паперової версії. До статей можуть додаватися ються ілюстрації, таблиці та карти, відео- та аудіоматеріали, а також бібліографія. У бібліографії подаються також електронні джерела та посилання на офіційний сайт, якщо такий є. Для статей про зарубіжні об'єкти, осіб, організації, установи тощо бажано подавати рубрику або абзац про зв'язок з Україною (за наявності такої інформації).

е-ВУЕ є енциклопедією відкритого доступу, що передбачає вільний та безкоштовний доступ читачів до її інформації з правом читати, завантажувати, копіювати, поширювати, друкувати, робити пошук, індексувати і посилатися на статті, тобто використовувати її з законними цілями без фінансових, юридичних і технічних перешкод. Але, на відміну від Вікіпедії, користувачі не мають права безпосередньо вносити зміни у статті енциклопедії – це мають змогу робити тільки автори статей та адміністратори е-ВУЕ. Проте користувачі мають можливість залишати свої зауваження та пропозиції щодо статей на сторінках обговорення. Це забезпечує зворотний зв'язок та має поліпшити якість контенту.

Завдяки використанню технологічної платформи Wiki виникає можливість інтегрувати е-ВУЕ з іншими інформаційними системами. Приміром, одна з функцій Semantic MediaWiki забезпечує генерацію онтології за обраною користувачем множиною Wiki-сторінок. Таким чином, е-ВУЕ виконують роль бази знань для інших застосунків. З іншого боку, статті енциклопедії можуть посилатися на зовнішні ресурси – сторінки різних Wiki-ресурсів, мультимедійні дані тощо.

е-ВУЕ як джерело інформації має забезпечувати інформаційні потреби у різних сферах людської діяльності, виконувати освітню та наукову діяльність. Для цього необхідно надавати користувачам розвинуті пошукові засоби.

Використання е-ВУЕ надає багато переваг у порівнянні з паперовим варіантом, найбільш значущими з яких є:

- *оперативність*: якщо у паперовому варіанті стаття публікується тоді, коли підготовлений до публікації відповідний том енциклопедії, то в е-ВУЕ стаття може бути опублікована відразу після її створення;

- *актуальність*: інформація, що міститься в статті, може застаріти вже через кілька років, та в паперовому варіанті стаття може бути перевидана тільки через великий час, тоді як в е-ВУЕ стаття може бути перевидана, у залежності від обраної процедури, або негайно, або при черговому перевиданні е-ВУЕ;
- *автоматизація створення*: використання сучасних методів та технологій керування знаннями, до яких відносяться семантичні Wiki, значно прискорює та полегшує процес створення статей е-ВУЕ;
- *приступність інформації*: до електронної версії енциклопедії може отримати доступ значно більша кількість користувачів у будь-якій точці світу.

Простір назв (англ. namespace) – це спеціальний механізм для класифікації сторінок за їх призначенням (енциклопедичних, користувацьких, службових тощо). Позначкою належності сторінки до певного простору назв є спеціальний префікс з двокрапкою.

У е-ВУЕ використовується схема з 18 стандартних просторів назв: основний простір назв (без префіксів – для енциклопедичних сторінок), 15 додаткових із префіксами та 2 псевдопростори назв.

Основний простір назв статей ВУЕ формуються з відповідних гасел Словника та реєстрів за напрямками. Сторінки енциклопедичного змісту: статті, неоднозначності, списки та переспрямування. Приклад використання – “Абетка”, або “Петренко, Петро Петрович”.

Простір назв «ВУЕ:» – це назва для самого Словника, а також для напрямів, спеціальностей, інших класифікаторів, реєстрів гасел за напрямками (загалом або по літерах), ліцензій, юридичної інформації ВУЕ, авторів і колективу ВУЕ та інших – формуються додаванням до назви приставки ВУЕ. Приклад використання – “ВУЕ:Автори ВУЕ”.

Простір назв «Довідка:» використовується для методичних і довідкових матеріалів ВУЕ.

Простір назв “Категорія” – службовий, він використовується редакторами для створення категорій статей.

Одним з важливих елементів електронної версії ВУЕ є класифікатор, що забезпечує впорядкування понять енциклопедії в ієрархічну структуру. Наявність класифікаторів полегшує навігацію і пошук статей. Кожна стаття може бути пов'язана з декількома класифікаторами, причому в кожному класифікатор – з багатьма його поняттями на різних рівнях. На технологічному рівні основою для класифікаторів можуть стати категорії Wiki і пов'язані з ними плагіни (наприклад, CategoryTree для відображення дерева категорій).

Щоб користувачі могли зручно та оперативно знаходити в цьому інформаційному масиві необхідні дані, їм надаються різні варіанти пошуку:

- за ключовими словами;
- за тематикою предметної області (PrO);
- за типом інформаційного об'єкта (IO);
- за семантикою інформаційного об'єкта (IO).

Пошук за ключовими словами (за повним гаслом статті або за початковими літерами гасла) пошуку здійснюється через пошуковий рядок — стандартний елемент Wiki-сторінки. Такий вид пошуку забезпечує найбільш швидкий доступ до інформації, але передбачає, що користувач має знати точну назву об'єкта.

Пошук за тематикою предметної області базується на загальноновживаних класифікаторах, що формалізують інформацію за допомогою категорій та підкатегорій, пов'язаних із різними науковими напрямками. Такий пошук зручний і зрозумілий для фахівців, але потребує певних знань щодо Про.

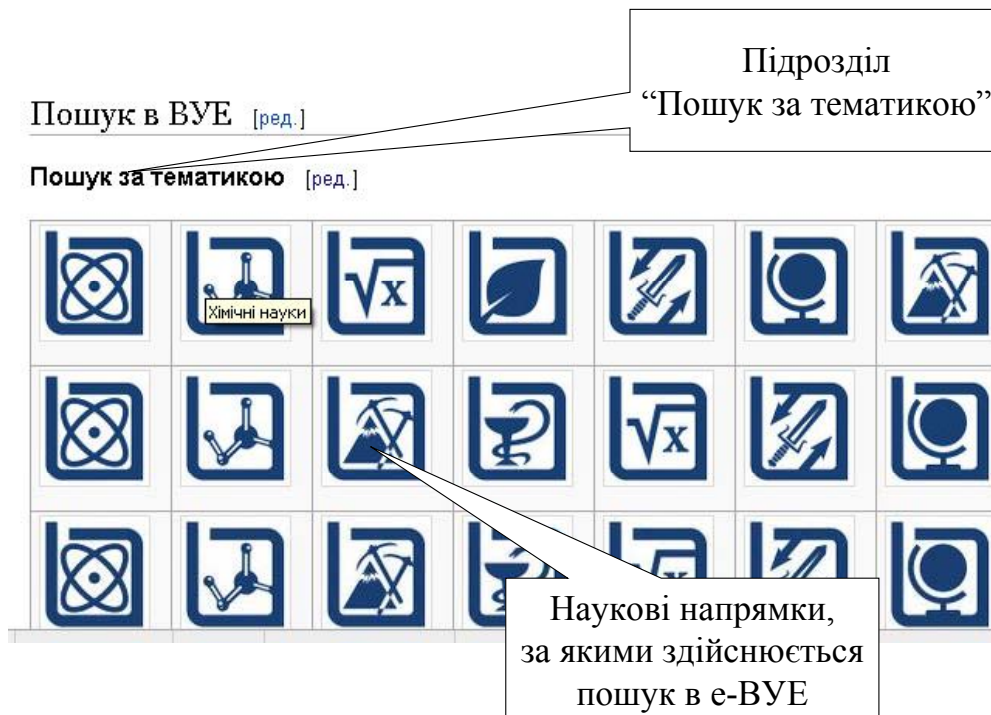


Рис.2.3. Пошук статей за науковими напрямками

Пошук за типом інформаційного об'єкта дозволяє спростити доступ до енциклопедичних статей з подібною структурою. Його ефективність значним чином визначається вдалою типизацією статей у певні тематичні групи. Приміром, користувачі можуть отримувати доступ до таких типових ІО, як «Державні діячі України» (як перетин категорій «Державні діячі» та «Персоналії України»), «Вид спорту» ((як перетин категорій «Діяльність» та «Спорт»)), «Міста України» тощо. Типи ІО можуть впорядковуватися ієрархічно, приміром, тип ІО «Державні діячі України» є підкатегорією ІО «Державні діячі», який усвою чергу є підкатегорією «Персоналії». В процесі аналізу користувацьких запитів можна виділити й більш специфічні ІО, які найбільш часто викликають інтерес, але не пов'язані безпосередньо із іншими системами класифікації – приміром, “Отруйні рослини” або “Небезпечні тварини України”. При цьому пошук за типом інформаційного об'єкта дає змогу читачеві не тільки знайти потрібну статтю, а й ознайомитися із змістово близькими статтями.

Пошук за семантикою Ю базується на використанні семантичних властивостей Wiki-сторінок, що надає можливість автоматизовано вибирати потрібні користувачеві дані з тих сторінок, що відповідають умовам запиту, та інтегрувати ці дані. Приміром, можна надати користувачеві перелік найбільших міст України (із зазначенням кількості їх населення) або перелік українських письменників 17-18 століття. Реалізація семантичного пошуку потребує встановлення додаткових семантичних плагінів, призначених для оперування із семантичними властивостями статей. Цей варіант пошуку наразі перебуває в процесі розробки, але у перспективі має перетворити е-ВУЕ на базу знань, яку зможуть автоматизовано використовувати інші застосунки.

MediaWiki дозволяє користувачам створювати довільні набори категорій, не накладаючи на їхні імена і зв'язки між ними які-небудь обмеження. Аналогічно обстоїть справа і із семантичними властивостями.

Однак, якщо різні автори будуть використовувати різні назви для семантичних властивостей і категорій, те це значно ускладнить виконання запитів до них. Тому часто при створенні Wiki-ресурсів спочатку розробляються різні методичні рекомендації та прототипи статей.

Наприклад, у [52] описані типові схеми статей для різних напрямків наук і типові схеми біографічних статей (зокрема, статей про вчених в області суспільних наук чи техніки, письменників, державних і політичних діячів), які планується застосовувати для створення е-ВУЕ. Якщо розробити на їхній основі єдину систему категорій і семантичних властивостей, то можна надати користувачу Wiki-ресурсу розвинуті можливості по інтеграції наявних відомостей і семантичному пошуку серед наявної інформації. Наприклад, можна автоматично генерувати переліки дослідників, що працювали у визначений час над тією чи іншою проблемою, що знаходяться в обраній місцевості організаціях тощо. При внесенні змін у ті ресурси, що є основою для побудови таких списків, самі переліки будуть динамічно обновлятися.

Методичні рекомендації з підготовки, редагування та оформлення статей до Великої української енциклопедії описують наукові засади та теоретико-методологічні принципи створення сучасних енциклопедій [50]. Вони формально характеризують різні види статей та їх структуру. Такий опис статті може бути використано для того, щоб визначити категорію статті, її шаблон та семантичні властивості.

Типові схеми статей дозволяють визначити заголовки та підзаголовки статті. Самі назви категорій теж можуть бути згенеровані за цими методичними вказівками а потім конкретизовані відповідно до специфіки ПрО.

Приміром, типова схема статті про науку або галузь науки містить наступні розділи:

Гасло

Дефініція

Структура

Нарис з історії розвитку науки

Місце, час і обставини зародження
Основні етапи історичного розвитку
Сучасний стан і тенденції подальшого розвитку
Визначні представники та наукові школи
Впливові наукові установи
Найважливіші теорії, вчення, напрями
Розвиток науки в Україні
Найважливіші форми та методи дослідження
Місце в системі наукового знання
Результативність
Найважливіші вітчизняні й міжнародні організації
Найважливіші друковані та електронні органи
Бібліографія

Така структура може бути використана як основа для побудови шаблону заповнення статті, а її елементи – використовуватися для семантичної розмітки.

Створюється наступний шаблон:

```
{{типова схема статті про науку або галузь науки
|Гасло=
|Дефініція=
|Структура=
|Нарис з історії розвитку науки=
|Найважливіші форми та методи дослідження=
|Місце в системі наукового знання=
|Результативність=
|Найважливіші вітчизняні й міжнародні організації=
|Найважливіші друковані та електронні органи=
|Бібліографія=
}}
```

Але у більшості випадків доцільно розробляти прототипи типових статей, які містять не тільки рубрикацію, але й елементи розмітки (як семантичної, так і елементів оформлення), шаблони, а також додаткові вказівки для авторів статей.

За об'єктом опису статті е-ВУЕ поділяються на дві основні групи:

1) Статті про персоналії (біографістика), в яких викладено життєпис, діяльність та її результати конкретних осіб (політиків, вчених, митців, спортсменів тощо);

2) Статті про поняття/термін, що статті розкривають суть власних і загальних імен, що становлять терміносистеми різноманітних наук.

Доступ до цих груп гасел надається на головній сторінці е-ВУЕ за посиланнями «Поняття» та «Персоналії».

3. Семантизація Wiki-ресурсів

3.1. Технології *Semantic Web* як засіб обробки розподілених знань

Основні проблеми керування знаннями в розподіленому інформаційному середовищі Web пов'язані з:

- *Інтеграцією* знань, отриманих від різних IP (наприклад, інтеграція онтологій кількох різних ПрО або IP в одній ПрО);
- Пошуком *протиріч* між знаннями, що відображені в контенті різних IP, оцінкою їх достовірності та надійності;
- *Здобуттям нових знань* з вже наявних та їх поданням у формі, зрозумілій користувачеві;
- *Пошуком* знань, потрібних конкретному користувачеві для вирішення певних задач;
- *Автоматизацією створення метаданих*, що коректно описують контент IP (як текстових, так і мультимедійних) на рівні змісту, та ефективним пошуком у таких метаописах.

Можна навести ще багато прикладів проблем, що виникають у процесі керування знаннями в Web, але всі вони зводяться до наступних:

1. Вибір засобів подання знань (досить потужних, щоб задовольнити різноманітні потреби користувачів, але придатних для швидкої обробки та розуміння людиною);

2. Методи створення нових знань за наявною інформацією (приміром, створення метаопису IP, формування онтології IP, логічне – індуктивне, традуктивне, дедуктивне – виведення, виконання запитів до БЗ);

3. Методи порівняння різних інформаційних об'єктів на семантичному рівні (приміром, інтеграція або пошук відмінностей двох онтологій, співвідношення інформаційного запиту та IP, що відповідають цьому запиту, визначення ПрО за контекстом IP);

4. Оцінка якості нових знань (достовірності, несуперечності, актуальності, повноти).

Усі ці проблеми потребують вирішення при створенні та використанні Wiki-ресурсів. Детальніше застосовні аспекти семантичних технологій розглянуто в [39].

Аналіз публікацій показує, що задачі збереження, здобуття й аналізу знань ефективно вирішуються в рамках проекту *Semantic Web*. Ця розробка, розпочата у 2000 році Т.Бернесом-Лі [2], спрямована на перетворення середовища Web на глобальну розподілену базу знань, яку можуть використовувати різноманітні ІС.

Керування знаннями в сучасному Web має певну специфіку, пов'язану як з величезними обсягами даних, що обробляються, так і з динамічністю, гетерогенністю і децентралізованістю даних. Тому й *Semantic Web* на основі

інформаційних ресурсів Web теж має враховувати особливості цього середовища та застосовувати відповідні технології та методи обробки знань, пов'язані з онтологіями.

Semantic Web пропонує потужний практичний підхід до використання семантики як інформаційних ресурсів, так і засобів їх обробки, що потребує нових методів та інструментів [4]. Це потужний напрямок для підвищення ефективності розподіленого та сумісного доступу до інформації та її використання прикладними програмами. Детальніше це розглянуто в [56]. Головними задачами Semantic Web є створення мов семантичної розмітки електронних документів; словників, які описують структуру і семантику елементів семантичної розмітки, та реалізація засобів автоматичного генерування та обробки цієї семантичної інформації.

За визначенням W3C [34], *Semantic Web* – це розширення WWW, у рамках якого інформація (Web-контент) представляється у форматах, що забезпечують її використання програмними агентами, дозволяючи їм, таким чином, шукати, розділяти і інтегрувати інформацію значно легше, ніж це відбувається зараз [10]. Для них вже існують формальні моделі, мови подання, методи обробки та програмні засоби: на сьогодні створено низку найважливіших технологій: розширену мову розмітки (Extensible Markup Language, XML), система опису ресурсів (Resource Description Framework, RDF) [12], мова Web-онтологій (Ontology Web Language, OWL) [20], що призначені для опису, збереження і поширення знань, а також SPARQL – мову запитів до RDF.

Вплив Semantic Web формує нові перспективи в розробці програмного забезпечення, що базуються на “data-centric” моделі, яка використовує велику кількість різноманітних, розподілених даних та має велику виразність, полегшене спільне використання та більшу гнучкість. Програмування в Semantic Web – це потужний новий підхід, що забезпечує більш ефективного використання великих обсягів інформації та робить доступними різноманітні сервіси. Модель знань Semantic Web може інтегруватися з прикладними програмами і використовуватися для окремих домено-орієнтованих бізнес-логік із самої програми.

Semantic Web можна уявити як Web даних, описаних і пов'язаних таким чином, щоб створити контекст або семантику, які представлені за допомогою явно визначених граматичних та мовних конструкцій. Розширені прикладні програми Semantic Web можуть здійснювати інтеграцію словників предметної області (та їх онтологій). Врахування семантики інформаційних ресурсів дозволяє більш ефективно використовувати наявні дані для вирішення проблем. Якщо семантика джерела інформації не формалізована, то це потребує від користувачів або складних програмних інструкцій певним чином виявляти її.

На відміну від інших програм, прикладні програми Semantic Web зосереджені не на програмних інструкціях, а на даних. Багатство даних Semantic Web спрощує цю задачу. Це дозволяє відокремлювати дані від програмних інструкцій та дозволяє створювати більш гнучкі рішення. Крім

того, прикладні програми Semantic Web є Web-орієнтованими: вони використовують переваги від масштабування, різноманітності і розподіленості WWW. Багато сучасних програм намагаються вирішити ці проблеми, але вони не в змозі повною мірою скористатися з WWW і працюють лише в обмеженому, ізольованому інформаційному просторі. Застосування Semantic Web здатні використовувати розмір і різноманітність WWW шляхом створення стандартизованої виразної інформації.

Semantic Web використовує набір нових відкритих інформаційних стандартів, які можуть застосовувати й всі інші. Прикладні програми, що базуються на цих стандартах, можуть швидко впроваджувати нові джерела інформації.

На сьогодні проект Semantic Web активно розвивається, з'являються нові мови та стандарти роботи з розподіленими знаннями й вдосконалюються існуючі. Тому доцільно у процесі розробки моделей, методів та засобів підтримки сервіс-орієнтованих прикладних програмних систем орієнтуватися саме на ці досягнення Semantic Web і створювати семантичні Web-сервіси, здатні ефективно використовувати всі переваги нового інформаційного середовища.

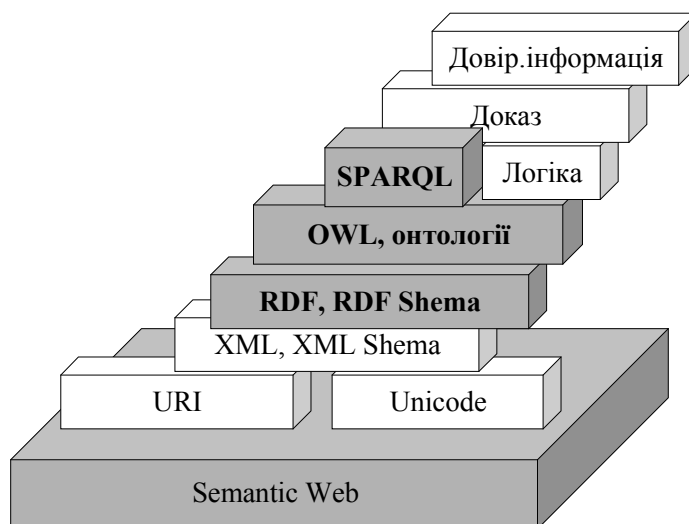


Рис.3.1. Стек технологій Semantic Web

Semantic Web – це злиття Web-технологій і науки про представлення знань (knowledge representation, KR), що є підобластю штучного інтелекту (artificial intelligence, AI), спрямоване на створення і підтримку потенційно складних моделей світу, що дозволяють міркувати про себе і про зв'язану з ними інформації.

З огляду на багаторічний досвід розробки, стандартизації і розвитку технологій Semantic Web у рамках World Wide Web Consortium, використання існуючих розробок для створення ІС є закономірним і потребує більш детального аналізу. Основними компонентами Semantic Web, на яких базуються ІС, є:

- онтології,
- сервіси;
- програмні агенти.

Онтологія – це формальний опис результатів концептуального моделювання різних предметних областей (ПрО), що забезпечує інтероперабельність та повторне використання знань в ІС. Онтології є ефективним засобом для моделювання уявлень про різноманітні ПрО і дають змогу формально відображати їх семантику. Зараз вони широко застосовуються в якості баз знань у різноманітних предметних областях [38].

Т.Грубер визначає онтологію як точну специфікацією концептуалізації [5], тобто основу онтології складають множини представлених у ній термінів та відношень, у той час як М.Ушольд [32] пов'язує онтологію з ПрО, визначаючи її є специфікацією концептуалізації, але тільки в тій її частині, що залежить від визначеної ПрО. За визначенням Н. Гуаріно [7], онтологія – це логічна теорія, аксіоми якої обмежують інтерпретації нелогічних символів мови, тобто онтологія має характеризувати концептуалізацію, обмежуючи можливі значення предикатів і функцій.

У загальному випадку *формальна модель онтології* – це впорядкована трійка

$$O = \langle T, R, F \rangle ,$$

де T – множина понять ПрО, що поділяється на класи та екземпляри; R – множина відношень між ними; F – множина функцій інтерпретації понять і відношень. Вона може бути конкретизована відповідно до специфіки задачі та ПрО.

Детальніший розгляд використання онтологій в ІС та їх інтеграції з іншими семантичними технологіями наведено в [56]. Слід відмітити, що обробка онтологічних структур є досить складною, і тому у багатьох практичних застосуваннях доцільно використовувати окремі випадки онтологій, які припускають простішу обробку, або здобуті з них знання.

Слід відмітити, що здобувати онтологічні знання значно простіше не з природномовних або мультимедійних ІР, а з тих, які вже структуровані та мають якусь семантичну розмітку. В той же час доцільно орієнтуватися на розповсюджені та зрозумілі для користувачів форми подання інформації. Обом цим умовам задовольняють семантичні Wiki-ресурси – інформаційні ресурси, що розробляються на основі технології Wiki, але розширені засобами для додавання семантичної розмітки інформації [29] та семантичного пошуку із виконанням складних запитів. Семантизація Wiki забезпечує інтеграцію інформації та її узгодженість, запобігає дублювання даних. Семантична розмітка дає змогу значно спростити всю структуру Wiki, допомагає користувачам швидше знаходити потрібну інформацію.

Найвідомішими прикладами застосувань Semantic Web є семантичні Wiki.

Онтології предметних областей

Онтології широко застосовуються зараз у різноманітних інтелектуальних застосуваннях для представлення знань щодо тієї предметної області, до якої відноситься задача, яку користувач вирішує за допомогою цієї ПС [40, 41].

У різноманітних наукових дослідженнях, пов'язаних з онтологічним аналізом, наводиться багато різних тлумачень терміна «онтологія предметної області», кожне з яких має певні недоліки, тобто на сьогодні немає загальноприйнятого визначення цього терміна. Це зумовлено тим, що для розв'язання різноманітних проблем, що стосуються онтологій ПрО, використовуються різні пояснення цього поняття, безпосередньо пов'язані зі специфікою задач, що розв'язуються за допомогою онтологій. Вибір найбільш придатного визначення залежить від цілей розробки онтології, які можуть бути такими:

- спільне використання загального розуміння інформації людьми та/або програмними агентами;
- повторне використання знань ПрО;
- можливість формулювати фактичні припущення щодо ПрО;
- відокремлення знань ПрО від операційного знання;
- аналіз знань у ПрО.

У широкому розумінні онтологія ПрО – це та частина знань стосовно ПрО, яка обмежує значення її термінів і не залежить від змінюваної частини знань цієї ПрО. Це набір угод щодо предметної області, що визначає ступінь погодження значень термінів серед фахівців предметної області.

Наявні підходи, що пов'язані з визначенням онтології ПрО, поділяються на гуманітарні, математичні та комп'ютерні. Детальніше їх переваги та недоліки проаналізовано в [47].

Гуманітарний підхід», до визначення поняття «онтологія предметної області» використовує неформалізовані терміни, які можна зрозуміти лише інтуїтивно. Тому такі визначення не придатні для використання в ІС. Наведемо кілька прикладів таких визначень. Онтологія – це:

- консенсус щодо ПрО для певних цілей;
- набір понять разом з їхніми визначеннями, їх ієрархічною організацією та відношеннями між поняттями (не тільки відношень «is-a» і «part-of»), а також аксіомами, які формалізують ці визначення й відношення ;
- фактична специфікація концептуалізації на рівні знань, тобто множина відмінних рис, що важливі для агента, де концептуалізація – це об'єкти, поняття та інші сутності, наявність яких передбачається в деякій ПрО, а також відношення, між ними;
- фактичний опис чи презентація деякої частини концептуалізації, яка може мати різні форми, але обов'язково містить словник термінів і деяку специфікацію їхнього змісту, визначення та вказівки на зв'язок понять, що в сукупності накладає структуру

на предметну область і обмежує можливі інтерпретації термінів. Онтологія фактично завжди є проявом розподіленого розуміння предметної області, з яким узгоджується деяке число агентів. Концептуалізація є поглядом на світ: вона відповідає способу мислення про деяку предметну область і може розглядатися як «множина неформальних правил, що обмежують структуру частини дійсності» ;

- опис об'єктів і відношень ПрО у вигляді речень формальної мови в термінах об'єктів і відношень, необхідних для визначення семантики мови подання, де екстенціональну концептуалізацію слід теж розуміється в змісті, подаючи точне й недвозначне визначення понять, необхідних для подання знань, без посилання на обчислювальні аспекти;
- наочна специфікація інтенціональної концептуалізації: знання ПрО подаються з використанням декларативного формалізму «універсум області» – множини об'єктів і опису співвідношення між ними, що відображається в словнику подання, де визначення пов'язують імена сутностей в універсумі зі зрозумілим для людини текстом, який пояснює значення цих імен, і з формальними аксіомами, що обмежують інтерпретацію та правильне використання цих термінів;
- формальна (машинно-читабельна), явна специфікація розподіленої концептуалізації, де концептуалізація – абстрактна модель деякого явища у світі, задана за допомогою понять, релевантних цьому явищу, а всі використані поняття й обмеження щодо їх застосування фактично визначені й прийняті деякою групою. Онтологія містить універсальні ґрунтовні знання предметної області, що не залежать від їх використання.

Хоча гуманітарний підхід до визначення онтології ПрО дає змогу певною мірою розкрити зміст цього поняття, але він неспроможний дати формальне визначення як самого цього терміна, так і тих понять, що використовуються у визначенні, як-от: «концептуалізація», «знання» і «подання знань».

У комп'ютерному підході , до визначення онтології ПрО під онтологією ПрО розуміють певну несуперечну конструкцію, створену відповідною комп'ютерною мовою для подання онтологій [6]. Наявність таких засобів опису онтологій забезпечує їх практичне застосування, проте таке визначення не дає можливості сформулювати відмінності між мовами опису онтологій і мовами подання знань, які за своєю семантикою еквівалентні мовам числення предикатів.

Математичний підхід , визначає поняття онтології ПрО в математичних термінах і через математичні теорії. Наприклад, у [7] онтологія визначається як логічна теорія, що обмежує припустимі моделі логічної мови, надаючи аксіоми, які не виходять за межі значень нелогічних символів (предикатів і функцій) логічної мови, що використовуються як «примітиви для визначених цілей подання». Мета онтології –

характеризувати концептуалізацію, обмежуючи можливі інтерпретації нелогічних символів логічної мови для встановлення консенсусу стосовно того, як відтворювати знання з використанням цієї мови. На жаль, такий підхід фактично не вказує на зв'язки між математичними термінами та змістовними аспектами предметних областей. Визначення онтологій ПрО, сформульовані в межах математичного підходу, мають певні переваги порівняно з комп'ютерним, оскільки за такої ж чіткості містять менше технічних деталей. Але вони не мають у собі явних припущень щодо властивостей онтологій ПрО.

Мови подання комп'ютерних онтологій

Комп'ютерний підхід до онтологій передбачає наявність формалізованих мов подання онтологічних знань. На сьогодні для цього найбільш широко використовуються створені в рамках проекту Semantic Web такі XML-орієнтовані мови, як RDF та OWL.

Стандарт *RDF* призначений для опису предметної області ресурсів. Він був затверджений у 1999 році консорціумом W3C і підтримується нині багатьма розробниками ПЗ і постачальниками контенту [3]. *RDF* являє собою модель опису метаданих. надає можливість формулювати твердження у вигляді, придатному для обробки комп'ютером.

RDF підтримує базову модель даних «об'єкт – атрибут – значення» для метаданих [11]. Як XML Schema використовується для визначення словника, так *RDF Schema (RDF(S))* дає змогу розробникам визначати конкретний словник для даних *RDF* (такий, як *authorOf*) і вказувати види об'єктів, до яких можуть застосовуватися ці атрибути. Іншими словами, механізм *RDF Schema* надає базову систему типів для моделей *RDF*. Ця система типів використовує деякі визначені терміни, такі, як *Class*, *subPropertyOf* і *subClassOf* для схеми, орієнтованої на конкретне застосування. Вираз схеми *RDF* також є коректним виразом *RDF* (як і вирази схеми XML – коректні вирази XML).

Мова подання онтологій *OWL* [19] розширює можливості XML, *RDF*, *RDF Schema* та *DAML+OIL*, де онтологія є колекцією *RDF*-трижок. *OWL* пропонує конструкції для вираження понять, відношень, потужності, анотацій та конкретизації понять тощо. *OWL* підтримується великою кількістю редакторів онтологій і вирішувачів. Онтологія *OWL (Web Ontology Language)* – це послідовність аксіом і фактів, а також посилань на інші онтології. Онтології *OWL* є документами Web, на які можна посилатися через URI. Вони можуть містити компонент для запису авторства та іншої подібної інформації.

OWL орієнтована на незалежну розподілену розробку онтологій. Її синтаксис дає змогу довизначати будь-який клас, екземпляр або властивість незалежно від того, як вони були визначені спочатку. Процес довизначення не потребує узгодження з автором початкового визначення й може здійснюватися в окремому документі й без зміни документа, де зафіксоване це початкове визначення. Так можна редагувати будь-яку онтологію

у форматі OWL і створювати власну версію, а знання про екземпляри, класи та властивості можуть накопичуватися й уточнюватися поступово, за участю великої кількості людей.

Одним з основних елементів в OWL є класи, які відповідають поняттям предметної області, яку відображає відповідна онтологія. У мові OWL передбачено кілька різних способів визначення нового класу:

- через ідентифікатор класу;
- через повний перелік екземплярів, які в сукупності складають цей клас;
- за допомогою обмеження властивості;
- через теоретико-множинні операції над іншими класами.

Кожен екземпляр в онтології OWL належить до класу `owl:Thing`, а кожен встановлений користувачем клас автоматично є підкласом `owl:Thing`. Специфічні для ПрО кореневі класи визначаються простим оголошенням іменованого класу. OWL також задає порожній клас: `owl:Nothing`. Основним конструктором для таксономії класів є `rdfs:subClassOf`, який зв'язує окремий клас із загальним. Це відношення є транзитивним.

Крім того, для опису ПрО використовуються такі елементи, як властивості та екземпляри класів.

Крім наведених вище засобів подання знань, OWL має набір конструкцій, які дають змогу встановлювати відношення еквівалентності й несумісності між класами, властивостями та екземплярами. Ці можливості певною мірою дають змогу перебороти труднощі, породжені принципом незалежної розподіленої розробки онтології.

- *еквівалентність класів*. `Owl:equivalentClass` визначає еквівалентність між двома класами: якщо між класами $C1$ і $C2$ є відношення еквівалентності, тоді, якщо X є екземпляром $C1$, тоді X є екземпляром і класу $C2$, і навпаки;
- *еквівалентність властивостей*. `Owl:equivalentProperty` визначає еквівалентність між двома властивостями: якщо між властивостями $P1$ і $P2$ є відношення еквівалентності, то наслідком того, що X пов'язано з Y відношенням $P1$, є той факт, що X пов'язано з Y відношенням $P2$, і навпаки;
- *несумісність класів*. `Owl:disjointWith` визначає несумісність класів: якщо класи $C1$ і $C2$ пов'язані відношенням несумісності, то наслідком того, що X – це екземпляр класу $C1$, є той факт, що X – не екземпляр $C2$, і навпаки;
- *еквівалентність екземплярів*. `Owl:sameAs` встановлює, що два імені екземпляра в онтології посилаються на той самий екземпляр з ПрО;
- *несумісність екземплярів*. `Owl:differentFrom` встановлює, що два імені екземпляра в онтології не можуть посилатися на один екземпляр з ПрО;

- *несумісність групи екземплярів*. Owl:AllDifferent встановлює, що жодні два екземпляри групи не можуть посилатися на один екземпляр з ПрО.

Визначення класу утворюється з його назви або посилання на неї та списку обмежень. Представники класу належать до перетину зазначених обмежень. Щоб визначити екземпляр, достатньо оголосити його членом якогось класу.

Властивості – це бінарні відношення, які дають змогу задавати факти про члени класів та екземпляри. Розрізняють два типи властивостей:

- *властивості-значення* – відношення між представниками класів і RDF-літералами або типами даних, обумовлених XML Schema;
- *властивості-об'єкти* – відношення між представниками двох класів.

Мова OWL має три діалекти, які різняться за виразністю:

- *OWL Lite* – це найпростіший варіант, орієнтований на користувачів, яким потрібні лише класифікація ієрархій і прості обмеження.
- *OWL DL* орієнтований на тих, кому необхідні максимальна виразність без втрати гарантованого завершення всіх обчислень за певний час. OWL DL містить усі мовні конструкції OWL з обмеженнями поділу типу (клас не може бути окремою властивістю, а властивість – індивідом або класом). OWL DL пов'язано з дескриптивними логіками.
- *OWL Full* надає максимальну виразність і синтаксичну потужність RDF без обчислювальних гарантій. Наприклад, в OWL Full клас може одночасно розглядатися і як сукупність екземплярів, і як екземпляр. Інша суттєва відмінність від OWL DL полягає в тому, що owl:DatatypeProperty може бути позначена як owl:InverseFunctionalProperty. OWL Full припускає такі онтології, що розширюють склад визначеного словника RDF або OWL.

Кожна онтологія OWL Lite є онтологією OWL DL, а онтологія OWL DL – онтологією OWL Full.

Важливо, що мова OWL передбачає відкритість, тобто клас спочатку може бути визначений в одній онтології, а потім розширений в інших. Унаслідок цього, судження є монотонними. Нова інформація не спростовує попередню: факти і наслідки можуть тільки додаватися до онтології й не можуть видалятися. Тому інформація може бути суперечливою, і розробник онтології має це враховувати.

Через великий інтерес до онтологічного аналізу сьогодні є також багато інших мов для подання онтологій. Хоча найбільшу популярність має мова OWL, розроблена в рамках проекту Semantic Web, та її остання версія – OWL 2.0 проте інші мови, що мають інакші виразові здатності, теж є досить популярними, коли йдеться про побудову різних типів онтологій. Більш детальний огляд мов подання онтологій наведено в [53].

3.2. Створення онтологій предметних областей за допомогою Protégé

Подання онтологій в Protégé

Protégé – локальна, вільно поширювана Java-програма, призначена для побудови (створення, редагування й перегляду) онтологій ПрО. Protégé містить редактор онтологій, що дає змогу проектувати онтології, розгортаючи ієрархічну структуру абстрактних і конкретних класів і слотів. На основі сформованої онтології Protégé дає можливість генерувати форми здобуття знань для введення екземплярів класів і підкласів [61, 1]. Наголосимо, що останні версії (починаючи з 4.0) значно відрізняються від Protégé 2000 і попередніх версій як форматом подання онтологій, так і можливостями для їх обробки [31].

Онтологія в Protégé – це явна специфікація концептуалізації. Цей інструментальний засіб підтримує спільне використання понять, яка містить засоби подання предметних знань і домовленості про методи міркувань. Protégé дозволяє перетворювати певний опис погляду на світ у конкретній сфері інтересів, який складається з набору термінів і правил їх використання, що обмежує їх значення в рамках конкретної ПрО, у таку форму, що вони стають придатними для машинної обробки, приміром, онтології, представлені мовою OWL.

Таким чином, створена за допомогою Protégé онтологія – це система, що складається з наборів понять і тверджень про ці поняття, на основі яких можна будувати класи, об'єкти, відношення, функції та теорії.

Таку онтологію можна розглядати як БЗ спеціального виду з семантичною інформацією певної ПрО. Компоненти, з яких складаються конкретні онтології, залежать від парадигми подання, але практично всі моделі онтологій містять певні *концепти* (поняття, класи), *властивості* концептів (атрибути, ролі), *відношення* між концептами (залежності, функції) та *обмеження застосування*, що визначаються аксіомами. Концептом може бути опис задачі, функції, дії, стратегії, процесу міркування тощо.

Онтологія у мові OWL (Web Ontology Language) є послідовністю аксіом і фактів, а також посилань на інші онтології. Вони також містять компонент для запису авторства та іншої подібної інформації. Онтології OWL є документами Web, на них можна посилатися через URI. Формально відобразити OWL-онтологію можна так:

```
<ontology> ::= Ontology ( [<authorship-etc>] {<directive>} )  
<authorship-etc> ::= ...  
<directive> ::= <imports>  
<directive> ::= <axiom>
```

<directive> ::= <fact>

<imports> ::= imports (<URI>)

Фундаментальні поняття певної ПрО мають відповідати класам, що містяться в коренях різних таксономічних дерев. Кожен екземпляр в онтології OWL належить до класу owl:Thing, а кожен встановлений користувачем клас автоматично є підкласом owl:Thing. Специфічні для ПрО кореневі класи визначаються простим оголошенням іменованого класу. OWL також задає порожній клас: owl:Nothing. Визначення можуть бути розширюваними й розподіленими [19].

Фундаментальним таксономічним конструктором для класів є rdfs:subClassOf. Він зв'язує окремий клас із загальним. Якщо X – підклас Y, то кожен представник X – також представник Y. Відношення rdfs:subClassOf є транзитивним. Якщо X – підклас Y і Y – підклас Z, тоді X – підклас Z.

Визначення класу складається з двох частин: назви (або посилання на неї) й списку обмежень. Кожний вираз, який безпосередньо міститься у визначенні класу, уточнює властивості представників цього класу. Представники класу належать до перетину зазначених обмежень. Для визначення екземпляра досить оголосити його членом якогось класу.

Властивості дають змогу утверджувати загальні факти про членів класів і про екземпляри. Вони становлять бінарні відношення. Розрізняють два типи властивостей:

- *властивості-значення* – відношення між представниками класів і RDF-літералами або типами даних, зумовлених XML Schema;
- *властивості-об'єкти* – відношення між представниками класів.

При наданні властивості є багато способів обмежити це відношення. Можна задати домен і діапазон. Властивість може бути визначена як спеціалізація (підвластивість) наявної властивості. Можливі й більш складні обмеження. Властивості, так само як класи, можуть бути організовані ієрархічно.

OWL використовує більшість убудованих типів XML Schema. Посилання на ці типи здійснюються за допомогою URI для <http://www.w3.org/2001/XMLSchema>.

Формальна семантика OWL містить опис того, як отримати логічні наслідки, маючи таку онтологію, тобто здобути факти, що не подаються в ній безпосередньо, проте зумовлені її семантикою. Ці наслідки можуть спиратися на один документ або множину розподілених документів, які комбінуються з використанням спеціальних механізмів OWL.

Онтологія OWL відрізняється від схеми XML тим, що вона є поданням знань, а не форматом повідомлень. Однією з її переваг є наявність інструментального ПЗ, призначеного для аналізу знань,

поданих мовою OWL. Ці інструменти забезпечують загальну домено-незалежну підтримку онтологічного аналізу.

Основні елементи онтології, створюваної в Protégé, – це [61]:

- *класи* (Classes) множини, елементами яких є екземпляри;
- *екземпляри* класів (Individuals), що являють собою конкретні об'єкти предметної області, яка цікавить користувача;
- *властивості* (Properties) бінарні відношення між екземплярами;
- *домени*, що визначають множини екземплярів, до яких застосовується властивість (область визначення властивості);
- *діапазон* (range) множина екземплярів, що можуть бути зв'язані цією властивістю з об'єктами з домену.

Властивості можуть бути зворотними (inverse) одна щодо одної, транзитивними та симетричними. Область значень властивості може бути обмежена єдиним об'єктом, тоді властивість називається функціональною.

Властивості відповідають слотам у Protege-фреймі. Вони також називаються ролями в описі логіки й відношень в UML та інших об'єктно-орієнтованих поняттях.

Процес створення нової онтології в Protégé полягає в створенні класів, їх властивостей та екземплярів цих класів [15].

Під час запуску Protégé 4. З відкривається нова онтологія. Її можна зберегти з обраним ім'ям у форматі RDF/XML, вибравши в меню «File» опцію «Save as».

Основні етапи створення онтології

У процесі створення застосовної онтології ПрО у Protégé потрібно послідовно виконати такі основні кроки:

- Задати основні класи ПрО, приміром, “Людина”, “Організація”, “Географічний об'єкт”;
- Визначити ієрархічні зв'язки між цими класами, приміром, клас “Дослідник” є підкласом класу “Людина”;
- Задати атрибути цих класів, значеннями яких є константи, та вказати тип цих даних, приміром, для класу «Людина» задати атрибути “Ім'я”, “Прізвище” зі значеннями типу “Рядок символів” та атрибут “Рік народження” зі значенням “Ціле”;
- Задати атрибути класів, що самі є класами. Наприклад, для класу “Студент” задати атрибут «вивчає», що належить до класу “Предмет, що вивчається”;
- Якщо потрібно, вказати додатково властивості класів, екземплярів класів та атрибутів, приміром, можна вказати, що класи “Людина” та “Особа” є еквівалентними, а класи “Студент” та “Викладач” не можуть перетинатися;

- Створити екземпляри класів, задавши їх ідентифікатори та визначивши значення їх атрибутів.

Після цього можна вдосконалювати онтологію, вносячи зміни в усі описані вище елементи, щоб більш точно та повно відобразити предметну область.

Розглянемо детальніше, як саме слід здійснювати ці операції.

Створення класів

Створення класів здійснюється в закладці «Classes». Порожня онтологія містить один клас з ім'ям **THING**, а всі створювані надалі класи є підкласами цього класу.

Створюючи новий клас, користувач має вказати, підкласом якого класу він є. Таким чином, ієрархія класів, що відповідає відношенню «є підкласом», задається безпосередньо та явно в процесі створення нових класів онтології. Якщо між класами існують інші типи ієрархічних відношень (приміром, відношення хронологічної впорядкованості), то вони задаються за допомогою інших виразних можливостей.

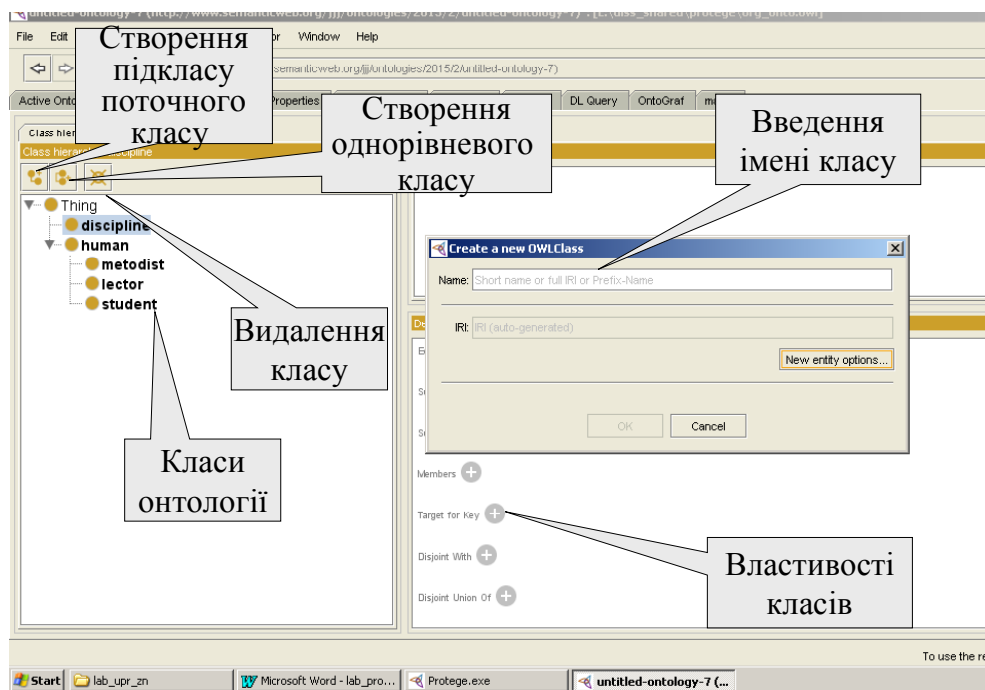


Рис. 3.2. Створення класів у Protégé

Після того, як класи створені, можна фактично здійснити опис деяких їхніх властивостей. Наприклад, у створеній раніше таксономії класів можна вказати, що підкласи «студент» і «викладач» не перетинаються. Для цього серед властивостей класів потрібно вибрати знак «+» біля властивості «Disjoint with», а потім у вікні, що з'явилася, з ієрархії класів вибрати мишею (при натиснутій клавіші «Ctrl») ті класи, що не перетинаються.

Крім того, у властивостях класу можна явно вказати, що класи еквівалентні («Equivalent To»). Це дає змогу використовувати різні синоніми на позначення груп об'єктів предметної області.

Вкладка «Annotation» дає змогу вводити й зберігати структуровані пояснення про клас природною мовою. Властивості анотації можуть використовуватися для додавання інформації (метадані – дані про дані) для класів, окремих індивідів і властивостей об'єктів/типів даних. Крім стандартних видів анотації, користувач може створювати нові підкласи й потім здійснювати опис їх значень.

Створення властивостей об'єктів

Щоб здійснити опис екземплярів класів, у Protege 4.3 використовується два основні типи властивостей: властивості об'єктів (ObjectProperties) і властивості даних (DataProperties).

Властивості об'єктів відображають відношення між екземплярами класів.

Наприклад, властивість об'єктів «вивчає» пов'язує «Іванов» (екземпляр класу «студент») і «Матаналіз» (екземпляр класу «дисципліна»).

Властивості даних дають змогу зв'язати екземпляри класів з константами різних типів. Наприклад, властивість даних «рік_народження» зв'язує «Іванов» (екземпляр класу «студент») і константу типу Integer «1994».

Властивості об'єкта й властивості даних можуть бути позначені як властивості Анотації.

Для того, щоб створити властивості об'єктів, потрібно скористатися вкладкою «ObjectProperties». У Protégé 4.3 за замовчуванням, автоматично створюються властивості об'єктів верхнього рівня (TopObjectProperty) за аналогією до класу THING.

У властивостей об'єктів можуть бути підвластивості. Наприклад, властивість «Використовувати» має підвластивості «Використовувати на практиці» й «Використовувати теоретично». Зазначимо: хоча аналогічно можна створювати підвластивості властивості в DataType, однак це не означає можливість змішувати й зіставляти властивості об'єктів і властивості типів даних з їх підвластивостями. Наприклад, неможливо створити властивість об'єкта, що є властивістю підвластивості DataType й навпаки. Це безпосередньо виходить з того, що властивості об'єктів зв'язуються з двома екземплярами класів, а властивості даних – з одним.

Створюючи нову властивість об'єкта, можна явно схарактеризувати такі особливості цієї властивості, як функціональність

та інверсна функціональність, транзитивність чи симетричність/асиметричність, рефлексивність чи іррефлексивність.

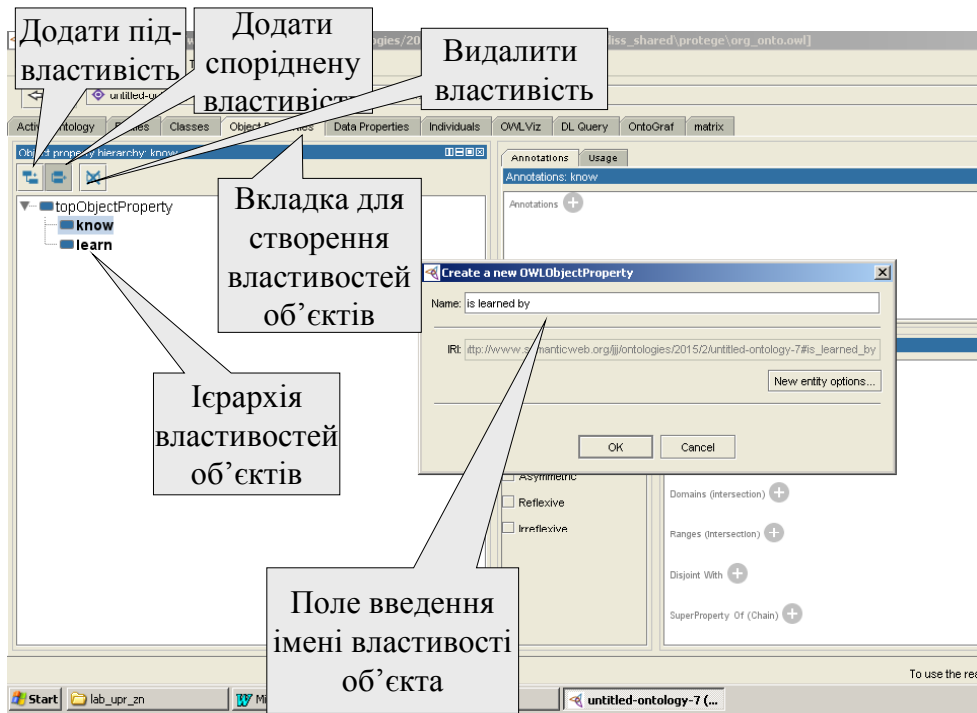


Рис.3.3. Створення властивостей об'єктів у Protégé

Кожна властивість об'єкта x може мати відповідну зворотну властивість y , таку, що, якщо властивість x зв'язує екземпляр A з екземпляром U , то y зв'язує екземпляр B з екземпляром A .

Наприклад, для властивості «learn» (вивчає) зворотною є властивість «is_learned» (вивчатися). Це означає, наприклад, якщо «learn» зв'язує «Іванов» (екземпляр класу «студент») з «Матаналіз» (екземпляр класу «дисципліна»), то зворотна йому властивість «is_learned» зв'язує «Матаналіз» (екземпляр класу «дисципліна») з «Іванов».

Щоб вказати при створенні онтології, що одна властивість є для іншої зворотною, необхідно, вибравши потрібну властивість об'єкта, у вікні «Description» обрати кнопку «+» біля напису «Inverse Of», а потім у вікні ієрархії властивостей об'єктів відзначити інверсну для нього властивість.

Дуже ефективним інструментом є використання функціональної характеристики властивості об'єкта: якщо властивість є функціональною, то кожен екземпляр класу може бути зв'язаний цією властивістю не більше, ніж з одним екземпляром класу. Наприклад, у людини може бути лише одна мати чи один безпосередній начальник; у конкретний момент часу людина проживає в якійсь одній країні тощо. Якщо ж один екземпляр класу зв'язаний цією властивістю з декількома екземплярами, щодо яких явно зазначено, що вони не ідентичні, то це є протиріччям.

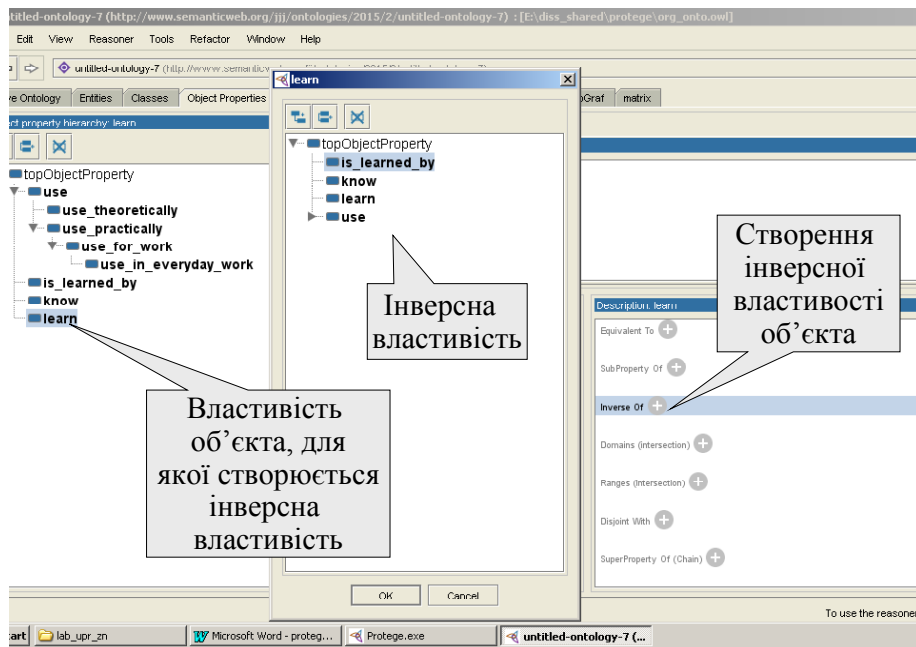


Рис.3.4. Створення інверсної характеристики властивостей об'єктів у Protégé

Для того, щоб позначити властивість об'єкта як функціональну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Functional».

Якщо властивість є зворотною щодо функціональної властивості, то це означає, що властивість є зворотно функціональною. Це значить, що для конкретного екземпляра може бути багато екземплярів, зв'язаних з ним цією властивістю, але кожний з них може зв'язуватися тільки один раз. Для того, щоб позначити властивість об'єкта як зворотно-функціональну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Inverse functional».

У вікні «Description» можна виділити область визначення й область значення властивості об'єкта: «Domain (intersection)» і «Range (intersection)» дають змогу вказати, до яких класів можуть належати екземпляри, що зв'язуються відношенням. Наприклад, властивість об'єкта «проживає» зв'язує екземпляр класу «людина» з екземпляром класу «країна».

Транзитивні властивості (transitive) об'єктів: якщо властивість x , що зв'язує екземпляр A з екземпляром B , а екземпляр B – з екземпляром C , є транзитивною, тоді ця ж властивість x зв'язує екземпляр A з екземпляром C . Приклад транзитивної властивості – «more_qualified» (більш кваліфікований), якою можуть бути зв'язані екземпляри класу людина: якщо Петров кваліфікованіший за Іванова, а Сидоров – кваліфікованіший від Петрова, то Петров кваліфікованіший за Іванова.

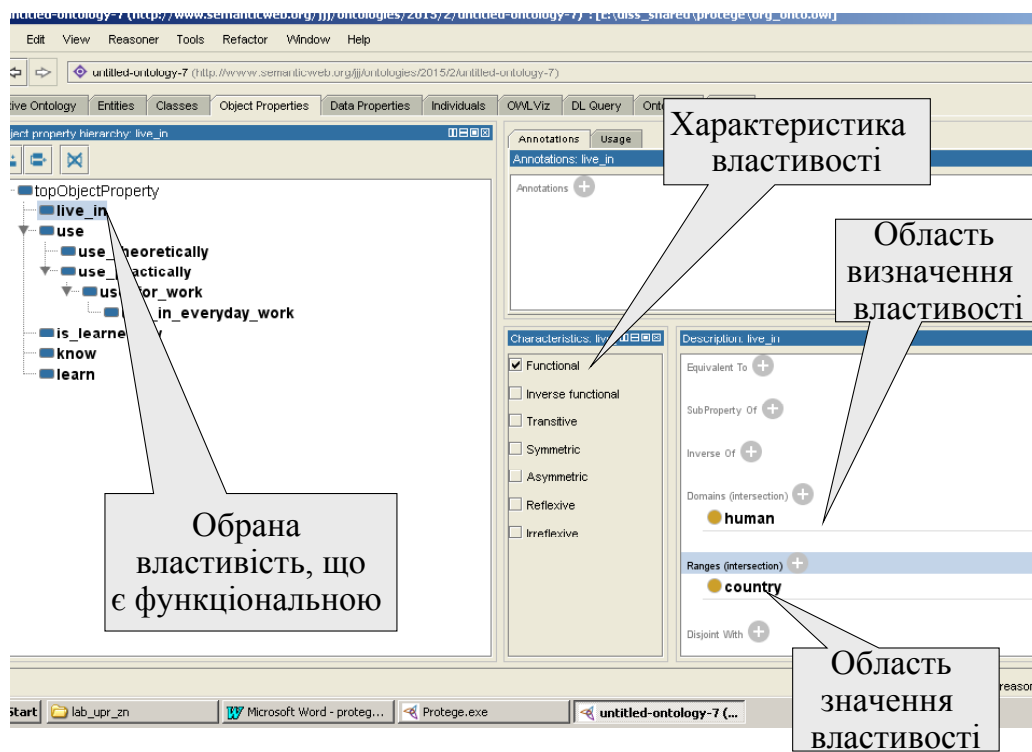


Рис.3.5. Створення функціональної характеристики властивостей об'єктів

Для того, щоб позначити властивість об'єкта як функціональну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Transitive».

Симетричні властивості (Symmetric) об'єктів: якщо властивість x , що зв'язує екземпляр A з екземпляром B , є симетричною, тоді ця ж властивість x зв'язує екземпляр B з екземпляром A . Приклад симетричної властивості – «more_qualified», якою можуть бути зв'язані екземпляри класу людина: якщо Петров кваліфікованіший за Іванова, то Іванов не може бути кваліфікованіший за Петрова.

Для того, щоб позначити властивість об'єкта як симетричну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Symmetric».

Асиметричні властивості (Asymmetric) об'єктів: якщо властивість x , що зв'язує екземпляр A з екземпляром B , є асиметричною, то ця ж властивість x не зв'язує екземпляр B з екземпляром A . Приклад асиметричної властивості – «familiar» (знайомий), яким можуть бути зв'язані екземпляри класу людина: якщо Петров знайомий з Івановим, то Іванов знайомий з Петровим.

Для того, щоб позначити властивість об'єкта як симетричну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Asymmetric».

Рефлексивні властивості (Reflexive) об'єктів: властивість x є рефлексивною, якщо вона зв'язує кожен екземпляр A з ним самим.

Приклад рефлексивної властивості – «familiar» (знайомий), якою можуть бути зв'язані екземпляри класу людина: кожна людина знайома сама з собою.

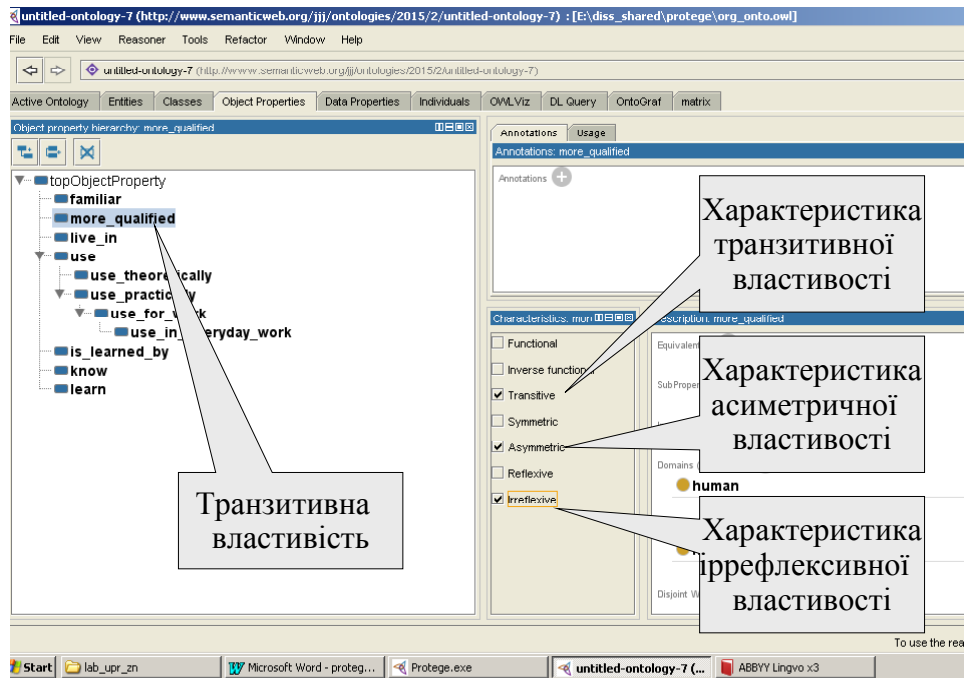


Рис.3.6. Створення асиметричної та іррефлексивної характеристик властивостей об'єктів у Protégé

Для того, щоб позначити властивість об'єкта як рефлексивну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Reflexive».

Іррефлексивні властивості (irreflexive) об'єктів: властивість x є іррефлексивною, якщо вона не може зв'язувати жоден екземпляр A з ним самим. Приклад іррефлексивної властивості – «more_qualified», якою можуть бути зв'язані екземпляри класу людина: жодна людина не може бути кваліфікованішою за себе саму.

Для того, щоб позначити властивість об'єкта як іррефлексивну, необхідно вибрати цю властивість в ієрархії властивостей, а у вікні «Characteristics» поставити позначку коло характеристики «Irreflexive».

Створення властивостей даних

За допомогою вкладки Data Properties можна відобразити властивості даних – відношення, що зв'язують екземпляри класів онтології не з іншими екземплярами класів, а з конкретними значеннями різних типів. Наприклад, для екземпляра класу «людина» це можуть бути такі властивості, як прізвище, ім'я, рік народження тощо. На відміну від Protege-фрейму, у Protege версії 4.1 і вище екземпляри отримують при створенні тільки ім'я, а згодом при здійсненні опису в них з'являються властивості.

Усі властивості даних є підкласами класу TopDataProperty. Створювати його підкласи можна у вікні «Data property hierarchy».

Наприклад, можна створити властивість даних з ім'ям «name»: у вікні «Description» указати його область визначення (кнопка «+» біля напису «domain (intersection)») – клас «human» і область значення (кнопка «+» біля напису «ranges») – тип String.

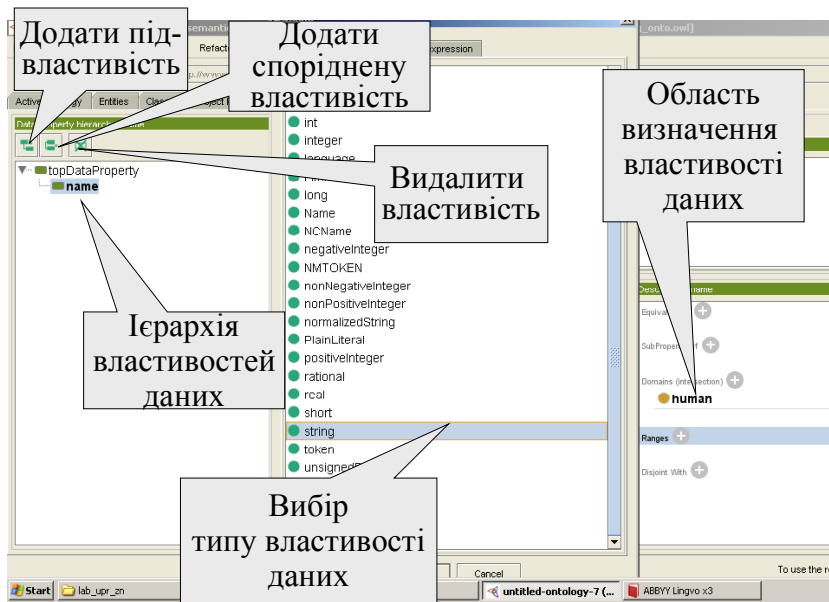


Рис.3.7. Створення властивостей даних у Protégé

Крім використання визначеного набору типів даних, можна також спеціалізувати використання властивостей даних шляхом вказівки на обмеження для можливих значень. Наприклад, це застосовують для вказівки на діапазон чисельних значень.

Властивість даних називається функціональною, якщо для кожного екземпляра вона може мати тільки одне значення. Наприклад, людина може мати тільки один рік народження.

Для того, щоб позначити властивість даних як функціональну, необхідно вибрати цю властивість в ієрархії властивостей даних, а у вікні «Characteristics» поставити позначку коло характеристики «Functional».

Створення екземплярів

Для створення екземплярів класів у Protégé 4.3 використовується вкладка «Individuals».

Для цього потрібно у вікні «Class hierarchy» вибрати клас, до якого належить створюваний екземпляр, а потім у вікні «Members list» створити новий екземпляр, указавши його ім'я.

Після цього можна здійснити опис значення властивостей екземпляра, використовуючи зв'язані з його класом властивості об'єктів і властивості даних.

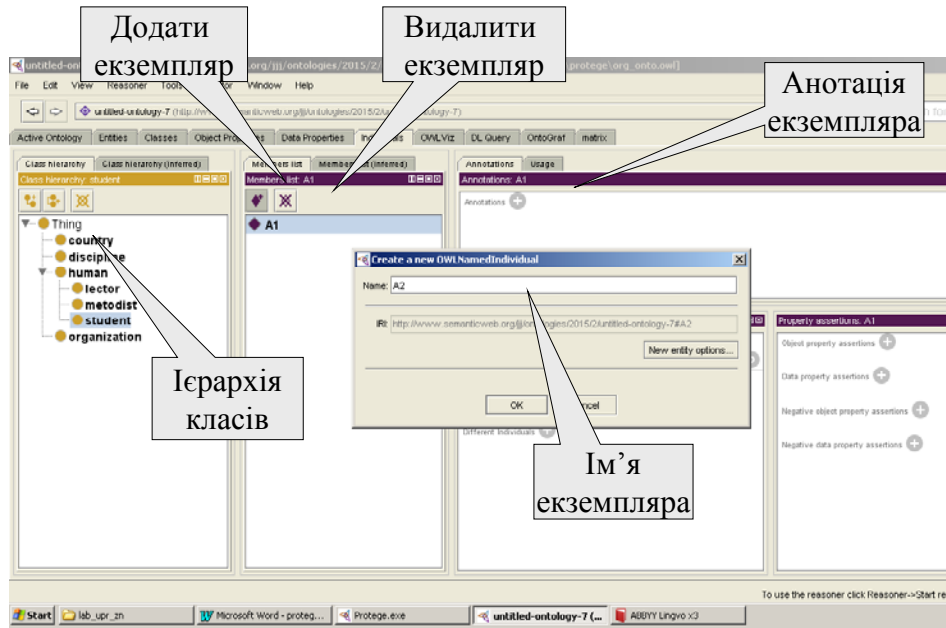


Рис.3.8. Створення екземплярів класів у Protégé

У вікні «Description» можна, вибравши знак «+» коло напису «Same individuals as», указати, що два екземпляри одного класу є ідентичними. Це дає змогу використовувати синоніми для іменування екземплярів.

Можна також явно вказати, що два екземпляри є різними. Для цього у вікні «Description» необхідно, вибравши знак «+» коло напису «Different individuals», указати, що два екземпляри одного класу не є ідентичними. Це дає можливість відстежувати протиріччя при створенні екземплярів.

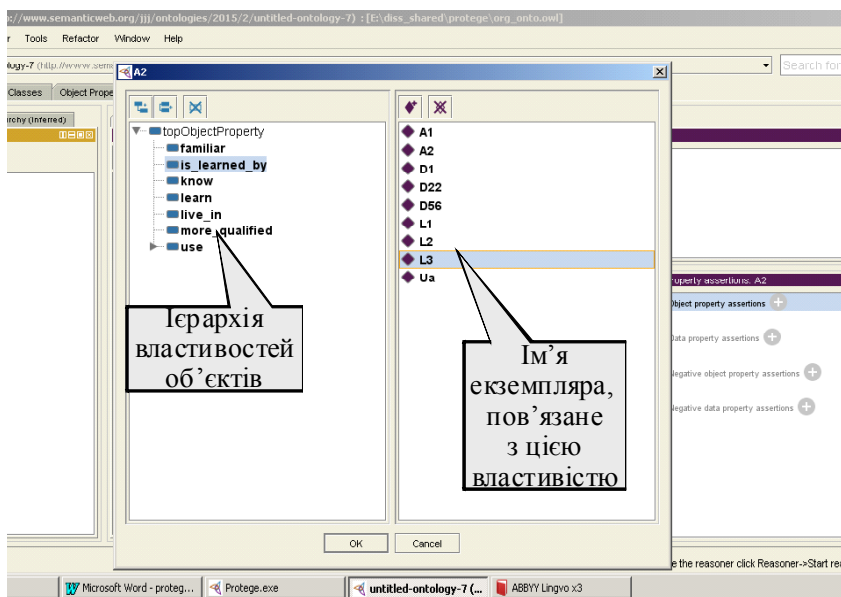


Рис.3.9. Створення властивостей об'єктів для екземплярів класів у Protégé

Візуалізація онтологій в Protégé

За допомогою вкладок «OntoViz» і «OntoGraph» можна візуалізувати ієрархію класів створеної онтології.

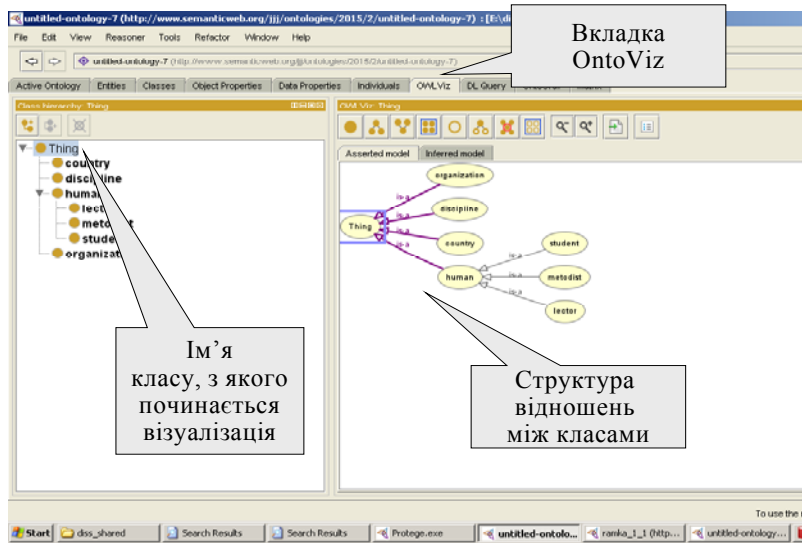


Рис.3.10. Візуалізація ієрархії класів у Protégé за допомогою OntoViz

Щоб побудувати візуалізацію онтології за допомогою вкладки «OntoGraph», потрібно в ієрархії класів онтології явно відзначити ті класи, зв'язки між якими потрібно візуалізувати. У цьому режимі візуалізуються не тільки ієрархічні відношення між класами, а й зв'язки, виражені через властивості об'єктів.

Для створення онтологій у редакторі Protégé потрібно явно надати відомості, які стосуються основних понять Про, що моделюється, й відношень між ними. Використання Protégé дає змогу створювати онтології мовою OWL, що забезпечує інтероперабельне подання знань і можливість їх повторного використання для інтелектуального аналізу інформації.

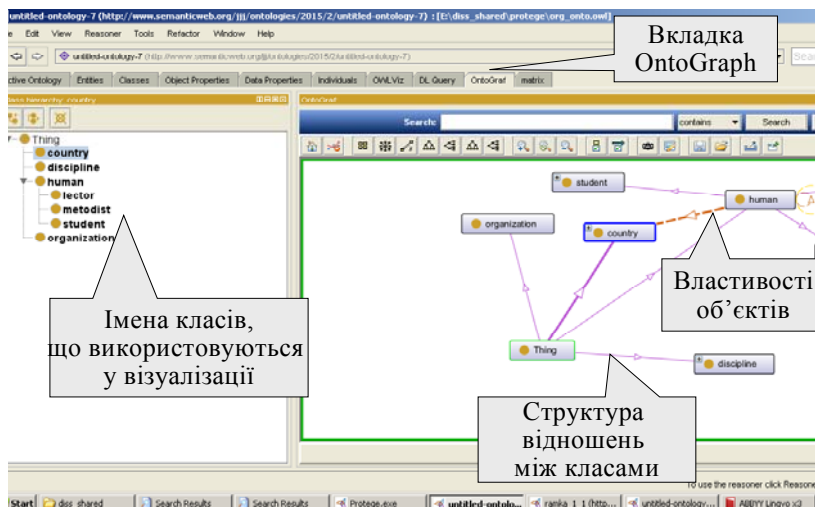


Рис.3.11. Візуалізація класів у Protégé за допомогою OntoGraph

Побудова запитів до онтологій

Для того, щоб задати запит SPARQL [46] в Protégé, необхідно відкрити відповідну вкладку з назвою SPARQL Query.

SPARQL – мова запитів до даних, представлених по моделі RDF, а також протокол для передачі цих запитів і відповідей на них.

SPARQL запит формується із:

- префіксних оголошень, наприклад:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

- джерела запиту;

- пункт результату, наприклад: SELECT ?subject ?object ;

- критерій запиту, наприклад: WHERE { ?subject rdfs:subClassOf ?object } ;

- модифікаторів запиту.

Префіксні оголошення служать для скорочення універсальних ідентифікаторів ресурсу. Джерела запиту визначають, які RDF графи запитуються. Пункт результату повертає набір даних (вибірку), які задовольняють умові. Критерії запиту визначає, що запитувати в базовому наборі даних. Модифікатори запиту обмежують, упорядковують, і інакше перетворюють результати запиту.

Пункт результату містить одну або кілька змінних, ім'я яких обирається користувачем, перед ім'ям змінної обов'язково ставлять символ "?". Кількість змінних збігається із кількістю стовпчиків в таблиці, що є результатом запиту.

В критерії запиту задається відношення між двома змінними, об'єктами, класами чи ін. Якщо таких критеріїв декілька, між ними ставиться крапка. Відношення типу: "Є підкласом класу" записується як "rdfs:subClassOf". Так, наприклад, щоб знайти всі підкласи класу Особа, слід ввести наступний запит:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?subject ?object
```

```
WHERE { ?subject rdfs:subClassOf ?object }
```

В даному запиті використовується PREFIX oo, що вказує власне на саму онтологію, відповідно всі об'єкти цієї онтології записуються з префіксом "oo:", що вказує, що вони належать саме цій онтології. Сам префікс може бути обраний користувачем і мати обране ім'я. Його можна знайти в вкладці Active Ontology, Ontology IRI, там же його можна і змінити. Для того, щоб можна було здійснювати SPARQL запити, префікс має закінчуватись на ".owl". Protégé сам генерує такий префікс при створенні онтології, та при такому генеруванні необхідного для запитів закінчення префікс скоріш за все не матиме.

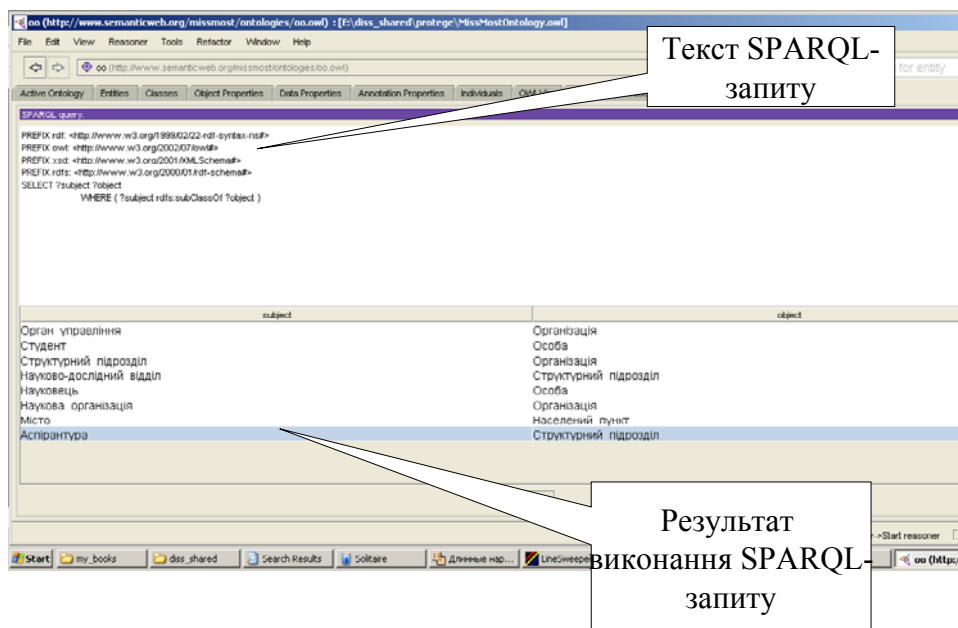


Рис.3.12. Результат виконання SPARQL-запиту в Protege

Для того, щоб знайти всі екземпляри деякого класу, використовується відношення: "rdf:type", тобто "є екземпляром". Так, наприклад, щоб знайти всі екземпляри класу "Науковці", можна використати наступний запит:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX oo: <http://www.semanticweb.org/missmost/ontologies/oo.owl#>
SELECT ?subject
WHERE { ?subject rdf:type oo:Науковець }
    
```

Для того, щоб вказати властивість в SPARQL запиті слід вказати префікс онтології і ім'я властивості. Так наприклад, для того, щоб знайти всіх науковців, що працюють в Інституті програмних систем НАН України, можна використати запит:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX oo: <http://www.semanticweb.org/missmost/ontologies/oo.owl#>
SELECT ?x ?y
WHERE { ?x rdf:type oo:Науковець.
       ?x oo:Працює_в oo:Інститут_програмних_систем_НАН_України
}
    
```

Також в SPARQL запитах можна задати, так звані, "не строгі" запити, з допомогою додавання до відношень символу "*". Тобто, додавши до

subClassOf символ "*", можна отримати не лише підкласи класу, а й підкласи підкласів тощо. Щоб зробити це можна використати запит:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX oo: <http://www.semanticweb.org/missmost/ontologies/oo.owl#>
```

```
SELECT ?subject
```

```
WHERE { ?subject rdfs:subClassOf* oo:Організація }
```

У порівнянні виразних можливостей мови вбудованих запитів Semantic Media Wiki значно поступається мові запитів SPARQL, але використання SPARQL потребує від користувачів значно більших знань.

Обидві мови дозволяють робити доволі складні запити, з можливістю сортувати, обмежувати, впорядковувати та виконувати інші операції над результатом, але мова вбудованих запитів SMW дає можливість працювати тільки з спеціальними категоріями та властивостями самої Semantic MediaWiki, а також категоріями та властивостями, які створили користувачі конкретної інсталяції Semantic MediaWiki. Таким чином вона дозволяє оперувати тільки тими даними, які розміщені на одному Wiki-ресурсі. Мова SPARQL, навпаки, з самого початку була орієнтована на інтероперабельність. Вона дозволяє використовувати різні онтології та їх поєднання, а також здійснювати операції над онтологіями.

В запитах до Semantic MediaWiki можуть використовуватися категорії та семантичні властивості, а в запитах до онтологій – класи та властивості, які поділяються на object properties та data properties.

Потужні можливості SPARQL з'являються завдяки можливості використовувати не тільки визначені користувачем властивості, але й властивості з інших онтологій, які позначаються різними префіксами. Також в SPARQL є можливість конструювати нові онтології. Форма CONSTRUCT вказує граф, який буде повернуто за допомогою змінних для заміни шаблону запиту, наприклад, у наступному прикладі, який повертає граф, що містить відомості щодо того, що особа на прізвище "Іванов" останніх двох людей, впорядкованих за алфавітом з даного URI (результат в RDF-графі не впорядкований, це граф і тому порядок триплетів не має значення).

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
CONSTRUCT { <http://example.org/person#Іванов> foaf:knows ?x }
```

```
FROM <http://example.org/foaf/people>
```

```
WHERE { ?x foaf:name ?name }
```

```
ORDER BY desc(?name)
```

```
LIMIT 2
```

Також може використовуватися форма DESCRIBE, що повертає інформацію про відповідні ресурси у формі графа RDF.

Для того, щоб користуватися цими можливостями SPARQL потрібно оволодіти більш складним синтаксисом в порівнянні з вбудованою мовою

запитів Semantic MediaWiki, а також витратити час на дослідження вже існуючих онтологій, присутніх у відкритому доступі через Web.

3.3. Напрямки семантизації Wiki

Семантичні Wiki здатні автоматизовано обробляти дані з чітко визначеною семантикою, а це дає змогу істотно розширити функціональність таких систем. Наприклад, можна обробляти типізовані посилання між Wiki-статтями й усередині статей [42]. У той час як звичайні Wiki-сторінки можуть містити тільки однакові посилання на інші сторінки, у семантичних Wiki можна явно вказувати значення (семантику) такого посилання, яка відповідає відношенню між тими поняттями, яким відповідають ці сторінки. Приміром, з Wiki-сторінки “Україна” можна не просто послатися на Wiki-сторінку “Київ”, а вказати в цьому посиланні, що Київ є столицею України.

Структуровані дані зберігаються в семантичних Wiki-ресурсах або безпосередньо в тексті сторінки за рахунок розширення Wiki-розмітки (зокрема, так зберігає метадані Semantic MediaWiki), або ж окремо в спеціальній формі (приміром, в Ontowiki).

Посилання між статтями містять у своєму імені інформацію про тип зв'язку. Приміром, у KiWi можна пов'язувати структуровані дані за допомогою засобів RDF, а потім співвідносити RDF-терміни з фрагментами тексту.

Більшість семантичних Wiki дають змогу здобувати додаткову інформацію за посиланням, що підтримує базовані на семантиці способи навігації та семантичний пошук. Так, користувач може побудувати багатокритеріальний запит, подібний до SPARQL, а використання семантичних анотацій дасть змогу відфільтрувати результати пошуку.

Семантичні Wiki орієнтовані на використання форматів та технологій Semantic Web для подання й збереження знань. Приміром, багато з них підтримують імпорт/експорт знань у RDF і OWL, а для користувацьких запитів до Wiki слугує SPARQL [30], що забезпечує підтримку логічного виведення в системі.

У перших семантичних Wiki (Platypus Wiki і Rhizome Wiki) RDF-дані презентувалися як текст, що вільно редагується й не пов'язується з неструктурованим контентом.

Найчастіше семантизація розмітки досягається шляхом збагачення посилань текстовою анотацією, що містять опис їх значень. Приміром, у статті про Київ, посилання на статтю Україна можна іменувати як «is capital» чи «located in». Таке анотування дає змогу значно поліпшити візуалізацію інформації (демонстрація контекстуальної інформації),

навігацію (доступ до релевантної інформації), пошук (пошук за контекстом, а не лише за текстом).

Для семантизації Wiki-ресурсів часто використовують онтології. Є два види взаємозбагачення онтологій і Wiki: системи, у яких Wiki використовуються для побудови онтологій, і системи, що використовують онтології для роботи Wiki. Іноді Wiki-системи використовуються як зовнішній інтерфейс для колаборативної розробки онтологій (Semantic MediaWiki, PlatypusWiki). В іншому підході онтології дають змогу поліпшити функціонування самої Wiki-системи (приміром, IkeWiki, SweetWiki, SWIM).

3.4. Реалізації семантичних Wiki

На сьогодні існує велика кількість Wiki-двигунів, які забезпечують обробку інформації на семантичному рівні. До них відносяться:

- COW
- Capital Market Wiki
- IkeWiki
- KawaWiki
- KendraBase
- Makna
- OntoWiki
- OpenRecord
- PlatypusWiki
- POWL
- Rhizome
- Rise
- Semantic MediaWiki
- Semantic Wiki as an Integrated Content and Metadata Management System
- SeMediaWiki
- SemPerWiki
- SemWiki
- SweetWiki
- Wekiwi
- WikiOnt
- WikSAR

Одна з найбільш поширених Wiki-систем – **Semantic Media Wiki** (SMW) – є розширенням двигуна MediaWiki й базується на використанні семантичних анотацій. Вона дає змогу користувачам додавати власні елементи до Wiki-розмітки. Wiki-посилання супроводжуються семантичними анотаціями мовою OWL DL у форматі OWL/RDF. Розмітка здійснюється за допомогою Web-інтерфейсу. Кожна Wiki-стаття відповідає одному онтологічному елементу, а кожна

анотація у статті містить твердження тільки про один елемент. Таке обмеження має ключове значення для експлуатації, оскільки знання використовуються повторно, й кінцеві споживачі повинні знати, звідки ця інформація була отримана. Таким чином, SMW створює додатковий семантичний шар, що дає можливість системі виглядати як Wiki, а функціонувати як Linked Data.

Основні семантичні поняття у Semantic MediaWiki – це *категорії*, що дозволяють користувачам класифікувати Web-сторінки (присутні й у звичайних Wiki-ресурсах), *семантичні властивості*, які дозволяють встановлювати зв'язки елементів контенту Wiki-сторінок з іншими сторінками Wiki ресурсу та з даними (дата, число, текстовий рядок, географічні координати тощо), визначаючи семантику цих зв'язків, та *семантичні запити*, в яких можуть використовуватися категорії та семантичні властивості. Результати семантичних запитів вставляються у відповідні Wiki-сторінки та автоматично оновлюються, якщо змінюється контент тих сторінок, з яких вони здобувають інформацію. Це забезпечує цілісність та актуальність Wiki-ресурсу, побудованого за допомогою Semantic MediaWiki.

У Semantic MediaWiki передбачена можливість трансформації цієї семантичної інформації в онтологічне подання.

IkeWiki використовує онтології для поліпшення самої системи. У системі реалізоване логічне виведення для підтримки користувача у виконанні завдань. Система пропонує і WYSIWYG-редактор, що призначений для користувачів, які не мають досвіду роботи з Wiki-редактором.

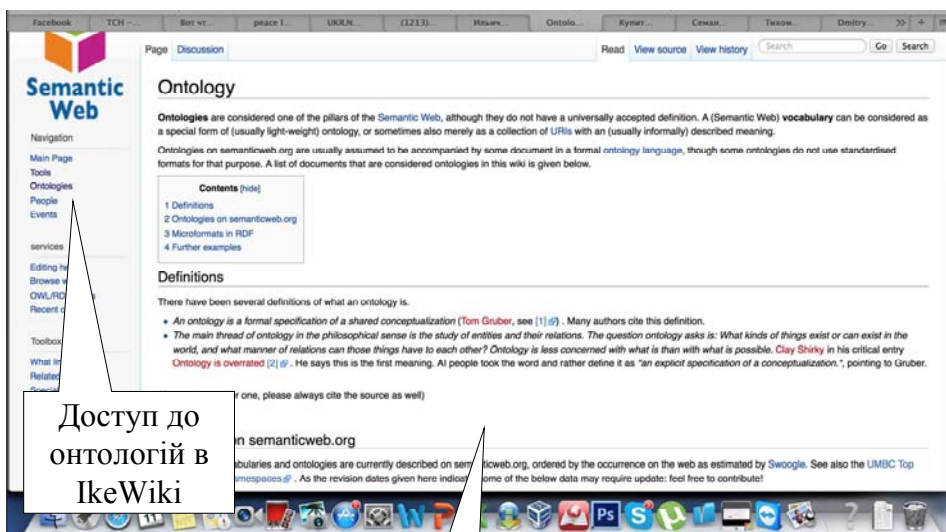


Рис.3.13. Інтерфейс IkeWiki

Платформа IkeWiki написана мовою Java. Дані зберігаються у СКБД Postgres, однак є розмежування тексту й структури документів. За необхідності, вони повертаються користувачеві в зручному форматі XML (для тексту) чи RDF (для структури).

База знань у системі презентована RDF фреймворком Jena. Частина RDF-сховища є SPARQL-двигуном, що забезпечує пошук у системі й базі.

Для анотацій є три види редакторів. Редактор метаданих дає змогу заповнювати текстуальні метадані, що стосуються сторінки (дані Dublin Core чи RDF-коментарі). Редактор типу дає змогу асоціювати сторінки з одним чи декількома типами, упровадженими в систему. Редактор посилань забезпечує керування анотаціями посилань. У IkeWiki доступні анотації, що визначаються логічним висновком. Приміром, якщо посилання зі статті «Kyiv» до «Ukraine» іменувати як «capitalOf», то система автоматично асоціює тип «Capital» зі сторінкою, що містить опис Києва, і цей тип не може бути вилучений користувачем.

IkeWiki більше не розвивається як самостійна система. Її продовженням стала система **KiWi**, що успадкувала більшість характеристик. Можна зробити KiWi-систему частиною Linked Open Data, що, так само, уможлиблює інтеграцію контенту з іншими сервісами Semantic Web.

Серед особливостей системи KiWi слід назвати сторінку користувача Dashboard. Вона дає змогу відстежувати й застосовувати в роботі: потік діяльності (приміром, відновлення елементів контенту), рекомендації (пропонує користувачеві інший контент за допомогою різних рекомендаційних алгоритмів), історію (список елементів, які переглядав чи редагував користувач), теги (список тегів, що використовувалися користувачем). Крім цього, Dashboard – це і профіль користувача, і список його друзів.

Система **OntoWiki** призначена для розробки баз знань і спирається на подання даних у форматі RDF. Для машинної обробки система підтримує різні RDF-формати, RDFa, Linked Data і SPARQL-інтерфейси. Знання в системі презентовані за допомогою «інформаційної карти», збагаченої зручними інтерфейсами для візуалізації й редагування контенту (WYSIWYG редактор для RDF, контроль версій, статистика, підтримка співтовариства тощо). Кожен вузол, представлений сторінкою системи, в інформаційній карті зв'язаний з відповідним цифровим джерелом (рис.3.14).

OntoWiki спроектована для роботи з онтологіями будь-якого обсягу й намагається підтримувати створення онтологій «з нуля». Ця платформа не тільки дає змогу застосовувати частково визначені шаблони з репозиторію шаблонів, а й створювати власні. Вона

підтримує колаборативну розробку шляхом відстеження змін, а також завдяки можливості коментувати й обговорювати кожну частину бази знань, оцінюючи й обмежуючи кількість контенту й заохочуючи користувальницьку активність.

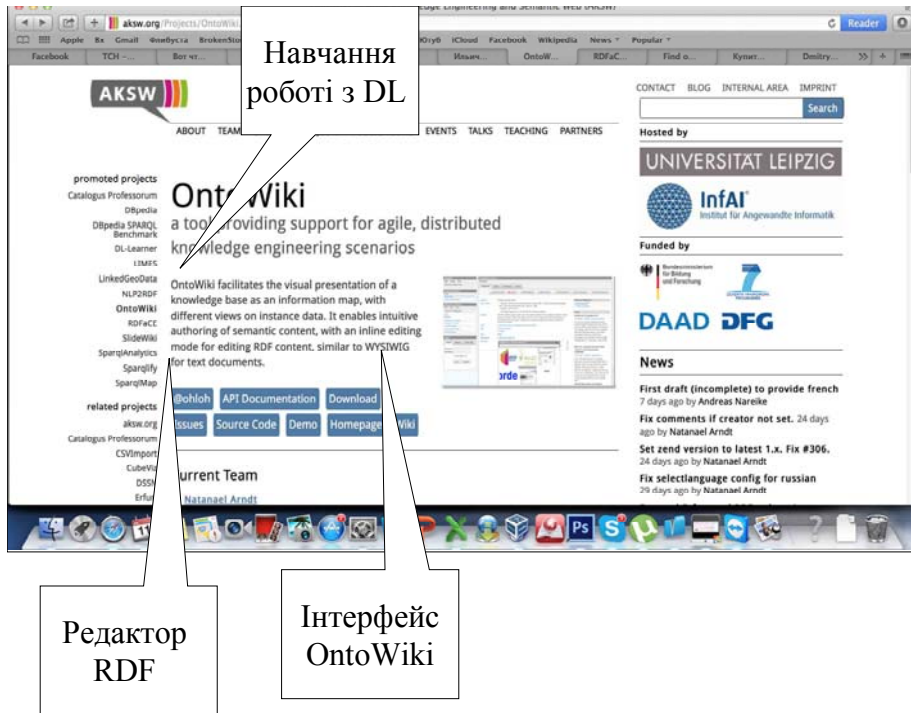


Рис.3.14. Інтерфейс OntoWiki

Семантичний пошук системи презентований як пошук у локальному RDF-сховищі за допомогою SPARQL-запитів. Для навігації пропонується кілька способів: таксономічний та ієрархічний пошук, фасетний пошук і текстовий пошук.

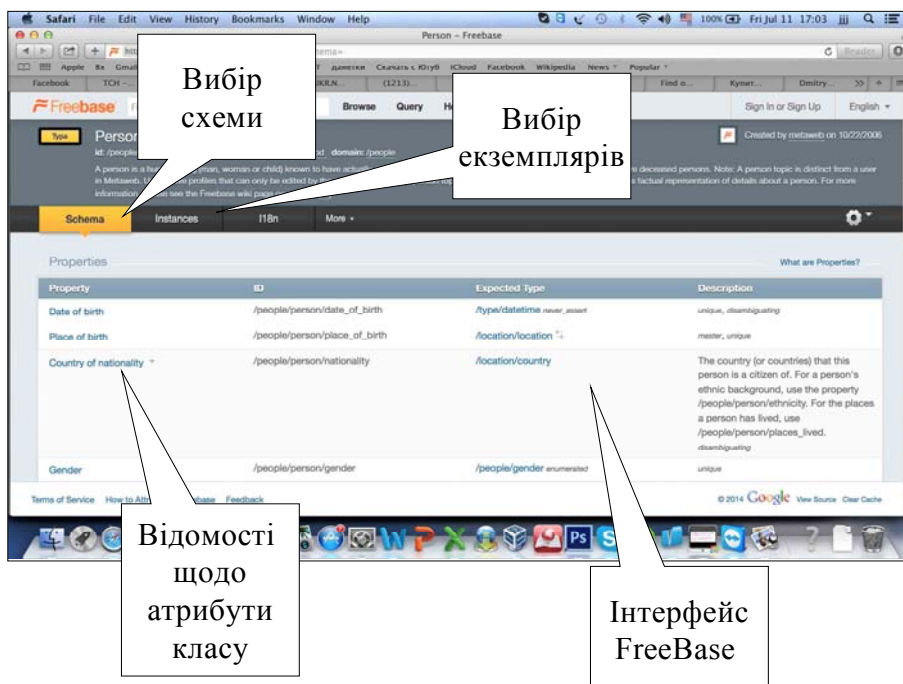


Рис.3.15. Інтерфейс FreeBase

Система **Freebase** дає змогу користувачам визначати власні схеми для моделювання різних типів даних і керувати зв'язаною структурованою інформацією. Це велика колаборативна база знань (рис.3.15).

Платформа намагається інтегрувати різні підходи до формування бази знань. Крім Wiki-стилю, вона дає змогу збирати інформацію автоматично з різних ресурсів, таких, як Wikipedia і MusicBrainz. Деякі сутності можуть бути змодельовані й додані до системи людьми. Однак у ній не просто створювати власні типи. Всі атрибути повинні мати типи й межі, визначені самою системою. Web-інтерфейс надає користувачам зручну можливість шукати, приймати, редагувати й організувати інформацію. Недоліком платформи є складність приєднання до зовнішнього ресурсу.

4. Створення та менеджмент Wiki-ресурсів у MediaWiki

Одним за найбільш поширених сьогодні Wiki-двигунів є MediaWiki. Ця програма дозволяє створювати та редагувати різноманітні Wiki-ресурси, надає користувачам зручний та інтуїтивно зрозумілий інтерфейс та підтримує більшість функцій, необхідних для зручної колективної роботи. Це потребує встановлення на сайті відповідного програмного забезпечення.

MediaWiki забезпечує інтерфейс для роботи з базою Wiki-сторінок, надаючи можливості для обробки текстового та мультимедійного контенту. Можна обробляти текст як у власному форматі Wiki, так і у форматах HTML та TeX (для формул). Щоб користувачі мали можливість використовувати на Wiki-сторінках графіку, аудіофайли та відео, надаються засоби для завантаження відповідних файлів. У системі підтримується розмежування прав доступу до адміністрування системи і доступу до контенту. Використання різноманітних плагінів дозволяє користувачам розширювати функції системи відповідно до власних потреб – приміром, відображати математичні формули, експортувати чи імпортувати різні формати даних. До переваг MediaWiki можна віднести:

- безкоштовність програмного забезпечення;
- можливість розподіленої роботи багатьох користувачів;
- можливість розгортання як в локальній мережі установи, так і у Web.

При редагуванні та доповненні статей використовується особлива мова розмітки, яка дозволяє створювати контент користувачам, які не мають спеціальних знань в галузі інформаційно-комунікаційних технологій.

Середовище MediaWiki забезпечує контроль за версіями Wiki-статей, тобто всі правки, внесені користувачами, фіксуються та супроводжуються інформацією щодо часу, дати та автора правки. Пропонується можливість порівнювати вихідний і відредагований текст статті, при цьому будь-яка версія статті може бути відновлена.

Щоб ефективно використовувати MediaWiki, користувачам потрібно знати його базові можливості. Деякі з них досить очевидні, але інші мають певну специфіку і тому потребують більш детального розгляду.

4.1. Базові можливості системи MediaWiki

Реєстрація в системі

Якщо користувач вже зареєстрований, він може ввести свій логін та пароль, обравши пункт меню «Увійти». Якщо користувач не зареєстрований в системі, то він має можливість проглядати вже існуючі сторінки, але не може редагувати їх, коментувати або створювати нові сторінки. Тому, щоб отримати всі переваги роботи з системою, доцільно зареєструватися. Ця проста процедура не потребує багато часу.

Створення та менеджмент Wiki-ресурсів у MediaWiki

Для того, щоб зареєструватися, потрібно обрати пункт меню «Створити обліковий запис».

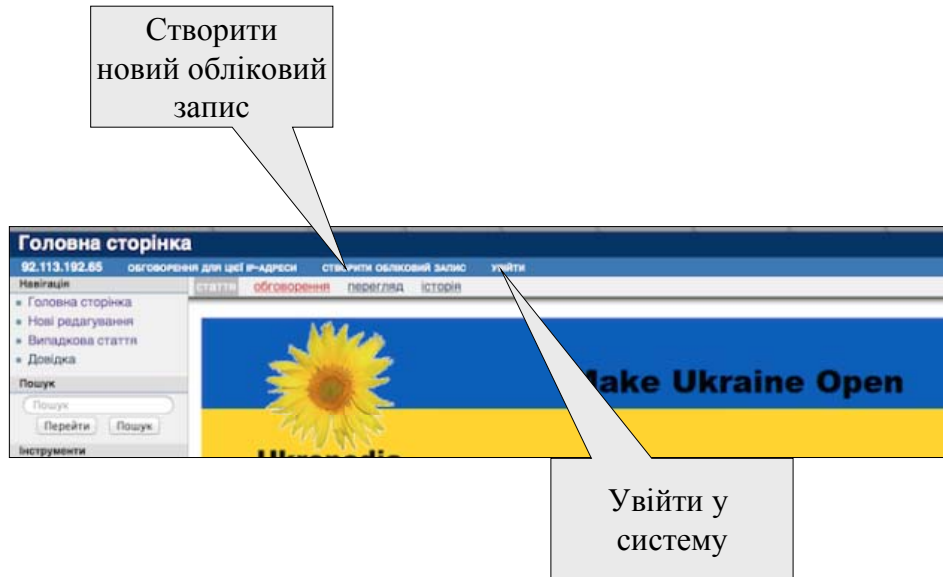


Рис.4.1. Створення облікового запису в Semantic MediaWiki.

В процесі реєстрації потрібно ввести свої логін, пароль та адресу електронної пошти. Після цього потрібно натиснути кнопку «Створити ваш обліковий запис».

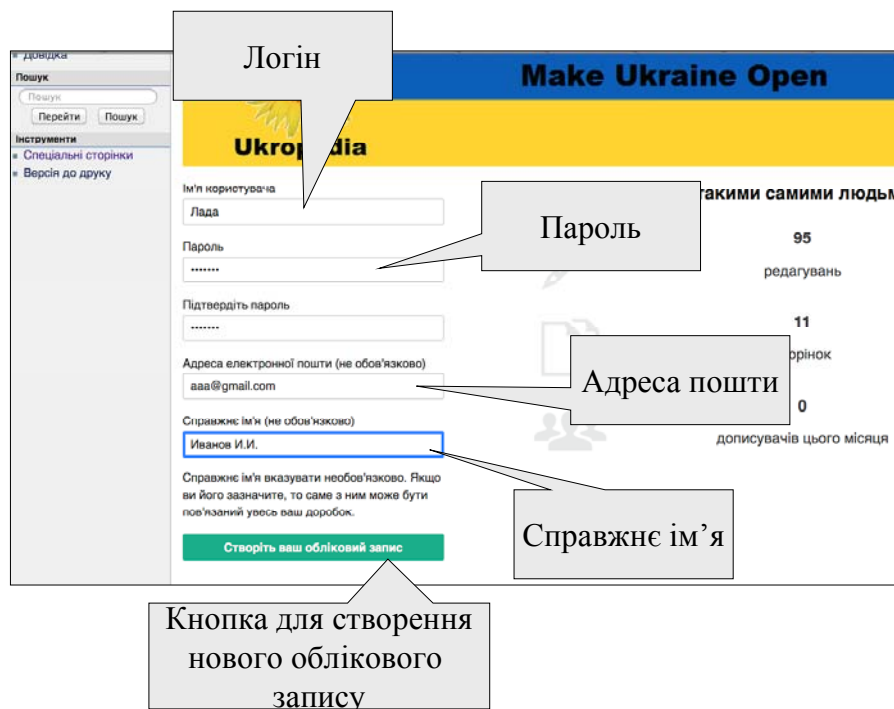


Рис.4.2. Елементи облікового запису в Semantic MediaWiki.

Вхід в систему для зареєстрованих користувачів

Після реєстрації логін та пароль вводяться для входу до системи. Якщо потрібно, можна дозволити системі запам'ятати ваш пароль, щоб надалі він вводився автоматично. Після цього потрібно натиснути кнопку «Вхід».

Для зареєстрованих користувачів стають доступними операції, пов'язані з редагуванням статей Wiki-ресурсу, завантаженням файлів, внесенням коментарів тощо.

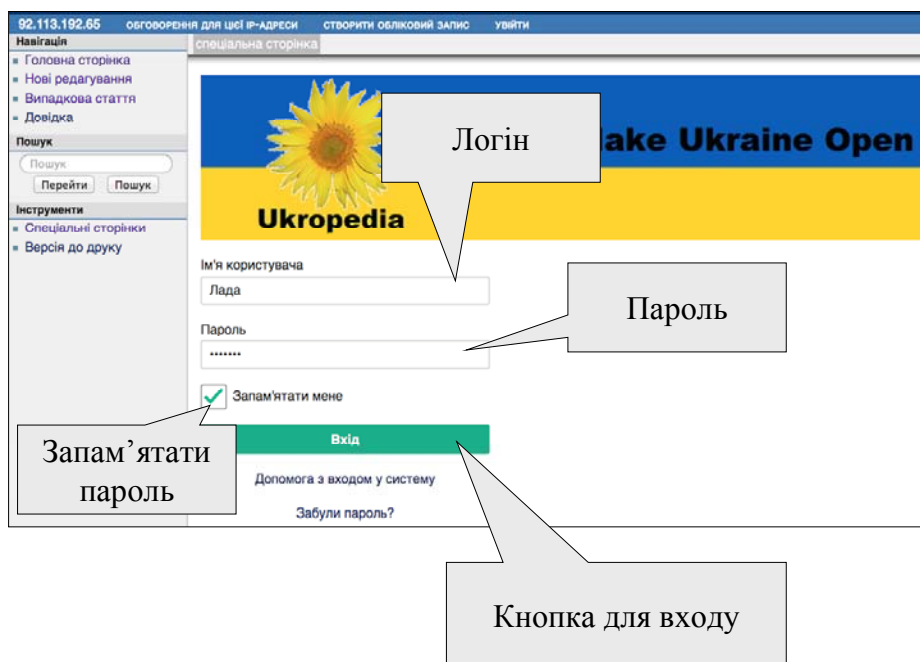


Рис.4.3. Вхід в систему зареєстрованого користувача

Пошук сторінок в Wiki-ресурсу

Якщо відоме ім'я сторінки, доступ до якої потрібно отримати, то користувачеві слід ввести це ім'я в меню "Пошук". Система дозволить обрати сторінку з тих, ім'я яких починається з введених літер або містить введений текст.

Щоб перейти на потрібну сторінку, слід обрати відповідну назва та натиснути на неї.

Створення нової сторінки

Створювати нову сторінку зареєстровані користувачі можуть різними способами. Розглянемо найпростіший з них.

Щоб створити нову сторінку Wiki-ресурсу, треба ввести в рядок браузера ім'я сайту, на якому міститься енциклопедія, й ім'я нової сторінки, яку потрібно створити.

Для цього потрібно до рядку "http://wiki.isoftware.kiev.ua/index.php/" додати ім'я сторінки.

Приміром, ми збираємося створити сторінку, присвячену В.П.Глушкову. Тоді отримаємо рядок "http://wiki.isoftware.kiev.ua/index.php/Глушков_В_П". Після цього потрібно

Створення та менеджмент Wiki-ресурсів у MediaWiki

перейти на цю сторінку, натиснувши “Enter”. Якщо така сторінка в Wiki-ресурсу відсутня, то буде запропоновано створити її, обравши пункт меню “Створити”. Слід відмітити, що використання великих та малих літер в імені сторінки має значення.

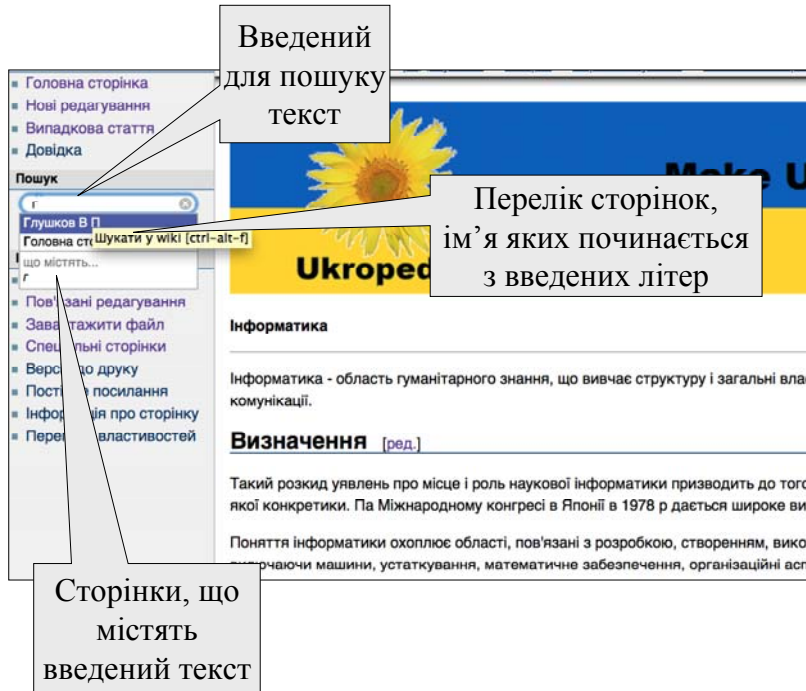


Рис.4.4. Пошук сторінок в Semantic MediaWiki.

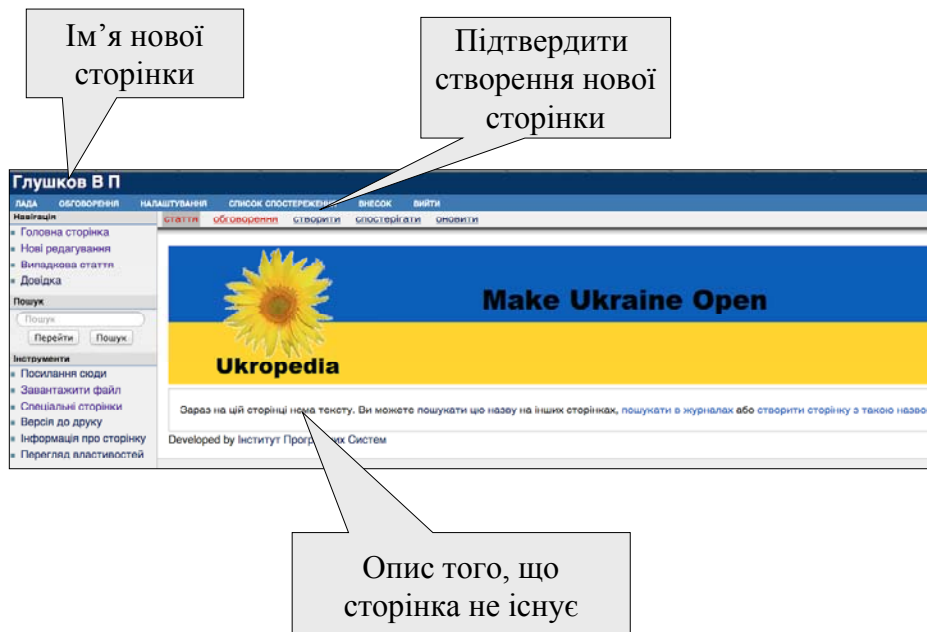


Рис.4.5. Створення нової Wiki-сторінки

Щоб створити нову сторінку, потрібно у вікні редагування ввести її контент – текст та інші елементи (рисунок, заголовки, таблиці та посилання на інші об'єкти тощо), які відповідають обраній темі.

Потім цей контент можна змінювати, доповнювати або повертатися до попередніх версій.



Рис.4.6. Збереження нової Wiki-сторінки

Після того, як інформація внесена, треба зберегти ці зміни – докладніше це описано в розділі «Редагування сторінки». Після цього нова сторінка буде створена і стане доступною для інших користувачів системи, які теж зможуть проглядати та редагувати її (якщо це не заборонено автором сторінки).

Редагування сторінки

Для створення та редагування контенту сторінки потрібно ввести текст – з клавіатури або скопіювати з іншого джерела. Потім до цього тексту можна застосувати певне форматування, яке відповідає змісту введеної інформації. Приміром, можна частину тексту виділити курсивом або жирним текстом. Таке форматування буде відображатися при перегляді сторінки.

У коді сторінки жирний шрифт позначається символами “**”** на початку і у кінці, а курсив – звичайними лапками.

Перед тим, як зберігати внесені зміни, доцільно попередньо переглянути, як ці зміни відображаються при перегляді сторінки. Це дозволяє позбутися помилок та вчасно виправити невдалі редагування.

Якщо внесені зміни задовольняють вас, необхідно зберегти їх за допомогою кнопки “Зберегти сторінку”. В іншому випадку зміни не будуть збережені.

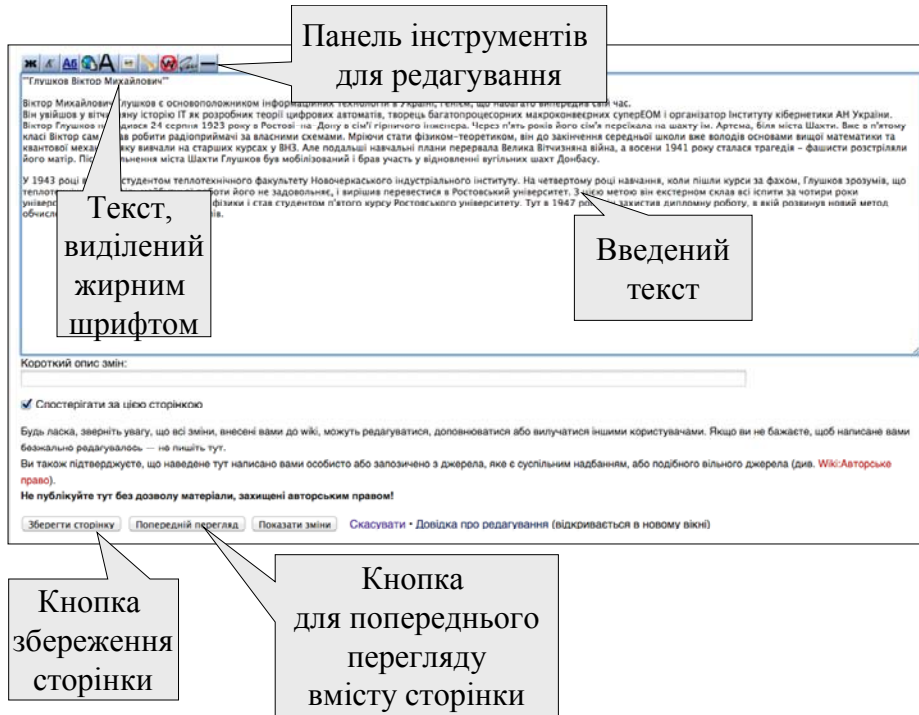


Рис.4.7. Редагування Wiki-сторінки



Рис.4.8. Попередній перегляд Wiki-сторінки перед збереженням змін

Обговорення сторінки

Для кожної звичайної сторінки Wiki-ресурсу існує відповідна сторінка її обговорення.

Пункт меню “Обговорення” дозволяє користувачам коментувати вміст сторінки, вносити свої зауваження та пропозиції навіть у тому разі, коли вони не мають повноважень для редагування сторінки. Крім того, наявність

можливості додавати часові мітки та підписи до коментарів дозволяє впорядковувати коментарі та відповіді на них, описувати внесені зміни тощо.

Щоб проглянути цю сторінку обговорення, потрібно, знаходячись на тій сторінці, коментарі до якої цікавлять користувача, обрати пункт меню “Обговорення”.

Якщо користувач хоче додати власні коментарі, то йому потрібно, знаходячись на сторінці обговорення, обрати пункт меню “Редагувати”.

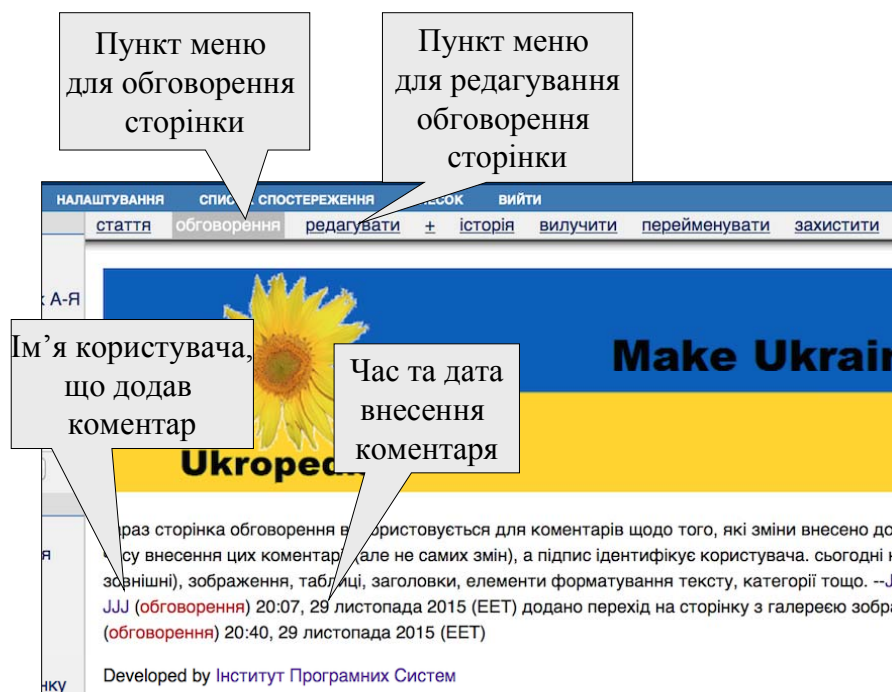


Рис.4.9. Обговорення Wiki-сторінки

Вилучення сторінки

Пункт меню “Вилучити” дозволяє видалити непотрібну сторінку – приміром, копію сторінки з іншою назвою, застарілу інформацію тощо. Виконати цю операцію можуть лише ті користувачі, кому адміністратор системи надав відповідні повноваження. Як правило, той користувач, що створив сторінку, має право її видалити. Це дозволяє позбутися неправильно створених сторінок.

За допомогою пункту меню “Перейменувати” можна змінити ім'я поточної сторінки – також за наявності відповідних повноважень в користувача

Пункт меню “Захистити” дозволяє заборонити іншим користувачам редагувати поточну сторінку.

Історія сторінки

Пункт меню “Історія сторінки” дозволяє проглянути історію того, хто й коли вносив зміни в дану сторінку.

Створення та менеджмент Wiki-ресурсів у MediaWiki

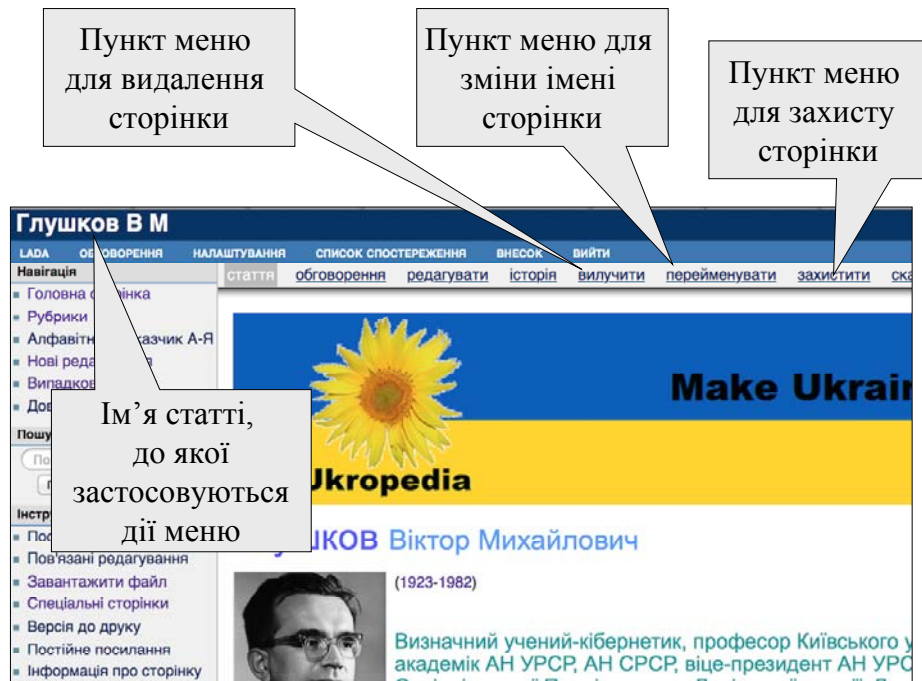


Рис.4.10. Вилучення Wiki-сторінки

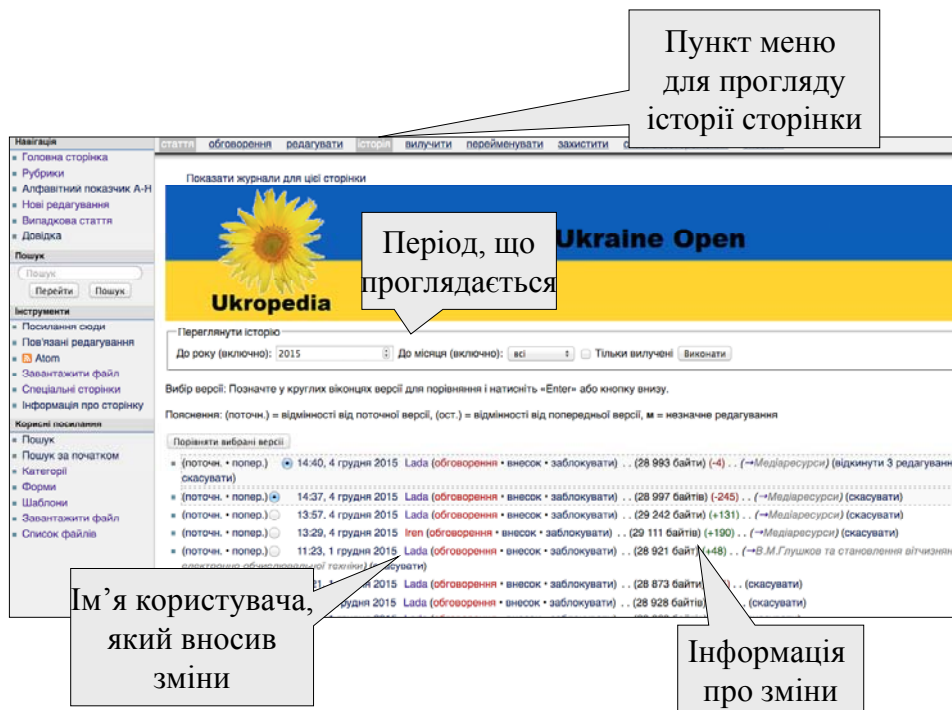


Рис.4.11. Історія Wiki-сторінки

Скасування редагування

Дана система дозволяє досить легко позбутися від помилково внесених редагувань. Для цього користувачеві достатньо повернутися до попередньої версії сторінки. Версію, до якої потрібно повернутися, слід обрати на сторінці "Історія".

Крім того, можна порівняти будь-які дві обрані версії сторінки.

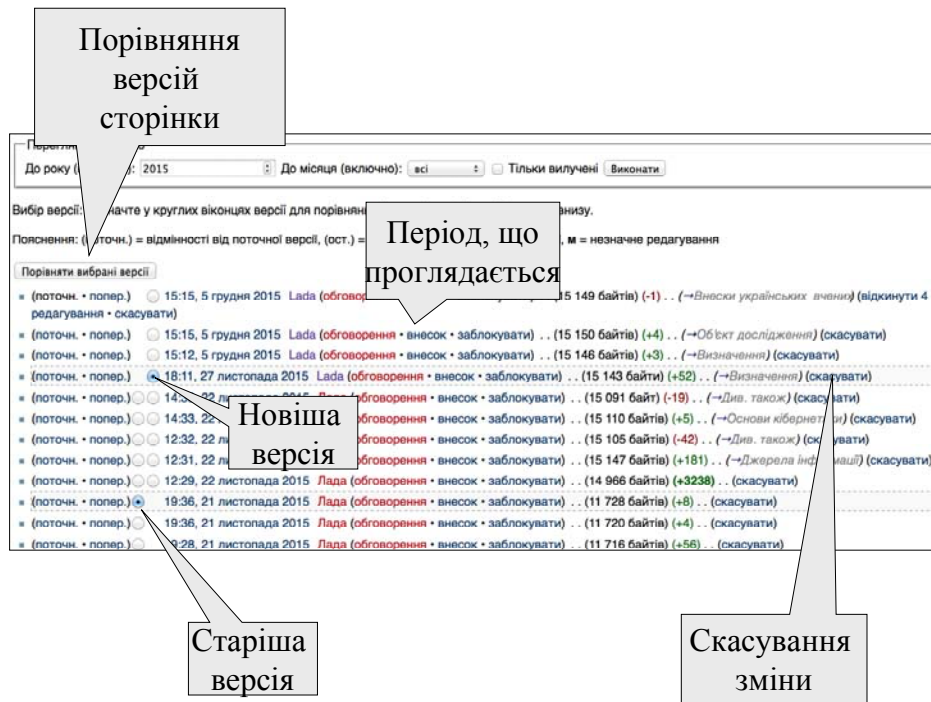


Рис.4.12. Порівняння версій Wiki-сторінки

Обравши версії сторінки, які потрібно порівняти, та натиснувши кнопку “Порівняти вибрані версії”, користувач отримує наступну інформацію:

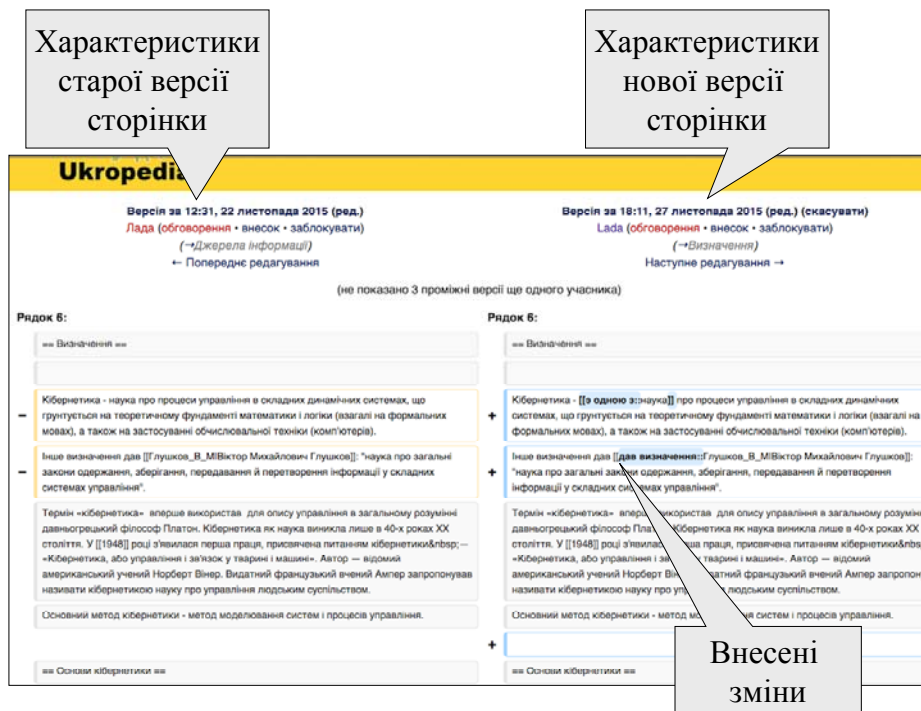


Рис.4.13. Вікно розбіжностей між версіями Wiki-сторінки

4.2. Розширені можливості системи

Крім базових можливостей, необхідних для створення Wiki-сторінок, у MediaWiki надаються додаткові функції, які дозволяють користувачам більш ефективно працювати з контентом.

Створення заголовків

Для кращого розуміння інформації до тексту сторінки доцільно додати заголовки різного рівня. Найпростіше зробити це за допомогою кнопки меню редагування.



Рис.4.14. Створення підзаголовків Wiki-сторінки

У коді сторінки заголовки 2 рівня, що на рівень нижче заголовку сторінки позначаються символами “==” на початку і у кінці. Щоб ввести заголовок 3 рівня, потрібно використовувати символи “===”, 4 рівня символи “====” і так далі.

Поділ документа на секції відбувається при додаванні в нього заголовків. Наприклад:

== Перший розділ ==

Текст.

=== Підрозділ ===

Ще текст.

==== Під-підрозділ ====

Ще якийсь текст.

== Другий розділ ==

Таблиця змісту автоматично додається до кожної сторінки, що має більше трьох заголовків, крім тих випадків, коли користувач відключив показ таблиці змісту у своїх налаштуваннях або коли у коді сторінки міститься спеціальний тег `__NOTOC__` (або ж `__БЕЗЗМІСТУ__`, при використанні

LanguageRu.php). Наявність такої таблиці значно спрощує навігацію всередині сторінки та дозволяє користувачам швидко отримати доступ до потрібних відомостей.

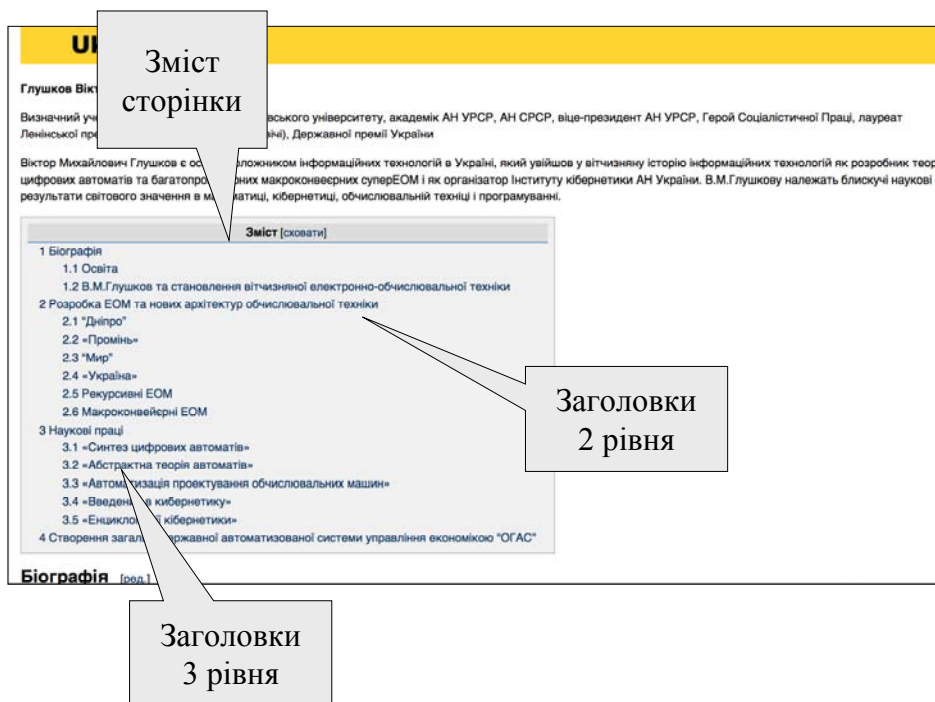


Рис.4.15. Відображення підзаголовків різних рівнів на Wiki-сторінці

Зміст відображається перед першим із заголовків, або ж у тій місці тексту, де розміщений тег `__TOC__`. Звичайно перед змістом міститься деякий додатковий текст із коротким оглядом статті. Хоча варіант за замовчуванням є найбільш зручним, іноді має сенс пересунути таблицю змісту, наприклад, зафіксувати її на самому початку статті, або ж додати її копію наприкінці.

Таблицю змісту зручно використовувати для навігації усередині сторінки.

Посилання на інформаційні ресурси

Текст сторінки може містити посилання як на зовнішні інформаційні ресурси (Web-сайти, сторінки інших Вікіпедій тощо), так і на інші сторінки даної Wiki-ресурсу. Натиснувши на таке посилання, можна перейти до перегляду обраного ресурсу.

В MediaWiki можливі п'ять типів посилань:

- Внутрішні посилання на інші сторінки Wiki;
- Зовнішні посилання на інші Web-сайти;
- Зовнішні посилання на внутрішні сторінки Wiki;
- ІнтерWiki-посилання на інші Web-сайти, попередньо зареєстровані у Wiki;
- Міжмовні посилання на інші Web-сайти, попередньо зареєстровані як іншомовні версії даного Wiki-проекту.

Посилання на інші сторінки Wiki-ресурсу

Для того, щоб послатися на іншу сторінку Wiki-ресурсу, потрібно додати у відповідному місці ім'я цієї сторінки у подвійних прямокутних дужках. Найпростіше зробити це за допомогою відповідної кнопки панелі редагування коду сторінки. Саме посилання буде виділятися у тексті іншим кольором. Після збереження сторінки ви побачите це посилання; воно буде червоним, якщо сторінка, на яку ви послались, не існує, і синім в іншому разі.

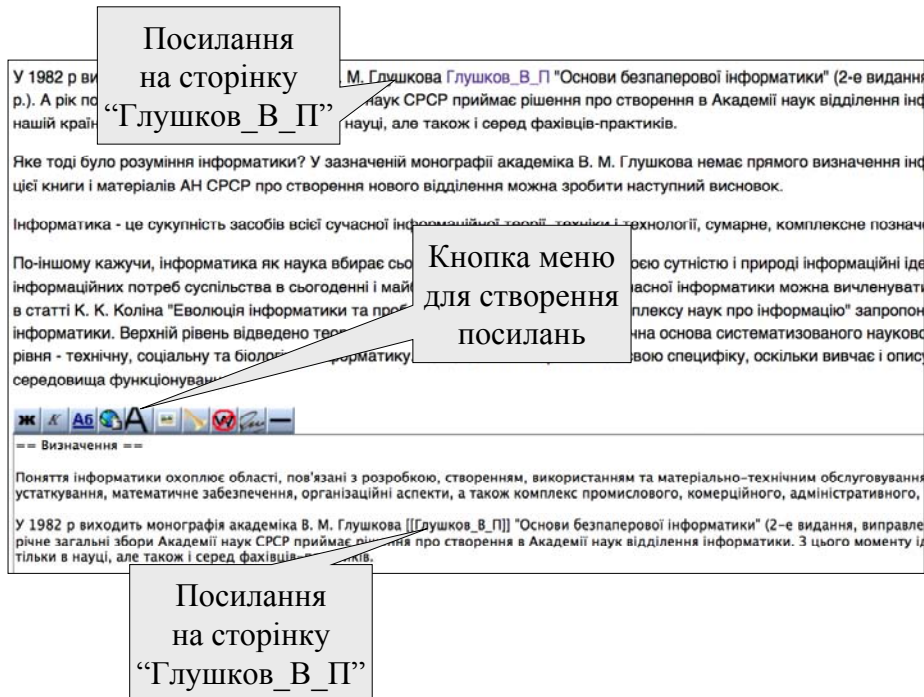


Рис.4.16. Посилання на інші Wiki-сторінки

Але у такому вигляді посилання у тексті не дуже зручно сприймати. Тому існує можливість співставляти посилання на сторінку з фрагментом тексту, який буде відповідати цьому посиланню, – приміром, це ж слово в іншому відмінку або його синонім. Для цього після посиланням усередині прямокутних дужок треба ввести цей фрагмент тексту, відокремивши його від імені сторінки вертикальною рисою.

Приміром, [[Глушков_В_П|Віктора Михайловича Глушкова]].

У такому варіанті посилання значно зручніше використовувати як складову тексту.

Крім того, деякі стандартні фрагменти тексту – приміром, дати, – теж доцільно оформляти як посилання. Це дозволяє групувати події за часом.

Посилання на зовнішні інформаційні ресурси

Для того щоб додати зовнішнє посилання, потрібно вставити в код сторінки його адресу – URL цього ресурсу, за яким через пробіл написано назву сторінки, на яку ви посилаетесь, в квадратних дужках. Коли ви збережете сторінку, ви побачите нове посилання з позначкою стрілочки після

нього. Ця позначка нагадує, що посилання може вести до іншого сайту, оскільки синтаксис квадратних дужок це дозволяє.

Приміром, можна вказати джерело інформації для створення сторінки, додавши адресу сайту та його назву: [http://ua.uacomputing.com/persons/glushkov/ ІТ в Україні. Історія та особистості].

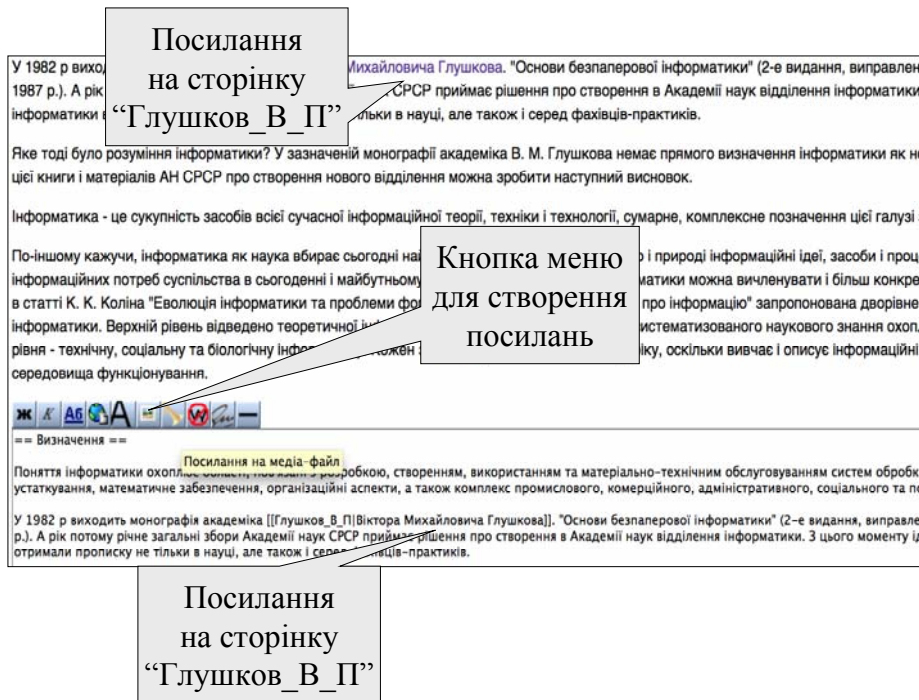


Рис.4.17. Оформлення посилань на інші Wiki-сторінки

Посилання на інші Вікіпедії

Можна досить легко створювати посилання на сторінки інших Вікіпедій. Це дозволяє отримувати доступ до додаткової довідкової інформації, не включаючи її до власної сторінки. Основна перевага такого рішення полягає в тому, що такі відомості будуть оновлюватися за рахунок зусиль користувачів цієї іншої Вікіпедії.

Для цього потрібно включити до коду сторінки наступний текст: [[wikipedia: ім'я Вікі-сторінки]]. Для створення такого рядку потрібно попередньо знайти в Вікіпедії потрібну статтю та довідатися її ім'я. Приміром, можна додати наступний рядок:

[[wikipedia: Victor_Glushkov]]

На жаль, такі посилання коректно працюють лише для англійської версії Вікіпедії <https://en.wikipedia.org>, а для посилань на сторінки української Вікіпедії доцільніше використовувати звичайні зовнішні посилання.

Крім тексту, можна пов'язати посилання – як на зовнішній ресурс, так і на сторінку Wiki-ресурсу – із зображенням.

Приміром, даний рядок

[[Файл:Инкиб.jpg|100px|50px|link=Інститут_кібернетики]]

[[Файл:Инкиб.jpg|100px|border|center|baseline|link=Галерея Глушкова|галерея зображень Інституту кібернетики]] щодо

дозволяє пов'язати зображення з файлу IC.JPG із сторінкою Wiki-ресурсу "Інститут_кібернетики".



Рис. 4.18. Посилання на зовнішнє джерело

Приклад зовнішнього посилання: цей рядок дозволяє пов'язати зображення БІ.JPG з сайтом, на якому представлено текст книги "Основи безбумажної інформатики"

[[Файл:БІ.JPG|400px|50px|link=https://www.google.com.ua/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjOhKeE6p_JAhWJ0RQKHZVmCzwQFggfMAA&url=http%3A%2F%2Fwww.pseudology.org%2Fscience%2FGlushkov_Osnovy_bezbumazhnoi_informatiki.pdf&usg=AFQjCNGXDyTh8K2uI-vQL5EmIS9vbliegw&sig2=HgbUBwRZ1bYeWAZMETmUbw]]

Категоризація сторінок

Зручним інструментом класифікації знань, представлених у Wiki-ресурсі, є категоризація сторінок.

У Wiki може використовуватися довільна кількість класифікаторів.

Основне призначення класифікаторів у системі:

- об'єднати групу статей, поєднаних спільним класифікатором;
- здійснювати пошук статей за класифікаторами.

Кожна сторінка Wiki-ресурсу може бути віднесена до однієї або до кількох категорій будь-якого з класифікаторів (залежно від специфіки самого класифікатора). Енциклопедія припускає співробітництво і співіснування різних точок зору без конкурентного витиснення там, де це не потрібно, тому система категорій у будь-якій області не зобов'язана являти собою

приналежне лише однієї класифікації дерево, а може бути системою деревоподібних структур, що накладаються одна на одну. Тобто та сама категорія чи стаття може бути віднесена до категорій з різних класифікацій, і при категоризації статті в рамках однієї класифікаційної системи практично не важливо, чи класифікована вона чи ні в рамках інших систем. Приміром, Wiki-сторінка, що описує певну організацію, може бути класифікована за типом діяльності установи (“належить до НАНУ”), за її місцезнаходженням (“організації міста Києва”), за кількістю співробітників (“середня організація”).

Можуть використовуватися як стандартні класифікатори, такі як УДК, так і введені самими користувачами.

Технічно сторінки категорій створюються і редагуються так само, як і звичайні сторінки. Контент сторінки категорій містить відсортований за алфавітом список статей Wiki, що відносяться до цієї категорії. На сторінках категорій сортування виробляється автоматично, а на сторінках статей категорії розташовуються в порядку проходження відповідних тегів. Сторінка з будь-якого простору імен включається в будь-яку категорію шляхом додавання тега-посилання. Щоб віднести сторінку до певної категорії, потрібно додати в її код [[Категорія:Назва категорії]].

Приміром, якщо додати до коду сторінки [[Категорія:Персоналії]] [[Категорія:Науковці]] [[Категорія:Українські дослідники]] [[Категорія:Особи, що працювали в Києві]], то це відобразиться наступним чином:



Рис.4.19. Вдображення категорій на Wiki-сторінці

Категорії, до яких належить сторінка, зазвичай перелічені внизу цієї сторінки.

Примітка. Починаючи з Mediawiki 1.13, категорію можна сховати від перегляду додаванням коду "`__HIDDENCAT__`".

Сторінки категорій дозволяють встановлювати ієрархію категорій: для цього слід вказати, що певна категорія є підкатегорією іншої категорії. Приміром, категорія “Науковці” є підкатегорією категорії “Персоналії”.

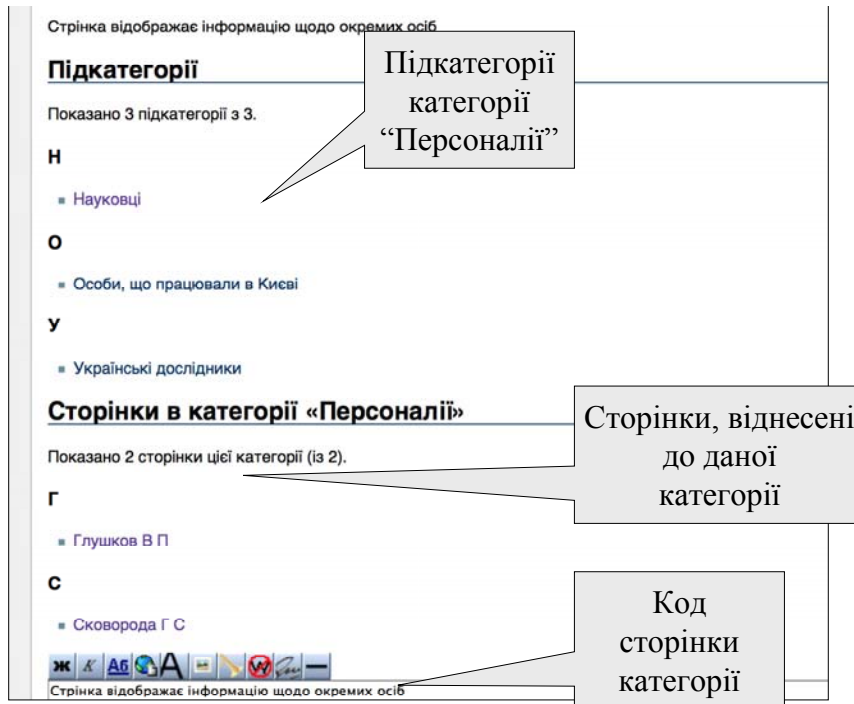


Рис.4.20. Сторінка категорії

Робота з шаблонами

Створення шаблону

Шаблони – це спеціальні Wiki-сторінки, вміст яких призначено для включення (вбудовування) в інші сторінки. Назві шаблону передуює слово "Template:", яке відносить шаблон до простору імен "Template". Знаючи це, можна створити шаблон так само, як і звичайну сторінку.

Основне призначення шаблонів – спростити та прискорити створення нових Wiki-сторінок та забезпечити однотипне подання інформації для користувачів.

Створювати новий шаблон можна як звичайну сторінку або створити із формою. Цей пункт «створити із формою» з'являється в меню при створенні шаблону з новим ім'ям.

Шаблон – це Wiki-сторінка, що є заготовленим оформленням тексту (тексти, рамки, зображення, змінювані тексти) для вставки в створювану Wiki-сторінку. Шаблони є потужним засобом, що допомагає поліпшити організацію Вікіпедії.

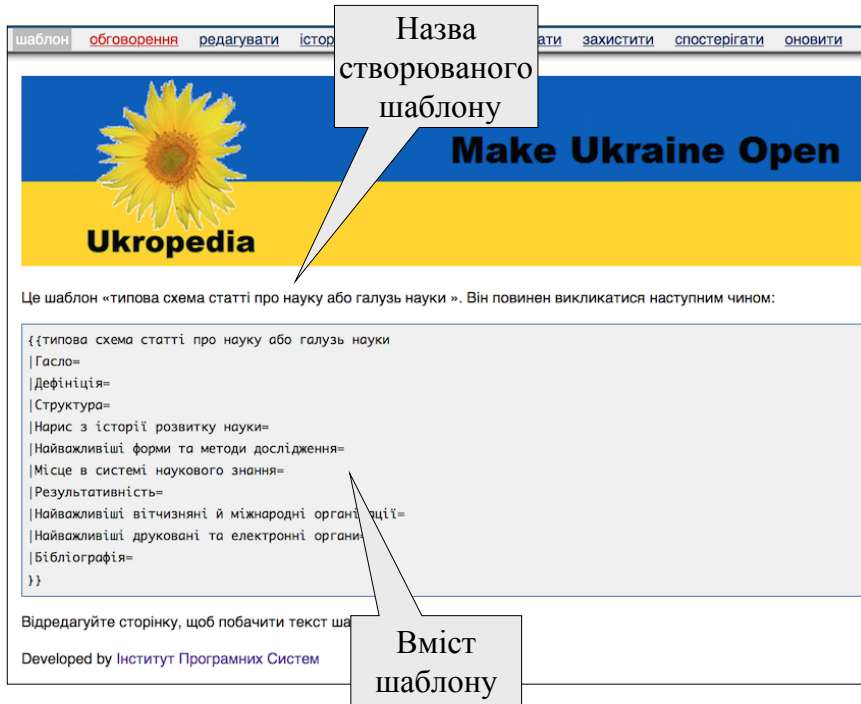


Рис.4.21. Сторінка шаблону

Це дозволяє позбутися проблем з різними формами представлення семантично однакових відомостей. Приміром, у Вікі, що присвячена історичним особам, частина людей буде описана властивістю “час народження”, інші учасники опишуть людей властивістю “рік народження”, а треті – “дата народження”. Для того, щоб одержати список історичних особистостей і вивести їх дати народження, потрібно написати три запити замість одного:

```

{{#ask: [[Категорія:Історична особистість]] | ?Час народження }}
{{#ask: [[Категорія:Історична особистість]] | ?Рік народження }}
{{#ask: [[Категорія:Історична особистість]] | ?Дата народження }}

```

Для рішення цих проблем застосовують шаблони і форми.

Шаблони у Вікіпедії часто застосовуються для того, щоб використовувати у Вікі-статтях складну розмітку, ховаючи при цьому деталі реалізації. Відомим прикладом шаблонів є інфобокси.

```

{{Назва шаблону
| параметр1=значення параметра 1
| параметр2=значення параметра 2
| параметр3=значення параметра 3
}}

```

Шаблони дозволяють однотипно оформляти та виводити в єдиному порядку подібні відомості на різних сторінках. Приміром, шаблон

«Науковець» дозволяє виділити основні відомості щодо персоналій, пов'язаних з науковими дослідженнями.

```

{{Науковець
|прізвище=
|ім`я=
|по батькові=
|сфера діяльності=
|наукові звання=
|рік народження=
|місце роботи=
|державні нагороди=
}}
```

Якщо редагувати сторінку цього шаблону, можна побачити наступний код:

```
<noinclude>
```

Це шаблон «Науковець». Він повинен викликатися наступним чином:

```

<pre>
{{Науковець
|прізвище=
|ім`я=
|по батькові=
|сфера діяльності=
|наукові звання=
|рік народження=
|місце роботи=
|державні нагороди=
}}
</pre>
```

щоб побачити текст шаблону, потрібно відредагувати сторінку.

```

</noinclude><includeonly>{\ style="width: 30em; font-size: 90%; border:
1px solid #aaaaaa; background-color: #f9f9f9; color: black; margin-bottom:
0.5em; margin-left: 1em; padding: 0.2em; float: right; clear: right; text-
align:left;"
```

```

! style="text-align: center; background-color:#ccccff;" colspan="2"
|<big>{{PAGENAME}}</big>
```

```

|-
! прізвище
| [[ПІБ:{{{прізвище}}}}]]
|-
! ім`я
| [[ПІБ:{{{ім`я}}}}]]
|-
! по батькові
| {{{по батькові}}}
|-
```

```

! сфера діяльності
| {{{сфера діяльності}}}
|-
! наукові звання
| {{{наукові звання}}}
|-
! рік народження
| {{{рік народження}}}
|-
! місце роботи
| {{{місце роботи}}}
|-
! державні нагороди
| {{{державні нагороди}}}
|}
[[Категорія:Персоналії]]
</includeonly>
    
```

Вставивши цей шаблон до сторінки, потрібно ввести значення всіх параметрів. Значення можуть бути одиничні (приміром, рік народження) або множинні (приміром, нагороди), текстові або посилання на зображення.

В результаті цього на сторінку буде виводитися наступна інформація:

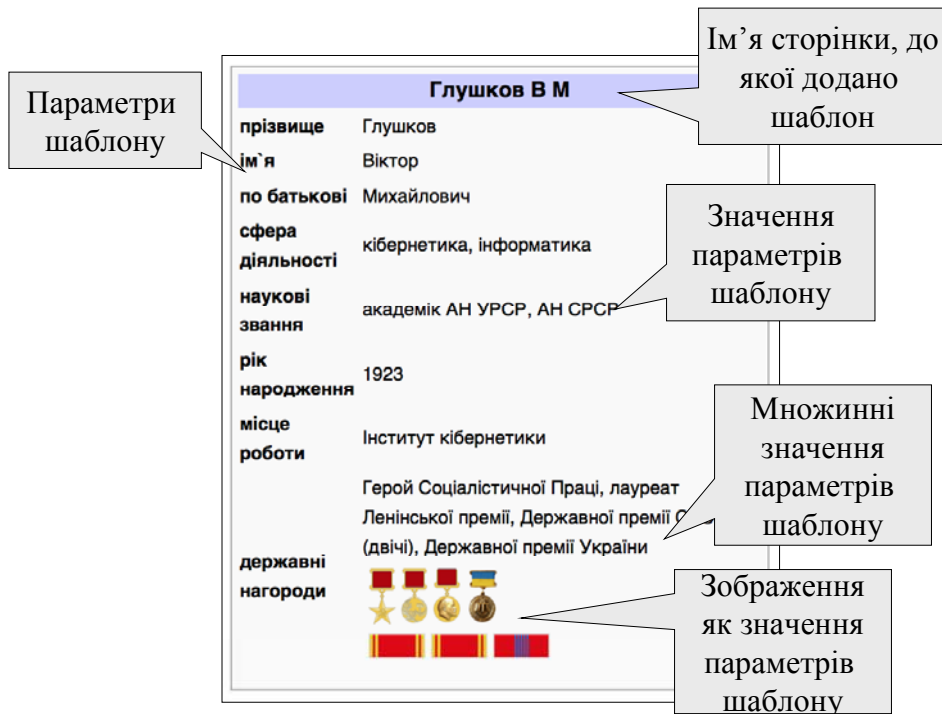


Рис.4.22. Відображення шаблону на Wiki-сторінці

Можна створювати значно складніші шаблони для введення інформації, відображення її властивостей та структури тощо.

В більшості випадків значна кількість шаблонів створюється заздалегідь та потім використовується користувачами. Перелік наявних шаблонів можна побачити за допомогою пункту меню "Шаблони".

У самому ж шаблоні можна визначити кольори, шрифти, відступи та інші речі, що стосуються оформлення.

Реалізація шаблону зберігається в просторі імен Template. Іменовані параметри шаблону беруть у потрібні фігурні дужки, а в момент виклику значення параметрів підставляються в потрібні місця. За допомогою шаблону можна одноманітно оформляти не тільки інфобокси.

Приклад застосування шаблону

Якщо створити сторінку "Template:Welcome", на якій записано:

"Привіт! Ласкаво просимо до Wiki"

то ця сторінка буде шаблоном. Тепер її включення за допомогою такого коду:

```
{{Welcome}}
```

до будь-якої іншої сторінки приведе до того, що при перегляді цієї сторінки на місці коду `{{Welcome}}` з'явиться текст "Привіт! Ласкаво просимо до Wiki ." Так вміст шаблону включається (інтегрується) до іншої сторінки.

Це працює лише при завантаженні сторінки, але не при її попередньому перегляді.

Якщо змінити код шаблону, то зміниться інформація на всіх сторінках, які містять цей шаблон.

Інші сторінки можуть використовувати сторінки шаблонів в такий спосіб:

`{{Назва}}` – таке посилання буде замінено поточним вмістом сторінки `[[Template:Назва]]`, який буде оновлюватися при кожному перегляді сторінки. Двигун MediaWiki залишає це посилання в Wiki -тексті сторінки незмінним. Якщо змінити вміст сторінки шаблону, то зміниться і інформація на всіх сторінках, де він використовується. Приміром, якщо замість розглянутого вище коду помістити перед святом "З Новим Роком!", то всі сторінки будуть оновлені цим надписом.

`{{subst:Назва}}` – коли використовується такий варіант, то виклик шаблону заміщується його вмістом один раз і не буде змінюватися при зміні вмісту шаблону до того, як сторінка з викликом шаблону не буде заново збережена після редагування. При цьому контент шаблону включається до самої сторінки, і його можна надалі редагувати як звичайний текст.

`{{safesubst:Назва}}` – ця конструкція поводить себе так саме, як і `subst.`, різниця виникає, лише коли вона зустрічається всередині іншого шаблону і цей шаблон або вставляють (за першою схемою), або прямо переглядають.

`{{msgnw:Назва}}` – включає шаблон у формі без Wiki-форматування (так само, як це робить `<nowiki>`) при перегляді сторінки.

Насправді будь-яка сторінка Wiki може бути використана як шаблон, але для цього слід явно вказати простір імен, в якому вона знаходиться:

- `{{Template:Назва}}` включає сторінку `[[Template:Назва]]`
- `{{Foo:Назва}}` включає сторінку `[[Foo:Назва]]`
- `{{:Назва}}` включає сторінку `[[Назва]]`
- `{{subst::Назва}}` заміщується вмістом `[[Назва]]`
- Якщо такого простору імен не існує, назву буде доповнено простором імен `Template:` тобто `{{Foo:Bar}}` включає сторінку `[[Template:Foo:Bar]]`

Параметри шаблону

Для збагачення механізму шаблонів в MediaWiki є можливість передавати параметри шаблону. Ці параметри дозволяють одному й тому самому шаблону генерувати різні результати.

Приміром, шаблон призначено для відповіді наступного вигляду “я прочитав (схвалюю) інформацію”. При цьому “я” та “інформація” є параметрами, і тому шаблоном можуть скористатися різні користувачі відносно різних інформаційних об’єктів.

Сторінка з цим шаблоном "Template:Схвалити", містить наступний текст:

```
{{{1}}} схвалює {{{2}}}
```

Нумерація – один із способів ідентифікувати параметри всередині шаблону. В шаблоні номер кожного параметру оточують три фігурних дужки: `{{{ }}}}`. Ця конвенція відрізняється від синтаксису використання назви шаблону.

Використовуючи цей шаблон на деякій сторінці, потрібно вказати значення параметрів, розділяючи їх символом вертикальної риски (`|`).

MediaWiki дозволяє передавати параметри шаблону трьома способами:

- анонімні,
- нумеровані,
- названі.

Анонімні параметри просто перелічують послідовно потрібні значення одне за одним у подвійних фігурних дужках після імені шаблону:

```
{{Схвалити|Я|дану роботу}}
```

Послідовність, в якій перелічено параметри, критична для правильного функціонування.

Звернення всередині шаблону до його аргументів за порядковим номером (`{{{1}}}` і так далі) дозволено лише для анонімних та нумерованих параметрів.

Для того щоб передати *нумеровані* параметри, слід просто позначити кожен з них номером:

```
{{Схвалити|2=дану роботу|1=Я}}
```

Цього разу код шаблону `{{Схвалити}}` отримає параметри `{{{1}}}` та `{{{2}}}` незалежно від порядку їх запису при використанні шаблону.

Третій спосіб передавати параметри шаблону – це надати їм *назви* (імена) замість номерів в самому описі шаблону. Приміром, це можна зробити так:

Хто `{{{1}}}` “схвалює” *що* `{{{2}}}`

Використання всередині шаблону назв `{{{хто}}}` та `{{{що}}}` замість номерів для позначення змісту кожного параметру спрощує його використання та робить шаблон більш зрозумілим. Крім того, що код шаблону стає більш зрозумілим, в ньому суттєво легше розібратися у випадку великої кількості параметрів.

Передавати ці параметри за назвою слід так:

`{{Схвалити|хто=Я|що=дану роботу}}`

Необхідність пам'ятати назви параметрів при використанні шаблону можна віднести як до переваг, так і до недоліків цього способу.

Не знайшовши вказаних параметрів, шаблон підставляє внутрішні назви цих параметрів замість відповідних значень.

Якщо при включенні шаблону, який має параметри, їх не вказати явно, приміром, записати:

`{{Схвалити}}`, то за замовчуванням значення будуть визначені наступним чином:

`{{{1}}}` “схвалює” `{{{2}}}`

Для того, щоб подібне використання шаблону зробити ефективним, слід задати значення параметрів за замовчуванням, тобто ті, які буде виведено у випадку, коли відповідні значення параметрів шаблону не передано. Так, якщо змінити сторінку шаблону таким чином:

`{{{хто|автор сторінки}}}` “схвалює” `{{{що|сторінку, про яку йде мова}}}`,

то виклик цього шаблону без параметрів викличе появу на сторінці, до якої включено цей шаблон, “автор сторінки схвалює сторінку, про яку йде мова”

Порядок оцінки шаблону: параметри шаблону повністю оцінюються перед тим, як вони входять до шаблону. Тому, якщо один з параметрів шаблону є викликом іншого шаблону, то цей шаблон буде викликаний всередині.

Керування включенням тексту шаблону до сторінок

Зазвичай текст шаблону повністю відображується як при прямому перегляді, так і на сторінці, яка включає шаблон. Але є можливість обирати, які частини сторінки шаблону будуть відображені чи включені, за допомогою тегів `<noinclude>` та `<includeonly>`.

Будь-який текст між `<noinclude>` та `</noinclude>` буде відображено при прямому перегляді сторінки шаблону, але не буде включено до будь-якої іншої сторінки. Це корисно, якщо необхідно включити до шаблону частини

коду, яка не буде розповсюджуватися на інші сторінки, які включають цей шаблон, такі як:

- Посилання категорій, до яких віднесений сам шаблон
- Міжмовні посилання на той самий шаблон іншими мовами
- Пояснювальний текст про використання шаблону

Аналогічно, будь-який текст між `<includeonly>` та `</includeonly>` буде відображено лише при включенні шаблону на тій сторінці, яка його включає. Але його не буде відображено при прямому перегляді сторінки шаблону.

Це корисно для категоризації сторінок, які включають шаблон.

Примітка: якщо в такий спосіб змінити категорії, до яких віднесено шаблон, на відображення цієї зміни на сторінках, що включають цей шаблон, необхідний певний час: це зумовлено чергою задач. Щоб примусити двигун MediaWiki терміново переглянути категорії певної сторінки, потрібно відкрити її для редагування та зберегти без змін.

Вся інформація шаблону поза тегам `<noinclude>` та `<includeonly>` обробляється та відображається звичайно.

Організація шаблонів

Для того, щоб ефективно користувалися шаблонами у Wiki-проекті, слід надати можливість знайти як самі шаблони, так і інформацію про їх застосування.

Для пошуку шаблонів використовують пункт меню “Шаблони” (або Спеціальні сторінки > Усі сторінки в Простір назв: list, Template and Виконати).

Для надання інформації про використання шаблону доцільно включити на його сторінці відповідні пояснення – між тегам `<noinclude>` та `</noinclude>` розмістити приклад виклику шаблону, який користувач може скопіювати і вставити на свою сторінку, а потім додати актуальні параметри.

Якщо потрібно використати посилання на шаблон, то слід використати шаблон `{{tl}}`, а в якості його параметру написати ім'я того шаблону, на який потрібно послатися. Приміром, `{{tl|Схвалити}}` є посиланням на шаблон “Схвалити”.

Копіювання шаблонів з одного Wiki-проекту до іншого

Шаблони зазвичай спираються на CSS чи на інші шаблони, тому користувачі часто стикаються із проблемами при переносі шаблону з одного проекту до іншого. Кроки, перелічені нижче, мають допомогти у більшості ситуацій.

Якщо є права на імпортування (конкретно, Import pages from a file upload) на цільовому проекті, то потрібно обрати Special:Export оригінальної Wiki-системи та завантажити файл .xml з історією усіх потрібних шаблонів.

- Ввести повну назву потрібного шаблону у текстове поле, приміром "Template:Welcome", звертаючи увагу на регістр літер та спеціальні символи – якщо назва шаблону не є абсолютно точною, експорт

відбудеться, але результируючий .xml-файл не міститиме очікуваних даних.

- включити "Включити шаблони".
- Відключити "Включати тільки поточну версію, без повної історії".
- обрати "Експорт".
- На цільовому Wiki-проекті перейти до сторінки Special:Import та завантажити отриманий .xml.

Якщо немає прав імпорту на цільовому Wiki -проекті:

- Перейти до сторінки необхідного шаблону, потім до сторінки його редагування, звідки скопіювати весь Wiki текст
- На цільовому проекті перейти до створення/редагування сторінки з такою самою назвою, як сторінка шаблону, текст якого скопійовано, та вставити скопійований Wiki-текст. В короткий опис змін нового шаблону записати посилання на його оригінал.
- На сторінці редагування шаблону оригінального Wiki -проекту, нижче вікна редагування, знайти перелік "Templates used on this page:". Для кожного шаблону з цього переліку теж ітеративно виконати інструкції, описані вище.

В такий спосіб буде скопійовано весь необхідний код, і цього виявиться досить для багатьох шаблонів. Якщо це не спрацює, на цільовому Wiki-проекті потрібно перевірити червоні посилання під "Pages transcluded onto the current version of this page:", що знаходиться під вікном редагування. Якщо такі посилання там є, слід виконати описані вище кроки і для цих сторінок.

Після успішного переносу з оригінального Wiki-проекту шаблону та всіх сторінок, що з ним пов'язані, можна відредагувати його на цільовому Wiki-проекті, налаштувавши його відповідно до власних потреб.

В шаблонах часто використовується розширення ParserFunctions. На сторінці ParserFunctions слід перевірити, чи використовує скопійований шаблон будь-які з цих функцій. Якщо так, слід встановити це розширення (для цього потрібно мати доступ системного адміністратора до сервера, на якому розміщено Wiki –проект).

Крім коду MediaWiki, багато які шаблони використовують CSS, а деяким для повноцінного функціонування потрібен JavaScript. Це може бути причиною неочікуваної поведінки скопійованого шаблону. Для того, щоб скопіювати необхідний код CSS або JavaScript, необхідний адміністративний доступ до цільового Wiki -проекту, оскільки доведеться редагувати системні повідомлення з простору імен "`{{cs:MediaWiki}}`".

Зображення

Використання зображень на сторінках Wiki-ресурсу робить подання інформації більш наочним та повним.

Для ілюстрування тексту можна використовувати графічні файли наступних форматів, що підтримуються за замовчуванням:

Створення та менеджмент Wiki-ресурсів у MediaWiki

використання завантажених зображень на інших сторінках та в іншому контексті.

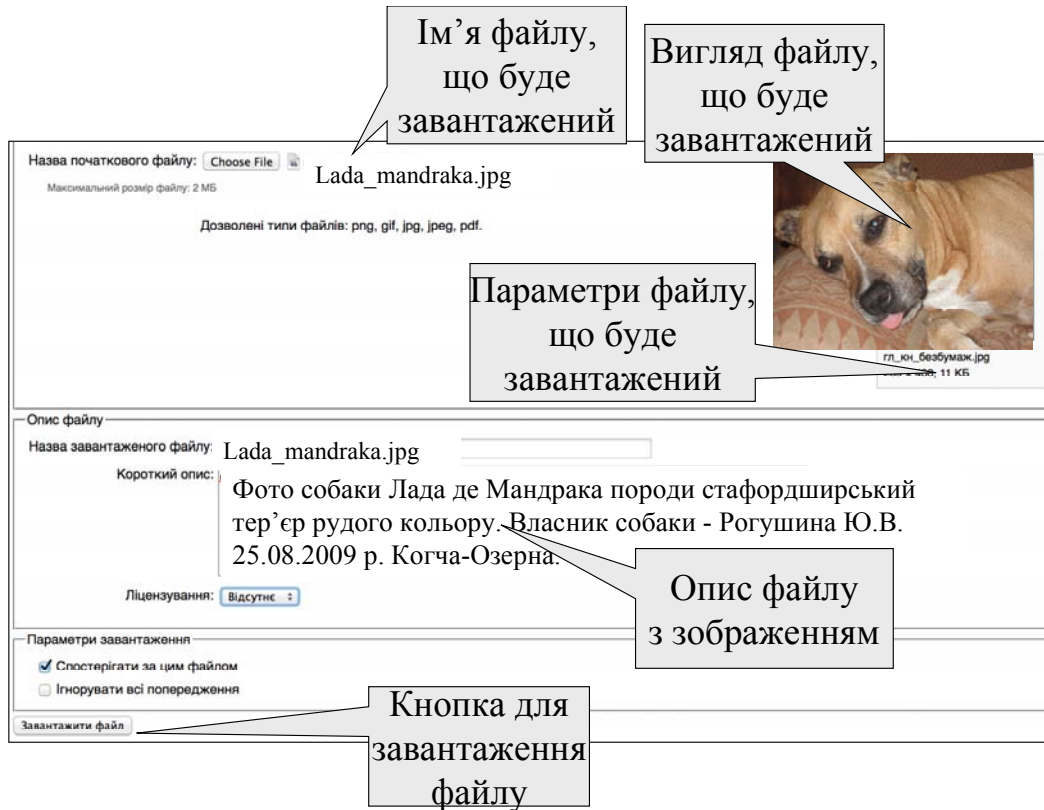


Рис.4.24. Опис файлу

Після того, як введено опис файлу та проглянуто його вигляд, треба натиснути кнопку “Завантажити файл”.

В даному випадку завантажено файл у форматі jpg, який містить фотографію собаки на ім'я Лада де Мандрака породи стафордширський тер'єр. Крім того, опис файлу містить характеристику кольору тварини (рудого кольору), ім'я її власника, дату створення зображення, назву місцевості, де було зроблено фото.

Надалі таке зображення може використовуватися, приміром, у сторінках щодо цієї місцевості або у сторінках за цю дату, або ж за змістом – в описах собак та їх підтипів (приміром, тер'єрів) та надтипів (приміром, хатніх тварин).

Коли завантаження виконане, відкривається сторінка даного зображення (її можна редагувати, як звичайну сторінку), що містить відомості щодо завантаженого файлу. Така сторінка мстить зображення, його опис, використання цього файлу в Wiki-ресурсу та метадані, пов'язані з цим файлом (якщо такі дані існують).

В процесі побудови Wiki-сторінок користувачі можуть знаходити цей опис та використовувати його для коректного застосування відповідних зображень.



Рис.4.25. Сторінка зображення

Розміщення зображення в тексті

Найпростіший варіант вставки зображення до тексту – через кнопку меню редагування.



Рис.4.26. Вставка зображення до Wiki-сторінки

Щоб знайти вже завантажені файли, потрібно зайти до сторінки "список завантажених файлів", яка знаходиться за адресою http://wiki.isoftware.kiev.ua/index.php/Спеціальна:Список_файлів. Доступ до цієї

Створення та менеджмент Wiki-ресурсів у MediaWiki

сторінки можна отримати наступним чином: в меню обрати пункту “Спеціальні сторінки”, зайти до переліку спеціальних сторінок і групі “Файли” обрати пункт “Список файлів”.

Якщо перейти на цю сторінку, то можна побачити інформацію щодо завантажених файлів, знайти потрібне зображення та скопіювати його коректне ім'я.

The screenshot shows the 'List of files' page in MediaWiki. At the top, there are two search filters: 'Пошук за назвою' (Search by name) and 'Пошук за ім'ям користувача' (Search by user name). Below these are input fields for search criteria. The main content is a table with columns: 'Дата' (Date), 'Назва' (Name), 'Мініатюра' (Thumbnail), 'Розмір (в байтах)' (Size in bytes), 'Користувач' (User), 'Опис' (Description), and 'Версії' (Versions). The table lists four files uploaded on 21 листопада 2015:

Дата	Назва	Мініатюра	Розмір (в байтах)	Користувач	Опис	Версії
14:18, 21 листопада 2015	Інститут.jpg (файл)		184 КБ	Лада	Інститут кіберетики ім.В.М.Глушкова, м.Київ	1
14:15, 21 листопада 2015	Глушков1.jpg (файл)		42 КБ	Лада	фото В.М.Глушкова	1
14:14, 21 листопада 2015	Глушків синтез.jpg (файл)		115 КБ	Лада	монографія Глушкова	1
14:13, 21 листопада 2015	Глушків беззубчик.jpg (файл)		11 КБ	Лада	обкладинка монографії Глушкова	1

Рис.4.27. прогляд завантажених зображень

Це ім'я потрібно вставити до коду сторінки, відмітити і натиснути кнопку меню редагування “Файл”. Приміром, якщо обрано ім'я файлу Глушков1.jpg, то буде згенерований рядок. [[Файл:Глушков1.jpg]]. Цей код буде виводити на сторінці зображення в оригінальному розмірі.

Розмір зображення. У більшості випадків доцільно змінити розмір зображення та вказати його місце на сторінці. Для цього використовуються такі параметри форматування.

Наприклад, [[Файл:Lilia.JPG|100px]] обмежує виведення зображення Lilia.JPG в 100 пікселів. Змінюючи цей параметр, можна змінити розмір зображення, що виводиться.

Рамка зображення. Для того щоб яке-небудь зображення взяти в рамку, зручно використовувати атрибут «border». Його потрібно додати у довільне місце усередині подвійних квадратних дужок, відокремивши від інших параметрів вертикальною рисою. Таким чином, наприклад, легко вирішується проблема зображення прапорів з білими полями, що зливаються з тлом сторінки.

Приклад: [[Файл:Flag of France.png|border|прапор Франції|200px]]

Пояснювальний підпис. Пояснювальний підпис біля зображення дозволяє уточнити, що саме показане на ньому. Щоб такий підпис з'явився

біля зображення, потрібно усередині подвійних квадратних дужок після останнього параметра ввести його, відокремивши від інших параметрів вертикальною рисою

```
[[Файл:Vroom icon.png|мини|пояснювальний підпис]]
```

Він укладає зображення в рамку і зміщає зображення вправо. В області «Пояснювальний підпис» можна застосовувати html- і Wiki-розмітку (наприклад, відцентрувати підпис за допомогою тегів `<center></center>`).

Атрибути «міні», «мініатюра» (чи «thumb», «thumbnail»), вставлені між ім'ям файлу і пояснювальним підписом, зменшують ширину зображення (у залежності від налаштувань користувача – від 100px до 300px; висота змінюється пропорційно), притискають його вправо і поміщають у рамку:

Крім того, праворуч від пояснювального підпису з'являється спеціальний значок: клацнувши по ньому, можна перейти на сторінку зображення і подивитися його в натуральну величину. Текст розташовується ліворуч від значка.

Для одержання зображення заданої ширини (висота змінюється пропорційно) потрібно вказати цей параметр в пікселях (у форматі Числорх). При цьому не слід зловживати надмірно великими значеннями параметра. Цей атрибут можна використовувати як окремо, так і разом з атрибутом «мини» – тоді «мини» додає рамку і підпис, але не впливає на розмір:

```
[[Файл:AAA.png|100px|підпис]]
```

Крім того, можна задати для зображення не ширину, а висоту:

```
[Файл:AAA.png|мини|слева|x100px|підпис]]
```

Якщо задати обидва параметри, але співвідношення сторін у зображення інше, чим співвідношення заданих цифр, то зображення буде вписано в прямокутник відповідного розміру, тобто буде узятє те значення, при якому розмір зображення буде найменшим.

Крім розглянутого вище атрибута «мини», помістити зображення праворуч можна атрибутом «право» («right»). Приклад:

```
[[Файл:AAA.png|справа|100пкс]]
```

На відміну від зображення з атрибутом «мини», у даному випадку зображення не виводиться у рамку, і текст підпису не вказується на екрані постійно, а використовується як спливаюча підказка (після наведення курсору миші на зображення), а також показується замість картинки для тих читачів, що відключили відображення зображень у своєму браузері.

Галереї

Для вставки декількох тематично пов'язаних зображень у статтю застосовується спеціальний службовий тег `<gallery>`, що групує зображення в галерею.

У найпростішому випадку використовується наступний синтаксис. Наприклад, наступний код відображається показом відповідних фото, в яких для всіх зображень відведено однакові за розміром фрагменти екрану, а зображення відображаються в них відповідно до пропорцій фото.

```
<gallery heights="200px">>
```

Файл:Г1.jpg|Глушков В.М. робить доповідь
Файл:Ик1.jpg|Інститут кібернетики ім.В.М.Глушкова
Файл:Ик2.jpg|зображення Інституту кібернетики ім.В.М.Глушкова
Файл:Гг2.jpg|Глушков В.М. зі співробітниками
Файл:Гг4.jpg|Глушков В.М. – директор Інституту кібернетики
Файл:Глушков_днепр.jpg|Глушков В.М. випробує нове обладнання
Файл:Инкиб.jpg|Інститут кібернетики ім.В.М.Глушкова сьогодні
Файл:Глушков1.jpg|Глушков В.М. – нові дослідження

</gallery>

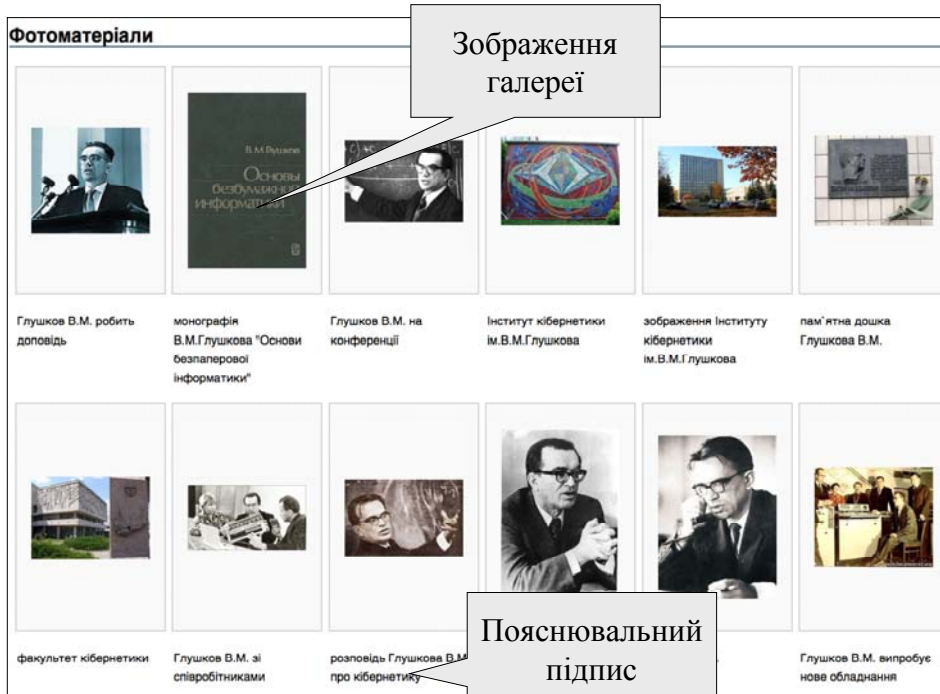


Рис.4.28. Галерея зображень у Wiki-сторінці

Більш сучасний вид галереї можна надати, якщо додати параметр `mode="packed"`. Приклад:

```
<gallery mode="packed" heights="160px">  
Файл:Гг2.jpg|Глушков В.М. зі співробітниками  
Файл:Гг3.jpg|виступ Глушкова В.М. на конференції  
Файл:Гг4.jpg|Віктор Михайлович Глушков  
Файл:Гг5.jpg| Глушков В.М.за роботою  
</gallery>
```

В такому випадку всі зображення матимуть однакову висоту, а їх ширина пропорційно залежить від розмірів похідного зображення.



Рис.4.29. Форматування галереї зображень

Є наступні параметри, що регулюють форматування галереї:

- perrow – кількість елементів у рядку (не працює в режимі packed)
- widths – розмір (кожної) картинки по горизонталі
- heights – розмір (кожної) картинки по вертикалі
- showfilename – з будь-яким значенням включає відображення імен файлів для кожного зображення.

Форматування зображень

Щоб додати до тексту сторінки зображення з елементами форматування, потрібно додати в код наступний рядок, де filename.extension – це ім'я файлу, яке треба обрати на розглянутій вище спецсторінці:

`[[File:filename.extension|options|caption]]`

або

`[[Файл:filename.extension|options|caption]]`

options – це параметри, які можуть бути відсутні чи приймати наступні значення, розділені вертикальною рисою (|):

- Format: border, frameless, frame, thumb (чи thumbnail) – Форматує зображення і визначає, як воно буде розташовано в частині сторінки, що залишилася.
- Resizing – одне з наступних:

{width}px – змінює розмір зображення відповідно до заданої максимальної ширини в пікселях, не обмежуючи його висоту;

x{height}px – змінює розмір зображення відповідно до заданої максимальної висоти в пікселях, не обмежуючи його ширину;

{width}x{height}px – змінює розмір зображення відповідно до заданої ширини і висоти в пікселях;

`upright` – корегує розміри зображення відповідно до переваг користувача (підходить для зображень, висота яких більше ширини).

Примітка. Зверніть увагу, що зображення завжди буде зберігати співвідношення сторін і може бути тільки зменшене (але не збільшено), якщо воно не масштабованого типу (растрові зображення не можуть бути збільшені).

За замовчуванням максимальний розмір залежить від значення *format* і внутрішніх розмірів зображення (у залежності від його типу).

- `Horizontal alignment` – одне з: `left`, `right`, `center`, `none` – Визначає горизонтальне вирівнювання (and inline/block or floating styles) зображення в межах тексту. Не має значення за замовчуванням.
- `Vertical alignment` – одне з: `baseline`, `sub`, `super`, `top`, `text-top`, `middle`, `bottom`, `text-bottom` – Визначає вертикальне вирівнювання неспливаючого зображення в блоці тексту. За замовчуванням вертикально вирівнюється по центрі (`middle`).
- `Link` – одне з:

`link={ціль}` – дозволяє перенаправляти (на заголовок довільної чи статті URL) посилання, зв'язану з зображенням; наприклад, `[[File:Example.jpg|20px|link=http://www.wikipedia.org]]` відображається як `Example.jpg` (зовнішнє посилання), чи `[[File:Example.jpg|20px|link=MediaWiki]]` відображається як `Example.jpg` (внутрішнє посилання).

`link=` (з порожнім значенням) – (MediaWiki 1.14+) показує зображення без зв'язаного посилання; наприклад, `[[File:Example.jpg|20px|link=]]` відображається як `Example.jpg`.

- Особливі параметри:

`alt={альтернативний текст}` – (MediaWiki 1.14+) визначає альтернативний текст (інтерпретується як HTML-атрибут `alt="..."` елемента ``), що буде показаний замість зображення у випадку невдачі при його чи завантаженні спробі вмонтувати в статтю, а також у разі потреби вказувати альтернативний опис (наприклад, при використанні читачем шрифту Брайля й інших спеціальних можливостей, встановлених у його браузері).

`page={номер}` – відображає сторінку з зазначеним номером (у даний час застосовується тільки при показі файлів `.djvu` чи `.pdf`).

`class={html class}` – (MediaWiki 1.20+) визначає класи (інтерпретується як HTML-атрибут `class="..."` елемента ``).

`lang={код мови}` – (MediaWiki 1.22+) для файлів SVG, що містять елементи `<switch>` з різними атрибутами `systemLanguage`, визначає мову, на якому файл буде відображатися. За замовчуванням – завжди англійський, навіть у неангломовних розділах Вікіпедії.

Якщо параметр не відповідає жодному з можливих, передбачається, що це підпис. Текст підпису відображається нижче зображення у форматах `thumb` і `frame`, чи як текст підказки в інших форматах. Текст підпису у

форматах thumb і frame може містити Wiki-посилання й інші елементи форматування. Розширення MediaWiki можуть мати додаткові параметри.

Таблиці

Дане середовище дозволяє включати в сторінки таблиці різного розміру.

Кожна таблиця в wiki-стилі складається з комірок. Найпростіша таблиця складається з однієї комірки.

Для таблиць використовується спеціальний код.

Таблиця починається з символів “{|”, після яких ідуть опції таблиці, та закінчується символами “|}”: вихідний код таблиці береться у фігурні дужки { } (перша фігурна дужка вихідного коду таблиці називається відкриваючою дужкою, остання – закриваючою); після відкриваючої дужки таблиці ставиться вертикальна риса |, така ж вертикальна риса також ставиться перед закриваючою дужкою таблиці. Відкриваюча дужка таблиці з вертикальною рисою називається відкриваючим елементом таблиці, а вертикальна риса з закриваючою дужкою – закриваючим елементом таблиці.

```
{|
|вміст таблиці
|}
```

Кожна таблиця може мати *заголовок*. Заголовок є зручним способом описати таблицю; він є необов'язковим, та включається в таблицю рядком “|+Заголовок”, що слідує за символами “{|” на початку таблиці.

Код для кожного нового рядка таблиці починається з символів “|-”, а окремі комірки одного рядку розділяються символом «|». Вмістом кожної комірки може бути текст, посилання на інші ресурси (внутрішні та зовнішні), посилання на файли (приміром, на зображення).

Рядок із заголовками колонок описується «!» замість «|», за винятком розділювача між параметрами клітинки та її значенням; різниця на екрані між заголовками та звичайним рядком залежить від браузера, часто заголовки колонок відображаються жирним шрифтом.

Перша комірка в рядку стає заголовком рядка, якщо рядок починається із символу «!» замість «|», та якщо інші дані в рядку починаються із наступного рядка.

Кожна комірка у таблиці починається з одинарної вертикальної риси |, після якої пишеться вміст комірку. Між вертикальною рисою і першою буквою тексту вмісту комірку допускається вставляти пробіл.

```
{| border="1"
|комірка А
|комірка Б
|комірка В
|}
```

Wiki-розмітка також дозволяє записувати комірки в один рядок. При цьому, всі комірки, що йдуть за першою коміркою у рядку, починаються з подвійної вертикальної риси: `||`. Наприклад:

```
{| border="1"  
|комірка А||комірка Б||комірка В  
|}
```

Рядок таблиці починається з вертикальної риси, після якої пишеться дефіс: `"|"`. Таблиця може мати довільну кількість рядків.

Кожна комірка у рядку може бути записана з нового рядка чи в один рядок.

```
{| border="1"  
|-  
|Комірка А-1  
|Комірка Б-1  
|Комірка В-1  
|-  
|Комірка А-2  
|Комірка Б-2  
|Комірка В-2  
|-  
|Комірка А-3  
|Комірка Б-3  
|Комірка В-3  
|}
```

Приклад коду таблиці:

```
{| class="wikitable" style="text-align:center"  
|-  
!початок  
!закінчення  
!місце навчання  
|-  
|1943  
|1946  
|теплотехнічний факультет Новочеркаського індустріального  
інститут  
|-  
|1946  
|1947  
|Ростовський університет
```

```

|-
|1949
|1949
|аспірантура Свердловського університету
|}
    
```

Цей код відображається на сторінці наступним чином:

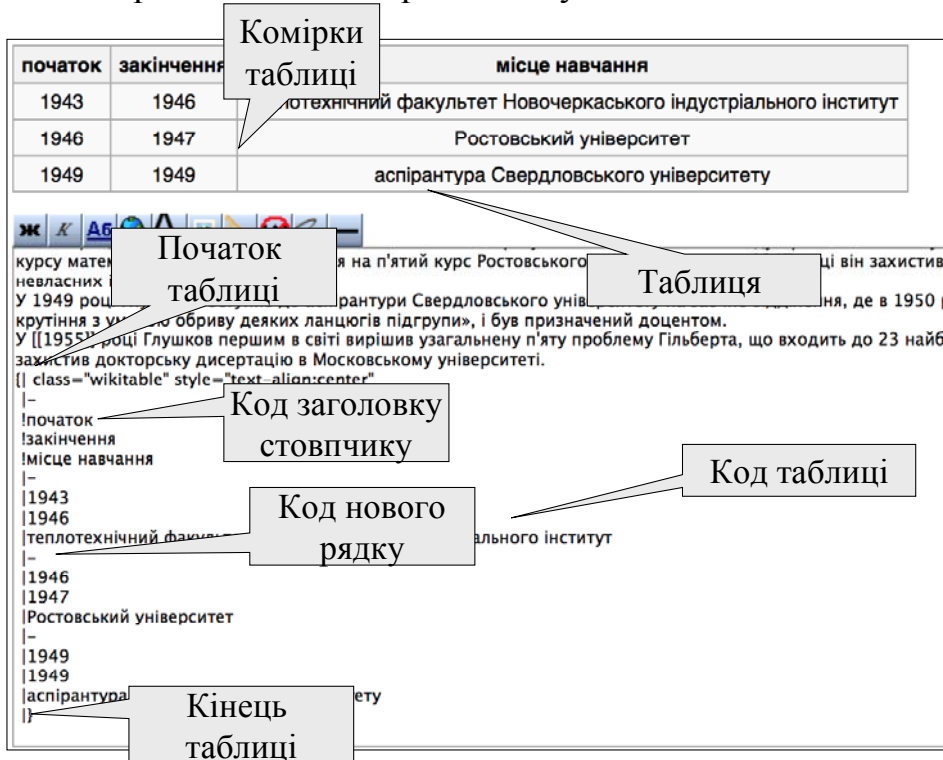


Рис. 4.30. Відображення таблиці на Wiki-сторінці

Можна використовувати різні варіанти оформлення таблиць.

Приміром, наступний код дозволяє вставляти в комірку зображення, що є посиланням на іншу сторінку, а для самої таблиці обрати більшу рамку.

```

{| border="3"
|-
| [[Файл:Инкиб.jpg|мини|центр|100пкс|link=Галерея щодо
Глушкова|Галерея ]]
|В даній галереї представлені різноманітні чорно-білі та кольорові
фотографії, пов'язані з діяльністю Віктора Михайловича Глушкова та
місцями, де він працював.
|}
    
```

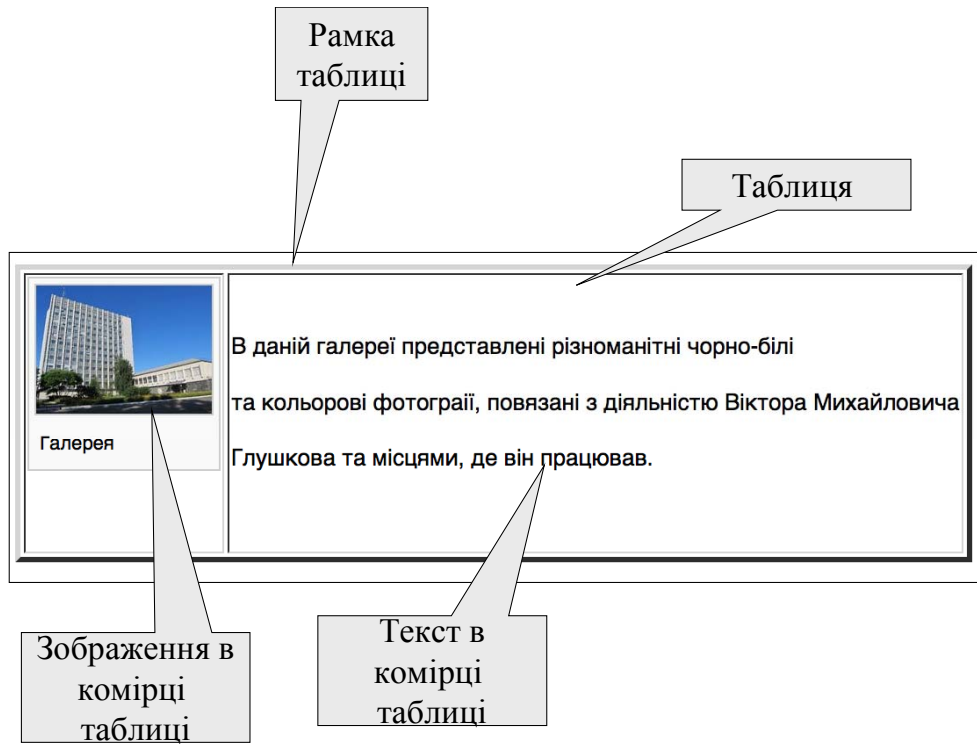


Рис.4.31. Відображення галереї на Wiki-сторінці

Використання таблиць у Wiki-сторінках робить подання інформації більш компактним та наочним.

5. Обробка семантики у *Semantic MediaWiki*

5.1. Цілі та задачі *Semantic MediaWiki*

Одним з поширених движків для створення і підтримки Wiki-ресурсів, що забезпечує семантизацію інформації, є *Semantic MediaWiki* (SMW). Це семантичне розширення *MediaWiki* – вільного програмного забезпечення з відкритим кодом для гіпертекстового Wiki-середовища, що є платформою для створення довідників, енциклопедій і каталогів. *Semantic MediaWiki* дозволяє користувачам додавати семантичні анотації до Wiki-сторінок, перетворюючи *MediaWiki* у семантичний ресурс.

Переваги використання SMW:

- Інтеграція інформації у різноманітних списках, що автоматично генеруються за результатами семантичних запитів;
- Візуалізація інформації (приміром, за допомогою додаткових розширень, такими як *Semantic Result Formats* і *Semantic Maps*), які дозволяють відображати інформацію в календарях, графіках і картах;
- Зручність введення структурованої інформації за допомогою семантичних шаблонів та форм;
- Персоніфікований пошук інформації – користувачі можуть шукати потрібну інформацію та створювати свої власні запити, використовуючи доповнення типу *Halo* і *Semantic Drilldown*
- Погодженість даних різними мовами та на різних Wiki-ресурсах;
- Можливість повторного використання інформації в інших застосунках – дані, створені в SMW, можуть легко передаватися назовні у форматах CSV, JSON і RDF, а з використанням *External Data* та *Semantic Result Formats* можна передавати дані з одного Wiki-ресурсу до іншого, усуваючи необхідність у дублюванні і ручній синхронізації.
- Інтеграція з технологіями *Semantic Web* – приміром, розширення *Triple Store Connector* дозволяє з'єднати Wiki-ресурс з RDF-сховищем і використовувати для запитів мову SPARQL .

Semantic MediaWiki отримала поширення на Wiki-фермах (так називають інструментальні засоби для створення Wiki-ресурсів), які потребують гнучкої і зручної обробки структурованих даних. Відомими впровадженнями є Wiki-співтовариства *Semantic Web* (semanticweb.org), офіційний портал інституту AIFB Технологічного інституту Карлсруе, Wiki-проект data.gov. *Semantic MediaWiki* може бути включена за запитом в популярній Wiki-фермі – Вікія, а у фермі *Referata* вона включена за замовчуванням. Відомий проект документування WWW як платформи webplatform.org також активно використовує SMW.

Дане Wiki-середовище надає ряд можливостей для роботи з інформацією на семантичному рівні, тобто підтримує семантичне анотування інформації та тематичний пошук і виконання складних запитів.

Semantic MediaWiki призначена для того, щоб вирішувати проблеми традиційних Wiki-систем і надавати користувачам можливість працювати з типізованими структурованими даними. Цій цілі в Semantic MediaWiki служать категорії, семантичні властивості і запити. З їхньою допомогою можна, наприклад, прив'язати до Wiki-статей, присвяченим містам, інформацію про їхнє населення, а потім автоматично сформувати список міст-мільйонерів, відсортованих по убутанню чисельності населення. Категорії, що використовуються для структурування традиційної MediaWiki, також можуть брати участь у запитах.

Semantic MediaWiki, додатково до звичайних властивостей Wiki-ресурсів, дозволяє користувачеві автоматично інтегрувати інформацію з різних Wiki-сторінок, генерувати відповіді на складні семантичні запити, будувати бази знань та візуалізувати їх результати, що відображають знання щодо семантичних властивостей та категорій інформаційних об'єктів, виконувати логічне виведення тощо, тобто обробляти інформацію на рівні знань.

Ці можливості базуються на таких інформаційних об'єктах, як семантичні властивості, категорії, шаблони та форми, які використовуються для структурування інформації, та семантичні запити.

У Semantic MediaWiki використовуються такі додаткові елементи розмітки, як

- *семантичні властивості* (для створення даних),
- *семантичні запити* (для використання даних).

Semantic MediaWiki дозволяє давати посиланням імена, перетворюючи їх у так звані "семантичні властивості" сторінки. Більш того, вона дозволяє розмічати й інші фрагменти тексту, прив'язуючи до сторінки дані різних типів. Можливість анотування тексту дає користувачам більше можливостей у порівнянні з традиційними категоріями. Інформація стає доступною не тільки для читання, але і для автоматичної машинної обробки. Семантична розмітка тексту Wiki-сторінок забезпечує таку обробку, і тому семантичні властивості і категорії, що входять до складу Wiki-статті, часто називають "семантичними анотаціями". Такі семантичні анотації в Semantic MediaWiki є узагальненням механізму категорій у MediaWiki. .

Семантичні властивості не схожі одна на одну і мають різні типи – вони можуть бути чисельними, рядковими і навіть датами і географічними координатами.

Механізм *семантичних властивостей* є важливим елементом Semantic MediaWiki.

Семантичні властивості і їх типи є головними інструментами введення семантичних даних у Semantic MediaWiki. Властивості схожі на категорії в MediaWiki, однак, на відміну від категорій, у властивостей є не тільки ім'я,

але і значення. Для того, щоб призначити Wiki-сторінці властивість, потрібно відредагувати текст цієї сторінки, додавши до неї наступну конструкцію:

```
[[ім'я властивості::значення властивості | що виводити на екран]].
```

Цей синтаксис дуже схожий на створення звичайних Wiki-посилань (нагадаємо, що для цього в MediaWiki використовується конструкція виду [[Ім'я посилання]]). При додаванні семантичної властивості до тексту Wiki-сторінки відображатися буде тільки значення цієї властивості.

Користувачі розмічають статті категоріями і властивостями для того, щоб інформація ставала доступною для запитів.

Семантичні можливості значно розширюють функції, які може надавати Wiki-ресурс, спрощує роботу з наповнення контенту та забезпечує актуальність та цілісність інформації. Крім того, вони дозволяють обробляти інформацію на рівні знань, що стає основою для поєднання Wiki-ресурсів з іншими розподіленими застосуваннями.

Основа підходу Semantic MediaWiki полягає в доповненні Wiki-розмітки новими елементами. Семантичні властивості дозволяють іменувати гіперпосилання між сторінками і прив'язувати до сторінок типізовані дані. Убудовані об'єкти дозволяють додавати структуровану інформацію, не створюючи додаткових сторінок. Убудовані запити і концепти допомагають здійснювати доступ до даних, озброюючи користувача могутньою мовою запитів.

5.2. Семантичні властивості Semantic MediaWiki

Семантичні властивості використовуються для прив'язування даних до Wiki-сторінок. Кожна властивість має тип, назву і значення, а також власну Wiki-сторінку в спеціальному просторі імен. Ця сторінка використовується для того, щоб задавати тип властивості, визначати її місце в ієрархії властивостей, а також документувати те, як цю властивість необхідно використовувати.

У текст Wiki-сторінки семантична властивість вставляється у форматі [[назва::значення]]. Приміром, якщо потрібно додати на Wiki-сторінку про Леонардо да Вінчі інформацію про те, що він був мешканцем Італії, досить додати в текст статті про нього конструкцію [[країна::Італія]].

Семантичні властивості використовуються для того, щоб позначати посилання на інші Wiki-сторінки, які несуть певне змістовне навантаження. Приміром, Wiki-ресурс може містити сторінки, пов'язані з різними географічними об'єктами, що пов'язані один з одним. Наприклад, для сторінок “Київ” та “Україна” цей зв'язок можна назвати “столиця”

Для того, щоб давати назви таким посиланням, у Semantic MediaWiki і використовуються семантичні властивості. Для того, щоб зафіксувати такий зв'язок, досить замінити текст, що міститься на сторінці “Київ”

```
[[Україна]]
```

на

[[є столицею.:Україна]]

Властивості можуть бути різних типів, і саме ці типи визначають порядок операцій, що можуть виконуватися зі значенням властивостей, наприклад, сортування і порівняння. Властивості типу “Сторінка” подібні до властивостей іменованих гіперпосилань між Wiki-статтями, і тому в ранніх версіях SMW їх виділяли в окрему групу. SMW підтримує 15 типів властивостей типу: Число, Рядок, Дата, Географічні координати тощо.

Значний інтерес представляє наступне: якщо в текст шаблону помістити семантичну властивість, то цю властивість будуть мати всі сторінки, що використовують цей шаблон. Шаблони не можуть, однак, вирішити проблему різного написання тих самих даних, приміром, дати.

Код шаблону

```
<div class="awesomebox">{{{ім'я}}} народився [[дата народження:{{{дата народження}}}]]...
```

Результат

Леонардо да Вінчі народився 15.4.1452...

Таким чином, шаблони можуть використовуватися для більш зручного керування та використання семантичних властивостей на Wiki-сторінках

5.3. Плагін семантичних форм *Semantic Forms*

Призначення плагіну Semantic Forms

Semantic MediaWiki викликало великий інтерес у співтовариства розроблювачів MediaWiki, про що свідчить поява цілого сімейства плагінів, що залежать від SMW. До найбільш популярних відносяться *Semantic Forms*, *Semantic Result Formats*, *Semantic Drilldown*, *Halo*. Розширення дозволяють додавати в SMW нові формати введення і виводу даних, поліпшують підтримку логічного виводу, дозволяють використовувати семантичну інформацію для різних цілей: голосувань, ідентифікації користувачів, розмежування доступу.

Одне з найбільш популярних розширень – *Semantic Forms* – дозволяє створювати html-форми, за допомогою яких заповнюються шаблони MediaWiki. Параметри цих шаблонів стають значеннями семантичних властивостей, а сторінка, що включає в себе виклик такого шаблону, стає доступною для запитів. Крім того, *Semantic Forms* забезпечує можливість автодоповнення на основі властивостей сторінок, введення даних за допомогою елементів введення, специфічних для типів даних: випадючих списків, радіокнопок, карт, календарів.

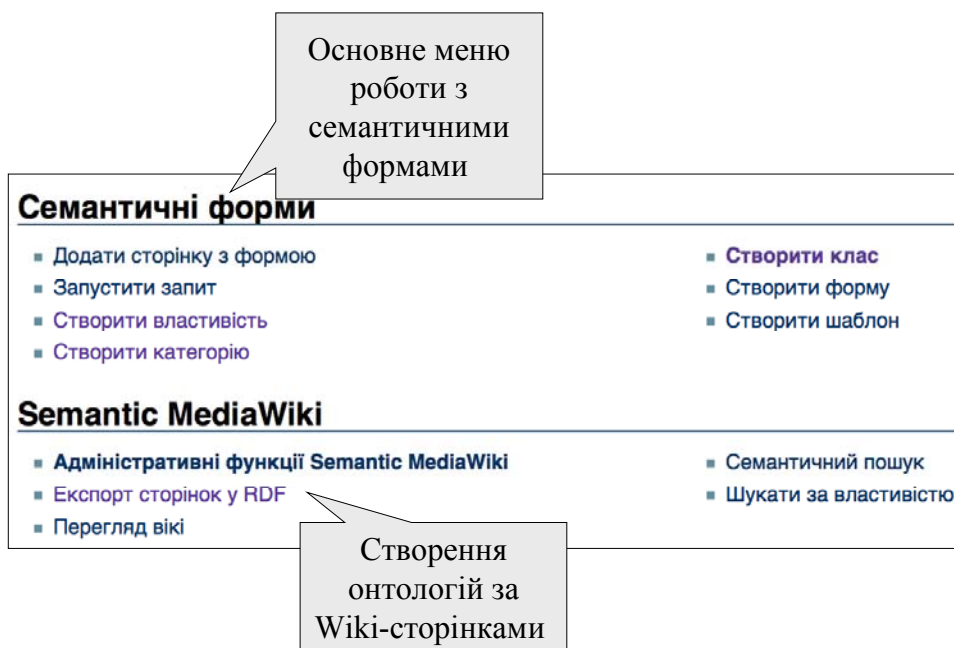


Рис.5.1. Основне меню плагіну *Semantic Forms*

Розширення *Semantic Result Formats* додає можливість виводити результати семантичних запитів у різних форматах, а також візуалізувати їх. У такий спосіб стає можливим розташовувати темпоральні дані на часовій осі і календарі, геодані – на картах (при включеному розширенні *Semantic Maps*), створювати хмари тегів, графи зв'язків Wiki-статей, інтерактивні графіки для кількісних даних тощо.

SMW доступна для завантаження як окремо, так і в складі пакета *Semantic Bundle*, що включає в себе саму SMW і семантичні розширення, що пройшли інтеграційне тестування. Комерційні компанії, що використовують MediaWiki як платформу, також часто включають SMW у свої збірки.

Semantic MediaWiki є ключовим компонентом Web-платформи *DataWiki* (у минулому *Enterprise Semantic MediaWiki+*), яку розробила компанія *DIQA-PM* для використання в корпораціях. Цей програмний продукт поставляється як за безкоштовною ліцензією для співтовариств, так і на основі оплати.

Основні елементи плагіну Semantic Forms

Плагін семантичних форм *Semantic Forms* надає Wiki-авторам різноманітні елементи введення, для того, щоб співтовариство не витратило час на ручну уніфікацію даних.

Він дозволяє створювати:

- шаблони;
- форми шаблонів;
- класи;
- семантичні властивості;
- категорії.

Крім того, він автоматизує включення форм та запитів до сторінок.

Синтаксис Semantic Forms викликає багато зауважень, як і весь синтаксис MediaWiki, однак це є одним зі способів уникнути створення форм вручну.

Створення семантичних властивостей

За допомогою сторінки “Створити властивість” можна досить просто створити нову семантичну властивість та пов’язати її з елементами певного типу.

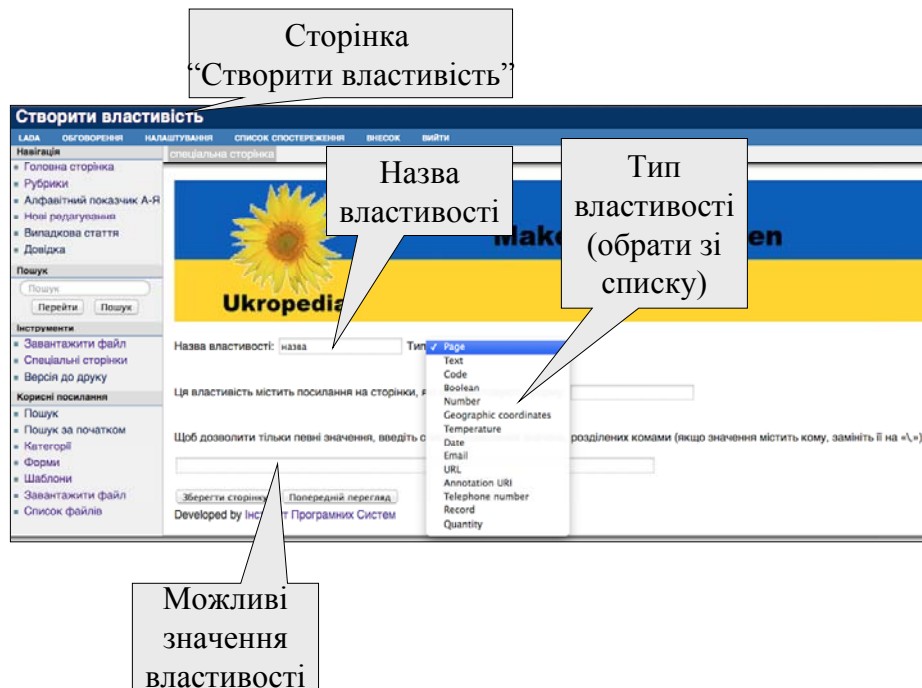


Рис.5.2. Створення семантичних властивостей

На основі такого вибору генерується наступний контент сторінки відповідної властивості:

Це властивість типу `[[Has type::Text]]`.

За необхідністю до неї можна додати будь-які інші пояснення або віднести її до однієї або кількох категорій.

Для простих властивостей це можна робити й вручну, але іноді зручніше обирати тип зі списку.

Надалі проглянути та відредагувати ці сторінки можна за допомогою засобів обробки спеціальних сторінок або вручну набравши її ім'я у браузері (у даному випадку – ім'я “`http://wiki.isoftware.kiev.ua/index.php/Property:назва`”).

У текст Вікі-сторінки семантична властивість вставляється у форматі `[[Ім'я властивості::Значення властивості]]`. Семантичні властивості використовуються для того, щоб визначити значення навантаження посилань на інші сторінки.

Наприклад, якщо на сторінці “Україна” зустрінеться `[[столиця::Київ]]`, те це означає, що поняття “Україна” і “Київ” зв'язані відношенням “столиця”.

Імена семантичних властивостей можуть використовуватися в шаблонах і в запитах до контенту Wiki-сторінок.

Створення форм шаблонів

Форма описується на Wiki-сторінці, що знаходиться в просторі імен Form, за допомогою html, Wiki-розмітки і спеціальних тегів. Її можна редагувати так само як і будь-яку іншу Wiki-сторінку. Форма – це засіб заповнення шаблонів, при її створенні потрібно враховувати параметри шаблонів.

Для кожного шаблону можна створити кілька різних форм, які визначають вигляд його представлення на екрані. Звичайно форму називають так само, як її головний шаблон, що використовується в ній.

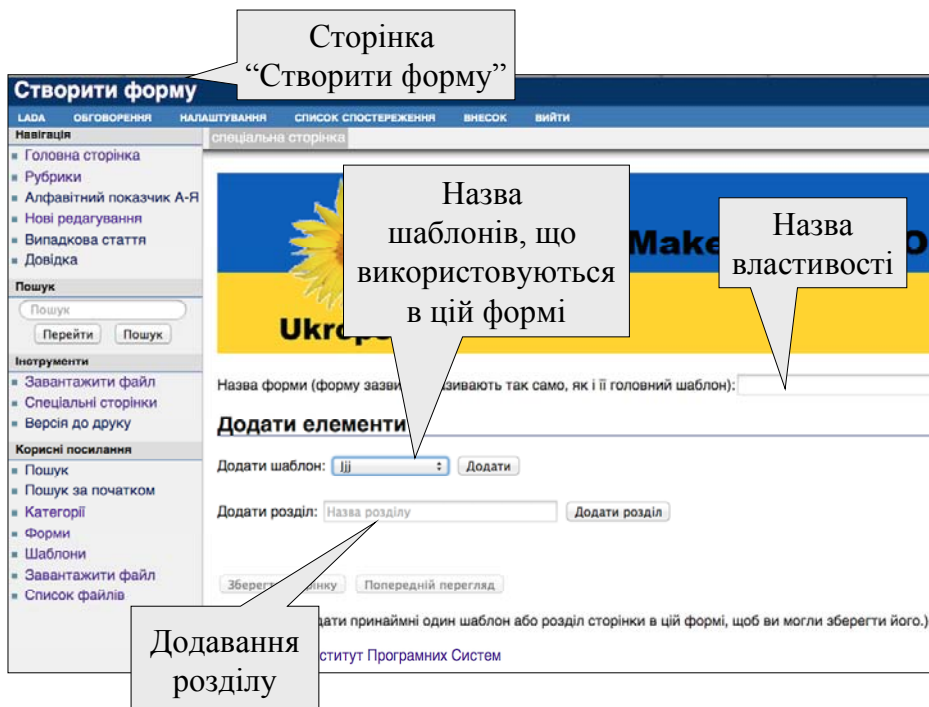


Рис.5.3. Створення форм

Форма дозволяє вказати розділ сторінки, до якого вона використовується. Можна вказати інші параметри, описати їх як обов'язкові, обмежені (які може редагувати тільки адміністратор) та приховані, вказати їх клас та задати кількість рядків та стовпчиків для їх введення або ж дозволити змінювати розмір поля введення відповідно до обсягу введеного тексту.

Після цього зберігається сторінка з відповідним ім'ям, приміром, “<http://wiki.isoftware.kiev.ua/index.php/Форма:Jjj>”.

Контент такої сторінки містить наступний текст

```
<noinclude>
```

Це форма «Jjj».

Обробка семантики у Semantic MediaWiki

Щоб створити сторінку з допомогою цієї форми, введіть нижче назву сторінки;

якщо сторінка з таким ім'ям вже існує, ви будете направлені на форму редагування цієї сторінки.



Рис.5.4. Основні елементи форми

```
{{#forminput:form=Jjj}}  
  
</noinclude><includeonly>  
<div id="wikiPreview" style="display: none; padding-bottom: 25px;  
margin-bottom: 25px; border-bottom: 1px solid #AAAAAA;"></div>  
==I==  
{{{section|I|level=2}}}  
"Довільний текст:"  
{{{standard input|free text|rows=10}}}  
{{{standard input|summary}}}  
{{{standard input|minor edit}}} {{{standard input|watch}}}  
{{{standard input|save}}} {{{standard input|preview}}} {{{standard  
input|changes}}} {{{standard input|cancel}}}  
</includeonly>
```

Звичайно текст форми має наступну структуру :

{{{info| <Історична особистість[Повне ім'я]> }}} <!-іотут описуємо форму в цілому: як буде виглядати заголовок форми, з яких полів буде братися назву сторінки й ін. -i>

```

{{{for template | Історична особистість}}} <!--івказуємо, тут, який шаблон буде заповнювати форма-і>

```

```

{{{field | Повне ім'я | input type=text }}} <!--іописуємо поля форми -і>

```

```

{{{field | Дата народження | input type=calendar}}}

```

```

{{{field | Дата смерті | input type=calendar}}}

```

```

{{{field | Досягнення | input type=textarea}}}

```

```

{{{end template}}} <!--ізакриваючий тег для шаблону-і>

```

```

{{{standard input|save}}} {{{standard input|cancel}}} <!--іМалюємо низ форми, наприклад, кнопки Зберегти і Скасування-і>

```

Головний тег у Semantic Forms – це звичайно field. Він формує елемент введення: текстове поле, випадаючий список, календар для введення дати тощо. Ім'я елемента введення повинне збігатися з ім'ям параметра шаблону, наприклад календар {{{field | Дата народження | input type=calendar}}} буде заповнювати параметр {{{Дата народження}}} шаблону Історична особистість. У цього тега є величезна кількість параметрів, що дозволяють завантажити деякий текст за замовчуванням, зробити його обов'язковим до заповнення, встановити список припустимих значень тощо.

Для Semantic Forms існує багато різних видів введення. Частина з них знаходиться безпосередньо в Semantic Forms, частина винесена в окреме розширення Semantic Forms Inputs.

Звичайно форму створюють у вигляді таблиці. Підписи до елементів введення робляться звичайною розміткою чи за допомогою html.

Сторінка в просторі імен Form – це лише вихідний код форми. Для того, щоб запустити форму, існує кілька способів.

Можна використовувати службову сторінку Special:FormEdit/Ім'я_форми.

Можна використовувати функцію {{#formlink: form=Ім'я_форми|link text=підпис_на_посиланні}}, що сформує посилання на форму, і дозволить передати їй різні параметри

Можна використовувати функцію {{#forminput: form=Ім'я_форми|button text=підпис на кнопці}}

6. Побудова запитів до Wiki-ресурсів

У Semantic MediaWiki є проста, але потужна мова запитів *SMW-QL*, що відкриває широкі можливості для семантичного пошуку у Wiki-ресурсах. У той час як семантичні властивості і категорії дозволяють структурувати дані у Wiki, запити потрібні для того, щоб використовувати цю інформацію: вони допомагають Wiki-користувачам і Wiki-адміністраторам комбінувати дані і візуалізувати їх. Зрозуміло, всі відвідувачі Wiki не зобов'язані вивчати цю мову запитів, і можуть навіть не знати про його існування. Однак і вони можуть відчувати різницю в роботі із сайтом на Semantic MediaWiki завдяки можливості зберігати вбудовані запити прямо в тексті Wiki-статті.

Мова запитів SMW-QL дозволяє, по-перше фільтрувати сторінки за заданими критеріями, і по-друге, виводити як результати запиту тільки цікавлячого користувача інформацію, а не весь текст Wiki-сторінки.

Результати таких запитів будуть динамічно інтегрувати актуальну інформацію з інших Wiki-сторінок і відображатися у зручному для користувача форматі – у вигляді діаграм, географічних карт, таблиць і схем тощо. Якщо сторінки, з яких здобуваються потрібні дані, будуть змінюватися, то результати запитів будуть також автоматично оновлюватися, забезпечуючи несуперечність і погодженість даних.

Як правило, такі запити будують автори Wiki-контенту – адміністратори і редактори, що знають мову запитів, у той час як споживати і дивитися на результати запитів може будь-який відвідувач Wiki-сайту.

6.1. Побудова запитів на основі функції *ask*

Найчастіше використовуються вбудовані запити, що сполучені з функцією *ask*. Ця функція використовується так само як і інші функції синтаксичного аналізатора MediaWiki: її виклик позначається подвійними фігурними дужками, перед іменем ставиться символ #, а після – двокрапка. Спочатку передається сам рядок запиту, що відбирає потрібну інформацію з Wiki, а потім усі параметри запиту, розділені символами вертикальної риси |.

Запити на SMW-QL найчастіше вбудовуються в код Wiki-сторінки в наступній формі:

```
{{#ask:Критерії вибору сторінок  
|?Запитувана властивість  
|інші параметри  
}}
```

Функція *ask* має три параметри.

Перший параметр відбирає з усієї множини сторінок лише ті, серед яких здійснюється пошук. Приміром, це може бути категорія або множина категорій, до яких входить сторінка, певні умови, яким має відповідати сторінка. У секції критеріїв вибору сторінок указуються значення

семантичних властивостей і категорій, якими володіють цікавлячі нас сторінки. При цьому для вказівки цих критеріїв використовується та ж форма запису, що і для завдання властивостей і категорій: `[[властивість::значення властивості]] [[Категорія:назва категорії]]`.

Другий параметр визначає, які саме семантичні властивості потрібно знайти на обраних сторінках

Третій параметр вказує на те, яку інформацію потрібно надати користувачеві, та форма її подання.

Зазвичай користувачів цікавлять не стільки назви сторінок, що відповідають умовам запитів, скільки інформація, яка представлена на цих сторінках. Іншими словами, потрібно отримати значення певних семантичних властивостей цих сторінок. Для цього можна використовувати принтауті (`printout statements`): після тіла запити перелічувати цікавлячі нас властивості, випереджаючи їх знаком питання:

```

}}
{{#ask: [[Категорія:Місто]] | ?Кількість населення | ?Час створення
}}
або
{{#ask: [[Категорія:Місто]] [[Населення::>1000000]]
|?Ім'я мера
|format=broadtable
}}

```

Наприклад, запит має визначити населення та площу всіх міст України. Він оброблятиме тільки ті сторінки, які відносяться до категорії “Місто”. У запиті використовується семантична властивість “Знаходиться в” та її значення “Україна”. Для тих сторінок, які мають таку властивість зі вказаним значенням, до результатів запити будуть додаватися відомості щодо значень семантичних властивостей “населення” та “площа”.

Такий запит матиме наступний вигляд:

```

{{#ask: [[Категорія:Місто]] [[Знаходиться в::Україна]]
| ?населення
| ?площа#km? = Size in km?
}}

```

У цьому простому прикладі семантичного пошуку задається єдина умова `[[Знаходиться в::Україна]]` для того, щоб описати, які саме сторінки потрібні.

Слід звернути увагу, що при запиті в SMW ми використовували в точності той же текст, як при описі сторінки. Це загальна схема: коли потрібно позначити сторінку як об'єкт, що знаходиться в Україні, ми пишемо на ній `[[Знаходиться в::Україна]]`; а якщо потрібно запитати у Wiki всі об'єкти, що знаходяться в Україні, у тілі запити слід написати те ж саме: `[[Знаходиться в::Україна]]`. Синтаксис для запиту сторінок, що

Побудова запитів до Wiki-ресурсів

задовольняють деякій умові, у точності такий самий як синтаксис для явного твердження, що ця умова виконується.

6.2. Кон'юнкція умов запитів

Можна використовувати умови, які стосуються різних семантичних властивостей. Іншими словами, коли багато умов зазначено в одному запиті, результат звужується до тих сторінок, що задовольняють усім вимогам. Таким чином, це є логічне ТА (кон'юнкція умов запитів).

Приміром, запит може містити

```
[[Категорія:Місто]]
[[Знаходиться в.:Україна]]
[[Населення.:1 000 000]]
```

Такі запити видають тільки ті статті, про які в точності відомо, що вони задовольняють необхідним властивостям: якщо сторінка не містить семантичної властивості “Населення”, то її не буде обрано.

Приклад запити:

```
Список міст-мільйонерів, заснованих більше тисячі років тому:
{{#ask: [[Категорія:Місто]]
[[населення.:>1000000]]
[[час заснування.:<1.01.2016]]
|format=ul
}}
```

При визначенні значень властивостей SMW звичайно ігнорує будь-які початкові і кінцеві пробіли.

Типи даних, такі як число, можуть мати додаткові властивості, такі як ігнорування ком (чи крапок), що можуть використовуватися для відділення розряду тисяч (залежить від регіональних налаштувань). Також SMW трактує синонімічні імена сторінок у такий же спосіб, як і MediaWiki: звичайно вважає Semantic wiki, Semantic_wiki, і semantic wiki посиланнями на ту саму сторінку.

Якщо часто використовувати однакові набори умов, то доцільно створити Концепт як скорочення. Концепти являють собою деякого роду віртуальні категорії і можуть, таким чином, використовуватися в умовах подібно категоріям.

Крім того, можна обмежити кількість результатів запити. Наприклад, такий запит видасть не більше 4 назв міст з населенням менше 20000.

```
У нас є маленькі міста, наприклад {{#ask: [[Категорія:Місто]]
[[населення.:<20000]] | limit=4 }}
```


6.3. Використання шаблонів і компараторів у запитах

У попередньому прикладі задані чіткі умови на властивості – приміром, міста з населенням рівно в один мільйон. На практиці частіше використовуються не конкретні значення, а цілий діапазон, наприклад "всі міста з населенням більше одного мільйона". У деяких випадках можуть навіть шукати всі сторінки, що мають довільні значення для заданої властивості. Наприклад, що вмерли люди – це всі ті, хто мають будь-яке значення для властивості «дата смерті». Задавати подібні загальні умови можна за допомогою компараторів і шаблонів.

Шаблони записуються як "+" і допускають довільне значення для заданої умови. Наприклад, [[Народитися в::+]] видасть усі сторінки, що мають яке-небудь значення для властивості «Народитися в».

Компаратори – це спеціальні символи, подібні < чи >. Вони містяться після :: в умовах на властивості. SMW підтримує наступні компаратори починаючи з версії 1.5.3:

>> і <<: "більше" і "менше";

> і <: "більше чи дорівнює" і "менше чи дорівнює" за замовчанням, але "більше" і "менше", якщо \$smwStrictComparators = true;;

? і ?: "більше чи дорівнює" і "менше чи дорівнює";

!: "не" ("не дорівнює");

~: порівняння «подібності» для рядків;

!~: порівняння «несхожості» для рядків.

Компаратори працюють тільки для значень властивостей, але не для умов на категорії. Wiki може обмежувати те, які компаратори доступні, що виконується адміністратором модифікацією значення \$smwgQComparators як це пояснено в розділі Help:Конфігурація.

У залежності від значення \$smwStrictComparators інтерпретація > і < може розрізнятися, це документоване на сторінці про строгі компаратори.

При застосуванні компараторів до сторінок використовуються заголовки сторінок (без префікса простору імен). Однак це можна змінити установкою іншого ключа сортування MediaWiki для цих сторінок, наприклад, {{DEFAULTSORTKEY: користувальницький ключ}}. це застосовується до всіх компараторів, включаючи ! і ~. Неможливо мати кілька ключів сортування для однієї сторінки. Зокрема, сторінки переспрямування не приймаються до уваги при застосуванні компараторів.

Компаратор "Не дорівнює" дозволяє вибрати сторінки, що мають значення властивості, відмінне від заданого значення. Наприклад,

[[Телефонний код::~!101]]

вибере всі сторінки, що мають телефонний код, відмінний від «101». Помітимо, що цей опис запиту не враховує сторінки, що взагалі не мають телефонного коду.

Як і у випадку зі стандартним компаратором рівності, використання користувальницьких одиниць виміру може вимагати округлення в числових перетвореннях, що може привести до несподіваних результатів. Наприклад, [[площа::~!5555.00 м]] може усе-таки вибирати об'єкт, площа якого

Побудова запитів до Wiki-ресурсів

відображається як «55555.00 м», просто тому що точне числове значення в дійсності не 55555. У подібних ситуаціях більш корисно запитувати сторінки, що мають значення властивості за межами визначеного діапазону, що виражається диз'юнкцією від умов $z < i >$.

Для числових значень часто потрібно вибирати сторінки зі значеннями властивості у певному діапазоні. Наприклад, запит з умовами

```
[[Категорія:Місто]]
[[населення::>100000]] [[населення::<1000000]]
```

знайде всі середні міста – з населенням від 100000 до мільйона.

Такі діапазонні умови на значення властивостей найбільш актуальні, якщо значення можуть бути упорядковані природним образом. Наприклад, доцільно шукати [[дата початку::>22 липня 2016]], але важко знайти доцільне використання умови [[URL домашньої сторінки::>http://www.somewhere.org]].

Якщо тип даних не має природного лінійного порядку, Semantic MediaWiki застосовує алфавітний порядок для нормалізованих значень даних, як вони використовуються в експорті RDF. Ви можете, отже, використовувати умови більше і менше для вибору алфавітних діапазонів строкової властивості. Наприклад, ви можете запросити [[Прізвище::>Д]] [[Прізвище::<М]] для вибору прізвищ від «Д» до «М». Для Wiki-сторінок компаратор посилається на ім'я даної сторінки (без префікса простору імен).

Тут і у всіх інших використаннях компараторів може случитися, що значення для пошуку в дійсності починається із символу, подібного $<$. У цьому випадку можна запобігти інтерпретації символу SMW як компаратора, якщо вставити пробіл після $::$. Наприклад, [[Властивість::
]] насправді шукає всі сторінки зі значенням «
» для даної властивості.

Іноді потрібно виключити саме точне значення з результату, наприклад, щоб знайти міста, населення яких перевищує мільйон (приміром, до яких стосуються певні законодавчі акти). Тоді можна використовувати в запиті комбінацію з компаратора $>$ і компаратора $!$:

```
[[Категорія:Місто]] [[населення ::>1000000]] [[населення ::!1000000]]
```

Компаратор \sim працює тільки для властивостей типу Рядок. В умові подібності використовуються шаблони $*$ для позначення довільної послідовності символів і $?$ для позначення довільного одиночного символу. Наприклад, можна використати "[[Адреса::~~*проспект*]]" для вибору адрес, що містять рядок «проспект».

Цей компаратор дозволяє динамічно оновлювати контент сторінки. Приміром, можна додати до сторінки наступний текст:

Багато міст у Мексиці починаються зі слова Сан:

{{#ask: [[Категорія:Місто]] [[Країна::Мексика]] [[Назва::~сан-]] }}*
Тоді користувачі побачать назви міст з відповідними іменами.

Стандартна поведінка SMW для компараторів “<” і “>”, яка означають "менше чи дорівнює" і "більше чи дорівнює", може бути незручною для людей, що знайомі з математичними значеннями < i >. Тому можна змусити SMW інтерпретувати < i > "строго".

6.4. Об'єднання результатів запитів: диз'юнкція

Диз'юнкція – це умова ЧИ, що допускає кілька альтернативних умов на результати запиту. SMW має два способи написання диз'юнкції в запитах:

Оператор OR використовується для об'єднання двох запитів.

Оператор || використовується для диз'юнкції значень, імен сторінок і категорій.

У будь-якому випадку, диз'юнкція вимагає, щоб по всьому менше однієї (але може бути більше, ніж одна) з можливих альтернатив була задоволена (логічне ЧИ). Наприклад, запит

[[Народитися в::Київ]] OR [[Народитися в::Львів]]

описує всі сторінки про людей, що народилися в Києві чи Львові.

Це також може бути записане за допомогою || як

[[Народитися в::Київ||Львів]].

В останньому випадку «Київ||Львів» описує значення, що може бути кожним із двох альтернатив.

Написання запитів за допомогою || звичайно коротше, але не всі диз'юнкції можуть бути записані цим способом. Наступний запит – приклад того, що не може бути виражене за допомогою ||:

[[Народитися в::Київ]] OR [[Категорія:Місто]]

Синтаксис || може бути використаний не тільки для значень властивостей, але також для категорій, як у запиті

[[Категорія:Місто||Країна]]

6.5. Умови запитів щодо окремих сторінок

Дотепер всі умови залежали від семантичних властивостей, зазначених на сторінці. Але існують також умови для безпосереднього вибору якихось сторінок чи сторінок із заданого простору імен.

Безпосередня вказівка заголовка деякої сторінки (можливо, включаючи префікс простору імен) чи списку заголовків таких сторінок, розділених `||`, вибирає сторінки з цими іменами.

Приклад – це запит

```
[[Київ||Україна||Інститут програмних систем]]
```

який має три результати (принаймні, якщо ці сторінки існують).

Відзначимо, що результат не відображає ніяких префіксів простору імен; для визначення простору імен дивитися спливаючу чи підказку статусний рядок чи браузера пройдіть по посиланню. Обмеживши множину, ґрунтуючись на значеннях атрибутів, можна додати додаткові умови щодо семантичних властивостей обраних сторінок, приміром, рік їх створення.

Для вибору категорії таким способом потрібно помістити `:` перед її ім'ям. Це запобігає змішуванню `[[Категорія:Місто]]` (видати всі міста) і `[[[:Категорія:Місто]]` (видати категорію «Місто»).

Менш конкретний спосіб вибору заданих сторінок – через простори імен. За замовчанням видаються сторінки у всіх просторах імен. Для видачі сторінок у визначеному просторі імен воно задається за допомогою «шаблону», наприклад, пишеться `[[Help:+]]` для видачі всіх сторінок у просторі імен «Help». Основний простір імен звичайно не має префікса, тому для вибору сторінок тільки в основному просторі імен пишуть `[[:+]]`.

Диз'юнкція також працює із синтаксисом `||`, як і вище. Наприклад, для видачі сторінок в основному просторі імен чи в «User» пишуть `[[:+||User:+]]`. Для видачі сторінок у просторі імен «Категорія» знову необхідно поставити `:` перед назвою простору імен для запобігання змішування, наприклад, `[[[:Категорія:+]]`.

6.6. Підзапити і ланцюжки властивостей

Перерахування множини сторінок для властивості обтяжливо і складно для підтримки.

Наприклад, для вибору всіх акторів, що народилися в містах України, можна написати:

```
[[Категорія:Науковець]]
```

```
[[Народитися в::Київ||Львів||Харків||...]]
```

Доцільніше для генерації списку всіх цих міст зробити ще один запит

```
[[Категорія:Місто]]
```

```
[[Розташоване в::Україна]]
```

а потім скопіювати і вставити результати в перший запит.

Але зручніше використовувати запит міст як підзапит у запиті щодо науковців для одержання бажаного результату безпосередньо. Такий підхід дозволить враховувати нову та актуальну інформацію – приміром, якщо якийсь населений пункт перейде до категорії «Місто». Замість фіксованого

списку імен сторінок для значення властивості новий запит, укладений у `<q>` і `</q>`, вставляється в умову властивості. У цьому прикладі можна написати так:

```
[[Категорія:Науковець]]
[[Народитися в.:
<q>
[[Категорія:Місто]]
[[Розташоване в.:Україна]]
</q>]]
```

Можливі довільні рівні вкладеності, хоча вкладеність повинна бути обмежена для окремих сайтів для забезпечення продуктивності. У ще одному прикладі для вибору всіх міст Європейського союзу можна написати:

```
[[Категорія:Місто]]
[[Розташований у.:
<q>
[[Є членом.:Європейський союз]]
</q>]]
```

У прикладі вище сконструювано ланцюжок властивостей «Розташований у» і «Є членом» для пошуку сутностей, що розташовані в чомусь, що є членом ЄС. Запити можуть бути написані в більш короткій формі для цього розповсюдженого випадку:

```
[[Категорія:Місто]]
[[Розташований в.Є членом.:Європейський союз]]
```

Цей запит має те ж саме значення, що і вище, але потрібно набагато менше спеціальних символів. Загалом, ланцюжки властивостей створюються списком усіх властивостей, розділених крапками.

У тих випадках, коли властивість повинна містити крапку у своєму імені, можна починати запит із пробілу для запобігання інтерпретації SMW цієї крапки спеціальним образом.

Слід зауважити, що неможливо використовувати підзапит для отримання списку властивостей, що потім використовуються в запиті.

6.7. Використання шаблонів і змінних у запитах

У запиті можуть бути використані довільні шаблони і змінні. Приклад – це критерії вибору, що відображають усі прийдешні події, ґрунтуючись на поточній даті:

```
[[Категорія:Подія]]
[[дата закінчення.:>
```

```
{{CURRENTYEAR}}-{{CURRENTMONTH}}-{{CURRENTDAY}}}}
```

Ще одна особливо корисна змінна для вбудованих запитів – це `{{FULLPAGENAME}}` – поточна сторінка з указівкою її простору імен, що дозволяє повторно використовувати загальний запит на багатьох сторінках.

6.8. Сортування результатів запиту

Часто корисно представляти результати запиту у певному порядку, наприклад, видавати список міст упорядкованим відповідно до чисельності населення. Для цього в Semantic MediaWiki можна використовувати формат виведення `Special:Ask`. Формат виведення визначає те, як повинні бути відображені результати запиту.

На спецсторінці `Special:Ask` при складанні запиту надається випадаючий список "Форматувати як", за допомогою якого можна вибрати потрібний формат виведення. Якщо писати запити вручну, то досить указати формат у параметрі запиту `format`. За замовчуванням результати запиту відображаються у вигляді таблиці, що відповідає `format=table`. Якщо ж результат запиту складається з одного стовпчика, за замовчанням він форматується як список, розділений комами (можна задати цей формат явно за допомогою `format=list`).

`Special:Ask` має простий інтерфейс для додавання однієї або більш умов для сортування результатів запиту. Ім'я властивості, за якою здійснюється сортування, вводиться в текстове поле, і можна вибрати порядок сортування за зростанням або за зменшенням значень.

За замовчанням SMW звичайно відсортовує результати в природному порядку для значень обраної властивості: числа сортуються чисельно, рядки сортуються за абеткою, дати сортуються хронологічно. Якщо не задані особливі умови сортування, результати будуть упорядковані за назвою своїх сторінок.

Можливо задати більше, ніж одну умова сортування. Якщо кілька результатів виявляться рівними за першою умовою сортування, тоді наступна умова буде використовуватися для їхнього упорядкування, і т.д. Наприклад, за допомогою наступного запиту можна побудувати список міст, впорядкованих за середньою кількістю дощових днів у році, а потім згрупувати їх за країнами розташування:

```
{{#ask: [[Категорія:Місто]]  
|?Кількість дощових днів  
|?Знаходиться в=Країна  
|sort=Знаходиться в,Кількість дощових днів  
|order=ASC,DESC }}
```

одержавши в результаті таблицю, в якій будуть наведені відомості щодо кількості дощових днів у містах, розташовані в алфавітному порядку щодо країн, в яких розташовані ці міста (приміром, спочатку це будуть міста

Австрії, потім Болгарії, Венісуели тощо. Для тих міст, в яких значення цього параметру співпадає, тобто розташованих в одній країні, порядок сортування буде залежати від кількості дощових днів у них.

Важливо відмітити, що семантичні розширення іноді надають додаткові формати виведення: приміром, якщо встановити розширення Semantic Maps, то можна виводити результати запиту на карті, а розширення Semantic Result Formats містить цілий список інших корисних форматів виведення: лінії часу, хмари тегів тощо.

Сортування запиту може впливати на результат запиту, тому що сортувати можна тільки за значеннями властивості, які сторінка дійсно має: у результаті будуть присутні дані тільки з тих сторінок, які містять значення семантичної властивості, яка використовується для сортування.

Отже, якщо запит упорядкований за певною властивістю (приміром, «Населення»), тоді SMW звичайно обмежить результати запиту тільки тими сторінками, що мають хоча б одне значення для цієї властивості (тобто тільки сторінками, для яких задане населення). З цієї причини, якщо запит не вимагає, щоб властивість було присутня в кожному результаті запиту, то SMW за замовчанням додає цю умова. Але SMW завжди буде намагатися спершу знайти властивість, що використовується для впорядкування, у заданому запиті і навіть можливо упорядкувати результати запиту по підвластивостях.

От деякі приклади, що ілюструють це:

```
[[Категорія:Місто]] [[Населення::+]]
```

Цей запит, впорядкований за властивістю «Населення», видасть назви міст з кількістю населенням, впорядкованою у зростаючому порядку. Результат запиту такої ж, як і без сортування.

```
[[Категорія:Місто]]
```

Цей запит, впорядкований за властивістю «Населення», також видасть назви міст з кількістю населення, впорядкованою у зростаючому порядку. Результат запиту може бути модифікований через умову сортування: якщо є сторінки міст, на яких на вказано кількість населення, то назви цих міст вже не з'являться в результаті.

```
[[Категорія:Місто]] [[знаходиться у країні.Населення::+]]
```

Цей запит, впорядкований за властивістю «Населення», видасть назви міст з кількістю населення, впорядкованою у зростаючому порядку щодо населення країн, де вони розташовані.

Результат запиту не змінився, але «Населення» тепер відноситься до властивості, що використовується у підзапиті.

Якщо властивість, що використовується для сортування, має для деяких сторінок більше, ніж одне значення, то ця сторінка все рівно з'явиться тільки один раз у списку результатів. Позиція, яку сторінка займе в цьому випадку, не визначається SMW і може відповідати кожному зі значень властивості. У вищенаведених прикладах це виникне, якщо якийсь місто буде кілька значень кількості населення або якщо якийсь місто розташоване в кількох країнах, для кожної з яких задане населення (приміром, наводиться кількість населення або державна приналежність міста у різні роки). Бажано уникати подібних ситуацій – можливо, за рахунок більш коректного вибору семантичних властивостей.

Результати запиту, відображувані в таблиці результатів, можуть також бути впорядковані динамічно шляхом натискання на сортувальні іконки, розташованих у заголовку кожного стовпчика таблиці. Ця функція вимагає включеного JavaScript у браузері і буде сортувати тільки відображувані результати. Тому, якщо, наприклад, запит видав двадцять найбільших міст світу за чисельністю населення, то такі засоби дозволяють відсортувати ці двадцять міст за алфавітом чи в зворотному порядку по чисельності населення, але запит природно не покаже двадцять найменших міст у світі при зворотному порядку сортування у стовпчику чисельності населення.

Динамічне сортування таблиць прагне використовувати той же порядок, який використовується в запитах SMW, і, особливо, упорядковувати числа і дати природним чином. Однак, алфавітний порядок рядків і імен сторінок може трохи відрізнитися від алфавітного порядку Wiki через те, що існує багато інтернаціональних алфавітів, що можуть бути впорядковані різним чином залежності від мовних налаштувань.

6.9. Посилання на результати семантичного пошуку

Посилання на результати семантичного запиту в Special:Ask можуть бути створені за допомогою убудованого запиту SMW. Не рекомендується створювати посилання безпосередньо, тому що вони дуже довгі і використовують специфічне кодування. Розроблювачам, що створюють розширення, що посилаються на Special:Ask, варто також використовувати внутрішні функції SMW для побудови посилань. Розуміння деталей кодування запитів SMW у посиланні є тому необов'язковим для використання SMW.

6.10. Приклади застосування запитів

Такий простий запит дозволяє знайти, де саме знаходяться навчальні заклади:

```
{{#ask:
[[Категорія:Competence_management]]
[[Категорія:educational_organization]]|
?situated_in}}
```


Для користувачів ця інформація візуалізується як таблиця наступним чином:

The image shows a search interface with the following elements:

- Callout 1:** "Імена знайдених Wiki-сторінок" (Names of found Wiki pages) pointing to the first row of the search results table.
- Callout 2:** "Знайдені значення семантичної властивості" (Found values of semantic properties) pointing to the "Situating in" column of the search results table.
- Callout 3:** "Текст запиту" (Search query text) pointing to the query text in the search box below the results.

пошук освітніх установ за умовами користувача	
Мелітопольський державний педагогічний університет імені Богдана Хмельницького	Situated in Мелітополь
Aspirantura IPS	Київ
Compet1 test	

Tit	Competence 300120
Мелітопольський державний педагогічний університет імені Богдана Хмельницького	
==Перелік базових компетенцій даного Wiki-ресурсу== {{{#ask: [[Категорія:Competence_management]] [[Категорія:Competence_management]] ?competence = назва компетенції }}	

Рис.6.1. Результат виконання запиту

Якщо потрібно, цей запит можна специфікувати додатковими умовами:

```

{{{#ask: [[Категорія:Competence_management]]
[[Категорія:Competence]]
[[Категорія:educational_organization]]
[[Знаходиться в::Україна]]
[[Рівень::вища освіта]] |
?situated_in}}}
    
```

Якщо користувачеві потрібно отримати більше інформації, то в запиті можна вказати більше параметрів для виведення, які дозволять специфікувати інформацію для користувача:

```

{{{#ask:
[[Категорія:Competence_management]]
[[Категорія:educational_organization]] |
?situated_in = знаходиться в |
?work_from = працює з |
?speciality_number = має ліцензію на спеціальність }}}
    
```

пошук освітніх ресурсів за умовами користувачів			
Aspirantura IPS	Situated in	Київ	
Compet1 test			
Мелітопольський державний педагогічний університет імені Богдана Хмельницького		Мелітополь	
Впорядкування			
	знаходиться в	працює з	мас ліцензію на спеціальність
Aspirantura IPS	Київ	1993	122
Compet1 test			
Мелітопольський державний педагогічний університет імені Богдана Хмельницького	Мелітополь		122
			118
			106

Рис.6.2. Результат виконання розширеного запиту

Можливості дозволяють впорядковувати результати запитів за значеннями всіх семантичних властивостей, що використовуються у запиті. Кожне семантичне значення типу “Сторінка є посиланням на відповідну Wiki-сторінку”.

назва ко	
Competence 000	
Competence 0003	
Competence 0004	
Competence 0005	
Competence 0006	
Competence 0007	
Competence 0008	
Competence 0009	
Competence 0010	
Competence 30011	
Competence 30012	
Competence 30013	
Competence 30014	
Competence 30015	
Competence 30016	
Competence 30017	122
Competence 30018	
Competence 30019	
Competence 30020	

Рис.6.3. Впорядкування результатів запиту

Семантичні запити надають можливість співставляти базові елементи моделі даних, такі як компетенції, спеціальності, кваліфікаційні рамки тощо,

та отримувати інтегровані відомості про ті інформаційні об'єкти, що задовольняють сукупності умов користувача.

Приміром, результат виконання наступного запиту щодо компетенцій у статтях:

```

{{#ask:
[[Категорія:Competence_management]] |
[[Категорія:Competence]] |
?competence_name = назва компетенції |
?speciality_number = номер спеціальності |
?qualification_frame_type = за класифікаційною рамкою }}
    
```

дозволяє знайти ті компетенції, що представлені в даному інформаційному ресурсі.

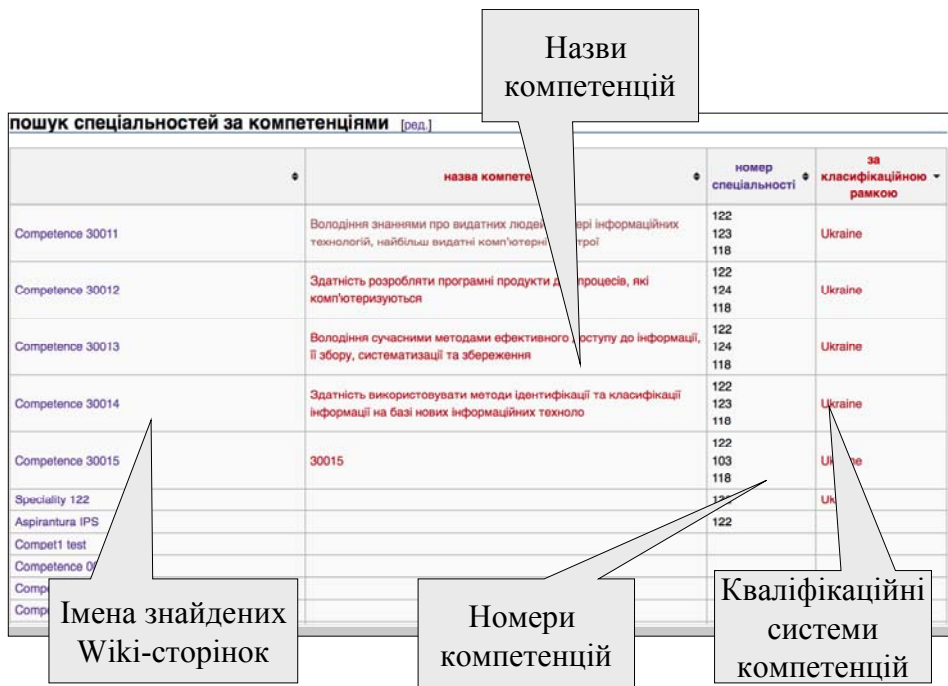


Рис.6.4. Приклад виконання запиту

Форму подання результатів та потрібну інформацію визначає автор запиту.

Він може зазначити категорії сторінок, серед яких здійснюється пошук, обрати необхідні семантичні властивості цих сторінок, та визначити бажані значення цих властивостей.

Внаслідок виконання такого запиту на відповідній Wiki-сторінці буде присутня інформація, яка потрібна користувачеві – значення обраних семантичних властивостей, які можна впорядкувати відповідно до потреб користувача.

	назва компетенції	номер спеціальності	за класифікаційною рамкою
Competence 30011	Володіння знаннями про видатних людей в історії інформаційних технологій, найбільш видатні комп'ютерні профі	122 123 118	Ukraine
Competence 30012	Здатність розробляти програмні продукти для процесів, які комп'ютеризуються	122 124 118	Ukraine
Competence 30013	Володіння сучасними методами ефективного доступу до інформації, її збору, систематизації та збереження	122 124 118	Ukraine
Competence 30014	Здатність використовувати методи ідентифікації та класифікації інформації на базі нових інформаційних техноло	122 123 118	Ukraine
Competence 30015	30015	122 103 118	Ukraine
Speciality 122		122	Ukraine
Aspirantura IPS		122	Ukraine
Compet1 test		122	Ukraine
Competence of			
Comp			
Comp			

Рис.6.5. Форма подання результатів запиту

Результати запиту, відображувані в таблиці результатів, можуть також бути впорядковані динамічно шляхом натискання на сортувальні іконки, розташованих у заголовку кожного стовпчика таблиці. Ця функція вимагає включеного JavaScript у браузері і буде сортувати тільки відображувані результати.

Динамічне сортування таблиць прагне використовувати той же порядок, який використовується в запитах SMW, і, особливо, упорядковувати числа і дати природним чином. Однак, алфавітний порядок рядків і імен сторінок може трохи відрізнитися від алфавітного порядку Wiki через те, що існує багато інтернаціональних алфавітів, що можуть бути впорядковані різним чином залежності від мовних налаштувань.

7. Пошук у Wiki-ресурсах

В ефективності використанні сучасних розподілених баз знань (БЗ) одним з визначних чинників є організація пошуку, що дозволяє користувачеві отримати доступ до потрібної інформації. При цьому важливі як час доступу, так і зрозумілість і зручність користувацького інтерфейсу. Знання, представлені інформаційними ресурсами, що організовані на основі Wiki-технологій, мають певну специфіку. Організація подання та збереження інформації на таких ресурсах визначає засоби та методи пошуку. Відомості, що представлені за допомогою Wiki-сторінок, з точки зору інформаційно-пошукових систем відносяться до частково структурованих: вони групуються за допомогою категорій Wiki, зазвичай містять посилання на інші Wiki-сторінки того самого ресурсу, та, якщо застосовуються семантичні Wiki, включають семантичну розмітку, тобто властивості інформації, що представлена на сторінці, охарактеризовані на семантичному рівні.

7.1. Типи пошуку у Wiki-ресурсах

Засоби пошуку у Wiki-ресурсах можна класифікувати на основні наступні групи:

1. **за ключовими словами** (для енциклопедій – за гаслом статті або за початковими літерами назви);
2. **за тематикою предметної області** (для енциклопедій – за категоріями та підкатегоріями статті);
3. **за типом інформаційного об'єкта** (для енциклопедій типи інформаційних об'єктів теж формалізуються через апарат категорій, конкретні значення яких здобуваються із узагальнення типів статей енциклопедії);
4. **за семантикою інформаційного об'єкта** (за його властивостями)

Перший варіант забезпечує найбільш швидкий доступ до інформації, другий базується на загальноновживаних класифікаторах (що дозволяє формалізувати класифікацію, але ускладнює пошук для пересічних користувачів), а третій – враховує семантику інформаційної потреби користувача та дозволяє ознайомитися із семантично близькими статтями.

Четвертий тип пошуку – це *семантичний пошук* [57], результатом якого має стати інформаційний об'єкт зі складною структурою, знання про яку використовуються в пошукових процедурах. Його можна розглядати як окремий випадок проблеми розпізнавання об'єктів, якщо Але у випадку енциклопедій повна реалізація потребує встановлення семантичних плагінів, призначених для оперування із семантичними властивостями статей.

У різних Wiki-ресурсах реалізуються різні підмножини видів пошуку. Це залежить від обсягу Wiki-ресурсу, складності його структури та від того, на яких саме користувачів орієнтовано цей ресурс. Приміром, планується, що всі ці підходи буде підтримувати електронна версія Великої української енциклопедії (далі – е-ВУЕ), яка має бути універсальною та поєднувати

різноманітні та гетерогенні джерела інформації. З іншого боку, невеликий спеціалізований Wiki-ресурс з досить простою структурою (приміром, сайт певної установи) може обмежуватися пошуком за ключовими словами.

7.2. Використання типів інформаційних об'єктів для пошуку у Wiki-ресурсах

У багатьох випадках для користувачів найзручнішим виявляється пошук за типом інформаційного об'єкта.

Інформаційний об'єкт (ІО) – це інформаційна модель об'єкта певної предметної області, що визначає структуру, атрибути, обмеження цілісності і, можливо, поведінку цього об'єкта.

У семантичному пошуку під ІО будемо розуміти ту інформацію, яку користувач отримує в результаті виконання процесу пошуку. Стосовно Wiki-ресурсів, в якості окремих ІО доцільно розглядати Wiki-сторінки – як звичайні, так і спеціальні (приміром, сторінки категорій)

Приміром, ІО у е-ВУЕ – це гасло енциклопедії та пов'язана з ним стаття. Кожен ІО однозначно ідентифікується унікальним ім'ям та характеризується наступними властивостями:

- Гасло;
- Контент;
- Підзаголовки;
- Категорії;
- Автор;
- Гіперпосилання на інші статті е-ВУЕ;
- Шаблони, що застосовуються у статтях;
- Семантичні зв'язки з іншими статтями е-ВУЕ (тільки для Wiki-ресурсів, що використовують Semantic MediaWiki).

Тип ІО – це множина ІО, що мають певні спільні властивості або призначені для подібних цілей. Приміром, кілька зображень можуть бути віднесені до типів ІО “Фотографія”, “Формат JPG” або “Зображення котів”. Для типізації ІО можуть використовуватися різноманітні критерії.

Стосовно енциклопедичних Wiki-ресурсів, *тип ІО* – це група статей енциклопедії, що мають спільні властивості. З точки зору пошуку доцільно групувати в один тип або статті однієї категорії (приміром, “Державні діячі”) чи множини категорій (приміром, “Фізика” та “Науковці”), або статті із подібними підзаголовками розділів. Типізація ІО може не співпадати з формально визначеними класифікаторами та відображати різні аспекти узагальнення інформації.

Доцільно виділити такі типи ІО, які найчастіше можуть зацікавити користувачів, що не знають точної назви потрібного терміну або цікавляться кількома близькими статтями, щоб спростити навігацію за базою знань (БЗ), якою з точки зору пошукових процедур є будь-який Wiki-ресурс.

У багатьох випадках доцільно групувати ІО ієрархічно. Приміром, у е-ВУЕ можна виділити наступні базові типи ІО верхнього рівня:

1. Особа

2. Група осіб
3. Організація
4. Інформаційний елемент
5. Документ
6. Абстрактне поняття
7. Діяльність
8. Предмет
9. Явище
10. Жива істота
11. Географічний об'єкт
12. Речовина
13. Властивість
14. Подія

У середовищі Wiki ці типи IO можуть бути представлені за допомогою апарату категорій.

Для цього потрібно створити базову категорію “Типовий IO”, а потім описати її підкатегорії.

На екрані сторінка такої категорії буде виглядати наступним чином:

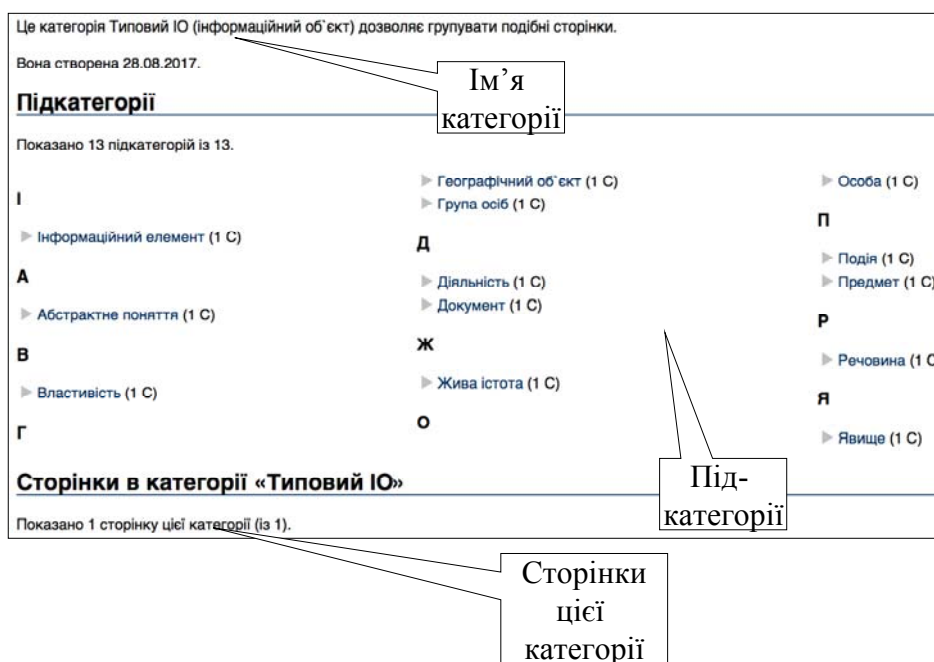


Рис.7.1. Wiki-сторінка категорії “Типовий IO”

У середовищі Wiki ці типи IO можуть бути представлені за допомогою апарату категорій.

Кожен з них може містити підтипи, що відповідають специфіці окремого типу IO або певній Про, до якої відносяться IO, що містяться у Wiki-ресурсі. Приміром, у е-ВУЕ типи IO верхнього рівня можуть бути спеціалізовані відповідно до окремих наукових напрямків.

Типи IO можуть бути конкретизовані для різних підгалузей. Приміром, для е-ВУЕ доцільно вводити спеціалізовані підтипи IO для кожного з 27

наукових напрямків (за класифікацією ВАК). Розглянемо це на прикладі. Для наукового напрямку “Географічні науки” можна наступним чином конкретизувати типи ІО:

- Особа
 - Вчені-географи
 - Зарубіжні вчені-географи
 - Вчені-географи
- Група осіб
- Інформаційний елемент
- Картографія
- Документ
- Картографія
- Абстрактне поняття
- Типи ландшафтів
- Географічний об’єкт
 - Річки
 - Річки світу
 - Річки України
 - Озера
 - Озера світу
 - Озера України
 - Водоспади
 - Пустелі
 - Світовий океан.
 - Моря
 - Океанічні течії
 - Острови. Архіпелаги
 - Півострови
 - Незалежні держави світу
 - Одиниці адміністративного поділу незалежних держав
 - Міста
 - Столиці незалежних держав
 - Міста України
 - Міста-мільйонники світу
 - Природно-заповідний фонд України
 - Залежні території
 - Материки.Континенти
 - ...

Деякі підтипи типових ІО не є специфічними для окремої Про, і тоді можна спочатку побудувати такі базові підтипи, а потім вже для таких загальних підтипів будувати підтипи, специфічні для окремих областей.

Приміром, для тип ІО “Особа” такими загальними підтипами може бути “Керівник”, “Науковець”, “Автор”, “Власник” та “Винахідник”. Але в конкретній Про такі підтипи можуть поділятися на більш специфічні, приміром, для наукового напрямку “Соціальні комунікації” тип ІО “Особа”

можна конкретизувати як “Персоналія-медійник”, що має підтипи “Видавець”, “Дизайнер”, “Журналіст” тощо.

Слід відмітити, що типи IO можуть перетинатися. Це пов’язано з тим, що той самий IO може відноситися одночасно до кількох різних категорій (приміром, стаття, що описує відому людину, може відноситися до категорій “Державний діяч”, “Науковець” та “Письменник”. Така багатокатегорійність пояснюється тим, що типізація IO є не таксономією, а *фолксономією*, тобто відображає не науковий підхід до класифікації IO, а тільки фіксує загальнозживані уявлення користувачів. Вона, крім того, не є вичерпною, і тому існує багато IO, для яких не побудовано специфічний тип IO.

Для типових IO доцільно виділити найбільш характерні параметри, які можна відобразити за допомогою семантичних властивостей Wiki. Для уніфікованого представлення таких властивостей доцільно розробити відповідні шаблони.

Наприклад, Wiki-сторінка, що описує прототип сторінки щодо видавництва, містить набір рубрикаторів та шаблон з основними параметрами такої сторінки (приміром, рік створення, країна тощо).

Зміст [сховати]

- 1 Гасло — дефініція
- 2 Історична довідка
 - 2.1 Заснування
 - 2.2 Етапи розвитку
- 3 Характеристика
 - 3.1 Специалізація
 - 3.2 Структура
 - 3.3 Книжкові серії
 - 3.4 Постати
- 4 N та Україна
- 5 Значення
- 6 Додатково
- 7 Дослідно
- 8 Література
- 9 категорії та типи інформаційних об'єктів

Гасло — дефініція [ред.]

Назва видавництва

У визначенні навести дані: стисла класифікаційна характеристика видавництва (велике, середнє, мале; універсальнє, спеціалізоване; діяльність).

Назва видавництва	Видавництво XXX
Логотип	
Рік заснування (період діяльності)	1990
Країна	Німеччина
Місто	Берлін
Спеціалізація	наукова література
Офіційний веб-сайт	www.abcd.org

Історична довідка [ред.]

Заснування [ред.]

Де, коли і ким було засновано; початкова назва.

Рис.7.2 Wiki-сторінка типового IO

Для цього створено шаблон “Видавництво”, який вставляється у сторінку наступним чином:

```

{{видавництво
|Назва видавництва=
|Рік створення=
|Країна=
|Тип=
}}
    
```

Для цього необхідно створити сторінку “Шаблон_Видавництво”, що містить наступний контент:

```
<noinclude>
```

Це шаблон «видавництво». Він повинен викликатися наступним чином:

```
<pre>
```

```
{{видавництво
```

```
|Назва видавництва=
```

```
|Рік створення=
```

```
|Країна=
```

```
|Тип=
```

```
}}
```

```
</pre>
```

Відредагуйте сторінку, щоб побачити текст шаблону.

```
</noinclude><includeonly>{| class="wikitable"
```

```
! Назва видавництва
```

```
| [[Назва::{{{Назва видавництва}}}}]
```

```
|-
```

```
! Рік створення
```

```
| [[Рік створення::{{{Рік створення}}}}]
```

```
|-
```

```
! Країна
```

```
| {{{Країна}}}
```

```
|-
```

```
! Тип
```

```
| {{{Тип}}}
```

```
|}
```

```
[[Категорія:соціальні комунікації]]
```

```
</includeonly>
```

Тоді кожна Wiki-сторінка, що відноситься до відповідного типу ІО, буде містити наступну інформацію (конкретні дані залежать від контенту сторінки та вводяться в процесі її редагування). Слід відмітити, що деякі параметри шаблону можуть бути пов’язані із семантичними властивостями, які посилаються на інші Wiki-сторінки (приміром, “Країна” та “Місто”) чи на дані (приміром, “Рік заснування”), а деякі – ні (приміром, “Логотип”).

Надалі значення ці семантичних властивостей можуть використовуватися у різноманітних запитах.

ІО також можуть бути згруповані за:

- Походженням;
- Формою існування;
- Призначенням;
- Часом виникнення

Необхідно постійно порівнювати розроблені типи ІО, призначені для загального вживання на Wiki-ресурсі, щоб досягнути уніфікації в іменах, що використовуються.

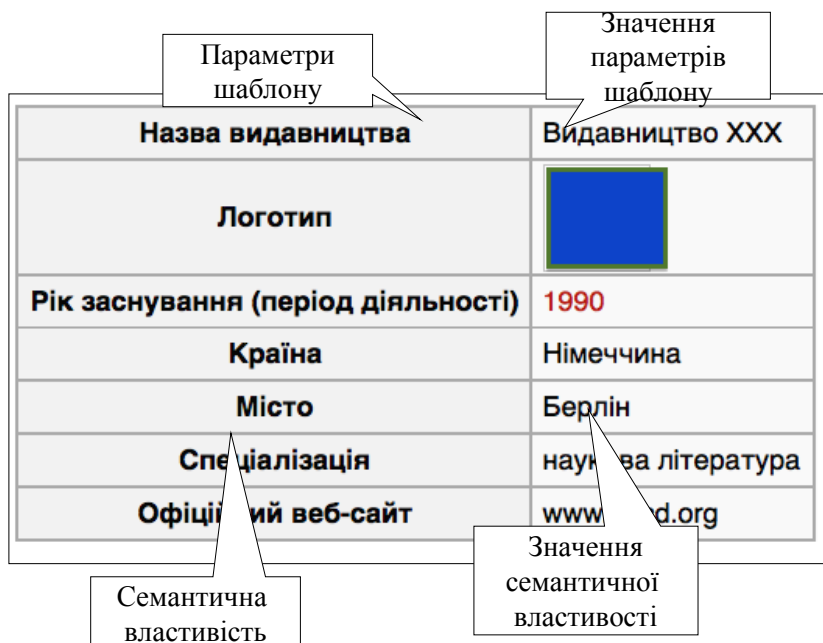


Рис.7.3 Подання шаблону на Wiki-сторінці типового ІО

Наявність різноманітних типів ІО значно покращить якість пошуку у Wiki-ресурсі, але створювати ці типи потрібно певною мірою системно.

Доцільно використовувати для цього два підходи:

- Зверху вниз;
- Знизу вверху.

Підхід “Зверху вниз” – від категорій верхнього рівня до їх підкатегорій, що більш чітко характеризують вміст (приміром, категорії соціальних комунікацій, які можна за потребою ще більше конкретизувати).

Підхід “Знизу вверху” – від статей до типів ІО. Створюючи нові статті або переробляючи існуючі, можна спробувати обирати для них тип ІО з точки зору користувачів. Приміром, за статтею “Австро-пруська війна 1866” е-ВУЕ можна створити тип ІО “Війна”, а за статтею “Автореферат” – тип ІО “Документ”. Потім ці типи можна використовувати й для опису інших статей е-ВУЕ.

7.3. Використання онтологій для побудови типів інформаційних об’єктів

Семантичне розпізнавання інформаційних об’єктів на основі онтологічного представлення знань про предметну область у задачах інтелектуального керування може використовуватися як теоретичний базис для побудови типів інформаційних об’єктів [43].

Створити несуперечну, повно та досить зрозумілу систему таких типів ІО надзвичайно складно навіть для людини, яка є експертом в певній ПрО, але не має навичок інженера зі знань. Крім того, при створенні системи з великою кількістю та складною структурою типів ІО недоцільно виконувати таку роботу вручну, а значно ефективніше використовувати методи та інструменти онтологічного аналізу, розглядаючи потрібні елементи

Пошук у Wiki-ресурсах

семантичної розмітки як класи, екземпляри та відношення онтології відповідної Про.

Важливо зафіксувати ієрархію типів ІО, тому що вона допомагає у навігації за БЗ та має стати основою для побудови складних запитів та сторінок-інтеграторів. Зручним засобом для цього може стати побудова онтології Wiki-ресурсу, яка формалізує семантику типів ІО та відношення між Wiki-сторінками. Таку онтологію можна надалі обробляти різноманітними програмними засобами, призначеними для менеджменту знань, визначати її властивості, візуалізувати знання, що містяться в ній. Саме ця онтологія може стати основним інструментом для спільного та розподіленого вдосконалення семантики Wiki-ресурсу.

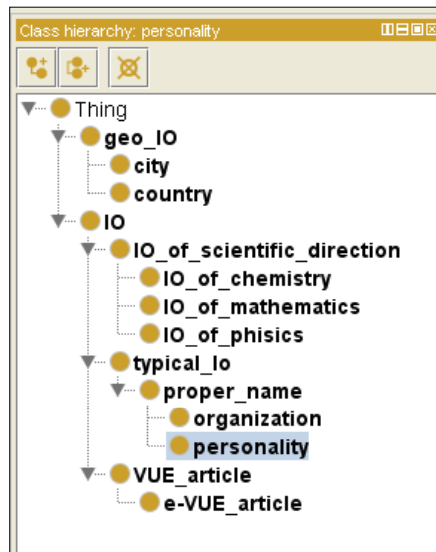


Рис.7.4 Класи онтології категорій в семантичному Wiki-ресурсі

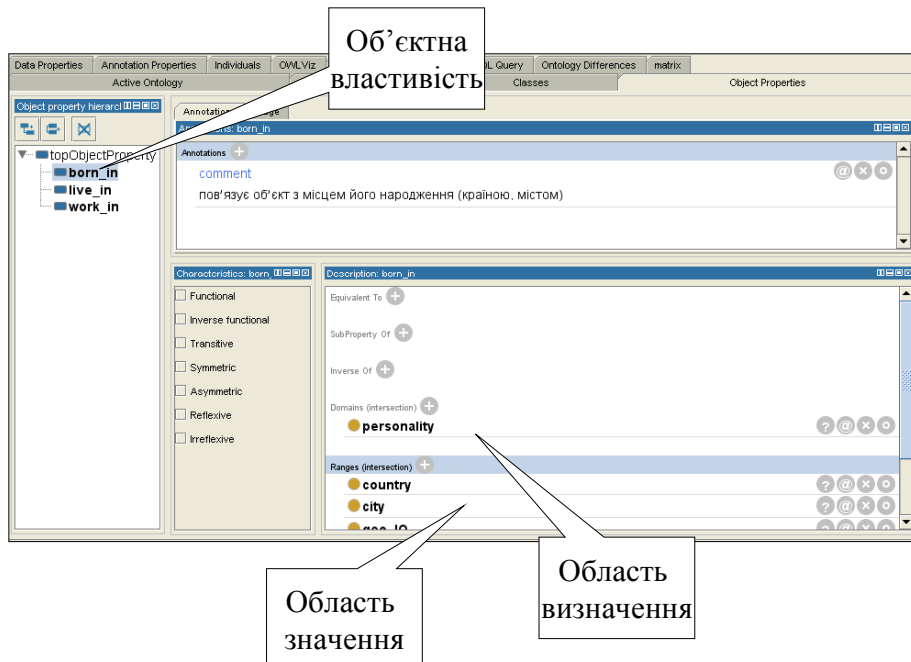


Рис.7.5 Об'єктні властивості онтології категорій в семантичному Wiki-ресурсі

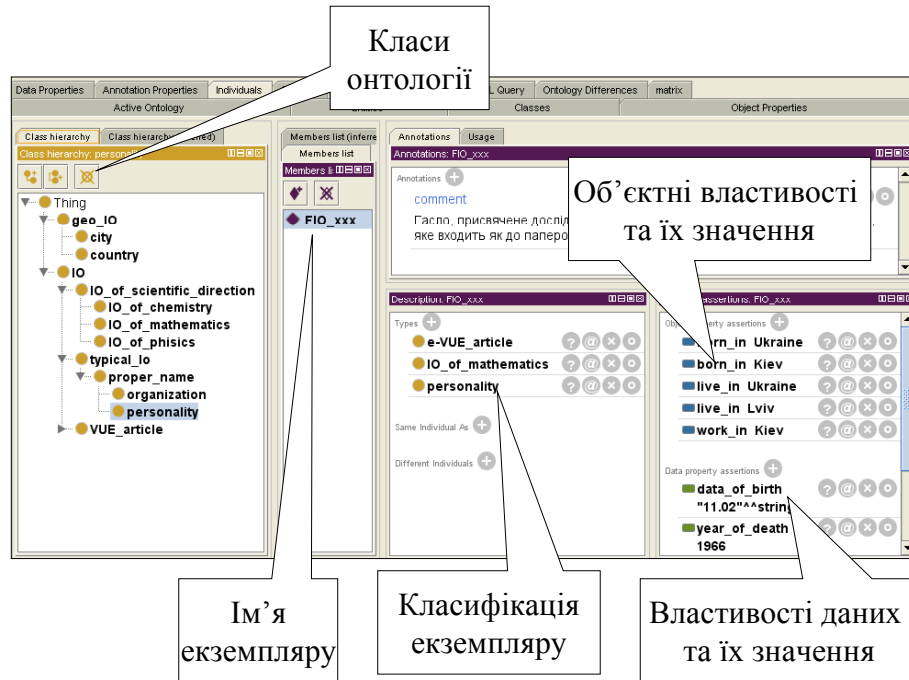


Рис.7.6 Екземпляри онтології категорій в семантичному Wiki-ресурсі

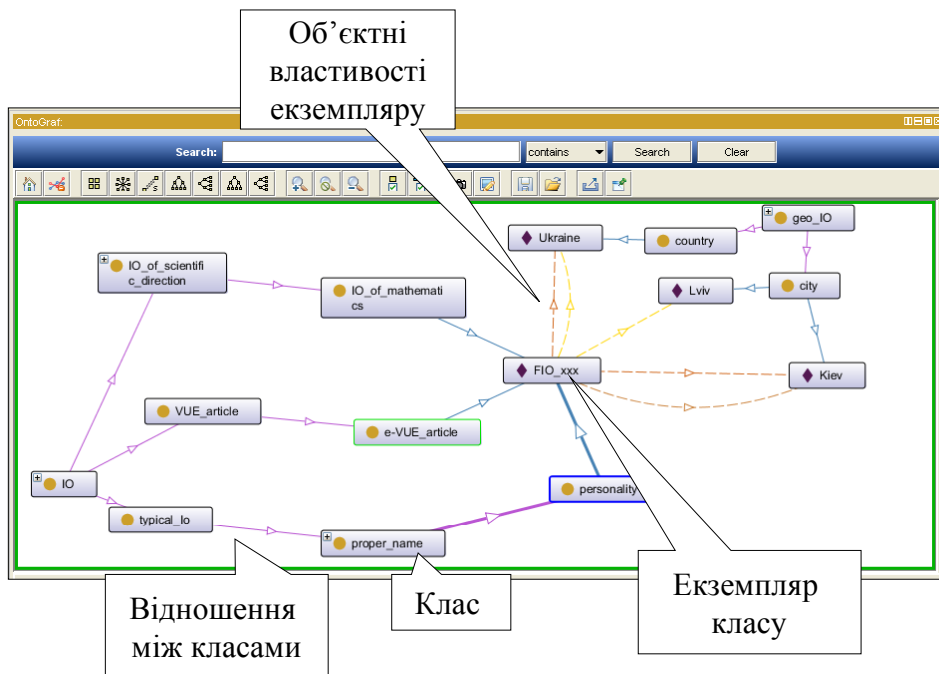


Рис.7.7 Візуалізація онтології категорій в семантичному Wiki-ресурсі

Для того, щоб надалі використовувати таку онтологічну структуру, потрібно розробити метод перетворення елементів онтології в конструкції Semantic MediaWiki (категорії і семантичні властивості). На наступному етапі необхідно забезпечити можливість уточнення й удосконалення початкової онтології ПрО на основі аналізу семантично розмічених Wiki-ресурсів, тобто розробити метод перетворення конструкцій Semantic MediaWiki в онтологію, представлену мовою OWL. Потім варто проаналізувати отриману онтологію, оцінити її властивості і відповідність представленням користувача щодо ПрО. Ітеративне повторення цих дій повинне забезпечити формування адекватної онтології ПрО, яку можна використовувати в різних ІІС як базу знань.

8. Інтеграція семантичних Wiki-ресурсів з онтологіями Semantic Web

Сучасні інтелектуальні інформаційні системи широко застосовують знання, здобуті з інформаційних ресурсів Web (приміром, за допомогою Web Mining та Text Mining). Наявність семантичної розмітки тих інформаційних ресурсів Web, що обробляються, значно спрощує цей складний процес, а застосування онтологій для представлення отриманих знань забезпечує їх повторне використання іншими системами.

Тому задача побудови семантичної розмітки на основі онтології певної Про та зворотна до неї задача побудови та вдосконалення онтологій на основі семантичних Wiki-ресурсів є сьогодні дуже важливими та актуальними.

8.1. Побудова онтологій засобами Semantic MediaWiki

Semantic MediaWiki орієнтовано на інтеграцію з технологіями Semantic Web. Дані, що задаються у Wiki-ресурсах семантичними властивостями, можуть бути експортовані у формат RDF. Типи даних SMW, у цьому випадку, перетворюються у відповідні типи даних з XML Schema, а унікальні ідентифікатори (URL) формуються шляхом приєднання суфіксів до URL Wiki. Також існує можливість явно вказати, які словники (OWL-онтології) слід використовувати при експорті тих або інших семантичних властивостей.

Семантичні властивості Semantic MediaWiki можна використовувати при створенні онтологій:

- для встановлення відношень між екземплярами класів (властивостей об'єктів) – для семантичних властивостей типу «Сторінка», семантика яких описана на сторінці відповідного властивості (наприклад, «працює в організації», «проживав у місті»),
- для виявлення властивостей даних для визначеної категорії сторінок (наприклад, на Wiki-сторінках категорії «Людина» можуть зустрічатися семантичні властивості «Рік народження», «Прізвище», «Ім'я»)

За замовчуванням значення семантичних властивостей зберігаються в додаткових таблицях бази даних MediaWiki, однак, починаючи з версії 1.6.0, для такої цілі можна також використовувати RDF-сховище (triplestore). Як основне сховище використовується 4Store, планується також підтримувати Virtuoso. У порівнянні з традиційним способом збереження, робота з RDF-сховищем дає наступні переваги:

- виникає можливість більш гнучко контролювати навантаження,
- стає можливою взаємодія з даними Wiki за допомогою мови запитів SPARQL,
- стає можливим використовувати движки логічного виводу по RDF-данам.



Рис.8.1. Wiki-сторінка семантичної властивості

У Semantic MediaWiki використовуються деякі можливості автоматичного логічного виводу нових фактів із уже наявних даних. Підтримується вивід по ієрархії семантичних властивостей і вивід по ієрархії категорій. Більш досконалі можливості логічного виводу стають можливими при установці розширення Halo і при використанні RDF-сховища.

Wiki-сторінки містять і інші елементи, що дозволяють формалізувати зв'язки між відповідними поняттями. До них відносяться:

- посилання на інші Wiki-сторінки;
- категорії.

У Semantic MediaWiki побудувати онтологію можна автоматизовано убудованими засобами системи.

Для цього в меню "Спеціальні сторінки" у підменю для "Semantic MediaWiki" і потрібно вибрати "Експорт сторінок у RDF".

За множиною обраних сторінок Wiki-ресурсу буде побудована онтологія мовою RDF. Таку онтологію можна надалі використовувати для семантичного пошуку як онтологію проблеми чи онтології Про та редагувати його контент.

Побудований у такий спосіб файл у форматі RDF можна переглядати, обробляти і редагувати за допомогою редактора онтологій Protege. При цьому певну проблему викликає лише кодування українських чи російських назв класів і екземплярів.

Цей файл можна використовувати як базу знань в інших інтелектуальних застосуваннях незалежно від того Wiki-ресурсу, на основі якого була побудована така онтологія.

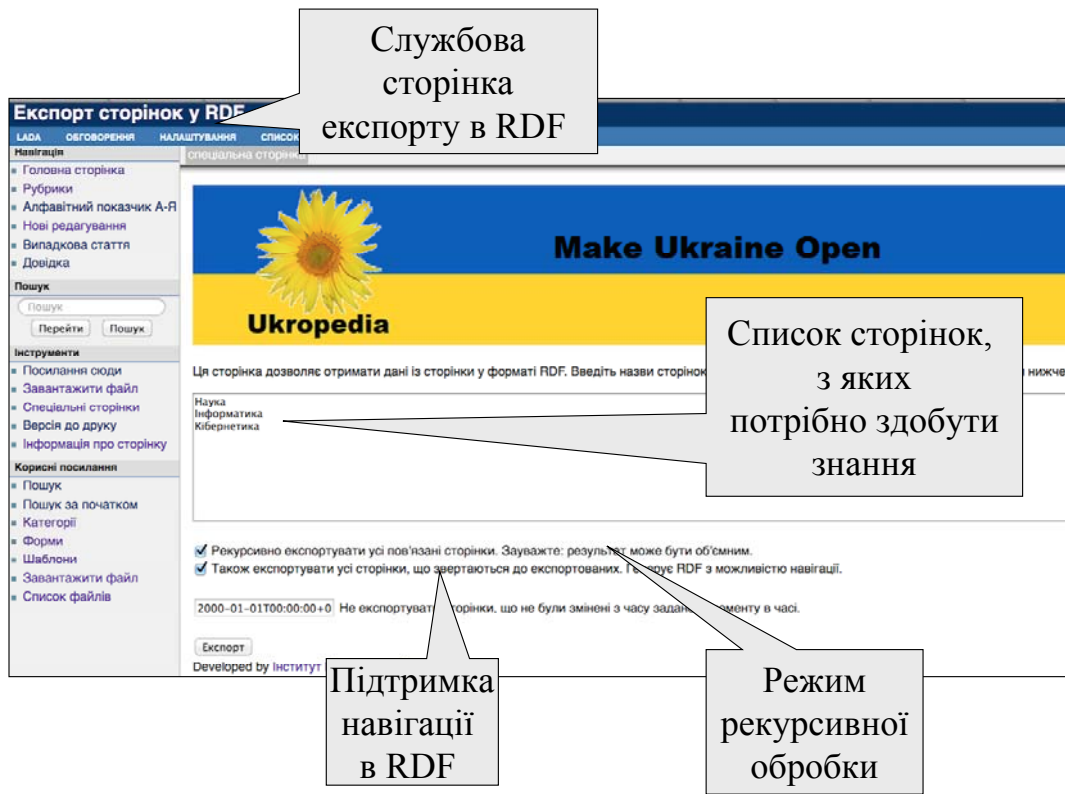


Рис.8.2. Експорт онтології з Semantic MediaWiki
Такий файл містить усі стандартні елементи RDF-файлу (рис.8.3).



Рис.8.3. RDF-файл, експортований з Semantic MediaWiki

Надалі цей файл можна проглядати та модифікувати за допомогою різних редакторів онтологій, приміром, за допомогою Protégé (рис.8.4).

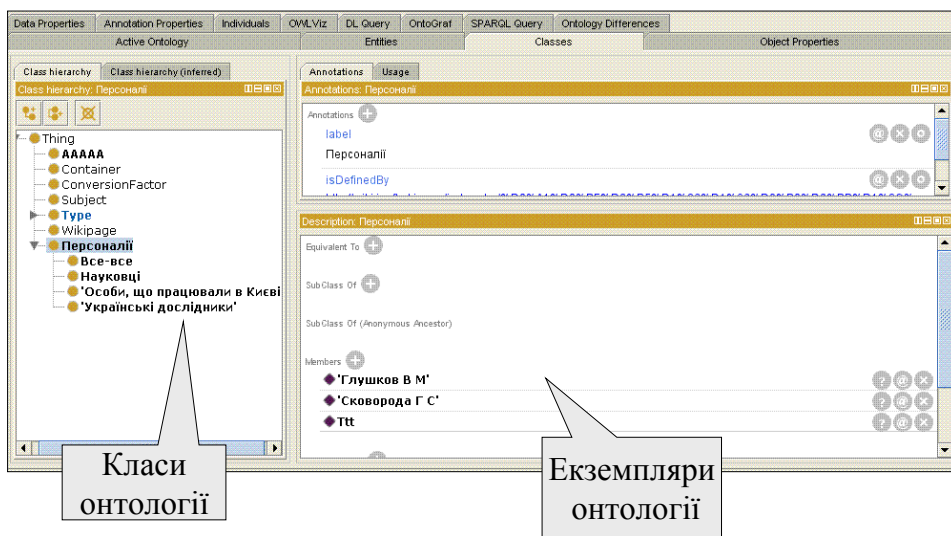


Рис.8.4. Прогляд RDF-файлу, побудованого в результаті експорту онтології з Semantic MediaWiki, за допомогою Protégé

Крім того, за допомогою убудованих засобів Protégé можна візуалізувати систему категорій, що застосовуються у Wiki-ресурсі, та представити її користувачам та розробникам контенту у наочній та зрозумілій формі.

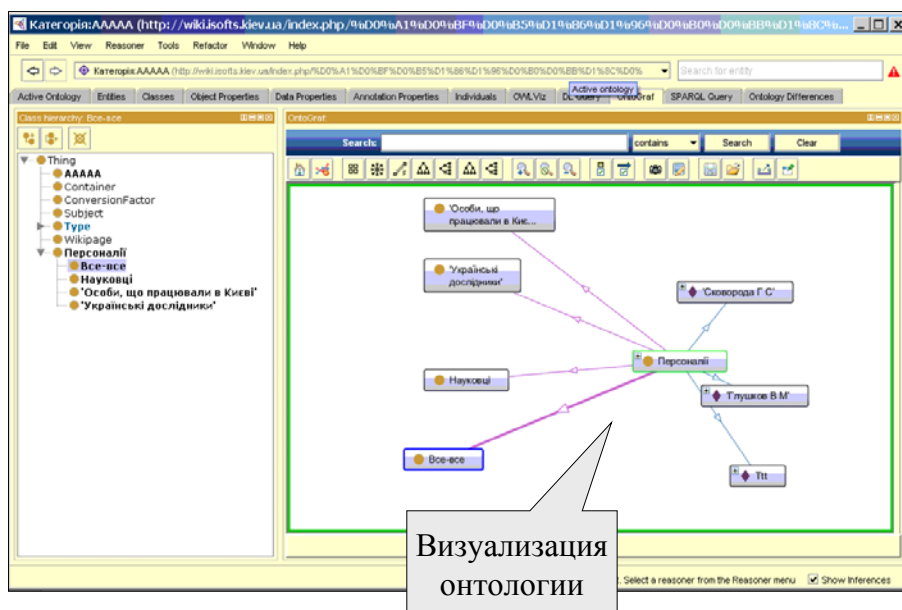


Рис.8.5. Візуалізація системи категорій, що застосовуються у Wiki-ресурсі, засобами Protégé

Таке подання знань про категорії Wiki-ресурсу забезпечить більш інтегровану та скоординовану роботу з інформаційними ресурсами.

8.2. Відповідності між елементами онтології та семантичного Wiki-ресурсу

Онтологічний аналіз дозволяє перетворювати опис певного представлення про зовнішній світ у набір термінів і правил їхнього використання, придатних для машинної обробки. Компоненти, з яких складаються конкретні онтології, залежать від парадигми представлення, але практично всі моделі онтологій містять набір концептів (поняття, класи та їх екземпляри), властивості концептів (атрибути, ролі), відношення між концептами (залежності, функції) і обмеження застосування, що визначаються аксіомами. Розглянуту вище формальну модель онтології $\text{PrO } O = \langle T, R, F \rangle$, де T – множина понять PrO ; R – множина відношень між ними; F – множина функцій інтерпретації понять і відношень можна конкретизувати для інтеграції з елементами Semantic MediaWiki наступним чином:

Будемо використовувати формальну модель [26]

$$O = \langle X = X_{cl} \cup X_{ind}, R = r_0 \cup \{r_i\} \cup \{p_j\}, F, T \rangle.$$

Ця модель складається з таких елементів:

- $X = X_{cl} \cup X_{ind}$ – множина концептів онтології, де X_{cl} – множина класів, X_{ind} – множина екземплярів класів, таких, що $\forall a \in X_{ind} \exists A \in X_{cl}, a \in A$;
- $R = r_{ier_cl} \cup \{r_i\} \cup \{p_j\}$ – множина відношень між елементами онтології, де r_{ier_cl} – ієрархічне відношення, що може встановлюватися між класами онтології і властивостями класів і характеризується такими властивостями, як антисиметричність і транзитивність, $r_{ier_cl} : X_{cl} \rightarrow X_{cl}$; $\{r_i\}$ – множина об'єктних властивостей, що встановлюють відношення між екземплярами класів: $r_i(a, a \in X_{ind}) = b, b \in X_{ind}$, $r_i : X_{ind} \rightarrow X_{ind}$; $\{p_j\}$ – множина властивостей даних, що встановлюють стосунки між екземплярами класів і значеннями: $p_j(a, a \in X_{ind}) = t, t \in T$, $p_j : X_{ind} \rightarrow \text{Const}$, причому усередині множин об'єктних властивостей і властивостей відносин також можуть існувати ієрархічні відносини $r_{ier_obj}, r_{ier_obj} : \{r_i\} \rightarrow \{r_i\}$ і $r_{ier_data}, r_{ier_data} : \{p_j\} \rightarrow \{p_j\}$;
- F – множина властивостей класів онтології, екземплярів класів і їхніх властивостей, що можуть застосовуватися для логічного виведення (наприклад, еквівалентність, відмінність);
- T – множина типів даних (наприклад, рядок, ціле).

Вибір саме такої моделі онтології обумовлюється наступними причинами. По-перше, вона найбільше повно відповідає інтуїтивному уявленню про

онтології, закладеному в користувацькому інтерфейсі найбільш популярного на даний час редактора онтологій Protege. По-друге, цю модель дозволяє зіставити онтологічне представлення знань щодо ПрО із семантичними конструкціями Semantic MediaWiki: для деяких елементів може бути сформоване взаємо-однозначна відповідність, а зіставлення інших потребує додаткових перетворень, але може бути частково автоматизовано.

Формальна модель Wiki-ресурсу складається з наступних елементів:

- множини Wiki-сторінок $P = P_{user} \cup P_{categ} \cup P_{template} \cup P_{spec}$, де P_{user} – множина сторінок, створених користувачами, P_{categ} – множина сторінок, що описують категорії, $P_{template}$ – множина сторінок, що описують шаблони, P_{spec} – множина інших спеціальних сторінок;
- $L = \{ "link" \}$ множина з одного елемента, що описує посилання однієї Wiki-сторінки цього ресурсу на іншу (хоча в Wiki-ресурсах передбачені і посилання на інші види сторінок, у рамках даної моделі вони не враховуються).

Формальна модель семантично збагачених Wiki-ресурсів є більш складною і включає ряд елементів, зв'язаних із семантичними властивостями:

$$W_s = \langle P = P_{user} \cup P_{categ} \cup P_{template} \cup P_{sem_prop} \cup P_{spec}, L = \{ "link" \} \cup L_{sem_prop} = \{ l_i \} \rangle$$

де P_{sem_prop} – множина сторінок, що описують семантичні властивості Wiki-сторінок, деякі з яких є семантично визначеними посиланнями на інші Wiki-сторінки: $P_{sem_prop_page} \subseteq P_{sem_prop}$, а інші зв'язують сторінки зі значеннями різних типів даних (ці типи даних визначаються на сторінках семантичних властивостей).

Фундаментальні поняття визначеної ПрО повинні відповідати класам онтології. Кожен екземпляр в онтології OWL належить до класу owl:Thing, а кожен створений користувачем клас автоматично є підкласом owl:Thing. Специфічні для ПрО кореневі класи визначаються простим оголошенням іменованого класу. OWL також дозволяє задати порожній клас: owl:Nothing. Визначення можуть бути розширюваними і розподіленими. Фундаментальним таксономічним конструктором для класів є rdfs:subClassOf. Він зв'язує окремий клас із загальним. Якщо X – підклас Y, то кожен представник X – також представник Y. Відношення rdfs:subClassOf є транзитивної. Якщо X – підклас Y і Y – підклас Z, тоді X – підклас Z.

Визначення класу складається з двох частин: назви (чи посилання на нього) і списку обмежень. Кожне вираження, що безпосередньо міститься у визначенні класу, уточнює властивості представників цього класу. Представники класу належать перетинанню зазначених обмежень. Для визначення *екземпляра* досить оголосити його членом якого-небудь класу.

Властивості дають можливість затверджувати загальні факти про члени класів і про екземпляри. Вони являють собою бінарні відношення.

Розрізняють відношення між представниками класів і RDF-літералами чи типами даних, визначеними XML Schema, і відношення між представниками класів.

При визначенні властивості існує багато способів обмежити це відношення. Можна задати його домен і діапазон. Властивість може бути визначена як спеціалізація (підвластивість) вже існуючої властивості. Можливі і більш складні обмеження. Властивості, також як і класи, можуть бути організовані ієрархічно. Формальна семантика OWL містить опис того, як одержати логічні висновки, маючи таку онтологію, тобто одержати факти, що не представлені в ній безпосередньо, але проте обумовлені її семантикою.

Семантику онтологічних мов звичайно розкривають через теорію моделей. Зокрема, вона визначає функцію інтерпретації, що проектує кожен елемент онтології на множину D , що називають областю інтерпретації.

Аналіз різних засобів представлення онтологій і їхніх формальних моделей, що пропонують технології Semantic Web, показує, що вони значно відрізняються своїми виразними можливостями і своєю складністю: RDF Schemas пропонує найпростіший рівень для представлення онтологій, а OWL Full – найбільш складний. Вибір засобу представлення онтології залежить від специфіки проблеми, для якої вона розробляється. Наприклад, для семантичного пошуку звичайно досить найпростішого рівня представлення знань, проте деякі задачі, зв'язані з перебуванням сукупності ІО зі складною структурою і численними умовами, вимагають більш складних можливостей для відображення знань відповідних ПрО.

При створенні семантичних Wiki-ресурсів необхідно спочатку сформувати набір категорій і семантичних властивостей. Але убудовані засоби Semantic MediaWiki не дозволяють ані візуалізувати цю інформацію, ані оцінити її цілісність і несуперечність. Тому доцільно спочатку побудувати онтологію ПрО, відображуваної в Semantic MediaWiki, а потім використовувати її як основу для семантичної розмітки. Крім того, семантично розмічені Wiki-ресурси набагато більш динамічні в порівнянні з онтологіями – у їхньому удосконаленні і відновленні може брати участь широке коло користувачів, і тому вони можуть бути корисні для удосконалення онтології ПрО. При цьому частина відповідностей взаємо-однозначні і можуть виявлятися автоматично, частину вимагають уточнень від користувача.

Таким чином, якщо вже побудована онтологія мовою OWL, тоді її досить просто використовувати в Semantic MediaWiki. Але зворотний процес не може бути цілком автоматизований. Більш того, при автоматичній генерації онтології по Semantic MediaWiki будуть загублені відомості, що містяться в OWL-онтології та стосуються характеристик класів і властивостей, які не мають аналогів Wiki (зокрема, про еквівалентність класів і властивостей, їхнього неперетину, про їхню область значення і визначення). У той же час, частину контенту Semantic MediaWiki не може бути безпосередньо трансформована в онтологію. Наприклад, той факт, що в сторінках використаний одні і той же шаблон, говорить про те, що на ці сторінках описані інформаційні об'єкти одного типу, але для відображення

цього в онтології треба створити специфічний клас і зв'язати його з елементом сторінки. Але потім по такій онтології неможливо зрозуміти, що треба створити в Wiki – шаблон чи категорію: вибір залежить від користувача, тому що для IO зі специфічної, але формалізованою структурою доцільно створювати шаблони, а у всіх інших випадках – сторінки, що будуть віднесені до якої-небудь категорії. Крім того, не можна зв'язати з класом онтології не всю сторінку, а її конкретний фрагмент (крім тих випадків, коли в нього є підзаголовок).

Розглянемо докладніше ці відповідності. Для цього треба проаналізувати основні семантичні елементи Semantic MediaWiki. Слід зазначити, що, хоча в Semantic MediaWiki передбачене автоматична побудова онтологій по семантично розмічених Wiki-сторінках, але в поточних версіях ця побудова враховує занадто мало семантичних параметрів, а його результати мало придатні для подальшого використання (таблиця 8.1).

Таблиця 8.1.

Semantic MediaWiki	Онтологія	Відображення з Wiki в онтологію	Відображення онтології в Wiki
Категорія	Клас	Взаємо-однозначне $P_{\text{categ}} \rightarrow X_{\text{cl}}$	Багатозначне $X_{\text{cl}} \rightarrow P_{\text{categ}} \cup P_{\text{template}}$
Ієрархія категорій	Ієрархія класів	Багатозначне	Взаємо-однозначне
Wiki-сторінка	Екземпляр класу	Багатозначне $P_{\text{user}} \rightarrow X_{\text{ind}}$	Взаємо-однозначне $X_{\text{ind}} \rightarrow P_{\text{user}}$
Посилання на іншу Wiki-сторінку	Об'єктне відношення	Взаємо-однозначне $L = \{\text{"link"}\} \rightarrow R$	Взаємо-однозначне $R \rightarrow L = \{\text{"link"}\}$
Семантична властивість типу «сторінка»	Об'єктна властивість	Взаємо-однозначне $P_{\text{sem_prop_page}} \rightarrow \{r_i\}$	Взаємо-однозначне $\{r_i\} \rightarrow P_{\text{sem_prop_page}}$
Семантична властивість будь-якого іншого типу	Властивість даних	Взаємо-однозначне $P_{\text{sem_prop}} \rightarrow \{p_i\}$	Взаємо-однозначне $\{p_i\} \rightarrow P_{\text{sem_prop}}$
Шаблон	Клас	Взаємо-однозначне $P_{\text{template}} \rightarrow X_{\text{cl}}$	Багатозначне $X_{\text{cl}} \rightarrow P_{\text{categ}} \cup P_{\text{template}}$

Для того, щоб ефективно використовувати семантичні властивості Semantic MediaWiki та інтегрувати такі Wiki-ресурси з іншими інтелектуальними системами, доцільно вміти створювати та обробляти онтології у редакторі Protégé. Тому розглянемо базові можливості цього редактору.

9. Використання Wiki-ресурсів в системах семантичного пошуку

У даному розділі розглядається, як Wiki-ресурси можуть використовуватися в якості джерела знань для семантичному пошуку у Web-середовищі [58]. На основі таких Wiki-ресурсів можна створювати онтології ПрО, що цікавить користувача, та тезауруси, що моделюють сферу його поточних інформаційних потреб [59, 60].

На даний час світовим співтовариством вже усвідомлений головний напрямок у боротьбі з інформаційним вибухом – перехід від збереження й обробки даних до накопичення й обробки знань. Для розробки інтелектуальних інформаційних систем (ІС), призначених для роботи зі знаннями, раніше зазвичай використовувалися логічні мови (Лісп, Пролог тощо), але в останні десятиріччя поширення Web – всесвітнього сховища, що відкриває доступ до величезної кількості різних інформаційних ресурсів (ІР), незалежно від їх географічної і національної локалізації та засобів подання – викликало потребу у нових, більш ефективних підходах до створення ІС, орієнтованих на функціонування у відкритому інформаційному середовищі та методів керування та обробки розподіленими знаннями в них.

9.1. Основні напрямки розвитку інформаційного пошуку

Однією з важливих проблем, пов'язаних із розробкою таких сучасних ІС, є створення засобів отримання актуальної та пертинентної інформації з середовища Web: тільки за наявності таких засобів доцільно вдосконалювати методи переробки отриманих відомостей. Таким чином, проблема пошуку в Web стає все більш актуальною зі зростанням обсягу Web та ускладненням його структури і тих програмних засобів, що діють у ньому. Значною мірою вирішенню цієї проблеми сприяє поступове перетворення Web на глобальну базу знань та розвиток технологій, що підтримують цей процес, але залишається ще дуже багато відкритих питань, що потребують створення ефективних рішень хоча б для окремих випадків цих проблем.

Інформаційний пошук у Web, що є складовою найрізноманітніших ІС, – це співставленні інформаційних потреб користувача (що формалізуються певним чином через пошукові запити) із відомостями щодо ІР та їх метаописів. Її рішення потребує як створення засобів для представлення відомостей щодо користувачів та ІР із достатньою виразною потужністю, так і у розробці ефективних методів їх співставлення. Термін "information retrieval") був введений американським математиком К.Н.Муерсом (Calvin Northrup Mooers), який визначив інформаційну потребу як причину для здійснення інформаційного пошуку. Як об'єкти інформаційного пошуку він розглядав документи та фактографічну інформацію.

Інформаційно-пошукові системи (ІПС) – це спеціалізовані програмні засоби, основне призначення яких полягає у здійсненні інформаційного пошуку (у Web, локальній або корпоративній мережі, на окремому

комп'ютері тощо). ПС можуть шукати як довільні ІР, так і спеціалізуватися на пошуку окремих підтипів ІО (приміром, Web-сервісів або фільмів). ПС може взаємодіяти безпосередньо із користувачем або входити до складу іншої ПС.

Через надзвичайно великий обсяг, гетерогенність та складну структуру Web тієї інформації, яку надає сам користувач – як набір ключових слів або природною мовою – для ідентифікації своїх інформаційних потреб, недостатньо для того, щоб знайти пертинентні відомості. Тому сучасні розвиваються у напрямку їх інтелектуалізації та персоніфікації.

Персоніфікація ПС – використання інформації щодо користувача, його інформаційних потреб. Така інформація може бути безпосередньо надана користувача, здобута з досвіду його взаємодії із системою або отримана від зовнішніх ІР та застосунків (приміром, соціальних мереж або персональних сторінок).

Інтелектуалізація ПС – це використання в процесі пошуку знань щодо його суб'єктів та об'єктів та отримання більш пертинентних результатів.

В ПС все частіше застосовують зовнішні бази знань (БЗ), в ролі яких часто використовують онтології різного рівня. Онтології дозволяють характеризувати:

- основні елементи ПС та зв'язки між ними;
- користувачів та сферу їх інтересів;
- ІР та їх ПрО;
- інформаційні об'єкти, що цікавлять користувача;
- інші ПС, які виступають в якості джерел інформації.

В ПС під ІО будемо розуміти ту інформацію, яку користувач отримує в результаті виконання процесу пошуку. Важливо, що сам об'єкт ПрО може бути як матеріальним (наприклад, людина, організація, місце), так і віртуальним, що не має конкретного матеріального еквівалента (наприклад, книга, програма), хоча і пов'язаний з тим чи іншим матеріальним носієм інформації. Приклади ІО, які можуть знаходити спеціалізовані ПС, – це Web-сервіс, мультимедійний об'єкт, товар в електронному магазині або онтологія.

Онтологічний опис ІО характеризує його структуру, відношення з іншими ІО, атрибути, їх типи та значення тощо. Тип ІО може бути формалізований як клас відповідної онтології, а потрібний користувачеві ІО – як екземпляр цього класу з певними властивостями. Доцільно використовувати для семантичного пошуку зовнішні спеціалізовані онтології та таксономії ІО. Приміром, якщо потрібно знайти Web-сервіс, то можна використовувати OWL-S, а для пошуку мультимедіа – таксономію мультимедійних ІО.

Інформаційний пошук у Web, що є складовою найрізноманітніших ПС, – це співставленні інформаційних потреб користувача (що формалізуються певним чином через пошукові запити) із відомостями щодо ІР та їх метаописів. Через надзвичайно великий обсяг, гетерогенність та складну структуру Web для ідентифікації інформаційних потреб користувача тієї

інформації, яку надає він сам, недостатньо. Тому сучасні інформаційно-пошукові системи (ІПС) розвиваються у напрямку їх семантизації, тобто використання знань для підвищення пертинентності пошуку.

9.2. Семантичний пошук у Web на основі онтологій

Різні підходи до пошуку інформації в Web, включаючи частково структуровані бази даних, машинне навчання та обробку природної мови, застосовувалися до проблеми аналізу та розуміння Web-сторінок у форматі HTML та XML. Проте відсутність семантики в цих засобах представлення інформації і в інструментах їх обробки значно обмежували якість методів. Тому виникає потреба у представленні знань, що пов'язані з тематикою пошуку.

Семантичний пошук – це метод інформаційного пошуку, у якому релевантність документа до запиту визначається семантично (за близькістю змісту), а не синтаксично (приміром, за частотою використання ключових слів у документі). Його метою є отримання з IP відомостей, що задовольняють інформаційну потребу користувача, пов'язану із розв'язанням певної проблем, в процесі якого застосовуються (наочно або приховано від користувача) знання щодо різних суб'єктів і об'єктів пошукової процедури

Системи семантичного пошуку (ССП) все частіше застосовують зовнішні бази знань (БЗ), в ролі яких часто використовують онтології різного рівня. Якщо система представлення знань визначає, як представляти поняття, то онтологія визначає, які саме поняття визначені в такій системі і як саме вони взаємопов'язані. такі формалізовані визначення дозволяють автоматизувати набагато глибші міркування, але такі визначення набагато важче побудувати. Онтології дозволяють формально охарактеризувати семантику:

- основних складових ССП та зв'язків між ними;
- області інтересів користувачів ;
- інформаційних ресурсів, серед яких здійснюється пошук;
- інформаційних об'єктів, що цікавлять користувача, тощо.

У семантичному пошуку в Web вирізняють підходи, що базується на структурованих мовах запитів; та підходи, орієнтовані на некваліфікованих користувачів, які не потребують знайомства зі спеціалізованими мовами запитів. У свою чергу, у другій групі можна виділити: – підходи, де запити складаються зі списків ключових слів; – підходи, де користувачі висловлюють запити природною мовою.

Сучасні ССП широко застосовують результати, отримані в проєкті Semantic Web, а саме – мову опису метаданих про ресурси (RDF), представлення онтологій (OWL) та запитів до них (SPARQL), а також інструментальні засоби для роботи з ними. Одна з першій систем створення семантичних запитів в Web – *SHOE* [9] – дозволяє користувачам задавати структуровані запити з використанням онтологій. Більшість ССП, що використовують онтології, орієнтовані на пошук лише у Semantic Web

(приміром, *Swoogle* [8]), тобто серед семантично анотованих IP. Але, на жаль, зараз такі IP складають лише незначну частину Web, і тому такий пошук не задовольняє потребам користувачів.

Для більшості користувачів досить складно використовувати структуровані мови запитів, і вони надають перевагу другій групі пошукових запитів. Але такі системи теж потребують застосування знань.

Приміром, *SemSearch* забезпечує подібний до Google інтерфейс запитів, що не потребує від користувачів мати знання щодо онтологій або мов. Запити користувачів складаються з двох або більше ключових слів, семантичне значення яких враховується для переформулювання запиту відповідно до синтаксису формальної мови запитів. Ключовим словам присвоюються семантичні значення шляхом співставлення їх з набором класів, властивостей і екземплярів в репозиторії семантичних даних. Кожному ключовому слову може відповідати клас, властивість або екземпляр. Розглядаються різні комбінації семантичних співставлень ключових слів. Наприклад, кожному ключовому слову може відповідати клас, або першому ключовому слову відповідає клас, а другому – властивість, і так далі. Всі комбінації співставлень враховуються в процесі переформулювання запиту, а за кожною комбінацією створюється вдосконалений формальний запит, який отримується із заздалегідь визначеного набору шаблонів запитів. Після переформулювання формальні запити точно оцінюються, і це забезпечує результати, які семантично пов'язані з усіма ключовими словами із запиту користувача.

Falcons відноситься до ПС на основі ключових слів для Semantic Web. Ця система підтримує пошук концептів та об'єктів. Пошук концептів здійснюється шляхом пошуку класів і властивостей в онтології, обраній користувачем, які відповідають умовам запиту.

Пошук в системі *QUICK* враховує онтологію відповідної ПрО. Коли користувач формулює запит з ключового слова, *QUICK* починає роботу з того, перетворює його в кілька семантичних запитів, кожен з яких отримується шляхом присвоєння якогось онтологічного поняття (властивість, об'єкт тощо з обраної онтології) для кожного з ключових слів. Потім користувач має обрати серед семантичних запитів, згенерованих системою, найбільш відповідний його інтересам. Якщо жоден семантичний запит, згенерований *QUICK*, не подобається користувачеві, то користувач самостійно може вести систему до генерації відповідного запиту, забезпечуючи додаткові характеристики (наприклад, вказуючи, що певне ключове слово має бути використане як властивість або як об'єкт).

Можна навести ще багато прикладів використання онтологій у пошукових системах. Можна виділити зовнішні та внутрішні онтології, що використовуються в них. Внутрішні онтології описують різні елементи ССП (обов'язкові, як користувачі, IP, IO та запити, та службові, що залежать від специфіки конкретної системи – приміром, тезауруси або лексичні онтології) та відношення між ними. Структура таких онтологій задається розробниками ССП, а екземпляри її класів створюються в процесі її функціонування та

можуть змінюватися. Зовнішні онтології описують ті ПрО, що цікавлять користувачів, або ж характеризують типи ІО, які вони прагнуть знайти. Такі зовнішні онтології можуть формуватися самими користувачами ССП, але зазвичай їх прагнуть знайти у Web або імпортувати із зовнішніх інформаційних ресурсів.

Незважаючи на різноманіття підходів до семантичного пошуку в Web, дослідження в цій області ще знаходяться на самому початку. Ще не знайдено методів для автоматичної трансформації природномовних запитів у формальні на основі онтологій, для семантичного анотування Web-контенту та автоматичного здобуття знань щодо певних ІО з обраних ІР. Крім того, потребують розробки засоби пошуку, створення та підтримка онтологій, потрібних для пошуку.

Семантична *модифікація запитів* користувача – це добре відомий метод пошуку інформації. В області семантичного пошуку часто використовує інформацію з онтологій. Він грає центральну роль у багатьох семантичних пошукових системах. Різні методи були розроблені для того, щоб збільшити повноту і точність пошуку. Наявність онтологій дозволяє відносно легко підвищити повноту запиту: онтологія стає для пошукової машини джерелом більш загальних, порівняно з ключовими словами запиту, термінів, – їх надкласів в онтології. Дещо важче поліпшити точність запиту. Це потребує вирішення проблеми омонімії та вибору певного підкласу для кожного з використаних у запиті термінів.

Одна з суттєвих проблем семантичного пошуку – це знаходження онтологій, яка найбільш адекватно відображає сферу інтересів користувача [55]. Якщо використовувати більш широкую онтологію, то це не тільки значно збільшує час обробки, але й ускладнює роботу самого користувача з відбору важливих для нього понять та відношень. З іншого боку, створювати онтології самостійно для більшості користувачів надто складно. Тому виникає потреба в засобах автоматизованої генерації онтологій за тими ІР, які користувач вважає релевантними цікавлячій його ПрО.

Чим точніше онтологічні моделі характеризують потреби користувача, тим більш ефективно виконується семантичний пошук. Але побудова персональних онтологій – надто складна задача для більшості користувачів, і це значно зменшує сферу застосування ССП.

Тому наявність різноманітних семантичних Wiki-ресурсів та засобів побудови за ними онтологій надзвичайно важливі для семантичного пошуку. Користувачам досить просто знайти відповідний Wiki-ресурс, обрати на ньому сторінки, що стосуються їх поточних задач, та згенерувати за знаннями, що містяться у семантичній розмітці цих сторінок, відповідну онтологію. Користувач може зберегти цю онтологію та потім використовувати її у своїх запитах, якщо ССП припускає явне підключення обраної користувачем онтології.

9.3. Використання тезаурусів у семантичному пошуку

Незважаючи на наявність великої кількості ІПС із розширеними можливостями для побудови запитів, більшість користувачів надає перевагу простим запитам з ключових слів, хоча прагнуть отримувати персоналізовані результати пошуку, у той час як пошукові запити на основі онтологій, які дозволяють значно краще формалізувати потреби користувачів, базуються на досить складних структурованих мовах. Тому виникає потреба у розробці певних проміжних засобів представлення знань, які в достатній мірі враховували семантику, але були б зрозумілі широкому колу користувачів. Доцільно використовувати для цього тезауруси [45, 51] та пов'язані з ними лексичні онтології [48].

Тезаурус IP є окремим випадком онтології, множина понять якого є підмножиною з тих понять онтології ПрО, які семантично пов'язані з певними елементами контенту цього IP, а множини відношень та аксіом є порожніми. Тезаурус можна розглядати як онтологічний опис IP, який характеризує його семантику. З точки зору користувача, такий тезаурус – це множина слів (можливо, з певними вагами).

Для природномовних IP тезаурус може будуватися за допомогою лексичних онтологій, для мультимедійних – за їх семантичним метаописом (приміром, в RDF). Для інших типів IP та IO потрібні спеціалізовані методи для аналізу їх семантики. Приміром, для Web-сервісів це можуть бути методи аналізу їх моделей.

Тезаурус задачі – це теж окремий випадок онтології, множина понять якого є підмножиною з понять онтології ПрО, які відповідають поточній задачі користувача (приміром, її природномовному опису).

Користувач може побудувати тезаурус задачі самостійно, обравши їх з множини понять онтології безпосередньо або за певним критерієм (приміром, всі підкласи певного класу) або знайти їх за природномовним описом задачі, використовуючи лексичну онтологію. Крім того, тезаурус може бути побудовано як об'єднання кількох раніше побудованих тезаурусів (при цьому можуть використовуватися тезауруси, побудовані за різними онтологіями).

Тезаурус не містить відношень між термінами, але ці відношення, присутні в базовій для нього онтології ПрО, використовуються для побудови тезаурусу. Слід відмітити, що тезауруси тієї самої задачі, побудовані за різними онтологіями ПрО, можуть значно різнитися.

Важливо, що тезаурусну модель IP досить просто зрозуміти навіть некваліфікованим користувачам, які можуть проглядати та модифікувати такі набори термінів.

У семантичному пошуку тезаурус дозволяє відфільтрувати ті IP, які семантично знаходяться ближче для інформаційних потреб користувача.

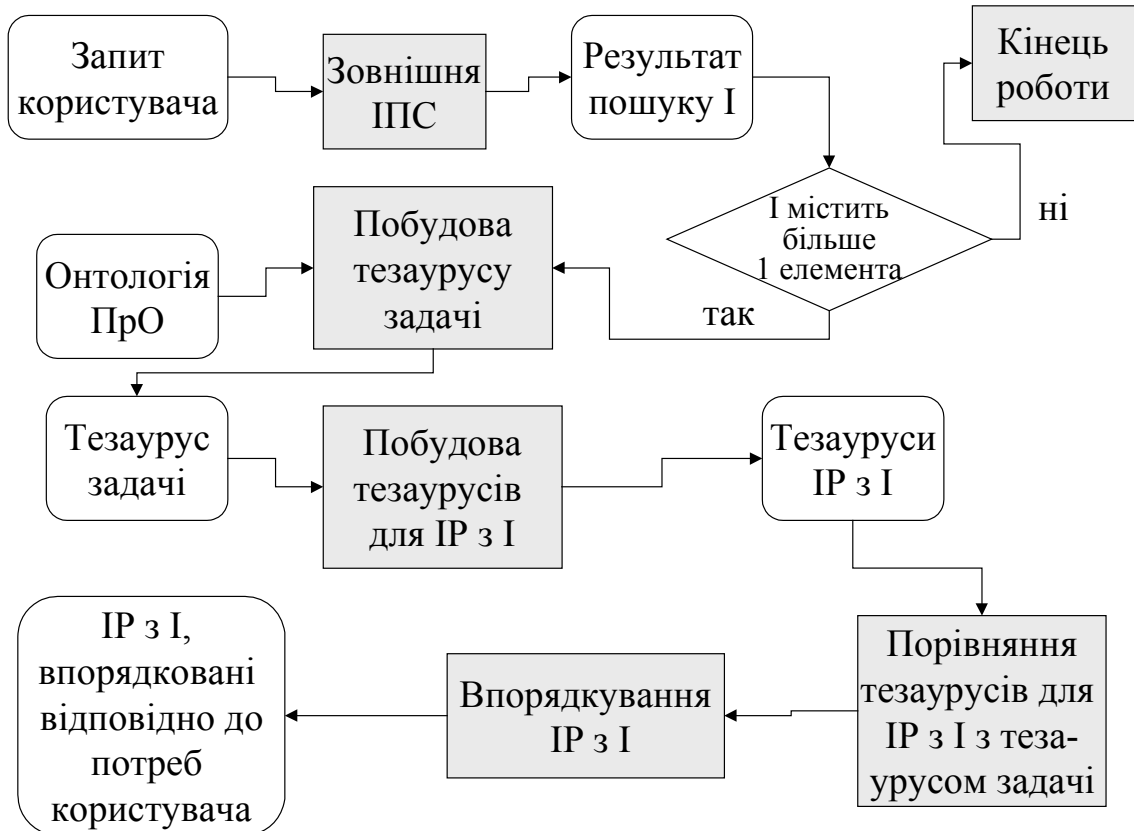


Рис.9.1. Алгоритм семантичного впорядкування результатів пошуку на основі тезаурусів

Одним з джерел для побудови тезаурусів є Wiki-ресурси. Користувач може обрати Wiki-сторінку або множину Wiki-сторінок та надалі використовувати в якості тезаурусу поняття та категорії з цих сторінок.

9.4. Wiki-ресурси як джерело інформації для побудови тезаурусів та онтологій в системі МАІПС

В деяких випадках у семантичному пошуку можна використовувати знання, здобуті не з семантичних, а зі звичайних Wiki-ресурсів. Для цього потрібно аналізувати систему категорій, що реалізовані в цьому Wiki-ресурсі, та імена статей, що вони групують.

Розглянемо це на прикладі системи МАІПС.

МАІПС [49, 50], що орієнтована на користувачів, які мають в мережі сталі інформаційні інтереси та потребують постійного надходження відповідної інформації. У розробці системи застосовувалися елементи Semantic Web.

МАІПС – це метапошукова система, що призначена для пошуку інформації у відносно вузьких ПрО, пов’язаних з професійними чи науковими інтересами користувачів. Основою МАІПС є технології Semantic Web, зокрема мова подання онтологій OWL, і засоби її обробки. Для подання знань, що цікавлять користувача, використовуються онтології ПрО та тезауруси задач, що базуються на них (рис.9.2). При цьому тезаурус будується користувачем за відповідною онтологією самостійно, а онтологія ПрО обирається з набору онтологій, запропонованих на сайті розробниками

системи. МАПС надає можливість зберігати й повторно виконувати запити, з огляду на реакцію користувача на раніше запропоновані йому IP (персональна фільтрація), відстежувати появу аналогічних запитів в інших користувачів (колаборативна фільтрація), зберігати формальний опис сфери інтересів користувача у вигляді онтології (семантична фільтрація) тощо.



Рис.9.2. Користувацький інтерфейс МАПС

МАПС дозволяє покращити результати пошуку, отримані від зовнішньої ІПС, завдяки:

- орієнтованості системи на користувачів, які мають в мережі сталі інформаційні інтереси та потребують постійного надходження відповідної інформації з обраної ПроО;
- застосуванню програмних агентів користувацького інтерфейсу, які здатні діяти в динамічному середовищі, враховуючи персональні особливості користувача;
- використанню зовнішніх онтологічних баз знань, які містять відомості щодо ПроО пошуку та структурі ІО, які шукає користувач.

Слід відмітити, що в МАПС агентний підхід використовується в першу чергу як парадигма опису поведінки складної системи, а не як технологічна платформа програмування. Поведінка інтелектуальних агентів описується через інтенціональні відношення, а їх функції – з використанням сервіс-орієнтованої архітектури.

Проведені експерименти дозволяють показати переваги семантичного пошуку в порівнянні з традиційними пошуковими машинами. Порівнювалися результати пошуку за тими самими запитами у Google та їх упорядкування у МАПС. Через те, що пошук здійснювався на одній індексній базі (Google), повнота та абсолютна точність пошуку співпадають, але значно відрізняється

Використання Wiki-ресурсів в системах семантичного пошуку

умовна точність пошуку для 10, 20 та 30 документів: МАПС автоматизує ту роботу, яку зазвичай виконує користувач, проглядаючи сторінки з результатами пошуку.



Рис.9.3. Архітектура МАПС

Найпростіший для користувача спосіб використання Wiki-ресурсів у МАПС – це використання в якості тезаурусу задачі множини категорій обраної Wiki-сторінки (або набору таких сторінок). Відповідно до власних потреб користувач може визначити вагу кожної з цих категорій у пошуковій процедурі.

10. Приклади використання Wiki-ресурсів у практичних застосунках

10.1. Використання Semantic MediaWiki для побудови семантичного порталу

Wiki-системи звичайно надають можливість вільного анотування контенту семантичними анотаціями, однак існують засоби, що дозволяють обмежити використовуваний при анотуванні словник термінів. Це реалізується за допомогою введення, заснованого на формах. Важливо дотримуватися балансу між використанням неструктурованого вмісту (це краще, ніж повна його відсутність) і роботою зі структурованих даними.

Одне з поширених застосувань Semantic MediaWiki і Wiki-систем у цілому – це організація спільного керування знаннями, приміром, порталів знань.

На відміну від традиційних Wiki-систем, Semantic MediaWiki дозволяє семантично анотувати вміст сайту. Ця вільна і незалежна модель анотування є достатньо гнучкою і легко розширюваною. Крім того, вона не потребує знань щодо визначеної заздалегідь схеми даних – така схема може бути додана у Wiki пізніше. Однак при створенні убудованих запитів необхідно використовувати точні назви семантичних властивостей.

Використання в Semantic MediaWiki шаблонів і форм введення дозволяють задавати обмеження на використовувану різних видів анотацій. Шаблони задають логіку і зовнішній вигляд частини сторінки. У середині шаблону використовуються перемінні, значення яких підставляються в процесі виклику шаблону на сторінці. Якщо в тексті шаблону зустрічаються семантичні властивості, то анотуються всі сторінки, що використовують цей шаблон. Отже, при зміні значень властивостей у шаблоні каскадно оновлюються всі залежні від нього сторінки. Це забезпечує гнучкий засіб для модифікації структурованих даних у Wiki. Форми введення надають графічний інтерфейс для заповнення шаблонів коректними значеннями, а їхнє використання не потребує навіть знання Wiki-розмітки. Таким чином, сполучення можливостей шаблонів і форм дозволяє мати множину визначених заздалегідь видів анотацій.

При розробці порталу доцільно створити шаблони для основних, часто повторюваних видів ресурсів, приміром, “особа”, “організація”, “діяльність”, “місце”, “дата”. Форми можуть містити різні типи областей введення: для тексту, дат, вибір зі списку тощо. Кожному полю відповідає анотація, тобто семантична властивість. Після введення значення в поле це значення привласнюється відповідному властивості. Можливо також забезпечити імпорт цих властивостей із вже існуючих словників, наприклад, використовувати FOAF для опису осіб. Для того, щоб залишити можливість для вільного анотування, у формах використовуються поля введення, у яких можна розміщати анотований текст. За допомогою цього можна забезпечити

баланс між керованим уведенням з визначеними заздалегідь семантичними властивостями і можливістю розмічати текст вільно.

Певні проблеми при створенні порталу можуть бути викликані тим, що у Wiki будь-який користувач зазвичай має можливість редагувати контент. Тому необхідно використовували систему розмежування доступу, убудовану в Media Wiki. Така система дозволяє створити чотири групи користувачів:

1. *Анонімні відвідувачі* порталу мають право на читання звичайних Wiki-сторінок, тобто статей, що знаходяться в основному просторі імен.
2. *Зареєстровані на порталі користувачі* одержують можливість читати сторінки в інших просторах імен. Вони також мають права на редагування Wiki-сторінок, за винятком тих з них, що були чи формами шаблонами.
3. *Адміністратори Wiki* мають також право на редагування форм і шаблонів.
4. *Бюрократи* мають ті ж права, що й адміністратори, але на додаток до цього вони мають право давати адміністраторські права іншим користувачам та позбавляти їх цих прав.

Необхідність реєстрації на Wiki-порталі свого облікового запису може здатися користувачам незручною, тому можливо застосовувати різні додаткові розширення Media Wiki, призначені для аутентифікації. Приміром, можуть використовуватися розширення для аутентифікації за допомогою полегшеного протокол доступу до каталогів LDAP (Lightweight Directory Access Protocol). З'єднання між LDAP-сервером і порталом здійснювалося за зашифрованим протоколом SSL. Це дозволяло використовувати вже існуючі облікові записи користувачів для аутентифікації на порталі.

Роботи з розробки порталу можна умовно розбити на наступні області [44]:

- налаштування системи,
- візуальне оформлення і кастомізація,
- імпорт даних,
- розробка шаблонів, що включає в себе моделювання структурованих даних, тобто властивостей і класів.

Найбільші труднощі під час розробки Wiki-порталу пов'язані за створенням шаблонів і їхнім візуальним оформленням відповідно до вимог порталу. Для багатьох Wiki-ресурсів невеликі помилки в оформленні цілком припустимі, однак для офіційних ресурсів, приміром, державних енциклопедій або порталів організацій пред'являються набагато більші вимоги. Приміром, у шаблонах усі змінні, що залишилися незаповненими, повинні бути сховані. Крім того, у зв'язку з високими вимогами до візуального оформлення, деякі анотації також можуть містити Wiki-розмітку (наприклад, тип чи розмір шрифту). Більш того, у зв'язку з тим, що шаблони є одночасно й інструментом візуального оформлення Wiki-статті і містять логіку відображення, маніпуляції із шаблонами потребують високого рівня

володіння Wiki-розміткою. У зв'язку з цим, маніпуляції із шаблонами дозволяються тільки певним групам користувачів порталу.

Найбільшою перевагою SMW, крім гнучкої парадигми анотацій, є можливість повторного використання даних усередині платформи. Це досягається шляхом використання запитів, що збирають інформацію з Wiki-сторінок. Ці убудовані запити дозволяють витягати множини даних (приміром, списки або таблиці) чи окремі значення семантичних властивостей, надаючи можливість їхнього візуального представлення за допомогою таблиць, списків, карт і т.д. Таке повторне використання даних дозволяє уникнути їхнього дублювання. Приміром, інформація про особу (її ім'я, адресу електронної пошти чи номер телефону) вводиться тільки один раз. Якщо ця інформація потрібно на інших Wiki-сторінках, що посилаються на цю особу, то відповідні дані автоматично запитуються і виводяться в необхідному форматі. При зміні даних на сторінці-джерелі відбувається відповідне їм відновлення на всіх сторінках, що запитують ці дані. Фактично, убудовані запити дозволяють створювати динамічні сторінки. і

10.2. Визначення компетентності спеціалістів

Сучасний етап розвитку ІТ характеризується великою кількістю міждисциплінарних досліджень, які поєднують результати з різних сфер знань, та швидким розвитком нових напрямків традиційних наукових напрямків. В різних сферах наукової і технічної діяльності пропонується багато різноманітних проектів – наукових досліджень, видання наукових праць, проведення міждисциплінарних досліджень, проведення наукових досліджень окремими вченими тощо. Це спричиняє деякі проблеми у класифікації та визначенні області експертизи як окремих осіб, так і різноманітних проектів та розробок.

Подібні проблеми призводять до того, що важко встановити та порівняти компетентність спеціалістів відносно певної роботи – технічної розробки, наукового проекту, публікації в реферованому журналі, відгуку на надані матеріали тощо. При цьому виникає проблема *оцінки компетентності* спеціалістів в конкретній Про [27]. Рішення цієї проблеми потребує співставлення природномовних (ПМ) документів, які характеризують кваліфікацію та досвід спеціалістів, із природномовним описом задачі, яку потрібно вирішити. Якщо ж замість ПМ співставляти документи з семантизованих Wiki-ресурсів, то час обробки і якість співставлення значно зростають. Тому доцільно для такого класу задач орієнтуватися на використання Wiki-сторінок, пов'язаних як з окремими особами та організаціями, так і з термінами та базовими поняттями обраної сфери знань. Це викликає потребу у розвитку як різноманітних локальних та спеціалізованих Wiki-ресурсів, так і у створенні енциклопедій та довідників на основі Wiki технологій із гарантованою якістю контенту (приміром, національних енциклопедій та стандартів).

10.3. Забезпечення прозорості рамок кваліфікацій

Система кваліфікацій України базується на Національній рамці кваліфікацій (НРК) – системному і структурованому за компетентностями описі кваліфікаційних рівнів. Вона необхідна для зіставлення та порівняння документів про освіту, міжнародного визнання кваліфікацій, які були здобуті в Україні [49]. Основні положення НРК повинні узгоджуватися з Європейською рамкою кваліфікацій (ЄРК), але механізм встановлення семантичного співвідношення між рівнями НРК та ЄРК і порівняння результатів навчання ще не розроблений.

Основою для цього може бути створення онтологічної моделі кваліфікацій, яка надалі буде використовуватися для семантичної розмітки довільних Wiki-ресурсів, пов'язаних з освітою. Саме Wiki-технологія дозволяє семантично поєднати природномовні тексти різного походження, які стосуються даної проблематики.

Еталонна модель рамки кваліфікацій (ЕМРК) [25] забезпечує порівнянні компетентностей, пов'язаних з різними інформаційними об'єктами [28], шляхом оцінки їх семантичної близькості. Базові класи ЕМРК – це ІО, які описують результати навчання, осіб, які отримують їх, та ті установи, хто їх забезпечують. ІО визначаються через свої властивості. Вони мають підкласи, в яких до властивостей базових класів додаються специфічні властивості, які визначають семантику цих підкласів та їх відношення з іншими ІО. Приміром, всі екземпляри класу «Особа» мають такі властивості даних, як «Ім'я» та «Рік народження», а екземпляри класу «Студент», що є підкласом класу «Особа», має також об'єктні властивості «Місце навчання» класу «Освітня організація», «Вивчає» класу «Спеціальність» та властивість даних «Рік навчання». Вісім рівнів кваліфікацій ЄРК можна представити як підкласи класу «Кваліфікація» з властивостями даних «Рівень кваліфікації» від 1 до 8 та значеннями властивості даних «Система кваліфікацій», що дорівнює текстовій константі «ЄРК».

Використання ЕМРК дозволяє автоматизувати обробку інформації, що стосується результатів навчання, інтегруючи знання, що відповідають потребам користувача [25]. Побудувавши таку онтологію, можна виконувати SPARQL-запити до неї для отримання потрібних знань. Але виникають проблеми, пов'язані з поповненням онтології відомостями щодо нових екземплярів ІО: для цього потрібні специфічні навички з онтологічного аналізу. Тому виникає потреба у використанні технологічної платформи, орієнтованої на користувачів, що не спеціалізуються в інженерії знань. Такою платформою є Wiki-середовище, яке дозволяє користувачам самостійно доповнювати та редагувати контент сторінок.

Онтологія ЕМРК описує семантичні властивості і відношення між ІО, що стосуються результатів навчання. Користувач семантично розмічає контент Wiki-сторінок елементами цієї онтології: визначає категорію (чи набір категорій) будь-якого документу й ідентифікує окремі елементи його контенту. Використання онтологічної моделі рамки кваліфікацій забезпечує автоматизувати обробку результатів навчання й інтегруючи знання, необхідні

користувачам. Зокрема, наявність онтології дозволяє користувачам представити структуру категорій і семантичних властивостей, який можна використовувати в запитах.

Візуалізація цієї онтології в Protégé, зроблена за допомогою плагіна OntoGraf, який завдяки різним кольорам дає змогу відтворити різні типи онтологічних відношень і візуалізувати екземпляри класів, представлена на рис.10.1.

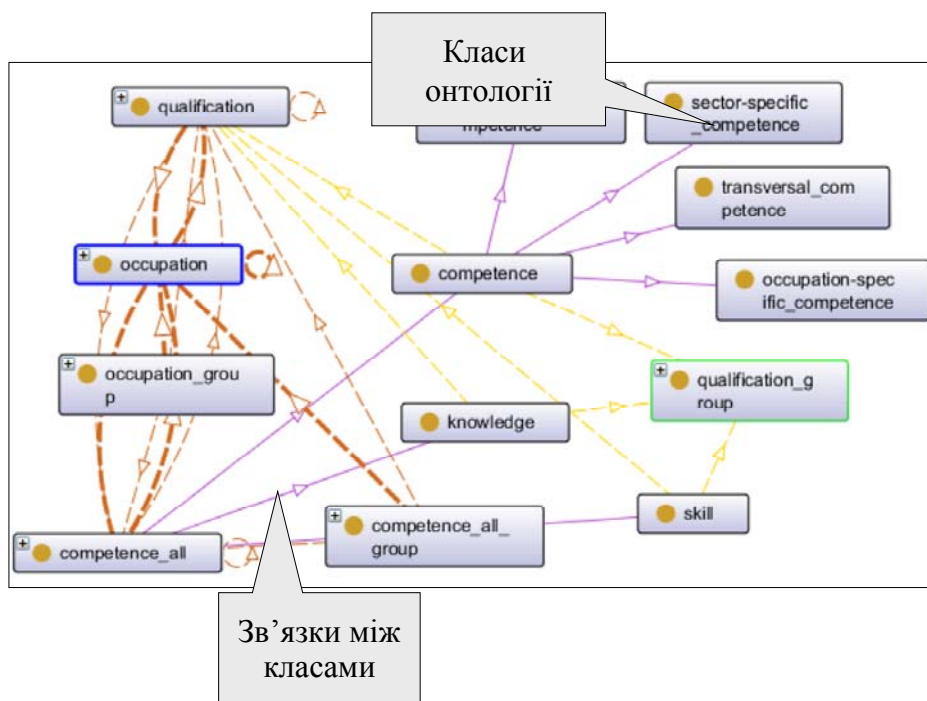


Рис.10.1. Онтологія ЕМРК

При цьому важливо, щоб онтологічна модель відповідала актуальному уявленню Wiki-ресурсу – інакше запити не будуть пертинентні потребам користувачів (наприклад, якщо семантична властивість була перейменована, а в запиті використовується старе ім'я, те значна частина інформації не буде виявлена). Така ситуація може виникнути й у тому випадку, якщо розроблювачі Wiki-ресурсу і його користувачів застосовують різні версії онтологічної моделі. Тому виникає необхідність зіставлення версій онтологій.

ІО можуть мати об'єктні властивості класу «атомарна компетентність». Важливо, що для одного ІО можуть існувати кілька різних об'єктних властивостей класу «атомарна компетентність». Приміром, ІО «спеціальність» має об'єктні властивості «обов'язкові навички» та «додаткові навички» класу «атомарна компетентність».

Атомарні компетентності призначаються для зіставлення екземплярів різних класів ІО у цій моделі. Одна й та сама атомарна компетентність може використовуватися різними екземплярами різних класів. З точки зору забезпечення прозорості рамок кваліфікацій важливо, що базові класи ЕМРК мають об'єктні властивості зі значеннями з класу «Атомарна компетентність», що дозволяє аналізувати та порівнювати ці ІО на рівні

знань. Важливою перевагою запропонованої моделі є те, що при такому описі явно вказується семантика інформаційної потреби. Унаслідок цього можна здійснювати пошук, що диференціює різні відносини між шуканим ІО і набором компетентностей. Наприклад, з тим самим екземпляром класу «особа» деякі різні підмножини атомарних компетентностей можуть бути зв'язані відносинами «володіє», «має сертифікат», «може викладати», «має досвід використання». Така диференціація дозволяє значно більш точно задовольняти інформаційну потребу користувача, виявляючи саме ті ІО, що відповідають його вимогам. Проблеми, що виникають у процесі рішення такої задачі, пов'язані з формуванням множини атомарних компетентностей та з поповненням бази знань про екземпляри ІО.

Класи онтології ЕМРК відповідають категоріям Wiki-ресурсу (рис.10.2).

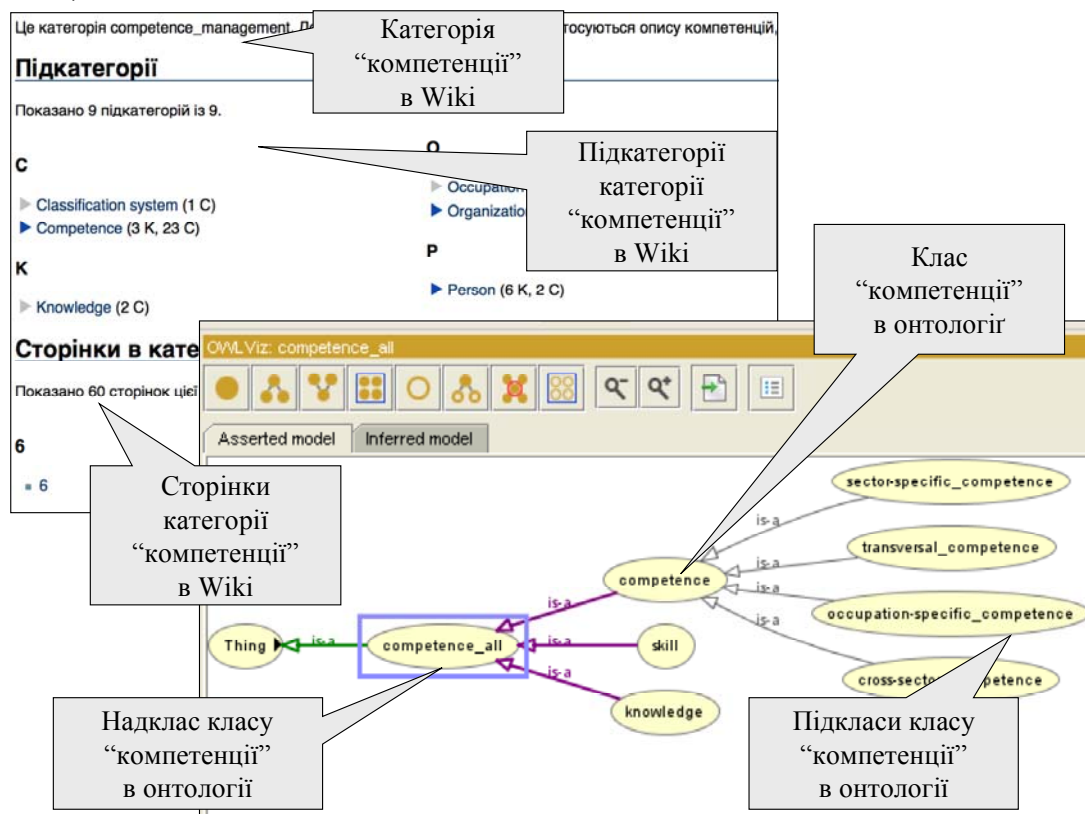


Рис.10.2. Відповідність між класом онтології і категорією Wiki «Компетенція»

Wiki значно розширює сферу використання такої системи порівняно з традиційними базами даних та системами управління контентом. Технологія Wiki забезпечує актуальність отриманих результатів – контент сторінок автоматично поновлюється після виникнення змін на сторінках, які є для них інформаційними джерелами. В процесі застосування онтологічної моделі компетентностей для семантичної розмітки на основі Semantic MediaWiki користувач семантично розмічає IP елементами цієї онтології. Користувач може визначити категорію (або набір категорій) кожного документу та ідентифікувати окремі елементи його контенту. Приміром, можна визначити

певну сторінку як опис спеціальності та вказати її рівень, назву, основні компетентності тощо. Це дозволяє виконувати складні семантичні запити – приміром, знайти всі навчальні заклади, що знаходяться в певному місті та де викладаються дисципліни, що містять певний визначений користувачем набір компетентностей.

Елементи семантичної розмітки сторінок в Semantic MediaWiki, що базуються на описаній вище моделі даних та еталонній онтології компетентностей, є основою для здобуття інформації з цих сторінок. Саме структура цієї онтології визначає, які саме знання можна автоматизовано отримати з Wiki-сторінок.

Користувач може отримувати результати вже побудованих запитів щодо компетентностей, спеціальностей, рівнів рамок кваліфікації тощо, які інтегровані до Wiki-сторінок. Він може також створювати такі запити самостійно, освоївши відповідну формальну мову.

Приміром, результат виконання наступного запиту:

```

{{#ask:
[[Категорія:Competence_management]] |
[[Категорія:Competence]] |
?competence_name = назва компетенції |
?speciality_number = номер спеціальності |
?qualification_frame_type = за класифікаційною рамкою }}
    
```

дозволяє знайти ті компетенції, що представлені в даному інформаційному ресурсі, та надати користувачеві їх назви, номери та класифікаційні рівні (рив.10.3).

пошук спеціальностей за компетенціями [ред.]			
	назва компетенції	номер спеціальності	за класифікаційною рамкою
Competence 30011	Володіння знаннями про видатних людей та технологій, найбільш видатні комп'ютерні пристрої	122 123 118	Ukraine
Competence 30012	Здатність розробляти програмні продукти для процесів, які комп'ютеризуються	122 124 118	Ukraine
Competence 30013	Володіння сучасними методами ефективного доступу до інформації, її збору, систематизації та збереження	122 124 118	Ukraine
Competence 30014	Здатність використовувати методи ідентифікації та класифікації інформації на базі нових інформаційних технологій	122 123 118	Ukraine
Competence 30015	30015	122 103 118	Ukraine
Speciality 122		122	Ukraine
Aspirantura IPS			
Compet1 test			
Competence 0			
Comp			
Comp			

Callouts in the image point to:

- Назви компетенцій**: points to the 'назва компетенції' column.
- Імена знайдених Wiki-сторінок**: points to the first column (ID).
- Номери компетенцій**: points to the 'номер спеціальності' column.
- Кваліфікаційні системи компетенцій**: points to the 'за класифікаційною рамкою' column.

Рис. 10.3. Результати виконання семантичного запиту з пошуку компетенцій у Wiki-ресурсі

Форму подання результатів та потрібну інформацію визначає автор запити, але користувач може керувати їх впорядкуванням на екрані.

Приміром, наступний запит дозволяє знайти, де саме знаходяться навчальні заклади:

```

{{#ask:
[[Категорія:Competence_management]]
[[Категорія:educational_organization]]|
?situated_in}}

```

Для користувачів ця інформація візуалізується як таблиця наступним чином (рис.10.4), причому користувач може сам обирати напрямок впорядкування результатів за значеннями кожної із семантичних властивостей:

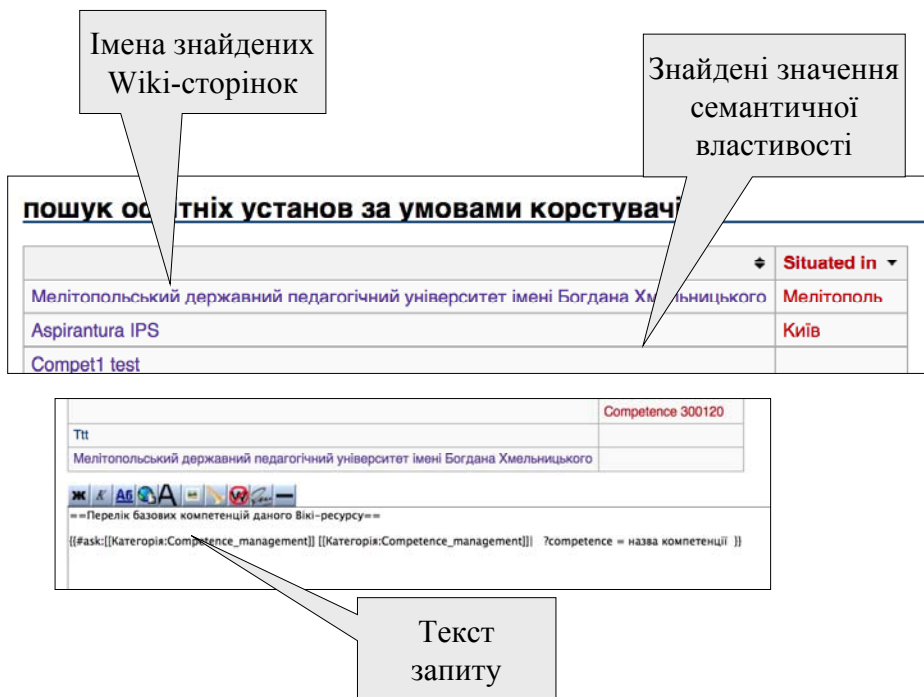


Рис. 10.4. Подання результатів запити до Wiki-ресурсу

Можна більш детально специфікувати тип Wiki-сторінок, інформація з яких задовольнить користувача, поєднати кілька різних умов тощо, як було описано вище. Якщо потрібно, цей запит можна специфікувати додатковими умовами:

```

{{#ask:
[[Категорія:Competence_management]] [[Категорія:Competence]]
[[Категорія:occupation_competence]]
[[Speciality_number::122]] |
?competence_name = назва компетенції |
?speciality_number = номер спеціальності |
?qualification_frame_type = за класифікаційною рамкою }}

```

Результати виконання такого запиту будуть містити тільки ті дані, що відповідають усім цим умовам (рис.10.5). Інформація на таких сторінках буде відновлюватися автоматично кожного разу після того, як буде змінюватися контент тих Wiki-сторінок, з яких запити здобувають інформацію. З результатів запитів можна за посиланнями перейти на ті сторінки, з яких отримано інформацію, та на сторінки властивостей та категорій, що використовуються у запиті. Таким чином, користувач завжди буде автоматично отримувати не тільки семантично оброблену, але й актуальну інформацію. Тексти запитів можна вставляти в інші Wiki-сторінки з відповідними поясненнями, поєднувати їх та імпортувати їх результати до інших інтелектуальних систем. Кожна сторінка може містити довільну кількість запитів та супроводжуватися потрібними відомостями.

Деталізований запит щодо компетенцій спеціальності

номер спеціальності	назва спеціальності	назва компетенції	номер компетенції	кваліфікаційна рамка
122	Інформаційні технології	Володіння знаннями комп'ютерних технологій	30011	Ukraine
123	Інформаційні технології	Здатність розробляти програмні продукти для процесів, які комп'ютеризують	30012	Ukraine
124	Інформаційні технології	Володіння спеціальними методами ефективного доступу до інформації, її збору, оброблення	30013	Ukraine
118	Інформаційні технології	Здатність використовувати методи ідентифікації та класифікації інформації на інформаційних техноло	30014	Ukraine
103	Інформаційні технології	30015	30015	Ukraine

Імена знайдених Wiki-сторінок

Назви компетенцій

Текст запиту

Номери компетенцій

Кваліфікаційні системи компетенцій

```

[[#ask:]]
[[Категорія:Competence_management]]
[[Категорія:Competence]]
[[Категорія:occupation_competence]]
[[Speciality_number=122]]
[[Competence_name=назва компетенції | ?speciality_number = номер спеціальності | ?qualification_frame_type = за класифікаційною рамкою ]]
    
```

Рис. 10.5. Подання результатів розширеного запиту до Wiki-ресурсів

Semantic MediaWiki дозволяє виконувати запити не тільки для знаходження ІО з певними властивостями, але й знаходити семантично близькі екземпляри ІО. Приміром, можна побудувати запит, який знаходить всі професії, в яких застосовуються компетентності певної спеціальності, та впорядковує їх відповідно до критерію (1) від найближчих до найдалших.

Додаток 1. Основні етапи побудови семантичного Wiki-ресурсу

Для того, щоб ефективно та якісно розробити семантичний Wiki-ресурс, використовуючи онтологічний аналіз та елементи менеджменту розподілених знань, доцільно виконувати наступні етапи:

- Виділити типові інформаційні об'єкти (ІО) верхнього рівню, які характеризують специфіку Wiki-ресурсу (Wiki-сторінки з подібною структурою та контентом);
- Побудувати ієрархію типових ІО верхнього рівню;
- Визначити семантичні властивості типових ІО верхнього рівню;
- Встановити семантичні відношення між різними типовими ІО;
- Виділити тематичні підрозділи Wiki-ресурсу;
- Для кожного тематичного підрозділу визначити специфічні для нього підтипи типових ІО вищого рівню;
- Визначити семантичні властивості тематичних підрозділів Wiki-ресурсу;
- Побудувати ієрархію типових ІО, специфічних для тематичних підрозділів Wiki-ресурсу;
- Визначити семантичні властивості типових ІО, специфічних для тематичних підрозділів Wiki-ресурсу;
- Встановити, якщо потрібно, семантичні відношення між цими типовими ІО;
- Побудувати онтологію, яка формалізує відношення між типовими ІО, тематичними підрозділами та їх семантичними властивостями;
- Проаналізувати побудовану онтологію та погодити її з усіма розробниками Wiki ресурсу;
- Внести до онтології потрібні уточнення;
- Використовувати уточнену онтологію для побудови спеціальних сторінок Wiki-ресурсу наступним чином: для класів онтології, згенерованих для типових ІО та тематичних підрозділів, побудувати категорії Wiki, а для об'єктних властивостей та властивостей даних онтології – семантичні властивості Wiki відповідних типів.
- Надалі в процесі створення нових Wiki-сторінок можливо вдосконалювати систему категорій Wiki ресурсу та вносити ці зміни до онтології, в якій звичайним сторінкам будуть відповідати екземпляри відповідних класів.

Додаток 2. Глосарій

- MediaWiki, 66
 - посилання на інформаційні ресурси, 78
 - створення заголовків, 76
- Open Source Intelligence, 6
- OWL, 40, 43, 142
 - 2.0, 42
 - діалекти, 42
 - екземпляри, 41
 - класи, 41
- Protégé, 43
 - візуалізація онтології, 55
 - властивості, 45
 - екземпляри класів, 45
 - класи, 45
- RDF, 40
- Semantic Forms, 110
- Semantic MediaWiki, 105, 109
 - запит, 115
 - семантичні властивості, 107
- Semantic Web, 34, 35, 36, 142
- SMW-QL*, 115
- Wiki
 - платформа, 10
 - реалізації, 13
 - ресурси, 6
 - сайт, 10
 - спільноти, 11
 - теоретичний базис, 16
 - технологія, 5
- Wikipedia, 10
- Wiki-двигун, 8
- Wiki-сторінка, 7
 - Формат, 8
- Wiki-технологія, 7
- XML, 40
- Велика українська енциклопедія
 - Електронна версія, 26, 27, 130, 131, 167
- Віківиди, 24
- Вікіджерела, 24
- Вікіновини, 24
- Вікіпедія, 10, 19
- Вікіпідручник, 24
- Вікісловник, 23
- Вікісховище, 25
- ВікіЦитати, 24
- Галерея*, 97
- Інформаційний пошук, 145
- Керування знаннями, 34
- МАПС, 151
 - архітектура, 153
- онтологія
 - формальна модель, 140
- Онтологія, 37, 62
 - Protégé, 45
 - в Protégé, 43
 - гуманітарний підхід, 38
 - комп'ютерний підхід, 39
 - математичний підхід, 39
 - предметної області, 38
 - формальна модель, 37
- Предметна область, 43
- Простір назв, 28
- Рекомендації, 63
- Семантичний пошук, 130, 147
 - семантичні елементи
 - семантичні елементи, 143
- Тезаурус
 - задачі, 150
 - IP, 150
- Тип IO*, 131
- Шаблон, 31, 84
 - параметри, 89

Додаток 3. Основні терміни і поняття

MediaWiki – вільне (програмне забезпечення, на основі якого можуть функціонувати Wiki-ресурси. Воно дає змогу, використовуючи Wiki-розмітку, створювати Wiki-сторінки та посилання між ними, а також шукати і проглядати ці сторінки. MediaWiki використовує мову програмування PHP для обробки і відображення інформації, що зберігається у власній базі даних MySQL.

Protégé – локальна, вільно поширювана Java-програма, призначена для побудови (створення, редагування й перегляду) онтологій ПрО, що дає змогу проектувати онтології, розгортаючи ієрархічну структуру абстрактних і конкретних класів і слотів.

Semantic MediaWiki – плагін для MediaWiki, який дає змогу використовувати у Wiki-ресурсі семантичні властивості.

Semantic Web – проект семантичного розширення WWW, у рамках якого інформація (Web-контент) представляється у форматах, що забезпечують її використання програмними агентами, дозволяючи їм, таким чином, шукати, розділяти і інтегрувати інформацію. Ця розробка, розпочата у 2000 році Т.Бернесом-Лі, спрямована на перетворення середовища Web на глобальну розподілену базу знань, яку можуть використовувати різноманітні інтелектуальні застосунки.

Wiki-двигун – спеціальне програмне забезпечення, яке забезпечує роботу відповідного Web-сайту, на якому розміщено Wiki-ресурс та забезпечує можливість того, щоб у будь-який час сторінку можна було б повернути до кожного з її попередніх станів.

Wiki-запит – елемент Wiki-розмітки, що дає можливість фільтрувати Wiki-сторінки за обраними в запиті критеріями (за категоріями та семантичними властивостями) та виводити на екран тільки необхідні користувачеві дані як результати запиту.

Wiki-категорія – група Wiki-сторінок, об'єднаних спільними властивостями. Кожна Wiki-сторінка може бути віднесена до довільної кількості категорій, які можуть бути взаємопов'язаними ієрархічними зв'язками.

Wiki-ресурс – контент Web-сайту, який працює на програмному забезпеченні, що підтримує технологію Wiki (приміром, на MediaWiki) та забезпечує вільний доступ до матеріалів таким чином, що користувачі можуть не тільки їх читати, а й додавати та редагувати.

Wiki-розмітка – розмітка, яка використовується для представлення інформації (тексту та мультимедіа) на Wiki-сторінках.

Wiki-сторінка – будь-яка окрема сторінка Wiki-ресурсу, яка має унікальне ім'я.

Wiki-технологія – така технологія побудови Wiki-ресурсу, яка дає змогу відвідувачам брати участь у редагуванні його вмісту – виправляти помилки, додавати нові матеріали, не використовуючи спеціальні програми, не реєструючись на сайті й не вивчаючи мову HTML

Вікіпедія – енциклопедія, доступ до якої здійснюється через Web. Це найбільший у світі Wiki-ресурс, який усі відвідувачі . можуть редагувати й доповнювати власними статтями

Гіперпосилання у Wiki-сторінці – посилання з однієї статті на іншу без формалізації семантики цього гіперпосилання. Базове програмного забезпечення підтримує посилання як на зовнішні інформаційні ресурси (Web-сайти, сторінки інших Wiki-ресурсів тощо), так і на інші сторінки даного Wiki-ресурсу. Натиснувши на таке посилання, можна перейти до перегляду обраного ресурсу.

e-VUE – електронна версія Великої української енциклопедії, яка містить більше інформації порівняно з паперовою версією – вона доповнюється мультимедійними ресурсами, сторінками-інтеграторами, системою семантичної навігації тощо. .

Інтелектуальна інформаційна система – один з видів автоматизованих інформаційних систем, функціонування якої базується на використанні знань та вирішує завдання аналізу та синтезу інформації. Функціонування ІС можна описати як постійне прийняття рішень на основі аналізу поточних ситуацій для досягнення певної цілі.

Інформаційний об'єкт – інформаційна модель об'єкта певної предметної області (ПрО), що описує структуру, атрибути, обмеження цілісності, поведінку цього об'єкта. У семантичному пошуку ІО – це інформація, яку користувач отримує в результаті пошуку.

Онтологія – формальний опис результатів концептуального моделювання різних предметних областей, що забезпечує інтероперабельність та повторне використання знань в інтелектуальних системах. Як правило, онтології описують основні поняття предметної області, відношення між ними та їх властивості.

Семантичні властивості – це властивості Wiki-сторінок, які використовуються для того, щоб позначати змістове навантаження посилання на інші Wiki-сторінки або дані.

Спеціальна Wiki-сторінка – сторінка Wiki-ресурсу, яка має спеціальне призначення та використовує окремий простір імен (це забезпечує від сплутування спеціальних Wiki-сторінок зі звичайними Wiki-сторінками). До спеціальних Wiki-сторінок належать сторінки Wiki-категорій, сторінки Wiki-шаблонів та сторінки семантичних властивостей.

Тезаурус інформаційного ресурсу – окремий випадок онтології, множина понять якого є підмножиною з тих понять онтології ПрО, які семантично пов'язані з певними елементами контенту цього ІР, а множини відношень та аксіом є порожніми. Тезаурус можна розглядати як онтологічний опис ІР, який характеризує його семантику. З точки зору користувача, такий тезаурус – це множина слів (можливо, з певними коефіцієнтами).

Шаблон – елемент Wiki-сторінки, що може містити як елементи форматування (приміром, розмір, колір та шрифт тексту), так і контент (текст, мультимедійні елементи), та призначений для уніфікованого оформлення однотипних даних. Це спеціальні Wiki-сторінки, вміст яких призначено для включення (вбудовування) в інші сторінки. Кожна Wiki-сторінка може містити довільну кількість шаблонів.

Додаток 4. Перелік скорочень

GFDL	GNU Free Documentation License
OS-INT	Open Source Intelligence
OWL	Ontology Web Language
RDF	Resource Description Framework
SMW	Semantic Media Wiki
XML	Extensible Markup Language
БЗ	база знань
ВУЕ	Велика українська енциклопедія
е-ВУЕ	Електронна версія Великої української енциклопедії
ПС	інтелектуальна інформаційна система
ПС	інтелектуальна інформаційна система
ІО	інформаційний об'єкт
ПІС	інформаційно-пошукова система
ІР	інформаційний ресурс
ПЗ	програмне забезпечення
ПрО	предметна область
ССП	система семантичного пошуку

ЛІТЕРАТУРА

1. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. Edition 1.2., 2009 [Електронний ресурс]. – Режим доступу : <http://phd.jabenitez.com/wp-content/uploads/2014/03/A-Practical-Guide-To-Building-OWL-Ontologies-Using-Protege-4.pdf>.
2. *Berners-Lee T., Hendle J., Lassila O.* The semantic web. Scientific american, 284(5), 2001, P.28-37. – <https://pdfs.semanticscholar.org/566c/1c6bd366b4c9e07fc37eb372771690d5ba31.pdf>.
3. *Cyganiak R.* RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation 25 February 2014 [Електронний ресурс] / Cyganiak R., Wood D., Lanthaler M. – Режим доступу : <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
4. *Davies J., Fensel D., van Harmelen F.* Towards the Semantic Web: Ontology-driven knowledge management. John Wiley & Sons Ltd., England, 2002, 288 p.
5. *Gruber T. R.* Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies, 1995, V. 43, Issues 5-6. P. 907-928.
6. *Gruber T. R.* What is an Ontology? [Електронний ресурс] / T. R. Gruber. – Режим доступу : <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
7. Guarino N. Formal Ontology in Information Systems / N. Guarino // Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, June 6-8, 1998 – Amsterdam : IOS Press, 1998. – P. 3—15.
8. *Finin T. W., Ding L., Pan R., Joshi A., Kolari P., Java A., Peng Y.* Swoogle: Searching for knowledge on the Semantic Web // Proc. AAAI-2005, AAAI Press / MIT Press, 2005. P. 1682-1683.
9. *Heflin J., Hendler J. A., Luke S.* SHOE: A blueprint for the Semantic Web. In D. Fensel, W. Wahlster, and H. Lieberman, editors. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, MIT Press, 2003. – P. 29-63.
10. Introduction to Semantic Web. – <http://www.mphasis.com/knowledge-center/white-papers-all.asp>.
11. *Lassila O.* Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation [Електронний ресурс] / O. Lassila, R. Swick. – Режим доступу : <http://www.w3.org/TR/REC-rdf-syntax>.
12. *Lassila O., Swick R.* Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation. – <http://www.w3.org/TR/REC-rdf-syntax>.
13. *Leuf B.* The Wiki way: collaboration and sharing on the Internet, 2001 [Електронний ресурс] / B. Leuf, W. Cunningham. – Режим доступу : <http://www.citeulike.org/group/13847/article/7659081>.
14. *Mizoguchi R.* A Step Towards Ontological Engineering [Електронний

- ресурс] / R. Mizoguchi. – Режим доступа : <http://www.ei.sanken.osaka-u.ac.jp/english/step-onteng.html>.
15. *Noy N. F.* Ontology Development 101: A Guide to Creating Your First Ontology [Электронный ресурс] / N. F. Noy, D. L. McGuinness. – Режим доступа : http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.
 16. Open Source Initiative [Электронный ресурс]. – Режим доступа : <http://opensource.org>.
 17. OSINT (Open Source Intelligence) – Online Strategies [Электронный ресурс]. – Режим доступа : <http://www.onstrat.com/osint/>.
 18. OWL 2 Web Ontology Language Document Overview. W3C. 2009 [Электронный ресурс]. – Режим доступа : <http://www.w3.org/TR/owl2-overview/>.
 19. OWL Web Ontology Language Semantics and Abstract Syntax. Section 2. Abstract Syntax [Электронный ресурс]. – Режим доступа : <http://www.w3.org/TR/owl-semantics/syntax.html>.
 20. OWL Web Ontology Language Semantics and Abstract Syntax. Section 2. Abstract Syntax. – <http://www.w3.org/TR/owl-semantics/syntax.html>.
 21. Protégé [Электронный ресурс]. – Режим доступа : <http://protege.stanford.edu/>.
 22. Protégé. W3C Semantic Web [Электронный ресурс]. – Режим доступа : <http://www.w3.org/2001/sw/wiki/Protégé>.
 23. RDF Vocabulary Description Language 1.0: RDF Schema. RDF Vocabulary Description Language 1.0. World Wide Web Consortium, 2010 [Электронный ресурс]. – Режим доступа : <http://www.w3.org/Metadata/Activity.html>.
 24. Resource Description Framework (RDF) Model and Syntax Specification. W3C Proposed Recommendation. January 1999 [Электронный ресурс]. – Режим доступа : <http://www.w3.org/TR/PR-rdf-syntax>.
 25. *Rogushina J., Pryima S.* Use of ontologies and the Semantic Web for qualifications framework transparency // Eastern-European Journal of Enterprise Technologies, 1/2 (85) 2017. – P.25-30.
 26. *Rogushina J.V.* The Use of Ontological Knowledge for Semantic Search of Complex Information Objects // Материалы VII международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2017, Минск, БГУИР, 2017. – С.127-132.
 27. *Rogushina J.V., Gladun A.Y.* Ontological Approach to Matching of the Domain Competence of Specialists for Research Projects // Материалы VII международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2017, Минск, БГУИР, 2017. – С.133-138.

28. *Rogushina J.* Ontology-based competency analyses in new research domains [Text] / J. Rogushina, A. Gladun // Journal of Computing and Information Technology. – 2012. – V. 20. – № 4. – P. 277–293.
29. *Souzis A.* Building a semantic wiki. Intelligent Systems [Електронний ресурс] / A. Souzis // IEEE, 2005, 20(5). – P. 87—91. – Режим доступу : http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1512004&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1512004.
30. SPARQL Query Language for RDF. W3C Recommendation, 2008 – <http://www.w3.org/TR/rdf-sparql-query/>.
31. *Tudorache T.* Collaborative Protege: Enabling Community-based Authoring of Ontologies [Електронний ресурс] / Tudorache T., Noy N. F., Musen M. A. // International Semantic Web Conference, 2008. – Режим доступу : http://ceur-ws.org/Vol-401/iswc2008pd_submission_60.pdf.
32. *Ushold M., Gruninger M.* Ontologies: Principles, Methods and Applications, Knowledge Engineering Review, V.11,N 2, 1996.
33. W3C Semantic Web Activity [Електронний ресурс]. – Режим доступу : <http://www.w3.org/2001/sw/Activity/>.
34. W3C Semantic Web Activity. – Режим доступу : <http://www.w3.org/2001/sw/Activity/>.
35. *Wagner C.* Wiki: A technology for conversational knowledge management and group collaboration [Електронний ресурс] // The Communications of the Association for Information Systems. – 2004. – V. 13(1). – P. 264—289. – Режим доступу : <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=3238&context=cais>.
36. WikipediA [Електронний ресурс]. – Режим доступу : <https://www.wikipedia.org>.
37. Вікіпедія. – <http://uk.m.wikipedia.org>.
38. *Гаврилова Т. А.* Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2001.
39. *Гладун А.Я.* Семантичні технології: принципи та практики / А.Я. Гладун, Ю.В. Рогущина – К.:ТОВ "ВД "АДЕФ-Україна", 2016. – 308 с
40. *Гладун А. Я.* Онтологии в корпоративных системах. Часть 1. / А. Я. Гладун, Ю. В. Рогущина // Корпоративные системы. – 2005. – № 6. – С. 23
41. *Гладун А. Я.* Онтологии в корпоративных системах. Часть 2. / А. Я. Гладун, Ю. В. Рогущина // Корпоративные системы. – 2006. – № 1. – С. 23–
42. *Глибовец А. Н.* Семантическая паутина и Wiki-системы / Глибовец А. Н., Глибовец Н. Н., Покопцев Д. Е., Сидоренко М. О. // Проблемы програмування. – 2013. – № 1. – С. 45–67.
43. *Гриценко В. И.* Семантическое распознавание информационных объектов на основе онтологического представления знаний о предметной области в

- задачах інтелектуального управління / Гриценко В. И., Гладун А. Я., Рогушина Ю. В. // Кибернетика и вычислительная техника. – 2014. – № 4. – Вып. 178. – С. 5–20.
44. *Херциг Д.М., Эль Б.* Semantic MediaWiki в действии: опыт построения семантического портала
45. *Добров Б. В.* Онтологии и тезаурусы: модели, инструменты, приложения [Электронный ресурс] / Добров Б. В., Иванов В. В., Лукашевич Н. В., Соловьев В. Д. – Электронная книга, 2006. – 220 с. – Режим доступа : http://catscpp.googlecode.com/svnhistory/r146/trunk/diploma/materials/ontologies_tesauresusespdf.
46. *Катков Ю. В.* Мастер-класс по SPARQL [Электронный ресурс] / Ю. В. Катков. – Режим доступа : http://test.wikivote.ru/index.php/Мастер-класс_по_SPARQL.
47. *Клещев А. С.* Математические модели онтологий предметных областей. Часть 1.: Существующие подходы к определению понятия «онтология» / А. С. Клещев, И. Л. Артемьева // Научно-техническая информация. – 2001. – Серия 2. – С. 20—27.
48. *Лесько О. Н.* Использование специализированной лексической онтологии для автоматизации формирования онтологии предметной области по естественно-языковым текстам / О. Н. Лесько, Ю. В. Рогушина. – Information Models of Knowledge. Edited by K. Markov, V. Velychko, O. Voloshin. – IT NE A, Kiev-Sofia, 2010. – P. 93–100.
49. *Луговий, В.* Національна рамка кваліфікацій: розуміння і реалізація / В. Луговий, Ж.Таланова [Текст] // Проф.-техн. освіта. – 2010. – № 1. – С. 5-9.
50. Методичні рекомендації з підготовки, редагування та оформлення статей до Великої української енциклопедії (проект) / За ред. д. і. н., проф. Киридон А.М. – К.: ДНУ «Енциклопедичне видавництво», 2015. – 120 с.
51. *Нариньяни А. С.* Кентавр по имени ТЕОН: Тезаурус + Онтология [Электронный ресурс] / А. С. Нариньяни. – Режим доступа : <http://www.artint.ru/articles/narin/teon.htm>.
52. Наукові засади та теоретико-методологічні принципи створення сучасних енциклопедій: колективна монографія / За ред. д. і. н., проф. Киридон А.М. – К.: ДНУ «Енциклопедичне видавництво», 2015. – 160 с.
53. *Никоненко А. А.* Обзор баз знаний онтологического типа [Электронный ресурс] / А. А. Никоненко // Штучний інтелект. – 2009. —№ 4. – С. 208—219. – Режим доступа : <http://dSPACE.nbuv.gov.ua/bitstream/handle/123456789/8144/27-Nikonenko.pdf>.
54. *Рогушина Ю. В.* Семантичний пошук у Web на основі онтологій: розробка моделей, засобів і методів / Ю. В. Рогушина. – Мелітополь: МДУПУ ім. Богдана Хмельницького, 2015. – 291 с.
55. *Рогушина Ю.В.* Класифікація засобів та методів семантичного пошуку в Web // Проблеми програмування. 2017. № 1. – С.30-50.
56. *Рогушина Ю.В., Гладун А.Я., Осадчий В.В., Прийма С.М.* Онтологічний

- аналіз у Web. Монографія. – Мелітополь: МДУПУ ім. Б.Хмельницького, 2015, 407 с.
57. Рогушина Ю. В. Знание-ориентированные средства поддержки семантического поиска в Web / Ю. В. Рогушина. – LAP LAMBERT Academic Publishing. – 214 с.
58. *Рогушина Ю. В.* Семантическая Википедия как источник онтологий для интеллектуальных поисковых систем / Ю. В. Рогушина, А. Я. Гладун // В кн.: Advanced Research in Artificial Intelligence. International Book Series «Information Science and Computing». – ITHEA, Sofia, 2008. – P. 172—178.
59. *Рогушина Ю. В.* Тезаурус предметной области – инструмент представления знаний при проблемно-ориентированном поиске в Web / Ю. В. Рогушина, А. Я. Гладун // Штучний інтелект. Інтелектуальні системи: збірник праць Міжнародної науково-технічної конференції. – К., 2010. – Т. 2. – С. 78—90.
60. *Рогушина Ю. В.* Формирование тезауруса предметной области как средства моделирования информационных потребностей пользователя при поиске в Интернете / Ю. В. Рогушина, А. Я. Гладун // Вестник компьютерных и информационных технологий. – 2007. – № 1.
61. Руководство по Protege 4. 2. [Электронный ресурс] / – Режим доступа : <https://docs.google.com/document/d/1Fg9u9pf5RXBu8bklLh48MkZ01-DLrd8hn3ZJbTh6xg4/edit#>.
62. Як редагувати Вікіпедію: Довідник для новачків [Текст] / Упоряд. С. Д. Петров; За заг. ред. Є. В. Букета. – К.: Український пріоритет, 2012. – 44 с.