

Ю. В. Білодрай; В. І. Месюра, к. т. н., доц.

КОМУНІКАЦІЙНА СИСТЕМА ПРОЕКТНОГО Й ОПЕРАЦІЙНОГО МЕНЕДЖМЕНТУ В СОЦІАЛЬНИХ МЕДІА

У роботі запропоновано новий тип комунікаційних систем проектного й операційного менеджменту, який передбачає наявність центрального комунікаційного середовища на основі соціальних медіа, спільного для людей і під'єднаних систем третіх сторін. Використано властивості соціальних медіа як складних адаптивних систем, які природно агрегують релевантні дані, мають значний інтеграційний та мобільний потенціал, а також мікроблогінг-повідомлення яких сприяють полегшеній семантичній анотації. Розробка відкриває новий економічно ефективний засіб комунікації різнорідних поточних та історичних даних для комбінованого аналізу й підтримки колективної цілеспрямованої діяльності. Система є основою інтеграції продуктів третіх сторін на основі популярної бізнес-моделі соціальних медіа, що суттєво збільшує набір можливих рішень, зазвичай складних і недоступних широкому колу користувачів. Розробка орієнтована на діяльність розподілених некомерційних, малобюджетних, наукових і освітніх організацій, публічних проектів та малого бізнесу.

Ключові слова: соціальні медіа, складні адаптивні системи, CAS, Enterprise 2.0, контрольована мова, мікроблогінг, наноформат, семантичне подання, секвенційний підхід, менеджмент.

Вступ і постановка завдання

Зі зростанням зацікавленості до соціальних медіа як до середовища накопичення капіталу знань та інструментарію організаційної діяльності, варто зазначити недостатній розвиток їхніх аналітичних можливостей для підтримки спрямованої групової діяльності в nearly-real-time-режимі. Однією з основних причин є складність технологій NLP/IE й семантичного подання інформації. У цьому дослідженні запропоновано підхід до створення систем на основі соціальних медіа з набором функцій «класичних» корпоративних систем проектного й операційного менеджменту. Така система повинна базуватись на соціальних медіа, які природно агрегують релевантні дані як складні адаптивні системи [1], мати ознаки Emergent Social Software Platform ESSP / «enterprise social software» [2]. Передбачається, що інформація акумулюється за допомогою спеціального агента й перетворює послідовності повідомлень користувачів щодо діяльності, подій, фізичних і абстрактних сутностей у цілісне семантичне подання з найменшою складністю процесів NLP/IE за умови задовільної точності й зберігає модель у спеціальному репозиторії. Система повинна мати швидкий наочний механізм фільтрації частин моделі з можливістю як агрегації даних, так і отримання їх індивідуальних елементів і візуальну репрезентаційну частину моделі на екранах комп'ютерів і мобільних приладів. Таким чином створюється динамічний процес, коли учасники поповнюють базу знань, а система підтримує цілісне подання знань, оптимізує його й надає учасникам, чим підвищує ефективність їхньої діяльності [3].

Основна частина

Загальна схема запропонованого рішення. На рис 1. зображено загальну схему системи. Поглинання даних (1А) відбувається в середовищі популярного мікроблогінгу, де користувачі і сторонні системи обмінюються поточною інформацією, а також поповнюють і використовують історичну інформацію, застосовуючи преанотацію тексту наноформатами й контрольовану мову [4]. Використовуючи візуальне конструювання (1Б), користувачі формують інформаційні дошки необхідної семантичної інформації як секвенції фільтрів і зберігають їх (2Б). Інформаційні дошки є наборами SPARQL- чи SeRQL-запитів до

семантичного репозиторія. Складність створення запитів прихована в механізмі автогенерації запитів з допомогою аналогу секвенційного підходу [5]. Поглинуті й частково структуровані дані (1А) обробляються методами NLP і ІЕ (2А) та перетворюються в RDF-тріпли, які зберігаються в семантичному репозиторії (3А). RDF-графи створені на основі зв'язків, заданих OWL-онтологіями. Онтології виступають в ролі схем не тільки для пошуку елементів інформації і їхніх зв'язків, але й керують перетворенням інформації в RDF, виступаючи в якості динамічних галузево-специфічних словників контрольованих мов. Інформаційні дошки генеруються чи періодично, чи в результаті подій (5А), і є сукупностями SPARQL- чи SeRQL-запитів до семантичного репозиторія (4А). Дошки демонструються користувачам у вигляді, подібному до того, в якому вони формуються (6А).

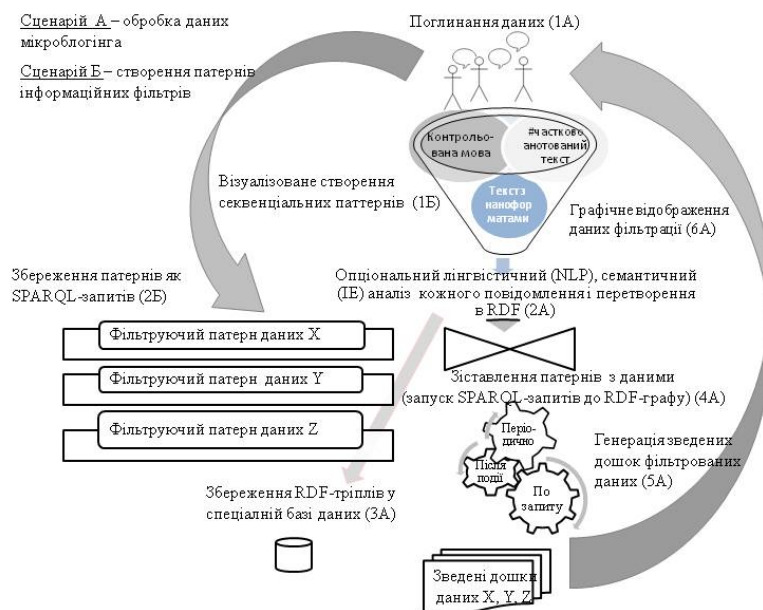


Рис. 1. Загальна схема системи підтримки й контролю колективної діяльності на основі мікроблогінгу

Приєднання до платформ мікроблогінгу. Схема підключення до мікроблогінгу зображена на рис. 2. Основні актори: користувачі (люди), користувачі (системи третіх сторін), користувач – інформаційний маршрутизатор (ІМ) і система проектного/операційного менеджменту (далі – «система»). Усі вони є суб'єктами двостороннього (щодо один одного) і багатостороннього (у загальному контексті) інформаційного обміну (показано як <<follows>>). Усі елементи зареєстровані стандартним способом як користувачі платформи мікроблогінгу й системи. ІМ є частиною системи, але зображений як окремий елемент для наочності його центральної функціональної ролі, – це єдина точка з'єднання середовища мікроблогінгу й системи. Відносно платформи мікроблогінгу і процесу комунікації всі елементи поєднані однорангово й технічно однаково. Зв'язки між користувачами і ІМ є обов'язковими, між користувачами – опціональними і бажаними. Незважаючи на те, що ІМ є функціонально центральним вузлом, система комунікації – однорангова, тому що мікроблогінг виявляє властивості інструменту Pull communication, а також Interactive communication одночасно [6]: ІМ є «слухачем» timelines користувачів, за якими він «слідуює», як інші користувачі. Також ІМ публікує власні повідомлення, які можуть бути прочитані користувачами в разі необхідності. Сукупність з'єднаних таким чином елементів формує простір комунікації групової операційної чи проектної діяльності.

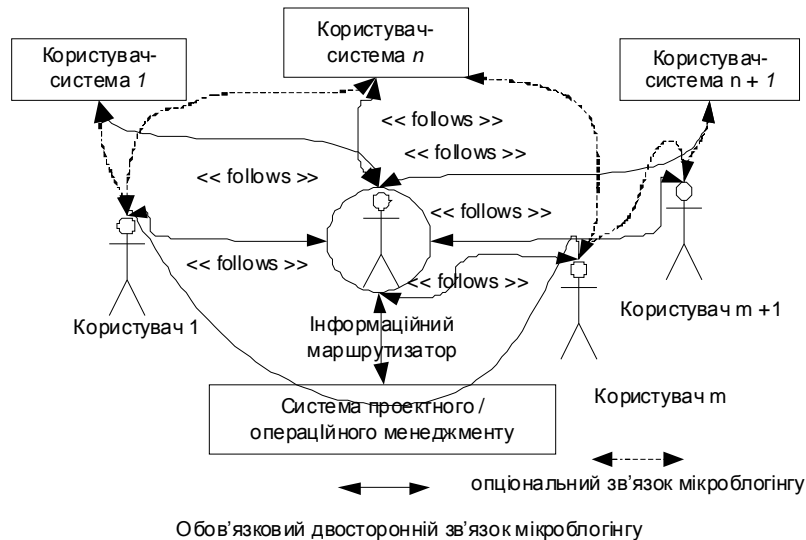


Рис. 2. Підключення користувачів до платформи мікроблогінгу

Щодо захисту інформації, більшість мікроблогінгових платформ загального призначення підтримують закриті профілі користувачів, і інформація такого користувача не може бути зчитана іншим неавторизованим користувачем. Якщо всі авторизовані користувачі мають закриті профілі, уся система буде закрыта для оперативного доступу ззовні, але інформація, що зберігається такими мікроблогінговими платформами, зазвичай має рівень захисту нижчий за прийнятий для підприємств. Це може бути проблемою для організацій, які володіють інформацією, що не підлягає розголошенню.

Таблиця 1

Сценарії використання платформи мікроблогінгу в системі

Сценарій використання	Пояснення
Користувач виконує запит на слідування – «follow» до інформаційного маршрутизатора (ІМ)	Дія формування робочого простору. Актори: усі користувачі, які беруть участь в інформаційному обміні
Інформаційний маршрутизатор підтверджує запит на слідування	Дія формування робочого простору. Актори: ІМ і користувачі, яким дозволена функція «follow»
Користувач публікує інформацію про діяльність чи знання	Актори: людина, система третьої сторони, ІМ
Користувач зчитує інформацію про діяльність чи знання	Актори: людина, система третьої сторони, ІМ (усі крім останнього – опціонально, останній – обов'язково)
Користувач відсилає команду чи запит про інформацію іншому користувачеві	Актори: людина, система третьої сторони, ІМ
Користувач реагує на команду чи запит	Актори: людина, система третьої сторони, ІМ

Збір даних повідомлень. За допомогою системи можна збирати будь-яку галузеву- чи проектно-специфічну інформацію (чи загальну, якщо це підтримано семантично): статус виконання задач, логістика, проблеми й рішення, інформація для обчислення, наприклад, EVM-аналіз [6], дані сенсорів, призначення завдань людям і системам, використання історичної інформації. Приклади можливих вхідних повідомлень:

Work package WP2 has EV=1.1 and AC = 0.9;

Project XYZ has workpackage WP1 and workpackage WP2. WP2 is half-complete; (1)

An accident on crossing of Street X and Street Y, Height of tidal wave is 1.5m near the east coast.

На вході системи (рис. 1) повинні знаходитись повідомлення, придатні для інтерпретації, а система повинна мати дані класів сутностей і їх властивостей для такої інтерпретації. Два типи вхідних повідомлень розглядаються в контексті цієї роботи: 1) повідомлення з контрольованою мовою CLIE; 2) повідомлення з наноформатами, подібними Twitter Data [7]:

#WP2 has new CR #change_data_format. Find details here < URL >, (2)

#WP2 \$hasNewCR Change data format. Find \$details here < URL>.

Вирази (2) демонструють, що текст із Twitter Data має схожий рівень описової складності з текстом контрольованої мови CLIE і є подібним до звичайних повідомлень мікроблогінгу. При цьому повідомлення є вже семантично анотованими сумісним з RDF і OWL поданням.

Семантична відповідність наноформатів і RDF-подання. Формат Twitter Data (різновид наноформатів), який дозволяє вмонтовувати семантичні визначення в повідомлення мікроблогінгу за допомогою відкриваючого і закриваючого символу “\$” (чи кінця повідомлення). Семантичні дані, структуровані як прості пари *name-value* послідовності символів, tuples *\$name1 value1 \$name2 value2*, чи triples *\$s subject \$name value* (де *\$s subject* може бути @<user name>, #hash-tag або URI):

discuss \$project New Project \$module New Module, (3)

<subject><predicate> <object> <predicate> <object>
#WP2 \$hasNewCR Change data format. Find \$details here < URL>.

У RDF-форматі з Notation3 серіалізацією запис (3) має такий вигляд:

murj:WP2 prj-ont:hasNewCR [prj-ont:subject “Change data format”; prj-ont:details “URL“]. (4)

Модифікований підхід CLIE. Робота [4] описує CLIE-підхід обробки повідомлення контрольованої мови для отримання онтологій. У цій роботі запропоновано новий підхід до контрольованої мови на основі CLIE і наноформатів, відмінність якого від CLIE полягає в наявності двох складників семантичних патернів речень: статичного і динамічного. Статичний складник визначає жорсткі правила взаємного розташування частин речення за правилами наноформату. Динамічний складник визначає характер семантичного зв'язку між сутностями речення і є властивістю онтологічних класів сутностей, які описані реченням (*predicate* для *object* і *subject*). Підхід дозволяє створювати RDF-описи текстових повідомлень з наноформатами на базі семантичних репозиторіїв RDF на основі RDF-S і OWL-схем.

Таблиця 2

Приклад патернів речень Twitter Data

Патерн речення	Використання / приклад
\$<predicate> <object>	Перетворення пари змінна – значення: <i>\$population 2300</i>
@<subject> \$<predicate> <object>	Перетворення тріпла опису події: @ChangeBoard \$hasMeeting tomorrow
#<subject> \$<predicate> <object> \$<predicate> <object>	Перетворення тріпла з додатковою інформацією щодо subject: #WP2 \$hasNewCR change_data_format. Find \$details here < URL>
@<?> \$<inferred/default_predicate> <?> Невідомі subject, object чи predicate.	Поведінка за замовчуванням 1) спроба знайти клас за ключовим словом – найменуванням сутності; 2) спроба знайти <i>predicate</i> -властивість за знайденим класом; 3) у протилежному випадку генерація нових об'єктів онтології і встановлення їх взаємозв'язку за замовчуванням, наприклад, «hasRelation»

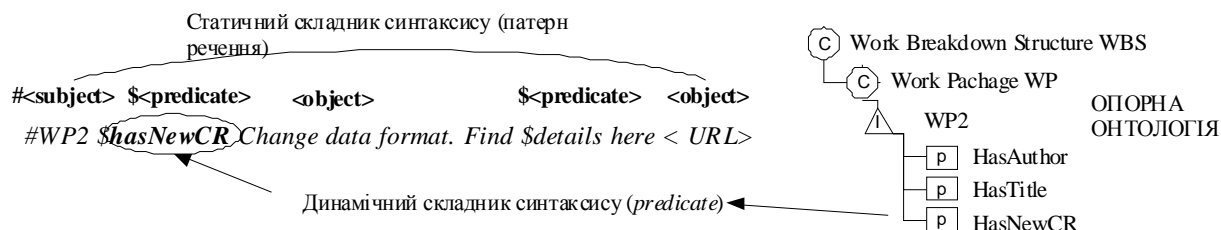


Рис. 3. Sentence patterns CLIE з статичним і динамічним складниками для Twitter Data

На рис. 4 подано схему NLP/ІЕ-аналізу нової комунікаційної системи, яка є модифікованою реалізацією CLIE для використання наноформатів мікроблогінгу. Якщо повідомлення преанотовані наноформатом, можна використати спрощені процедури NLP для фільтрування від незначних символів, згодом тегуючи текст означеннями *subject*, *predicate* і *object*, для подальшого семантичного перетворення в RDF. Для цього можна застосовувати тегування Брілля чи Common Pattern Specification Language (CPSL) [8].

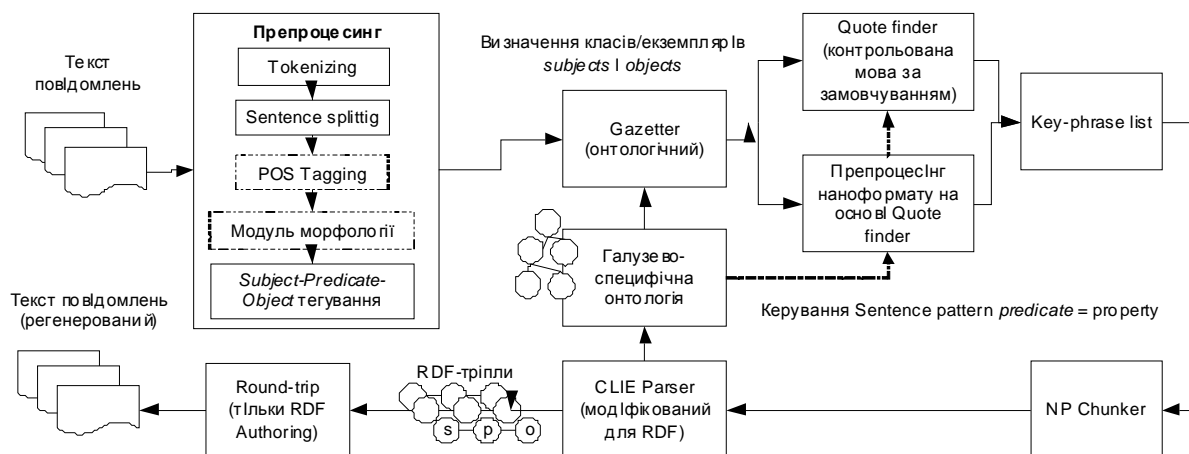


Рис. 4. Перетворення тексту з наноформатами й контрольованою мовою в RDF з CLIE

Ontologic gazetter є новим елементом у схемі CLIE, він анотує текст посиланнями на IRI онтологічних ресурсів. Ними можуть бути галузеві і проектні онтології чи їхні об'єднання з онтологіями більш загального характеру. Цей інструмент повинен мати можливість анотації не тільки значеннями класів онтології (Classes), але й їх екземплярів (Instances). За основу передбачається взяти підхід OBIE (Ontology Based Information Extraction) [9].

Quote finder є модифікованим модулем CLIE, анотує текст за принципом таблиці 2, окрім останнього випадку в таблиці варіантів використання, і знаходить ключові фрази повідомлень, які відповідають комбінації: 1) дозволених патернів наноформату; 2) властивостей сутностей з онтологій як *predicates*. Система може працювати і зі стандартним підходом CLIE, якщо не знаходить наноформатування в повідомленні. *Key-phrase list* містить знайдені *Quote finder* ключові фрази. *NP Chunker* є допоміжним інструментом.

Модифікований CLIE Parser: 1) знаходить неідентифіковані *Ontologic gazetter* і *Quote finder* елементи (<*subject*>, <*predicates*> і <*objects*>) і поповнює онтологію новими згенерованими об'єктами, обов'язково вказуючи значення *predicate* за замовчуванням, наприклад, «hasRelation» (див. останній випадок в таблиці 2); 2) генерує RDF-тріпли корисної інформації.

Модифікований Round-trip-модуль модуль регенерує текст вхідного повідомлення з RDF-тріплів для синтезу індивідуальних елементів даних (опціонально, в залежності від

задачі візуалізації інформації).

Потенціал використання даних RDF-подання. У межах цієї роботи існує завдання створення простого у використанні механізму пошуку інформації, що накопичується в подібній системі. Як зазначено вище, семантично оброблену інформацію повідомлень мікроблогінгу передбачається зберігати у вигляді RDF-тріплів у семантичному репозиторії, альтернативі реляційній базі даних (RDBMS), що дозволяє схожі функції зберігання й опитування даних. Інформація в них зберігається не у вигляді даних у зв'язаних таблицях, а в репозиторії triple statements, де кожен рядок зберігає один тріпл *<object, predicate, subject>*, який має посилання на різноманітні дані за допомогою IRI / URI (рис. 5).

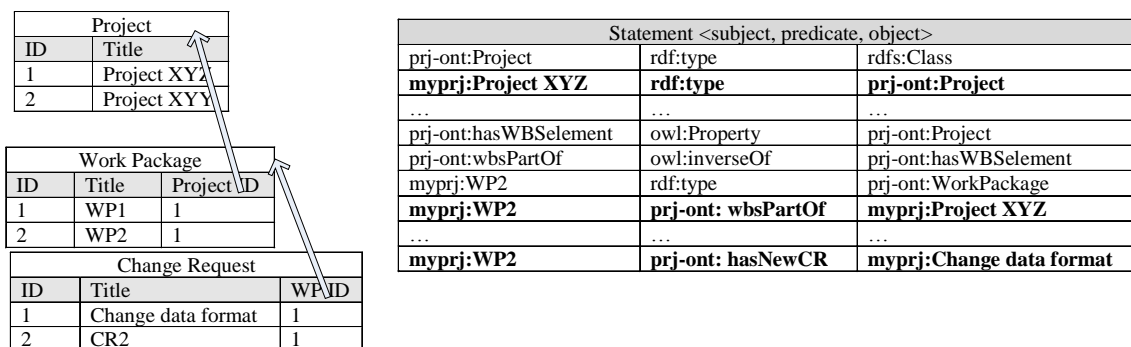


Рис. 5. Відмінність подання даних у RDBMS і семантичних репозиторіях

Семантичні репозиторії більше підходять для ефективного використання в якості бази знань, ніж реляційні бази даних, оскільки мають закладені механізми здобуття неявних знань [11]. Сучасні семантичні репозиторії зазвичай використовують онтології як семантичні схеми. Семантичні схеми онтологій визначають моделі даних: типи даних, можливі зв'язки, а також правила, за якими знаходять неявні дані, *entailment rules*.

Передумова: prj-ont:hasElement a owl:TransitiveProperty .

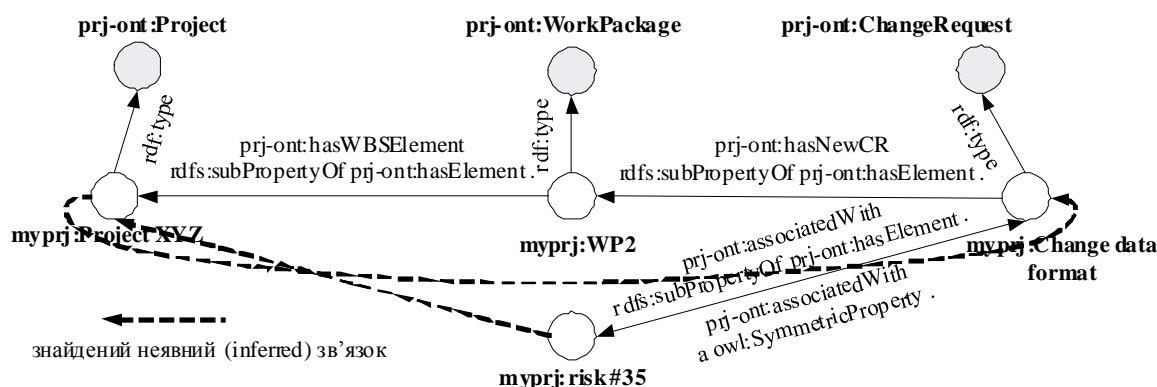


Рис. 6. Приклад знаходження неявного зв'язку RDF-даних за допомогою OWL-схеми

На рис. 6 зображено частину RDF-графу, який використовує семантику схем-онтологій для встановлення зв'язків між сутностями. Завдяки транзитивним властивостям предиката *hasElement* і властивостям *hasWBSelement*, *hasNewCR* та *associatedWith* як його *subProperty*, можна зробити висновки: 1) запит на зміну в пакеті робіт WP2 з назвою «Change data format» належить проекту Project XYZ (тому що WP2 належить цьому проекту); 2) ризик #35, який виник внаслідок появи запиту на зміну «Change data format», належить проекту Project XYZ.

Також у схемі представлено двосторонній зв'язок між запитом на зміну «*Change data format*» і ризиком *risk #35*.

Особливістю сучасного підходу до використання мов семантичного подання для семантичних репозиторіїв є те, що набір можливостей пошуку прихованих даних, який ґрунтується на відомій інформації (*inference*), залежить від обраної мови семантичної схеми [11]. Отже, використовуючи різні мови і діалекти схеми (RDF-S, OWL Light, OWL DLP, OWL Horst, OWL 2 EL/QL/RL/DL), залежно від потреб можна вибирати різні набори *entailment rules* і різні можливості «глибини» пошуку неявної інформації. Використання схеми з більш потужними *entailment rules* призводить до більших витрат. OWL DLP [12], що передбачається використати, і є діалектом на основі потужного OWL DL з деякими обмеженнями та логічним програмуванням, яке дозволяє дещо спростити використання схем.

Секвенційні інформаційні фільтри й подання результатів пошуку. У роботі [5] Мокіна В. Б і Ящолта А. Р. розроблено теоретичні та алгоритмічні основи нового методу автоматизованого формування складних звітів реляційних баз даних інформаційних систем, який є аналогом секвенційного методу проектування технічних систем. Метод базується на зображенні вхідних і вихідних даних у вигляді секвенційної моделі, дозволяє синтезувати програмний код для формування звіту для заданого набору вхідних і вихідних даних і не вимагає знань мови SQL.

Мови запитів до семантичних даних SPARQL і SeRQL мають багато спільних рис з SQL. Існує можливість застосувати аналогію й використати розроблений в [5] теоретичний і алгоритмічний апарат для автоматичного формування запитів до семантичних репозиторіїв з онтологічними OWL-схемами і з інкапсуляцією складності формування запитів для користувачів.

Як зазначено в [5], ключовий пункт алгоритму спеціального апарату секвенційного методу полягає у створенні моделі звіту, яка пов'язує його вхідні і вихідні дані. Спеціальний запис (5), на який є покликання в [5], описує основний елемент апарату секвенцій, що формалізує зв'язки входів і виходів кінцевих автоматів:

$$\phi | - \varphi, \quad (5)$$

де ϕ та φ – функції алгебри логіки. Права частина цього запису (функція φ) приймає одиничне значення в разі, якщо одиничного значення набуває ліва частина (функція ϕ). Це також може інтерпретуватись, як «висловлення φ істинне, якщо істинне висловлювання ϕ ». У роботі [5] представлено секвенційний опис моделі інформаційних систем з реляційними базами даних для двох вхідних і однієї вихідної змінної в такому вигляді:

$$x_1 x_2 \Big|_{\substack{\tilde{s}_1 \\ \tilde{s}_2}} y_1, \quad (6)$$

де вхідні змінні x_1 , x_2 , які є полями таблиці \tilde{s}_1 , відповідають (мають логічний зв'язок) вихідній змінній y_1 , яка є полем таблиці \tilde{s}_2 .

Ключовим фактом є те, що дані семантичного репозиторія RDF є сукупністю RDF-тріплів (граф), на відміну від реляційних таблиць з полями даних. (див. рис. 5). Однією з основних ідей семантичних web-технологій є можливість унікально ідентифікувати дані за допомогою IRI, незалежно від фізичного місця знаходження даних. Тобто, у семантичних репозиторіях оператор SELECT і речення FROM мають дещо інший зміст. У SPARQL [13] речення FROM призначене для звернення до RDF-графів і є опціональним. У SeRQL [14] FROM

покликається на paths expressions і також є опціональним.

Проводячи аналогію з реляційними базами даних, де таблиці описують сутності програмного додатка (наприклад, projects, work packages, change requests), можна вибрати класи *objects* і *subjects* RDF-тріплів як аналоги таблицям \tilde{s}_1 , \tilde{s}_2 формули (6). Класи є ключовими і стійкими елементами RDF-графів (подібно таблиць у схемах реляційних баз даних), завжди асоційованими з інформацією зв'язків у графах (*predicates*). І вхідними, і вихідними змінними SPARQL- і SeRQL-запитів є посилання на актуалізовані екземпляри класів (*instances*) чи властивості цих екземплярів (*properties* чи *attributes*), описаних за допомогою RDF-S / OWL.

Формула (6) для RDF-графів може бути подана так: вхідні змінні x_1 , x_2 , які є екземплярами класу чи властивостями класу подання RDF \tilde{s}_1 , відповідають (мають логічний зв'язок з) вихідній змінній x_1 , яка є властивістю чи екземпляром класу \tilde{s}_2 . Наприклад, класом \tilde{s}_1 може бути *prj-ont:Project* (клас проекту), а властивостями-змінними *rdf:label* (назва проекту) і *prj-ont:hasWBSElement* (індикатор наявності асоційованих з проектом Work Breakdown Structure елементів). Класом \tilde{s}_2 може бути *prj-ont:ChangeRequest* (клас запиту на зміну проекту), а вихідною змінною – список змін проекту (change requests). Важливо зазначити, що, незалежно від форми запису запитів, незмінно буде працювати принцип пошуку прихованої інформації в семантичних репозиторіях зі схемами-онтологіями (рис. 6).

Використовуючи аналогію з викладками роботи [4], продемонструємо побудову SPARQL-запитів і секвенцій для двох послідовних фільтрів інформаційної дошки, зображених на рис. 7 за допомогою виразів (7).

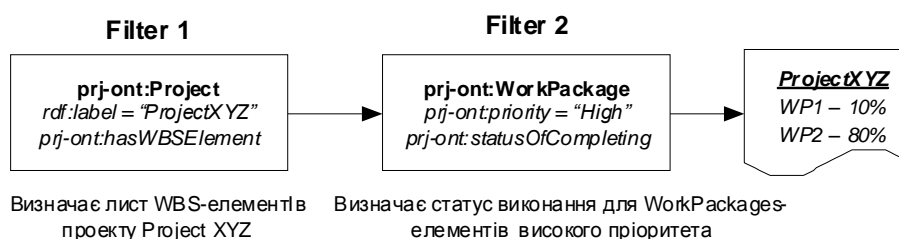


Рис. 7. Приклад семантичного патерна інформаційної дошки з двома фільтрами

SELECT ?workpackage

WHERE prj-ont:Project rdf:label "ProjectXYZ" ;prj-ont:hasWBSElement ?workpackage . (7)

SELECT ?statusOfCompleting

WHERE<[вихідна інформація першого запиту]> prj-ont:hasPriorityLabel "High";
prj-ont:hasStatusOfCompleting ?statusOfCompleting .

Перетворимо описи вхідних, вихідних даних і семантичних класів з (7) за допомогою символічних умовних позначень у форму, яка дозволяє записати запити як секвенції:

$$\begin{aligned}
 x_1 &= (rdf:label "ProjectXYZ"), x_2 = (prj-ont:hasWBSElement ?workpackage), \\
 x_3 &= (prj-ont:hasPriorityLabel "High"), \\
 x_4 &= (prj-ont:hasStatusOfCompleting ?statusOfCompleting), \\
 \tilde{s}_1 &= (prj-ont:Project), \tilde{s}_2 = (prj-ont:WorkPackage),
 \end{aligned}
 \tag{8}$$

$$y_1 = [myprj : WP_n \text{ rdf : type pri-ont : WorkPackage}],$$

$$y_2 = [pri-ont : statusOfCompleting \text{ rdf : type rdfs : Literal}],$$

де записи в дужках [] вказують, що змінна містить множину значень.

Як зазначено вище, особливістю подання схем семантичних даних є їхня структура як класів, їхніх екземплярів і властивостей. Для розробки теорії секвенційних форм-запитів до семантичних даних доцільно враховувати цю особливість. За допомогою виразів типу $A(B)$, який символізує належність об'єкта B об'єкту A, запишемо приклад опису елементів запитів:

$$\tilde{s}_1 = (C_1, x_2 = P(I(C_1)), y_1 = [I(C_1)]), \quad (9)$$

де C – клас, I – екземпляр класу, P – властивість екземпляра класу.

Тоді в секвенційній узагальнюючій формі записи запитів (7), у яких вихідна змінна першого запиту є вхідною змінною другого запиту, набудуть такого вигляду:

$$P(I(C_1))_1 P(I(C_1)) \left| \frac{C_1}{C_2} [I(C_2)] \text{ та } [I(C_2)] P(I(C_2))_1 P(I(C_2))_2 \right| \frac{C_2}{C_1} [I(C_1)]. \quad (10)$$

Важливо зазначити, що секвенції для запитів до семантичних даних на основі класів може мати один і той самий вхідний і вихідний клас. Це видно з запису другої секвенції (10).

Секвенційний апарат дозволяє виконувати ряд операцій над секвенціями, таких як: «І» та «Або», що дозволяє переходити до їх скорочених записів. Спрощені секвенції дозволяють автоматично генерувати вже оптимізований код SPARQL.

UML Component діаграма прототипу системи. Для створення прототипу систем вибрано набір NLP/IE інструментів GATE і семантичний репозиторій Ontotext SwiftOWLIM, завдяки їхньому великому набору інструментів, ефективній інтеграції, production-рівню технології, безперервному вдосконаленню за допомогою дослідницької діяльності, безкоштовним рішенням для дослідницьких проектів [8]. На рис. 8 зображено діаграму компонентів системи.

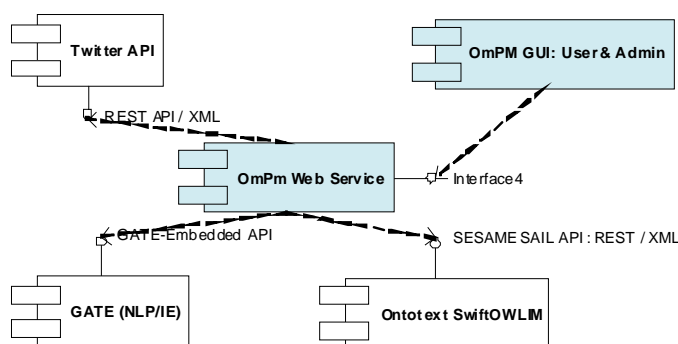


Рис. 8. Діаграма компонентів системи

Висновки

У роботі запропоновано новий тип системи менеджменту, яка використовує властивості соціальних медіа як CAS. Її основні риси: комунікаційно-центрична концепція; використання неспеціалізованих соціальних медіа як центрального комунікаційного середовища менеджменту організацій і проектів; використання коротких повідомлень

мікроблогінгу, а не великих масивів тексту, що спрощує процес семантичної анотації і збільшує точність; використання секвенційних фільтрів інформаційного пошуку в семантичних поданнях, що спрощує використання. Подібні системи призначені, насамперед, для розподілених некомерційних та малих комерційних підприємств, наукових і освітніх організацій, імплементації підходу communicative rationality [15] в громадських проектах. Наступним кроком є виготовлення прототипу системи.

СПИСОК ЛІТЕРАТУРИ

1. John H. Miller. Complex Adaptive Systems: An Introduction to Computational Models of Social Life / H. Miller John, E. Page Scott. – Princeton University Press, 2007. – 284 p.
2. McAfee Andrew. Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges / Andrew McAfee. – 1st edition. – Harvard Business School Press, 2009. – 240 p.
3. Manas Jerry. Managing the Gray Areas: Essential Insights for Leading People, Projects & Organizations / Jerry Manas. – RMC Publications, Inc, 2007. – 232 p.
4. User-friendly ontology authoring using a controlled language [Електронний ресурс] / V. Talban, T. Polanjnar, H. Cunningham, K. Botcheva // Department of Computer Science, University of Sheffield, 2006. – Режим доступу: <http://gate.ac.uk/sale/irec2006/ctie/ctie.pdf>.
5. Мокін В. Б. Новий метод автоматизованого формування звітів інформаційних систем на основі секвенційного підходу / В. Б. Мокін, А. Р. Яцолт // Вісник Вінницького політехнічного інституту. – 2008. – № 3. – С. 50 – 56
6. A guide to the project management body of knowledge (PMBOK Guide)-Forth Edition : ANSI/PMI 99-001-2008 An American National Standard. – Project Management Institute, Inc., 2008. – 470 p.
7. Twitter Data – a simple, open proposal for embedding data in Twitter messages [Електронний ресурс] / Todd Fast, Jiri Copsa. – 2009. – Режим доступу : <http://twitterdata.org/>.
8. The Common Pattern Specification Language [Електронний ресурс] / Douglas E. Appelt, Boyan Onyshkevych // Artificial Intelligence Center SRI International. – Режим доступу: <http://www.aclweb.org/anthology/X/X98/X98-1004.pdf>.
9. OBIE Ontology Based Information Extraction [Електронний ресурс] // The University of Sheffield. 1995 – 2001. – Режим доступу: <http://gate.ac.uk/demos/obie/obie.html>.
10. Wilcock Graham. Introduction to Linguistic Annotation and Text Analytics / Graham Wilcock. – 1st edition. – Morgan & Claypool Publishers, 2009. – 160 p.
11. Ontotext and GATE Training (Module 15) – RDF, SPARQL and Semantic Repositories : GATE Training Course August 2010 [Електронний ресурс] // The University of Sheffield. 1995 – 2001. – Режим доступу : <http://gate.ac.uk/wiki/TrainingCourseAug2010/track4-slides/module-15.pdf>.
12. DLP – An introduction [Електронний ресурс] / Denny Vrandecic, Peter Haase, Pascal Hitzler, York Sure, Rudi Studer // Insitute AIFB University of Karlsruhe, Germany. – Режим доступу: <http://logic.aifb.uni-karlsruhe.de/dlp/download/dlpintro.pdf>.
13. SPARQL Query Language for RDF, W3C Recommendation [Електронний ресурс] // W3C. – 2008. – Режим доступу : <http://www.w3.org/TR/rdf-sparql-query/>.
14. The SeRQL query language (revision 1.2) [Електронний ресурс] // Aduna. 1997 – 2009. – Режим доступу : <http://www.openrdf.org/doc/sesame/users/ch06.html#section-blanknodes>.
15. Erwann Michael-Kerjan. The Irrational Economist: Making Decisions in a Dangerous World / Erwann Michael-Kerjan, Paul Slovic. – Public Affairs, 2010. – 336 p.

Білодрай Юрій Валентинович – провідний інженер-програміст, PMP, Senior Engineering Project Manager (Eng.), Yuriy.Belodray@gmail.com.

Арісент Текнолоджис Україна.

Месюра Володимир Іванович – к. т. н., професор кафедри комп'ютерних наук, інституту інформаційних технологій і комп'ютерної інженерії.

Вінницький національний технічний університет.