

Science. LNCS 1665. – Springer-Verlag, 1999. – P. 122–134. 7. Stein B., Meyer zu Eissen S., Wißbrock F. On Cluster Validity and the Information Need of Users // *Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA'03)*. – Benalmadena, Spain: September 8-10, 2003. – P. 216–221. 8. Wassenberg J., Middelman W., Sanders P. An Efficient Parallel Algorithm for Graph-Based Image Segmentation // *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns (CAIP'09)*. – Münster, Germany: September 2–4, 2009. – P. 1003–1010.

УДК 004.89

В.В. Литвин

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

ЗАДАЧІ ОПТИМІЗАЦІЇ СТРУКТУРИ ТА ЗМІСТУ ОНТОЛОГІЇ ТА МЕТОДИ ЇХ РОЗВ’ЯЗУВАННЯ

© Литвин В.В., 2011

Розглянуто оптимізаційні задачі, які виникають під час автоматичної розбудови онтології щодо її структури та змісту. Сформовано низку таких задач. Розглянуто методи та алгоритми їх розв’язування.

Ключові слова: онтологія, концепт, інтелектуальний агент, структура онтології, зміст онтології, оптимізаційна задача.

In the paper the evaluation of the optimization problems that arise during the automatic ontology building on its structure and content. Formed a series of such problems. The methods and algorithms for their solution.

Key words: ontology, concept, intelligent agent, ontology structure, ontology content, optimization task.

Вступ. Постановка проблеми у загальному вигляді

Автоматична розбудова онтології призводить до виникнення недоліків у її структурі та змісті, невідповідності її наповнення інформаційним потребам користувача [1, 2]. Тому подібні системи необхідно “укомплектовувати” відповідним набором процедур оптимізації онтології, тобто оптимізаційною компонентою. Актуальність задачі оптимізації внутрішнього подання баз знань (БЗ), що дає змогу істотно підвищити ефективність функціонування інтелектуальних систем, також зазначається в роботі Пузанкова [3]. Нижче розглянуто методику динамічної побудови онтології, а також процедури її оптимізації.

Складна структура взаємозв’язків між поняттями, відображеними в онтології, а також її динамічне наповнення під час експлуатації вимагає застосування певних оптимізаційних процедур з метою мінімізації часу реакції на запити; неперевищення місця, відведеного для розміщення онтології; вирішення конфліктів між даними, внесеними з різних джерел, а також задоволення інших вимог та критеріїв, які належить визначити. Оптимізація онтології також виконується з метою адаптації її змісту до інформаційних потреб користувачів, вилучаючи елементи, які рідко використовуються або ж не використовуються взагалі, тобто не належать до певної предметної області (ПО).

У роботах [4, 5] обґрунтовано доцільність подання онтології у вигляді концептуального графу, вершинам якого присвоєно певні семантичні та числові характеристики. Він є зваженим графом, в якому на етапі формування допускається існування паралельних ребер, циклів, петель, дублювання вершин з аналогічними параметрами та інших особливостей. Виявлення й усунення таких особливостей з метою нормалізації графу онтології, а також оптимізація графу за визначени-

ми критеріями з урахуванням вагових коефіцієнтів ребер та вершин мають входити до переліку обов'язкових процедур забезпечення ефективного функціонування інтелектуальних систем підтримки прийняття рішень (ІСППР).

У термінах теорії графів структурна оптимізація онтології (усунення конфліктів, збереження цілісності, дотримання обмежень на максимальний об'єм) полягає у чергуванні процедур доповнення та редукції графу онтології в межах заданого діапазону значень кількості вершин у разі максимізації суми коефіцієнтів важливості його вершин та ребер.

Під час функціонування ІСППР відбувається постійне наповнення її онтології новими поняттями, що, своєю чергою, вимагає періодичного прийняття рішення щодо вибору елементів для вилучення з неї (редукції графу) у разі збереження цілісності її семантичної структури. У зв'язку з цим під час внесення змін виникає потреба в корекції структури та змісту онтології.

З міркувань підвищення ефективності функціонування системи, загалом, процедуру оптимізації доцільно застосовувати не кожного разу, а періодично за досягнення моменту виконання певних умов (перевищення ліміту часу чи об'єму, доступного для заповнення), чергуючи етапи доповнення онтології та вивільнення надлишкових елементів під час оптимізації. Критерії, які потрібно обрати для визначення моментів переходу від режиму до режиму, потрібно дослідити окремо, ґрунтуючись на результатах чисельного моделювання. Отже, динамічне наповнення онтології та періодична оптимізація її структури і змісту забезпечують перебудову й удосконалення моделі ПО, яка відображається її онтологією.

Аналіз останніх досліджень та публікацій

Проект Гасті [6] являє собою систему автоматичної побудови онтології природомовними текстами. За цим підходом система розпочинає роботу з малого ядра онтології, що містить базові поняття, зв'язки та оператори, і продовжує конструювати онтологію автоматичним опрацюванням тексту, використовуючи символічні методи, а також так званий гібридний підхід, що складається з методів логічного, лінгвістичного, методу керування шаблоном і семантичного аналізу.

Особливістю проекту Гасті є наявність у складі системи процедур, завданням яких є періодичне уточнення та реорганізація онтології. Реорганізація полягає у кластеризації (об'єднання подібних понять або розщеплення поняття до відповідних підпонять), перенесенні спільних властивостей догори за таксономією, відсікання неважливих понять і зв'язків та виключення деяких надлишковостей. Процедура уточнення реалізується чотирма основними модулями: модуль “внесення за дужки”, модуль “відсікання”, модуль “об'єднання гілок” та модуль “відкидання надлишкових шляхів”.

Гнучкість щодо змін користувач/ПО під час побудови динамічної онтології пропонується забезпечувати присвоєнням певної важливості зв'язкам та пов'язаним термам. Важливість показує важливість зваженого елемента (поняття, зв'язку чи їх комбінацію) у ПО з погляду користувача системи. Автори зазначають необхідність автоматизації визначення важливості, проте сьогодні у системі Гасті це роблять вручну.

Переваги цієї моделі полягають у пристосовуваності онтології до потреб користувача і змін в ПО, охопленні широкого діапазону знань, а також у зменшенні часових та фінансових затрат на побудову онтології. До недоліків треба зарахувати сповільнення опрацювання запитів та труднощі навчання онтології, що зумовлені специфікою методу подання знань та природою навчання онтології “з нуля”. Відсутність у системі процедури автоматичного присвоєння важливості елементам онтології унеможливорює її повну автоматизацію.

Загалом огляд наявних підходів виявив невелику кількість розроблених та придатних для практичного застосування методів динамічної оптимізації онтології БЗ. Запропоновано новий підхід до виконання цього актуального науково-технічного завдання.

Формування цілей

Метою роботи є побудова оптимізаційних задач, розв'язання яких покращує структуру та зміст онтології ПО під час її автоматичної розбудови. *Об'єкт дослідження* – процес оптимізації

структури та змісту онтології. *Предмет дослідження* – методи та алгоритми розв'язування оптимізаційних задач покращення властивостей автоматично формованої онтології.

Основний матеріал

1. Вибір критеріїв оптимальності структури і змісту онтології бази знань

Прийmemo такі критерії оптимальності структури і змісту онтології:

- фізичний об'єм пам'яті, яку займає онтологія;
 - швидкодія як час відгуку ІСППР на зовнішнє звертання (час реакції на зміну параметрів зовнішнього середовища, до яких чутлива система);
 - повнота онтології, яка може бути визначена через середній відсоток нетривіальних (ненульових) відповідей на запити до неї;
 - цілісність онтології, тобто відсутність в її тілі взаємозаперечувальних тверджень і дублювання;
 - збалансованість ПО, виражена у рівномірному поданні її окремих підрозділів в онтології.
- Очевидно, для оптимізації онтологій БЗ прикладних ІСППР доцільно обирати не один критерій, а певне їх поєднання, причому вибір способу такого поєднання доводиться здійснювати дослідно, ґрунтуючись на реальних вимогах до системи.

1.1. Цілісність онтології бази знань

Здатність інтелектуальної системи приймати обґрунтовані рішення (давати відповіді) з питань, які ставлять перед нею розробники чи користувачі, передбачає наявність у системи онтології, яка б забезпечувала обґрунтованість таких рішень. Зокрема, логічність висновків, однакову відповідь на однакові, але по-різному сформульовані запитання. Така онтологія повинна задовольняти вимоги цілісності.

Поняття цілісності об'єднує ознаки чи вимоги, серед яких:

- контрольована надлишковість;
- зв'язність графу онтології;
- відсутність взаємозаперечувальних тверджень.

У системах, що ґрунтуються на знаннях, під час доповнення їхньої онтології може виникати надлишковість, яка полягає у присутності дублювальних структур: понять та тверджень, виражених через зв'язки між поняттями. Коли надлишкове знання необхідне, такі системи здійснюють контроль надлишковості.

Зв'язність графу онтології – властивість, яка означає, що між будь-якими двома вершинами такого графу існує простий ланцюг. Зв'язність свідчить, що всі елементи БЗ перебувають у межах досяжності інтелектуальної системи і можуть бути задіяні під час генерування відгуку на звертання до неї. Під час впорядкування та редукції онтології система має контролювати збереження умови зв'язності її графу і не допускати операцій, які цю умову порушують.

Перевірку на зв'язність графу можна здійснити за допомогою наслідків з теореми про оцінення кількості ребер через кількість вершин і кількість компонентів зв'язності [7]. Якщо позначити p та q – число вершин та ребер графу відповідно, то повинні виконуватись такі дві умови:

- 1) якщо $q > (p-1)(p-2)/2$, то граф зв'язний;
- 2) у зв'язному графі $p-1 \leq q \leq p(p-1)/2$.

Під час внесення змін – доповнення онтології новими елементами, модифікації, видалення елементів – система повинна виконувати перевірку на її цілісність, тобто відсутність в ній дублювальних і/або взаємозаперечувальних тверджень. Цю процедуру можна реалізувати через механізм виявлення протилежних за змістом відгуків тестовим їх порівнянням (зіставленням) у разі послідовного логічного інвертування одного з тверджень-відгуків методом резолюцій. У разі збігу прямого та інвертованого тверджень система отримує сигнал про порушення цілісності та необхідність усунення суперечності тверджень. У разі виявлення взаємосуперечливих тверджень, конфлікт усувається видаленням того з них, для якого важливість є нижчою.

У разі перевищення очікуваної за контекстом кількості відгуків, система генерує процедуру узагальнення понять, що дали такий відгук, або їх властивостей-опрацьовувачів з метою редукції онтології.

Базу знань вважатимемо внутрішньоузгодженою, якщо виконуються такі умови:

- 1) класи, екземпляри та їхні атрибути мають унікальні імена в межах області визначеності;
- 2) всі класи в таксономії зв'язані ієрархічними зв'язками "IS-A", на найвищому рівні таксономічної ієрархії міститься лише один базовий клас;
- 3) всі класи містять принаймні один об'єкт-екземпляр певного класу, що забезпечує функціональність фреймової моделі подання знань;
- 4) усі опрацьовувачі повідомлень понять онтології на момент їх виклику мають визначені атрибути (конкретні значення формальних параметрів).

Деякі ознаки і вимоги, які містить поняття цілісності (контрольована надлишковість, зв'язність графу, відсутність взаємозаперечувальних тверджень), враховані під час формулювання обмежень на структуру онтології та оптимізаційну задачу.

1.2. Обмеження на фізичний об'єм пам'яті

Система має бути реалізована на базі певного програмно-апаратного комплексу, для якого існує реальне обмеження на об'єм оперативної пам'яті. З іншого боку, надмірне зростання об'єму БЗ сповільнює її швидкодію, що може мати вирішальне значення у разі систем, які працюють у реальному масштабі часу.

На етапі первинного формування онтології ІСППР у момент її створення подібна проблема не виникає, проте під час експлуатації відведений об'єм фізичної пам'яті заповнюється, тому доводиться вдаватися до процедур вивільнення тієї її частини, яка використовується системою найменш ефективно. Отже, система працює поперемінно у двох режимах:

- 1) доповнення онтології новими знаннями;
- 2) вилучення з онтології інформації, яка за певними ознаками становить для користувача найменшу цінність.

Одним із підходів до вибору ознак, що ідентифікують знання, які доцільно вилучити з онтології, є зважування понять і зв'язків між ними під час їх додавання та використання під час експлуатації системи.

Для підтримання системи у робочому стані треба залишати визначений об'єм вільної оперативної пам'яті. У роботі обрано 10 % загального об'єму. Якщо під час наповнення цей контрольний показник перевищено, система переходить у режим оптимізації, під час якого виконується послідовне вибирання і вилучення з онтології тих її елементів, для яких відношення важливості до займаного в оперативній пам'яті місця буде мінімальним:

$$J_1 = \min_{1 \leq j \leq K} \frac{W_j}{m_j}, \quad (1)$$

де W_j – важливість елемента C_j ; m_j – місце елемента C_j в оперативній пам'яті; K – кількість понять онтології.

1.3. Час відгуку на зовнішнє звертання

Швидкодію, що визначається часом відгуку ІСППР на зовнішнє звертання, можна оцінити за максимальною кількістю дуг графу онтології у можливій траєкторії поширення повідомлення між поняттями, які задіяні у разі генерування відгуку. Просту та ефективну оцінку швидкодії системи, означеної у такий спосіб, дає ексцентриситет вершин графу онтології (рис. 1). Ексцентриситет E_j вершини C_j у зв'язному графі G – це максимальна відстань від вершини C_j до інших вершин у графі G . Тоді найгірша швидкодія системи:

$$J_2 = \max_{1 \leq j \leq K} E_j, \quad (2)$$

де E_j – ексцентриситет вершини C_j у зв’язному графі G , K – кількість понять онтології.

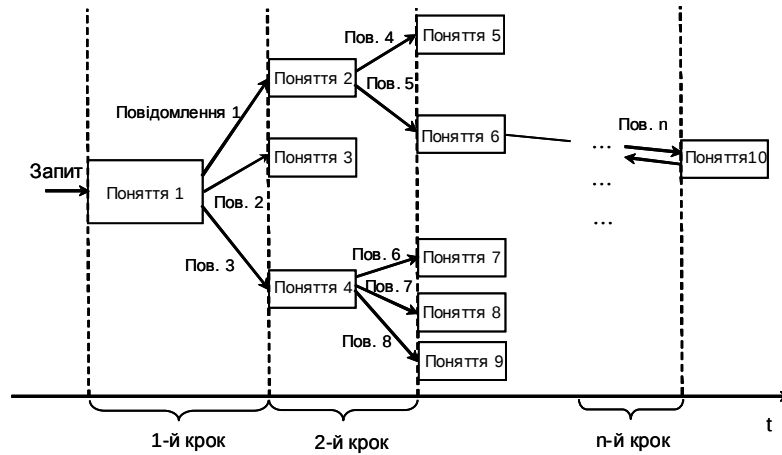


Рис. 1. Схема генерування відповіді опрацьовувачами повідомлень понять на зовнішнє звертання

1.4. Повнота онтології системи

Повноту онтології можна визначити як середній відсоток нетривіальних відповідей на зовнішні запити до системи. Під тривіальною розумітимемо відповідь, яка не надає її отримувачу нової інформації. Зокрема, до тривіальних можна зарахувати відповідь типу “Інформація відсутня”. Оцінити відсоток нетривіальних відповідей певної системи може інша тестова інтелектуальна система, повноту якої ми знаємо, або людина-експерт.

Оптимізації підлягає приведена повнота онтології, як відношення повноти онтології до кількості її понять. Якщо кількість понять порівнюваних онтологій чи БЗ однакові, що має місце у разі їх насичення під час експлуатації, достатньо порівнювати лише їх повноту.

Принцип визначення повноти онтології ґрунтується на методиці порівняння й оцінювання пошукових систем, запропонованої Американським інститутом стандартів (NIST) – одним з авторитетних органів стандартизації інформаційних технологій у США. Методика використовує корпус тестових запитань і документів, нагромаджених протягом конференцій з оцінювання систем текстового пошуку (TREC – text retrieval evaluation conference), що проводяться NIST.

Такий критерій є інтегральним і не дає змоги виконувати регулярну оптимізацію структури онтології на його основі, а призначений для оцінювання та порівняння функціонування інформаційних систем загалом.

1.5. Збалансованість предметної області

Під час автоматичної розбудови онтології можливі випадки, що в результаті деталізації понять ПО, подана кількістю класів, підкласів, їх екземплярів і властивостей, для одного класу може значно відрізнятися від відображення іншого класу. Збалансованість ПО виражається у рівномірному поданні її окремих розділів в онтології. Вимога збалансованості може бути перед комерційними універсальними ІСППР, сферу застосування яких не можна апіорно визначити.

Формальним критерієм збалансованості подання поняття-класу в онтології ІСППР може слугувати дисперсія важливості його підкласів:

$$s_j^{<k>} = \sqrt{\overline{W_i^{<k+1>}} - W_i^{<k+1>}}, \quad i = \overline{1..n}, \quad j = \overline{1..K}, \quad (3)$$

де $W_i^{<k+1>}$, $\overline{W_i^{<k+1>}}$ – важливість та усереднена важливість підкласів $k+1$ -рівня, відповідно; n – кількість підкласів у j -му класі, N – кількість класів у онтології.

Під час оптимізації онтології критерій збалансованості можна використовувати під час вибору тематики вірцевих текстів для доповнення онтології. У цій роботі алгоритм оптимізації змісту онтології не містить такий критерій до цільової функції, тому детально не розглядається.

2. Формулювання та розв’язування оптимізаційної задачі

Сформулюємо оптимізаційну задачу, яка буде покладена в основу процедур доповнення й оптимізації онтології. Автоматичне генерування онтології зумовлює необхідність здійснювати її локальну оптимізацію під час наповнення, і глобальну – на етапі впорядкування, коли наповнення призупинено до завершення процедури оптимізації. Метод оптимізації онтології містить задачу нормалізації її структури та задачу оптимізації змісту (рис. 2), до того ж:

1) усунення паралельних ребер, дублювання вершин з однаковими параметрами та інших особливостей структури графу онтології, які можуть порушити її цілісність і знижують ефективність функціонування інтелектуальної системи, є задачею нормалізації структури онтології;

2) оптимізація змістової частини онтології виконується з метою збільшення її швидкодії та інформаційної насиченості за заданих обмежень на фізичний об’єм пам’яті системи.

Передбачається, що розв’язування цих задач необхідно розділити в часі, причому, з метою збереження цілісності онтології, спершу треба виконувати її структурну перевірку, і лише потім – оптимізацію змістової частини послідовною редукцією її графу до виконання вимог вибраних критеріїв за допомогою максимізації суми важливостей вершин та ребер такого графу.

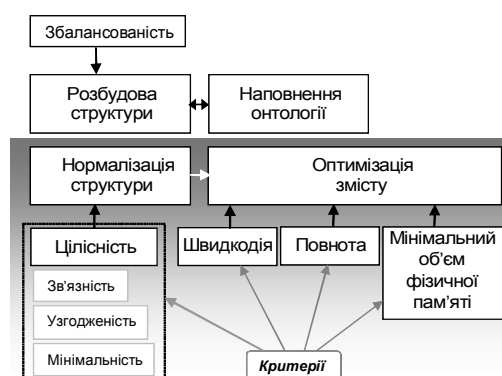


Рис. 2. Застосування критеріїв оптимальності онтології на етапах її наповнення та впорядкування

Деякі задачі показані на рисунку, а саме розбудова структури та наповнення онтології (на білому фоні) не потребують оптимізації, тому не є предметом детального розгляду в межах цієї роботи.

2.1. Задача нормалізації структури

Визначені вище критерії, які забезпечують цілісність онтології – зв’язність, узгодженість (відсутність взаємосуперечливих тверджень) та мінімальність (контрольована надлишковість), покладені в основу оптимізаційної задачі. Один із вказаних критеріїв – зв’язність – виступатиме як обмеження, яке треба задовольнити, натомість критерії мінімальності й узгодженості визначають дві окремі оптимізаційні задачі. В основу задачі мінімізації структури графу онтології покладена типова оптимізаційна задача теорії графів про відшукування мінімального кістяка, яка полягає у пошуку кістяка мінімальної важливості у зваженому графі. Натомість задача забезпечення узгодженості в структурі графу ефективно розв’язується, застосовуючи метод резолюцій. Важливим є порядок застосування таких оптимізаційних процедур, оскільки мінімізація структури частково досягається під час усунення логічних неузгодженостей. Отже, розглянемо їх послідовно в порядку застосування.

2.2. Забезпечення узгодженості структури онтології методом резолюцій

Наявність взаємнозаперечувальних тверджень у структурі онтології призводить до виникнення внутрішніх логічних конфліктів, порушуючи її цілісність, тому система повинна мати здатність своєчасно виявляти та вилучати ті з них, які визнаються помилковими. Таку задачу можна ефективно розв'язати, застосовуючи метод резолюцій – класичний метод автоматичного доведення теорем. Цей метод ґрунтується на доведенні від протилежного несуперечливості множини тверджень. Для цього формули числення предикатів за допомогою відповідних перетворень приводяться до диз'юнктивної форми, тобто подаються у вигляді набору речень – визначеного класу правильно побудованих формул (ППФ), безкванторної диз'юнкції літералів, кожний з яких є або предикатом, або запереченням предиката.

До недоліку методу резолюцій за його безпосереднього застосування можна зарахувати “сліпий” пошук, оскільки метод резолюцій є перебірною процедурою. Безпосереднє його застосування є неефективним, оскільки для великих онтологій він породжує значну кількість безперспективних резолюцій. Для запобігання “експоненційного вибуху” доводиться застосовувати ефективніші модифікації методу: семантична резолюція, лок-резолюція, лінійна резолюція. Суть цих модифікацій полягає у введенні тих чи інших критеріїв, за якими відбираються диз'юнкти, що беруть участь у черговій резолюції. Зазначені модифікації загалом не усувають бектрекінгового перебору, але дають змогу істотно скоротити цей перебір і роблять метод резолюцій цілком придатним для практичного застосування. У роботі застосовано алгоритм лінійної резолюції з огляду на його ефективність і простоту реалізації [8].

2.3. Нормалізація структури за допомогою пошуку мінімального кістяка

Нормалізація структури онтології, передусім, полягає у забезпеченні її мінімальності, тобто відсутності інформаційної надлишковості. У разі подання структури онтології зваженим графом, де важливість ребра відображає важливість відображеного ним зв'язку, а надлишковість може проявлятися у вигляді паралельних ребер, процедура усунення таких особливостей може полягати у послідовному вилученні ребер з мінімальною важливістю за збереження зв'язності усього графу. Для розв'язання цієї задачі запропоновано використати алгоритм побудови мінімального кістяка графу [7]. Кістяк – це підграф деякого графу $G(V, E)$, що містить всі його вершини і є деревом. Коли ребра графу мають важливість, можна побудувати мінімальне кістякове дерево (MST – Minimal Spanning Tree) зваженого графу – це кістякове дерево, важливість якого (сума важливостей його ребер) не перевищує важливості будь-якого іншого кістякового дерева.

У нашому випадку замість виділення ребер з мінімальною важливістю необхідно виділяти ребра з максимальною важливістю, враховуючи особливості подання концептуальним графом семантичної мережі з різними типами семантичних зв'язків.

Загальний граф онтології БЗ, як граф семантичної мережі, складається з підграфів, що мають спільні вершини-сутності, але різні типи ребер, які відображають різні семантичні зв'язки між цими сутностями. Процедура побудови мінімального кістякового дерева виконується роздільно, послідовно для кожного з підграфів, виділених за типом семантичних зв'язків.

Отже, загальна стратегія побудови мінімального кістяка в задачі мінімізації структури графу полягає в такому: на кожному кроці до фрагмента кістякового дерева, сформованого на попередньому кроці, додається ребро, максимальне за важливістю серед тих ребер, які з'єднують вже побудований фрагмент з вершинами, ще не доданими у цей фрагмент. З метою економної реалізації кроків цього процесу, зв'яжемо з кожною вершиною $x \in VG$ дві мітки $\beta(x)$ і $\gamma(x)$ – важливість максимального за важливістю ребра, що з'єднує вершину x з побудованим фрагментом максимального кістяка, та ім'я другої вершини цього ребра відповідно. Мітки $\beta(x)$ і $\gamma(x)$ дають змогу швидко знаходити на кожному кроці ребро максимальної важливості.

Прийmemo, що досліджуваний граф G , котрий відображає структуру онтології БЗ певного семантичного типу, заданий матрицею важливостей. Тоді алгоритм побудови мінімального кістяка матиме вигляд, описаний нижче.

1. Вибираємо будь-яку вершину графу бази знань G і знаходимо сусідню з нею з найбільшою важливістю ребра (зв'язку). Нехай $ET := \emptyset$, $VT := \{a\}$, де VT , ET – множини вершин і ребер фрагмента мінімального кістяка, що будується; a – будь-яка вершина з VG .

2. Присвоюємо мітці решті вершин $x \neq a$ за принципом $\beta(x) = W(x, a)$, якщо $x \in N_a$, і $\beta(x) = 0$, якщо $x \notin N_a$, $\gamma(x) = a$, N_a – оточення вершини a .

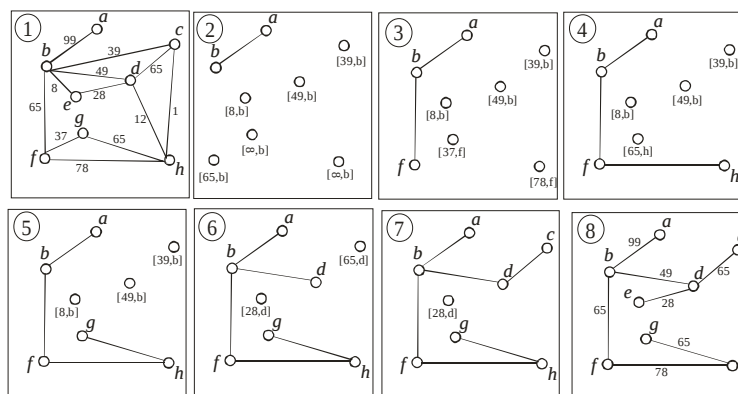
3. Вибираємо вершину $v \in VG \setminus VT$, найближчу до фрагмента за умовою:

$$b(v^*) = \min_{v \in VG \setminus VT} b(v).$$

4. Збільшуємо фрагмент. Серед незадіяних вершин знаходимо вершину з максимальним значенням x і з'єднуємо її з відповідною вершиною кістяка. Решті незадіяним вершинам переприсвоюємо значення $\beta(x)$ із $\gamma(x)$ з врахуванням збільшення кістяка, тобто, якщо $\beta(v) > W(v^*, v)$, то $\beta(v) := W(v^*, v)$, $\gamma(v) := v^*$. Якщо ж $\beta(v) \leq W(v^*, v)$, то мітку вершини v не змінюємо. Переходимо до п. 3.

5. Операції повторюємо доти, поки не залишиться жодної незадіяної вершини, тобто, якщо $|VT| = n$, процедуру треба завершити. Ребра, що містяться в множині ET , становлять мінімальний кістяк.

На рис. 3 зображено граф G і послідовність побудови фрагментів мінімального кістяка, які утворюються після кожної ітерації алгоритму.



• Рис. 3. Послідовність побудови мінімального кістяка графу G

Оскільки кожного разу до ET додається ребро, один кінець якого належить VT , а інший ні, то граф $T = (VT, ET)$ на кожному кроці залишається деревом. Після завершення роботи алгоритму це дерево буде кістяком, оскільки алгоритм припиняє роботу тоді, коли $VT = VG$.

Покажемо, що такий алгоритм буде мінімальний кістяк за час $O(|G|^2)$. Одноразове виконання п. 1 потребує $O(|G|)$ часу. Стільки ж часу необхідно для оновлення міток в п. 4, а збільшення фрагмента виконується за час $O(1)$. Оскільки кожен із п. 2–5 виконується $n-1$ разів, то оцінка трудомісткості алгоритму – $O(|G|^2)$.

2.4. Задача оптимізації змісту

Вище обрано та обгрунтовано критерії, за якими досягається оптимальність змісту онтології: швидкодія, повнота і мінімальний об'єм фізичної пам'яті. Процедура оптимізації онтології полягає у послідовному вилученні з неї понять, які мають найменшу важливість. На цьому етапі, ґрунтуючись на вказаних критеріях, необхідно сформулювати оптимізаційну задачу, тобто сформулювати цільову функцію та накласти відповідні обмеження, які впливають зі способу реалізації ІСППР. Оскільки оптимальність змісту онтології такої системи оцінюється суперечливими показниками, то одночасно “привести до спільного знаменника” їх не вдається. У цьому разі маємо ситуацію невизначеності цілей, яка може бути усунена застосуванням одного з методів усунення невизначеності (контрольних показників, методу лінійної згортки, введення метрики в просторі цільових функцій, компромісу Парето).

Отже, спершу необхідно розв'язати задачу пошуку оптимальної кількості понять у онтології, а тоді, використовуючи отримані результати, реалізувати процедуру періодичного почергового

доповнення онтології новими поняттями та вилучення з онтології елементів, семантичне значення яких для системи найменше. Така процедура має бути покладена в основу методу адаптації онтології до заданої користувачем ПО, як функціональний елемент відповідної інтелектуальної системи. Задача вилучення з онтології найменш важливих елементів може бути сформульована, як дискретна лінійна оптимізаційна задача, про що свідчать результати імітаційного моделювання процедур доповнення та зважування. Подібну умову такої оптимізаційної задачі має задача про наплічник [7]. Тільки тут наплічником виступатиме частка понять онтології, які підлягають вилученню.

Розглянемо обмеження, які накладаються на таку оптимізаційну задачу, визначимо оптимальний за критеріями швидкодія/повнота об'єм онтології та формалізуємо, подавши у вигляді алгоритму, розв'язування задачі вибору елементів для вилучення з онтології.

2.5. Накладання обмежень на оптимізаційну задачу

Як обмеження на кількість понять оптимізованої онтології введемо параметр мінімально допустимого об'єму вільної оперативної пам'яті. Вибір цього параметра зумовлений необхідністю підтримання системи у робочому стані під час доповнення її онтології. Його величина залежить від конкретних умов реалізації інтелектуальної системи (апаратного і програмного забезпечення). Для визначеності значення параметра приймемо 1 % від загального об'єму пам'яті, зайнятої онтологією:

$$\Delta M = 0,01 \sum_{j=1}^K m_j, \quad (4)$$

де m_j – об'єм, який займає j -й елемент в оперативній пам'яті ІСППР.

Якщо під час наповнення цей контрольний показник перевищено, запускається процес оптимізації змісту, під час якого виконується послідовне вилучення з онтології елементів, вибраних вказаним вище способом.

Зупинення оптимізації можна здійснювати як у разі досягнення цільовою функцією оптимального значення, так і досягнення параметрами оптимізації (цільової функції) критичних, тобто, обмежувальних значень. У разі оптимізації змісту онтології вилучення понять, які несуть мале змістове навантаження, а тому мають найменшу важливість серед усіх понять, що розглядаються, зупиняється, коли відносний об'єм вільної оперативної пам'яті перевищив друге порогове значення – 10 %. Вибір числового значення також залежить від умов реалізації ІСППР та, зокрема, її онтології.

2.6. Задача пошуку оптимальної кількості понять онтології

Під час експлуатації онтології виникає дилема забезпечення, з одного боку, якнайповнішого відтворення семантичної структури аналізованих текстів у разі їх подання у вигляді концептуальних графів, а з іншого, максимальної швидкодії системи їх інтелектуального аналізу.

Відомим лексично-семантичним джерелом, придатним для побудови онтології загального призначення є база даних WordNet, розроблена та підтримувана науковцями Стенфордського університету (США). Вона де-факто вважається стандартом семантики англійської мови і дозволяє відтворити “IS-A” – ієрархію понять обсягом близько 100 тис. термінів. Проте складність алгоритмів, які доводиться реалізувати на графах онтології з метою перерахунку коефіцієнтів важливості її елементів та побудови семантичних образів досліджуваних текстів, як показали дослідження, є степеневою, зокрема, пошук семантичного центра важливостей концептуального графу становить $O(n^3)$. Тому розмірність онтології істотно впливає на швидкодію ІСППР.

Існує клас прикладних задач, які вимагають опрацювання текстових даних в реальному масштабі часу (наприклад, моніторинг та фільтрація неперервного поштового трафіку). У кожний наступний момент надходить нова порція даних, що потребують аналізу. У таких випадках виникає дилема між вибором повноти такого аналізу та швидкодією. Зменшення розмірності онтології до певних оптимальних меж становить важливу оптимізаційну задачу етапу її розроблення, на якому є можливість врахувати особливості умов експлуатації системи, віддаючи перевагу або швидкодії, або точності отриманого розв'язку ІСППР.

Точність отриманого розв'язку ІСППР залежить від повноти подання в онтології певної ПО. У межах цієї роботи за основу взято просту лінійну модель залежності, яка забезпечує нас верхньою межею оцінки падіння точності, спричиненого зменшенням об'єму онтології:

$$t = \frac{r_e}{r_m} = f(G) \approx xG, \quad (5)$$

де $r_e = |R_e|$ – потужність множини R_e – правильних рішень, виданих ІСППР (наприклад, згідно з оцінкою експерта з певної ПО), знайдених за відповідним методом; $r_m = |R_m|$ – потужність множини R_m всіх виданих системою рішень, знайдених за відповідним методом; $R_e \subseteq R_m$; G – відносна кількість понять у онтології; $G = K/N$ (K – кількість понять онтології; N – кількість понять ПО); x – коефіцієнт лінійного зв'язку.

Отже, баланс між швидкодією та точністю пошуку можна у модельному наближенні звести до пошуку оптимального співвідношення швидкодії та повноти.

Практична реалізація онтології як семантичної мережі фреймів, у вершинах якої розміщені об'єктні модулі, а зв'язки реалізовані на базі опрацювачів повідомлень цих модулів, зумовлює необхідність оцінювати швидкодію системи як час реакції системи на зовнішнє звертання. В опрацюванні такого звертання може брати участь довільна кількість об'єктних модулів через звертання один до одного. Час реакції загалом визначається найповільнішим компонентом системи. Тому швидкодію такої системи можна оцінити через усереднену найповільнішу реакцію її компонентів.

Отже, як вже було згадано, для розв'язання такої задачі необхідно на основі критеріїв швидкодії та повноти визначити оптимальну кількість елементів в онтології БЗ ІСППР. З цією метою запишемо цільову функцію

$$f = \bar{E} + \frac{1}{G} \rightarrow \min. \quad (6)$$

Ексцентриситет вершини графу онтології, зв'язність якого забезпечується за рахунок обов'язковості існування “IS-A”-зв'язку між усіма елементами онтології, буде не більшим, ніж сума рівнів таксономічної ієрархії між заданим і кореневим елементами та максимальна кількість рівнів в онтології. Довільний додатковий семантичний зв'язок між елементами онтології знижує середній ексцентриситет вершин її онтології. Отже, для певного статистичного розподілу числа елементів за рівнями “IS-A”-ієрархії $P(i)$ середній ексцентриситет вершин графу, що відображає онтологію:

$$\bar{E} = \frac{N \sum_{i=1}^k [P(i) \cdot (i+k)]}{K}, \quad K = N \sum_{i=1}^k P(i), \quad (7)$$

де k – кількість рівнів у графі; K – кількість понять в онтології; N – кількість понять у словнику ПО; $P(i)$ – імовірність знаходження елемента онтології на i -му рівні таксономічної ієрархії понять, встановлена на основі аналізу лексичної бази даних WordNet.

Встановлено, що для такого фактичного розподілу означених критеріїв оптимальності та словника, який містить 100000 понять мінімум цільової функції відповідає оптимальній кількості 31000 понять в адаптивній онтології ІСППР.

2.7. Подання задачі видалення елементів з онтології як оберненої задачі про наплечник

Для практичного застосування дискретної оптимізаційної задачі про наплечник замість інтегральних критеріїв оптимальності онтології загалом необхідно застосовувати критерії, що характеризують кожен окремий елемент онтології та дають змогу приймати щодо нього рішення

про необхідність або недоцільність долучення його до онтології (вилучення з неї). Для цього розроблені критерії доводиться адаптувати для застосування у цьому методі.

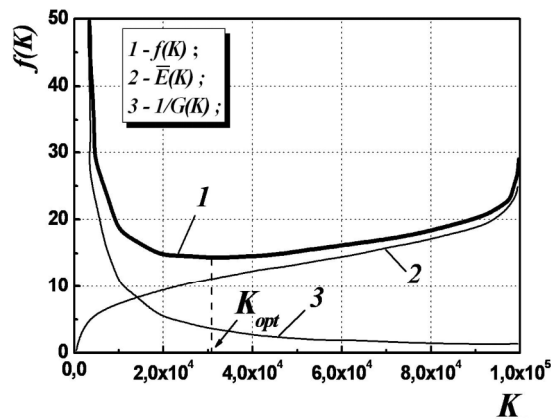


Рис. 4. Цільова функція пошуку оптимальної кількості понять онтології

Приймаємо, що кожен елемент онтології займає однаковий об’єм оперативної пам’яті і тому загальний зайнятий об’єм лінійно залежить від кількості визначених в онтології понять. Середній ексцентриситет вершин графу онтології враховується у власній важливості класів і важливості їх підкласів, спадаючи з рівнем. Якщо взяти до уваги факт, що в онтології міститься понад 10–100 тисяч окремих понять, розташованих у таксономічній 15–25-рівневій ієрархії, то відношення між важливістю понять, близьких до кореневого, та важливістю понять нижніх рівнів становить кілька порядків. Так для семантичних задач вилучення чи внесення до онтології понять нижніх рівнів під час оптимізації не змінює відчутно важливість інших понять. Отже, можна використати модельне припущення про незалежність середньої важливості понять онтології від їх загальної кількості, а лінійну дискретну задачу оптимізації змісту можна подати, як задачу про напличник. Напличником виступатиме та частка об’єму онтології, до якої треба зарахувати найменш цінні елементи з метою подальшого їх вилучення.

Застосовуючи простий перебір для онтології з n елементами, верхня оцінка кількості можливих комбінацій буде $2n$. До того ж для розв’язання такої задачі достатньо застосувати методи динамічного програмування, жадібний алгоритм чи метод гілок і границь.

Під час дослідження ефективності реалізації окремих методів вибір було зупинено на використанні жадібного алгоритму. Цей алгоритм має складність $O(n \log(n))$ і гарантує знаходження рішення не гірше ніж у два рази від оптимального, що є прийнятним для цієї задачі.

Нехай для онтології з n елементами надлишковий об’єм становить $N=1/10M$, де M – максимально допустимий об’єм. Якщо внесення i -го елемента до “напличника” x_i , $0 \leq x_i \leq 1$ визначає зиск x_i/W_i , то оптимальним наповненням “напличника” вважається таке наповнення елементами, яке максимізує загальний сумарний зиск. Задачу можна сформулювати так: максимізувати $\sum_{i=1}^n \frac{1}{W_i} x_i$ таких i елементів, для яких:

$$\sum_{i=1}^n m_i x_i \leq N, W_i > 0, m_i > 0, i = 1, \dots, n, \begin{cases} x_i = 0, & \text{якщо залишається,} \\ x_i = 1, & \text{якщо вилучається.} \end{cases} \quad (8)$$

Задача спрощується, якщо прийняти, $m_i = m = \text{const}$, що як правило відповідає умовам реалізації онтології.

Реалізація жадібного алгоритму в цій задачі зводиться до послідовного виконання таких операцій (рис. 5):

- 1) пошуку елемента з найменшою важливістю W_i ;
- 2) внесення його до множини тих, що вилучаються (присвоєння $x_i=1$);
- 3) обчислення та перевірки умови (8);

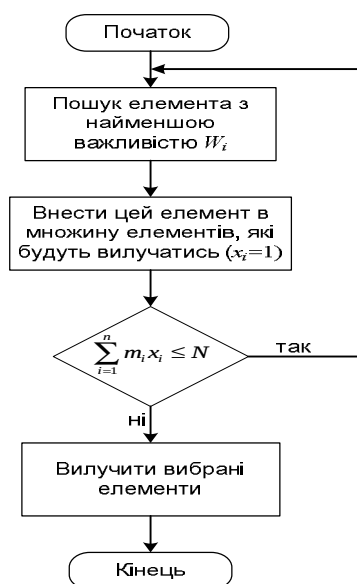


Рис. 5. Блок-схема реалізації жадібною алгоритму

- 4) у разі виконання умови (8) перейти до п.1;
- 5) у разі порушення умови припинити пошук та вилучити вибрані елементи.

Часова оцінка алгоритму цілком залежить від часової оцінки використаного алгоритму впорядкування, оскільки для реалізації самої стратегії наповнення після впорядкування предметів потрібен час $O(n)$.

Висновки

Автоматичне наповнення онтології спричиняє актуальність проблеми її періодичної оптимізації. Обґрунтовано множину критеріїв оптимальності структури та змісту онтології, які відображають вимоги до експлуатаційних характеристик і технічних можливостей реалізації відповідних систем. Задачу оптимізації онтології запропоновано розділити на задачу нормалізації структури та задачу оптимізації змісту. Усунення паралельних ребер, циклів, петель, дублювання вершин з аналогічними параметрами та інших особливостей структури графу онтології, а саме: логічних суперечностей, які

порушують її цілісність та знижують ефективність функціонування, є завданням нормалізації структури. Під час розв'язування цієї задачі застосовано метод резолюцій та алгоритм побудови мінімального кістяка. Оптимізація змісту онтології виконується для збільшення її інформаційної насиченості та забезпечення адаптації до інформаційних потреб користувача через періодичне зменшення її об'єму до заданих меж вилученням елементів, семантичне значення яких є найменше. Показано, що оптимізаційну задачу можна звести до дискретної оптимізаційної задачі про наплекник, алгоритми ефективного розв'язання якої відомі.

1. Литвин В.В. Автоматизація процесу розвитку базової онтології на основі аналізу текстових ресурсів / В.В. Литвин // Комп'ютерна та математична лінгвістика. Вісник НУ "Львівська політехніка" № 673, 2010. – С. 319–325.
2. Литвин В.В. Математичне забезпечення розвитку базової онтології предметної області / В.В. Литвин, Д.Г. Досин, Н.В. Шкутяк // 2-га Міжнар. наук.-практ. конф. „Сучасні інформаційні та інноваційні технології на транспорті”. – MINTT-2010. – Херсон. – 2010. – С. 345–348.
3. Пузанков Д.В. Оптимизация продукционных баз знаний для интеллектуальных агентов реального времени / Д.В. Пузанков, М.Г. Пантелеев, В.В. Денисов // УсиМ, 2002. – № 2 – С. 66–73.
4. Литвин В.В. Базы знаний интеллектуальных систем підтримки прийняття рішень / В.В. Литвин. – Львів: Вид-во Львівської політехніки. – 2011. – 240 с.
5. Інтелектуальні системи, базовані на онтологіях // Д.Г.Досин, В.В.Литвин, Ю.В.Нікольський, В.В.Пасічник – Львів: Цивілізація, 2009. – 414 с.
6. Shamsfard M. Learning ontologies from natural language texts / M. Shamsfard, A. Barforoush // Int. J. Hum. – Comput. Stud. 60(1). – 2004. – P. 17–63.
7. Свами М. Графы, сети и алгоритмы / М. Свами, К. Тхуласираман. – М.: Наука, 1984.
8. Нильсон Н. Принципы искусственного интеллекта / Н. Нильсон. – М.: Радио и связь, 1985. – 373 с.