УДК 681.3.

UWN: УНІВЕРСАЛЬНА ОНТОЛОГІЧНА БАЗА ЗНАНЬ УКРАЇНСЬКОЇ МОВИ

А.В. Анісімов, О.О. Марченко, А.О. Никоненко

Київський національний університет імені Тараса Шевченка, факультет кібернетики,
03680, Київ, проспект Академіка Глушкова, 2, корпус 6
Тел.: (044) 259 0119; Факс: (044) 259 0439; E-mail: dean@unicyb.kiev.ua

Дана стаття описує мету та основні задачі, які ставилися розробниками при створенні онтології UWN. Детально розглянуто архітектуру створеної системи, структуру збереження знань, механізми доступу до знань, внутрішню логіку роботи системи та деякі ключові програмні модулі. Також стаття містить опис двох системних утиліт, що забезпечують доступ до даних онтології: онтокоректора та онторедактора.

This article describes the purpose and main objectives that were set by UWN ontology developers. The architecture of the created system, structure of knowledge storage, knowledge access mechanism and internal logic key elements are considered in details. The article also describes two system utilities which provide access to the ontology data: ontocorrector and ontoeditor.

Вступ

Впродовж останніх десятиріч одним з найбільш важливих і актуальних напрямків у штучному інтелекті традиційно вважалася інженерія та розробка баз знань. Бази знань є фундаментальною основою для створення різноманітних інтелектуальних інформаційних систем – від експертних систем до систем інтелектуальної обробки природномовної інформації. За цей час бази знань значно розвинулися і еволюціонували: семантичні мережі, фреймові моделі, сценарії та онтології. Без застосування технологій баз знань неможливо сьогодні уявити побудову «розумної» програми для ефективного розв'язання складних інформаційних задач в реальному часі.

Серед багатьох сучасних баз знань вирізняється одна з найбільш популярних – лексична семантична база знань, онтологія WordNet [1], яка вже три десятки років розробляється командою вчених-дослідників Прінстонського університету (США). За цей час WordNet стала фактично загальноприйнятим стандартом в своєму класі онтологічних баз. Вона використовувалася в якості ядра бази знань при створенні численних інтелектуальних систем широкого спектру застосувань. Дуже скоро майже у всіх розвинутих країнах Європи було відкрито ряд проектів щодо створення баз знань WordNet для своїх власних національних мов. Черговий етап розвитку WordNet-баз був пов'язаний з проектом EuroWordNet [2] (1996─1999р.), в рамках якого не тільки було створено декілька мереж для європейських мов (голландської, іспанської, італійської, німецької, французької, чеської і естонської), але і вперше була реалізована ідея про об'єднання окремих WordNet-представлень в загальну систему. Всі компоненти EuroWordNet були побудовані за єдиною моделлю. Перед розробниками стояла задача відобразити всі особливості лексичних систем національних мов. Сумісність компонентів EuroWordNet забезпечувалася єдністю принципів і заданим набором загальних понять, на яких визначалася система міжмовних посилань, що дають можливість переходити від лексикалізованих значень однієї мови до схожих, але не обов'язково тотожних значень в іншій мові. Успішне завершення проекту EuroWordNet стало поштовхом до створення великого числа WordNet-баз для мов різних типів (угорської, турецької, арабської, тамільської, китайської і ін.), а також багатомовних ресурсів типу EuroWordNet (наприклад, проект BalkaNet націлений на об'єднання грецького, румунського, болгарського, сербського, турецького і чеського wordnet-словників). У 2001р. створена Всесвітня Асоціація WordNet (Global WordNet Association), метою якої є об'єднання вже існуючих ресурсів цього типу, вдосконалення системи міжмовних відношень і розробка загальних стандартів, що дозволяють використовувати модель WordNet для мов різних типів.

Дана стаття висвітлює основні результати роботи над проектом створення української онтологічної лексико-семантичної бази знань UkrWordNet (UWN). Дослідження ведуться на кафедрі математичної інформатики факультету кібернетики Київського національного університету імені Тараса Шевченка. Проект триває вже декілька років і досягнуто значних успіхів щодо мовної локалізації лексичних вузлів мережі, побудови та наповнення власних структур бази UWN, створення інструментарію для автоматизації заповнення структур бази знань. Розроблено багаторівневий програмний комплекс для поповнення, редагування даних бази та модерації роботи редакторів у мережі. Створено ряд програм для візуалізації вмісту бази знань та забезпечення зручного інтерфейсу доступу до даних мережі.

Архітектура системи

Розробка будь-якого великого проекту починається з планування, а розробка інформаційної системи зі створення списку вимог та вибору архітектури, що здатна задовольнити цим вимогам. У проекті UWN ми

ставили собі за мету створити не ще одну онтологію чи мову опису онтологій, не онторедактор і не проект по спільному створенню бази знань, ─ наразі існує досить багато об'єктів кожного з описаних класів. У основі проекту лежить ідея про об'єднання в одному місці даних про мову та алгоритмів їх обробки, даний підхід дозволяє значно підвищити швидкість опрацювання даних, а тому є принципово корисним для використання в складних лінгвістичних методах аналізу природномовних текстів. Нашою головною задачею було створити технологічну платформу, що поєднувала б у собі онтологічну базу знань, морфологічну базу знань, логіку для обробки цих знань, була доступною онлайн, забезпечувала можливості спільної роботи, дозволяла проводити розробку лінгвістичних додатків та містила набір інструментів для роботи зі своїм вмістом. Тому розробка правильного дизайну системи була однією з ключових задач реалізації проекту. Спираючись на вищеописану ідею до системи було висунуто наступні вимоги:

- доступність даних онлайн;
- швидкість доступу до даних;
- простота редагування та поповнення бази;
- доступ до даних як через спеціальні утиліти так і в «ручному» режимі;
- створення гнучкої структури збереження даних, що могла б бути легко змінена або розширена;
- забезпечення автоматичного перетворення даних вбудованими засобами при зміні структури метаданих;
- підтримка зв'язності між концептами різномовних онтології;
- забезпечення безпеки доступу до даних;
- забезпечення контролю змін даних вбудованими засобами;
- можливість об'єднання даних та алгоритмів з їх обробки в єдиній точці;
- стабільна робота в багатокористувацькому режимі.

Детальний аналіз вимог показав, що для побудови системи найбільш вдало можна використати дворівневу клієнт-серверну архітектуру. В даному випадку основною перевагою дворівневої архітектури над трирівневою є можливість розгортання серверної логіки на тому ж мережевому вузлі, де відбувається збереження даних. Роль програмного сервера (application server) та бази даних виконує СУБД Оракл. Він являє собою типове рішення великих корпорацій для ведення обліку користувачів та білінгу. Висока надійність, якість, швидкість роботи та широкі можливості внутрішньої мови PL/SQL забезпечили популярність цієї СУБД серед мобільних операторів, бірж та банків. Нами застосовано Оракл у вирішенні задач комп'ютерної лінгвістики. Як базову версію обрано Oracle Database 10g Express Edition (також відому як Oracle Database XE), доступну за ліцензією [3].

Основні структурні елементи. Основними структурними елементами для розміщення логіки та даних в СУБД Оракл виступають схеми. Кожна схема може містити таблиці з даними, модулі з програмною логікою, окремі процедури та функції, користувацькі типи даних, тригери та механізми запуску процесів за розкладом. Крім того, кожна схема може мати свої політики безпеки та доступу до об'єктів інших схем. Основні елементи програмної логіки та збереження даних UWN, а також зв'язки між ними показано на рис. 1.

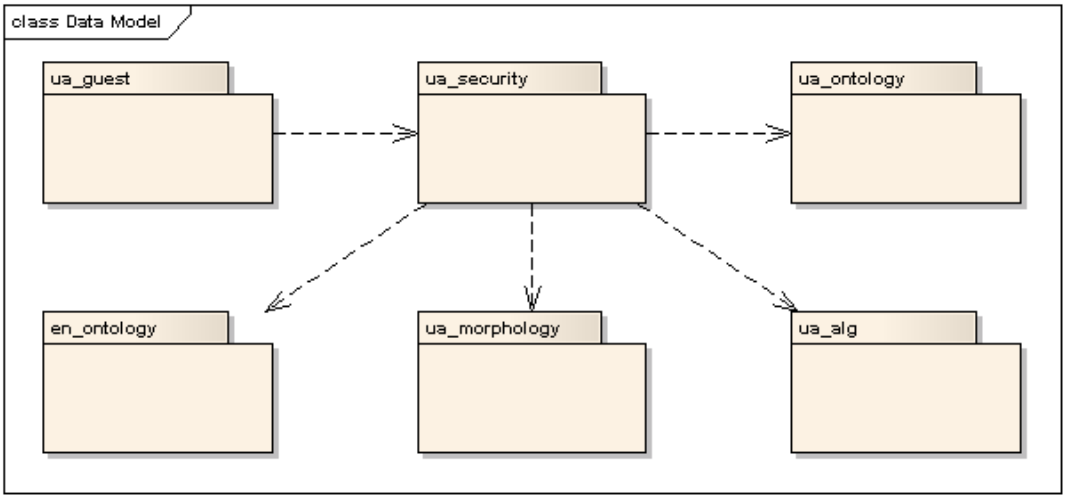

Рис. 1. Схема взаємодії основних структурних елементів UWN

Розглянемо детально призначення ключових блоків:

- ***ua_guest*** – технологічна схема, що використовується для підключення клієнтів до БД. СУБД Оракл не розрізняє поняття схеми та користувача, тобто кожен користувач, що встановлює з'єднання з БД в якості

логіну має використовувати ім'я схеми. Як єдину точку підключення зовнішніх систем до UWN нами було створено схему ***ua_guest***, що виконує роль буферної зони для системи безпеки і дозволяє встановити з'єднання з UWN будь-якому користувачу чи системі. Дана схема не містить жодного об'єкта та прав на їх створення, також вона не має прав доступу до об'єктів інших схем. Єдине виключення становлять інтерфейси доступу до системи безпеки UWN. Детально один з інтерфейсів описано в [4];

- ***ua_security*** – схема, що відповідає за аутентифікацію систем, що підключаються до UWN, визначення набору повноважень та надання прав доступу відповідно до профілю системи-додатку. Також дана схема виконує роль єдиного інтеграційного вузла доступу до всіх внутрішніх інтерфейсів UWN, тобто до серверної логіки, що здійснює обробку та модифікацію даних онтологічної та морфологічної баз. Додатково, для підвищення рівня безпеки системи, схема містить механізми логування доступів (включаючи час, IP-адресу та інші характеристики машини-клієнта) та логування списку команд, що виконує додаток;
- ***ua_ontology*** – схема, що зберігає інформацію про наповнення україномовної онтології. Також схема містить серверну логіку для роботи з даними онтології: пошук синсетів, отримання інформації синсету, зміна даних синсету, отримання та виставлення спеціальних ознак синсету, отримання інформації про зв'язки в онтології та зміна цієї інформації. Крім логіки для роботи з онтологією, самої онтології та механізму відстеження та логування змін даних в схемі також розгорнуто серверну логіку кількох програм-додатків призначених для опрацювання знань з онтології;
- ***en_ontology*** – схема, що зберігає інформацію про наповнення англомовної онтології. Аналогічно до ***ua_ontology*** схема містить серверну логіку для роботи з даними онтології та механізм відстеження і логування змін даних;
- ***ru_ontology*** – схема, що запланована для зберігання інформації російськомовної онтології;
- ***ua_alg*** – схема, що використовується для зберігання різноманітних семантичних алгоритмів та серверної логіки лінгвістичних додатків, а також для розміщення програмно-алгоритмічних частин лінгвістичних наукових досліджень (наприклад, програмних реалізацій методів вимірювання ступеню семантичної зв'язності та схожості);
- ***ua_morphology*** – схема, що зберігає морфологічний словник української мови та містить механізми доступу до його інформації, також в даній схемі розміщено алгоритми перевірки правопису та підбору варіантів правильного написання слова.

Представлення знань. Розробка структури онтології (або побудова мови представлення онтологічних знань), як було нами показано в [5], є досить складним і тривалим процесом, тому, замість розробки власного формату онтології, за основу нами було взято формат знань характерний для сімейства WordNet. Відповідно, логічна структура представлення знань в UWN відповідає формату WordNet та EuroWordNet ─ знання подано у вигляді концептів (синсетів) та зв'язків між ними. Фізично ж дані розміщено в спеціальній табличній структурі бази даних. Дещо спрощена діаграма основних елементів схеми ***ua_ontology***, в якій зберігаються дані україномовної онтології, показано на рис. 2.

Вкажемо призначення основних таблиць:

- ***synsets*** – таблиця, що зберігає основну інформацію про синсет: ідентифікатор синсету; частину мови; глосарій; стан коректності глосарію; стан коректності синсету; дату останньої зміни даних синсету; ідентифікатор редактора, що вносив зміни; ідентифікатор модератора, що перевіряв зміни; та деякі додаткові дані;
- ***synsets_log*** - таблиця, що зберігає інформацію про зміни синсету: попередній глосарій; попередній стан коректності глосарію; попередній стан коректності синсету; попередній редактор; попередній модератор; час попередніх змін та ідентифікатор виконаної над синсетом дії (створення, зміна, видалення);
- ***words*** – таблиця, що містить слова синсету. Основні поля таблиці: ідентифікатор слова; ідентифікатор синсету; частина мови; слово або словосполучення, що належить синсету; ідентифікатор редактора, що вносив зміни; стан коректності слова; дата останньої зміни даних слова; ідентифікатор модератора, що перевіряв зміни; та деякі додаткові дані;
- ***words_log*** – таблиця, що зберігає інформацію про зміни слів: попереднє слово (словосполучення), що належить синсету; попередній стан коректності слова; попередній редактор; попередній модератор; час попередніх змін та ідентифікатор виконаної над словом дії (створення, зміна, видалення);
- ***semantic_relations*** – таблиця, що зберігає інформацію про семантичні зв'язки між синсетами. Основними полями є ідентифікатор і частина мови синсету з якого виходить зв'язок; тип зв'язку та ідентифікатор і частина мови синсету в який входить зв'язок;
- ***lexical_relations*** – таблиця, що зберігає інформацію про лексичні зв'язки між словами синсетів. Основними полями є ідентифікатор і частина мови слова з якого виходить зв'язок; тип зв'язку та ідентифікатор і частина мови слова в яке входить зв'язок;
- ***lexicograph_dict*** – таблиця-словник, що зберігає інформацію про лексикографічні типи синсетів. Основні поля: лексикографічний ідентифікатор типу; частина мови, якій відповідає даний лексикографічний тип; назва секції (наприклад, «мотив», «почуття» і т. д.) та її опис (наприклад, «іменники що позначають цілі», «іменники що позначають почуття і емоції» і т. д.).

- ***system_users*** – таблиця, що зберігає інформацію про профілі користувачів, які можуть користуватися програмами-додатками, призначеними для роботи зі знаннями онтології, серверна частина яких розміщена в схемі ***ua_ontology***.

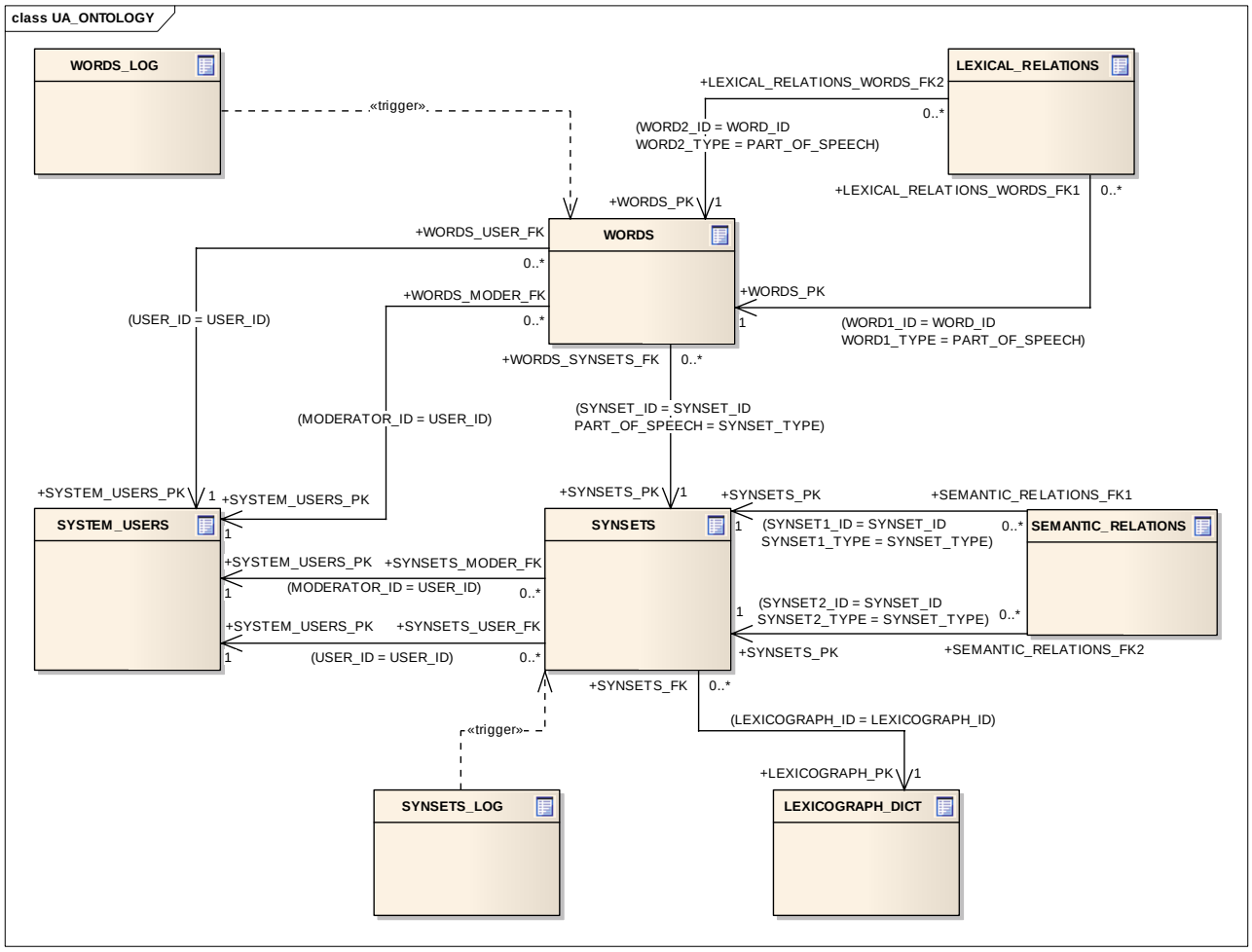

Рис. 2. Структура розміщення знань у схемі ua_ontology

Ключові логічні модулі. Алгоритмічно-логічна складова системи розгорнута на базі ключових структурних вузлів зображених на рисунку 1 і складається з кількох типів модулів.

1. Перевірки прав та забезпечення доступу до об'єктів системи.
2. Логування входу/виходу користувача та списку виконаних дій.
3. Логування змін.
4. Модулі роботи з даними онтології.
5. Модулі з лінгвістичною логікою.
6. Модулі підсистем, що містять серверну логіку для клієнтських додатків.

Розглянемо структуру та використання цих модулів детально:

1. Перевіркою прав доступу клієнтських додатків до серверної логіки займається модуль **security_pkg**, котрий виконує перевірку прав на виклик кожної функції, доступ до якої намагається отримати клієнт. Розрізнення клієнтів відбувається за допомогою функції реєстрації, тобто будь-який клієнтський додаток, що встановив з'єднання з сервером UWN через технологічний аккаунт ***ua_guest***, має вказати системі клієнтом якої підсистеми він є (наприклад, онторедактору, онтокоректору чи інших). Якщо сервером успішно ідентифіковано тип додатку, то він отримує права на запуск процедур та функцій у відповідності до профілю доступу даної підсистеми. Якщо додаток не ідентифікував себе в системі, то він не отримує жодних прав і не може виконувати ніяких дій в системі.

За доступ до внутрішніх об'єктів системи відповідає модуль **run**, що містить у собі інтерфейси до внутрішніх пакетів та процедур системи. Фактично, модуль **run** представляє собою єдину точку доступу до всієї серверної логіки розгорнутої всередині UWN. Детально процес аутентифікації клієнта, видачі йому

системних прав та реалізація механізмів доступу до серверної логіки описано в [6]. В даний момент, система має два типи загальнодоступних (тобто таких, доступ до яких має будь-який клієнт, що під'єднався до серверної частини UWN через технологічний аккаунт ***ua_guest***) інтерфейсів:

- інтерфейс типу ***get***, також відомий як «старий інтерфейс» - прошарок, що містить лише методи отримання інформації з онтології. Серед них: пошук синсетів, у які входить певне слово; побудова ієрархічних дерев за різними типами зв'язку; пошук синонімів і т.д. Даний інтерфейс використовується в ескізному проекті семантичної пошукової системи та альфа-версіях клієнтів для перегляду наповнення онтології. Детально інтерфейс описано в роботі [4].
- інтерфейс типу ***get/set***, також відомий як «новий інтерфейс» - прошарок, призначений для двонаправленої роботи з БД, тобто, як для отримання інформації з бази так і для внесення змін. Даний інтерфейс використовується в більшості клієнтських додатків, в тому числі в останніх версіях онтокоректорів та онторедакторів, а також в підсистемах призначених для вирішення конкретних задач інтелектуальної обробки тексту. Саме цей інтерфейс реалізує модуль **run**.

2. Логування підключення клієнтської машини до сервера (в якості зареєстрованого додатку) та ведення списку виконаних дій здійснює модуль **security_pkg.** Запис про підключення нового додатку створюється відразу після успішної реєстрації підсистеми, в разі відсутності реєстрації, або у випадку невдалого проходження реєстрації запис про підключення додатку не вноситься. Питаннями збору та аналізу інформації про нелегальні спроби доступу до системи та спроби підключення незареєстрованих типів додатків займається окрема підсистема безпеки, розгорнута на базі адміністративної схеми, опис якої виходить за межі даної роботи.

Логування списку виконаних дій також забезпечує модуль **security_pkg**, а саме його підсистема перевірки прав доступу клієнтського додатку до серверних функцій. Дана підсистема відповідає за виклик процедур логування дій додатку. У зв'язку з великою кількістю команд, що надходять від клієнту до сервера, та великою кількістю клієнтів, що одночасно працюють з сервером, даний функціонал здатен генерувати великі об'єми рідко використовуваних даних. Тому цей тип логування, в основному, використовується на етапі тестування нових додатків, і може бути сконфігуровано для відстеження лише певних команд чи підсистем. Налаштування системи логування дій відбувається шляхом внесення даних про цільову команду чи підсистему до спеціальної таблиці параметрів.

3. На відміну від модулів забезпечення контролю, безпеки та доступу, вся логіка яких зосереджена в одній точці, механізм логування змін ключових даних має децентралізований характер, його елементи розміщено в різних схемах. Кожен вузол системи, що містить важливі дані, зміни яких необхідно відстежувати, має свій екземпляр механізм логування. Ключову роль в роботі механізму виконують тригери та журнали змін даних, які дозволяють відстежити час зміни, характер внесених правок, користувача що їх зробив та деякі інші параметри. У зв'язку зі значним розміром журналів змін, що формуються при багатокористувацькій роботі з системою, їх використано лише для ключових даних. Як видно з рисунку 2, активне застосування механізм знайшов в схемі ***ua_ontology,*** схожа ситуація в інших онтологічних схемах та в схемі ***ua_morphology***.

4. Модулі для роботи з даними кожної конкретної онтології розміщено в тій же схемі, що й онтологію. Загалом в системі існує кілька типів таких модулей, що відрізняються складністю реалізованої логіки. Найбільш простими є модулі, що представляють собою інтерфейси для роботи зі знаннями онтології, такі модулі містять функції з пошуку, видалення, створення та редагування даних. Цей тип модулів представляє собою прошарок між клієнтами, що працюють з даними онтології, та самою онтологією, їх основна мета - гарантувати коректність роботи інших систем з онтологією. Іншим поширеним типом є модулі, що реалізують цілі підсистеми для роботи з даними онтології. Класичний приклад таких підсистем – онторедактор та онтокоректор, побудовані для роботи з україномовною онтологією і розгорнуті в схемі ***ua_ontology***. Також у вигляді окремих модулів можуть бути реалізовані частини систем дослідження особливостей онтологій та проведення експериментальних випробувань з використання їх вмісту.

5. Модулі з лінгвістичною логікою представляють собою окремий клас допоміжних засобів з проведення різних етапів інтелектуального аналізу і розгорнуті в схемі ***ua_alg***. Типовим прикладом таких модулів буде модуль, що містить програмну реалізацію різних методів вимірювання ступеню семантичної зв'язності та схожості на основі онтології.

6. Модулі підсистем використовуються для реалізації серверної частини конкретної прикладної системи, що вирішує певну комп'ютерно-лінгвістичну задачу. Такі модулі містять усю програмну логіку взаємодії кінцевої системи з UWN та розміщуються в схемі ***ua_alg***, або в окремих, спеціально виділених схемах (у випадку високої складності серверної частини та необхідності в розгортанні додаткової інфраструктури).

Модель доступу та захист інформації.

Основною проблемою при розробці загальнодоступних клієнт-серверних систем є безпека даних від несанкціонованого доступу та забезпечення сумісної роботи великої кількості користувачів з одніми й тими ж даними, також важливими виступають питання розподілу та обмеження серверних ресурсів між клієнтами. Відповідь на всі ці питання має давати прийнята в системі модель доступу та політики безпеки. В UWN усі додатки використовують однакову модель доступу, основний принцип якої говорить про неможливість встановлення прямого доступу до даних та про необхідність використання проміжних спеціалізованих інтерфейсів.

Детально розглянемо механізм взаємодії клієнтського додатку з серверною частиною UWN в рамках цієї моделі [6] (опис наведено в термінах нового інтерфейсу ***get/set***, модель доступу старого інтерфейсу ***get*** дещо відрізняється).

1. На першому кроці для встановлення з'єднання з серверною частиною UWN будь-який клієнтський додаток має використовувати технологічний аккаунт ***ua_guest***. Вхід в систему під цим аккаунтом надає додатку право доступу до схеми ***ua_security*** для подальшої ідентифікації.
2. На другому кроці додаток має провести виклик процедур аутентифікації схеми ***security*** та пройти реєстрацію в системі за допомогою модуля **security_pkg**. Після успішного проходження реєстрації клієнт отримує права на роботу з об'єктами UWN у відповідності до профілю своєї системи-додатку, також відбувається внесення записів до журналу входу та журналу виконаних команд (якщо логування поведінки даної системи задано системними параметрами).
3. За умови успішно виконаного другого кроку система-додаток отримує права на виконання певної підмножини команд модуля **run**. Оскільки модуль **run** представляє собою інтерфейс доступу до внутрішніх об'єктів UWN, то це означає що додаток отримує права на виклик процедур та функцій з інших схем системи: ***ua_ontology, en_ontology, ua_alg, ua_morphology***, та інших.
4. На четвертому кроці відбувається безпосередня робота додатку з системою, що може складатися як з виклику окремих лінгвістичних функцій, та і з взаємодії з пакетами серверної логіки (наприклад, в так UWN організовано системні утиліти).

Інструменти обробки та редагування знань

Важливим моментом при створенні онтології є забезпечення доступу до її вмісту. Мало розробити структуру онтології та програмно її реалізувати, потрібно ще створити набір додатків для забезпечення взаємодії користувача з онтологією та надати йому засоби перегляду даних в базі. Крім програмних засобів для роботи зі знаннями онтології необхідно також передбачити можливості прямого доступу до даних, причому такі можливості мають закладатися ще на етапі проектування онтології. Так, наприклад, вся база знань WordNet зберігається у текстовому вигляді і доступна для перегляду та редагування в звичайному текстовому редакторі.

При проектуванні UWN ми також ставили перед собою задачу забезпечення доступу до даних з використанням простих утиліт. Структура системи є досить складною і розподіленою в мережі, що унеможливлює побудову доступ до даних через текстовий редактор. Тому, в якості засобу забезпечення прямого доступу до даних онтології обрана мова SQL, а в якості програми для зв'язку з базою може бути використано будь-який типовий додаток для роботи з СУБД. Також існує можливість доступу до онтології без використання десктопних додатків – через веб-інтерфейс з сайту проекту [7].

Корекція наявних даних. З метою забезпечення користувачів можливістю редагувати, перевіряти якість наповнення україномовної онтології, порівнювати зміст україномовних синсетів з їх англомовними відповідниками було створено утиліту **UWNCorrector**. Утиліта представляє собою підсистему-онтокоректор для україномовного вмісту UWN. Метою підсистеми є надання користувачам можливості корекції невірно заданих слів та глосаріїв синсетів україномовної онтології та можливість позначати певні елементи синсету (глосарії, слова) спеціальними ознаками. Також додаток має вбудовану систему перевірки орфографії, що заснована на морфологічній базі UWN. Підсистема дозволяє обмежити доступ користувачів до даних шляхом конфігурації профілів користувача, причому конфігурація здійснюється на рівні БД, а не на рівні клієнта. Робота з UWN виконується через клієнтський додаток відповідно до встановленого користувацького профілю.

Задачі, що вирішуються в рамках системи **UWNCorrector**.

1. Надання користувачам інтерфейсу для переходу між синсетами україномовної онтології.
2. Надання інформації україномовного синсету: глосарію та списку слів.
3. Надання інформації англомовного синсету: глосарію та списку слів.
4. Надання можливості зміни слів україномовного синсету (редагування, видалення, створення нових) з подальшим збереженням змін в БД.
5. Надання можливості корегування глосарію синсету з подальшим збереженням змін в БД.
6. Надання можливості проставлення відміток коректності для глосарію та слів синсету з подальшим збереженням в БД.
7. Перевірка орфографії україномовних елементів глосаріїв та слів.
8. Забезпечення користувачам рівня доступу у відповідності до їх профілю (Читач, Редактор, Модератор).
9. Підтримка одночасної роботи багатьох користувачів.

Дана утиліта становить собою клієнт-серверний додаток в якому клієнтська частина реалізована на Java, а серверна частина розгорнута у вигляді окремого модуля в UWN. Основна ціль системи – забезпечити користувачам можливість роботи з україномовними онтологічними знаннями UWN. Для виконання цієї цілі система надає користувачу можливість переходу між синсетами що мають помилки (для Редакторів), що не мають помилок (для Модераторів), між будь-якими синсетами (для Читачів). Загальний вигляд системи показано на рис. 3. На момент написання статті використовується версія програми 1.55.2, яка доступна для завантаження з сайту проекту [7].

Засоби візуалізації та редагування онтології. Утиліта **UWNEditor** призначена для виконання широкого спектру робіт з україномовною онтологією. Основними з них є: пошук синсетів за ключовим словом, побудова семантичних дерев на основі структури онтології та обраного типу зв'язку, візуальне представлення елементів онтології в графічному вигляді з можливістю переміщення та приховання елементів (зв'язків та синсетів), пошук та графічне представлення зв'язаних синсетів, пошук та графічне представлення найкоротших шляхів між довільними синсетами, редагування україномовних синсетів, редагування семантичних зв'язків в україномовній онтології, перегляд англомовних відповідників україномовних синсетів та інші операції. Утиліта представляє собою підсистему-редактор для україномовного вмісту UWN. Основною ціллю підсистеми є надання користувачам можливості перегляду в зручному та наочному вигляді інформації, а також корегування та розширення існуючої онтологічної бази української мови. Система має зручний, простий та легкий у використанні інтерфейс. Не зважаючи та широкі можливості по роботі з онтологією робота з системою є інтуїтивно зрозумілою і не вимагає попередньої підготовки користувача. Як і в попередньому додатку, дана підсистема дозволяє обмежити доступ користувачів до даних шляхом конфігурації профілю, що здійснюється на рівні БД. Вся робота з UWN виконується через клієнтський додаток відповідно до встановленого користувацького профілю.

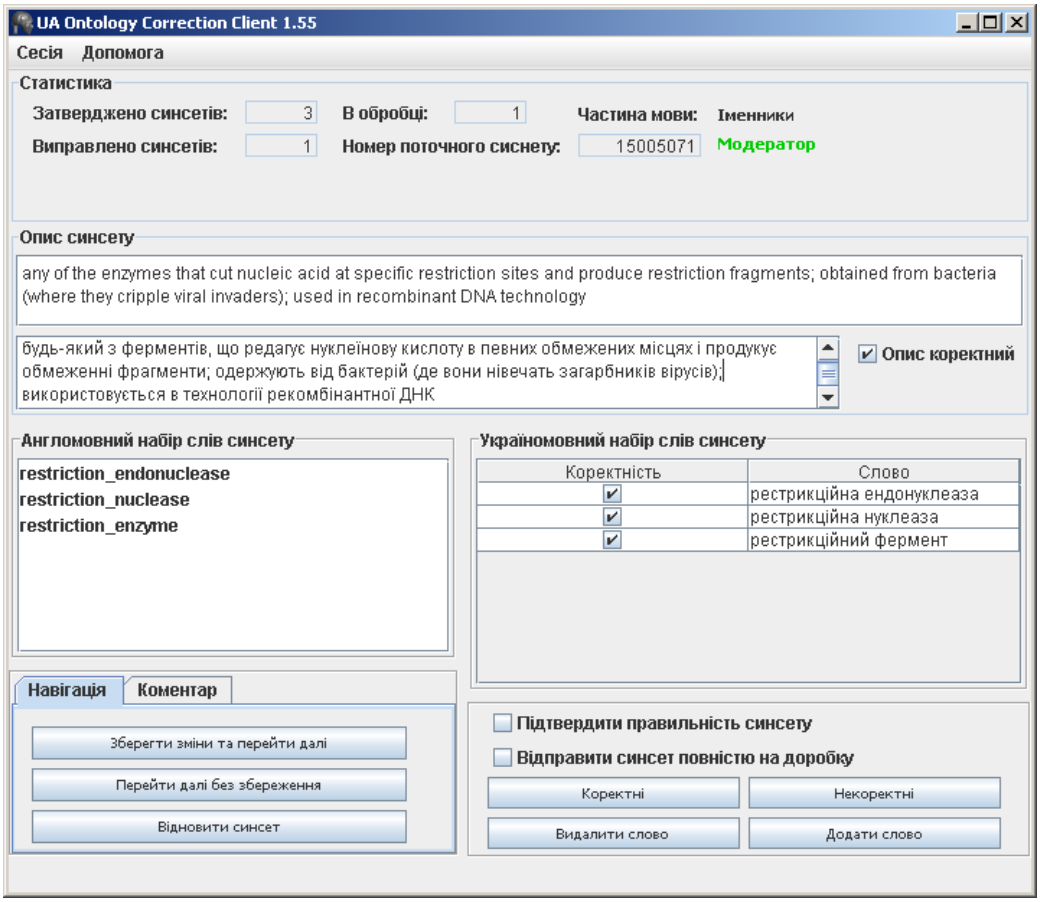

Рис. 3. Загальний вигляд утиліти UWNCorrector

Задачі, що вирішуються в рамках системи **UWNEditor**.

1. Видача інформації з онтологічної бази про синсети та їх зв'язки.
2. Представлення інформації у графічному та текстовому вигляді.
3. Організація пошуку синсетів за ключовим словом.
4. Організація пошуку зв'язаних синсетів.
5. Забезпечення можливості редагування та збереження інформації онтології.
6. Забезпечення користувача інструментарієм роботи над даними, відповідно до його профілю.
7. Підтримка одночасної роботи багатьох користувачів.

Система реалізована на базі клієнт-серверної архітектури, де клієнтська частина реалізована на Java, а серверна частина розгорнута у вигляді окремого модуля в UWN. Основна задача системи – забезпечити користувачам можливість роботи з онтологічними даними UWN. Для виконання цієї цілі система надає користувачу можливості перегляду синсетів та зв'язків між ними (для Читачів), а також можливості редагування цих даних (для Модераторів). Загальний вигляд системи показано на рис. 4.

Висновки

Описуються основні результати роботи команди фахівців з комп'ютерної лінгвістики кафедри математичної інформатики факультету кібернетики Київського національного університету імені Тараса Шевченка. Робота по створенню єдиної платформи призначеної для розробки прикладних лінгвістичних додатків ведеться з 2009 року і на даний момент наближається до фінальної стадії. Наразі всі інфраструктурні питання вирішено, система успішно функціонує в багатокористувацькому режимі і забезпечує користувачів інструментами для роботи з україномовною онтологією. Процес наповнення онтології триває, активно використовуються описані в статті онтокоректор та онторедактор, на даний момент онтологія містить близько 80000 концептів. Паралельно на базі UWN іде розробка прикладних систем для вирішення прикладних задач комп'ютерної лінгвістики.

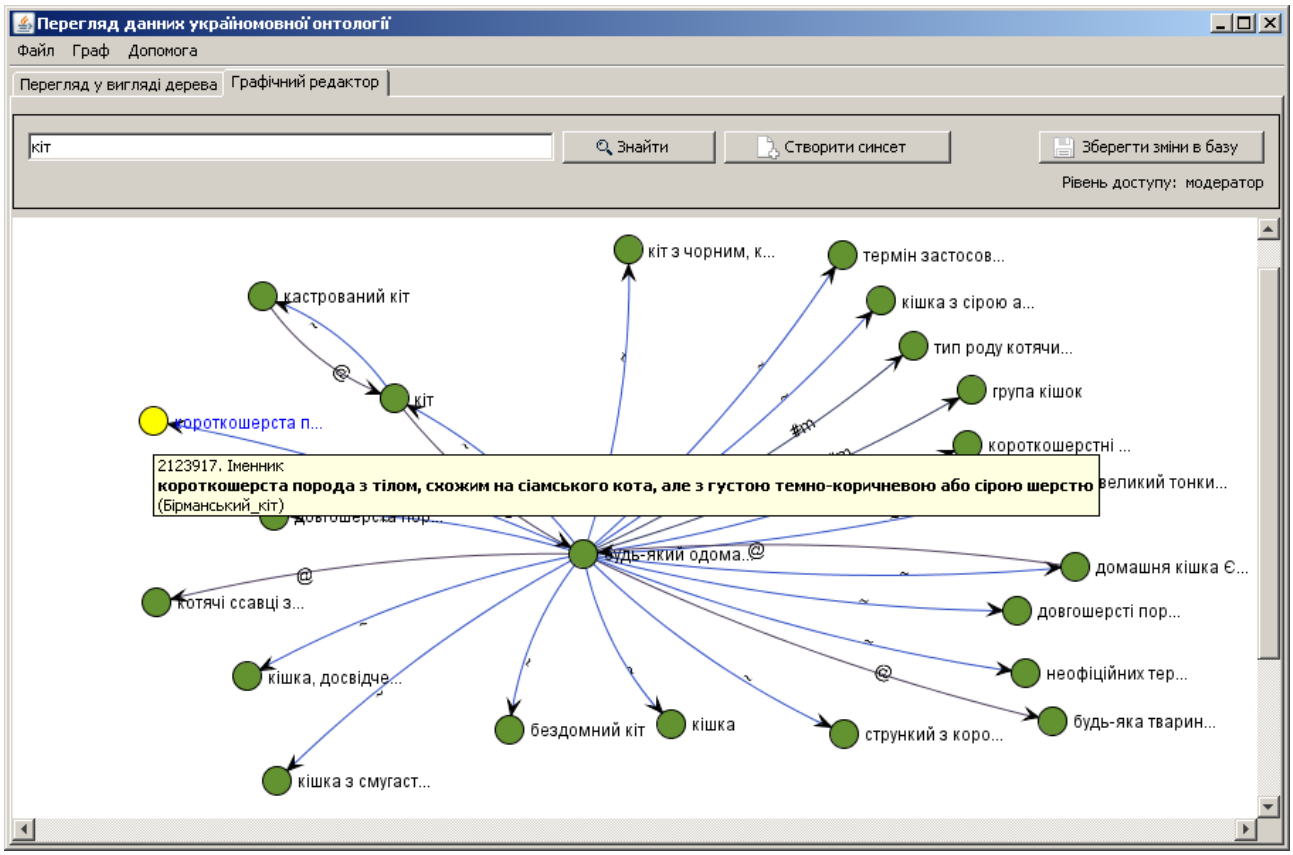

Рис. 4. Загальний вигляд утиліти UWNEditor

1. *George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller.* Introduction to WordNet: An On-line Lexical Database // http://wordnetcode.princeton.edu/5papers.pdf
2. *Piek Vossen, Pedro Diez-Orzas, Wim Peters*. Multilingual design of EuroWordNet // http://acl.ldc.upenn.edu/W/W97/W97-0801.pdf
3. *Oracle* Technology Network Developer License // http://www.oracle.com/technetwork/licenses/xe-license-152020.html
4. *Никоненко А.А.* Проект UWN: публичные интерфейсы для создания приложений сторонними разработчиками // Тезисы Международной научно-практической конференции «Информационные технологии и информационная безопасность в науке, технике и образовании "ИНФОТЕХ - 2011"». 5─10 сентября 2011, Севастополь, Крым, Украина ─ С. 33─34.
5. *Никоненко А.А.* Обзор баз знаний онтологического типа // Штучний інтелект. ─ 2009. ─ № 4. ─ С. 208─219.
6. *Никоненко А.А.* Проект UWN: Платформа для ускоренной разработки лингвистических приложений // Тезисы 21-ой Международной Крымской конференции "СВЧ-техника и телекоммуникационные технологии" (КрыМиКо' 2011), Севастополь, Крым, Украина, 2011 – С. 63─64.
7. *Сайт* проекту UWN // http://lingvoworks.org.ua