

pub.org/jlsc/vol1/iss1/3/; 12. Rebecca Kennison Point & Counterpoint: The Purpose of Institutional Repositories: Green OA or Beyond? [Електронний ресурс] / Rebecca Kennison, Sarah L. Shreeves, Stevan Harnad // Journal of Librarianship and Scholarly Communication. – 2012. – Vol. 1, № 4. – Режим доступу: <http://jlsc-pub.org/jlsc/vol1/iss4/5/>; 13. Simons N. New Roles, New Responsibilities: Examining Training Needs of Repository Staff [Електронний ресурс] / Natasha Simons, Joanna Richardson // Journal of Librarianship and Scholarly Communication. – 2012. – Vol. 1, № 2. – Режим доступу: <http://jlsc-pub.org/jlsc/vol1/iss2/7/>; 14. Thomas L. Reinsfelder Open Access Publishing Practices in a Complex Environment: Conditions, Barriers, and Bases of Power [Електронний ресурс] / Thomas L. Reinsfelder // Journal of Librarianship and Scholarly Communication. – 2012. – Vol. 1, № 1. – Режим доступу: <http://jlsc-pub.org/jlsc/vol1/iss1/10/>; 15. Башун О. Бібліотеки США. Погляд українського фахівця / О. Башун. – К.: Нора друк, 2004. – 55 с.; 16. Копанєва В. Бібліотека як центр збереження інформаційних ресурсів Інтернету: [монографія] / НАН України; Нац. б-ка України ім. В. І. Вернадського. – К., 2009. – 198 с.; 17. Ляшенко Л. В. Вивчення, забезпечення та розвиток інформаційних потреб бібліотекарів в умовах формування інформаційного суспільства: автореф. дис. на здобуття наук. ступеня канд. пед. наук: спец. 07.00.08 “Книгознавство, бібліотекознавство, бібліографознавство” / Л. В. Ляшенко. – К., 2002. – 22 с.

УДК 004.89

В.В. Литвин, С.Б. Хрущ\*

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж,

\*кафедра прикладної лінгвістики

## ПІДХІД ДО АВТОМАТИЧНОЇ ПОБУДОВИ ФУНКЦІЙ ІНТЕРПРЕТАЦІЇ ПІД ЧАС НАВЧАННЯ ОНТОЛОГІЙ

© Литвин В.В., Хрущ С.Б., 2014

Розглянуто підхід до автоматизованої побудови функцій інтерпретації (аксіом) концептів та відношень онтологій предметної області під час її навчання. Розроблено метод розпізнавання семантики природномовних текстів та її подання у вигляді описової логіки.

**Ключові слова:** онтологія, навчання онтологій, інтелектуальна система, база знань, функції інтерпретації, описова логіка.

**In the paper the approach to automated construction of interpretation functions (axioms) of concepts and relations of the domain ontology during its training are considered. The method of recognizing the semantics of natural language texts and its representation in the form of descriptive logic.**

**Key words:** ontology, learning ontologies, intelligent system, knowledge base, interpretation, descriptive logic.

### Вступ. Постановка проблеми у загальному вигляді

Формування галузевих предметних онтологій та засобів їх навчання (розроблення методів автоматизованої розбудови онтологій, перевірки їх достовірності та несуперечливості) – актуальний сучасний напрям наукових досліджень побудови інтелектуальних систем, зокрема інтелектуальних пошукових систем [1].

Онтологія – це знання, формально відображені на базі концептуалізації. Концептуалізація – опис множини об'єктів і понять, знань про них і зв'язків між ними. Формально онтологія складається з термінів (понять, концептів), організованих у таксономію, їхніх визначень і атрибутів, а також пов'язаних з ними аксіом і правил виведення. Онтологія визначає загальний словник для

користувачів, які спільно використовують інформацію про деяку предметну область (ПО). Враховуючи вищенаведене, під формальною моделлю онтології  $O$  розуміють трійку такого вигляду:

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

де  $C$  – скінченна множина понять (концептів, термінів) ПО, яку задає онтологія  $O$ ;  $R: C \rightarrow C$  – скінченна множина відношень між концептами (поняттями, термінами) заданої ПО;  $F$  – скінченна множина функцій інтерпретації (аксіоматизація, обмеження), заданих на концептах чи відношеннях онтології  $O$  [2].

Онтологію можна подати у вигляді графу, де вершини графу – концепти ПО, дуги вказують напрями відношень між концептами. Приклад такого графу наведено на рис. 1. Вершини можуть бути як інтерпретованими (визначені аксіоми понять), так й неінтерпретованими. Інтерпретовані вершини позначені темним кольором. Своєю чергою, відношення діляться на вертикальні (суцільні лінії) та горизонтальні (штрихпунктирні лінії).

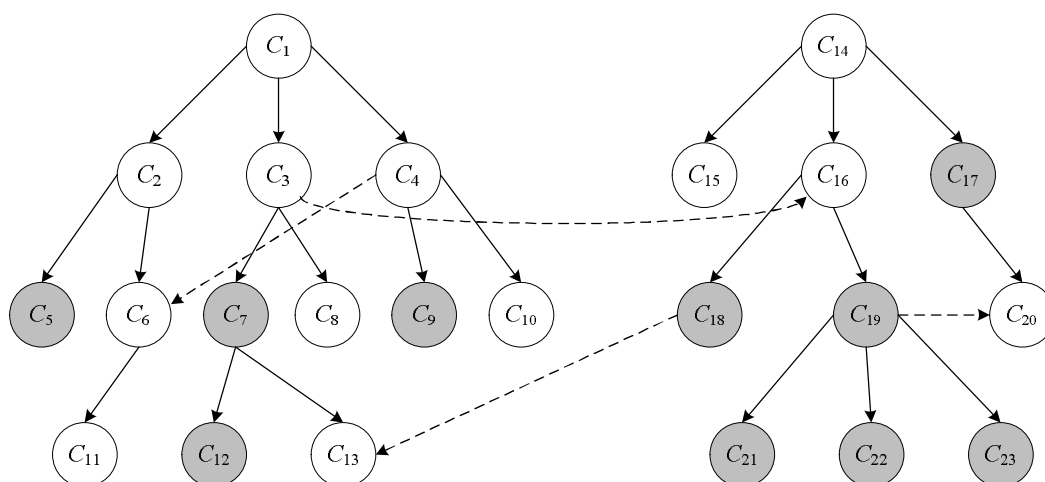


Рис. 1. Приклад графу онтології

У структуру онтології входить скінченна множина функцій інтерпретації (набір аксіом), однак автоматизовано побудувати таку множину функцій дуже складно. Тому ця задача все ще не розв’язана. У цій статті ми пропонуємо один із підходів до автоматизації побудови таких функцій інтерпретації.

### Аналіз останніх досліджень та публікацій

Відомі такі методи автоматичної побудови (навчання) онтологій:

#### 1. Метод Агіра

Цей метод призначений для збагачення великих онтологій за допомогою тексту, що видобувається з мережі Інтернет. Метою цього методу є подолання недоліків великої онтології: відсутність актуальних зв’язків між поняттями, різні смислові значення одного терміна тощо. Спершу інформація, що міститься в онтології, використовується для побудови запитів. Для кожного поняття з Інтернету отримують документи, з яких видобуваються тексти, що формуються в колекції. Кожна колекція порівнюється з іншими. Ці колекції документів використовують для побудови тематичних ознак та для побудови ієрархічних кластерів понять. В результаті формуємо ознаки тем, які мають збігатися із нашими запитамі [3].

#### 2. Метод Альфонсека і Манандара

Це метод автоматизованого набуття контексту слів, які вживаються спільно із кожним з множини понять. Значення слова змінюється залежно від контексту, в якому воно вжито. Метод описує алгоритм класифікації для розширення онтології. Ознаки тем можна покращити, ввівши лише ті контекстні слова, які мають синтаксичні зв’язки з поняттями в онтології. Дивимось, чи є в тексті слова із зв’язки, що вже є в онтології, і додаємо ті слова, яких нема. В такий спосіб шукаємо нові поняття і порівнюємо, чи вони нам підходять [4].

### **3. Метод Наталі Асенак-Гілз**

Метод дає змогу створити модель предметної області за допомогою аналізу корпусу, використовуючи обробку природної мови та лінгвістичні методи. Центральну роль у цьому методі відіграють технічні тексти. Метод поєднує в собі інструменти видобування знань, оснований на методах моделювання, що дозволяє підтримувати зв'язок між моделями і текстами. Метод використовує тексти і може використовувати інші наявні онтології або термінологічні ресурси для створення онтології. Цей метод оснований на видобуванні термінів на основі дистрибутивного аналізу (розподіл окремих одиниць у тексті). Текст для корпусу вибирають дуже ретельно, він має чітко відповідати вимогам користувача [5].

### **4. Метод Бакімонта**

Цей метод запропоновано для побудови онтологій з урахуванням мовних прийомів.

Користувач повинен вибрати відповідні терміни з ПО та нормалізувати їх значення, виражаючи схожість і відмінності кожного поняття щодо своїх сусідів (з наявними зв'язками). Потім необхідно усунути неоднозначність понять і уточнити їх значення для експертів для проведення формалізації знань. Отже, користувач може визначити нові поняття і зв'язки, додати їх властивості та загальні аксіомы. І наприкінці слід переписати онтологію на певну мову представлення знань. Недолік цього методу – користувач робить все вручну [6].

### **5. Метод Фетза та Стрейнметза**

Фетз і Стрейнметз представляють підхід, який наповнює наявну онтологію, видобуваючи поняття з Інтернету. Процес наповнення оснований на порівнянні між статистичною інформацією слововживання і структурою самої онтології. Кожне поняття в онтології повинно мати одне або більше слів чи фраз на природній мові, пов'язаних з ним. Метод пропонує розрахунок семантичної відстані між словами, щоб збагатити значення понять і створити кластери слів, пов'язаних з новими поняттями. Нові поняття будуть запропоновані експертів з певної предметної області, який вирішить, чи додавати їх до онтології [7].

### **6. Метод Гана**

Ган і його колеги пропонують метод для підтримки предметно-орієнтованих таксономій, побудованих на природній мові. Дані таксономії постійно оновлюються, як нові поняття, що взяті із реальних текстів.

Дії, що пропонуються у цьому методі, описують такими кроками:

1. Мова обробки (визначає структурну залежність інформації від граматичних конструкцій, у яких невідомі лексичні одиниці трапляються у термінах відповідного синтаксичного дерева).
2. Визначення знаків якості: (1) лінгвістичний, що відображає структурні властивості патернів; (2) концептуальний, що впливає з порівняння структур гіпотез).
3. Оцінка якості (отримання доступного набору знаків якостей для кожної гіпотези. Результат являє собою класифікований список понять гіпотез).
4. Аналіз (емпірично оцінюється процес набуття знань з тексту за допомогою різних методів, які оцінюють точність і швидкість навчання) [8].

Однак жодний із сучасних методів навчання онтологій автоматично не буде функцій інтерпретації.

### **Формування цілей**

Розробити метод автоматизованої побудови функцій інтерпретації (аксіом) концептів та відношень онтологій предметної області під час її навчання.

### **Основний матеріал**

Для автоматизації процесу побудови онтологій застосовують кілька основних підходів до опрацювання тексту – символний, статистичний та змішаний. Серед найпоширеніших символних підходів – застосування лексико-семантичних патернів (lexico-semantic pattern – LSP) [9]. За таким підходом опрацювання тексту виконують, виявляючи певні наперед відомі або встановлені шляхом машинного навчання реляційні маркери, які існують у природній мові і дозволяють розпізнати семантичні ролі синтаксичних конструкцій, а у поєднанні з ідентифікацією онтологічних сутностей,

які у цьому тексті представляють такі синтаксичні конструкції, виконувати проекцію тексту на онтологію, отримуючи розпізнаний зміст, а за ним – оцінювати новизну, достовірність і корисність отриманих за цим змістом знань. Методи, основані лише на статистичних лінгвістичних моделях, здатні тільки поверхнево розпізнавати дискурс, але не можуть виявляти зміст тексту, тобто відображену там логіку семантичного взаємозв'язку між поняттями конкретної проблемної області.

Знання набувають змісту лише в контексті певної ПО, заданої у цьому випадку її онтологією. Одержані з текстового документа нові знання набувають форми змін у первинній онтології, яку слід попередньо сформулювати вручну або із застосуванням процедур навчання. Аналіз кожного наступного тексту ґрунтується на застосуванні онтології, доповненої в процесі аналізу попередніх текстів у тій частині, яка стосувалася заданої проблемної області.

Корпус текстів для заданої ПО складається з окремих текстових документів (далі – текстів), кожен з яких містить від одного до 10–20 речень. Ці речення перебувають у послідовному логічному зв'язку, тому не можуть розглядатися відокремлено без втрати змісту. Саме тому текст розбивається на впорядковану множину речень, над якими надалі послідовно будуть виконуватись основні процедури розпізнавання. Складні речення розділяють на прості засобами синтаксичного аналізу. В процесі розділення виконується підстановка займенників замість іменників з першої частини речення, на які ці займенники посилаються. Підготовка речень виконується так, щоб надалі однозначно ідентифікувати усі поняття, задіяні в сформульованих у реченнях твердженнях, а також відрізнити узагальнені поняття, які будуть розпізнані як класи, та конкретні поняття (зокрема, власні назви), що є унікальними, а тому будуть розпізнані як екземпляри відповідних класів. Кожне з таких декомпонованих і формалізованих речень слугуватиме матеріалом, вхідними даними для побудови на його основі концептуального графу [10] і/або предиката, який відображає логічний взаємозв'язок між поняттями цієї ПО. Тобто попереднє опрацювання тексту складається з таких кроків:

1. Розбиття тексту на окремі речення розпізнаванням розділювачів “крапка”, “кінець абзацу”, “кінець тексту”.

2. Поділ складних речень на прості із застосуванням синтаксичного аналізу з одночасною заміною займенників наступних речень відповідними їм іменниками з попередніх.

3. Переведення букв тексту в нижній регістр з маркуванням слів всередині речення, відсутніх у словнику загальних назв понять, як власних назв. Вилучення зайвих розділових знаків.

4. Занесення отриманих текстів до масиву речень цього документа.

Результатом роботи процедури попереднього опрацювання тексту є масив послідовно логічно пов'язаних формалізованих та очищених речень, готових для опрацювання семантичним парсером з метою виділення ознак семантичних зв'язків у цих реченнях.

Природномовний текст необхідно перетворити на послідовність предикатів логіки першого порядку. Цю задачу можна розкласти на кілька підзадач:

- розпізнавання іменникових груп;
- визначення їх ролі суб'єкта чи об'єкта семантичного зв'язку та відповідного цій ролі місця у предикаті;
- розпізнавання дієслівної групи кожного речення як типу семантичного зв'язку і відповідного цьому типу зв'язку предиката.

Розпізнавання іменникових груп виконується з виділенням базового іменника і послідовним уточненням (звуженням) його значення внаслідок додавання до нього описових іменників та прикметників з виділеної іменникової групи. Відповідність певного іменника поняттю в онтології визначається за написанням цього слова після його стемінгу (нормалізації). Послідовне уточнення значення іменникової групи здійснюється від останнього іменника (усталеного словосполучення іменникового типу) в групі до першого доти, доки в онтології буде відповідне такому поєднанню слів поняття. Усі наступні поєднання іменників і прикметників у іменниковій групі відображаються в онтології як підкласи понять, які були впізнані у іменниковому словосполученні на попередньому кроці – до додавання до цього словосполучення нового уточнюючого іменника або прикметника. Наприклад, якщо в онтології є поняття `garden`, але немає `apple tree garden`, хоча є або поняття `apple`

tree, або лише окремі іменники tree та apple, система вносить до онтології уточнюючим підкласом класу garden одразу поняття apple tree garden або лише tree garden відповідно у кожному з двох випадків. При цьому у другому випадку вже на наступному кроці до онтології буде внесено поняття apple tree garden як підклас класу tree garden (див. рис. 2).

Визначення ролі розпізнаного поняття у семантичному зв'язку у найпростішому випадку здійснюється на підставі місця, яке займає відповідна іменникова група відносно групи дієслова у реченні та залежно від виду семантичного зв'язку, який представлений цією дієслівною групою. Якщо іменникова група стоїть перед дієслівною, основне дієслово якої закінчується на “-ed + by”, ця іменникова група є об'єктом семантичного зв'язку. В інших випадках – суб'єктом цього зв'язку.

Для формального подання природномовного речення у термінах онтології та описової (дескриптивної) логіки предикатів 1-го порядку необхідно визначити тип предиката. Розпізнавання типу можна виконати за дієслівною групою цього речення та службовими словами, які до дієслівної групи можуть не входити. Для цього необхідно застосувати метод машинного навчання системи розпізнавання, вхідними даними для якої слугуватимуть результати розбору природномовного речення спеціальним синтаксично-семантичним парсером. Такий парсер розділяє речення на пари слів, пов'язані деяким метасемантичним зв'язком. В результаті кожне речення парсер представляє множиною триплетів, що складаються із суб'єкта такого зв'язку, об'єкта зв'язку і самого метасемантичного зв'язку певного виду. Ці триплети можуть бути використані як ознаки наявності в реченні того чи іншого семантичного зв'язку, на основі якого має бути збудований предикат як логічне формальне представлення цього речення.

У спрощеному викладі суть алгоритму виглядає так: шукаємо перше відоме дієслово (LSP дає відповідну розмітку) і найближчий іменник, що передує цьому дієслову. За англійською граматику у переважній більшості випадків (винятки згадано вище у цьому параграфі) цей іменник (іменникова група) і є підметом – суб'єктом дії. Якщо це займенник, шукаємо у попередньому реченні підмет за аналогічною процедурою і підставляємо у реченні, що аналізується, на місце займенника. Дія і є семантичним зв'язком (видом предиката), проте таким зв'язком може виступати не саме дієслово а ціла дієслівна група в реченні, тому її належить виявити і зафіксувати в онтології одним терміном на зразок "is-a" чи "is-a-part-of".

Після такого навчання сигнатура (формула) семантичного зв'язку вноситься до онтології для подальшого використання та можливого уточнення. Для цього в лінгвістичній частині онтології зберігаються екземпляри класів: 1) метасемантичних зв'язків; 2) об'єктів та 3) суб'єктів цих зв'язків, які додаються в міру машинного навчання онтології кожному з семантичних зв'язків як деякі патерни і враховуються під час розрахунку імовірності виявлення котрогось із зв'язків у тексті, що аналізується.

Розроблено цілу низку ефективних інструментів опрацювання ПМ текстових документів, проте методи комп'ютерної лінгвістики, штучного інтелекту, інформаційних технологій, що існують нині, загалом досі не мають ефективних моделей текстів природними мовами, придатних для автоматичної числової оцінки їх потрібності (корисності) потенційному споживачу.

Під час виконання роботи ми використовували середовище планування та інженерії знань загального призначення без прив'язки до конкретної предметної області, яке забезпечує побудову систем планування, базованих на OWL-онтологіях, та інтегрування процесу планування з логічним виведенням, оснований на описовій логіці (description logic – DL) [5-6].

Стан системи описується множиною OWL-фактів. Вони представлені як RDF-граф [11]. Дії описуються через трансформації RDF-графів. Ці факти та дії використовуються для здійснення логічного виведення (планування) інтелектуальною системою на основі описової (DL) логіки за двоетапним алгоритмом: на першому етапі в режимі "off-line" планувальник транслює DL-описи дій

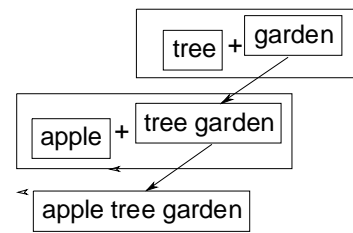


Рис. 2. Уточнення понять за складовими іменникової групи

на мову SPPL, в процесі чого виконує необхідне логічне виведення і додає необхідні факти до множини результатів планованих дій. На другому етапі використовується сам планувальник SPPL.

В описовій логіці використовують два типи фактів: термінологічні – *TBox* та факти предметної області – *ABox*.

Множина RDF-термів (RDFT) містить множини URI ( $U$ ) та множини RDF-літералів (RDFL)

RDF-триплет є елементом множини  $RDFT \times U \times RDFT$

RDF-граф є множиною RDF-триплетів. Вузли графа є суб'єктами і об'єктами, а ребра позначаються як властивості.

Застосована в роботі система підтримує підмножину OWL-DL, яку називають Description Logic Programs (DLP) [5]. DLP є перетином DL та Horn Logic Programs.

Модель світу подається RDF-графом, який містить триплети, що відображають OWL *ABox* факти.

Модель дії базується на ідеї, що дія має спричинити зміну стану світу. Передумови та результати моделюються як патерни RDF-графу. Дія виражається через трансформацію RDF-графу.

Унарний предикат (Пристрій ?A1) на RDF-графі виражається через вузол екземпляра ?A1, зв'язаний ребром "типу" з вузлом класу Пристрій.

Бінарний предикат відображається на RDF-графі через два вузли екземплярів, з'єднаних ребром властивості.

Формально опишемо модель дії:

Змінна є елементом множини  $V$ , де  $V$  – необмежена і несумісна з RDFT. Триплетний патерн є елементом множини  $(RDFT \ U \ V) \times U \times (RDFT \ U \ V)$ . Патерн графа – це множина триплетних патернів. Дія має форму  $A(P, E, C, Q)$ , де:

$P$  – патерн RDF-графу, що виражає передумови дії;

$E$  – патерн RDF-графу, що виражає результати дії.

Множина змінних в  $P$  є підмножиною множини змінних у  $E$ . Це гарантує відсутність невизначених змінних у результуючому описі.

Дія може бути застосована, якщо у цьому стані виконуються всі передумови. Нехай поточним є стан  $G$ . Визначимо, що  $P$  можна застосовувати до  $G$ , базуючись на онтології  $O$ , тоді і тільки тоді, коли існує функція заміни змінних ( $q: V \rightarrow RDFT$ ), визначена для всіх змінних у  $P$  так, що:

$$G \cup O \models q(P).$$

Результат застосування дії описується як результат трансформації RDF-графів. Нехай  $L$  та  $R$  – передумови та результати дії відповідно,  $L$  задовольняє поточний стан, описаний RDF-графом  $X$ ,  $q$  – функція заміни змінних в  $L$ . Нехай також наступний після застосування дії стан описується RDF-графом  $Y$ . Кожний  $Y$  визначається із застосуванням гомоморфізму графів  $f$ , описаного так:

$$f: q(L) \ U \ q(R) \rightarrow X \ U \ Y, \text{ де } f \text{ задовольняє такі властивості:}$$

$$1. f(q(L)) \subseteq X,$$

$$2. f(q(R)) \subseteq Y,$$

$$3. f(q(L) \setminus q(R)) = X \setminus Y \text{ та } f(q(R) \setminus q(L)) = Y \setminus X.$$

Це значить, що саме та частина графу  $X$  вилучена, котра відповідає елементам  $q(L)$ , але якої немає в  $q(R)$ , а також саме та частина графу  $Y$  створюється, яка відповідає новим елементам в  $q(R)$ .

Застосовуючи властивості 2 і 3, можна визначити наступний стан  $Y$  як результат застосування дії до  $X$ . Операція виконується за два кроки. На першому кроці усуваємо всі вузли і ребра з  $X$ , які відповідають  $q(L) \setminus q(R)$ , щоб отримати граф  $D$ , для якого  $D = X \setminus (q(L) \setminus q(R))$ . Впевнюємося, що  $D$  – коректний граф, тобто не має висячих вузлів після видалення вихідних чи вхідних ребер. На другому кроці ми склеюємо  $D$  з  $R \setminus L$ , щоб отримати  $Y$ .

Програмний засіб Protégé-OWL містить вбудовані предикати DL, які наведено на рис. 3.

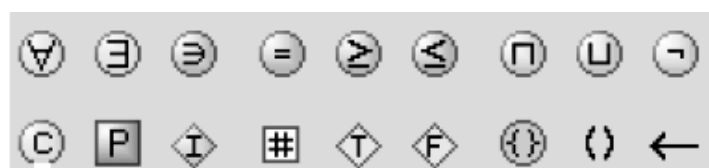


Рис. 3. Вбудовані предикати DL у ПЗ Protégé-OWL

Окремим з цих предикатів відповідають такі слова англійської мови

∀	any, anybody, anyone, each, whichever, whichever, either, all, which
∃	exist, there, one, only-begotten
ε	is, belongs, behove, effer, owned
=	equal, level, is

Подальшою нашою роботою є формалізація методу побудови функцій інтерпретації з метою повноцінної автоматизації процесу навчання онтологій.

### Висновки

Проаналізовано стан досліджень та розробок у галузі видобування знань з природномовних текстів та машинного навчання онтологій. Обґрунтовано необхідність автоматизації побудови функцій інтерпретації концептів та відношень онтологій. Запропоновано метод виділення предикатів описової логіки (DL) з природномовного тексту, показано, що такий алгоритм має бути багатоступінчастим і містити ієрархічну кількарівневу процедуру розпізнавання понять, зв'язків, предикатів та правил, які в результаті вносяться до онтології. Визначено слова англійської мови, яким відповідають окремі вбудовані предикати програмного засобу Protégé-OWL.

1. Литвин В.В. *Бази знань інтелектуальних систем підтримки прийняття рішень* / В.В.Литвин. – Львів: Видавництво Львівської політехніки, 2011. – 240 с. 2. *Інтелектуальні системи, базовані на онтологіях* // Д.Г. Досин, В.В. Литвин, Ю.В. Нікольський, В.В. Пасічник. – Львів: Цивілізація, 2009. – 414 с. 3. Agirre E. (2000). *Enriching very large ontologies using the WWW* / E.Agirre, O.Ansa, E.Hovy, D.Martinez // *In Proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00)*. – 2000. – P. 347–349. 4. Alfonseca E. *Extending a Lexical Ontology by a Combination of Distributional Semantics Signatures* / E.Alfonseca, S.Manandhar // *EKAW-2002, Siguenza, Spain. Published in Lecture Notes in Artificial Intelligence*. – 2002. – P. 2473 (Springer Verlag). 5. Aussenac-Gilles N. *Corpus Analysis For Conceptual Modelling* / N.Aussenac-Gilles, B.Biebow, S.Szulman // *Workshop on Ontologies and Text, Knowledge Engineering and Knowledge Management: Methods, Models and Tools, 12th International Conference EKAW'2000, Juan-les-pins, France, Springer-Verlag*. – 2000. – P. 847-849. 6. Bachimont B. *Semantic commitment for designing ontologies: a proposal* / B.Bachimont, A.Isaac, R.Trancy // *In A. Gomez-Perez and V.R. Benjamins (Eds.): EKAW 2002, LNAI 2473*. – P. 114–121. 7. Faatz A. *Ontology enrichment with texts from the WWW* / A.Faatz, R.Steinmetz // *Semantic Web Mining 2nd Workshop at ECML/PKDD-2002, 20th August 2002, Helsinki, Finland*. – P. 217-219. 8. Hahn U. *Towards text knowledge engineering* / U.Hahn, K.Schnattinger // *In: AAAI'98-IAAI'98 Proceedings of the 15th National Conference on Artificial Intelligence & 10th Conference on Innovative Applications of Artificial Intelligence. Madison, Wisconsin, July 26-30, 1998*. – P. 524–531. 9. Hearst M.A. *Automatic acquisition of hyponyms from large text corpora* / M.A. Hearst // *In: Proceedings of the 12th Conference on Computational Linguistics, 1992*. – P. 579-581. 10. Ganter B. *Conceptual Structures: Logical, Linguistic, and Computational Issues* / B.Ganter, W.Guy. – *Lecture Notes in AI 1867, Springer-Verlag, Berlin, 2000*. – 569 p. 11. Beckett, D. 2004. *Rdf/xml syntax specification [Електронний ресурс]* / D.Beckett. – Режим доступу: <http://www.w3.org/TR/rdf-syntax-grammar>.