УДК 004.414

РОЗРОБКА ІНФОРМАЦІЙНО-СТАТИСТИЧНОЇ СИСТЕМИ НА ПРИКЛАДІ АВТОМОБІЛЬНОГО РИНКУ УКРАЇНИ

Соловйов В.А., Сафонова С.О.

DEVELOPMENT OF THE INFORMATION AND STATISTICAL SYSTEM AT THE CASE OF AUTOMOBILE MARKET OF UKRAINE

Solovjov V., Safonova S.

У статті розглянуто питання автоматизованого пошуку інформації на прикладі автомобільного ринку України. Розглянуті аналоги, які можливо використовувати для рішення проблеми пошуку інформації. Надані рекомендації щодо розробки інформаційно-статистичної системи у вигляді програмного додатку. Розроблено та розглянуто основний алгоритм роботи додатку.

Ключові слова: *інформаційна модель, база даних, пошук інформації, аналіз інформації.*

Вступ. Розробка програмного забезпечення (ПЗ), яке вирішує проблеми пошуку та автоматизованої обробки інформації, нині дуже потрібна й актуальна. Великі обсяги інформації змушують користувачів таких сайтів витрачати все більше і більше часу на її вивчення. Представлене програмне забезпечення дозволяє володіти певною інформацією і не витрачати на це час.

Постановка проблеми. Для моніторингу або пошуку необхідної інформації людина може витратити дуже багато часу. Також необхідна інформація може бути випадково пропущена або втрачена у результаті великого обсягу даних. У зв’язку з цим розробка спрямована на пошук способу швидкого та автоматизованого збору великого обсягу даних за допомогою електронно-обчислювальної машини.

Мета статті. Для полегшення роботи з інформацією проаналізувати предметну область галузі та наявні рішення, розробити алгоритм програмного додатку для вирішення проблеми.

Основний текст. Останні роки в Україні спостерігається сплеск активності створення різних сайтів з продажу одягу, техніки, нерухомості, авто. Використання таких систем змушує програмістів писати все нові й нові системи управління сайтами для того, щоб максимально полегшити їх використання та зручність роботи з великими наборами даних, з якими так чи інакше стикаються інтернет-продавці. Якщо у сайту є список товарів, значить в тому чи іншому вигляді є база даних з інформацією, з якою працює система управління сайту. Недолік будь-якої такої системи управління сайтом це її код. Система управління сайтом автоматично генерує сторінки з товаром згідно з шаблоном. З цього випливає, якщо є шаблон, то є можливість написати «зворотню» програму, яка зможе розібрати цей шаблон на початкові складові та використовувати їх за своїм призначенням. Такі програми називаються парсери даних.

Парсер сайтів - послідовний синтаксичний аналіз інформації, яка розміщена на інтернет-сторінках. Текст інтернет-сторінок – це ієрархічний набір даних, структурований за допомогою людських і комп'ютерних мов. Людською мовою надана інформація, знання, заради яких люди й користуються Інтернетом. Комп'ютерні мови (html, JavaScript, css) визначають як інформація виглядає на моніторі.

Парсинг сайтів є ефективним рішенням для автоматизації збору і зміни інформації.

У порівнянні з людиною комп'ютерна програма-парсер:

- швидко обійде тисячі веб-сторінок;
- акуратно відокремить технічну інформацію від «людської»;
- безпомилково відбере потрібне і відкине зайве;
- ефективно упакує кінцеві дані в необхідному вигляді.

Результат (будь то база даних або електронні таблиці), звичайно ж, потребує подальшої обробки. Втім, подальші маніпуляції із зібраною інформацією вже до теми парсинга не належать.

Навіщо потрібен парсинг?

Таблиця

Порівняльний аналіз додатків-аналогів

Назва додатку	Переваги	Недоліки	Ціна	Моя оцінка (0-5)	Примітка
Datacol	Широкі можливості налаштування, помічник складання рядків з інформацією, можливість автоматизації постинга даних на свій сайт, розширення функціональності за допомогою плагінів	Дуже заплутаний інтерфейс, розібратися в програмі дуже складно, отримання даних методом POST	$28 за три місяці використання	3	Datacol вміє отримувати сторінки тільки методом GET. На деяких сайтах при посиланні форми використовується метод POST і в цьому випадку не можливо скористатися цією програмою
X-Parser Light	Зручний графічний інтерфейс, швидка праця, інтелектуальний алгоритм обробки тексту	Орієнтовність на пошукові системи (Google,Yahoo) та сайти новин, ціна	$45	4	Спрямована фактично на «ключові запити» та сайти з новинами. Trial 7 діб
Content Downloader X1	Зручний інтерфейс, вміє парсити не тільки текст, але й зображення або файли, виготовлена в Україні, підтримка авторизації на сайтах	Великий та складний комбайн для парсингу, швидкість роботи	$20 рік	5	Багато зайвого для рішення поставленої задачі

Створюючи веб-сайт, його власник неминуче стикається з проблемою - де брати контент? Оптимальний варіант: знайти інформацію там де її дуже багато - в Інтернеті. Але при цьому доводиться вирішувати такі завдання:

- Великі обсяги. В епоху бурхливого зростання Мережі та жорстокої конкуренції вже всім ясно, що успішний веб-проект немислимий без розміщення великої кількості інформації на сайті. Сучасні темпи життя призводять до того, що контенту має бути не просто багато, а дуже багато, в кількостях, які набагато перевищують межі, можливі при ручному заповненні.
- Часте оновлення. Обслуговування величезного потоку динамічно мінливої інформації несила забезпечити одна людина або навіть злагоджена команда операторів. Часом інформація змінюється щохвилини й в ручному режимі оновлювати її навряд чи доцільно.

У нашому випадку, інформацію яку оброблює парсер потрібно записати у базу даних та повідомити про появу нового запису. Повідомляти про новий запис можливо різними шляхами, але ми будемо це робити за допомогою email повідомлення.

Будемо створювати парсер з функцією повідомлення про появи нових записів. Нижче наведено приклад розробленого парсеру з можливістю інформування щодо отриманої інформації на прикладі автомобільного ринку.

Програма являє собою парсер даних з автомобільного сайту rst.ua. Початкове завдання парсера було в пошуку та повідомленні мене поштою про розміщені авто в моєму місті та містах поблизу. Завдання на момент написання програми - купити авто максимально дешево і дізнатися про це одним з перших. Це і є ідея додатку. Особливістю програми є можливість працювати на будь-якій системі, де є інтерпретатор Python і вихід в інтернет. Модуль з графічним інтерфейсом має обмежену можливість у роботі тільки з тими системами, де підтримується графічний інтерфейс і можливості управління. Програму не складно змінити для роботи з нерухомістю або з іншими майданчиками, тому що вони мають схожий принцип роботи управління сайтами. Для правильної й повної функціональності програма повинна працювати цілодобово.

Нижче представлено порівняльний аналіз додатків-аналогів.

Постановка задачі

Проаналізувавши програми аналоги, було прийнято рішення написати свій варіант тому що:

1. Інтерфейс деяких аналогічних додатків переповнений.
2. Всі додатки працюють тільки в операційній системі Windows.
3. Shareware.
4. Деякі із додатків не використовують класичні бази даних, а зберігають отриману інформацію у своїх форматах.
5. Інтерфейс та ядро додатку не відокремлено.
6. Не мають можливості працювати в термінальному режимі без графічного інтерфейсу.

Тому було прийнято рішення створити власне програмне забезпечення, яке має відповідати таким вимогам:

1. У ПЗ повинен бути простий інтерфейс (для конфігуратора).
2. Ядро та інтерфейс повинні бути відокремленими.
3. ПЗ повинно мати можливість працювати у термінальному режимі.

4. ПЗ повинно працювати на платформі Linux / Windows / MacOs..

5. ПЗ повинно вміти завантажувати html сторінки, а так само аналізувати дані та вибирати інформацію, яка необхідна для подальшої роботи.

6. ПЗ повинно зберігати інформацію в класичну базу даних і обробляти її за певними алгоритмами.

7. ПЗ повинно мати можливість відправляти повідомлення на електронну поштову скриньку відразу після поновлення інформації.

8. ПЗ повинно мати можливість налаштування електронної поштової скриньки, з якої буде відправлятися інформація.

9. Додаток повинен мати можливість вибирати, яку саме інформацію надсилати на електронну поштову скриньку.

10. Додаток повинен мати інтерфейс для перевірки інформації за певними даними. У нашому випадку це номер телефону продавця, після введення якого ми отримаємо інформацію.

11. ПЗ повинно мати простий і зрозумілий інтерфейс додавання сторінок, які будуть оброблятися.

12. ПЗ повинно правильно реагувати на позаштатні ситуації, наприклад: відсутність інтернету, втрата файлу бази даних та інше.

Для реалізації даного програмного забезпечення можна вибрати такі мови програмування, як php, python, C #, C ++. Ці мови програмування досить різні. Php - це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP сконструйований спеціально для ведення Web-розробок і її код може впроваджуватися безпосередньо в HTML. Python - це одна з найбільш популярних сучасних мов програмування. Вона придатна для вирішення різноманітних завдань і пропонує ті ж можливості, що й інші мови програмування: динамічність, підтримку ООП і крос-платформеність. C# - мова програмування, призначена для розробки найрізноманітніших додатків, призначених для виконання в середовищі .NET Framework. C # проста та об'єктно-орієнтована. C ++ - компільована строго типізована мова програмування загального призначення. Підтримує різні парадигми програмування: процедурну, узагальнену, функціональну; найбільшу увагу приділено підтримці об'єктно-орієнтованого програмування.

Графічний інтерфейс залежить від реалізації. Якщо писати програму мовою програмування php, то інтерфейс буде повністю написаний на html + css. Якщо мовою програмування python, C ++ або C #, то однозначно вибором буде Qt. Розглядати інші побудови графічного інтерфейсу немає сенсу.

Для завантаження html сторінок можна використовувати такі засоби як Selenium, однак, на мій погляд, це ускладнить програму. В разі використання мов C++, C# краще використовувати бібліотеки curl. Php має власні вбудовані засоби для обробки та завантаження html сторінок. У разі мови програмування python можна використовувати бібліотеки urllib або request. Для синтаксичного розбору html коду існує ряд різних бібліотек. Для C++, C# - це htmlcxx, для php - це Simple HTML DOM, в python використовуються такі бібліотеки як lxml, BeautifulSoup, Grab. Для того, щоб повідомляти клієнта про нову інформацію, найпростішим і найзручнішим є формування та посилання електронного листа. Найпростіша реалізація такого завдання за допомогою мови програмування php - така можливість реалізована на рівні стандартної бібліотеки mail (). У мові програмування C++ - це бібліотека SMTP-Client (досить складно і невдало реалізована). В мові програмування C# є бібліотека mail send. Реалізована вона набагато краще, ніж в C++. Мова програмування python має бібліотеку smtplib, яка дозволяє надати листу практично будь-який вид. Регулярні вирази. Регулярні вирази - це широковживаний спосіб опису шаблонів для пошуку тексту. Спеціальні метасимволи дозволяють визначати, наприклад, що шукається підстрока на початку вхідного рядка або певне число повторень підрядка. У мові програмування php є вбудована бібліотека для роботи з регулярними виразами. C++ має цілу низку бібліотек для цих цілей: boost, cpp_regex, poco, pcre. C# має у своєму арсеналі вбудовану бібліотеку Regular Expressions. Мова програмування python має бібліотеку - re. База даних. База даних має дуже вагоме значення в роботі програмного забезпечення. Вся інформація, яка потрібна для роботи програмного забезпечення, зберігається в базі даних. Вибір був між mysql і sqlite3. Будь-яка з запропонованих мов програмування дозволяє реалізувати роботу з цими видами баз даних без особливих проблем. Таким чином, визначено приблизний набір бібліотек і мов програмування, які дозволять написати ПЗ найбільш правильно і зручно для програміста і клієнта.

Алгоритм роботи додатку

В програмі використовуються 2 основних алгоритми:

1. алгоритм завантаження сторінок сайту rst.ua з наступною обробкою;
2. алгоритми завантаження сторінок автомобілів з подальшим розбором інформації.

Додатковий модуль, відповідальний за показ статистичної інформації, теж містить в собі один алгоритм.Зауважую, що наведені нижче назви бібліотек належать до мови програмування Python. Перший алгоритм завантаження сторінок сайту rst.ua з наступною обробкою:

1. За допомогою вбудованих засобів бази даних sqlite3 зчитується інформація про веб посилання, назву міста, тип оголошення, посилання повідомлення, кількість автівок, знайдених раніше, та інша інформація.

2. Далі бібліотека urllib переходить за веб посиланням, отриманим в пункті 1, і завантажує html-сторінку.

3. Завантажена html-сторінка передається далі й за допомогою бібліотеки beautifulsoup розбирається на компоненти, відокремлюючи при цьому текст, посилання, зображення та іншу інформацію. Для вирішення постановленої задачі бібліотека beautifulsoup на виході дає таку інформацію як: марка автомобіля, його вартість, час розміщення автомобіля, рік випуску, місто розміщення автомобіля, посилання на більш детальний опис.

4. Отримана в пункті 3 інформація за допомогою бібліотеки sqlite3 записується в базу даних та містить у собі поля: назва авто, ціна, рік випуску, час розміщення, дата розміщення, місто та посилання на сторінку з інформацією.

5. Після запису інформації в базу даних програма переходить до завантаження 2 і 3 html-сторінки з автомобілями. Таким чином дії повторюються з пункту 2.

6. Якщо інформація отримана в пункті 1 говорить про те, що місто є великим за населенням, тоді завантажується 4 і 5 сторінки з автомобілями в даному місті.

7. Алгоритм звертається до отриманої інформації в розділі 1, а точніше інформації про кількість машин в місті перед початком обробки сторінок, і порівнює її з кількістю машин, яка отримана після поточної обробки. Якщо кількість знайдених машин відрізняється від кількості машин записаних в базі даних по оброблюваному місту, то згідно з отриманою інформацією пункту 1 відправляється або не відправляється email повідомлення про нове авто. У випадку, якщо інформація не відправляється, вона просто записується у базу даних. Вигляд повідомлення представлено на рисунку 1.

8. Цикл починається знову до тих пір, поки не закінчиться список веб посилань для обробки.

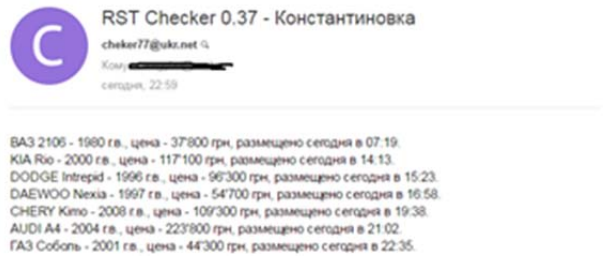

Рис.1. Зовнішній вигляд повідомлення

Для коректної роботи програми алгоритм необхідно запускати кожні 10-15 хвилин. Таким чином, можливо отримувати актуальну інформацію вчасно, до того ж знижується шанс втратити потрібний автомобіль. Робота другого алгоритму неможлива без відпрацювання першого алгоритму, оскільки другий алгоритм використовує інформацію отриману за допомогою першого алгоритму. Другий алгоритм завантаження сторінок автомобілів з подальшим розбором інформації:

1. За допомогою вбудованих засобів бази даних sqlite3 зчитується інформація про веб посилання знайдених у першому алгоритмі автівок.

2. За допомогою бібліотеки urllib завантажується html-сторінка кожного автомобіля для подальшого синтаксичного розбору.

3. За допомогою бібліотеки beautifulsoup отримана інформація аналізується й обробляється. На виході отримуємо таку інформацію як: модель автомобіля, ціна, рік випуску, пробіг, тип двигуна, тип коробки передач, ім'я власника автомобіля, його номер телефону, а також його опис.

4. За допомогою бібліотеки sqlite3 отримана інформація (а саме тип авто, ціна, рік випуску, двигун, тип трансмісії, кілометраж, а також ім'я та телефон продавця) записується до бази даних.

Для найбільш повної роботи цього алгоритму рекомендується запускати його вночі, тому що вдень перший алгоритм збирає багато інформації, яку краще обробити вночі, коли люди не продають автомобілі.

Останній алгоритм виводить всю статистичну інформацію користувачеві. Третій алгоритм на відміну від перших двох дуже тісно пов'язаний з графічним інтерфейсом (рисунок 2).

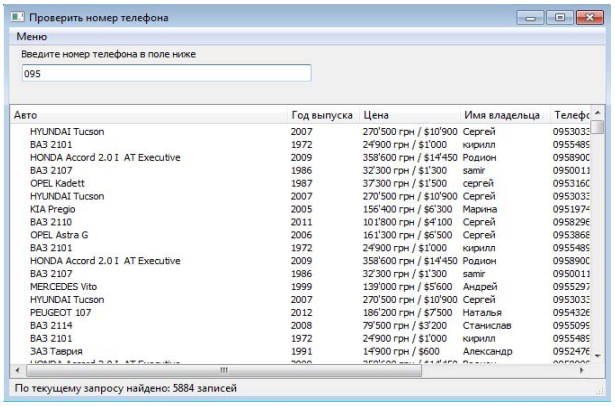

Рис. 2. Перевірка номерів

Опис алгоритму:

1. Після запуску програми перевірки номера телефону по базі даних алгоритм очікує введення номера телефону у відповідному полі.

2. Як тільки вводиться хоча б одна цифра, алгоритм відразу звертається до бази даних і шукає всі номери телефонів, які починаються або містять в собі введену цифру.

Алгоритм виводить інформацію за допомогою графічного інтерфейсу користувача (рисунок 2). Після розробки алгоритмів роботи програми можна приступити безпосередньо до реалізації кінцевого продукту.

Висновки. У статті були розглянуті питання пошуку, збору та аналізу інформації за допомогою різних програмних додатків та алгоритмів. Було прийнято рішення створити більш спеціалізований

додаток для пошуку та роботи з інформацією на прикладі автомобільного ринку України на основі описаних алгоритмів. Розроблений алгоритм та поняття проблематики є основою для створення ефективного програмного засобу.

Розробивши додаток можливо вирішити такі проблеми як:

1. Збереження часу людини
2. Повідомлення про нове оголошення
3. Швидкий та автоматизований пошук інформації
4. Ведення історії автівок, які продавались
5. Ведення історії продавців за номером телефону
6. Зниження до 0% фактору людської помилки

Література

1 . Python. Detailed Reference, 4th Edition, David M. Beezli, 2012 ISBN: 978-5-93286-157-8., 864 с.
2. Mark Lutz - "Study Python" 4th edition 2010, Ill. ISBN: 978-5-93286-159-2
3. Когаловский, М. Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002. 800 с. ISBN 5-279-02276-4.
4. Korneev V.V. Databases. Intellectual processing of information / VV Korneev, AF Gareev, SV Vasyutin, VV Reich - M .: Publisher SV Molchacheva, Publishing House, 2001.

References

1 . Python. Detailed Reference, 4th Edition, David M. Beezli, 2012 ISBN: 978-5-93286-157-8., 864 p.
2. Mark Lutz - "Study Python" 4th edition 2010, Ill. ISBN: 978-5-93286-159-2
3. Kongalovsky M.R. Encyclopedia of databases technologies. - Moscow: Finance and Statistics, 2002. - 800 s .: ill. ISBN 5-279-02276-4.
4. Korneev V.V. Databases. Intellectual processing of information / VV Korneev, AF Gareev, SV Vasyutin, VV Reich - M .: Publisher SV Molchacheva, Publishing House, 2001

Соловьев В.А., Сафонова С.О. Разработка информационно-статистической системы на примере автомобильного ринка Украины

В статье рассмотрены вопросы автоматизированного поиска информации на примере автомобильного рынка Украины. Рассмотрены аналоги, которые можно использовать для решения проблемы поиска информации. Предоставлены рекомендации по разработке информационно-статистической системы в виде программного приложения. Разработан и рассмотрен основной алгоритм работы приложения.

Ключевые слова: *информационная модель, база данных, поиск информации, анализ информации.*

Solovjov V., Safonova S. Development of the information and statistical system at the case of automobile market of Ukraine

The article discusses the issues of automated information search on the example of the automotive market of Ukraine. Considered analogues that can be used to solve the problem of information retrieval. Provided recommendations for the development of information and statistical system in the form of software applications. Developed and reviewed the basic algorithm of the application.

Keywords: *information model, database, information search, information analysis.*

Сафонова Світлана Олександрівна – к.т.н., доцент, доцент кафедри комп'ютерних наук та інженерії, Східноукраїнський національний університет ім. В. Даля. safonovasa@ukr.net

Соловйов Владислав Андрійович – студент групи КІ-17зм; Східноукраїнський національний університет ім. В. Даля. msienery@gmail.com

Рецензент: д.т.н., проф. ***Чернецька-Білецька Н.Б.***

Стаття подана 30.11.2018