

Л.О. Березко, Б.І. Рутило
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

РЕАЛІЗАЦІЯ КЕРУВАННЯ БАГАТОСЕРВЕРНОЮ СИСТЕМОЮ

© Березко Л.О., Рутило Б.І., 2013

Розроблено систему, за допомогою якої здійснюється обробка даних з багатьох серверів, кожен з яких надає певний сервіс. У результаті трансформації та змішування даних і представлення їх в іншому форматі отримують повністю новий сервіс, мета якого – вдосконалити роботу з соціальними мережами. Система може легко масштабуватись завдяки підходу до розробки, що ґрунтується на поєднанні мало-пов’язаних модулів.

Ключові слова: веб-сервери, веб-сервіси.

System which is developed, provides possibilities for process data from many servers, each one gives his service. In results of transformation and mix of data, and representation they in new format, system gives us new service. The main purpose of that system is optimization work process in social networks. The system cans easy growth up through using programming methodology, which based on weakly bound modules.

Key words: web-servers, web-services.

Вступ

Невеликий (за історичними мірками) термін існування “світової павутини” WWW (World Wide Web, або просто Web) показав її ефективність та популярність у великої кількості користувачів. Це стало стимулом для розвитку Web-орієнтованих концепцій і технологій, які розширюють можливості користувачів. Масове впровадження і використання цих технологій викликало якісні зміни, які можна оцінювати як появу нових “версій” Web, наприклад, Web 2.0.

Стан задачі

Нові компоненти Web, з новими можливостями і застосуваннями надають Web нової якості. До них належать:

1. Web-сервіси.

Web-сервіси (Web-служби) – це мережеві додатки, доступні за протоколом HTTP, які використовують формати даних (RPC, SOA і подібні), що ґрунтуються на XML.

2. Web-синдикація.

Одночасне поширення інформації, зокрема аудіо та відео, на різні сторінки або Web-сайти, як правило, з використанням технологій RSS або Atom.

3. Mash-up.

Web mash-up (дослівний переклад – “змішування”) – сервіс, який повністю або частково використовує як джерела інформації інші сервіси, надаючи користувачу нову функціональність. У результаті такий сервіс може ставати також новим джерелом інформації для інших Web mash-up-сервісів. У такий спосіб утворюється мережа залежних один від одного сервісів, інтегрованих один в інший [1].

4. Мітки (теги).

Ключові слова, що описують цей об'єкт або зараховують його до якоїсь категорії, – це свого роду мітки, які присвоюються об'єкту, щоб визначити його місце серед інших об'єктів. З поняттям “міток” тісно пов'язане поняття “фолксономії” – терміна, про який заговорили саме у зв'язку із зростанням сервісів Web 2.0, таких як Flickr, del.icio.us та Wink.

5. Розумні (динамічні) помічники.

Використовуються технології, які дають змогу під час користування сторінкою сайту надавати користувачеві потрібну інформацію, враховуючи його або загальний досвід користування цим сайтом.

6. Соціалізація.

Використання розробок, які дають можливість створювати співтовариство. До поняття соціалізації сайту можна також включити можливість індивідуальних налаштувань сайту і створення персональної зони (особисті файли, зображення, відео, блоги) для користувача, що дає можливість йому відчувати свою унікальність.

З огляду на те, що сьогодні інформаційні технології використовуються фактично в усіх сферах і галузях людської діяльності, виникає проблема організації інформації. Оскільки її обсяг може бути дуже великим, людині важко сприйняти увесь потік у повному обсязі. Розглянемо ситуацію, коли один користувач отримує певні дані з різних серверів. У такому випадку йому потрібно вручну генерувати запити до кожного сервера. Очевидно, що це незручно, вимагає доволі багато часу, тому актуальним є питання вдосконалення цього процесу.

Вдосконалення повинно стосуватись організації спільної роботи усіх серверів, з якими працює користувач. Коли дані з усіх серверів будуть сформовані в інше (зручне) подання, це дасть змогу отримати нову функціональність.

З глобальної точки зору, кожен з серверів надає свій сервіс, поєднання яких уможливорює створити новий інноваційний сервіс. Така багатосерверна система керуватиметься одним центральним сервером, який розподілятиме завдання, керуватиме завантаженістю та агрегуватиме отримані дані в спеціальний (в контексті інноваційного сервісу) формат їх подання.

Найкращим підходом для реалізації цієї пропозиції буде використання сервісу mash-up. Цей підхід об'єднає багато серверів, використовуючи лише їх відкриті інтерфейси.

Постановка завдання

Система, що пропонується, буде корисною насамперед для користувачів, які підтримують зв'язок з різними групами інших користувачів. Як відомо, соціальні мережі широко використовуються для рекламування, розповсюдження інформації, пошуку нових клієнтів. Отже, це може бути корисним як для корпоративних, так і для звичайних користувачів. Підхід, що пропонується, уможливить зекономити час, значно зменшити кількість дій, які виконує користувач, працюючи з соціальною мережею.

Система повинна забезпечувати надання інформації та засоби керування нею у доступному для загального кола користувачів вигляді. Необхідно спроектувати та реалізувати клієнти для мобільних пристроїв. Дані, якими система буде оперувати, повинні бути захищені від несанкціонованого доступу і модифікацій.

Також повинен існувати Web-сервер, який на запит користувача надаватиме йому певну інформацію, або оброблятиме надану користувачем інформацію. Крім того, що Web-сервер взаємодіятиме з користувачем, він повинен також взаємодіяти з іншими серверами, які будуть для нього джерелом отримання інформації, або об'єктом, куди треба інформацію відправляти. На рис. 1 показано структурну схему такої взаємодії.

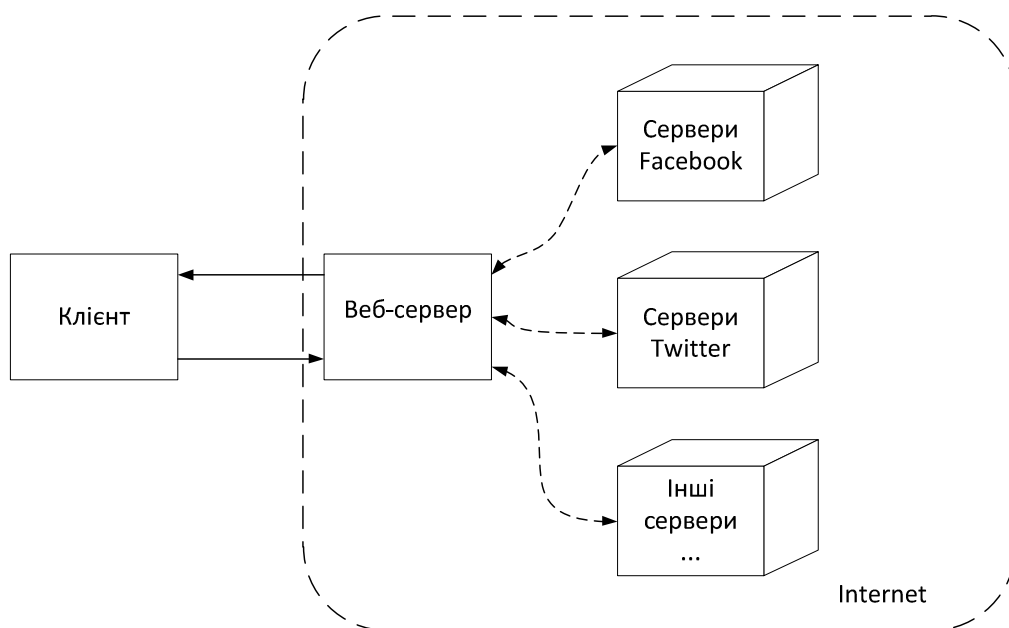


Рис. 1. Узагальнена структурна схема системи

Об'єднання серверів за допомогою “mash-up”

Mash-up (інакше “меш-ап”, від англ. *mash-up* — “змішувати”) — концепція побудови Web-сервісів шляхом змішування функціональності різних програмних інтерфейсів та джерел даних. З технічного погляду Mash-up – це Web-сайт, який об'єднує дані з кількох джерел в одному сайті. Наприклад, використовує картографічні дані Google Maps та додає до них дані про нерухомість з Cragislist, у підсумку створюючи новий Web-сервіс.

Основними інструментами концепції є комбінація, візуалізація, агрегація. Важливо зробити існуючу інформацію кориснішою для особистого чи професійного використання. Mash-up відіграє активну роль у розвитку соціального програмного забезпечення та розвитку Web 2.0.

Вміст, що використовується в mash-up, переважно отримується від третіх осіб через відкритий інтерфейс або API. Mash-up створює нові споживчі сервіси, своєю чергою, створюючи для них безліч нових можливостей. Кінцевим користувачам стає зрозуміліше, як створювати mash-up-сайти, незалежно від рівня їх технічних здатностей, використовуючи API сайтів. Це робить реалізацію mash-up простою і нові споживачі можуть легко створювати mash-up. З усіма доступними API виробники можуть легше і дешевше будувати програми, які повторно використовуються і підтримуються.

Розв'язання задачі

Запропонована система реалізована засобами мови програмування Java, за допомогою поєднання трьох шаблонів проектування програмного забезпечення: Фасад, Модель-вигляд-контролер, Стратегія [2, 3].

Перший шаблон дає змогу спроектувати систему так, що одна система зможе включати в себе багато інших підсистем. Це обов'язковий шаблон під час використання концепції mash-up .

Основним шаблоном проектування для цієї задачі є Модель-вид-контролер (або Модель-вигляд-контролер, англ. *Model-view-controller*, MVC). Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета цього шаблону – уможливити гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонент програми. Крім того, використання цього шаблону у великих

системах приводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

І останній використаний шаблон – це Стратегія. Стратегія (англ. *Strategy* — шаблон проектування, що належить до класу шаблонів поведінки). Відомий ще під назвою “Policy”. Його функція полягає у тому, щоб створити кілька схем поведінки для одного об'єкта та винести його в окремий клас. Шаблон Стратегія дає змогу змінювати вибраний алгоритм незалежно від об'єктив-клієнтів, які його використовують.

На рис. 2 показано структурну схему реалізації системи.

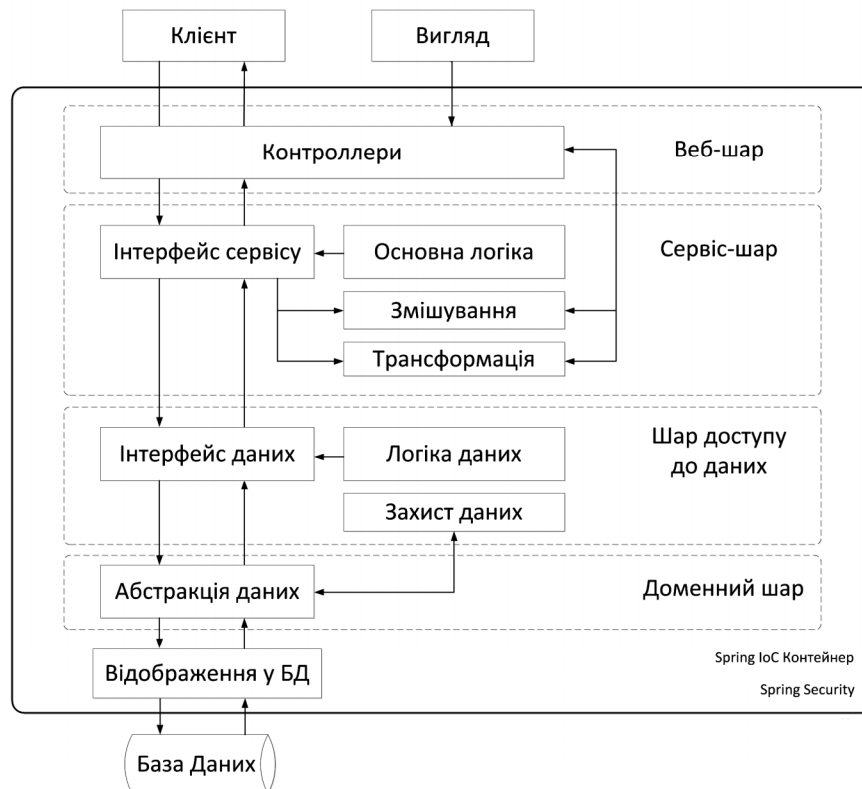


Рис. 2. Структурна схема реалізації системи

Висновки

Запропоновано підходи та розроблено засоби, які дають змогу керувати багатосерверною системою, кожний сервер в якій надає свій сервіс. Запропонована реалізація дає можливість об'єднувати ці сервіси в один, з новими якостями та можливостями, що забезпечує вищу продуктивність роботи з даними та легко масштабується завдяки використанню концепції mash-up.

1. Mash-up. – [Електронний ресурс]. – Режим доступу до статті:<http://uk.wikipedia.org/wiki/Mashup>. 2. Шаблиони проектування програмного забезпечення. – [Електронний ресурс]. – Режим доступу до статті:http://en.wikipedia.org/wiki/Software_design_pattern. 3. Dhrubojyoti Kayal. Pro Java™ EE Spring Patterns: Best Practices and Design Strategies Implementing Java™ EE Patterns with the Spring Framework, 2008.