УДК 683.1

В.М.Галайко, А.М.Пелещишин

НУ "Львівська політехніка", кафедра інформаційних систем та мереж

ІНТЕЛЕКТУАЛЬНА ПОШУКОВО-ІНФОРМАЦІЙНА СИСТЕМА ATHENA

This paper considers intelligent search information system called Athena. Authors describe Athena's components, their structure and functioning algorithms.

Формальна модель Web-системи є теоретичною основою для проектування та розроблення складних INTERNET-орієнтованих інформаційних систем. До таких Web-систем належать й інформаційно-пошукові системи, зокрема система, реалізована в межах проекту "Athena". Серед технологічних та системних особливостей інтелектуальної інформаційної Web-системи Athena, при аналізі та проектуванні яких використовувалась формальна модель, виділимо такі:

- Інформаційно-пошукова система Athena є надбудовою над глобальною множиною Web-серверів і разом з ними (як верхній рівень) формує глобальну Web-систему збереження, пошуку та класифікації інформаційних ресурсів.
- Доступ до всіх послуг системи може здійснюватися з будь-якої точки глобальної мережі, у якій вона встановлена. Всі користувачі цієї мережі є потенційними користувачами загальнодоступних послуг Web-системи Athena.
- Користувачі інформаційної Web-системи Athena поділяються на декілька категорій, що відрізняються за повноваженнями та функціональністю.
- Інформаційно-пошукова система Athena має складну багаторівневу компонентну структуру та містить компоненти із різним рівнем автоматизації та розподілу роботи між машинними алгоритмами та операторами.
- Інформаційно-пошукова система Athena містить інтелектуальну складову, яка використовується для автоматизації процесу опрацювання та класифікації інформації.
- Кожна з компонент Web-системи Athena є автономною, має закінчену структуру та може використовуватися окремо від Web-системи загалом. Компоненти Web-системи Athena є замінними на аналогічні програмно-інформаційні продукти іншого виробництва.
- Інформаційно-пошукова система Athena використовує широкий спектр інформаційних технологій, зокрема: HTML, JavaScript для створення інтерактивного інтерфейсу користувача та відображення даних; СУБД реляційного

Рис. 1. Діаграма потоків даних Web-системи Athena

типу для збереження даних; Java та Internet–роботи для збору даних та засоби штучного інтелекту для їх автоматизованої класифікації.

- Інформаційно-пошукова система Athena може використовуватися як у глобальних мережах типу INTERNET, так і локальних Intranet-мережах.

На рис. 1 наведено діаграму потоків даних Web-системи Athena.

На рис. 2 зображено контекстну діаграму Web-системи Athena.

Розглянемо детальніше компоненти інформаційно-пошукової системи Athena.

1. Каталог інформаційних ресурсів (базова компонента Web-системи)

Каталог інформаційних ресурсів є базою даних про розміщені в INTERNET (або Intranet-мережі) інформаційні ресурси. Разом з Web-серверами мережі утворює глобальну Web-систему обліку, пошуку та збереження інформаційних ресурсів.

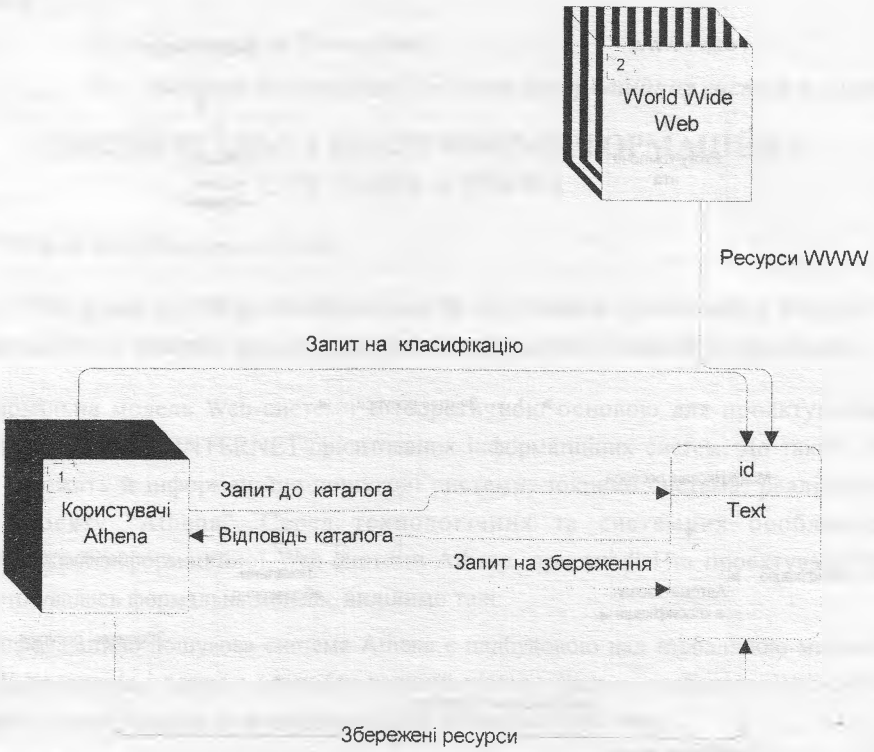

Рис. 2.Контекстна діаграма Web-системи Athena

Каталог передбачає збереження такої інформації про ресурс:

· тип категорії інформаційного ресурсу;

· категорія, до якої належить інформація;

· назва інформаційного ресурсу;

· ключові слова до інформаційного ресурсу;

· універсальний локатор ресурсу (URL), за яким можна знайти оригінал;

· автори інформаційних ресурсів;

· співвідношення між інформаційними ресурсами та авторами.

Категорії інформаційних ресурсів можуть бути вкладеними, тобто певні категорії можуть містити інші (наприклад: до категорії "Розподілені інформаційні системи" може входити категорія "Web-системи").

Каталог передбачає оброблення таких запитів:

· занесення нового типу категорії інформаційного ресурсу;

· занесення нової категорії інформаційного ресурсу;

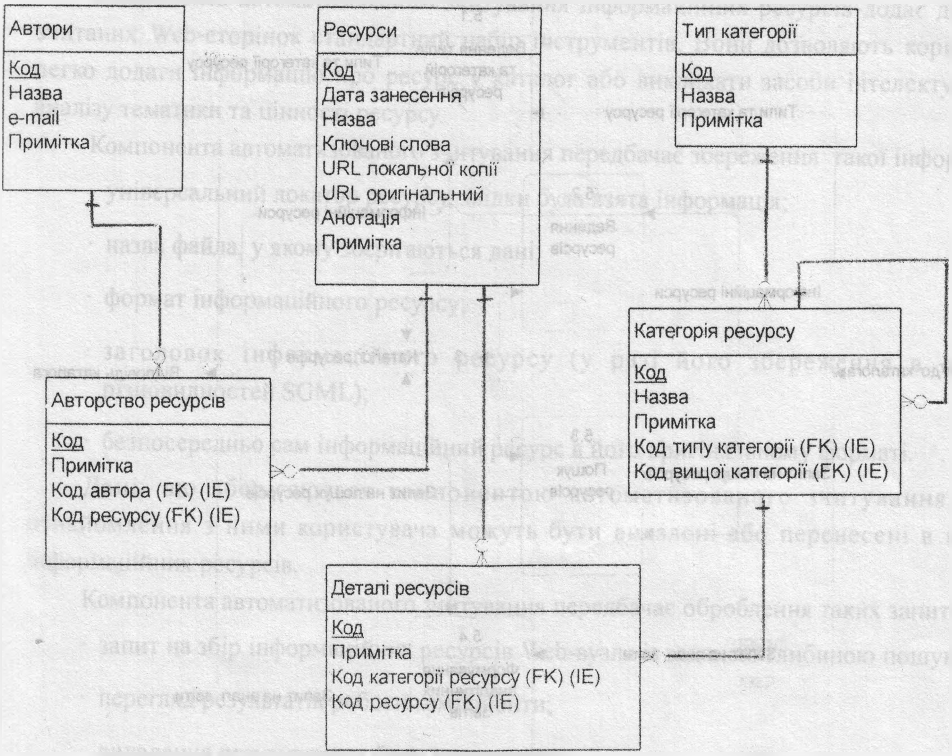

Рис. 3. Інформаційна схема каталогу ресурсів

- занесення даних про новий інформаційного ресурс в базу даних;
- видалення застарілої чи неактуальної інформації про типи, категорії та ресурси;
- пошук необхідних інформаційних ресурсів згідно із заданими критеріями;
- формування аналітичних звітів по вмісту каталогу інформаційних ресурсів.

Пошук необхідних інформаційних ресурсів є найскладнішою функцією каталогу. Він здійснюється шляхом формування складного запиту до бази даних на отримання інформації. Запит формується за допомогою спеціальної QBE-форми. Передбачається неповне визначення певних атрибутів, використання складних критеріїв, масок для текстових стрічок та використання ієрархії категорій. Параметрами запиту є тип категорії, категорія ресурсу, його авторів та ключові слова. Загальна інформаційна схема каталогу ресурсів наведена на рис. 3.

Схему потоків даних каталогу інформаційних ресурсів наведено на рис. 4

Для збереження інформації використовується СКБД Oracle8. Інтерфейс користувача каталогу інформаційних ресурсів реалізується на основі динамічно створюваних сервером Athena HTML-сторінок з використанням технології JavaScript.

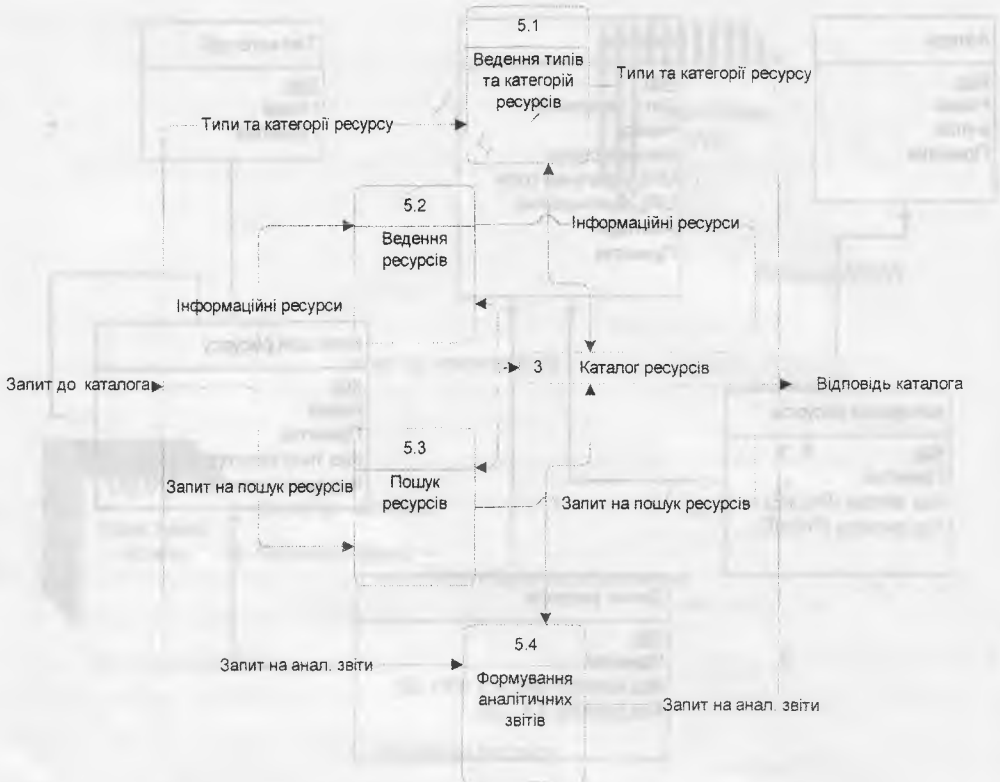

Рис. 4. Потоки даних каталогу інформаційних ресурсів

2. Компонента автоматизованого зчитування інформаційних ресурсів

Компонента автоматизованого зчитування інформаційних ресурсів використовується для спрощення роботи користувача в INTERNET. Під час роботі користувача в глобальній мережі типу INTERNET часто виникають такі проблеми:

- низька пропускна здатність мережі призводить до значних втрат робочого часу користувача при отриманні ресурсів;
- нерівномірне завантаження мережі у різні частини доби призводить до неповного її використання;
- у користувача виникають складнощі при перегляді багатих на гіпертекстові посилання документів (користувач може не простежити усіх посилань і не проаналізувати усіх ресурсів).

Компонента автоматизованого зчитування інформаційних ресурсів дозволяє частково розв'язати усі наведені проблеми. Користувач використовує її для зчитування ресурсів у неробочий час, а сам працює уже з локальними копіями отриманих документів, що при певних умовах (зокрема, якщо значна частина отриманих ресурсів є корисною) економить його час та дозволяє не пропустити цінної інформації. Використання цієї компоненти для високопродуктивних та добре організованих Intranet-систем є недоцільним.

Компонента автоматизованого зчитування інформаційних ресурсів додає до змісту зчитаних Web-сторінок стандартний набір інструментів. Вони дозволяють користувачу легко додати інформацію про ресурс у каталог або викликати засоби інтелектуального аналізу тематики та цінності ресурсу.

Компонента автоматизованого зчитування передбачає збереження такої інформації:

- універсальний локатор ресурсу, звідки була взята інформація;
- назва файла, у якому зберігаються дані;
- формат інформаційного ресурсу;
- заголовок інформаційного ресурсу (у разі його збереження в одній з різновидностей SGML);
- безпосередньо сам інформаційний ресурс в його оригінальному форматі.

Дані, що зберігаються компонентою автоматизованого зчитування, після ознайомлення з ними користувача можуть бути видалені або перенесені в каталог інформаційних ресурсів.

Компонента автоматизованого зчитування передбачає оброблення таких запитів:

- запит на збір інформаційних ресурсів Web-вузла із заданою глибиною пошуку;
- перегляд результатів роботи компоненти;
- видалення результатів роботи компоненти;
- передача знайдених даних каталогу інформаційних ресурсів;
- передача знайдених даних інтелектуальній компоненті для опрацювання та подальшого збереження в каталозі інформаційних ресурсів;
- отримання аналітичних звітів про використання компоненти.

Компонента автоматизованого зчитування передбачає формування запиту на отримання інформаційного ресурсу з World Wide Web.

Описувана компонента реалізована на основі Java-технології, що забезпечує їй високий рівень крос-платформності. Тому ця компонента може отримуватися користувачами безпосередньо через мережу INTERNET.

Компонента автоматизованого зчитування інформаційних ресурсів належить до програмних продуктів класу Web-роботів і може бути у разі потреби замінена на аналогічний продукт іншого виробника.

3. Компонента автоматизованої класифікації інформаційних ресурсів

Опрацювання великих об'ємів інформації, отриманих за допомогою компоненти автоматизованого збору інформаційних ресурсів, вимагає значних людських затрат для їх коректного впорядкування та правильної класифікації.

Для спрощення і автоматизації роботи користувача із класифікації отриманої інформації використовується інтелектуальна компонента. У своїй роботі вона застосовує

технології штучного інтелекту, зокрема механізм штучних нейронних мереж. Інтелектуальна компонента виступає в ролі системи підтримки і прийняття рішень на етапі класифікації. Вона застосовується до окремо взятого ресурсу та, використовуючи засоби та методи семантичного аналізу, пробує визначити категорії, до яких належить розглядуваний ресурс.

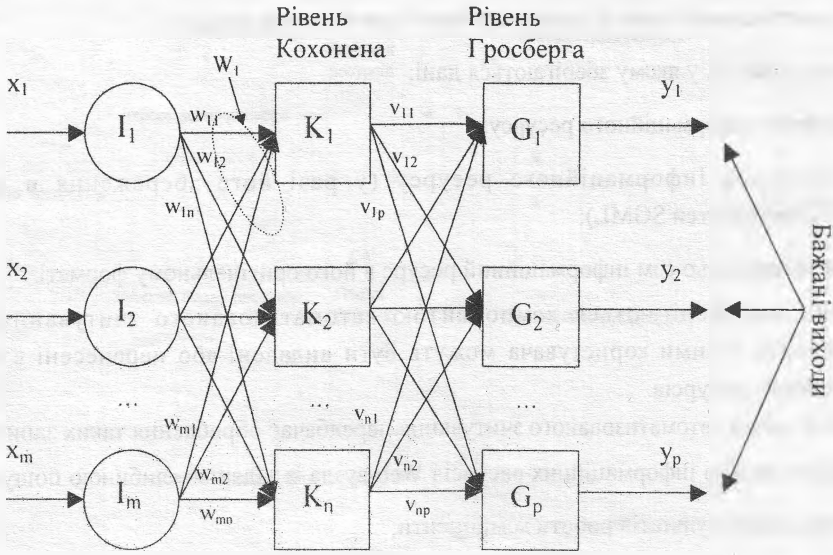

Рис. 5. Модель нейронної мережі за методом прямого поширення

В своїй роботі інтелектуальна компонента використовує нейронні мережі за методом прямого поширення. Метод прямого поширення є комбінацією двох добре відомих алгоритмів: карт самоорганізації Кохонена та зірки Гросберга. Особливістю цього методу є те, що він потребує меншого часу на навчання, а тому застосовується для вирішення проблем, які не терплять довгого навчального процесу. Загальна структура мережі подана на рис. 5.

Робота нейронної мережі складається з двох етапів: початкового навчання і безпосередньої роботи з опрацювання вхідної інформації. Навчивши мережу один раз, надалі її використовують для інтелектуального опрацювання вхідної інформації. Інтелектуальна робота нейронної мережі полягає в тому, що вона продукує коректний результат навіть у тому випадку, коли вхідна інформація є частково неповною або спотвореною.

Розглянемо детальніше етапи навчання і роботи нейронної мережі для класифікації інформаційних ресурсів.

3.1 Робота нейронної мережі

3.1.1 Робота рівня Кохонена

У найпростішому вигляді на рівні Кохонена для кожного вхідного вектора один і лише один нейрон продукує логічну одиницю, а всі інші — нулі. З кожним нейроном

рівня Кохонена пов'язується набір вагових коефіцієнтів по кожному із входів. Так, наприклад, нейрон k_1 має вагові коефіцієнти w_{11}, w_{21}, ... w_{m1}, які об'єднуються у вектор W_1.

Вхідні сигнали у вхідному рівні об'єднуються у вхідний вектор $X=\{x_1,x_2,.. x_m\}$. Вихід NET кожного нейрона є простим підсумовуванням його входів, перемножених на відповідні вагові коефіцієнти. Так,

$$NETj = X_1W_{1j} + X_2W_{2j} + ... + X_mW_{mj} \quad (1)$$

де NET_j – це є NET вихід j-го нейрона Кохонена

$$NET_j = \sum_i X_iW_{ij} \quad (2)$$

або у векторному вигляді

$$N = XW \quad (3),$$

де N — вектор NET виходів рівня Кохонена. Вихід нейрона з найбільшим значенням NET встановлюється в одиницю, а всі інші — в нуль.

3.1.2 Рівень Гросберга

Дії, що відбуваються на цьому рівні, подібні до функцій попереднього рівня. NET виходи рівня Кохонена k_1, k_2, ... k_n формують вектор K. Вагові коефіцієнти цього рівня v_{11}, v_{21}, ... v_{np} об'єднуються у вектор V. Тоді NET виходи кожного нейрона рівня Гросберга мають вигляд:

$$NET_j = \sum_i k_j v_{ij} \quad (4)$$

або у векторній формі

$$Y = KV \quad (5),$$

де:

Y — вихідний вектор рівня Гросберга

K — вихідний вектор рівня Кохонена

V — матриця вагових коефіцієнтів рівня Гросберга.

Оскільки лише один нейрон рівня Кохонена дає на виході одиницю, то фактично вся робота на рівні Гросберга полягає в обчисленні значень вагових коефіцієнтів, що взаємодіють з ненульовим нейроном Кохонена, а отже, обчислення є найпростішими.

3.2 Навчання нейронної мережі

Для того, щоб мережа правильно працювала, потрібно попередньо її навчити на визначеному наборі вхідної інформації. Процес навчання для кожного з рівнів відбувається наступним чином.

3.2.1 Навчання рівня Кохонена

На рівні Кохонена вхідні вектори розподіляються по групах подібності. Вхідні коефіцієнти встановлюються таким чином, що подібні вхідні вектори активізують

однакові нейрони. Це дає можливість надалі на рівні Гросберга продукувати бажані виходи.

Навчання рівня Кохонена є самоорганізуючим алгоритмом, що відбувається без контролю з боку людини. З цієї причини важко прогнозувати, який нейрон буде активований вхідним вектором. Необхідно лише впевнитися у тому, що в процесі тренування розрізняються неподібні вхідні вектори.

Бажано нормалізувати всі вхідні вектори перед їх використанням в мережі. Це здійснюється діленням кожної компоненти вхідного вектора на векторну довжину.

$$x_i' = x_i / \sqrt{x_1^2 + x_2^2 + ... + x_n^2} \qquad (6)$$

Тренувальний процес полягає у тому, що вибираються нейрони, чиї вагові коефіцієнти є найбільш подібні до вхідного вектора з їх подальшим коригуванням з метою відповідності вхідному набору. Мережа самомодифікується таким чином, що нейрон Кохонена дає максимальне вихідне значення для вхідного вектора. Тренувальний процес записується у вигляді:

$$w_{new} = w_{old} + \alpha(x - w_{old}) \qquad (7)$$

де w_{new} — нове значення вагового коефіцієнта нейрона, який продукує максимальне значення на виході.

w_{old} — старе значення цього нейрона.

α — тренувальний коефіцієнт.

Ваговий коефіцієнт, що асоціюється з мажоритарним нейроном, змінюється пропорційно до різниці між ним і значенням вхідного вектора.

Змінна α є тренувальниим коефіцієнтом, що набуває спочатку значення 0.7, поступово зменшуючись в процесі тренування. Це забезпечує великі початкові кроки для швидкості при "грубому" навчанні і менші кроки для навчання на кінцевому етапі.

Всім ваговим коефіцієнтам мережі мусить бути присвоєно початкове значення до початку тренування. Переважно ваговим коефіцієнтам присвоюються випадкові значення малої розмірності. Для рівня Кохонена випадкові вагові вектори мають бути нормалізованими. Після навчання вагові вектори мусять дорівнювати нормалізованим вхідним векторам. Отож нормалізовані одиничні вектори наблизять вагові коефіцієнти до їх кінцевого стану і таким чином зменшать час тренувального процесу.

3.2.2 Навчання рівня Гросберга

Рівень Гросберга доволі простий в тренуванні. Застосовуючи вхідний вектор, отримується вихід (виходи) рівня Кохонена і обчислюється кожен ваговий коефіцієнт рівня Гросберга. Далі встановлюється кожен ваговий коефіцієнт, який з'єднаний з нейроном рівня Кохонена, що має ненульовий вихід. Значення коефіцієнта буде пропорційним до різниці між ваговим коефіцієнтом і бажаним виходом нейрона рівня Гросберга.

$$v_{ijnew} = v_{ijold} + \beta(y_j - v_{ijold})k_i \qquad (8)$$

де k_i — вихід нейрона і рівня Кохонена (лише один нейрон Кохонена з ненульовим виходом),

y_j — компонента j вектора бажаних виходів.

Навчальний коефіцієнт β встановлюється в 0.1 і поступово зменшується в процесі навчання.

Отож, навчання рівня Гросберга є підконтрольним. Для нього задаються бажані виходи, до досягнення яких тренується мережа.

В інформаційно-пошуковій системі Athena класифікація інформації за допомогою нейронної мережі виконується у такий спосіб. На перших етапах роботи інтелектуальної компоненти інформаційно-пошукової системи Athena користувач повинен навчити нейронну мережу класифікації вхідної інформації. На вхід системи подається текстова інформація (заголовки HTML-сторінок або їх короткий опис), яка сприймається нейронною мережею у бінарному вигляді. У випадку навчання користувач повинен сам визначити, до якої категорії віднести інформаційний ресурс. Якість класифікації інформаційного ресурсу безпосередньо залежить від рівня та якості навчальних зразків і часу навчання компоненти, яке здійснюється в процесі її використання. Чим більше прикладів буде запропоновано нейронній мережі на етапі навчання — тим кращі будуть результати її роботи.

У процесі роботи нейронна мережа намагається самостійно ідентифікувати категорію, до якої слід віднести інформаційний ресурс. Якщо результат класифікації не задовольняє користувача, він може тут же скоректувати роботу інтелектуальної компоненти, здійснивши тим самим процес навчання новому прикладу.

Далі результати роботи передаються в каталог інформаційних ресурсів, де відбувається занесення і збереження нової інформації.

Компонента автоматизованої класифікації інформаційних ресурсів значно спрощує роботу користувача Web-системи "Athena" при її спільному використанні із компонентою автоматизованого зчитування.

4. Компонента локального збереження інформаційних ресурсів

Компонента локального збереження інформаційних ресурсів використовується для зменшення сумарного трафіку в INTERNET. Ця компонента використовується, зокрема, у випадку, коли користувачі системи Athena використовують для доступу в INTERNET високопродуктивну локальну мережу, приєднану до INTERNET. У такому разі доцільним є реплікація ресурсу у локальній мережі, де він буде доступним для користувачів без використання INTERNET-з'єднання.

База даних компоненти локального збереження містить таку інформацію:

- оригінальне значення універсального локатора ресурсу;
- безпосередньо сам інформаційний ресурс в його оригінальному форматі.

Компонента локального збереження передбачає оброблення таких запитів:

- запит на локальне розміщення інформаційного ресурсу;
- запит на отримання локальної копії інформаційного ресурсу;
- видалення локальної копії інформаційного ресурсу;
- отримання аналітичних звітів про використання компоненти.

Компонента автоматизованого зчитування передбачає формування запиту на отримання інформаційного ресурсу з World Wide Web.

Описувана компонента реалізована на основі Java-технології, що забезпечує їй високий рівень крос-платформності. Тому цю компоненту можуть отримувати користувачі безпосередньо через мережу INTERNET.

Компонента автоматизованого зчитування інформаційних ресурсів належить до програмних продуктів класу Web-роботів і може бути у разі потреби замінена на аналогічний продукт іншого виробника.

5. Proxy-компонента

Proxy-компонента використовується для модифікації зчитуваних користувачем інформаційних ресурсів глобальної мережі з метою додавання до них стандартного набору інструментів. Вони забезпечують користувачу легкий доступ до основних функцій компонент Web-системи "Athena".

Інструменти передбачають доступ до запитів:

- на поповнення каталогу інформаційних ресурсів;
- на запуск компоненти автоматизованого зчитування інформаційних ресурсів;
- на навчання нейронної мережі компоненти автоматизованої класифікації інформаційних ресурсів;
- на автоматизовану класифікацію інформаційного ресурсу;
- на отримання допомоги по системі Athena.

Proxy-компонента не заміняє собою існуючих Proxy-серверів, основною метою яких є оптимізація мережевого трафіку Web-системи та підвищення рівня безпеки Web-системи. При встановленні Proxy-компонента каскадно приєднується до інших Proxy-серверів.

Proxy-компонента передбачає оброблення запиту на отримання інформаційних ресурсів INTERNET. Отримання запиту ініціює формування компонентою запиту до системи World Wide Web (або аналогічної глобальної Web-системи). Отриманий інформаційний ресурс модифікується Proxy-компонентою через доповнення ресурсу панеллю інструментів.

1. Konopnicki D., Shmueli O. W3QS: A Query System for the World-Wide Web. *VLDB'95, Proceedings of 21th International Conference on VeryLarge Data Bases, September 11-15, 1995, Zurich, Switzerland.54-65 phttp://www.informatik.uni-rier.de/~ley/db/conf/vldb/KonopnickiS95.html. 2. Галайко В.М.* Розроблення інтелектуальних Web-систем *//Вісник Державного університету "Львівська політехніка" 1998, №330. С.53-62. 3. Горбань А.* Нейроинформатика и ее приложения *//Открытые системы 1998. №4-5. С.36-41. 4. Пелещишин А.* Використання апарату абстрактних автоматів для моделювання Web-систем *//Вісник Державного університету "Львівська політехніка" 1998. №330. С.188-201. 5. Пелещишин А.* Побудова формальної моделі Web-системи. *Задачі та методи прикладної математики //Вісник Львівського університету. Львів 1998. С.182-185. 6. Пелещишин А.* Розробка інформаційних систем у Web-середовищах *//Вісник Державного університету "Львівська політехніка" 1997. №315. С.193-208. 7. Пелещишин А.М.* Напрямки використання Web-технологій при побудові інформаційних систем у сфері виробництва та бізнесу *//Вісник Державного Університету "Львівська Політехніка" 1998. №330. С.202-211. 8. Уоссермен Ф.* Нейрокомпьютерная техника: Теория и практика.- *М.: Мир.- 1992 - 240 с. 9. Эйнджел Дж.* Proxy-серверы. *LAN//Журнал сетевых решений. 1999. №6, http://www.osp.ru/lan/1999/06/016.htm. 10. Флореску Д., Леви А.,Мендельсон А.* Технологии баз данных для World Wide Web: обзор. *СУБД №4-5 1998. http://www.osp.ru/dbms/1998/04_05/01.htm. 11. Хехт-Нильсен Р.* Нейрокомпьютинг: история, состояние, перспективы. *//Открытые системы 1998. №4-5. С.23-28. 12. Шапот М., Рощупкина В.* Интеллектуальный анализ данных и управление процессами. *// Открытые системы.№4-5, 1998. С.29-35. 13. Hecht-Nielsen R. Neurocomputing. Addison-Wesley, 1989. 14. Konopnicki D., Shmueli O. W3QS:* A query system for the World Wide Web *// In Proc. of the Int. Conf, on Very Large Data Bases (VLDB), Pp. 54-65, Zurich, Switzerland,1995. 15. Konopnicki D. Shmueli O.* Bringing database functionality to the WWW / *In Proceedings of the International Workshop on the Web and Databases, Valencia,Spain, Pp. 49-55, 1998. 16. Mendelzon A., Mihaila G., MiloT.* Querying the world wide web. *International Journal on Digital Libraries, 1(1):54-67, April 1997. 17. Li W., Shim J., Selcuk K., Hara Y. WebDB*: A web query system and its modeling, language, and implementation *//In Proc. IEEE Advances in Digital Libraries'98, 1998.*