



ОРГАНІЗАЦІЯ ДОСТУПУ ДО ГЕТЕРОГЕННИХ БАЗ ДАНИХ З ВИКОРИСТАННЯМ АГЕНТНИХ ТЕХНОЛОГІЙ

В'ячеслав Письменний, Максим Терновой

Національний технічний університет України "Київський політехнічний інститут"
Індустріальний пров., 2, Київ, 03056, Україна,
e-mail: vpysmennyi@gmail.com, maximter@mail.ru

Резюме: В роботі запропоновано підхід до реалізації доступу до інформації в розподіленому інформаційно-телекомунікаційному середовищі з гетерогенних баз даних з використанням мультиагентних систем. Розглянуто різні варіанти створення мультиагентної системи. Розроблено схему взаємодії між агентами для реалізації доступу до інформації з гетерогенних баз даних. Запропонований підхід націлений на підвищення ефективності роботи з даними з гетерогенних інформаційних систем без необхідності їх зміни.

Ключові слова: бази даних, розподілена система, агент, мультиагентна система, FIPA, JADE.

AGENT BASED ACCESS TO HETEROGENEOUS DATABASES

Vyacheslav Y. Pysmennyi, Maksym Y. Ternovoy

National Technical University of Ukraine "Kyiv Polytechnic Institute"
Industrialnyy Al., 2, Kyiv, 03056, Ukraine,
e-mail: vpysmennyi@gmail.com, maximter@mail.ru

Abstract: The paper presents the implementation of system for access to the information in distributed information-telecommunication environment with heterogeneous databases using multi-agent approach. Different ways for multi-agent system construction were considered. It was developed a scheme of interaction between agents in the proposed system. The approach aims to improve the efficiency of data access in heterogeneous information systems without the necessity to change them.

Keywords: databases, distributed system, multi-agent system, FIPA, JADE.

1. ВСТУП

Інтеграція інформаційних і обчислювальних ресурсів у єдине середовище й організація доступу до них є одним з найважливіших напрямків розвитку сучасних інформаційних технологій. Стрімкий розвиток глобальних інформаційних і обчислювальних мереж веде до зміни парадигм збору, збереження, обробки та доступу до даних. При цьому спостерігаються тенденції до винятково розподіленої схеми створення, підтримки й зберігання ресурсів [1]. У той же час існує прагнення до віртуального об'єднання інформаційних ресурсів на рівні надання доступу до них [2,3].

Найчастіше інформаційні ресурси зберігаються в реляційних базах даних (БД) різних гетерогенних автоматизованих систем [4]. Це обумовлює складність, а в деяких випадках неможливість, об'єднання таких баз даних в одну логічну базу даних за допомогою стандартних засобів, що надаються системами керування базами даних (СКБД) [5]. З іншого боку, існуючі підходи до інтеграції інформаційних ресурсів таких систем вимагають переробки або зміни існуючого програмного забезпечення [5,6], що потребує значних фінансових витрат. Отже, актуальною є задача об'єднання на рівні доступу до даних баз даних, що є частинами існуючих автоматизованих систем, яке не буде вимагати зміни цих систем.

2. ОСНОВНА ЧАСТИНА

Поставлену задачу в загальному випадку неможливо вирішити за допомогою монолітної системи. Рішення цієї задачі, яке пропонується в даній роботі, полягає в розробці підходу до створення мультиагентної системи (МАС), яка утворена групою взаємодіючих інтелектуальних агентів [7]. Основна функціональність системи, що розробляється, полягатиме в наданні доступу до інформації з різних баз даних в розподіленому середовищі. Однією з головних переваг МАС є гнучкість, оскільки мультиагентна система може бути доповнена і модифікована без зміни значної частини програми. Також мультиагентні системи мають здатність до самовідновлення і володіють стійкістю до збоїв завдяки достатньому запасу компонентів та самоорганізації. Серед основних характеристик компонентів такої системи, агентів, можна виділити [8]: автономність, обмеженість уявлення агентів про структуру всієї системи та децентралізацію керування. В якості середовища функціонування МАС використовується агентна платформа, яка є базовим інструментом розробки МАС та дозволяє створювати, знищувати, інтерпретувати, запускати і переміщати агентів. Агенти, які реєструються на платформі можуть взаємодіяти з вже зареєстрованими агентами на тій же платформі. Також є можливість взаємодії агентів зареєстрованих на різних платформах, якщо реалізована схема взаємодії між самими платформами.

Згідно специфікації Foundation for Intelligent Physical Agents (FIPA) [9], на агентній платформі мають бути також реалізовані такі компоненти: Agent Management System (AMS), Message Transport Service (MTS) і Directory Facilitator (DF). AMS веде диспетчерський контроль за доступом та використанням платформи. MTS реалізує механізм передачі повідомлень між агентами. DF надає сервіс жовтих сторінок іншим агентам. Сервіс жовтих сторінок дозволяє будь-якому агенту, зареєстрованому на платформі, отримати дані щодо сервісів, які надаються іншими агентами.

При розробці МАС слід врахувати вимоги, які обумовлені задачею:

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.

В процесі вирішення поставленої задачі було запропоновано два архітектурні підходи до побудови МАС: багатоплатформовий та

одноплатформовий.

Багатоплатформовий підхід, включає в себе встановлення на кожному вузлі розподіленої системи окремої агентної платформи. Локальний для окремого вузла агент буде приєднуватись до платформи, яка запущена на тому ж вузлі. Взаємодія між агентами при такій реалізації здійснюється через зв'язок між платформами.

Переваги реалізації такого підходу полягають в тому, що на кожному вузлі системи знаходиться своя окрема платформа. В разі знищення або виходу з ладу одного з вузлів, недоступним для інших агентів стане лише цей вузол, але система в цілому буде функціонувати. На Рис.1 зображено мультиагентну систему побудовану з використанням багатоплатформового підходу.

Основним недоліком такого підходу є складність реалізації та складність впровадження сервісу жовтих сторінок. Оскільки для реалізації такого підходу доведеться об'єднувати декілька DF у федерацію, для того, щоб агенти з інших вузлів мали доступ до сервісу. Також, використання даного підходу збільшить використання ресурсів на вузлах системи.

Структура системи з використанням єдиної платформи зображена на Рис. 2. При такому підході побудови системи кожен з агентів, які існують в системі, реєструються на єдиній платформі, що запущена на одному з вузлів системи.

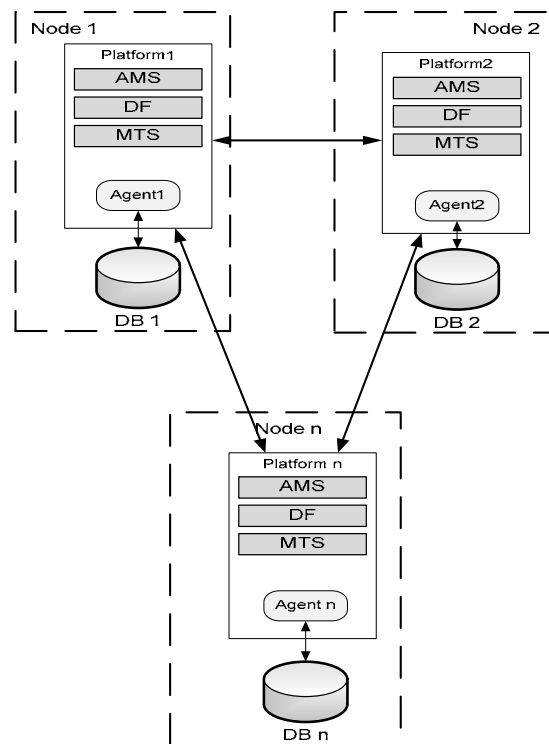


Рис. 1 – Структура системи з використанням багатоплатформового підходу до побудови

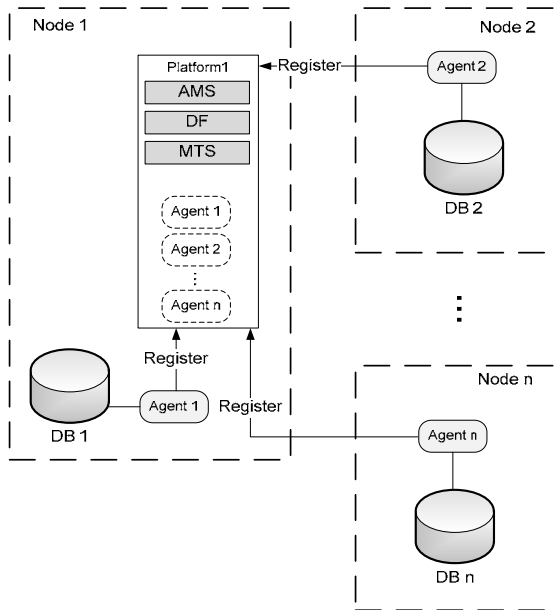


Рис. 2 – Структура системи з використанням єдиної платформи

Використання даного підходу до реалізації зменшує використання ресурсів на елементах системи, оскільки лише на одному вузлі необхідно буде запускати платформу. На інших вузлах будуть запуснені лише основні агенти, які після реєстрації будуть готові до роботи. До того ж зникає проблема реалізації

розподіленого сервісу жовтих сторінок, оскільки всі агенти системи реєструються на єдиному DF, до якого кожен з них має доступ.

Основним недоліком такої системи є те, що в разі знищення або виходу з ладу вузла, на якому запуснена платформа вся система дасть збій і перестане функціонувати.

Для реалізації поставленої задачі було обрано спосіб побудови з використанням єдиної платформи, тому що в порівнянні з першим способом він є менш ресурсоємним та більш простим в реалізації. Надійність системи була підвищена шляхом запуску резервної платформи, на яку будуть автоматично переключені всі агенти системи у разі недоступності основної платформи.

Для побудови мультиагентної системи обміну інформацією було запропоновано використовувати два типи агентів MainClientAgent та ListenerAgent. Основним завданням MainClientAgent є взаємодія з користувачем, передача запитів до групи агентів ListenerAgent, отримання та узагальнення інформації від цих агентів. Основним завданням кожного з агентів ListenerAgent є взаємодія з базою даних на вузлах системи і передача даних агенту MainClientAgent у відповідь на його запит. На кожному з вузлів системи, де знаходяться бази даних необхідно встановити один

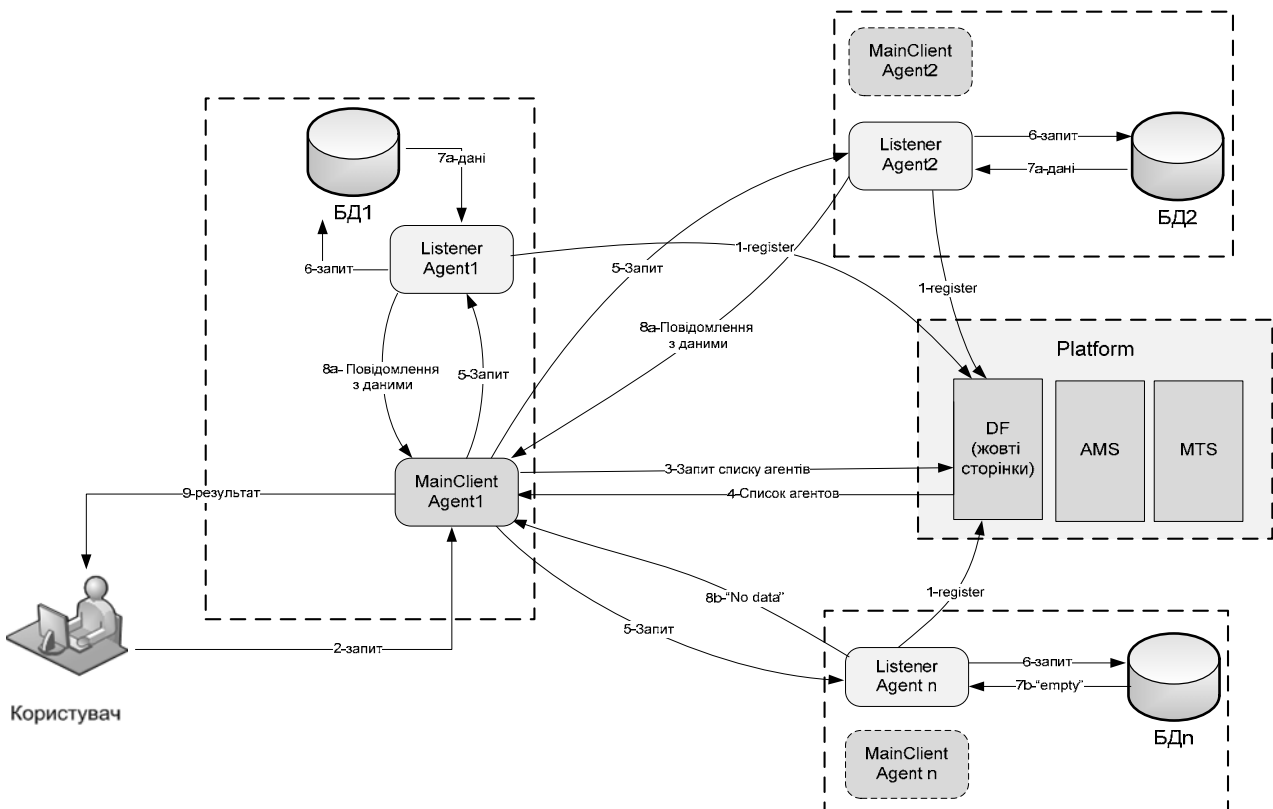


Рис. 3 – Схема взаємодії агентів в системі

ListenerAgent. MainClientAgent встановлюється на тих вузлах, з яких необхідно надати доступ до розподілених даних.

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.

Схема взаємодії агентів показана на Рис. 3 та описана нижче.

Взаємодія агентів відбувається наступним чином:

1. Під час запуску всі агенти ListenerAgent реєструються на сервісі жовтих сторінок.

2. За допомогою графічного інтерфейсу користувач вводить дані для пошуку в базі даних, на основі яких MainClientAgent формує запит.

3. MainClientAgent надсилає запит DF для отримання списку активних агентів (в даному випадку ListenerAgent), які відповідають заданому критерію пошуку.

4. DF формує відповідь і надсилає MainClientAgent список агентів, які зареєстровані в DF і відповідають заданому критерію пошуку.

5. MainClientAgent формує повідомлення з запитом та розсилає його всім агентам вказаним у списку отриманому від DF.

6. ListenerAgent отримує повідомлення з запитом та передає запит до СКБД.

7. СКБД оброблює запит та передає результат агенту ListenerAgent: дані (7a) або пустий набір (7b).

8. ListenerAgent формує відповідь до MainClientAgent і надсилає йому повідомлення з вкладеними даними (8a) або з текстом “No data” (8b), якщо в базі даних були відсутні дані.

9. MainClientAgent проводить обробку повідомлень отриманих від кожного з агентів, формує результуючий набір даних як єдину таблицю та виводить її на екран.

3. РЕАЛІЗАЦІЯ ТА ЗАСТОСУВАННЯ

Для реалізації мультиагентної системи було обрано платформу Java Agent Development Framework (JADE), яка створена згідно специфікацій FIPA [10,11] та побудована на мові Java [12]. Платформа JADE (як середовище функціонування агентів) складається з системи контейнерів, розподілених у мережі. Кожна платформа обов'язково має головний контейнер, який представляє собою точку початкового завантаження платформи. 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.

Запити на отримання даних формуються агентом MainClientAgent у вигляді запиту стандартизованою мовою SQL і передаються до агентів ListenerAgent. Агенти ListenerAgent використовують цей запит для отримання даних з БД. Оскільки SQL є основною мовою запитів до баз даних в переважній більшості сучасних реляційних СКБД, то це полегшує завдання розробки функції взаємодії агентів мультиагентної системи з різними типами СКБД. Тобто незалежно від типу СКБД, кожна з них розуміє один і той же SQL запит однаково.

Взаємодія між агентами відбувається за допомогою мови Agent Communication Language (ACL), стандартизованої FIPA [10]. Нижче показана структура повідомлення описана в стандарті FIPA, в якій повідомлення вкладаються одне в інше, знизу вгору, утворюючи свого роду “надмережний стек” протоколів обміну знаннями [13]:

- Envelope – конверт, який містить поля:
 - відправник;
 - одержувач;
 - тип повідомлення.
- Message – повідомлення, описане мовою ACL, яке містить в собі поля:
 - мова вираження даних;
 - тип кодування даних;
 - онтологія – описує набір символів повідомлення.
- Content – вміст повідомлення, описується мовою CL та містить в собі поля:
 - дані;
 - знання (факти, правила, запити).
- Symbol – символи, що містяться в описі CL, можуть відповідати певній онтології.

В запропонованій мультиагентній системі дані в повідомленнях передаються описаним нижче способом. Запит від MainClientAgent до ListenerAgent та повідомлення від ListenerAgent про відсутність даних і помилки, отримані від СКБД, передаються як текстові дані на рівні Content описаної вище ієрархії. Повідомлення, які містять дані, отримані з БД, передаються від ListenerAgent до MainClientAgent у вигляді об'єкта Java Collection на рівні Envelope.

Дані, отримані від кожного агента ListenerAgent консолідується і виводяться у вікно графічного інтерфейсу у вигляді таблиці (рис. 4). Сортування і фільтрація отриманих даних відбувається з використанням стандартних

методів графічної компоненти `javax.swing.JTable`.

Параметри підключення агента `ListenerAgent` до конкретної СКБД задаються у файлі конфігурації, де і вказується тип самої СКБД.

Реалізація запропонованого підходу розглядалась на прикладі створення системи доступу до розподілених баз даних управління у справах захисту населення від наслідків аварії на Чорнобильській АЕС м. Києва. Існуюче інформаційне середовище управління в цілому представляє собою сукупність незв'язаних застарілих інформаційних систем підрозділів, які територіально розподілені по районах міста та в

якості СКБД використовують настільну СКБД MS Access. Інформаційна система центрального підрозділу консолідує інформацію в СКБД Oracle, для чого вручну проводяться наступні операції: інформація вивантажується з баз даних районних підрозділів, передається на зовнішніх носіях до центрального підрозділу, а потім завантажується до СКБД центрального підрозділу. Це обумовлює суттєві затримки в оновленні інформації та неактуальність інформації, що зберігається в базі даних центрального підрозділу.

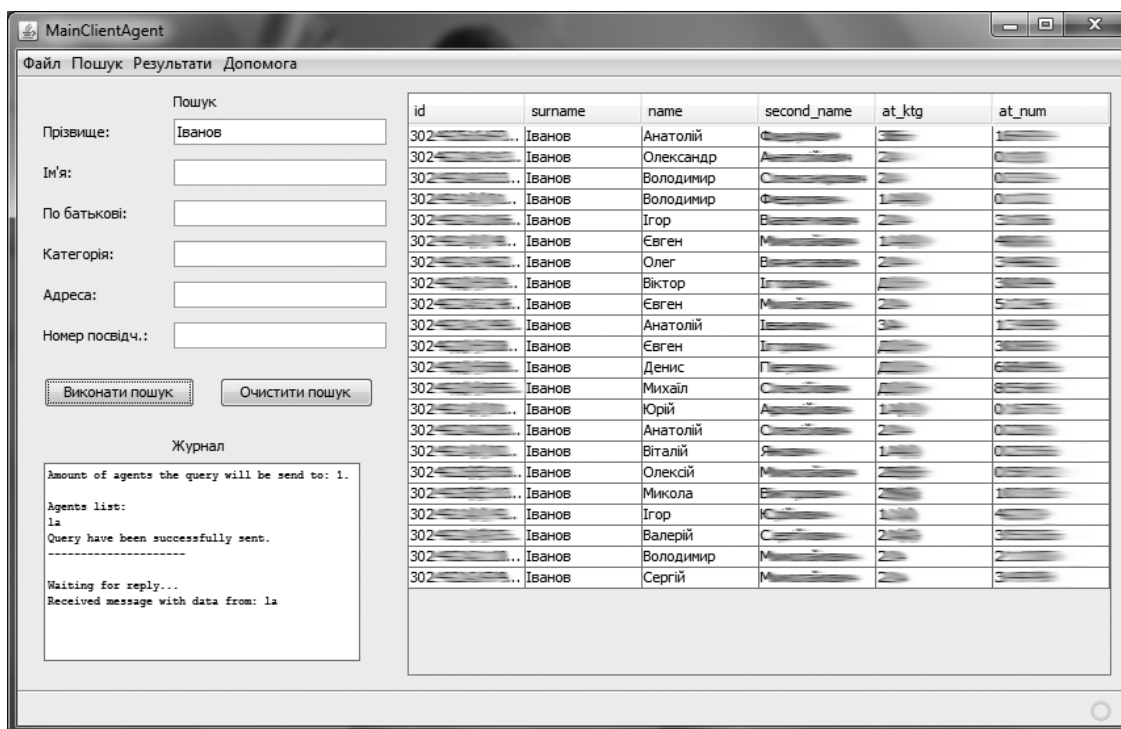


Рис. 4. Графічний інтерфейс користувача.

Для організації доступу до інформації з баз даних районних підрозділів запропоновано використовувати описану вище систему. Оскільки вона не буде потребувати зміни існуючих інформаційних систем та дозволить суттєво підвищити оперативність отримання та актуальність інформації для роботи посадовців центрального підрозділу.

4. ВИСНОВКИ

Запропонований в роботі підхід до створення розподілених систем обміну даними з використанням агентних технологій дозволяє вирішувати задачі доступу та роботи з інформацією з гетерогенних баз даних та зняти проблему взаємодії з даними застарілих інформаційних систем без необхідності їх переробки. Реалізована на базі цього підходу

система побудована з використанням технологій Java, що дозволяє використовувати таку систему на будь-якій платформі, на якій є можливість запустити Java Virtual Machine. Для з'єднання агентів з СКБД використовуються драйвери JDBC, що дозволяє підключити агента до будь-якої СКБД, для якої реалізовані драйвери JDBC та яка підтримує мову запитів SQL.

5. СПИСОК ЛІТЕРАТУРИ

- [1] Разработка фундаментальных основ создания распределенных информационно-вычислительных ресурсов. [Електронний ресурс]. – Електрон. текстові дані (205 131 bytes). – Режим доступу: <http://www.ict.nsc.ru/sitepage.php?PageID=14> Thursday, 31.03.2011 17:49.

- [2] Mbarka B. Control access policies for distributed resources. [Електронний ресурс]. – Електрон. текстові дані (130 853 bytes). – Режим доступу: http://www.fisoft1.com/sources/policies_report.pdf Thursday, 31.03.2011 17:53.
- [3] Nunes C., Kulesza U., Sant'Anna C., Nunes I., Garcia A., Lucena C., Comparing Stability of Implementation Techniques for Multi-agent System Product Lines, European Conference on Software Maintenance and Reengineering, 2009. – с.229-232.
- [4] Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика. 2-е изд. М.: Вильямс, 2000. –1120 с.
- [5] Яковлев Ю.С. О концепции построения и выбора распределенных баз данных информационно-поисковых систем. [Електронний ресурс]. – Електрон. текстові дані (230 675 bytes). – Режим доступу: <http://dspace.nbu.gov.ua:8080/dspace/bitstream/handle/123456789/727/6-Yakovlev.pdf?sequence=1> Thursday, 31.03.2011 17:55.
- [6] Силин А. В. Исследование и разработка моделей распределенных баз данных информационных систем корпоративного типа : Дис. ... канд. техн. наук : 05.13.11 Москва, 2002 134 с. РГБ ОД, 61:03-5/919-5
- [7] Neruda R., Cooperation of Computational Intelligence Agents, International Symposium on Collaborative Technologies and Systems (CTS'06), 2006. – с. 256-263
- [8] Распределенные интеллектуальные системы на основе агентов. [Електронний ресурс]. – Електрон. текстові дані (659 077 bytes). – Режим доступу: <http://www.intuit.ru/department/algorithms/distrsa/10/> Thursday, 31.03.2011 18:01.
- [9] FIPA Agent Management Specification. [Електронний ресурс]. – Електрон. текстові дані (1 457 286 bytes). – Режим доступу: <http://fipa.org/specs/fipa00023/SC00023K.html> Thursday, 31.03.2011 18:11.
- [10] FIPA Specifications. [Електронний ресурс]. – Електрон. текстові дані (85 946 bytes). – Режим доступу: <http://fipa.org/specifications/index.html> Thursday, 31.03.2011 18:16.
- [11] Bellifemine F., Poggi A., Rimassa G. JADE – a FIPA-compliant agent framework. [Електронний ресурс]. – Електрон. текстові дані (154 953 bytes). – Режим доступу: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=8FDC78AFCC8CAC3AC365050D78617C59?doi=10.1.1.86.6371&rep=rep1&type=pdf> Thursday, 31.03.2011 18:06.
- [12] Introduction. [Електронний ресурс]. – Електрон. текстові дані (5 950 bytes). – Режим доступу: <http://jade.tilab.com/doc/html/intro.htm> Thursday, 31.03.2011 18:14.
- [13] Гореликов М.В., Иванов А.М. Архитектура и реализация платформы JADE. [Електронний ресурс]. – Електрон. текстові дані (1 470 431 bytes). – Режим доступу: <http://agentlab.ru/confluence/pages/viewpage.action?pageId=17367045> Monday, 23.05.2011 18:32.



Письменный Вячеслав Юрійович, студент Інституту телекомунікаційних систем Національного технічного університету України "Київський політехнічний інститут". В 2009 році отримав кваліфікацію бакалавра з телекомунікацій. На даний час навчається в магістратурі Інституту телекомунікаційних систем Національного технічного університету України "Київський політехнічний інститут". Наукові інтереси: розподілені інформаційні системи та бази даних, агентні системи.



Терновий Максим Юрійович, к.т.н., с.н.с. В 2002 році закінчив Національний технічний університет України "Київський політехнічний інститут" та отримав кваліфікацію магістра прикладної математики за спеціальністю інформатика. В 2007 році захистив кандидатську дисертацію за спеціальністю "Автоматизовані системи управління і прогресивні інформаційні технології". В 2010 році отримав звання старшого наукового співробітника. Зараз навчається в докторантурі Національного технічного університету України "Київський політехнічний інститут". Має близько 80 наукових робіт. Наукові інтереси: розподілені інформаційні системи, інтелектуальні системи підтримки прийняття рішень, mobile grid, pervasive computing.