

ІНТЕЛЕКТУАЛЬНИЙ СЕРВІС ЗАВАНТАЖЕННЯ, КЕРОВАННИЙ МОДЕЛЯМИ

© Буров С.В., 2008

Розглянуто принципи реалізації та структуру інтелектуального сервісу завантаження у комп’ютерній мережі. В основу реалізації сервісу покладено архітектурний підхід Адаптивних об’єктних моделей (АОМ). Запропоновано структуру системи та визначено типи моделей керування.

Current trends in information system architectures promote intellectual and service-oriented approaches. This paper analyses file download service as an example of intellectual, model-driven service implementation. This implementation is based on Adaptive Object Models approach. The structure of system and corresponding models set along with their functions is proposed.

Постановка проблеми та аналіз останніх досліджень

Програми автоматизації завантаження мережевих ресурсів (менеджери завантажень, Downloaders) сьогодні належать до одного з найрозвиненіших та широко використовуваних класів мережевих утиліт. Найпотужніші менеджери завантажень, такі як Reget, Flashget, Download Manager надають можливість багатопотокових завантажень, диспетчерування, пріоритетності, використання макросів та багато інших.

Головною задачею типового менеджера завантажень є для заданої адреси файла-ресурса (URI) завантажити цей файл на локальний комп’ютер.

Для адаптації процесу виконання конкретного завантаження до вимог користувача та до стану сполучення комп’ютерної мережі в менеджерах завантажень є велика кількість налаштувань, що встановлюються користувачем вручну. Ці налаштування належать до таких груп:

- завдань на завантаження (URI завдання, локальна адреса, категорія, графік завантаження);
- стану мережі зв’язку (режим завантаження – обмежений, кооперативний, необмежений, максимальна кількість потоків, кількість повторних спроб при розриві сполучення);
- взаємодії з користувачем (оформлення інтерфейсу, реагування на події, збереження історії, логування повідомлень);
- пошуку;
- окремих протоколів (налаштування параметрів для конкретних протоколів, наприклад, HTTP, FTP);
- встановлення зв’язку з провайдером (параметри додзвону, VPN – сполучення);
- взаємодія з проксі – серверами.

Як правило, зазначені параметри встановлюються користувачем вручну. Вони не змінюються залежно від конкретного завантаження, стану мережі або відгуку конкретного сайту. Функції підтримки анонімних проксі-серверів, пошуку контенту, роботи з дзеркалами розвинені недостатньо.

Сьогодні архітектурні вирішення мережевих інформаційних систем поступово переходять на підтримку сервісного підходу та інтелектуальних семантично-орієнтованих мереж [1,2]. Подальший розвиток менеджерів завантажень у цих умовах полягає у їх трансформації в інтелектуальні сервіси завантажень. Такі сервіси порівняно з існуючими менеджерами завантажень надають такі додаткові послуги та можливості:

- встановлення оптимального графіка завантаження, враховуючи умови провайдера;

- рекомендація або вибір найвигіднішого пакета послуг провайдера з врахуванням історії попередніх завантажень;
- відслідковування та налаштування параметрів завантажень, враховуючи специфіку окремих сайтів – визначення максимальної кількості потоків, швидкості, застосування анонімних проксі-серверів у разі потреби;
- гнучкий пошук дзеркал та перемикання завантажень на інші сервери;
- підтримка завантажень в однорангових мережах, вибір серверів завантажень;
- взаємодія з аналогічними сервісами та утворення груп (community) сервісів з врахуванням інтересів їх власників, сумісне використання скачаних файлів у межах групи;
- пошук контенту у мережі, взаємодія з сервісами пошуку;
- інтелектуальне кешування ресурсів, взаємодія з серверами кешування файлів з метою ефективнішого доступу до ресурсів;
- автоматична авторизація на сайтах, взаємодія з сервісами авторизації;
- врахування досвіду роботи інших сервісів (користувачів) з певними сайтами, що виражається у зміні правил закачування з цих сайтів. Зміни до правил та відповідні програми передаються між учасниками групи користувачів сервісів.

Структура та принципи побудови інтелектуального сервісу завантаження

Розглянемо контекстну діаграму інтелектуального сервісу завантаження. На вхід сервісу надходять запити на завантаження. У результаті роботи сервісу отримують завантажені файли. У процесі роботи сервіс взаємодіє з іншими сервісами та сайтами. Крім того, сервіс отримує керівну інформацію про зміну моделей, що керують його роботою (рис. 1).

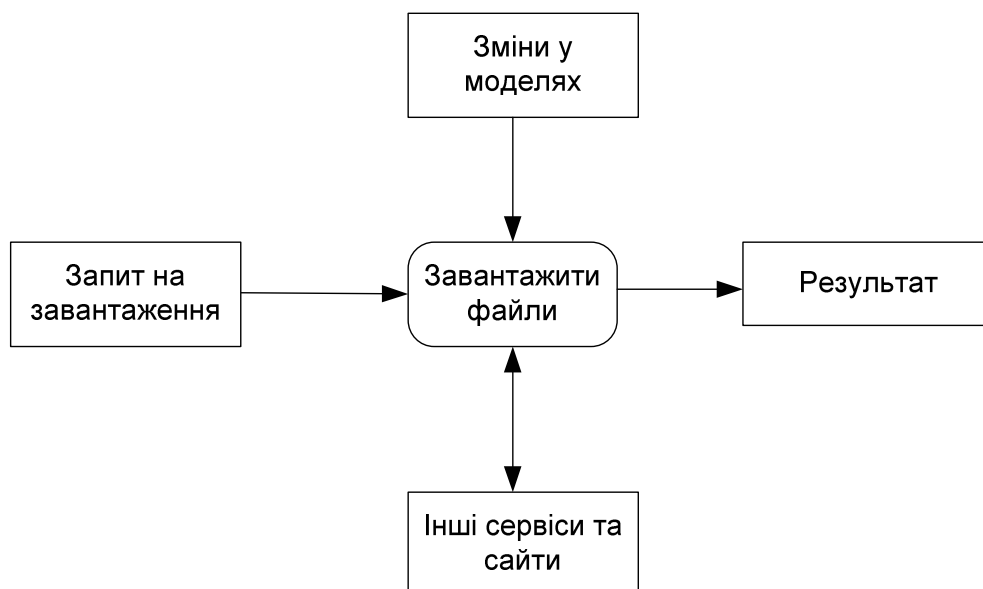


Рис. 1. Контекстна діаграма сервісу завантаження

Розглянемо структуру сервісу детальніше (рис. 2). На вхід сервісу надходять запити на завантаження. Запити можуть відповідати різним типам завантажень. Так, завантаження може бути одноразовим, періодичним (повторюватися з певним часовим інтервалом) або умовним (тобто виконуватися, якщо дійсна певна умова, наприклад, на сайті викладено нову версію документа).

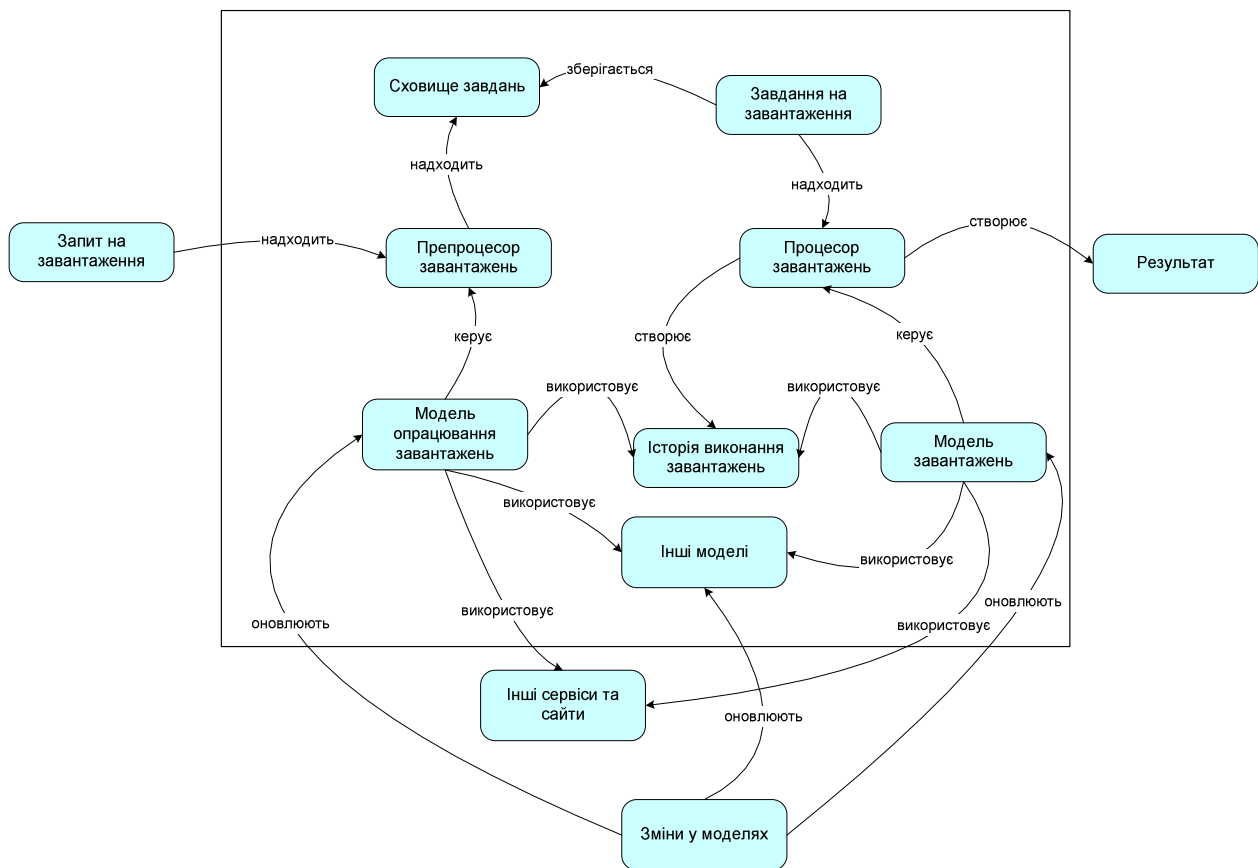


Рис. 2. Структура інтелектуального сервісу завантаження

Запит, що надійшов, опрацьовується препроцесором завантажень. Цей препроцесор безпосередньо не здійснює завантаження, а тільки готує завдання до завантаження. Він, наприклад, планує час проведення завантаження, шукає додаткову інформацію, перевіряє правильність та виконуваність завдання. Опрацьовані завдання зберігають у сховищі завдань на завантаження. За принципом роботи препроцесор завантажень є процесором правил, тобто весь алгоритм його роботи сформульований як набір правил. Правила роботи препроцесора зберігаються в окремій моделі – моделі попереднього опрацювання завантажень. Як кожна інша модель системи, модель попереднього опрацювання завантажень може змінюватися, додаючи системі необхідної гнучкості (система не потребує зміни коду). Під час роботи окремі правила використовують дані з інших моделей, а також з історії попередніх завантажень. Правила можуть стосуватися як окремих завантажень, так і їх груп, створених за певною ознакою (наприклад, всіх завантажень з певного сайта або всіх завантажень певної категорії контенту).

Процесор завантажень безпосередньо виконує завантаження згідно з графіком завантажень та додатковими умовами, сформульованими у правилах моделі завантажень. Аналогічно до препроцесора завантажень процесор завантажень є процесором правил, і весь алгоритм його роботи сформульовано як набір правил моделі завантаження.

Центральною частиною інтелектуального сервісу завантаження є моделі, що керують його роботою. Для реалізації модельно-керованої архітектури сервісу завантаження пропонується використовувати архітектурні принципи Адаптивних об'єктних моделей (AOM – Objective Object Model) [3,4]. AOM має такі відмінності порівняно з традиційним об'єктно-орієнтованим підходом до моделювання[5].

- система розглядається на вищому рівні абстракції. Якщо, наприклад, в традиційному підході використовують клас та суперклас (метаклас), то в AOM, замість них – об'єкт та клас;

- для моделювання властивостей та відношень між класами використовують окремі класи певних типів, так що кожна властивість є об'єктом класу властивості певного типу;

- для моделювання бізнес- правил, складних залежностей між об’єктами використовуються Об’єкти Правил (Rule Objects). Такі об’єкти дають змогу описувати не тільки прості залежності між даними, але специфікувати процеси;

- визначення класів та об’єктів зберігається у базі даних. Об’єкти цього класу є динамічними – вони створюються та отримують значення під час роботи системи.

Динамічний характер об’єктних моделей дає змогу не тільки налаштовувати параметри моделей залежно від ситуації, але й створювати нові об’єкти та моделі, а також шляхом розповсюдження нових моделей по мережі обмінюватися досвідом та гнучко реагувати на зміну поточної ситуації.

Схема класів моделей наведена на рис. 3.

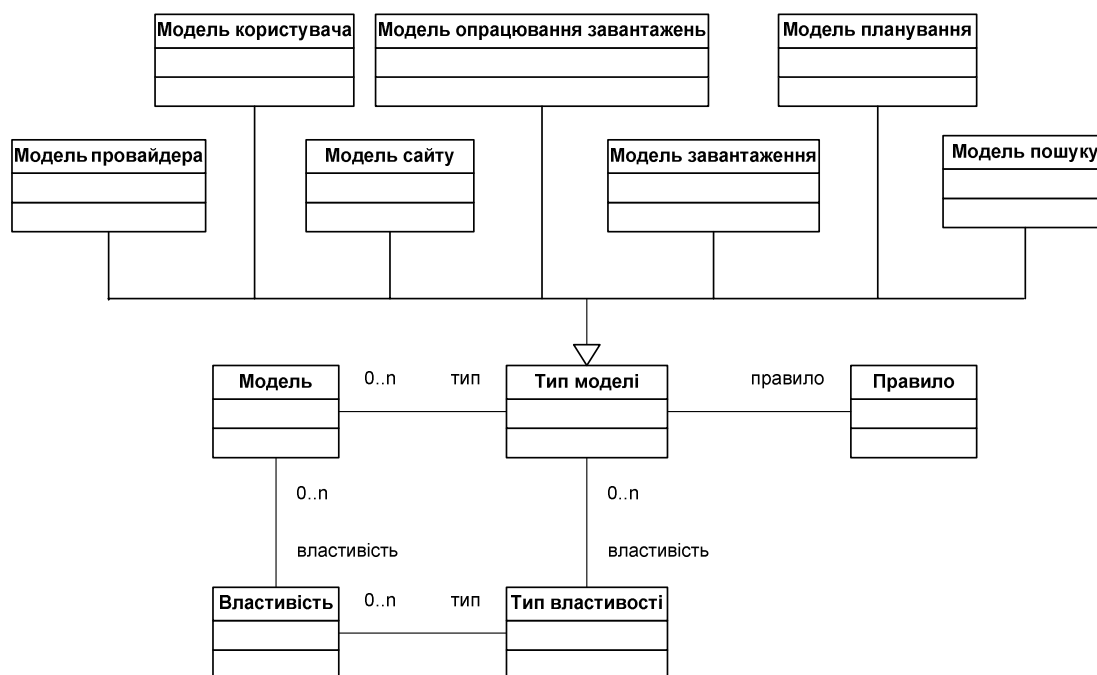


Рис. 3. Схема класів моделей

Інтелектуальний сервіс завантаження використовує такі типи моделей.

- Модель провайдера зв’язку

Модель містить такі дані: назва провайдера, адреса сайту (URL), тип каналу, наявні типи послуг, перелік послуг, на які підписаний абонент, вартість послуг, тип сполучення (dial-up, VPN), інформація для встановлення сполучення (номер телефону додзвону, IP- адреса сервера VPN). Якщо провайдер підтримує довідковий сервіс, то модель провайдера також містить операції звертання до довідкового сервісу провайдера з метою отримання інформації про стан рахунку, типи наданих послуг, тимчасові обмеження на передавання (від’єднання), зміну переліку доступних послуг та ін.

Інформація моделі провайдера зв’язку використовується моделлю планування завантажень, а також для автоматичного встановлення (перевстановлення) сполучення. Сервіс завантаження повинен підтримувати декілька моделей провайдерів, оскільки користувач може користуватися послугами декількох провайдерів.

- Модель користувача

Модель користувача зберігає інформацію про користувача системи. Вона містить, наприклад, такі дані: улюблені сайти користувача, анкетні дані, необхідні для автоматичної реєстрації на сайтах (заповнення форм), підписка на канали новин (RSS), переваги щодо використання сервісів пошуку та ін. До моделі користувача належать такі операції, як аналіз історії завантажень у розрізі

категорій його контенту і пропонування користувачеві новин та сайтів, що належать до категорій, що його цікавлять (побудова профілю користувача).

- **Модель сайту**

Зберігає інформацію та правила роботи з конкретними сайтами. Модель містить такі дані: максимальна кількість потоків скачування, які підтримує сайт, обмеження на підтримку протоколів, необхідність використання анонімних проксі-серверів для доступу до сайту, поточний стан сайту (працює, не працює, знаходиться під DDOS атакою, тимчасово недоступний), наявні дзеркала сайту (URL). До операцій моделі сайту можна віднести автоматичну реєстрацію на сайті, заповнення форм.

- **Модель опрацювання завантажень**

Модель опрацювання завантажень використовується для попереднього опрацювання отриманих запитів на завантаження. Зокрема, цей процесор класифікує завантаження відповідно до певної категорії, планує час проведення завантаження, якщо необхідно, отримує додаткову інформацію з зовнішніх сервісів.

- **Модель завантаження**

Модель завантаження використовується безпосередньо для організації завантаження. Вона містить параметри за замовчуванням, які використовуються, якщо модель сайту не передбачає іншого, кількість повторних спроб, якщо ресурс недоступний, інтервали між спробами; чи треба записувати інформацію про події у log-файл. Модель завантаження визначає також правила поведінки сервісу у разі виникнення певних подій. Наприклад, якщо певна кількість завантажень з різних сайтів підряд були неуспішними, має сенс перевірити стан каналу до провайдера на предмет розриву. Інше правило може перевіряти перед початком завантаження розмір файла та відмовлятися від завантаження, якщо розмір перевищує визначену межу, або провести таке завантаження не дозволяє стан рахунку.

- **Модель планування завантажень**

Модель планування завантажень використовується для побудови графіка завантаження або групи завантажень. При цьому враховуються: категорії завантаження та правила і обмеження, сформульовані відносно певних категорій, ступінь терміновості, наявність інших передавань, тарифи та доступні послуги провайдера. Наприклад, якщо провайдер надає можливість нічних завантажень за пільговим тарифом, метод планування завантажень (який є частиною моделі) запланує нічне завантаження.

- **Модель пошуку контенту**

Модель пошуку контенту визначає параметри, порядок та правила пошуку потрібного контенту. При цьому організується взаємодія з пошуковими серверами та враховується модель користувача та його потреби. В основі моделі пошуку контенту є семантична мережа з ключовими словами (концептами) та відношеннями між ними. Користувач, наприклад, може передбачити пошук та завантаження матеріалів (або тільки анотацій) про задані концепти та пов'язані з ними концепти.

Формальна специфікація інтелектуального сервісу завантаження

У роботі [6] розглянута формальна специфікація інформаційної комп'ютерної мережі, яка використовується для проектування та керування розподіленими інформаційними системами і складається з декількох мережевих специфікацій. Доповнимо цю специфікацію специфікацією мережі сервісів:

$$S = \{NBp, NSe, NPc, NDv\},$$

де NBp – мережа бізнес-процесів, NSe – мережа сервісів, NPc – мережа процесорів, NDv – мережа пристроїв.

Мережа сервісів (Реєстр сервісів, service inventory [1]) – це множина сервісів:

$$NSe = M(Se)$$

Сервіс завантаження DnSe визначається специфікаціями його інтерфейсу та реалізації.

$$DnSe = \{InDnSe, Re DnSe\},$$

де *InDnSe* – інтерфейс сервісу, *Re DnSe* – внутрішня реалізація сервісу

Реалізація запропонованого інтелектуального сервісу завантаження містить такі частини:

$$Re DnSe = \{PpcDnSe, LiDnSe, LiMoDnSe, PcDnSe, HiDnSe\}$$

де *PpcDnSe* – препроцесор опрацювання завантажень, *LiDnSe* – сховище завдань, *LiMoDnSe* – набір моделей, *PcDnSe* – процесор завантажень, *HiDnSe* – історія виконання завантажень.

$$LiMoDnSe = \{PpcMo, PcMo, PiMo, StMo, PlMo, ShMo\}$$

де *PpcMo* – модель опрацювання завантажень, *PcMo* – модель завантаження, *PiMo* – модель провайдера(ів) зв’язку, *StMo* – модель сайту(ів), *PlMo* – модель планування завантажень, *ShMo* – модель пошуку контенту.

Висновки

Запропонована у роботі функціональна структура та принципи роботи інтелектуального сервісу завантаження дають змогу досягти більшого рівня адаптивності до зміни середовища та більшої ефективності у використанні ресурсів, ніж існуючі менеджери завантажень.

1. Thomas Erl. *SOA: principles of service design*, Prentice Hall, 2007. 2. Jorge Cardoso. *Semantic Web services : theory, tools and applications*, Hershey, New York., 2006. 3. Joseph W. Yoder, Ralph Johnson. *The Adaptive Object-Model Architectural Style*. The Refactory, Inc. and Software Architecture Group – Department of Computer Science Univ. Of Illinois at Urbana-Champaign. 4. Ralph Johnson. *Dynamic Object Model*. Department of Computer Science University of Illinois at Urbana-Champaign. 5. Jim Alateras. *The Adaptive Object Model*. O’Reilly XML Blog, August 25, 2005, <http://www.oreillynet.com/xml/blog/index.xml>. 6. Буров Є.В. Система формальних специфікацій для проектування розподілених інформаційних систем // Вісник Держ. ун-ту “Львівська політехніка” “Інформаційні системи та мережі”. – 2000. – № 406.