

УДК 004.89

М.М. Глибовець,

А.А. Жигмановський

Національний університет «Києво-Могилянська академія»

СЕМАНТИЧНА КООРДИНАЦІЯ ЗНАНЬ НА БАЗІ АРХІТЕКТУРИ PEER-TO-PEER

У роботі описано реалізацію прототипу системи розподіленого обміну інформацією на основі архітектури Peer-to-Peer (P2P). Система складається з модулів опису локальних даних, автоматичної побудови схем та відповідностей і трансформації запитів. Прототип реалізовано на мові С# із використанням технології WPF. Можлива інтеграція із сучасним Web 2.0 середовищем. Зв'язок між клієнтами зорієнтований на користувачів мобільних пристроїв.

В работе описано реализации прототипа системы распределенного обмена информацией на базе архитектуры Peer-to-Peer (P2P). Система состоит из модулей описания локальных данных, автоматического построения схем, соответствия и трансформации запросов.

Прототип реалізований на мові С # з використанням технології WPF. Можлива інтеграція з сучасним Web 2.0 середой. Св'язок між клієнтами базується на використанні мобільних пристроїв.

In this paper we propose the prototype implementation of the distributed information interchange system built on P2P architecture. The system itself consists of the following modules: creation of description metadata for the local data, automated schema building, automated building of the mappings and query translation. The prototype is implemented using C# language, graphic interface is built with WPF. The system is capable of integrating with the modern Web 2.0 application. Besides, the prototype can communicate with clients with mobile devices.

Ключові слова: Peer-to-Peer мережа, семантична координація, розподілена система управління знаннями.

Вступ

Значне розповсюдження цифрового контенту за допомогою Інтернет привело до пошуку альтернативних шляхів побудови систем обміну інформацією. Одним з таких підходів є Peer-to-Peer (P2P) архітектура [1]. Наразі відомі P2P системи або нав'язують примітивну семантичну структуру априорі (наприклад, Napster чи Kazaa), залишаючи непросту задачу семантичної анотації та інтеграції користувачу, або не торкаються питань семантики взагалі (наприклад, Gnutella чи практично будь-які системи, що будуються на основі Distributed Hash Table). Останні підтримують неструктуровані з семантичної точки зору подання даних і залишають інтерпретацію цих подань користувачеві.

Ця ситуація є типовою ознакою ключової проблеми в сучасних інформаційних системах – браку семантичної інтероперабельності [2], яка необхідна для структурованого розподіленого пошуку, обміну та інтеграції даних. Інтероперабельність слугує і основою створення функціональності вищого рівня та більш складної обробки даних. Класичні спроби досягнути семантичної інтероперабельності інформаційних ресурсів, зокрема в областях інтеграції баз да-

них, погано масштабуються для використання в глобальних інформаційних системах, таких як, наприклад, P2P системи [3].

Існує два основних підходи до вирішення проблем забезпечення семантичної інтероперабельності: побудова глобальних схем (онтологій) [2] або створення локальних відповідностей між різними схемами [6]. Другий підхід, як правило, досягається шляхом написання спеціалізованих трансляторів між відносно невеликим початковим набором джерел даних, які є тісно пов'язаними та мають спільне призначення. В подальшому, до початкової множини трансляторів додаються нові, що пов'язують початкові з щойно доданими джерелами. Ця практика надає основу для побудови великої системи на кшталт семантичного вебу.

Для останнього варіанту необхідна архітектура, яка дозволяє побудову системи організованих даних шляхом інкрементального додавання джерел, причому кожне нове джерело буде відповідності в першу чергу між тими джерелами, які воно вважає найбільш підходящими для цього. Це більш ефективний шлях, ніж підхід, що полягає у вимозі – одразу ж приєднанні повністю інтегруватися до мережі

шляхом побудови усієї структури відповідностей. Звісно, у випадку з семантичним вебом відповідності між джерелами мають задаватися декларативно. Також, мають бути розроблені ефективні алгоритми, які можуть проходити семантичними маршрутами (semantic paths) використовуючи дані з відповідностей для отримання даних з віддалених, але пов'язаних вузлів мережі.

Семантична координація

Побудова розподілених систем управління знаннями (PCY3) вимагає більш чіткого формулювання понять, які лежать в їхній основі. Однією з найголовніших складових таких систем є якісна реалізація процесу семантичної координації. Сутність цього процесу полягає насамперед у знаходженні та формулюванні відповідностей між різними схемами, у яких подаються дані різних учасників. Для вирішення задачі семантичної координації, досить недавно був запропонований агентний підхід [7]. Модифікуємо цю формальну модель з акцентом на використання архітектури P2P системи для опису правил, яким підкорюється процес семантичної координації у PCY3.

По-перше, необхідно мати визначення поняття схеми, яке буде незалежним від деталей реалізації [7].

Нехай L_{ext} - множина міток. Схемою S називають четвірку $\langle N, E, lab_N, lab_E \rangle$, де: (N, E) - граф; $lab_N : N \rightarrow L_{ext}$ - функція, яка приписує кожній вершині мітку з множини L_{ext} та $lab_E : E \rightarrow L_{ext} \cup \emptyset$ - функція, що або приписує кожній дузі мітку, або нічого не змінює. Тобто, множина L_{ext} використовується для побудови зовнішнього подання схеми, яка на початковий момент існує виключно у агента. Це подання буде використовуватися для обміну схемою з іншими агентами та для проведення різноманітних операцій зі схемою, в тому числі – для побудови відповідностей.

Оскільки у багатьох випадках схема і на практиці реалізована у вигляді подібного орієнтованого графу, можна такі схеми інтерпретувати як контексти. У таких схемах, фактичне значення текстових міток на вузлах (та на дугах, якщо такі є) залежить не тільки безпосередньо від свого лінгвістичного значення, а й від контексту, в якому вони використовуються. Часто контекст зображають в якості своєрідного «ящика», вміст якого – явне подання певного домену, а межі визначаються набором припущень, які робляться щодо цього явного подан-

ня. Фактично, контекст розглядають як пару $c = (R_c, A_c)$, де R_c - безпосередньо схема у вигляді графа, A_c - набір явних припущень, а саме – множина атрибутів (пар «ключ-значення»), які містять метаінформацію про контекст. Серед цих атрибутів можуть бути, наприклад, ідентифікатор мови комунікації L_{ext} , деталізація структури R_c , набір даних про домен тощо.

Далі необхідно формалізувати те, що агент a може асоціювати множину об'єктів з елементом схеми e [7], базуючись на своєму розумінні e і те, що різні агенти можуть мати різні розуміння e . Нехай a_1 та a_2 - агенти та L_{ext} - мова комунікації. Окрім, необхідно ввести поняття мови подання знань, яке відображає внутрішню мову опису знань агентів про своє оточення. Позначимо L^i мову подання знань агента a_i , W - множину можливих світів, C - множину контекстів, які використовуються. Інтуїтивно, W - множина усіх можливих інтерпретацій L^i , а C - набір різних контекстів, в яких використовуються вирази з L_{ext} . Єдина умова - L^i та L_{ext} мають бути різними мовами. Для зручності будемо вважати, що мови подання L^i є мовами декларативної логіки (description logic) [8].

Зв'язок між елементами схеми та реальними даними відбувається у два кроки: побудова подання елемента схеми у мові L^i та інтерпретація цього подання на множині W . Розглянемо ці етапи більш формально.

Нехай L^i - мова подання знань, L_{ext} - мова комунікації та c - контекст з C . Тоді функція трансляції $T_c^i : L_{ext} \rightarrow L^i$ - це функція, яка зіставляє виразу з L_{ext} , який використовується у контексті c , вираз з L^i . Ця функція формалізує той факт, що агент надає певну сутність поняттям, якими він обмінюється в процесі комунікації у певному контексті. Позначимо $T^i = \{T_c^i \mid c \in C\}$ сімейство функцій трансляції для усіх контекстів.

Нехай L^i - мова подання, $w \in W$. Тоді функція проєкції $P_w^i : L^i \rightarrow 2^w$ - це функція, яка зіставляє кожному опису поняття на мові L^i набір світів, з яким це поняття пов'язано. Позначимо $P^i = \{P_w^i \mid w \in W\}$ сімейство усіх функцій трансляції для усіх можливих світів.

Зрозуміло, що потрібно мати засоби опису і того факту, що агенти мають знання про концепції, яким зіставлені елементи схеми. Для цього в [7] введемо поняття *онтології агента*, яка є набором аксіом O^i у мові подання агента: $O^i = \{t_j \prec t_k \mid t_j, t_k \in L^i\}$. Тут використовується позначення \prec описової логіки, яке означає, що два об'єкти знаходяться у відношенні *is-a*.

Потім, в [7] дають визначення агента $a_i^{L_{ext}}$, який є користувачем певної системи обміну інформації, в якій використовується мова комунікації L_{ext} .

Агент $a_i^{L_{ext}}$ є четвіркою $\langle T^i, P^i, L^i, O^i \rangle$, де T^i - сімейство функцій трансляції, P^i - сімейство функцій проєкції, L^i - мова подання агента, O^i - знання (онтологія) агента; і виконується умова

$$\forall w \in W : t_j \prec t_k \in O^i \Rightarrow P_w^i(t_j) \subseteq P_w^k(t_j).$$

Наразі можна перейти до уточнення понять семантичної координації та відповідності між схемами. Неформально можна вважати, що два агенти є семантично координованими у використанні двох виразів мови комунікації L_{ext} при відношенні R та контексті c , якщо інтерпретація, яку вони надають цим виразам є сумісною.

Незважаючи на загальність моделі, можна показати [7], що степінь семантичної координованості демонструється трьома типами правил, які використовуються агентами чи інструментами, що проводять автоматичну побудову відповідностей. Кожне з цих правил спирається на певні відношення між різними об'єктами, що беруть участь у процесах координації (синтаксичне правило – відношення між виразами на мові комунікації, прагматичне правило – відношення між підмножинами множини світів (тобто відношення між підмножинами предметної області), семантичне правило – відношення між концепціями, які виражені на мові подання агентів) та на основі цих відношень роблять висновки про семантичну координованість агентів.

Згідно з викладеною моделлю, можна розділяти усі методи семантичної координації на три базові типи: синтаксичні, прагматичні та концептуальні. Синтаксичні методи встановлюють відповідності виключно за допомогою синтаксичного аналізу виразів, що зустрічаються у різних схемах, а саме – за допомогою порівняння властивостей рядків, які використову-

ються для маркування вузлів схеми, та аналізу причин, за яких вони були організовані у схемі [9]. Прагматичні методи – методи, які припускають, що відношення між елементами схеми може бути виведене з відношень на даних, які пов'язані з ними [10]. Концептуальні методи намагаються знайти відповідність шляхом порівняння концепції, які асоційовані з елементами схеми, тими, хто її створював [11].

В усіх цих методах є свої особливості, переваги та недоліки.

Метод автоматичної побудови відповідностей, який пропонується в даній роботі не тільки має в своєму складі елементи усіх трьох правил, зазначених вище, але й використовує четвертий тип правила, який не викладений в загальних теоретичних побудовах через свою специфічність та прив'язку до конкретної реалізації. Мається на увазі використання особливостей формату подання схеми, які включають в себе особливості вибору типу вузла (для XML це вузол елемента чи атрибуту), порівняння адресації різних елементів схеми тощо. Хоча з іншого боку можна розглядати подібний тип правила як різновид синтаксичного.

Конкретний опис використання різних правил у методі автоматичної побудови схеми буде розглянуто далі.

Архітектура та реалізація складових системи семантичної координації

Наша система, з акценту використання архітектури P2P, має складатися з таких частин: опис локальних наборів даних, що містяться у кожного піра; методи для напівавтоматичної побудови схем та/чи онтологій; побудова та підтримка наборів відповідностей між різними локальними схемами; реалізація процесу переформулювання запитів кожним клієнтом.

Одразу ж треба відмітити, що в цій роботі не приділялося уваги аспектам, які не є суттєвими для розгляду інформаційного рівня системи: вибір конкретних протоколів комунікації, мережених технологій, алгоритмів розповсюдження та кешування запитів.

Система обміну інформацією реалізована на мові C# 3.0. Складається з трьох основних блоків, що визначають базові простори імен:

- **DataShare** – містить основну функціональність: реалізацію автоматичної побудови схеми, реалізацію автоматичної побудови відповідностей, бібліотеку класів, що інкапсулюють різні типи XPath-виразів та частину, що відповідає функціональності подання та переформулювання запитів до даних, а також формування множини результатів;

• **DataShare. Communication** – містить функціональність, необхідну для комунікації між двома клієнтами; зокрема, реалізацію протоколу комунікації;

• **DataShare. UI** – графічний клієнт, що встановлюється на комп'ютері користувача та надає можливості по викладенню даних у загальний доступ пошуку; клієнт реалізований на технології WPF [12].

Локальні набори даних

Кожен користувач з точки зору рівня обміну інформації має дві основні потреби: зробити свою інформацію доступною якомога швидше, затративши на це мінімальні зусилля; мати можливість зручно та ефективно шукати інформацію інших. Для простоти, ми розглядали систему яка об'єднує користувачів-науковців, що мають потребу чи бажання поділитися інформацією як наукового (персональні чи спільні розробки, статті, презентації, медіа-матеріали), так і обчислювального характеру (тексти програм, скомпільовані бібліотеки, програмні модулі, дані для обчислювальних систем) тощо.

У роботі [13] було з'ясовано, що будуючи систему управління знаннями на основі P2P архітектури, ми стикаємося з фактом відсутності глобальної уніфікованої схеми даних, і маємо використати віртуальну глобальну схему (онтологію). Замість наявності конкретних описів та правил, які мають складати схему, віртуальна онтологія будується неявно, імпліцитно. Фундаментом для її побудови є формування наборів локальних даних, які є описаними в спосіб, зручний для користувача, тобто, описані в термінах користувача. Одним із зручних форматів такого опису є формат XML, за допомогою якого можна подавати ієрархічну структуру документів. Разом з тим, цей формат є практично незалежним від фактичної реалізації збереження даних: база даних (окремі файли та описи).

Наявність інтелектуальних описів уможливує побудову зв'язків між даними, організацію даних в великі пов'язані між собою масиви, які, з одного боку, дозволяють комплексно висвітлити проблему, а з іншого – дозволяють здійснювати ефективні запити до даних.

Головною проблемою такого способу подання описів даних в системі є забезпечення їхньої семантичної інтероперабельності. Вимагається і постійна підтримка узгодженості різних описів на рівні комунікації між користувачами – кожен має розуміти, про що йдеться.

Процес залучення користувача до системи є настільки простим, наскільки його можна було зробити. Завдяки ситуативній природі P2P

мереж, клієнт може під'єднатися до системи в будь-який час і зробити це досить просто. Як приклад можна розглянути людину, яка, маючи мобільний пристрій, хоче в певний момент під'єднатися до мережі для отримання певної інформації чи для її передачі. Можливі й більш традиційні варіанти під'єднання зі стаціонарних комп'ютерів.

Але ще перед під'єднанням існує етап, який є фундаментом усієї системи. Хоча більшість користувачів, скоріш за все, буде під'єднуватися для того, щоб просто пошукати певну інформацію, серед них будуть і такі, які бажаються поділитися своєю інформацією з іншими. Тому дуже важливим є процес опису даних, який має відбуватися за допомогою зручного інтерфейсу, зрозумілого користувачеві. Звісно, позиціонуючи систему як засіб інтелектуального обміну даними неможливо зробити процес опису настільки ж елементарним, яким він є у традиційних пірінгових мережах. Від користувачів вимагатиметься елементарні знання концепції XML та розуміння, якою саме є структура даних, що викладаються у розподілений доступ. Передбачається наявність візуального конструктора XML описів.

Автоматична побудова схем та онтологій

Першим кроком до забезпечення семантичної інтероперабельності є організація побудови фактичних схем на основі існуючих даних. Ще раз зазначимо, що поняття «схема» розуміється у широкому сенсі – сюди включається перелік сутностей домену, формалізований опис їхніх властивостей, зв'язків між ними, властивостей цих зв'язків тощо.

Фактично, в системі може бути декілька (принаймні 2) рівнів, на яких сформована абстрактна схема – рівень опису структурних та синтаксичних властивостей даних (наприклад, XML Schema) та рівень опису семантики даних (наприклад, RDF Schema).

На даному етапі побудови системи було прийняте рішення обмежитися тільки рівнем XML-схеми. Реалізація включає в себе класи `DataShare.Schema`, `SchemaTreeSubstitute` і `DataShare.Schema.SchemaTreeNode`.

Перша задача системи при приєднанні нового користувача (чи при зміні набору даних вже існуючого) – зробити аналіз його даних та побудувати по них можливу схему. Це завжди можливо, принаймні в синтаксично-структурному аспекті. Звісно, побудовані таким чином схеми не можуть бути реальними та вичерпними і використовуватися при валідації як єдиний коректний взірць. Вони лише дають максимально наближене формалізоване подання властивостей описів даних. Для XML-схеми

це можуть бути імена вузлів, ієрархічні відношення між ними, можливі типи і числові та статистичні характеристики даних у вузлах тощо. Для семантичного рівня схем можлива побудова приблизних відношень між різними сутностями в описах, певне уточнення цих відношень та їхніх характеристик. Звісно, після отримання запропонованих варіантів користувач має змогу вносити корективи для досягнення максимальної достовірності з реальними схемами даних. Крім того, можливе використання вже існуючих схем.

Побудова та підтримка наборів відповідностей

Процес інтеграції нового користувача у систему є багатоетапним і крім організації описаних локальних наборів даних включає до себе як мінімум фазу побудови наборів локальних відповідностей між схемою даного користувача та схемами певного набору інших учасників системи, що є найбільш вирішальним моментом, який забезпечує семантичну інтероперабельність. Під набором відповідностей між двома користувачами розуміється формальний опис, який у певному форматі подає, як елементи схеми одного зіставляються з елементами схеми іншого. У системі кожен вузол схеми однозначно ідентифікується за допомогою XPath-виразу [14], що описує шлях до даного елемента в XML-документі, тож відповідність між вузлами фактично реалізована у вигляді пари XPath-виразів.

Множина відповідностей між двома конкретними користувачами організується у таблицю, яка подається класом `DataShareMapping.MappingTable`.

Таблиця відповідностей має декілька функцій: виявлення степені узгодженості двох різних схем, виявлення степені координованості та зв'язності системи в цілому тощо. Але головне призначення таблиць відповідностей – їхнє використання у процесах переформулювання запитів в системі.

Побудова відповідностей загалом складається з двох етапів: знаходження схожих моментів між схемами та власне побудова відповідностей, яке засноване як на використанні цих подібностей, так і на втручанні користувачів у цей процес. Можна виділити такі основні два моменти, що забезпечують ефективну побудову відповідностей: комбінування різних ознак та використання вже побудованих відповідностей.

Комбінування різних ознак

Існує безліч ознак, за якими можна порівнювати схеми у різних евристичних та алгоритмах. Зокрема це можуть бути імена елементів схеми, типи та властивості даних, що знахо-

дяться у відповідних вузлах, додаткові текстові описи, схожості у загальній структурі схеми тощо. Тим не менш, треба враховувати те, що усі ці ознаки є досить неточними, оскільки імена можуть містити скорочення, бути синонімічними, структура може бути схожою випадково тощо.

Використання вже побудованих відповідностей

Процеси знаходження відповідностей між схемами є часто повторюваними. Наприклад, усі схеми, для яких будувалися відповідності у конкретному застосуванні, скоріш за все відносяться до одного й того ж самого домену, а отже, мають багато схожих елементів. Тож, має бути можливим повторне використання вже існуючих знань з побудованих відповідностей для формування нових. Це реалізовано за допомогою потужного Web 2.0 середовища, яке надає можливості по пошуку відповідностей, дозволяє отримувати їх повністю чи частково у визначеному форматі. Також процес повторного використання уможливується шляхом спеціальних команд у протоколі взаємодії між окремими пірамі.

Як вже було сказано, формування таблиці відповідностей і її окремих записів в загальному випадку відбувається у два етапи: автоматична побудова відповідностей з використанням евристичних методів та коригування побудованих таким чином відповідностей безпосередньо користувачем. У розширених можливостях системи, яка інтегрована з певною надбудовою для полегшення роботи в системі, координації користувачів тощо, передбачається використання, можливо частково, вже існуючих відповідностей, які теж пропонуються користувачеві для затвердження.

Для автоматичного формування відповідностей запропоновано метод, що базується на виділенні набору критеріїв, за якими порівнюються окремі елементи схеми та приписуванні їм спеціальним чином ваги. В процесі встановлення степені схожості двох окремих елементів схеми підраховується сумарна вага та з усіх можливих пар як найбільш відповідна обирається пара з максимальною сумарною вагою. Порівняння за кожною з цих ознак відноситься до певного типу методів семантичної координації. На даний момент використовуються такі ознаки схем, які порівнюються: ідентифікатор схеми (тобто XPath-вираз, який однозначно ідентифікує даний елемент) (*синтаксичний метод*); шаблон ідентифікатора схеми (ступінь схожості ідентифікаторів) (*синтаксичний метод*); ім'я елемента (*синтаксичний метод*); тип елемента (атрибут чи вузол елемента) (*синтак-*

сичний метод); базовий тип даних елементу (порожній, дерево чи текстовий) (*прагматичний метод*); можливий (додатково) тип даних елементу (рядковий, цілочисельний, часовий, логічний тощо) (*прагматичний та концептуальний метод*); кількість входжень екземплярів елементу у один батьківський (одне, багато) (*синтаксичний метод*); середня довжина імені елементу (*синтаксичний метод*); середня довжина даних елементу (для текстових елементів) / середня кількість дочірніх елементів (для елементів з типом даних дерево) (*прагматичний метод*).

Необхідна функціональність для побудови відповідності міститься у класі `DataShare.Mapping.SimilarityResolver`.

Можна використовувати й більш інтелектуальні критерії, які відносяться до типу концептуальних методів семантичної координації, як от: використання словнику синонімів для визначення схожості імен, використання таксономій та онтологій у різних доменах для виявлення схожості імен та/чи даних елементів. У випадку подання схеми як схеми онтології можливі додаткові критерії, які порівнюють зв'язки між елементами, виявляють степінь їхньої схожості, наявність схожих батьківських зв'язків тощо.

Процес переформулювання запитів

Один з головних принципів зручної та здатної до ефективної семантичної інтероперабельності агентів системи – користувач має якомога менше опікуватися тим, яким чином структуровані та описані дані інших. Тому запити у системі користувачі роблять у термінах власної схеми, але завдяки функціональності, яка забезпечує трансляцію запитів в терміни інших користувачів, пошук у системі прагне до максимальної ефективності та вичерпаності. Сам запит складається з трьох частин – умовна частина (в загальному випадку – складений XPath-вираз), частина формування результуючої множини (один простий XPath-вираз) та частина сортування результатів (один чи декілька простих XPath-виразів).

Спеціально для потреб формування та переформулювання запитів був розроблений набір класів, що дозволяє маніпуляцію з XPath-виразами, точніше, з певною їхньою підмножиною, яка є достатньою для даного випадку. Вони включають в себе інтерфейс `DataShare.XPath.IXPathExpression` такі класи з простору імен `DataShare.XPath`: `XPathExpressionPart`, `XPathExpression`, `XPathConstExpression`, `XPathUnaryExpression`, `XPathBinaryExpression`, `XPathGroupExpression`.

Інтерфейс `IXPathExpression` реалізує інтерфейс `ICloneable`, що дозволяє зручно робити переформулювання запитів рекурсивним чином.

Оскільки відповідності встановлюються між елементами схеми, кожен з яких ідентифікується власним XPath-виразом, процес переформулювання використовує це для заміни відповідних виразів в оригінальному запиті. Однак, крім заміни відбувається коригування контексту виразів, адже при переформулюванні він може змінюватися.

Отримавши запит, користувач переформулює його у термінах власної схеми, використовуючи набори відповідностей. Це реалізовано за допомогою класу `DataShare.Query.XmlQueryTransformer`.

Далі, цей запит ставиться до локального набору даних користувача, отримується певна відповідь и в залежності від неї, налаштувань системи та конкретного запиту, від або надсилається далі (вже у двох варіантах, або оригінальному користувачу надсилається коротка інформація про результати).

Комунікація користувачів

Хоч питання по комунікації висвітлюються в іншій роботі, необхідно зазначити декілька слів щодо реалізації протоколу комунікації у десктопному клієнті. Процес обміну командами інкапсулюється у класах `DataShare.Communication.Blatocol.CommunicationItem` та `DataShare.Communication.Blatocol.CommunicationReply`, в яких міститься функціональність обробки команди, виконання локальної дії чи запиту до сервера та формування відповіді із подальшим її відсиланням.

Висновок

Система, що запропонована у цій роботі є унікальною в своєму роді, оскільки на даний момент нам невідома реалізацій систем обміну інформацією на архітектурі P2P, які б використовували методи інтелектуального опису даних. Фундаментальною для побудови системи є ідея поєднання P2P архітектури з технологіями семантичного вебу, що дозволяє з одного боку підкреслити та використати сильні місця обох концепцій, а з іншого – нівелювати їхні найбільш серйозні недоліки.

Поряд із визнаними перевагами, у розподілених системах обміну інформації виникають і проблеми, однією з найголовніших з яких є проблема узгодження різних локальних схем користувачів. Запропонована реалізація методу, що виконує таку роботу. Він використовує різні кількісні характеристики, отримані після порівняння різних автоматично побудованих схем,

комбінує їх та відшукує найбільш вдалі відповідності, які пропонуються користувачеві. В подальшому є можливості по коригуванню цих відповідей, зберіганню в банку відповідей та їхньому використанню, повністю або частково, у подальшій роботі. Побудовані таким чином відповідності використовуються у процесі поновлення пошукових запитів у системі.

Ця робота є ще одним кроком, спробою активізації практичних впроваджень архітектури P2P.

Список літератури

1. Mika P.A Methodology for Distributed Knowledge Management Using Ontologies and Peer-to-Peer, Semantic Web And Peer-To-Peer /Mika P. – 2006. – P. 283-300.

2. Aberer K. Semantic Gossiping: Fostering Semantic Interoperability In Peer Data Management Systems, Semantic Web And Peer-To-Peer /Aberer K., Cudre-Mauroux P. and Hauswirth. – M., 2006, – P. 259-275.

3. Wiederhold G. Mediators in the Architecture of Future Information Systems / Wiederhold G. – IEEE Computer 25(3) – 38-49 p.

<http://ieeexplore.ieee.org/Xplore/login.jsp?url=http://ieeexplore.ieee.org/iel1/2/3459/00121508.pdf%3Farnumber%3D121508&authDecision=-203> (May 1, 2008).

4. Hull R. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective, Symposium on Principles of Database Systems (PODS97) / Hull R. – 1997. – 11p. <http://w3.msi.vxu.se/~per/IVC743/p51-hull.pdf> (May 1, 2008).

5. Ouksel A. Ontologies are not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction /Ouksel A., Ahmed I. – Distributed and Parallel Databases 7(1). –1999. – 7-35p. <http://www.springerlink.com/content/w811m5x7765t2648/> (May 1, 2008).

6. Tatarinov I. The Piazza Peer Data Management Project /Tatarinov I., Ives Z., Madhavan J., Halevy A., Suciu D., Dalvi N., Dong X., Kadiyska Y., Mikalu G., Mork P. – 2006. – 47-52p.

<http://portal.acm.org/citation.cfm?doid=945721.945732> (May 1, 2008).

7. Bouquet P. Contexts and Ontologies in Schema Matching / Bouquet P. – 2004. – 2-10p.

8. Baader F. The Description Logic Handbook. Theory, Implementation and Applications /Baader F., Calvanese D., McGuinness D., Nardi D. Patel-Schneider P. – Cambridge University Press.– 2007. – 622 p.

9. Carroll J. Matching RDF graphs / Carroll J. – Proceedings in the first International Semantic Web Conference (ISWC 2002). – 2002. – 3-15p.

10. Doan A. Reconciling schemas of disparate datasources: A machine-learning approach /Doan A., Domingos P., Halevy A. Y. – SIGMOD Conference. – 2001. –1-11 p.

11. Oram A. Peer-to-Peer: Harnessing The Power of Disrupting Technologies / Oram A. – O'Reilly. – 2001. – 430 p.

12. Windows Presentation Foundation. <http://msdn.microsoft.com/enus/library/ms754130.aspx>. (May 1, 2008).

13. Глибовець М.М. Розподілене управління знаннями на основі архітектури Peer-to-Peer / Глибовець М.М. Жигмановський А.А. – «Інженерія програмного забезпечення. – №1. – НАУ. – 2012. – 59-65 с.

14. Berglund A. XML Path Language (XPath) 2.0 / Berglund A., Boag S., Chamberlin D., Fernandez M. F., Kay M., Robie J., Simeon J. – W3C Recommendation. – 2007. <http://www.w3.org/TR/xpath20/> (May 1, 2008).

15. Hummel J. Business models and system architectures of virtual communities. From a sociological phenomenon to peer-to-peer architectures / Hummel J., Lechner, U. – International Journal of Electronic Commerce 6(3). – 2002.

<http://portal.acm.org/citation.cfm?id=1286964> (May 1, 2008).

16. Stuckenschmidt H. Peer-to-Peer and Semantic Web / Stuckenschmidt H., van Harmelen, F., Siberski W., Staab S. – Semantic Web And Peer-To-Peer. – 2006. – 1-21 p.

Відомості про авторів



Глибовець Микола Миколайович - доктор фіз.-мат. наук, професор, декан факультету інформатики Національний університет "Києво-Могилянська Академія". Напрямок наукових інтересів – розподілені системи інтелектуального типу, програмні системи підтримки електронного навчання.
E-mail: glib@ukma.kiev.ua



Жигмановський Андрій Андрійович - магістр комп'ютерних наук, напрям наукових інтересів - автоматизація бізнес процесів, розподілені системи, семантичний веб.
Email: andrew.zhygmanovsky@gmail.com

Стаття надійшла до редакції 23.11.2011 р.