



РОЗРОБКА СИСТЕМИ КЕРУВАННЯ СЕМАНТИЧНИМИ МЕРЕЖАМИ

Юрій Семчишин, Іван Кульпа, Ігор Колосовський,
Олександр Гречніков, Петро Гайда

Національний університет «Львівська політехніка»
вул. Степана Бандери, 12, м. Львів, 79013, Україна
e-mail: 7th@ukr.net, ivan.kulpa@gmail.com, igor.kolosovski@gmail.com,
sansofter@gmail.com, fezundo@gmail.com

Резюме: Робота присвячена семантичним мережам – способу представлення знань у виді набору вузлів-сутностей об'єднаних за допомогою ребер-зв'язків. Здійснено огляд історії розвитку та сучасного стану семантичних мереж, обґрунтовано актуальність проблеми. Описано процес проектування та розробки системи керування семантичними мережами: проектування структур даних, розробку алгоритмів семантичного аналізу текстів українською мовою, вибір алгоритмів візуалізації графових структур даних, реалізацію користувацького інтерфейсу у вигляді Web-додатку. Розроблена система є повністю функціональним засобом побудови та дослідження семантичних мереж, що підтвердив свою ефективність та послужить базовою платформою для подальших досліджень.

Ключові слова: система керування семантичними мережами, семантична мережа, семантичний аналіз, візуалізація графів.

DEVELOPING SEMANTIC NETWORK MANAGEMENT SYSTEM

Yuriy Semchyshyn, Ivan Kulpa, Igor Kolosovskyi,
Oleksandr Hrechnikov, Petro Hayda

Lviv National Technical University
12 Stepana Bandery Str., Lviv, 79013, Ukraine
e-mail: 7th@ukr.net, ivan.kulpa@gmail.com, igor.kolosovski@gmail.com,
sansofter@gmail.com, fezundo@gmail.com

Abstract: The paper considers semantic networks. It is a way of representing knowledge as a set of nodes-concepts connected by edges-relations. The history and current state-of-the-art of semantic networks is analyzed. The experience of designing and developing semantic network management system is described in four sections covering the following topics: data structures design, semantic analysis algorithm implementation, graph visualization methods selection and Web-based user interface development. The developed system is fully-operational tool for building and studying semantic networks. The system will be used for the further research.

Keywords: Semantic Network Management System, Semantic Network, Semantic Analysis, Graph Visualization.

ВСТУП

Семантична мережа (Semantic Network) – це спосіб представлення знань у виді набору вузлів, які представляють собою деякі сутності та є об'єднаними за допомогою зв'язків, що описують відношення між ними.

Ідея систематизації засобами семантичних відношень зародилась ще у далекому минулому. Наприклад, біологічна класифікація Карла Лінея

(Carl Linné) 1735 року, яка описує принципи поділу живих істот на класи та визначає відношення між цими класами [1].

Одним із новаторів свого часу був Фрідріх Людвіг Готлоб Фреге (Friedrich Ludwig Gottlob Frege) – німецький математик і філософ, який представив логічні формули у вигляді дерев. Фреге відзначив різницю між сенсом і значенням поняття, що задається деяким іменем. Під

значенням він розумів предметну область, яка має відношення до заданого імені, а під сенсом – деякий аспект завдяки якому відбувається перегляд даної предметної області. Пари імен, що мають однакове значення але різний сенс отримали назву «семантичних трикутників», або «трикутників Фреге» [2].

Прототипом семантичних мереж можна назвати екзистенціальні графи Чарльза Сандерса Пірса (Charles Sanders Peirce), які він запропонував в 1909 році [3]. Ці графи були вже більш схожі на сучасні семантичні мережі, в яких за допомогою діаграм зображувалися логічні твердження. Пірс вважав, що таке відображення можна назвати «логікою майбутнього».

В розвитку семантичних мереж важливу роль відіграли роботи відомого німецького психолога Отто Зельца (Otto Selz), який вивчав процеси мислення. Для зображення асоціацій і понять він використовував вузли і дуги, тобто графи. Роботи Зельца мали великий вплив на інших вчених, які використовували його ідеї для моделювання людського розуму.

Значного поштовху розвитку семантичних мереж надала робота одного з видатних лінгвістів ХХ століття француза Люсена Теньєра (Lucien Tesnière) під назвою «Основи структурного синтаксису» [4]. В цій книзі йдеться про синтаксичні залежності, зв'язки та їх відображення у вигляді дерева залежностей. Саме Теньєр вперше описав поділ синтаксису на статичний і динамічний а також ввів поняття валентності, що визначає способи поєднання слів з іншими елементами.

Першу діючу семантичну мережу розробив у 1956 році Річард Річенс (Richard Richens) при роботі над проектом машинного перекладу Кембриджського дослідницького мовного центру (Cambridge Language Research Unit). Система Річенса використовувала семантичну мережу в якості проміжної форми представлення при здійсненні машинного перекладу.

Наступну версію системи розробила англійська дослідниця Маргарет Мастерман (Margaret Masterman) в 1961 році; система включала в себе близько 100 концептуальних примітивів, за допомогою яких було описано близько 15000 понять. Система Мастерман призначалася для напівавтоматичного перекладу окремих абзаців урядових документів з англійської мови на французьку та давала можливість здійснювати переклад особам, які не знають французької мови [5]. Дослідження в цьому напрямку продовжував Моріс Вінсент Вілкс (Maurice Vincent Wilkes).

Цікавою є робота Роса Квіліана (Ross Quillian) 1968 року, в якій він описує власну семантичну

модель. Основним припущенням Квіліана було те, що значення кожного слова можна представити у вигляді великої кількості асоціацій, кожна з яких, в свою чергу, також можна деталізувати. В результаті утвориться величезна павутина вузлів і відношень між ними [6].

Аж до кінця 1970-х років семантичні мережі не мали великої популярності. Проте на початку 1980-х вони набувають значної поширеності, зокрема в системах штучного інтелекту для пошуку відповідей на поставлені запитання та дослідження процесів навчання. На жаль, апаратні засоби того часу не давали можливості створення масштабних систем на базі семантичних мереж.

А вже на початку 1990-х років на основі теорії семантичних мереж виникає концепція семантичної павутини (Semantic Web), автором якої вважається Тім Бернерс-Лі (Tim Berners-Lee) [7]. Ця концепція має на меті здійснення семантичної розмітки інформаційних ресурсів мережі Internet. Роботу над стандартизацією семантичної павутини було доручено організації «RDF Interest Group», назва якої 2004 року змінилась на «Semantic Web Interest Group». Першим проектом, заснованим на концепції семантичної павутини був проект під назвою «Dublin Core», в межах якого було розроблено кросс-платформенні стандарти метаданих, які можуть використовуватись для вирішення широкого спектру задач [8].

Комп'ютерне програмне забезпечення, що забезпечує користувачам можливість створення, збереження, оновлення, пошуку інформації та контролю доступу в семантичних мережах можна, за аналогією до загальноприйнятого терміну «система керування базами даних», називати системами керування семантичними мережами (СКСМ).

Сучасні системи керування семантичними мережами покликані вирішити проблемні задачі штучного інтелекту: здійснення семантичного аналізу тексту, структуризація специфічної інформації, створення засобів інтелектуального машинного перекладу тощо; серед найбільш вдалих проектів з реалізації систем керування семантичними мережами варто згадати такі:

- WordNet (<http://wordnet.princeton.edu/>) – лексична база даних, що з 1985 року розробляється у Принстонському університеті (Princeton University); станом на 2012 рік база даних вміщує 155287 слів згрупованих у 117659 синсетів (synsets, груп синонімів); основною проблемою WordNet є неможливість візуалізації отриманих результатів;
- LexiPedia (<http://lexipedia.com/>) – шестимовний (англійська, іспанська,

німецька, французька, італійська) тлумачний словник; містить відомості не лише про синоніми та антоніми, але й про фазиніми (fuzzynyms, семантично пов'язані слова); здійснює візуальне представлення зв'язків у вигляді семантичної мережі за допомогою інтерактивного Flash-додатку; до недоліків LexiPedia можна віднести її закритість та відсутність будь-якої технічної документації;

- MLSN (MultiLingual Semantic Network, <http://dcook.org/mlsn/>) – п'ятимовне (англійська, японська, німецька, китайська, арабська) відгалуження проекту WordNet, що розробляється з 2005 року; володіє базою даних більшого обсягу, ніж батьківський проект; серед недоліків системи варто відзначити можливість роботи виключно з простими іменниками; станом на першу половину 2012 року нестабільність системи унеможливило візуалізацію результатів;
- TextAnalyst (<http://megaputer.com/textanalyst.php>) – пропрієтарна система аналізу тексту з функціями формування семантичної мережі та тематичного древа, а також автоматичного реферування, кластеризації та індексації тексту; реалізовано підтримку зовнішніх словників; закритість системи робить неможливими її дослідження та подальший розвиток;
- UMLS (Unified Medical Language System, <http://www.nlm.nih.gov/research/umls/>) – медична система, що з 1986 року розробляється Національною медичною бібліотекою США (US National Library of Medicine); система призначена для структуризації біомедичної інформації; очевидно, що система не може мати жодних інших застосувань, окрім як в сфері охорони здоров'я.

Як бачимо, жоден з проектів не може претендувати на роль універсальної системи керування семантичними мережами загального призначення; проте потреба в такій системі є на часі, а в майбутньому актуальність досліджень в галузі семантичних мереж лише підвищуватиметься, особливо з огляду на стрімку популяризацію концепції семантичної павутини.

1. ОРГАНІЗАЦІЯ СЕМАНТИЧНОЇ МЕРЕЖІ

Розробку системи керування семантичними мережами було здійснено на мові програмування C# для програмної платформи Microsoft .NET 4.0 засобами середовища розробки Microsoft Visual Studio 2010. Таку комбінацію засобів розробки було обрано завдяки їх гнучкості та

потужності, зокрема з огляду на можливість використання запитів, інтегрованих у мову (Language INtegrated Queries, LINQ), автоматичного вивільнення виділеної пам'яті (Garbage Collector, GC) та вбудованих засобів для роботи з розширюваною мовою розмітки (eXtensible Markup Language, XML). Окрім того, вибраний набір засобів розробки робить можливою легку інтеграцію розробленої системи до складу наступних проектів.

В ході вибору структур даних, призначених для представлення семантичної мережі в пам'яті комп'ютера було розглянуто такі варіанти:

- за допомогою матриці суміжності;
- за допомогою списків інцидентності;
- за допомогою деревовидних структур.

Використання матриці суміжності могло б бути доцільним лише у випадку сильнозв'язних графів невеликої розмірності, у випадку ж слабозв'язних графів великої розмірності, які зазвичай виникають при роботі з семантичними мережами накладні витрати пам'яті стали б неприпустимо високими. Використання списків інцидентності дещо сповільнює навігацію по вершинам графа, проте водночас спрощує внутрішню реалізацію системи керування семантичними мережами та забезпечує вищий рівень її масштабованості (scalability). Використання деревовидних структур забезпечило б дещо вищу швидкість доступу до даних, проте неминуче спричинило б зниження масштабованості програмної системи а також виникнення деяких ускладнень в процесі серіалізації графа в сховище даних.

Відтак після аналізу всіх згаданих способів було вирішено використовувати представлення семантичної мережі в пам'яті комп'ютера за допомогою списків інцидентності.

При проектуванні сховища даних для системи керування семантичними мережами було вирішено відмовитись від використання централізованої бази даних, яка б надмірно ускладнила архітектуру програмної системи та викликала б певні труднощі при її розробці, впровадженні та підтримці. З огляду на структурований характер опрацьовуваної інформації було вирішено реалізувати сховище даних у вигляді локальних файлів у текстовому форматі XML, призначеному для зберігання структурованих даних.

Структуру тестового файлу даних в форматі XML представлено на Рис. 1.

```

<?xml version="1.0" ?>
<!-- Developed by Jewana Team -->
<!-- There are all your analyzed relationships -->
- <SemanticNetwork>
+ <Relationship1>
- <Relationship2>
  <Node1>Ваня</Node1>
  <Relation>любить</Relation>
  <Node2>Машу</Node2>
</Relationship2>
- <Relationship3>
  <Node1>Я</Node1>
  <Relation>їду</Relation>
  <Node2>додому</Node2>
</Relationship3>
- <Relationship4>
  <Node1>Колобок</Node1>
  <Relation>втік</Relation>
  <Node2>баби</Node2>
</Relationship4>
    
```

Рис. 1 – Структура XML-файлу

Для зберігання та опрацювання даних такого виду в пам'яті комп'ютера було на основі стандартної реалізації хеш-таблиць (hash tables) розроблено набір спеціалізованих структур даних, що забезпечують високу швидкість доступу до інформації про семантичні зв'язки на основі повної назви або фрагменту назви однієї з сутностей чи зв'язку між ними.

2. СЕМАНТИЧНИЙ АНАЛІЗ ТЕКСТІВ

В ході проектування програмного забезпечення для семантичного розпізнавання текстів природньою мовою було здійснено аналіз існуючих систем, виділено їх переваги та недоліки. Огляд існуючих продуктів виявив, що ринок засобів семантичного аналізу на сьогоднішній день перебуває в стані занепаду, особливо це стосується української частини ринку. Наразі не існує жодного комерційного програмного забезпечення для семантичного розпізнавання текстів українською мовою. Іншомовні системи переважно базуються на статистичних алгоритмах та зазвичай не можуть бути використані для побудови семантичних мереж на основі заданого тексту природньою мовою.

Загалом можна виділити три основні алгоритмічні підходи до семантичного аналізу текстів природньою мовою:

1. Спосіб генерації всіх граматично правильних словосполучень з подальшим обходом мережі синтактико-семантичних відношень [9]. Такий спосіб не можна вважати достатньо достовірним: генерування всіх граматично правильних словосполучень на основі введеного тексту не гарантує наявності семантичного зв'язку між обраними генератором сутностями.
2. Статистичний спосіб встановлення зв'язків між сутностями на основі підрахунку частоти

їх спільної появи в тексті. Даний спосіб є застосовним лише для текстів значного обсягу, оскільки не дає змоги отримати достовірний результат для сутностей, які згадуються в тексті статистично незначну кількість разів. Водночас спосіб є надзвичайно універсальним та може виявляти асоціативні зв'язки між сутностями навіть у граматично некоректному або особливого стилю тексті.

3. Синтаксичний спосіб встановлення зв'язків передбачає виявлення сутностей, змістовно пов'язаних з іншими сутностями на основі визначення синтаксичних конфігурацій в реченні згідно набору апріорних шаблонів [10]. Оскільки реалізація цього способу для кожної нової мови є складною алгоритмічною задачею, його не можна назвати достатньо універсальним. Разом з тим спосіб володіє високою точністю виявлення зв'язків та дає змогу отримати достовірні результати навіть для текстів вкрай малого розміру.

При розробці програмного забезпечення для семантичного розпізнавання текстів природньою мовою було вирішено використовувати синтаксичний спосіб встановлення зв'язків, який ґрунтується на синтаксичному аналізі, який в свою чергу вимагає попереднього виконання лексичного аналізу (визначення частини мови, до якої належить кожна лексема).

Синтаксичний аналіз – це процес співставлення лінійної послідовності лексем мови з її формальною граматику. Зазвичай результатом такого співставлення є синтаксичне дерево – структура даних, яка відображає синтаксичну структуру вхідного речення. Приклад синтаксичного розбору простого речення представлено в графічній формі на Рис. 2.



Рис. 2 – Синтаксичний розбір простого речення

Семантичний аналіз текстів природньою мовою є складною проблемою з галузі штучного інтелекту. Вичерпне вирішення цієї проблеми виходить далеко за межі даної роботи. Тим не менше було знайдено часткове рішення, яке дало

задовільні результати на використовуваному корпусі текстів. Запропонований алгоритм складається з таких кроків:

- [A1] Розбити речення на лексеми.
- [A2] Визначити для кожної лексеми її частину мови.
- [A3] Визначити імовірний семантичний зв'язок у реченні.
- [A4] Визначити імовірний суб'єкт у реченні.
- [A5] Визначити імовірний об'єкт у реченні.
- [A6] Опрацювати виявлений семантичний зв'язок між суб'єктом та об'єктом.

Визначення частини мови, до якої належить певне слово (крок A2) є нетривіальною задачею. Для її розв'язання розроблене програмне забезпечення здійснює інтеграцію з текстовим процесором Microsoft Word 2010 та використовує можливості вбудованого тезауруса. Недоліком такого способу є неможливість коректного опрацювання власних назв з некапіталізованою першою літерою.

3. ВІЗУАЛІЗАЦІЯ ГРАФОВИХ СТРУКТУР

Графова структура даних – це множина вершин та ребер, що з'єднують ці вершини. В контексті семантичних мереж графова структура представляє собою онтологію, її вершини – множину сутностей, а ребра – відношення, в яких перебувають ці сутності.

Задача візуалізації структур такого виду почала набувати все більшої актуальності разом з розвитком засобів графічного виводу обчислювальних машин, а саме з кінця ХХ століття. У зв'язку з невинним ростом графічних можливостей сучасних комп'ютерів та стрімким ускладненням моделей, що використовуються в різних галузях науки і техніки, ця задача потребує з кожним роком все більш якісного розв'язку.

Серед значної кількості різноманітних алгоритмів візуалізації графів найчастіше використовуються модифікації таких шести способів:

1. Найпростішим алгоритмом візуалізації графових структур є хаотичний (Randomized) алгоритм. Він передбачає розміщення вершин та ребер у довільному порядку та положенні. Очевидними перевагами цього алгоритму є вкрай високі швидкодія та простота програмної реалізації. Проте алгоритм володіє також значною кількістю недоліків: кількість ребер що перетинаються та сума відстаней між контекстно-близькими вершинами є неприпустимо високими, а візуальна картина представлення даних не

адекватна структурі взаємозв'язків. Типовий вигляд графа при візуалізації за допомогою хаотичного алгоритму представлено на Рис. 3а.

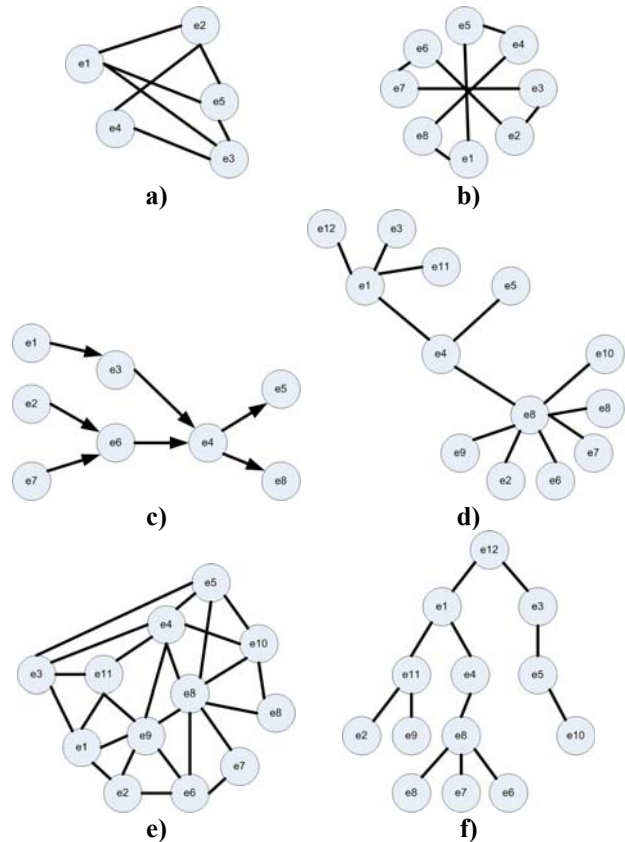


Рис. 3 – Типовий вигляд графа при візуалізації за допомогою алгоритму:

- а) хаотичного; б) кругового;
- в) Сугіяма; д) Фрухтермана–Райнголда;
- е) Камади–Кавая; ф) променевого дерева

2. Достатньо простим в реалізації є круговий (Circular) алгоритм візуалізації графів. Вершини розміщуються на колі через рівні проміжки в порядку, який дає можливість мінімізувати кількість ребер що перетинаються. Цей алгоритм дає змогу отримати задовільний результат за прийнятний час. Попри це застосовність алгоритму є обмеженою у випадку складних графів великої розмірності, оскільки не відображає належним чином структурні особливості графа. Типовий вигляд графа при візуалізації за допомогою кругового алгоритму представлено на Рис. 3б.
3. Алгоритм Козо Сугіяма (Kozo Sugiyama) було описано в роботі [11]. Схема Сугіяма відноситься до методів порівневої візуалізації та призначена перш за все для оптимізації візуальних критеріїв, таких як кількість перетинів між ребрами та їх сумарна довжина. У статті [12] автор описує новий евристичний метод для зображення графів, названий

«Magnetic Spring Model», та, на відміну від попередніх, придатний також для візуалізації орієнтованих графів. Алгоритми Сугіями довели свою ефективність та застосовуються для візуалізації графових структур даних в багатьох програмних системах. Типовий вигляд графа при візуалізації за допомогою алгоритму Сугіями представлено на Рис. 3с.

4. Алгоритм Фрухтермана–Райнголда (Fruchterman–Reingold) є одним із представників класу «керованих силами» («Force-Directed») алгоритмів візуалізації графів [13]. Цей клас алгоритмів спрямований на оптимізацію візуально-естетичних характеристик (зокрема на мінімізацію кількості ребер що перетинаються) при вирішенні задачі візуалізації графів із хаотично-розміщеними вузлами; такі алгоритми досліджував також Джузеппе ді Батіста (Giuseppe di Battista) [14]. Застосування алгоритму Фрухтермана–Райнголда є доцільним у випадку великих неорієнтованих графів, оскільки гарантується геометрична близькість пов'язаних вузлів. Проте низька швидкодія алгоритму унеможлиблює його застосування для графових структур надто великої розмірності. Типовий вигляд графа при візуалізації за допомогою алгоритму Фрухтермана–Райнголда представлено на Рис. 3д.

5. Іншим представником класу «керованих силою» алгоритмів є алгоритм Камади–Кавая (Kamada–Kawai). Попри деяку подібність до попереднього алгоритму, основним його принципом є мінімізація енергії шляхом пошуку похідної від рівнянь сили [15]. Алгоритм Камади–Кавая виконує впорядкування вузлів графу досить швидко і тому не накладає обмежень на розмір графів. Попри це, згідно візуально-естетичних критеріїв результат застосування цього алгоритму може потребувати деякого покращення, яке здійснюється, переважно, засобами алгоритму Фрухтермана–Райнголда. Типовий вигляд графа при візуалізації за допомогою алгоритму Камади–Кавая представлено на Рис. 3е.

6. Алгоритм променевого дерева (Radial Tree) є одним із методів, що візуалізують інформацію у вигляді деревовидних структур [16]. Суть алгоритму полягає у розміщенні самостійної вершини в геометричному центрі області побудови та в наступному послідовному радіальному розміщенні залежних вершин. Типовий вигляд графа при візуалізації за допомогою променевого алгоритму представлено на Рис. 3ф.

При розробці підсистеми візуалізації даних

системи керування семантичними мережами було здійснено програмну реалізацію трьох алгоритмів візуалізації графових структур, а саме: хаотичного, кругового та «керованого силами» алгоритму Фрухтермана–Райнголда. Вибір того чи іншого алгоритму візуалізації здійснюється користувачем та не залежить від характеристик семантичної мережі. Всі реалізовані алгоритми підтвердили свою застосовність та ефективність для вирішення задачі візуалізації семантичних мереж.

4. WEB-ІНТЕРФЕЙС СИСТЕМИ

В ході проектування системи керування семантичними мережами було прийнято рішення про реалізацію користувацького інтерфейсу у вигляді Web-додатку. Такий підхід, перш за все, забезпечить зручність використання та кросплатформенність системи, а, крім того, спростить імплементацію розширюваності та балансування навантаження в майбутньому.

Розробку користувацького інтерфейсу було здійснено для платформи Microsoft ASP .NET 4.0 з використанням технології асинхронних JavaScript та XML (Asynchronous Javascript And Xml, AJAX), яка є частиною концепції динамічного HTML (Dynamic HTML, DHTML) та дозволяє фонове виконання запитів до сервера без перезавантаження усієї сторінки [17, 18].

При розробці користувацького інтерфейсу системи було проведено аналіз зручності використання згідно рекомендацій, наведених у [19, 20]. Загальний вигляд користувацького інтерфесу під час роботи з системою представлено на Рис. 4.

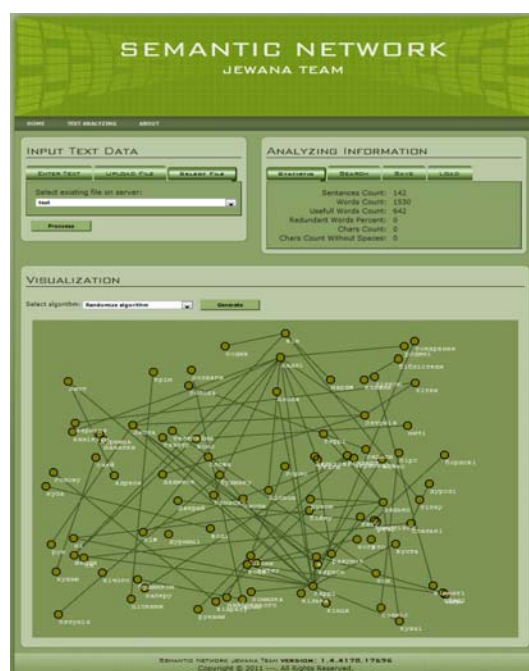


Рис. 4 – Вигляд користувацького інтерфейсу

Після завершення розробки користувацького інтерфейсу було виконано навантажене тестування (Load Test) та стресс-тестування (Stress Test) з наступною оптимізацією швидкодії, що дало змогу підвищити стабільність та надійність Web-додатку.

ВИСНОВКИ

Розроблена система керування семантичними мережами є повністю функціональним засобом побудови та дослідження семантичних мереж, що підтвердив свою ефективність.

Реалізовані алгоритми семантичного аналізу текстів природньою мовою та спеціалізовані структури для збереження даних підтвердили можливість опрацювання за прийнятний час (десятки секунд) значних обсягів тексту українською мовою (сотні тисяч знаків), створення на його основі достатньо великих онтологій (тисячі сутностей) та, що найголовніше, швидкого (долі секунди) виконання запитів на отриманих даних. Всі реалізовані алгоритми мають обчислювальну складність лінійно або сублінійно залежну від розміру опрацюваного тексту.

Спосіб реалізації користувацького інтерфейсу виправдав себе, а обрані та реалізовані методи візуалізації графових структур зробили можливим представлення даних в інтуїтивно-зрозумілому вигляді.

Попри те, що розроблена система навряд чи може бути без додаткових змін використана для вирішення реальних бізнес-задач контент-аналізу, вона послужить базовою платформою для подальших досліджень авторського колективу, зокрема в напрямку використання гіперграфів для реалізації семантичних мереж з n -арними зв'язками, навантажування ребер мітками дати і часу для розв'язання семантичних конфліктів тощо.

СПИСОК ЛІТЕРАТУРИ

- [1] Biological classification, *Wikipedia*, http://en.wikipedia.org/wiki/Biological_classification, – 2012-01-01.
- [2] Gottlob Frege, *Wikipedia*, http://en.wikipedia.org/wiki/Gottlob_Frege, 2012-01-01.
- [3] Existential Graph, *Wikipedia*, http://en.wikipedia.org/wiki/Existential_graph, 2012-01-01.
- [4] Tesnière L., *Elements of Structural Syntax*, Second Edition, Klincksieck, 1976, 674 p. (In French)
- [5] Yngve V., Yates D. M., Masterman M., von Glasersfeld E., Machine translation researchers: their works and contribution into the development of machine translation, <http://www.br.com.ua/referats/Computers/10099.html>. – 2012-01-01. (In Ukrainian)
- [6] What are Semantic Networks? <http://www.cs.bham.ac.uk/research/projects/poplog/computers-and-thought/chap6/node5.html>. – 2012-01-01.
- [7] Tim Berners-Lee, http://en.wikipedia.org/wiki/Tim_Berners-Lee. – 2012-01-01.
- [8] Dublin Core, http://en.wikipedia.org/wiki/Dublin_Core. – 2012-01-01.
- [9] Yermakov A. E., Explication of meaningful elements from text by the means of syntactic analysis-synthesis, *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue»*. Nauka, (2003), pp. 136-140. (In Russian)
- [10] Kisilyov S. L., Yermakov A. E. and Plyeshko V. V., Search for facts in natural language text based on network descriptions, *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue»*, Nauka, (2004), pp. 282-285. (In Russian)
- [11] Sugiyama K., Tagawa S., Toda M., Methods for visual understanding of hierarchical systems, *IEEE Transactions on Systems, Man and Cybernetics*, (11) 2 (1981), pp. 109-125.
- [12] Sugiyama K. and Misue K., Graph drawing by the magnetic spring model, *Journal of Visual Languages and Computing*, (6) 3 (1995), pp. 217-231.
- [13] Fruchterman T. M. J. and Reingold E. M., Graph drawing by force-directed placement, *Software: Practice and Experience*, (21) 11 (1991).
- [14] Battista G., Eades P., Tamassia R. and Tollis I. G., Algorithms for drawing graphs: an annotated bibliography, *Computational Geometry: Theory and Applications*, (4) 5 (1994), pp. 235-282.
- [15] Kamada T. and Kawai S., An algorithm for drawing general undirected graphs, *Information Processing Letters*, 31 (1989), pp. 7-15.
- [16] Di Battista G., Eades P., Tamassia R. and Tollis I. G., *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1998, 397 p.
- [17] Holzner S. *AJAX Bible*, Wiley, 2007, 695 p.
- [18] Woolston D., *Pro AJAX and the .NET 2.0 Platform.*, Apress, 2006, 488 p.
- [19] Borodayev D. V., *Website as Object of Graphic Design*, Septima, 2006, 288 p. (In Russian)
- [20] Nielsen J. and Pernice K., *Eye-tracking Web Usability*, New Riders Press, 2009, 456 p.



Юрій Семчишин, кандидат технічних наук, старший викладач кафедри програмного забезпечення Національного університету «Львівська політехніка». Народився 1984 року в місті Івано-Франківську, 2005 року отримав диплом бакалавра комп'ютерних наук з відзнакою, а 2006 року – диплом

магістра програмного забезпечення автоматизованих систем з відзнакою, захистив кандидатську дисертацію в 2010 році. Наукові інтереси зосереджені навколо вирішення різноманітних задач штучного інтелекту.



Іван Кульпа, бакалавр програмної інженерії, студент магістратури кафедри програмного забезпечення Національного університету «Львівська політехніка». Народився 1990 року в місті Івано-Франківську, захистив бакалаврську роботу в 2011 році.

Цікавиться вирішенням задач візуалізації даних.



Ігор Колосовський, бакалавр програмної інженерії, студент магістратури кафедри програмного забезпечення Національного університету «Львівська політехніка». Народився 1990 року в місті Львові, захистив бакалаврську роботу в 2011 році.

Цікавиться опрацюванням текстів природньою мовою.



Олександр Гречніков, бакалавр програмної інженерії, студент магістратури кафедри програмного забезпечення Національного університету «Львівська політехніка». Народився 1990 року в місті Львові, захистив бакалаврську роботу в 2011 році.

Цікавиться розробкою Web-додатків.



Петро Гайда, бакалавр програмної інженерії, студент магістратури кафедри програмного забезпечення Національного університету «Львівська політехніка». Народився 1990 року в місті Львові, захистив бакалаврську роботу в 2011 році.

Цікавиться розробкою високооптимізованих структур даних.



DEVELOPING SEMANTIC NETWORK MANAGEMENT SYSTEM

Yuriy Semchyshyn, Ivan Kulpa, Igor Kolosovskiy,
Oleksandr Hrechnikov, Petro Hayda

Lviv National Technical University
12 Stepana Bandery Str., Lviv, 79013, Ukraine
e-mail: 7th@ukr.net, ivan.kulpa@gmail.com, igor.kolosovski@gmail.com,
sansofter@gmail.com, fezundo@gmail.com

Abstract: *The paper considers semantic networks. It is a way of representing knowledge as a set of nodes-concepts connected by edges-relations. The history and current state-of-the-art of semantic networks is analyzed. The experience of designing and developing semantic network management system is described in four sections covering the following topics: data structures design, semantic analysis algorithm implementation, graph visualization methods selection and Web-based user interface development. The developed system is fully-operational tool for building and studying semantic networks. The system will be used for the further research.*

Keywords: *Semantic Network Management System, Semantic Network, Semantic Analysis, Graph Visualization.*

1. INTRODUCTION

Semantic Network is a way of representing knowledge as a set of nodes (some concepts), which are connected by edges (corresponding relations).

The idea of using semantic relations to classify entities have been known since the first half of XVIII century, but only in the late XX century did semantic networks become applicable for solving such problems of artificial intelligence as machine translation and information retrieval [1 – 8].

Amongst the modern semantic network management systems the most sophisticated ones are the following five:

- WordNet (<http://wordnet.princeton.edu>);
- LexiPedia (<http://lexipedia.com>);
- MLSN (<http://dcook.org/mlsn/>);
- TextAnalyst (<http://megaputer.com/textanalyst.php>);
- UMLS (<http://www.nlm.nih.gov/research/umls/>).

None of the mentioned projects can be considered as general-purpose semantic network management system. Development of such system is a topical task which will become even more urgent with further popularization of the Semantic Web conception.

2. DATA STRUCTURES

The semantic network management system was developed in accordance with principles of the object-oriented programming paradigm in the C# programming language for the Microsoft .NET 4.0 software platform using the Microsoft

Visual Studio 2010 development environment.

The following three ways of storing semantic network in computer memory were carefully evaluated:

- using Adjacency Matrix;
- using Incidence Lists;
- using Tree-Like Structures.

Usage of incidence lists appeared to be the most efficient way; hence, a set of specialized hash-based speed-optimized data structures were designed.

Considering structural nature of data being processed, it was settled to use a set of local files of specific structure as data storage (Fig. 1).

3. SEMANTIC ANALYSIS

There are three possible approaches to semantic analysis of texts in natural language:

1. Combinatorial Approach [9].
2. Statistic Approach.
3. Syntactic Approach [10].

It was decided to implement syntactic approach to semantic analysis which implies lexical analysis of source sentence and recognizes some heuristic patterns so as to build syntactic tree (Fig. 2).

In order to define which part of speech the particular word is, the developed semantic network management system integrates with the Microsoft Word 2010 text processor to utilize its thesaurus.

4. GRAPH VISUALIZATION

The most frequently used graph visualization algorithms are modifications of the following six methods:

1. Randomized Algorithm (Fig. 3a).
2. Circular Algorithm (Fig. 3b).
3. Sugiyama Algorithm (Fig. 3c) [11, 12].
4. Fruchterman–Reingold Algorithm (Fig. 3d) [13, 14].
5. Kamada–Kawai Algorithm (Fig. 3e) [15].
6. Radial Tree Algorithm (Fig. 3f) [16].

In the developed semantic network management system randomized, circular and Fruchterman–Reingold algorithms are available for user's choice.

5. WEB INTERFACE

The Web-based user interface of the developed semantic network management system were designed, implemented and tested using the Microsoft ASP .NET web technology in accordance with all the best practices recommended (Fig. 4) [17 – 20].

6. CONCLUSION

The developed semantic network management system is fully-operational and well-tried tool for building and studying semantic networks.

Implemented algorithms for semantic analysis of texts in natural language and corresponding data structures have proven their efficiency. All the implemented algorithms do have computational complexity that depends linearly or sublinearly on the size of input text.

Developed Web-based user interface implements three different graph visualization algorithms and assures intuitive and friendly interaction.

The developed system will be used as a platform for further research specifically in the directions of using hypergraphs for representing semantic networks with n -ary relations, weighting the edges with datetime stamps for resolving semantic conflicts et cetera.

7. REFERENCES

- [1] Biological classification, *Wikipedia*, http://en.wikipedia.org/wiki/Biological_classification, – 2012-01-01.
- [2] Gottlob Frege, *Wikipedia*, http://en.wikipedia.org/wiki/Gottlob_Frege, 2012-01-01.
- [3] Existential Graph, *Wikipedia*, http://en.wikipedia.org/wiki/Existential_graph, 2012-01-01.
- [4] Tesnière L., *Elements of Structural Syntax*, Second Edition, Klincksieck, 1976, 674 p. (In French)
- [5] Yngve V., Yates D. M., Masterman M., von Glasersfeld E., Machine translation researchers: their works and contribution into the development of machine translation, <http://www.br.com.ua/referats/Computers/10099.html>. – 2012-01-01. (In Ukrainian)
- [6] What are Semantic Networks? <http://www.cs.bham.ac.uk/research/projects/poplog/computers-and-thought/chap6/node5.html>. – 2012-01-01.
- [7] Tim Berners-Lee, http://en.wikipedia.org/wiki/Tim_Berners-Lee. – 2012-01-01.
- [8] Dublin Core, http://en.wikipedia.org/wiki/Dublin_Core. – 2012-01-01.
- [9] Yermakov A. E., Explication of meaningful elements from text by the means of syntactic analysis-synthesis, *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue»*. Nauka, (2003), pp. 136-140. (In Russian)
- [10] Kisilyov S. L., Yermakov A. E. and Plyeshko V. V., Search for facts in natural language text based on network descriptions, *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue»*, Nauka, (2004), pp. 282-285. (In Russian)
- [11] Sugiyama K., Tagawa S., Toda M., Methods for visual understanding of hierarchical systems, *IEEE Transactions on Systems, Man and Cybernetics*, (11) 2 (1981), pp. 109-125.
- [12] Sugiyama K. and Misue K., Graph drawing by the magnetic spring model, *Journal of Visual Languages and Computing*, (6) 3 (1995), pp. 217-231.
- [13] Fruchterman T. M. J. and Reingold E. M., Graph drawing by force-directed placement, *Software: Practice and Experience*, (21) 11 (1991).
- [14] Battista G., Eades P., Tamassia R. and Tollis I. G., Algorithms for drawing graphs: an annotated bibliography, *Computational Geometry: Theory and Applications*, (4) 5 (1994), pp. 235-282.
- [15] Kamada T. and Kawai S., An algorithm for drawing general undirected graphs, *Information Processing Letters*, 31 (1989), pp. 7-15.
- [16] Di Battista G., Eades P., Tamassia R. and Tollis I. G., *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 1998, 397 p.
- [17] Holzner S. *AJAX Bible*, Wiley, 2007, 695 p.
- [18] Woolston D., *Pro AJAX and the .NET 2.0 Platform.*, Apress, 2006, 488 p.
- [19] Borodayev D. V., *Website as Object of Graphic Design*, Septima, 2006, 288 p. (In Russian)
- [20] Nielsen J. and Pernice K., *Eyetracking Web Usability*, New Riders Press, 2009, 456 p.