

ПРОТОТИПУВАННЯ АВТОМАТИЗОВАНОЇ МУЛЬТИАГЕНТНОЇ ПОШУКОВОЇ СИСТЕМИ УНІФІКОВАНОЮ МОВОЮ МОДЕЛЮВАННЯ АГЕНТІВ

Запропоновано модель мультиагентної пошукової системи по мережі Інтернет засобами уніфікованої мови моделювання агентів (AURL). Змодельована пошукова система класифікує веб-сторінки за певними типами та виконує спеціалізований пошук залежно від типу пошукового запиту.

Ключові слова: агент, пошук, AURL, типізація, мультиагентні системи

Вступ

Через зростання залежності людини від інформації, що надходить з Інтернет (її актуальності, своєчасності, повноти і т. д.) пошук і розпізнавання інформації в мережі перетворюється із разової операції у тривалий або навіть постійний процес. Для автоматизації такого пошуку необхідно використовувати спеціалізовані прикладні програми, які відстежують зміни інформації, що перебуває у сфері інтересів конкретного користувача. Такі програми дістали назву *агенти Інтернет*.

Агентами Інтернет називають програми, що автоматизують пошук, розпізнавання, отримання та аналіз інформації у мережі, орієнтовані на потреби конкретного користувача (або групи користувачів).

Програмування агентів Інтернет є одним із найцікавіших напрямків у сфері пошуку й розпізнавання інформації в мережі, що активно розвивається останнє десятиліття. Перспективність

цього підходу продемонстрована в ряді дослідницьких і комерційних проектів [3, с. 115–118; 6, с. 2–3; 8, с.4–8; 4, с. 9–10].

У цій статті автор моделює основні модулі автоматизованої мультиагентної пошукової системи (АМАПС), яка здійснює спеціалізований пошук на основі визначених типів сайтів у Інтернет [опис системи в статті: 2, с. 234–239]. Новизна роботи полягає в створенні пошукової системи, що здійснює розбиття множини всіх сайтів на спеціалізовані підмножини, які відповідають певному типу (наприклад, магазини, розважальні сайти тощо), та виконує типізований пошук залежно від типу пошукового запиту. Автор створює прототипи основних агентів АМАПС за допомогою уніфікованої мови моделювання агентів (AURL [4, с. 13]), що є стандартом організації зі стандартизації агентних технологій FIPA [4, с. 10–27] для моделювання мультиагентних систем.

Прототипування мультиагентної пошукової системи

АМАПС складається із агента-брокера (управляє розподілом пошукових завдань та аналізує запит), агента типізатора (типізує та індексує сторінки Інтернет) та спеціалізованих для кожного типу сайтів пошукових агентів [детальніша специфікація описана у: 6, с. 237–238]. АМАПС взаємодіє із інтерфейсом користувача (GUI) та базою знань (*KnowledgeBase*).

Для моделювання системи використано середовище розробки INGENIAS [7, с. 394–403].

Агент-брокер в АМАПС отримує пошуковий запит, аналізує його та делегує пошук за ним агенту-шукачу визначеного брокером типу, який відповідає запиту. Якщо брокер не може визначити тип пошукового агента, запит делегується агенту за замовчуванням.

Агент-брокер у загальному вигляді представлений роллю «Брокер» (*Broker*), оскільки в системі не обов'язково має бути єдиний агент-брокер. Для діаграм розгортання використовуються реалізація ролі «Брокер», яка називається «Агент-брокер» (*AgentBroker*). Завдання та цілі агента-брокера наведено в табл. 1.

Початковий ментальний стан агента-брокера не містить жодних фактів дії. Додатки, з якими співпрацює агент-брокер: база знань, інтерфейс користувача. Роль брокера та реалізація агента-брокера представлено діаграмами на рис. 1–4.

Агент-типізатор здійснює індексацію сторінок та визначення їх типів. На цьому етапі реалізації системи для оптимізації роботи агента-типізатора його було зв'язано із пошуковою системою *Google*. Завдання та цілі агента-типізатора зібрано в табл. 2.

Початковий ментальний стан агента-типізатора містить факт дії «розклад», що описує розклад оновлення індексів та визначених типів сторінок в базі знань. Додатки, з якими співпрацює агент-типізатор (*AgentTyper*): база знань, набір інструментів для роботи з пошуковою системою *Google* [5].

Реалізація агента-типізатора представлена діаграмами на рис. 5, 6.

Для всіх агентів-шукачів та агента за замовчуванням характерні однакові завдання та цілі (див. табл. 3) – це допомога брокерові в пошуку та загальний виграш системи з погляду якості пошуку.

Додатки, з якими співпрацює роль агента-шукача: база знань. Реалізація агента магазинів представлена на діаграмах (див. рис. 7 та 8).

Додаток, що відповідає за діалоговий інтерфейс для взаємодії агента з користувачем є внутрішнім, називається «GUI» і має два методи: *showGUI()* – відкрити діалоговий інтерфейс для роботи з користувачем та *actionPerformed()* – запускає дію «GUIInputQueryEvent», яку прослуховує агент-брокер.

Таблиця 1. Завдання та цілі агента-брокера

Завдання	Мета	Входи	Виходи
Отримати завдання від інтерфейсу користувача (<i>ReceiveQuery FromGUI</i>)	Ефективний пошук у мережі Інтернет (<i>Search</i>)	Подія «надходження запиту від інтерфейсу користувача» (<i>GUIInputQueryEvent</i>)	Запит (<i>Query</i>)
Аналізувати запит (<i>AnalyzeQuery</i>)	Ефективний пошук у мережі Інтернет (<i>Search</i>)	Запит (<i>Query</i>), база знань (<i>Knowledge BaseWrapper</i>)	Проаналізований запит (<i>AnalyzedData</i>)
Знайти агента, який найкраще виконає пошук із наявних (<i>FindAgentToSearch</i>)	Ефективний пошук у мережі Інтернет (<i>Search</i>)	Проаналізований запит (<i>AnalyzedData</i>)	Діалог (<i>SearchingForAgent Interaction</i>), Контракт на пошук (<i>QuestionFF</i>)
Делегувати пошук (<i>DelegateSearch</i>)	Ефективний пошук у мережі Інтернет (<i>Search</i>)	Протокол діалогу (<i>SearchingForAgent Interaction</i>), Відповіді на контракт (<i>AnswerFF</i>)	Ім'я агента для пошуку (<i>BrokerDelegation Decision</i>)
Повернути результати інтерфейсу користувача (<i>ReturnResultsToGUI</i>)	Ефективний пошук у мережі Інтернет (<i>Search</i>)	Результати, що потрібно повернути (<i>ResultsToReturn</i>), інтерфейс користувача (<i>GUI</i>)	–

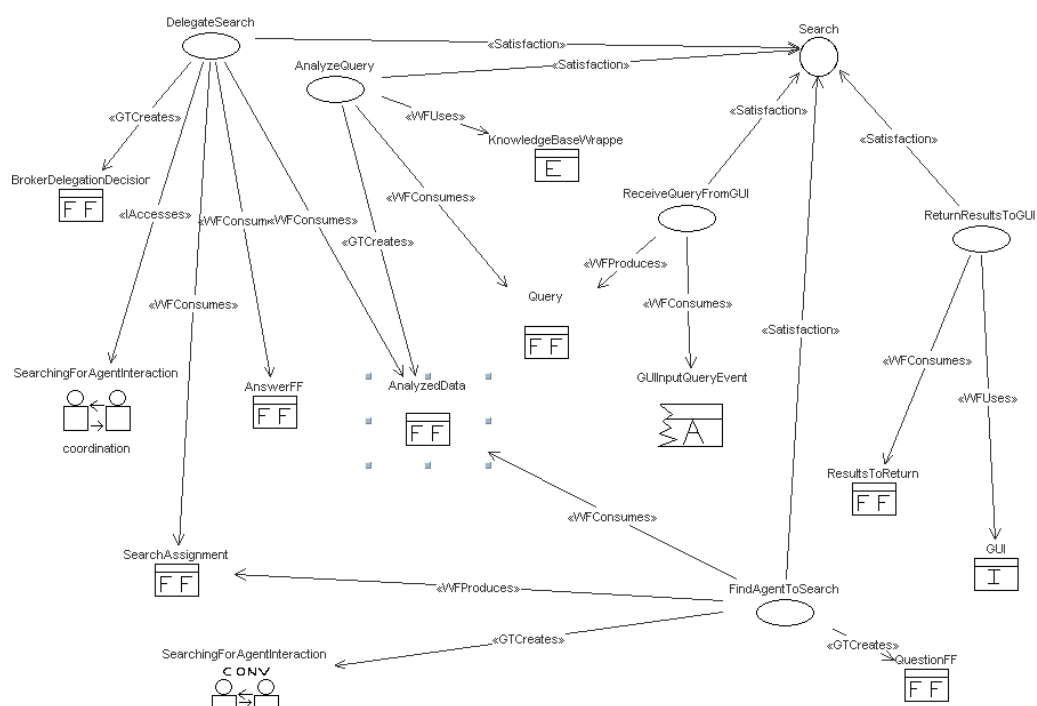


Рис. 1. Визначення завдань та цілей ролі «брокер»

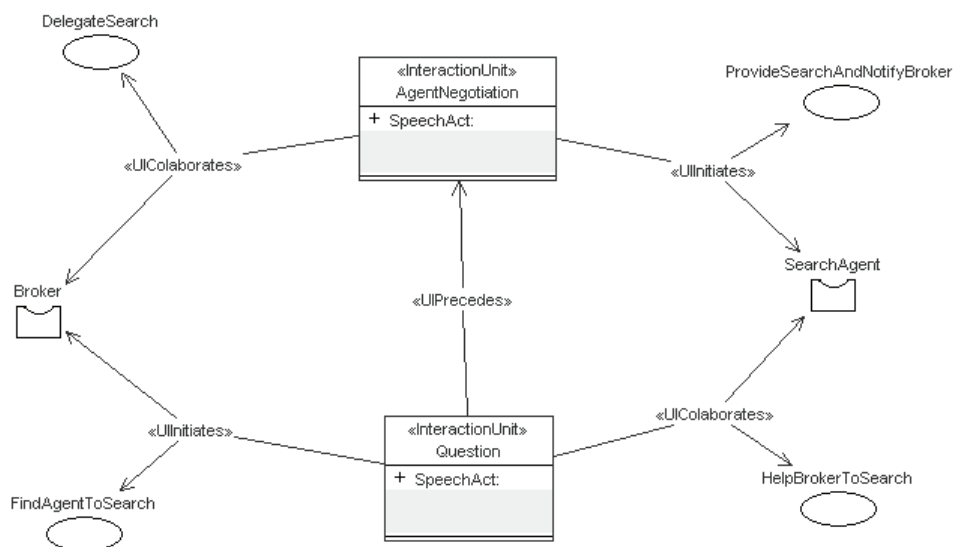


Рис. 2. Реалізація протоколу взаємодії агента-брокера із агентами-шукачами

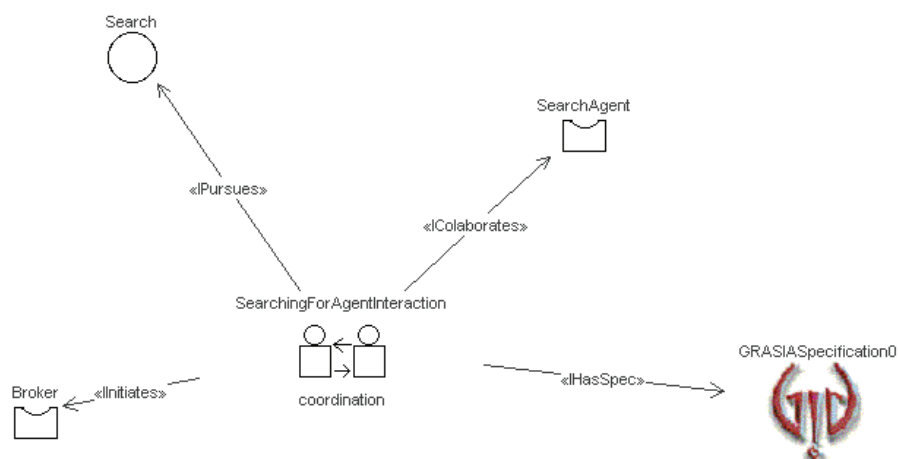


Рис. 3. Взаємодія агента-брокера із агентами-шукачами

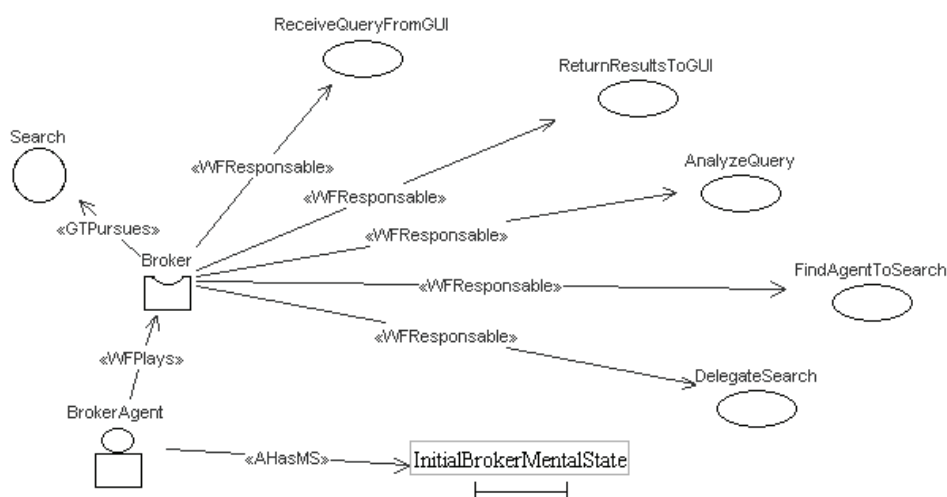


Рис. 4. Визначення агента-брокера

Таблиця 2. Завдання та цілі агента-типізатора

Завдання	Ціль завдання	Входи	Виходи
Отримати посилання на індексовані Google сторінки (GetIndexesFrom Google)	Типізувати веб-сторінки та тримати індекси «свіжими» (KeepTypedIndexes UpToDate)	Розклад оновлення (TyperSchedule), набір інструментів для роботи із пошуковою системою Google (Google Search API)	Колекція сайтів (MSFrameFact)
Типізувати отримані сторінки (TypizateGoogle Indexes)	Типізувати веб-сторінки та тримати індекси «свіжими» (KeepTypedIndexes UpToDate)	Колекція сайтів (MSFrameFact), база знань (KnowledgeBase Wrappper)	Типізована колекція сайтів (TypizatedSite Collection)
Оновити базу знань (UpdateKB)	Типізувати веб-сторінки та тримати індекси «свіжими» (KeepTypedIndexes UpToDate)	Типізована колекція сайтів (TypizatedSite Collection), база знань (KnowledgeBase Wrappper)	—

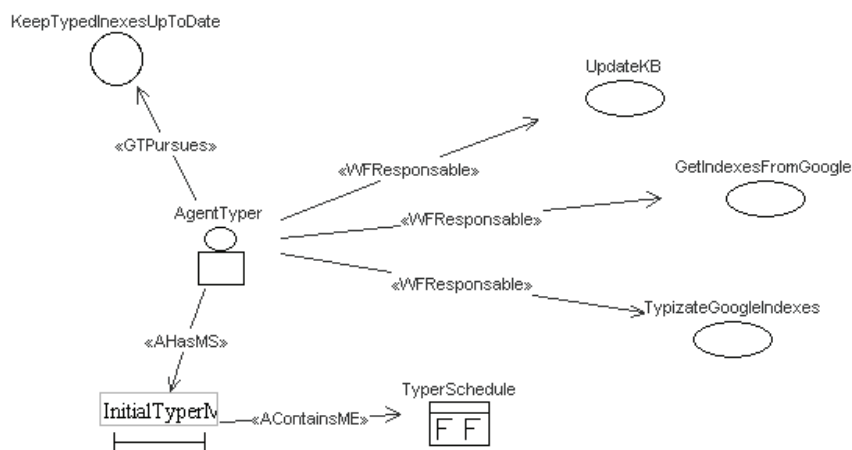


Рис. 5. Визначення агента-типізатора

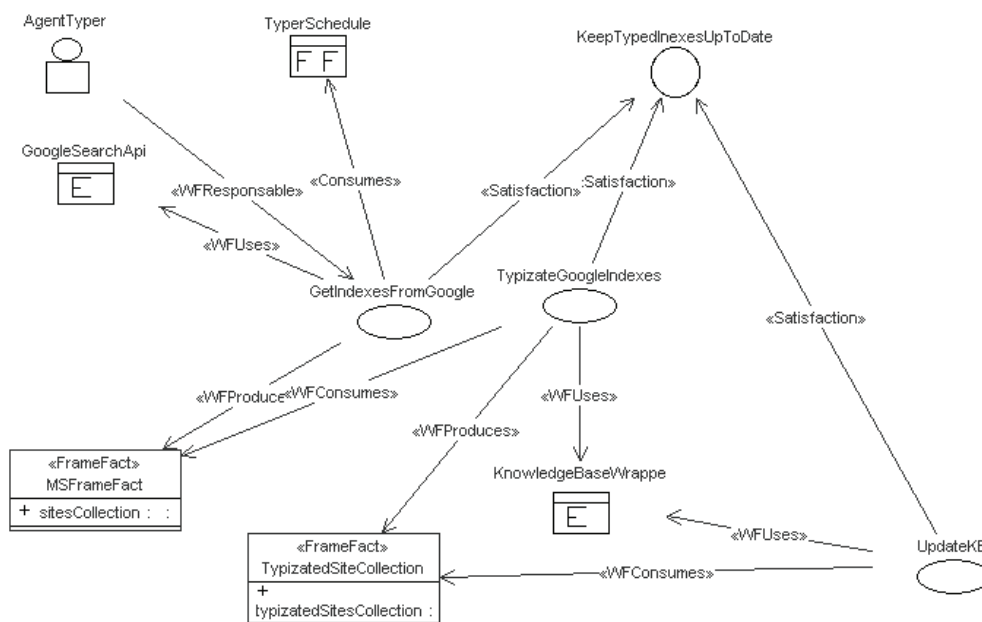


Рис. 6. Завдання та цілі агента-типізатора

Таблиця 3. Завдання та цілі агента ролі агента-шукача

Завдання	Мета	Входи	Виходи
Провести пошук та сповістити брокера про результати (ProvideSearchAnd NotifyBroker)	Ефективний пошук у мережі Інтернет (Search)	Факт дії «результат пошуку агентом-шукачем» (SearchAgent SearchResults)	Факт дії «результат, що повертається брокеру» (ResultToReturn)
Допомогти брокерові у пошуку (Help BrokerToSearch)	Ефективний пошук у мережі Інтернет (Search)	Факт дії «контракт на пошук» (QuestionFF), діалог брокера з агентами-шукачами (SearchingForAgent Interaction)	Факт дії «відповідь на контракт» (AnswerFF)

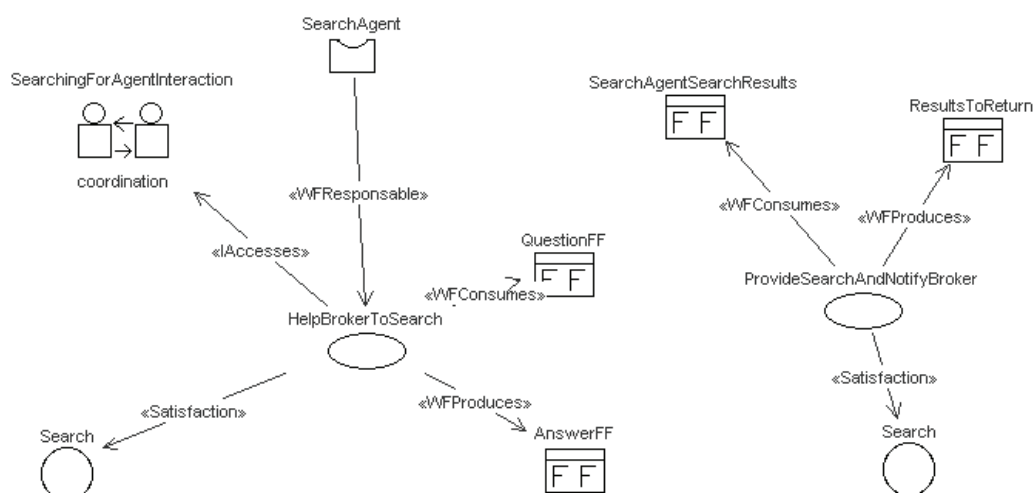


Рис. 7. Завдання ролі агента-шукача

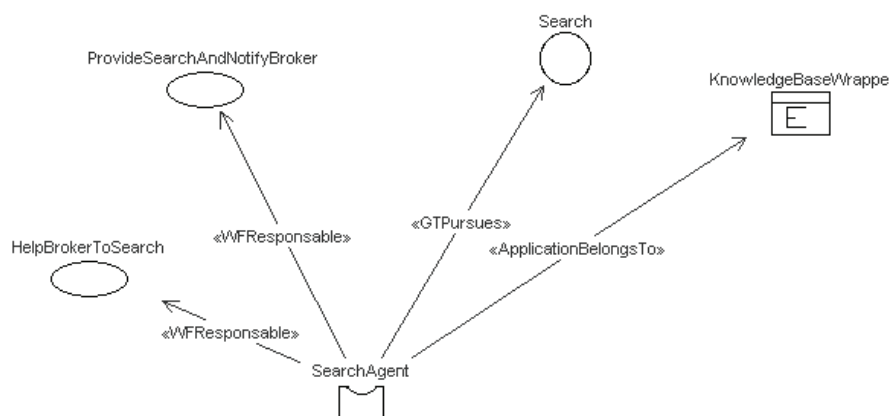


Рис. 8. Визначення ролі агента-шукача

Додаток, що відповідає за роботу з базою знань, є зовнішнім і називається KnowledgeBaseWrapper, він реалізує доступ до бази знань на зрозумілій агентам мові.

Висновки

У цій статті описано прототипи основних агентів автоматизованої мультиагентної пошукової системи уніфікованою мовою моделювання агентів: агента-брокера, агента-типизатора та ролі агента-шукача, яку виконують всі агенти системи, що допомагають агенту-брокеру покращити якість пошуку з точки зору агента-брокера.

З отриманої реалізації мультиагентної системи у середовищі розробки INGENIAS може бути згенеровано працюючий Java-код [1, с. 89–94].

Перевагами цієї архітектури перед класичними пошуковими системами є можливість вносити зміни будь-якої складності без перебоїв в роботі. Такими змінами, наприклад, можуть бути

додавання нового типу сайтів у систему та реалізації агента, що буде виконувати спеціалізований пошук по сайтах нового типу, введення нових версій спеціалізованих агентів тощо. Ще однією суттєвою перевагою є можливість видачі результатів у найбільш відповідній формі для конкретного типу запиту, наприклад, для пошуку товару – це видача товарів, ранжованих за ціною. З іншого боку, недоліком даної системи може бути швидкість роботи, оскільки пошуковий запит оброблюється і делегується агенту-шукачу, також певний час займає визначення найкращого агента-шукача для пошуку.

Зараз для індексування та типізації веб-сторінок реалізована пошукова система використовує базу даних веб-сайтів Google за допомогою відповідного набору інструментів. Надалі автоматизовану мультиагентну пошукову систему варто робити самостійною, тобто незалежною від інструментів Google.

Література

1. Замковий О. В. Модельний підхід до створення пошукової МАС за допомогою середовища розробки INGENIAS / О. В. Замковий // Вісник Київського національного університету імені Тараса Шевченка. Серія : Фіз.-мат. науки. – 2009. – Випуск № 4. – С. 89–94.
2. Замковий О. В. Типізація веб-простору як засіб виключення реклами з пошукового процесу / О. В. Замковий // Сьома міжнародна конференція «Теоретичні та прикладні аспекти побудови програмних систем» (TAAPSD'2010). – К., 2010. – С. 234–239.
3. Adding Logic Programming Behaviour to the WorldWideWeb / Loke S. W. // CiteSeerX. – Режим доступу : <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.9131&rank=1>. – Назва з екрана.
4. Bellifemine F. Developing Multi Agent Systems with JADE / F. Bellifemine, G. Caire, D. Greenwood. – West Sussex : John Wiley & Sons, 2007. – 286 p.
5. Google AJAX Search API. Руководство. [Електронний ресурс] / От новичка до профессионала. – Режим доступу: <http://i-novice.net/google-ajax-search-api-rukovodstvo/>. – Назва з екрана.
6. On The Problem Of Using Logic Object-Oriented Programming In The World Wide Web [Електронний ресурс] / A. A. Morozov, Yu. V. Obukhov, Yu. V. Gulyaev // Institute of Radio Engineering and Electronics of RAS. – Режим доступу: <http://www.cplire.ru/Lab144/internet.pdf> — Назва з екрана.
7. Pavón J. Agent Oriented Software Engineering with INGENIAS / J. Pavón, J. Gómez-Sanz // MultiAgent Systems and Applications III. – 2003. – P. 394–403.
8. Wooldridge M. Introduction to Multiagent Systems / M. Wooldridge. – West Sussex : John Wiley & Sons, 2002. – 349 p.

A. Zamkovyi

AN AGENT UNIFIED MODELING LANGUAGE PROTOTYPE OF SEMI-AUTOMATED MULTI-AGENT SEARCH SYSTEM

In this article the author creates an agent unified modeling language (AUML) prototype of multi-agent search engine. The given search engine divides all web-pages into a various types and provides Internet search depending on the search query type.

Keywords: agent, search, AUML, website type classification.

Матеріал надійшов 11 травня 2011 р.