2004. – Pp. 1083–1086. 7. Dave K. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification in Product Reviews / K. Dave, S. Lawrence, D. Pennock // Proceedings of ACM WWW2003. – Budapest, 2003. – Pp. 519–528. 8. Turney P. Measuring Praise and Criticism: Inference of Semantic Orientation from Association / P. Turney, M. Littman // ACM Transactions on Information Systems. – 2003. – 4(21). – pp. 315–346. 9. Яндекс Допомога: Пошуковий контекст [Електронний ресурс]. – Режим доступу: http://help.yandex.ua/search/?id=1111438. 10. Google Guide Quick Reference: Google Advanced Operators [Електронний ресурс]. – Режим доступу: http://www.googleguide.com/using_advanced_operators.html. 11. Market Share of Forums: A Pie Chart of The Most Common Forums [Електронний ресурс]. – Режим доступу: http://www.qualityposts.com/ForumMarketShare.php. 12. The Top 5 Forum Platforms Compared [Електронний ресурс]. – Режим доступу: http://www.webhostingreport.com/learn/top-five-forum-platforms-compared.html. 13. Yahoo! Search Help: Search Tips [Електронний ресурс]. – Режим доступу : http://help.yahoo.com/l/us/yahoo/search/basics/basics-04.html.

УДК 004.415, 004.6, 004.738.5

В.В. Шендрик, С.М. Ващенко
Сумський державний університет,
кафедра комп'ютерних наук

СИСТЕМА ЗБИРАННЯ, РОЗМІЩЕННЯ ТА АНАЛІЗУ ДАНИХ

Розглянуті особливості структури Web-сторінок, запропановано метод структурування та розміщення в базу даних неструктурованої інформації з Інтернету для виконання подальшого аналізу.

Ключові слова: HTML, база даних, парсер, спектральний аналіз Фур'є, вейвлет-перетворення, кальманівська фільтрація.

This paper is deals with peculiarities of the structure of Web-pages, proposes a method of structuring and placement in the database of unstructured information from the Internet to perform the analysis.

Key words: HTML, database, parser, Fourier spectral analysis, wavelet transform, Kalman filter.

Вступ

Питання про володіння інформацією, створювання та керування інформаційними потоками стає все більш актуальним, адже підприємства та організації в сучасних умовах змушені працювати з великими обсягами інформації, джерелом якої може слугувати всесвітня мережа. Нагромаджена інформація з надзвичайною швидкістю помножується та змінюється, створюючи при цьому «інформаційний хаос». Використання та поширення значних масивів різноманітної інформації спонукає до створення нових автоматизованих систем збору, упорядкування та подальшого аналізу необхідних даних.

Першочерговим завданням під час опрацювання інформації є створення системи, що здатна перетворювати інформацію з неактивної форми на web-сторінках в активну. Тобто інформація може бути перетворена в згруповану та структуровану за допомогою переведення в реляційну форму та розміщена у базі даних, що зручно для подальшого аналізу.

Постановка проблеми та аналіз останніх досліджень

Наразі сформувалися дві основні технології опрацювання інформації, розміщеної на web-сторінках. Це пошукові системи та парсери.

Пошукові системи працюють за таким алгоритмом:

- Браузероподібна програма–павук викачує web-сторінки.
- Програма-краулер автоматично проходить за всіма посиланнями, знайденими на сторінці.

- Програма-індексатор аналізує web-сторінки, викачані павуками.
- Інформація з викачаних та опрацьованих сторінок розміщується у сховище даних.
- Система видачі результатів отримує результати пошуку з бази даних.
- Web-сервер здійснює взаємодію між користувачем та іншими компонентами пошукової системи.

Реалізації пошукових механізмів можуть відрізнятися один від одного у деяких деталях, проте всім пошуковим системам властиві описані загальні ознаки.

Парсери являють собою програми або частину програм, які виконують синтаксичний аналіз web-сторінок, виокремлюють потрібні елементи. У зв'язку з тим, що кожна сторінка має унікальну структуру – не існує єдиного універсального парсера, який би працював із будь-якою web-сторінкою.

При використанні web-сторінок, як джерела інформації, та при автоматизації процесу збору даних виникають труднощі, а саме:

- сайти не мають чіткої структури;
- відсутність схеми, яка б описувала структуру сайта;
- відсутність універсального алгоритму зчитування даних з неструктурованої інформації.

Все це зумовлює необхідність розробки систем, які б базувалися на універсальних методах структурування та опрацювання інформації.

Мета та задачі дослідження

У сучасних інформаційних технологіях роль такої процедури, як витяг інформації, усе більше зростає – через стрімке збільшення кількості неструктурованої інформації, зокрема, в Інтернеті.

Тому об'єктом дослідження є структура web-сторінок з неупорядкованою табличною інформацією, а предметом дослідження – метод структурування неупорядкованої інформації з web-сторінок.

Мета роботи полягає у створенні нового універсального способу структурування незгрупованої та неструктурованої HTML-інформації у вигляді таблиць для виділення необхідних даних та переведення їх у згруповану структуру баз даних, проведенні аналізу даних. При цьому треба врахувати, що запропоновану технологію обробки інформації необхідно реалізувати у вигляді системи, яка повинна бути універсальною, тобто працювати з будь-яким сайтом та будь-якою базою даних.

Задачі дослідження:

- Вивчити структуру сайтів в Інтернеті, які можуть бути джерелами корисної інформації.
- Виділити загальну закономірність в структурі джерел інформації.
- Створити універсальний парсер, здатний розбирати цю структуру та зчитувати необхідні табличні дані.
- Створити систему, здатну в зручній формі представляти дані, наповнювати вибрані таблиці баз даних або створювати нові таблиці необхідні для подальшого аналізу.

Розроблювана система збору даних і аналізу інформації повинна:

- Забезпечувати необхідну оперативність процесу збору даних від моменту появи нової потреби у звітній інформації до повного збору і консолідації даних з усіх джерел.
- Знижувати обсяг ручної праці щодо заповнення та контролю данних за рахунок надання розвинених засобів контролю.
- Надавати розвинені засоби контролю за дотриманням встановленого порядку підготовки даних до аналізу.

Відповідно автоматизована система збору даних та аналізу інформації повинна дозволяти:

- Знижувати велике навантаження на персонал в частині ручного введення, збору і контролю даних.
- Підвищувати оперативність процесу збору.
- Запобігати дублювання операцій з підготовки звітних матеріалів.
- Підвищувати ефективність використання зібраних звітних даних, зокрема, за рахунок введення доступних гнучких засобів аналізу даних.

Аналіз структури web-сторінок

Витяг інформації полягає в скануванні набору документів, написаних мовою HTML, та заповненні баз даних відібраною корисною інформацією.

Будь-який документ мовою HTML є набором елементів, які позначаються спеціальними позначками (тегами). Ім'я тегу визначає тип елемента та правила розмітки. Регістр, в якому набрано ім'я тегу, в HTML значення не має. Набір і рекомендовані інтерпретації тегів визначені організацією W3C. Використовують тільки два види тегів – відкриваючий, або початковий, і закриваючий, або кінцевий, або ще додатково залежно від реалізації мови можливе застосування одиночного тегу та тегу порожнього елемента (що не містить ніякого тексту та інших даних – у цьому випадку зазвичай не вказується закриваючий тег). Крім того, елементи можуть мати атрибути, що визначають будь-які їх властивості. Атрибути вказуються в відкриваючому тегу та дають додаткові можливості форматування тексту. Вони записуються у вигляді пари ім'я-значення, причому нечислове значення розміщується у лапках.

HTML не має суворої синтаксичної структури та є неструктурованим текстом, і ця властивість не дає можливість обробити документ – виконати трансформацію даних, пошук потрібних елементів документа і т.д. Також у мові гіперпосилання спостерігається вкладення елементів вищого рівня. Все це дещо ускладнює розбір синтаксису для пошуку та зчитування даних.

Наше завдання – визначити ієрархію секцій в HTML документі, використовуючи різні HTML теги. HTML був створений не тільки для визначення, але й для відображення даних, таким чином, більшість HTML документів не сприяють організації компонентів HTML у секції або блоки відповідно до ієрархії. Тому перше завдання полягає в ідентифікації HTML тегів, які можуть бути використані для конструювання ієрархічної структури HTML документів (тип 1), та тегів, які слугують для подання даних (тип 2). Список тегів з поділом за типами можна знайти в табл. 1.

Таблиця 1

Список тегів за типами

HTML-теги			Тип 1	Тип 2
Head			TITLE, META	ISINDEX, BASE, LINK, SCRIPT, STYLE, META
Body	Заголовки		H1,H2,H3,H4,H5,H6	
	Блоки		P, CENTER, BLOCKQUOTE, PRE, DIR, MENU, DL, DT, DD, UL, OL, LI, TABLE, CAPTION, THEAD, TBODY, TR, TH, TD	ISINDEX, HR, DIV
	Текст	Шрифт		TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP
		Фраза		EM, STRONG, DFN, CODE, SAMP, KBD,VAR, CITE
		Спеціальний	IMG	A, APPLET, FONT, BASEFONT, BR, SCRIPT, MAP
		Форма		FORM, INPUT, SELECT, TEXTAREA
	Адреса		ADDRESS	

Принципи створення синтаксичного аналізатора

Об'єктна модель документа (англ. Document Object Model, DOM) – специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами. З точки зору об'єктно-орієнтованого програмування, DOM визначає класи, методи та атрибути цих методів для аналізу структури документів та роботи з представленням документів у вигляді дерева. Все це призначено для того, аби надати можливість комп'ютерній програмі виконувати доступ та динамічну модифікацію структури, змісту та оформлення документа.

Синтаксичний аналізатор (парсер) – це програма або частина програми, яка виконує синтаксичний аналіз. Під час парсингу текст оформлюється у структуру даних, зазвичай – в DOM-дерево, яке відображає синтаксичну структуру вхідної послідовності, та зручніше для подальшої обробки.

Зазвичай парсери працюють в два етапи: на першому ідентифікуються осмислені токени (виконується лексичний аналіз), на другому створюється дерево розбору.

Через те, що структура документа представляється у вигляді дерева, повний зміст документа аналізується та зберігається в пам'яті комп'ютера. Тому, DOM підходить для застосувань в програмах, які вимагають багаторазового доступу до елементів документа в довільному порядку.

Оскільки у цій роботі необхідно структурувати табличну інформацію з web-сторінок, розглянемо табличні елементи HTML. Серед табличних елементів TR визначає число рядків, тоді як TH та TD визначають число стовпців в HTML таблиці.

Елемент TH використовується для задавання одного або більше заголовків.

Елемент TD використовується для внесення даних в комірки таблиці. Будемо надалі називати дані в елементах TD табличними даними на відміну від даних, що знаходяться в елементах TH, які будемо називати заголовками.

Типова HTML таблиця має, як мінімум, один стовпець – заголовок у верхній частині таблиці, і як мінімум, один рядок заголовок у лівій частині. Такий тип таблиць назвемо строково-стовпцевим.

Інший тип таблиці містить, як мінімум, один стовпець заголовок або один рядок заголовок і називається в цьому випадку стовпцевим або рядковим типом таблиці.

Заголовки в рядкових та стовпцевих таблицях задають схему таблиці. Для будь-яких таблиць, які не мають елементів TH, у ході аналізу було виявлено, що перший рядок використається як заголовок.

Серед табличних елементів TH та TD два атрибути – ROWSPAN й COLSPAN – відіграють істотну роль у визначенні ієрархії HTML таблиць. Коли TH або TD включає ROWSPAN = "n" (або COLSPAN = "n"), зв'язування комірок таблиці поширюється на n стовпців униз (або n рядків вправо).

Для визначення семантичної ієрархії, що розширює синтаксичне дерево будь-якої HTML таблиці, у першу чергу визначимо ієрархічні залежності даних. Коли вони визначені, залишаються тільки дані, і всі теги з таблиці виключаються. Семантична ієрархія HTML таблиці визначається відповідно до нотації псевдотаблиці, тому що властивості псевдотаблиці легкі для сприйняття. Псевдотаблиця може розглядатися як особливий тип HTML таблиці та може бути використана для вираження строково-стовпцевих, рядкових і стовпцевих таблиць. Загальний підхід побудови семантичної ієрархії – це, у першу чергу, відбиття таблиці T в псевдотаблицю та потім одержання з неї ієрархії.

Як уже згадувалося, HTML таблиця може мати різну кількість стовпців у рядках відповідно до використання атрибутів COLSPAN та ROWSPAN. Якщо елемент TH або TD містить COLSPAN = "n", то відповідна комірка TH або TD розширюється на n стовпців і займає, таким чином, n комірок, включаючи поточну комірку у поточному рядку. Отже, можна вважати, що вставлено n–1 комірок вправо від поточної комірки, і в них знаходяться дані поточної комірки. ROWSPAN функціонує інакше. Якщо елемент TH містить ROWSPAN = "n", то конкретна комірка розширюється на наступні n–1 рядків і займає n комірок. У цьому випадку вставляються n–1 комірок нижче поточної комірки, і вміст поточної комірки h не займає всі комірки, а розміщує тільки у вставлену n–1 комірку записане h. Отже, h з'являється тільки в комірці n–1, всі інші вставлені комірки залишаються порожніми. Це необхідно для збереження коректних взаємозв'язків табличних даних у всіх рядках у стовпцях та уникнення повторення того самого заголовка, тому що об'єднані заголовки в стовпці HTML таблиці перетворяться в заголовок стовпця псевдотаблиці. Однак якщо TD містить ROWSPAN, то додаються n–1 нових комірок нижче поточної комірки TD, і в них розміщуються дані поточної комірки для того, щоб дані в кожному з n різних рядків того самого стовпця були однаковими. Після того, як обробка COLSPAN та ROWSPAN пройшла успішно, необхідно пересвідчитися, що результуюча таблиця задовольняє визначенню псевдотаблиці.

Псевдотаблиця – це таблиця, яка має правильну структуру і доступна для зчитування даних. Семантична ієрархія HTML таблиці визначається відповідно до нотації псевдотаблиці. Псевдотаблиця може розглядатися як особливий тип HTML таблиці та може бути використана для вираження рядково-стовпцевих, рядкових і стовпцевих таблиць. До псевдотаблиці можна

звертатися за індексами. З кожною таблицею може бути зв'язаний заголовок. Рядки таблиці можуть групуватися в розділи заголовків, нижні заголовки і тіла. При відображенні довгих таблиць інформація із заголовків може повторюватися на кожній сторінці таблиці.

Модель таблиць HTML дозволяє упорядковувати дані та текст. Із кожною таблицею може бути зв'язаний заголовок. Рядки таблиці можуть групуватися в розділи заголовків, нижніх заголовків і тіла. Групи рядків несуть додаткову структурну інформацію та можуть генеруватися агентами користувачів різними способами, що відображають цю структуру. Агенти користувачів можуть розбити підрозділ на заголовки/тіло/нижні заголовки для підтримки прокручування тіла таблиці незалежно від заголовків. При відображенні довгих таблиць інформація із заголовків може повторюватися на кожній сторінці таблиці.

Для початку парсингу потрібно отримати доступ до документу та отримати кореневий елемент дерева тегів – у даному випадку це тег <html>. Потім, за алгоритмом потрібно дізнатися, які елементи містяться в структурі під кореневим елементом. Якщо вони є, то зчитуємо їх, звіряємо чи є той елемент потрібною таблицею, якщо так, то вибираємо з неї всі необхідні дані, інакше перевіряємо чи має поточний елемент дочірні, якщо так, то робимо його поточним елементом і запускаємо функцію рекурсивно. Цей процес повторюється доти, доки не буде знайдено потрібну таблицю або елемент, який не має дочірніх елементів. Після цього повертаємося до того елементу, де почалося розгалуження. Процес, алгоритм якого зображений на рис. 1 триває доти, поки не буде знайдена таблиця з необхідними даними.

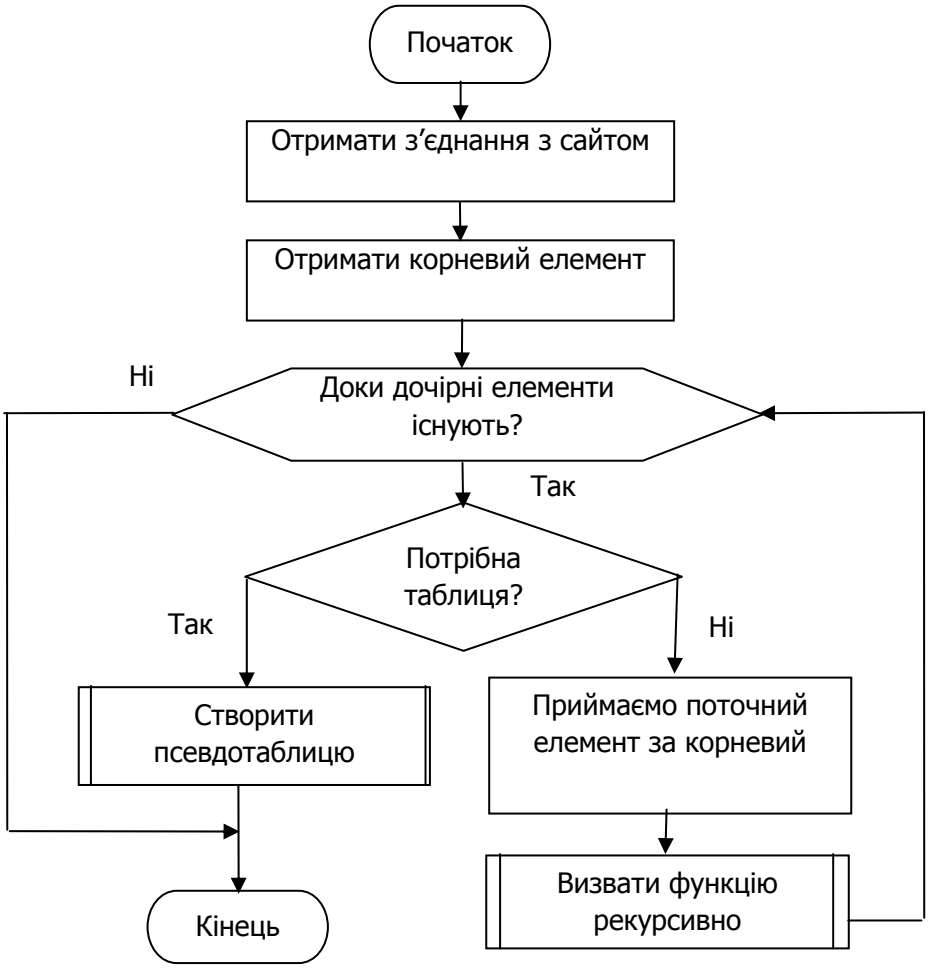

Рис. 1. Алгоритм зчитування даних

Коли парсер знаходить потрібну таблицю для зчитування, він передає керування наступній функції, яка аналізує структуру таблиці та створює псевдотаблицю. З псевдотаблиці можна звертатися до елементів за індексом рядка та стовпця.

При створенні псевдотаблиці парсер враховує атрибут тегу <TD> COLSPAN=n, який об'єднує n комірок в одну по горизонталі, та ROWSPAN=n, який об'єднує n комірок в одну по вертикалі. Також парсер враховує ситуацію, коли таблиця вставлена в другу таблицю. При цьому дані передаються на вищий рівень в основну комірку псевдотаблиці. Алгоритм роботи парсера по створенні псевдотаблиці наведений на рис. 2.

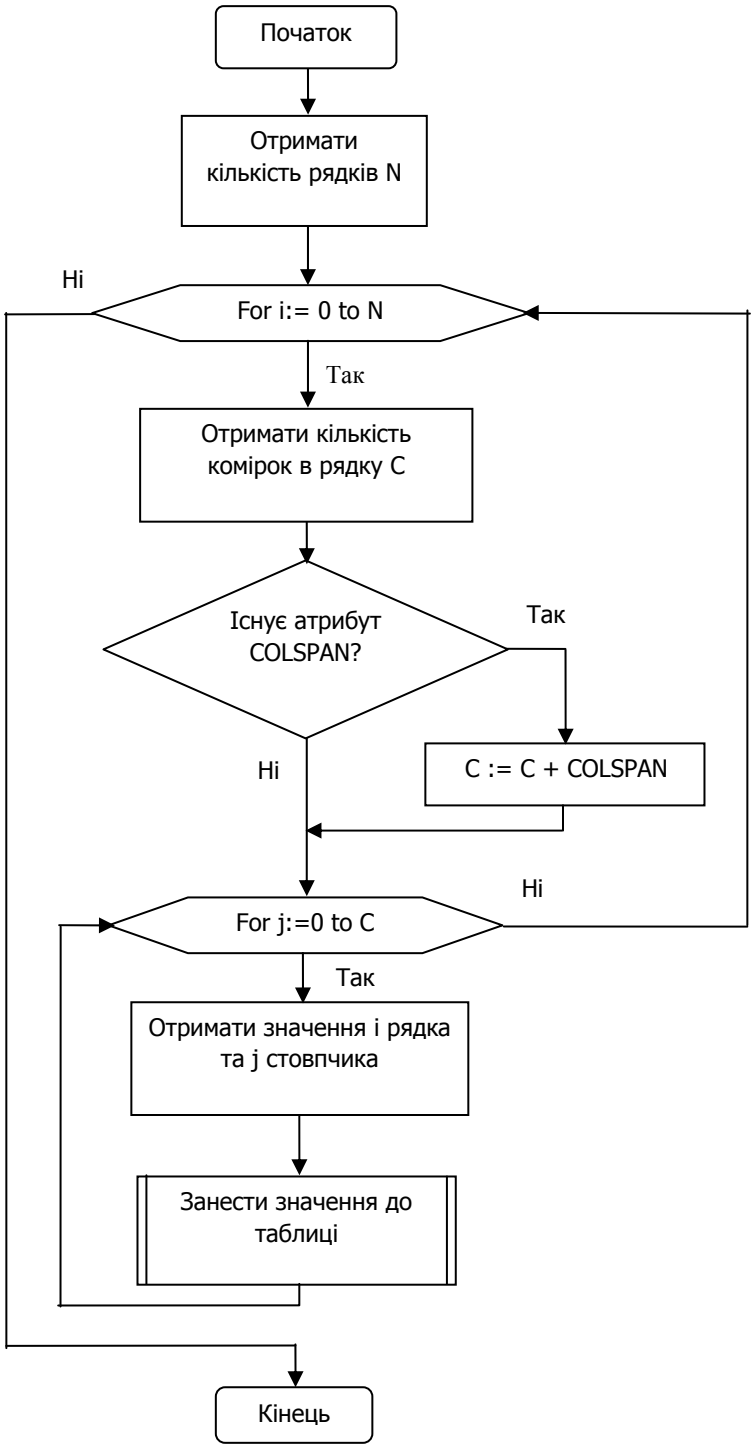

Рис. 2. Алгоритм роботи парсера під час створення псевдотаблиці

Загальна модель функціонування системи збирання, розміщення та аналізу даних

Система складається з декількох окремих компонентів-модулів (рис. 3), що дозволяє гнучко налаштувати параметри збору інформації для баз даних різного структурного та інформаційного наповнення, а саме:

- Модуль зчитування - здійснює зчитування web-сторінок відповідно до завдання.
- Модулі структурування - перетворює дані неструктуровані в структуровані.
- База даних (БД), у якій зберігаються результати зчитування та перетворення.
- Планувальник – управляє процесом збору даних: формує завдання на зчитування і обробку відповідно до настроювань.
- Аналітичний модуль – модуль, який дає можливість проводити аналіз даних, переданих для аналізу з бази даних.

Рис. 3. Схема роботи системи збирання, розміщення та аналізу даних

Модуль зчитування (парсер) отримує доступ до сайта в режимі зчитування, після чого він розбирає елементи web-сторінки, знаходить вказану таблицю та зчитує дані, які передаються на обробку системі управління базою даних. Система має планувальник, який організовує зчитування інформації та передачу її для аналізу в автоматичному режимі через заданий час. Таким чином дані, зчитані з web-сторінки, проходять процедуру структурування розміщуються у базі даних, а потім вже структурована інформація передається на обробку аналітичному модулю.

Аналіз даних

Аналітичний модуль містить набір бібліотек та компонентів, які дають змогу проводити аналіз та прогнозування інформації. Аналітичний модуль можна модифікувати залежно до потреб користувача та налаштувани на виконання необхідного йому виду аналіза.

На теперішній час у системі передбачено використання методів для проведення фінансового аналізу (коливання курсу валют).

У цій програмі було використано такі види аналізу:

- Спектральний аналіз Фур’є.
- Вейвлет-перетворення.
- Кальманівська фільтрація.

Основна ідея спектрального аналізу Фур’є зводиться до того, щоб розбити дані, що аналізуються, на синусоїди з різними величинами (довжиною) циклів. Кожний цикл – це частина фундаментального або загального циклу.

Кожна точка часової шкали, має свої унікальні параметри:

- Амплітуду (максимальне значення).
- Частоту (норму вібрації).
- Фазу.

Дані, що аналізуються, розділяють на задану кількість синусоїд, кожна з яких має власну амплітуду, фазу та частоту. Таким чином перетворені дані при спектральному аналізі Фур'є, виражаються амплітудою кожної з синусоїд на відміну від частоти синусоїди. Метод Фур'є, на жаль, має кілька недоліків і може звести всі зусилля нанівець. Основна проблема полягає у тому, що фінансовий ринок не постійний і на ньому є дуже багато шумів. Хаотичність і мінливість дають сильні коливання спектру, приховуючи основні цикли. Шуми на ринку формують помилкові западини і піки. Ще однією проблемою є викривлення аналізу Фур'є при різних значеннях початку і кінця досліджуваного періоду. Це трапляється через те, що спектральний аналіз наближає періодичне розширення даних, і в разі різних початкових і кінцевих котирувань утворюється неоднорідність у реалтайм даних, що надходять. Саме тому рекомендується використовувати довгі інтервали отримуваних даних, які містять як мінімум 64 спостереження на день. Частково проблеми вирішуються різними фільтрами, наприклад, підбором ковзних середніх для пошуку частотних піків спектра.

Вейвлет-перетворення – перетворення, схоже на перетворення Фур'є (або набагато більше на віконне перетворення Фур'є) із зовсім іншою оцінною функцією. Основна відмінність полягає у такому: перетворення Фур'є розкладає дані на складові у вигляді синусів і косинусів, тобто функцій, локалізованих у Фур'є-просторі; напроти, вейвлет-перетворення використовує функції, локалізовані як у реальному, так і в у Фур'є-просторі. Вейвлет-перетворення насправді є нескінченною множиною різних перетворень залежно від оцінної функції, використаної для його розрахунків. У роботі аналітичного модуля використані два види перетворення, що мають такі властивості:

- Дискретне вейвлет-перетворення повертає вектор даних тієї самої довжини, що й вхідний. Звичайно, навіть у цьому векторі багато даних майже дорівнюють нулю. Це відповідає факту, що він розкладається на набір вейвлетів (функцій), які ортогональні до їхнього паралельного переносу та масштабуванню. Отже, розкладаємо подібні дані на ту саму або меншу кількість коефіцієнтів вейвлет-спектра, що й кількість точок даних. Подібний вейвлет-спектр доволі зручний, оскільки не одержуємо надлишкової інформації.
- Безперервне вейвлет-перетворення, навпаки, повертає масив на один вимір більше від вхідних даних. Для одномірних даних одержуємо зображення площини. Можна легко простежити зміну частот даних протягом тривалості та порівнювати цей спектр зі спектрами інших даних. Оскільки тут використовується неортогональний набір вейвлетів, дані високо корельовані та мають велику надмірність. Це допомагає бачити результат у ближчому до людського сприйняття вигляді.

Калмановська фільтрація (фільтр Калмана) – ефективний (що має спосіб гарантовано досягати результату за кінцеве число дій) рекурсивний фільтр, що оцінює вектор стану динамічної системи (у цьому випадку стан фінансового ринку), використовуючи ряд неповних та зашумлених вимірювань. Фільтр Калмана призначений для рекурсивного дооцінювання вектора стану апріорно відомою динамічної системи. Тобто для розрахунку поточного стану системи необхідно знати поточний вимір, а також попередній стан самого фільтра. Таким чином фільтр Калмана, як і безліч інших рекурсивних фільтрів, реалізований в тимчасовому поданні, а не в частотному. Для обчислення оцінки стану системи на поточний такт роботи йому необхідні оцінка стану (у вигляді оцінки стану системи і оцінки похибки визначення цього стану) на попередньому такті роботи та вимірювання на поточному такті. Ця властивість відрізняє його від пакетних фільтрів, що вимагають у поточний такт роботи знання історії вимірювань та/або оцінок.

Результати аналізу, виконані аналітичним модулем, надаються у вигляді графічних залежностей, зручних для сприйняття людиною.

Реалізація системи

Запропанована система може зчитувати з сайтів різну табличну інформацію, яка буде передана для аналізу. Здебільшого це різні економічні показники, які змінюються доволі часто. Програма сканує вказану сторінку і кожного разу проводить аналіз, враховуючи зміни. Продемонструємо роботу програми на сайті, який містить показники змін курсу іноземної валюти. Після запуску програми відкривається вікно з трьома вкладками, на яких розміщуються результати аналізу (рис. 4).

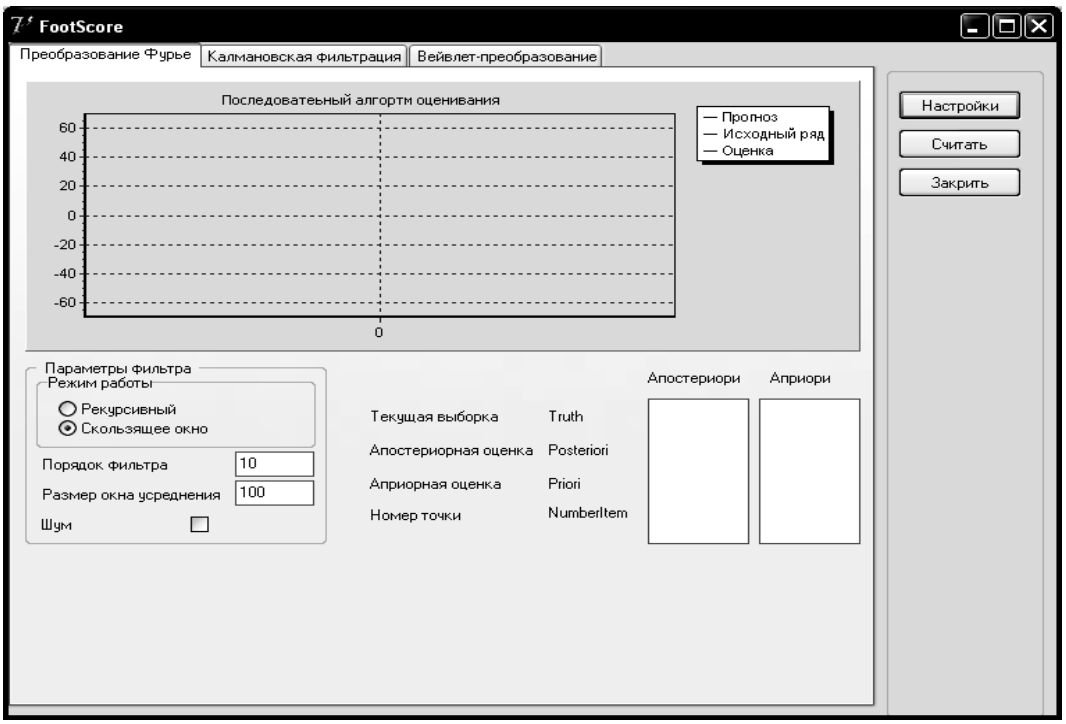

Рис. 4. Інтерфейс програми

Для початку роботи необхідно налаштувати зчитування даних. Для цього натиснемо кнопку «Настройки». Відріється однойменне вікно.

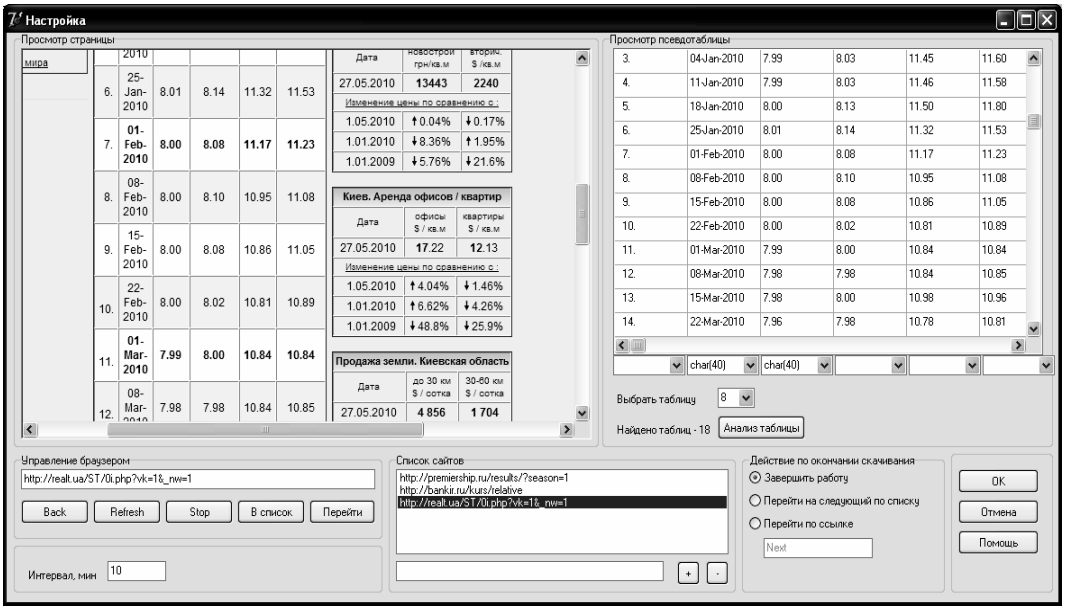

Рис. 5. Діалогове вікно «Настройка»

У цьому вікні можна вказати адреси сайтів, з яких потрібно збирати інформацію, переглянути вказані сайти та задати дії, які повинна виконати система після завершення зчитування сторінки: закінчити роботу, перейти на інший сайт зі списку або перейти за посиланням вказаним у полі «Список сайтов». Також у цьому вікні можна налаштувати інтервал часу, після завершення якого програма самостійно виконує сканування сайта та зчитування з нього інформації.

Після того як сайт завантажився в переглядачі, в полі «Выбрать таблицу» з’явиться список доступних таблиць. Далі необхідно вибрати потрібну таблицю і натиснути кнопку «Анализ таблицы». Потім для кожного потрібного поля необхідно зазначити, до якого типу даних належить

інформація. Надалі за інформацією з цих полів і буде проводитися аналіз. Натискання кнопки «ОК» приводить до закриття вікна налаштувань, дані передаються для аналізу. Результати аналізу виводяться на вкладках, які наведені на рис. 6–8.

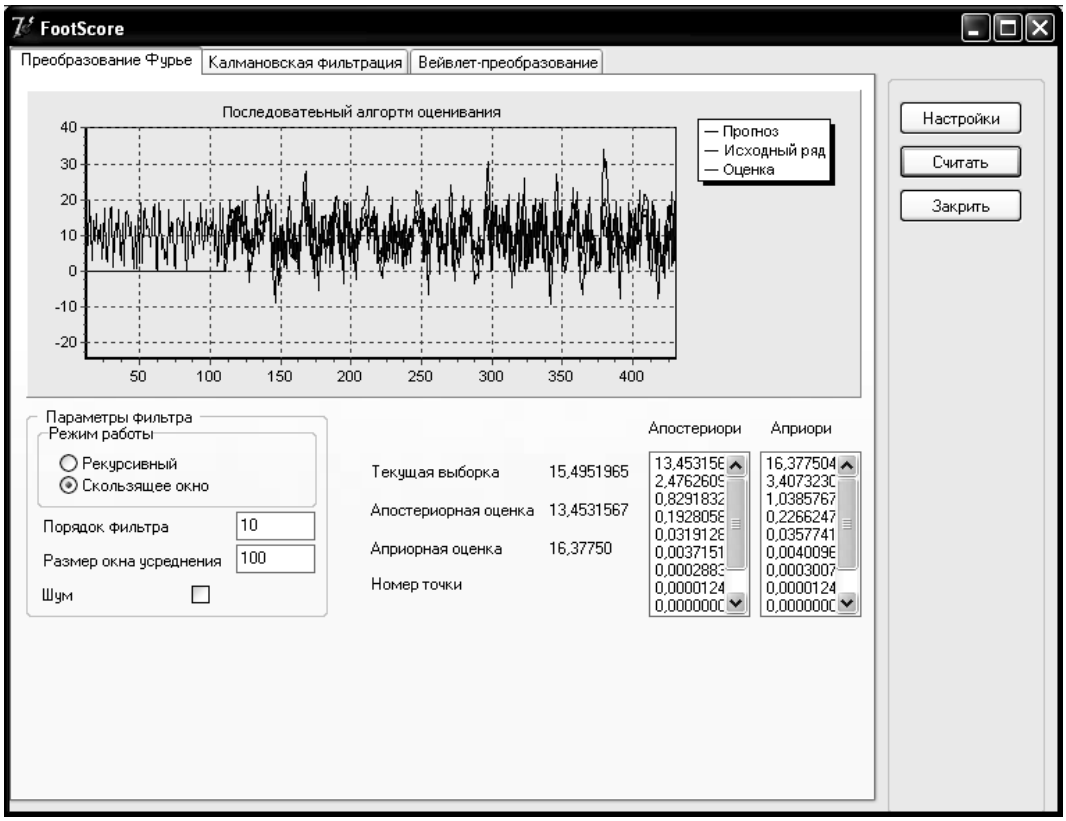

Рис. 6. Вигляд вкладки "Преобразование Фурье"

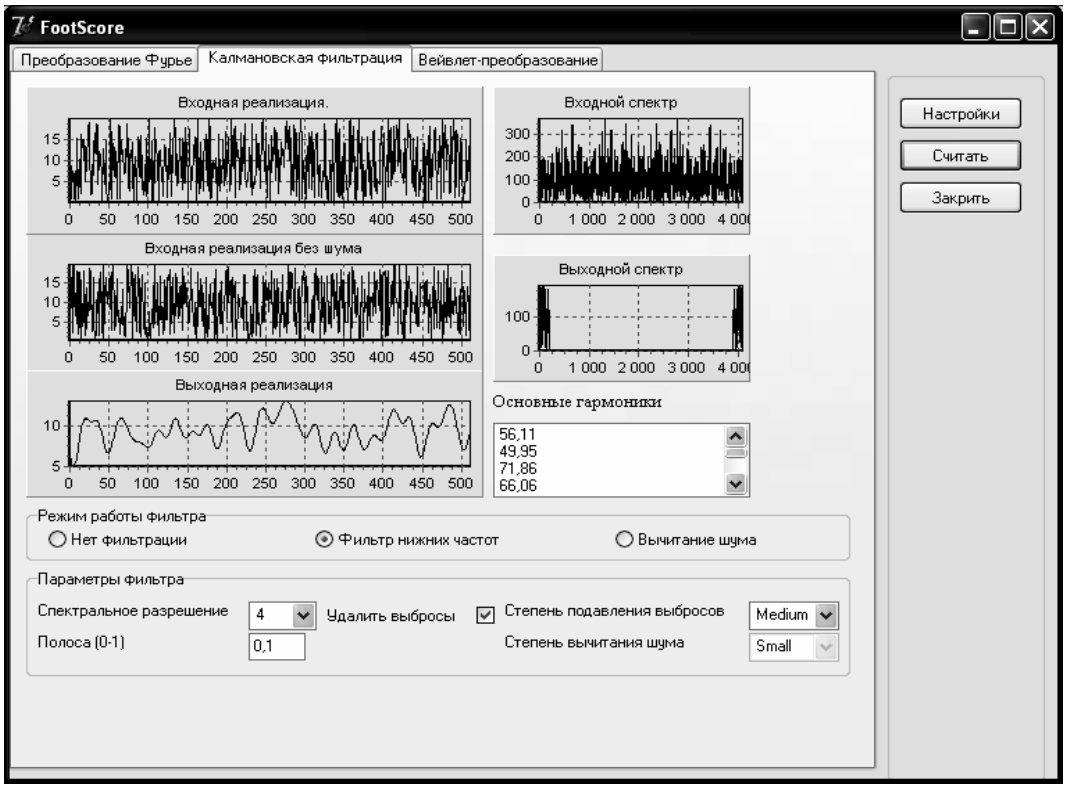

Рис. 7. Вигляд вкладки "Калмановская фильтрация"

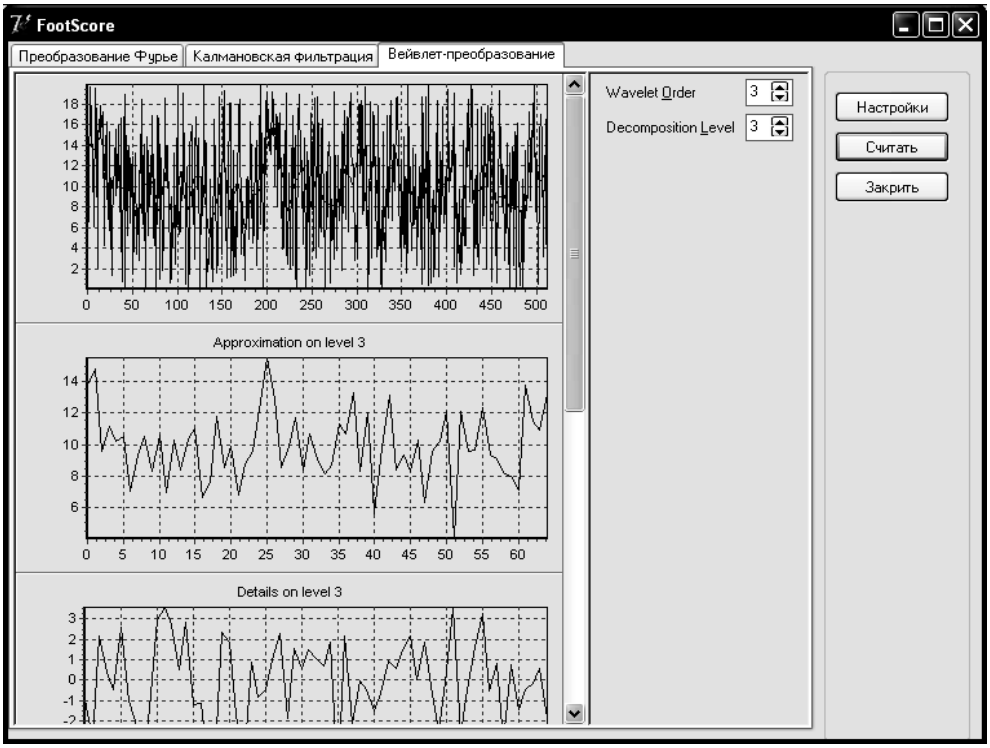

Рис. 8. Вигляд вкладки «Вейвлет-преобразование»

Система заносить до файла всі запити роботи з базами даних, створюючи скрипт, який можна використовувати для експортування даних в іншу систему або до іншої бази даних.

Висновки

У роботі був виконаний аналіз структури web-сторінок, визначені особливості структури HTML таблиць, запропоновано універсальний метод збирання інформації та розміщення у базу даних табличної інформації з web-сторінок. Розроблена система, яка надає можливість отримати доступ до необхідної інформації з будь-якого сайта. Ця система містить вбудований парсер, який створює чітку ієрархію елементів, аналізує дерево елементів та вишукує необхідну інформацію, трансформує її в допустимий для зберігання в базі даних вигляд та заповнює нею таблиці бази даних. Програма працює із усіма сайтами, незалежно від їх структури та має можливість заносити інформацію в бази даних з будь-якими таблицями або створювати нові, залежно від потреб аналізу. В системі також передбачено виконання аналізу інформації, отриманої з мережі Інтернет.

Запропонована система має переваги над іншими системами збору даних, а саме:

- Відстеження змін інформації в джерелах інформації, збереження історії змін та подальший аналіз.
- Можливість інтеграції в корпоративну інформаційну систему.
- Сферою застосування цієї системи можуть стати:
- Моніторинг товарів і послуг.
- Дослідження ринків.

1. Change detection in hierarchically structured information / Rajaraman A., GarciaMolina H., Widom J. // Proc. of the ACM SIGMOD Int. Conf. on Management of Data. – Montreal, Quebec, 1996. – V. 25. № 2. – P. 493–504. 2. Навarро Э. HTML Учебный курс. [Пер. с англ. И. Синицин] / Навarро Э. – СПб.: Питер, 2001. – 336 с. 3. Фаронов Валерий. Delphi 2005. Разработка приложений для баз данных и Интернета. / В.Фаронов. – СПб.: Питер, 2006. – 603 с. 4. Загоруйко Николай Григорьевич. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. – Новосибирск: Изд-во ин-та математики, 1999. – 340 с. 5. Тюрин Юрий Николаевич. Статистический анализ данных на компьютере / Ю.Н. Тюрин, А.А. Макаров. – М.: Инфра-М, 2003. – 544 с.