УДК 004.415, 004.738.5

Н.О. Думанський, О.В. Марковець
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

МОДЕЛЬ АГЕНТА ПОБУДОВИ ЗАПИТУ ДЛЯ ТЕМАТИЧНОЇ ПОШУКОВОЇ СИСТЕМИ

Розглянуто особливості роботи агента побудови запиту для тематичної пошукової системи, описано математичну модель агента та запропоновано критерії створення агента побудови запиту.

Ключові слова: агент, модель, пошук інформації.

The article deals with peculiar features of request building agent for subject-based search engine. Mathematical model of agent is described and criteria for developing request building agent are suggested.

Keywords: agent, model, search information.

Вступ

Оскільки Інтернет використовує різнорідні ресурси й різні методи доступу до цих ресурсів, то системи пошуку документів за ключовими словами, авторами, тематикою постійно доповнювалися можливостями складніших тематичних запитів. Такі системи надавали розподілений доступ до документів, що розташовані на веб-серверах, на місцевих і віддалених серверах, до яких був можливий доступ через мережу.

Із збільшенням кількості інформації в мережі Інтернет потреба в релевантному тематичному пошуку загострилась. Але, крім величезної кількості інформації, ускладнює пошук потрібної інформації ще й складна, різнорідна структура даних, а різноманіття методів доступу до них створює нові проблеми, пов’язані з необхідністю отримання за запитом точнішої тематичної інформації. Одним з підходів до розв’язання задач швидшого пошуку інформації є використання агентної технології. Агенти відрізняються від поширених універсальних систем тим, що:

- спроможні самостійно виконувати завдання користувача (без уточнювальних звертань до користувача чи адміністратора) протягом тривалого часу (дні, тижні);
- один раз створений агент може бути використаний декілька разів у майбутньому, на відміну від запиту універсальної системи, який збирає інформацію лише один раз.

Як правило, агентами називають програми, що автоматизують пошук, розпізнавання, здобування та аналіз інформації, орієнтованої на потреби певних користувачів (груп користувачів) та демонструють автономність, реактивність (реакцію на зовнішні фактори), проактивність (дії з власної ініціативи) і, у випадку багатоагентних систем, цілеспрямовану групову поведінку.

Використання агентного підходу допоможе реалізувати такі функції системи:

- *автономність* – здатність функціонувати без втручання людини і при цьому здійснювати самоконтроль над своїми діями та внутрішнім станом;
- *суспільна поведінка* (social ability) – здатність функціонувати разом з іншими агентами, обмінюючись повідомленнями з "колегами" за допомогою якоїсь загальнозрозумілої мови комунікації;
- *реактивність* (reactivity) – здатність сприймати стан середовища і своєчасно реагувати на зміни, що там відбуваються;
- *проактивність*, або позитивна активність (pro-activity), – здатність агента брати на себе ініціативу, тобто здатність генерувати мету і діяти раціонально для її досягнення, а не лише реагувати на зовнішні події.

Остання властивість і є саме тим, що відрізняє агента від простої програми.

Також агент має ряд понять, які використовуються для реалізації його властивостей та функціональних можливостей:

- *знання* (knowledge) – це стала частина знань агента про себе, про середовище та про інших агентів; ці знання не змінюються в процесі його функціонування;
- *переконання* (beliefs, віра) – знання агента про середовище, зокрема про інших агентів; ці знання можуть змінюватися в часі, навіть ставати помилковими, однак агент може нічого про це не знати і продовжувати вірити, що на цих знаннях можуть ґрунтуватися висновки;
- *бажання* (desires) – це стани, досягнення яких з різних причин бажане для агента;
- *наміри* (intentions) – це те, що агент має зробити, оскільки або зобов'язався, або це випливає з його бажань;
- *цілі* (goals) – конкретна множина станів, досягнення яких агент вибрав як поточну стратегію поведінки;
- *обов'язки* (commitments) відносно інших агентів – завдання, які агент бере на себе на прохання (доручення) інших агентів у межах кооперативних цілей.

Існує думка, що до властивостей агента також можна додати:

- *мобільність* (mobility) – здатність агента мігрувати, рухатись мережею в пошуках необхідної інформації для розв'язання своїх завдань, при кооперативному розв'язанні їх разом або за допомогою інших агентів;
- *доброзичливість* (benevolence) – готовність агентів допомагати один одному та готовність агента розв'язувати саме ті завдання, що йому доручив користувач;
- *правдивість* (veracity) – властивість агента не маніпулювати недостовірною інформацією, знаючи про те, що вона неправдива;
- *раціональність* (rationality) – властивість діяти так, щоб досягти своїх цілей, а не уникати цього, принаймні, у межах своїх знань і переконань.

Агент розглядається як об'єкт, що діє автономно, цілеспрямовано і виконує задані функції. Він функціонує, ґрунтуючись на індивідуальних і загальнодоступних знаннях, переконаннях і намірах. Агент формування запиту – це об'єкт, якщо можна запрограмувати і який може обмінюватися повідомленнями в межах структурованих планів з іншими агентами. Ці плани задаються в класах повідомлень і виконуються екземплярами повідомлень. Класи повідомлень визначають разом з агентом. Агента визначають, призначаючи йому ім’я, встановлюючи змінні, які утворюють його локальну базу даних, і встановлюючи початкові повідомлення (initial conversation). Після визначення агента виконується його початкова ініціалізація повідомлень і впродовж цього часу він перебуває в активному стані. Якщо початкові повідомлення не були визначені, то агент залишається в стані очікування доти, доки не одержить повідомлення від іншого агента.

Розглянемо основні принципи роботи агента з використанням мови Java. У такому разі клас агент являє собою загальний базовий клас для агентів, обумовлених користувачем. З погляду програміста агент JADE (Java Agent DEvelopment Framework) – звичайний екземпляр Java-класу, що розширює базовий клас Agent. Мається на увазі успадкування властивостей для здійснення основних взаємодій з агентною платформою (реєстрація, конфігурація, віддалене керування тощо) і основного набору методів, що можуть викликатися для реалізації заданого поводження агента.

Обчислювальна модель агента є багатозадачною і паралельною, у якій задачі (чи поведінка) виконуються одночасно. Кожна функціональна можливість і/або сервіс, наданий агентом, повинні бути реалізовані як одна чи кілька поведінок. Внутрішній планувальник, схований від програміста, автоматично керує плануванням поведінки.

Агент JADE може перебувати в одному з кількох станів, представлених у класі Agent такими об’єктами:

– AP_INITIATED: об'єкт класу Agent створений, але поки не зареєстрований у AMS, тому він не має ні імені, ні адреси і не може вести обмін повідомленнями з іншими агентами;

– AP_ACTIVE: об'єкт Agent зареєстрований у AMS, має постійне ім'я й адресу і може використовувати всі сервіси JADE;

– AP_SUSPENDED: об'єкт Agent у цей момент часу зупинений і не виконує ніякої дії;

– AP_WAITING: Agent блокований, його внутрішній стан – без дії і він «прокинеться» у разі виконання деякої умови (наприклад, надходження повідомлення);

– AP_DELETED: внутрішній стан – завершений, і агент не зареєстрований у AM;

– AP_TRANSIT: мобільний агент вводить цей стан для того, щоб, поки агент переміщається до нового місця розташування, система продовжувала накопичувати і зберігати повідомлення, що будуть надіслані йому, коли він прибуде на нове місце;

– AP_COPY: внутрішній стан використовується платформою JADE для створення копії агента;

– AP_GONE: внутрішній стан використовується платформою JADE для повідомлення про те, що мобільний агент перемістився до нового місця розташування і встановив стійкий стан.

Агенти можуть бути: інформаційними, що створюють ядро тематичних індексів і запитів; такими, що виконують фільтрацію завантаженої тематичної інформації; агентами архіву документів, агентами з обслуговування проходження запитів за мережею, агентами інтеграції інформації про пошук. У загальному випадку агенти отримують запити від користувачів з інформацією про необхідні інформаційні ресурси і вимоги до критеріїв пошуку.

Огляд наявних рішень

Відомі методи інтелектуального пошуку усувають певні недоліки пошуку за ключовими словами:

1) урахування числових параметрів документа вирішується за допомогою надання спеціального шаблону для створення документа. Окремі поля цього шаблону відповідатимуть числовим даним, і ці дані зберігатимуться у внутрішній базі даних у числовому вигляді, а показуватимуться користувачеві у вигляді звичайного текстового документа;

2) врахування важливості ключових слів, близьких до предметної області, забезпечується за допомогою створення в базі даних спеціальних груп синонімів для таких ключових слів і подальшого врахування їх пошуковою системою;

3) кожен користувач може створити персонального пошукового інтелектуального агента, і, отже, орієнтувати пошук не на предметну область, а на свої особисті дані.

Загальна вага пропозиції за статистичними і семантичними характеристиками підраховується як сума ваг всіх знайдених слів у цій бізнес-пропозиції.

У результаті розроблення тематичної пошукової системи було сформульовано принцип роботи агента формування запитів до тематичних веб-сайтів і алгоритм створення запиту з використанням інформації від користувача. Для роботи агента потрібна така інформація:

- перелік ключових слів;
- список тематичних веб-сайтів;
- правила створення запитів до сайтів.

Загальні пошукові системи замість такого агента використовують Spider (павук) та Crawler (“мандрівний” павук). Їх основна відмінність в тому, що вони шукають всі веб-сторінки та інформацію про карту сайта цих сторінок. Потім завантажують їх і передають іншим агентам пошукової системи для аналізу. Для тематичного пошуку вони не можуть використовуватись, оскільки не мають змоги оцінювати, до якої саме тематики належить та чи інша сторінка. Адже наявність тематичних слів не завжди означає наявність самої тематики. Окрім цього, у пошукових системах сайт, перед тим як потрапити в список пошуку, не дуже детально аналізується для відсіювання сайтів, які не відповідають рівню цензури певної пошукової системи. У нашому ж випадку це є одним з основних правил відбору тематичних сайтів.

Також є системи, які дають змогу здійснювати пошук та аналізувати слабкоструктуровані веб-ресурси. Вони діють за принципом стандартних пошукових систем і дають змогу структурувати ресурси веб-сайтів. Ці системи можна використовувати для пошуку нових джерел, що можуть застосовуватись тематичною пошуковою системою. База даних, утворена інтелектуальною системою аналізу слабкоструктурованих веб-ресурсів, може використовуватись як основа для побудови бази правил і як джерело нових сайтів пошуку. Структуризація веб-ресурсів допоможе виокремити відповідні правила звернення до бази даних конкретного сервера.

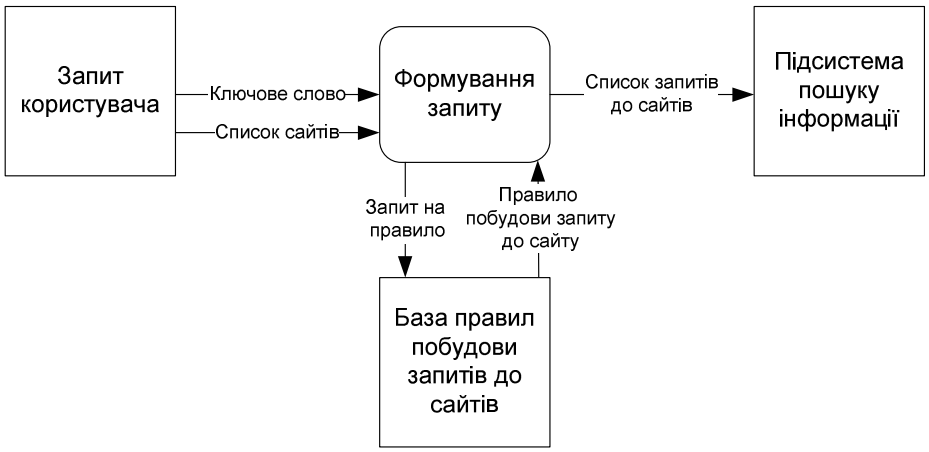

Рис. 1. Контекстна діаграма

Важливу роль в процесі створення моделі агента відіграє контекстна діаграма, що моделює систему найзагальніше. Контекстна діаграма зображає інтерфейс системи із зовнішнім світом, а саме інформаційні потоки між системою та зовнішніми сутностями, з якими вона повинна бути зв'язана. Вона ідентифікує ці зовнішні сутності, а також, як правило, єдиний процес, що зображає головну мету або природу системи, наскільки це можливо. В нашому випадку на контекстній діаграмі зображено процес формування запиту, три зовнішні сутності: запит користувача, підсистема пошуку інформації та база правил побудови запитів до сайтів, а також інформаційні потоки між системою та зовнішніми сутностями.

Від «Запиту користувача» система отримує набір ключових слів та список адрес сайтів, на яких потрібно знайти інформацію за ключовими словами. Використовуючи одержані адреси сайтів, система формує запит до «Бази правил побудови запитів до сайтів» про правила створення запиту до кожного з потрібних сайтів. Отримавши інформацію з бази правил, система підставляє ключові слова у кожний запит і сформований список запитів відправляє «Підсистемі пошуку інформації».

Контекстна діаграма може бути деталізована за допомогою DFD першого рівня (рис. 2).

Ця діаграма містить шість процесів та чотири сховища даних.

Процес 1.1 (отримання вхідної інформації). Здійснює приймання та запис вхідної інформації і має на вході/виході такі потоки:

зовнішній вхідний потік – список сайтів для створення переліку сайтів, з яким працюватиме система;

зовнішній вхідний потік – ключове слово для створення списку слів, за якими здійснюватиметься пошук інформації;

вхідні потоки – адреса сайта та ключове слово для запису адрес сайтів і ключових слів у відповідні сховища даних.

Процес 1.2 (формування запиту до Бази правил). Формує запит на отримання правила для кожного сайта на пошук інформації. Вхідним потоком є адреса сайта, на якому потрібно здійснити пошук, а вихідним потоком є запит на правило створення запиту до цього сайта.

Процес 1.3 (одержання правила створення запиту). Здійснює приймання і запис у сховище правила створення запиту до сайта. Вхідним і вихідним потоком є правило створення запиту.

Процес 1.4 (підстановка ключових слів у правило). Здійснює підстановку ключових слів у правило створення запиту. Вхідними потоками є правила створення запитів та ключові слова, які отримують з відповідних сховищ, а вихідним потоком є правило з ключовими словами.

Процес 1.5 (формування запиту до сайту). Формує запит до сайта, використовуючи правило створення запиту і ключові слова. Вхідним потоком є правила створення запитів з ключовими словами, а вихідним потоком є запит до сайта.

Процес 1.6 (формування списку запитів до сайтів). Формує список всіх запитів до сайтів. Вхідним потоком є запити до сайтів, а вихідним потоком є список зі всіх сформованих запитів до сайта.

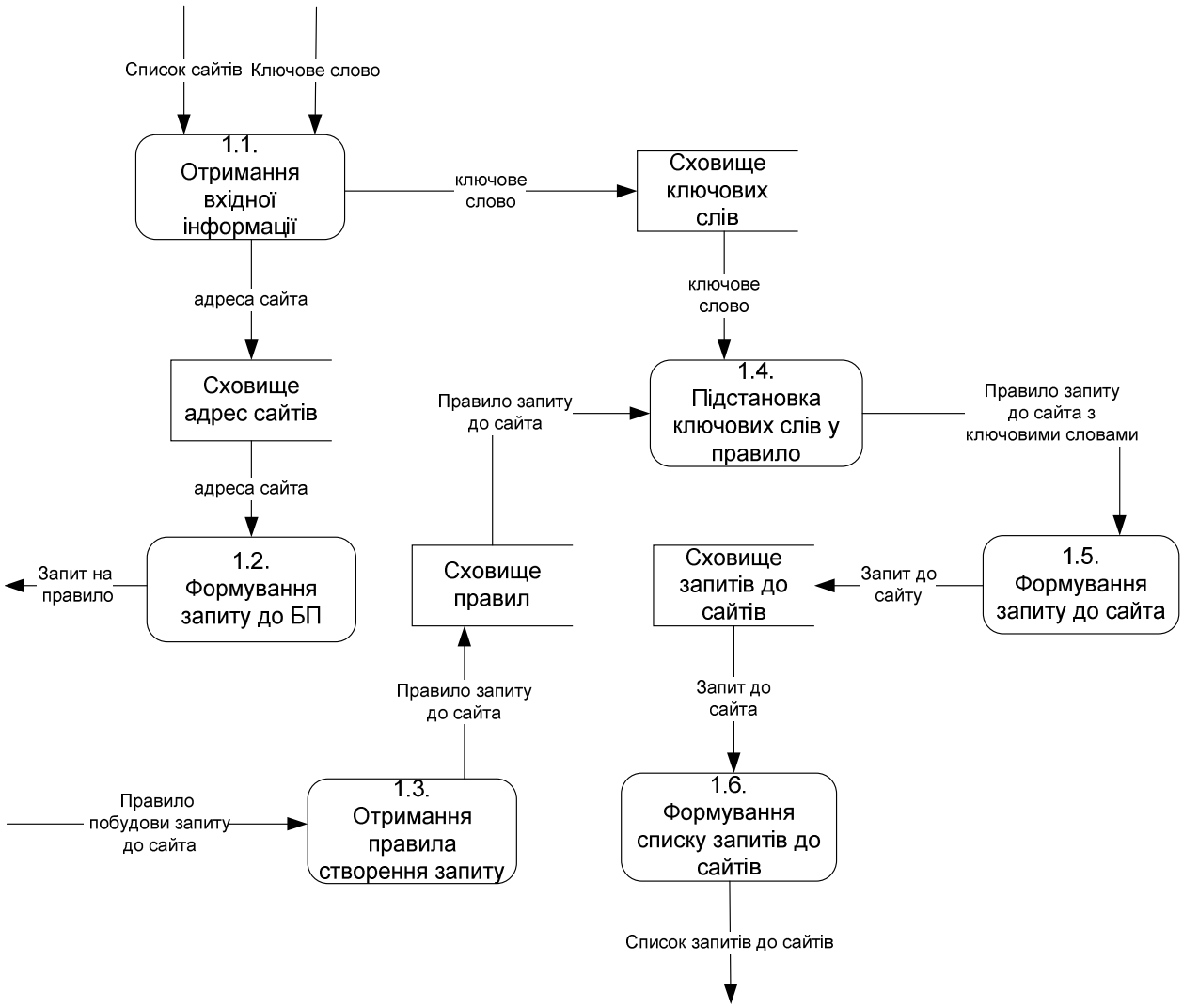

Рис. 2. DFD першого рівня

Запит до сервера формується на основі правил, сформованих адміністратором системи. Правила створюють відповідно до структури та типізації сервера, на якому провадиться пошук. Залежно від поставленої задачі та вибраних критеріїв пошуку певні правила формування запиту можуть не використовуватись або набувати значення за замовчуванням. Відповідно, не всі сервери дають змогу однаково формалізувати параметри пошуку і деякі параметри для їх запитів будуть просто зайвими. Формування списку сайтів та правил звернення до них відбувається за участю людського фактора. Резонним буде питання: наскільки актуальною буде така пошукова система? Враховуючи, що найбільший рейтинг мають здавна відомі сайти, а щойно створені завжди у кінці списку пошуку (якщо вони взагалі там є), наша система нічим не гірша, з урахуванням оновлення бази даних сайтів, за стандартні системи пошуку. Додатково потрібно сказати, що використання людського фактора відбору сайтів дасть змогу унеможливити потрапляння в базу даних ненадійних та недостовірних сервісів.

Агент формування запиту, отримавши інформацію від користувача, звертається до бази правил та вибирає з неї правила формування запитів до сайтів вибраної користувачем тематики. Згідно з одержаними правилами відбувається побудова звернення до сервера тематичного ресурсу з використанням ключових слів та правила побудови запиту. Створений запит долучається до загального списку запитів, який, своєю чергою, передається до підсистеми пошуку інформації.

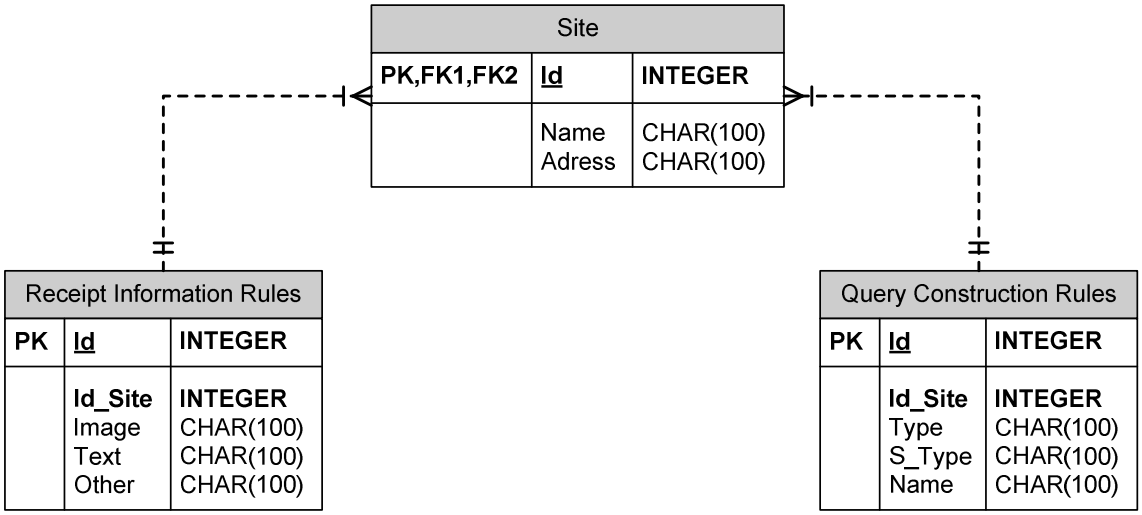

Рис. 3. IR-діаграма

У загальному випадку модель агента формування запиту являє собою кортеж:

$$Mg = (Algorithm, Inut, Output), \quad (1)$$

де Algoriptm – набір команд роботи агента; Input – множина вхідних даних; Output – параметри створеного запиту.

Алгоритм роботи агента описується як послідовність кроків:

1) на основі вхідної інформації про місце пошуку визначає електронну адресу сайта;
2) отримує шаблон запиту до потрібного сайта з бази правил;
3) формує з одержаних ключових слів вхідні значення для запиту;
4) здійснює пошук змінних в шаблоні запиту для заміни їх вхідними значеннями;
5) підставляє вхідні значення в шаблон замість змінних;
6) створює запит до тематичного сайта.

У множині вхідних даних виділимо три складові:

$$Input = (Str(Input), Var(Input), Int(Input)), \quad (2)$$

де Str(Input) – структура Input; Val(Input)– значення параметрів Input; Int(Input) – інтерпретація Input.

Структура Str(Input) визначає форму виразів, що входять в Input. Конкретні значення параметрів фіксуються в Val(Input). Така декомпозиція дає змогу диференціювати формування запиту, варіюючи значення параметрів за незмінної структури моделі вхідних даних і генерацію, що передбачає трансформацію Str(Input).

Інтерпретація Int(Input) характеризує суть (тобто предметний зміст) елементів Str(Input) і значень з Val(Input).

Таблиця 1

Приклад запиту до тематичного сайта

Str(Input)	Ноутбуки продаж		
Val(Input)	Name=Acer	Type=TravelMate	S_Type=4152
Int(Input)	Name – назва продукту	Type – тип продукту	S_Type – номер
A	ebay.com		
V	http://shop.ebay.com/ifems/?_nkw=acer+travelmate+4152&_sacat=0&_trksid=m270&_odkw=acer+travelmate&_osacat=0		

Враховуючи, що пошук буде здійснюватись не на одному, а на кількох сайтах вибраної тематики, можна розширити модель агента формування запиту, замінивши компоненти A та V на відповідні масиви значень A(n) та V(n).

Висновок

Інтернет як джерело інформації характеризується величезним обсягом, динамічністю та дублюванням інформації. Розроблення нових підходів та методів до побудови пошукових систем залишається актуальним на сучасному етапі. В роботі розглянуто принципи побудови агентів пошукових систем, запропоновано новий агент формування запитів до тематичної пошукової системи. Розглянуто спеціалізованого агента системи, який дасть змогу підвищити якісний рівень пошукових систем.

1. Andon Ph., Deretsky V. Control Oriented Ontology and Process Description for Cooperation Agents in Information Retrieval // Sixth International Scientific Conference „Electronic Computers and Informatics ECI'2004". − Kosice – Herlany, Slovakia; September 22–24, 2004. − P. 14–18. 2. Patel, L. Petrosjan, and Rosenstiel W., editors. OASIS: Distributed Search System in the Internet. St. Petersburg State University Published Press, St. Petersburg, 1999. 3. Lieberman Henry. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence // Morgan Kaufmann publishers Inc. − San Mateo, CA, USA, 1995. − P. 924−929. 4. Martin E. Meller. Machine learning based user modeling for www search, http://citeseer.nj.nec.com. Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web http://www.sciam.com/article. cfm?articleid =000A0919. 5. Дерецький В.О. Підхід до автоматичної побудови тематичної онтології документу для удосконалення інформаційного пошуку. − 2005. − № 3. − С. 76−82. 6. OWL Technical Committee. Web Ontology Language (OWL). http://www.w3.org/TR/2004/WD-owlref. 7. Salton G., Buckley C. Term-Weighting Approaches in Automatic Text Retrieval // Information Processing and Management. − 1988. V. 24. − P. 513−523. 8. Козлов Е.Б., Метелкин А.В., Хорошевский В.Ф. Мультиагентная система поиска информации в Интернет. – М.: Физматлит, 2000. – С. 840–850. 9. Куршев Е.П., Осипов Г.С., Рябков О.В., Самбу Е.И., Соловьева Н.В., Трофимов И.В. Интеллектуальная метапоисковая система. – М.: Наука, 2002. – С. 320–330. 10. Кормалев Д.А., Куршев Е.П., Осипов Г.С., Сулейманова Е.А., Трофимов И.В.: Препринт // Методы поиска и анализа информации. Автоматическое извлечение данных. – Переславль-Залесский, ИПС РАН, 2003. – С. 260–263.