УДК 004.6

Мультиагентна система інтеграції розподілених баз даних

В. Ю. Письменний[1], М. Ю. Терновой[2]
[1]*Національний технічний університет України «Київський політехнічний інститут»*,
[2]*Інститут телекомунікаційних систем, Україна*

В статті запропоновано математичну модель мультиагентної системи інтеграції баз даних в розподіленому гетерогенному середовищі, яка дозволяє проводити інтеграцію даних на рівні доступу до них. Розглянуто процес побудови та прототип мультиагентної системи. Запропонована в роботі математична модель мультиагентної системи надає формальний апарат для побудови розподіленої інформаційної системи на основі агентних технологій. Розглянутий підхід до реалізації дозволяє отримати досить гнучку, платформо незалежну та стійку до змін в інформаційному середовищі систему. Реалізований прототип мультиагентної системи показав адекватність моделі та можливість її використання в реальних розподілених системах.

Ключові слова: *мультиагентна система, система керування базами даних, програмний агент, інформаційний простір.*

В статье предложена математическая модель мультиагентной системы интеграции баз данных в распределенной гетерогенной среде, которая позволяет проводить интеграцию данных на уровне доступа к ним. Рассмотрен процесс построения и прототип мультиагентной системы.

Ключевые слова: *мультиагентная система, система управления базами данных, программный агент, информационное пространство.*

This paper proposes mathematical model of multi-agent system for databases integration in the distributed heterogeneous environment. Proposed multi-agent system allows user to integrate a data stored in databases at access level. Process of construction and a prototype of multi-agent system are described.

Key words*: multi-agent system, database management system, the program agent, an information environment.*

1. Загальна постановка задачі та її актуальність

На сучасному рівні розвитку інформаційних технологій існує багато способів організації доступу до розподілених ресурсів, а також розподіленого доступу до єдиного інформаційного ресурсу. Це як і системи керування базами даних (СКБД), на основі яких можна побудувати розподілену базу даних та управляти і користуватись нею з використанням вбудованого в СКБД програмного забезпечення, так і окремі програмні рішення, часто базовані на веб-технологіях, які створені незалежно від СКБД і можуть працювати з різними їх типами. Зазвичай таке програмне забезпечення зв‘язується з конкретною базою даних за допомогою спеціальних шлюзів підключення або драйверів та дозволяє управляти та користуватись базою даних в незалежності від її типу.

Зважаючи на стрімкий розвиток інформаційних технологій та швидкість виникнення нових і удосконалення вже існуючих технологій обробки та обміну інформацією часто виникає необхідність в оновленні існуючих корпоративних інформаційних систем до належного рівня для покращення швидкості та автоматизації обробки та обміну інформацією, що в кінцевому результаті потребує значних фінансових витрат та часто повної реконфігурації та заміни програмного забезпечення існуючої інформаційної системи. Це обумовлює

зростання вірогідності виникнення проблем в роботі організації, які, як правило, з'являються при переході на нові програмні засоби, що обумовлюється довготривалістю такого процесу та необхідністю перенесення даних до нових інформаційних ресурсів.

В даній роботі запропоновано математичну модель та прототип мультиагентної системи, яка може бути використана як посередник між різними типами баз даних в розподіленому інформаційно-телекомунікаційному середовищі і здійснювати інтеграцію до баз даних на рівні доступу до них. Такий підхід не вимагає зміни існуючої інфраструктури, що зменшує фінансові витрати та час впровадження.

На даний час, агентні технології є одним з найбільш перспективних напрямків для інтеграції інформаційних ресурсів. Така тенденція спостерігається тому, що агентні технології пропонують високорівневу абстракцію програмного забезпечення, яке необхідне для управління складними програмними комплексами, а також тому, що вони були розроблені, щоб підтримувати розподіленість та інтероперабельність [1].

В роботах [2, 3] запропоновано використовувати мультиагентні системи для вирішення задач об'єднання інформаційних ресурсів. Однак в них більше уваги приділено саме реалізації мультиагентної системи без розробки відповідної математичної бази. В роботі [4] запропоновано математичну модель інформаційного середовища з використанням агентних технологій, однак вона описана в загальному вигляді без врахування особливостей збору інформації з СКБД та вимог, що були пізніше висунені до агентних систем FIPA [5].

Отже, актуальною є задача розробки математичної моделі мультиагентної системи для об'єднання на рівні доступу до даних баз даних, що є частинами існуючих автоматизованих систем, яке не буде вимагати зміни цих систем та враховувати сучасні вимоги до агентних систем.

2. Модель мультиагентної системи

При розробці мультиагентної системи (МАС) слід врахувати вимоги, які обумовлені задачею, а саме:

- об'єднання на рівні доступу розподілених баз даних;
- територіально розподілений характер корпоративного середовища;
- необхідність взаємодії з різними типами СКБД, в тому числі і настільними СКБД;
- необхідність підтримки масштабованості та гнучкості щодо змін в структурі.

Інформаційний простір можна представити у вигляді абстрактної алгебраїчної системи [4]:

$$E = < O, S, \Omega >, \qquad (1)$$

де O – об'єкти інформаційного простору;

S – відносини між об'єктами O;

Ω – операції маніпулювання об'єктами в просторі E.

Для задачі, що розглядається, у якості об'єктів моделі (1) будуть виступати компоненти сучасної інформаційної системи – різні типи сховищ даних, в тому числі реляційні бази даних, $O = \{db_n \mid n = \overline{1, N_O}\}$. Відношення $S = \{s_n \mid n = \overline{1, N_S}\}$

між об'єктами інформаційного простору визначає конкретну конфігурацію обчислювального середовища E, орієнтовану на конкретного користувача або користувачів, $G = \{g_n \mid n = \overline{1, N_G}\}$ – множина користувачів.

Залежно від цілей, поставлених перед користувачем і обчислювальним середовищем, усі дії виконуються на основі операцій $\Omega = \{q_n \mid n = \overline{1, N_\Omega}\}$. Прикладом множини елементарних операцій $\Omega = \{q_1, q_2, .., q_n\}$ можуть бути такі операції маніпулювання даними в базах даних, як $\{< \text{select} >, < \text{insert} >, < \text{update} >, < \text{delete} >\}$.

Для забезпечення доступу до інформації та обміну інформацією в інформаційному просторі запропоновано використовувати мультиагентну систему, математичне представлення якої має такий вигляд:

$$U_A =< A, S_A, \Omega_A >, \quad (2)$$

де A – множина програмних агентів;

S_A – відносини в мультиагентній системі;

Ω_A – множина операцій, які можуть виконувати агенти.

Відносини S_A в мультиагентній системі (2) діляться на взаємовідносини S_{AA} між самими агентами та відносини S_{AE} між агентами A та об'єктами інформаційного простору: $S_A = S_{AA} \cup S_{AE}$, $S_{AA} \cap S_{AE} = \varnothing$.

Для мультиагентної системи, яка не прив'язана до конкретної конфігурації інформаційного простору $S_{AE} = \varnothing$.

Визначимо поняття мультиагентного простору як розширення інформаційного простору. Тоді модель мультиагентного простору можна представити у вигляді (3):

$$E_A = E \cup U_A =< O_g, S_g, \Omega_g >. \quad (3)$$

При цьому множину об'єктів інформаційного простору можна представити у вигляді $O_g = O \cup A$.

Множина відносин $S_g = S \cup S_A = S \cup S_{AA} \cup S_{AE}$, при чому $S_{AE} \neq \varnothing$. Множина операцій $\Omega_g = \Omega \cup \Omega_A$.

В мультиагентному просторі всі операції над об'єктами в довільні моменти часу $t_1, t_2, ..., t_m$, $t_{m+1} > t_m$ ініціює користувач g_n мультиагентного простору на основі формалізованого плану дій $z^*_{n1}(t_1), z^*_{n2}(t_2), ...z^*_{nm}(t_m)$ або унітарної дії $z^*_1(t)$. Взаємодія користувача з мультиагентним простором реалізуються через агента користувача a_C, який взаємодіє з відповідними системними агентами $A_{S_i} = \{a_{S1}, a_{S2}, .., a_{Sn}\}$, кожен з яких зіставлений з деяким інформаційним ресурсом простору E. Множина агентів A мультиагентного простору буде складатись з множини агентів користувача A_C та множини системних агентів A_S:

$$A = A_C \cup A_S$$

Оскільки, модель мультиагентного простору, що розглядається, не передбачає взаємодію між однотипними агентами, то компонент відносин в мультиагентній системі матиме наступний вид (4):

$$S_A = S_{CS} \cup S_{SE}, \tag{4}$$

де S_{CS} – відносини між різними типами агентів (агент-агент);

S_{SE} – відносини між агентами множини A_S та об'єктами інформаційного простору (агент-система).

Відповідно, множина операцій матиме вигляд (5):

$$\Omega_A = \Omega_{CS} \cup \Omega_{SE}, \tag{5}$$

де Ω_{CS} – множина операцій, які виконують агенти у відносинах S_{CS};

Ω_{SE} – множина операцій у відносинах S_{SE}, для доступу агентів множини A_S до об'єктів інформаційного простору.

Формалізована задача z_i, що поставлена користувачем, перетворюється агентом користувача a_C у послідовність команд, що відсилаються системним агентам A_{S_i}, та наступної обробки інформації після отримання відповіді від агентів A_{S_i}. Наприклад, задача $z_i =< find\_data\_in\_database >$, яку ставить користувач, буде реалізована послідовністю дій агента користувача q_{a_C} (6) та системних агентів q_{A_S} (7), та показана на рис. 1.

$$q_{a_C} = q_{a_C 1} \cup q_{a_C 2}, \tag{6}$$

де $q_{a_C 1} = (\text{get\_search\_parameters}, \text{search\_available\_agents}, \text{form\_query}, \text{send\_query})$;

$q_{a_C 2} = (\text{collect\_all\_results}, \text{merge\_results}, \text{show\_final\_result})$.

$$q_{A_S} = q_{A_S 1} \cup q_{A_S 2}, \tag{7}$$

де $q_{A_S 1} = (\text{receive\_query}, \text{execute\_query}, \text{get\_result})$,

$q_{A_S 2} = (\text{form\_result}, \text{send\_result})$.

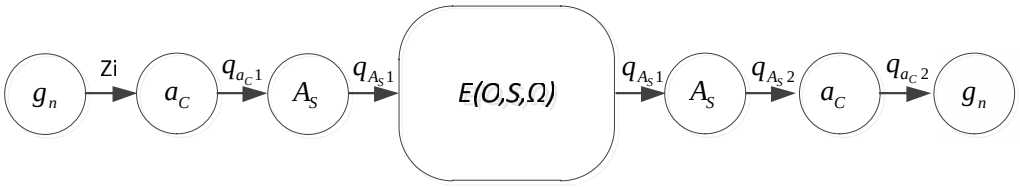

Рис. 1. Схема взаємодії користувача з мультиагентним простором

4. Реалізація системи

Як вказано вище, в системі використовується два типи агентів a_C та a_S. В реалізації назвемо агента a_C – MainClientAgent, основна задача якого є взаємодія з користувачем, передача запитів до та отримання і консолідація інформації від групи агентів A_{S_i}. Агента a_S назвемо ListenerAgent – це

системний агент, основне завдання якого є взаємодія з базою даних на вузлах системи і передача даних агенту MainClientAgent у відповідь на його запит. На кожному з вузлів інформаційного простору, на якому знаходиться база даних до якої необхідно мати доступ, має бути встановлений один ListenerAgent. MainClientAgent встановлюється на тих вузлах, з яких необхідно надати доступ користувачу до розподілених даних.

За специфікацією FIPA [5] реалізація мультиагентної системи передбачає наявність додаткового агента в системі, агента Directory Facilitator (DF), який впроваджує сервіс «жовтих сторінок». За допомогою цього сервісу агенти системи можуть дізнатись, які ще агенти є в цій системі і які сервіси вони пропонують.

Після включення агентів, всі агенти ListenerAgent, які є в системі, реєструються в сервісі «жовтих сторінок», тобто у DF. Це потрібно для того, щоб MainClientAgent при надсиланні запитів міг мати актуальну інформацію про існуючих на даний момент активних агентів в системі. В даній реалізації ListenerAgent пропонує сервіс доступу до бази даних, а MainClientAgent використовує цей сервіс для вирішення задач поставлених користувачем. Взаємодія між агентами відбувається за допомогою мови Agent Communication Language (ACL), стандартизованої FIPA [6].

Запити на отримання даних формуються агентом MainClientAgent у вигляді запиту стандартизованою мовою SQL і передаються до агентів ListenerAgent, а ListenerAgent використовує цей запит для отримання даних з БД. Використання SQL, як основної мови запитів до БД в переважній більшості сучасних реляційних СКБД дозволяє виключити зі списку обмежуючих параметр вибору методу отримання даних від різних типів СКБД при побудові та налаштуванні мультиагентної системи.

Алгоритми роботи агентів показані на блок-схемах, зображених на рис. 2, а для MainClientAgent та на рис. 2, б для ListenerAgent.

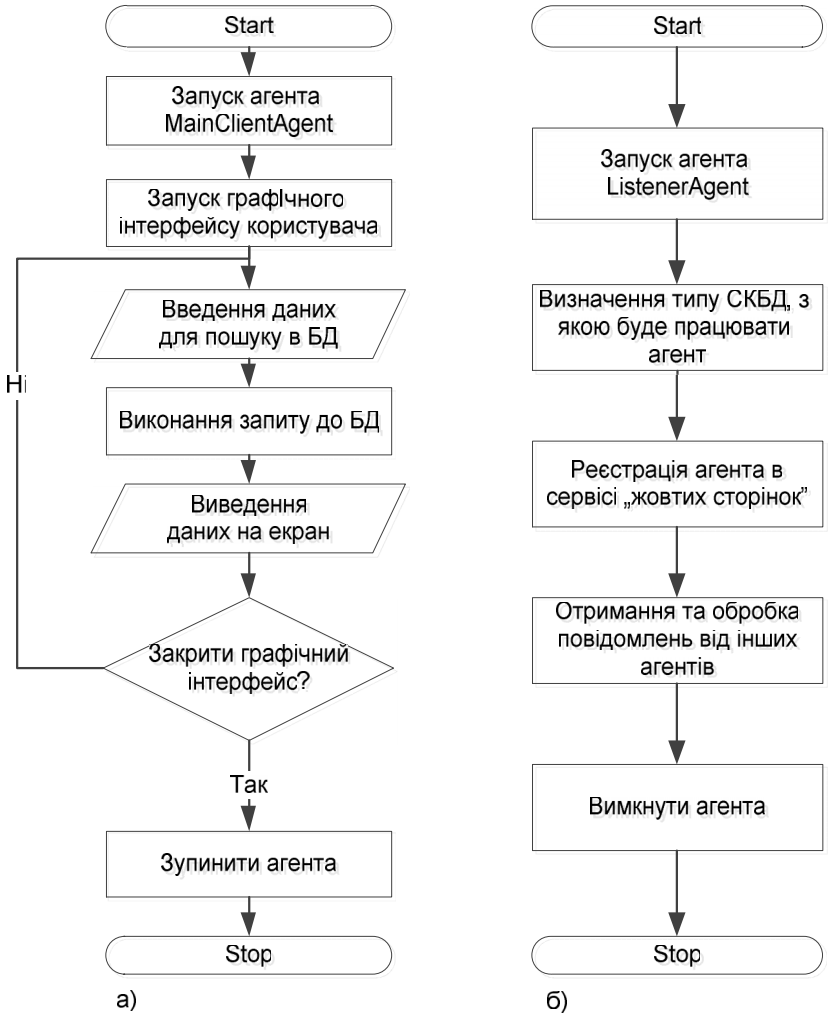

Рис. 2. Алгоритм роботи агентів мультиагентної системи: а) MainClientAgent; б) ListenerAgent

На рис. 3 показано алгоритм створення запиту та обробки вхідних повідомлень агентом MainClientAgent.

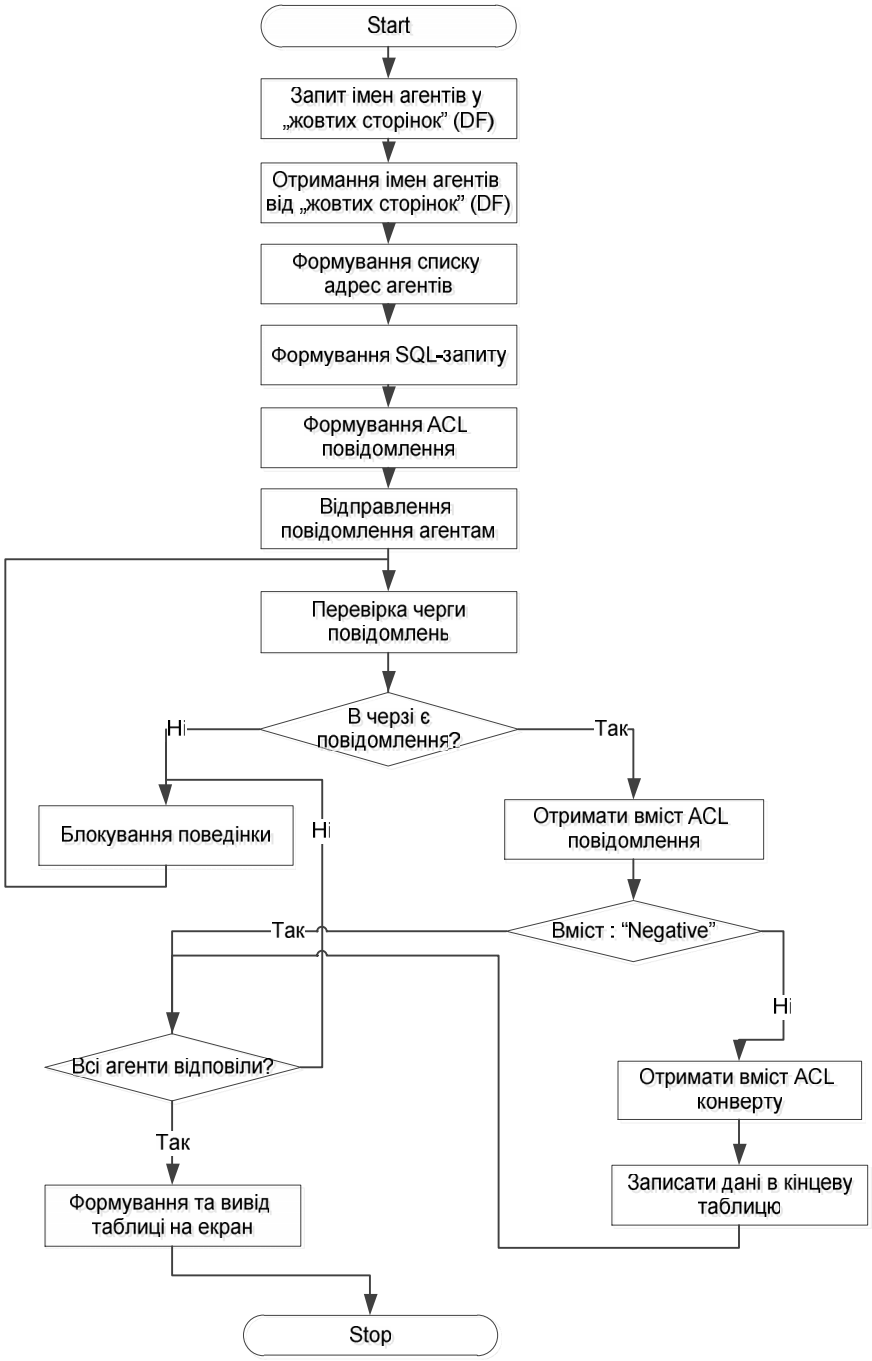

Рис. 3. Внутрішній алгоритм роботи MainClientAgent

Алгоритм обробки повідомлень агентом ListenerAgent показано на рис. 4.

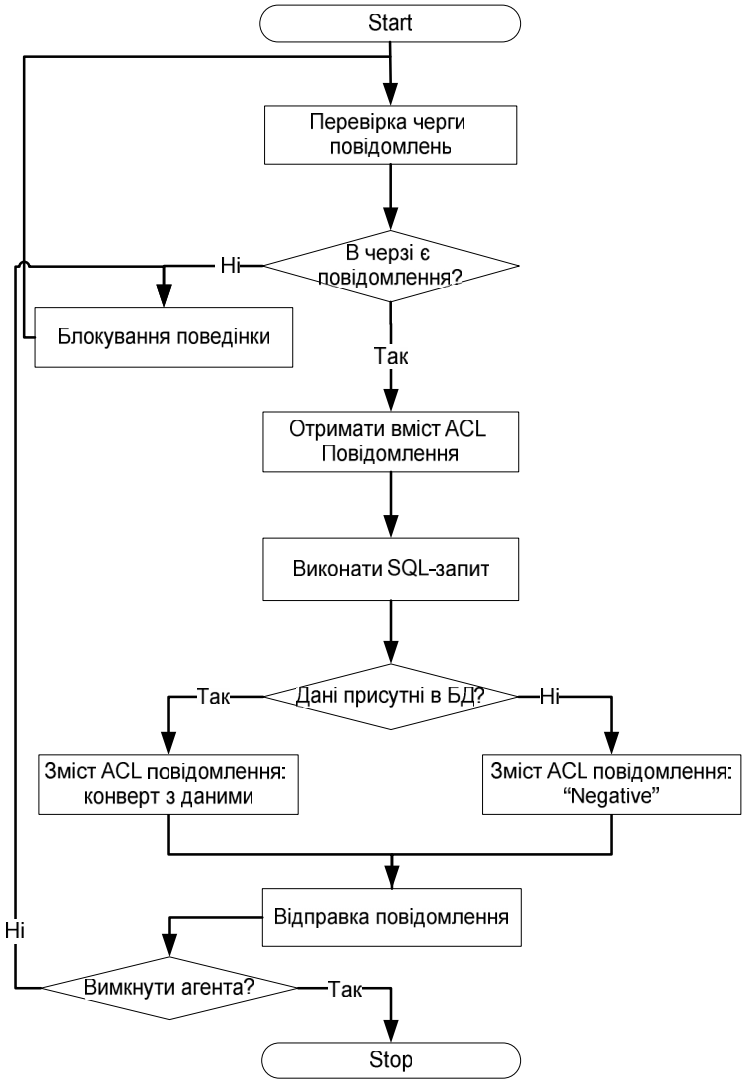

Рис. 4. Внутрішній алгоритм роботи ListenerAgent

5. Прототип мультиагентної системи

Розробка прототипу запропонованої мультиагентної системи базується на методології описаній у [7], яка використовує принципи побудови мультиагентних систем описаних у специфікаціях FIPA [5].

Для запропонованої мультиагентної системи було розроблено діаграму прецедентів, яка зображена на рис. 5.

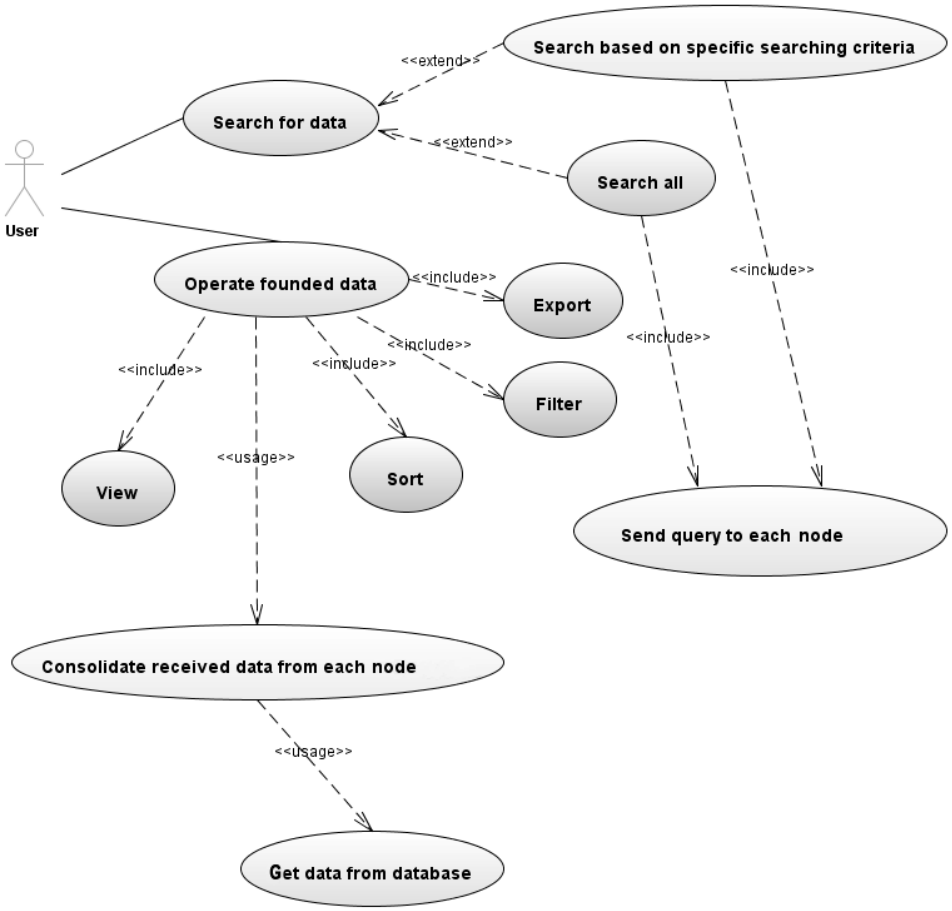

Рис. 5. Діаграма прецедентів (Use Case diagram)

Функціональний опис прецедентів, показаних на рис. 5 наведено нижче.

Користувачі:

User – людина, співробітник організації, яка напряму взаємодіє з системою.

Прецеденти:

Search for data – активується, коли користувач намагається зробити пошук інформації.

Search based on a specific searching criteria – пошук інформації по заданим критеріям.

Search all – отримання всіх даних, які є в кожній з баз даних.

Send query to each node – надсилання запиту на отримання інформації з бази даних кожному вузлу.

Operate founded data – проведення операцій з отриманими даними.

Consolidate received data from each node – консолідація даних отриманих від кожного з вузлів.

View – перегляд отриманих даних.

Sort – сортування отриманих даних.

Filter – фільтрація отриманих даних.

Export – експорт отриманих даних до файлової системи.

Get data from database – отримання інформації безпосередньо з бази даних, як результат запиту до БД.

Для чіткого розуміння схеми взаємодії між об‘єктами в мультиагентній системі було складено діаграму агентів.

Потрібно відзначити, що в діаграмі агентів, на відміну від UML Use Case діаграм, відмінність зроблена явно між людьми й зовнішніми системами.

Діаграма агентів для запропонованої мультиагентної системи обміну інформацією показана на рис. 6.

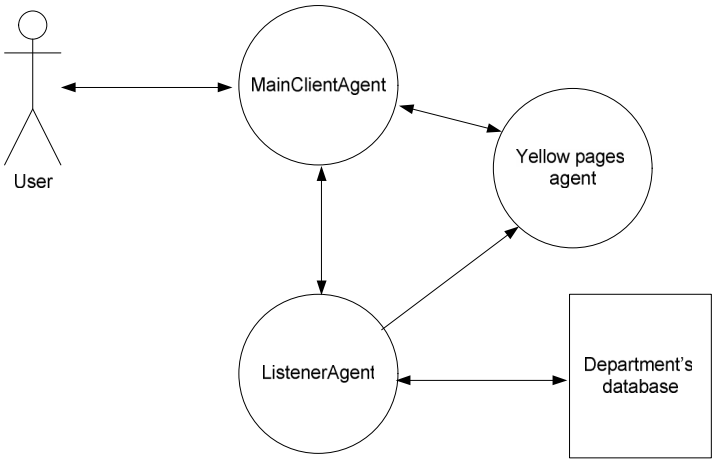

Рис. 6. Діаграма агентів

Для створення запропонованої мультиагентної системи, було обрано агентну платформу та інструментарій розробки агентів Java Agent Development Framework (JADE), який створено згідно специфікацій FIPA [6, 8] та побудовано на мові Java [9]. Використання платформи JADE для вирішення поставленої задачі додатково обумовлюється тим, що Java дозволяє використовувати драйвери JDBC для з‘єднання з базою даних, що в свою чергу дозволяє агенту, побудованому на платформі JADE, бути незалежним від типу СКБД, яка використовується в системі.

Тестування системи проводилось з використанням тестової бази даних на мережі створеній за допомогою віртуальних машин, яка мала конфігурацію показану на рис. 7. Платформа JADE була встановлена на окремому вузлі D. На вузлах A, B і C розміщені бази даних з різними типами СКБД: Microsoft Access для вузлів A і C та Oracle на вузлі B. На цих же вузлах було розміщено по одному агенту ListenerAgent з іменами «ListenerAgent A», «ListenerAgent B», «ListenerAgent C» відповідно. На окремому вузлі E було запущено агента MainClientAgent.

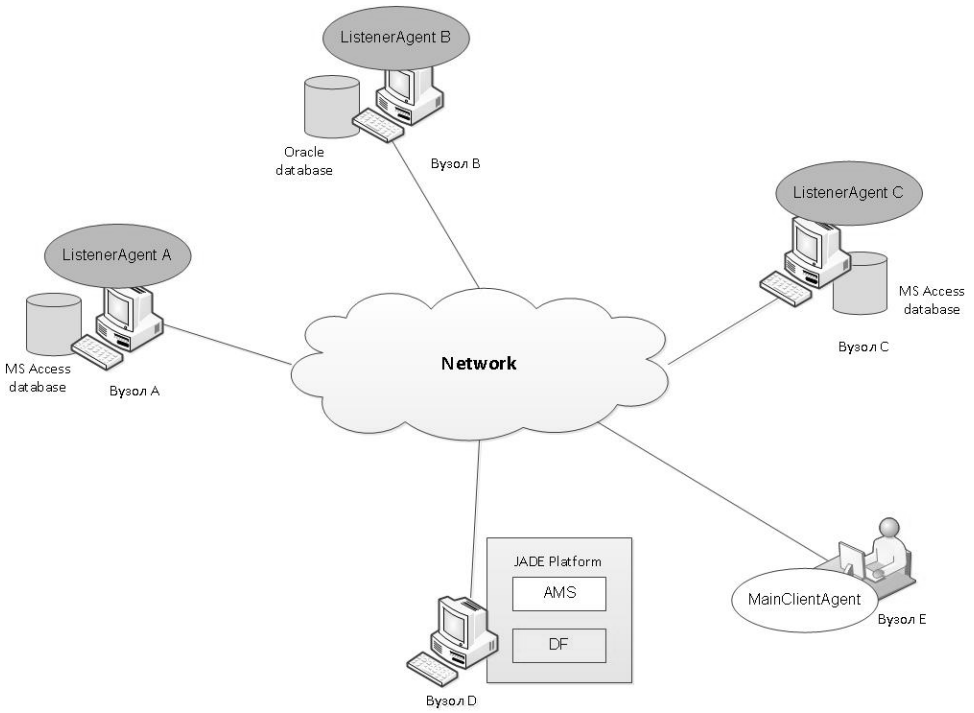

Рис. 7. Конфігурація, яка була використана для тестування роботи мультиагентної системи

За допомогою вбудованих інструментів управління та тестування в JADE можемо прослідкувати які агенти зареєстровані на платформі та які з них зареєстрували свої сервіси в «жовтих сторінках» (рис. 8).

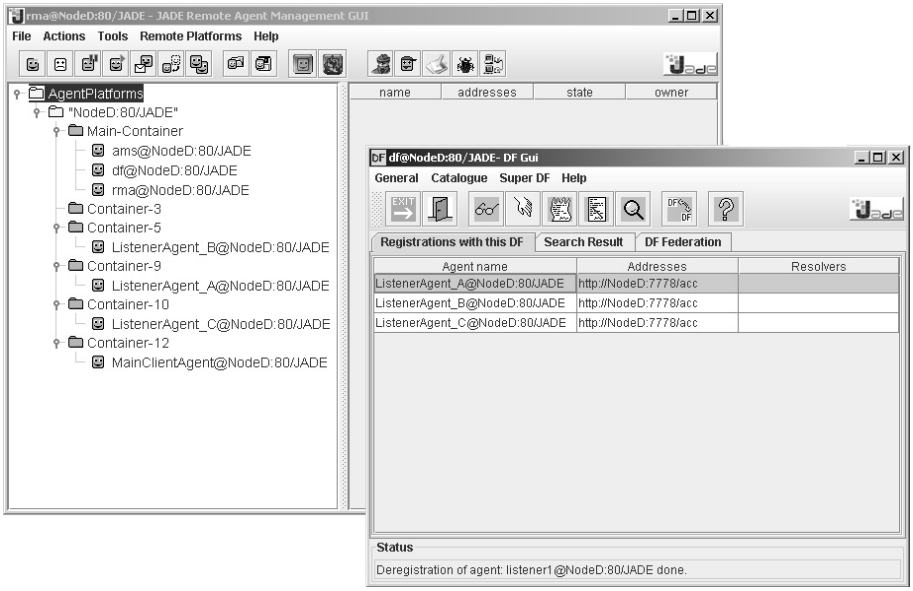

Рис. 8. Перегляд запущених та зареєстрованих агентів в системі

Використовуючи Sniffer агента вбудованого в інструментарій JADE можна побачити процес обміну повідомленнями між агентами в тестовій мультиагентній системі (рис. 9).

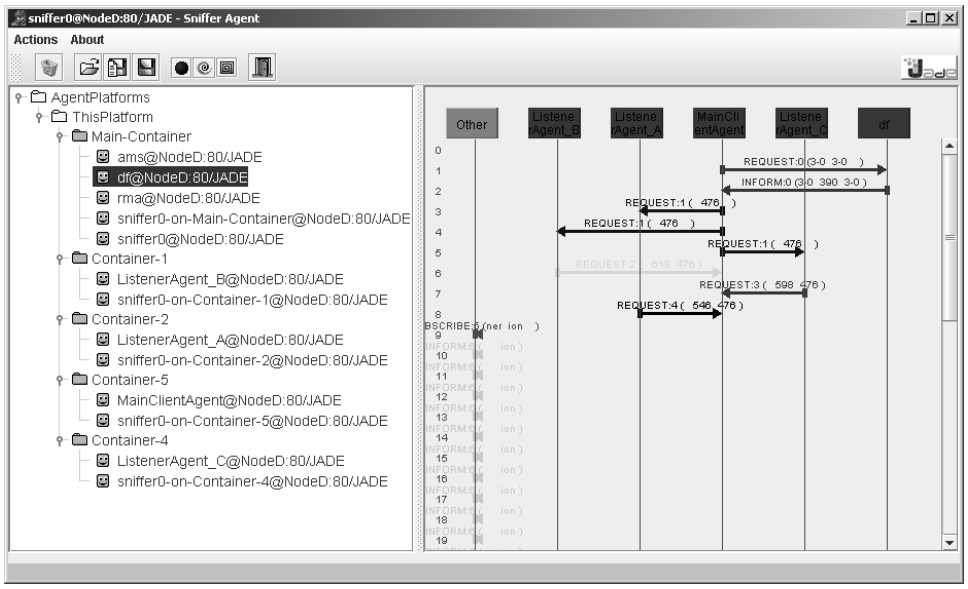

Рис. 9. Перегляд обміну повідомленнями між агентами використовуючи Sniffer агента

На рис. 9 можна побачити, що кожен з віддалених агентів розміщується на платформі в окремому віртуальному контейнері. На діаграмі послідовності, яка розміщується на правій панелі вікна Sniffer агента видно послідовність взаємодії агентів в системі під час тестування. При отриманні запиту від користувача MainClientAgent в першу чергу збирає інформацію про всіх агентів ListenerAgent, які готові отримувати від нього запити, звертаючись за списком наявних агентів до сервісу «жовтих сторінок» (агент DF). Отримавши список агентів, MainClientAgent надсилає запит всім агентам зі списку та чекає доки кожен з них дасть відповідь, після чого консолідує отримані дані (якщо дані були знайдені) і виводить їх користувачу. Детальніше процедура обміну інформацією між агентами була описана вище.

Система легко адаптується до змін в середовищі. Завдяки сервісу «жовтих сторінок» агент користувача завжди отримує актуальні дані про наявних в системі активних агентів – агент DF веде постійне спостереження за статусом зареєстрованих агентів. При відключенні зареєстрованого в «жовтих сторінках» агента, він автоматично знімається з реєстрації. У випадку, якщо в системі з’являється новий вузол, його підключення до системи не потребує багато зусиль та в основному полягає в правильній конфігурації підключення агента до СКБД.

6. Висновки

Запропонована в роботі математична модель мультиагентної системи надає формальний апарат для побудови розподіленої інформаційної системи на основі агентних технологій. Розглянутий підхід до реалізації дозволяє отримати досить гнучку, платформо незалежну та стійку до змін в інформаційному середовищі систему. Реалізований прототип мультиагентної системи показав адекватність моделі та можливість її використання в реальних розподілених системах.

ЛІТЕРАТУРА

1. Developing Multi-agent Systems with JADE. [Електронний ресурс]. – Режим доступу: http://www.imamu.edu.sa/dcontent/IT_Topics/java/bellifemine01.pdf
2. Письменный В. Ю., Терновой М. Ю. Организация доступа к гетерогенным базам данных с использованием агентных технологий // Международный научный журнал "Компьютинг". – 2011. – Т.10, Вып. 2. – С. 183-191.
3. Paes Leme L. A. P., Marco A. Casanova, Karin K. Breitman, Antonio L. Furtado. Database Mediation Using Multi-agent Systems // Proceeding SEW '08 Proceedings of the 2008 32nd Annual IEEE Software Engineering Workshop.
4. Филатов В. А. Мультиагентные технологии интеграции гетерогенных информационных систем и распределенных баз данных : Дис. д-ра техн. наук: 05.13.06 / Харьковский национальный ун-т радиоэлектроники. – Х., 2004. – 341л. : рис. – Библиогр.: л. 313-336.
5. FIPA Agent Management Specification. [Електронний ресурс]. – Режим доступу: http://www.fipa.org/specs/fipa00023/SC00023K.html#_Toc75951006 (2.04.2011).
6. FIPA Specifications. [Електронний ресурс]. – Режим доступу: http://fipa.org/specifications/index.html (31.03.2011).
7. A Methodology for the Analysis and Design of Multi-Agent Systems using JADE. [Електронний ресурс]. – Режим доступу: http://jade.tilab.com/doc/tutorials/JADE_methodology_website_version.pdf
8. Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. JADE: A FIPA-compliant agent framework. In Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM), pages 97–108, 1999.
9. Introduction. JADE. [Електронний ресурс]. – Режим доступу: http://jade.tilab.com/doc/html/intro.htm

Надійшла – 13.03.2012.