

ІНТЕРНЕТ: СУЧАСНІ ТЕХНОЛОГІЇ ТА БІЗНЕС-МОДЕЛІ

© Кравець Р.Б., Скицький Т.Р., 2008

Подано короткий огляд Інтернет-технологій та найуспішніших бізнес-моделей, які сьогодні ефективно використовуються у Всесвітній мережі, та розглянуто еволюцію цих моделей у зв'язку із тими можливостями, які надає інтенсивний розвиток програмного забезпечення (новітні Web-технології).

The short review of WWW Development methodologies and the most successful business models, which are effectively utilized in WWW for today is given in the article, and the evolution of these models in connection with those possibilities, providing by intensive development of software is examined (the newest Web-technologies) .

Постановка проблеми

Сучасний потужний технологічний розвиток програмного забезпечення у Всесвітній мережі насамперед сильно вплинув на розвиток та принципові зміни у технологіях провадження бізнесу. Тому у цій статті цьому напрямку діяльності приділяється увага і подаються конкретні приклади. Розглядаються різні аспекти, можливості та впливи, які має або буде мати наступне покоління Web, і здійснено це на основі функціональності і соціалізації, які можна розглядати як певний фундаментальний “навігаційний простір” для Web.

Оскільки еволюція Всесвітньої Павутини та її розвиток за низку років має різноманітні аспекти, зокрема й значний технологічний аспект, то в цій статті розглянемо з множини цих аспектів досить важливі перспективи для індивідуального користувача, суспільства користувачів та з погляду бізнес-користувачів. Зокрема треба звернути увагу і на такий аспект Павутини, як її вплив на індивідуальних користувачів, зокрема в плані взаємодії розміщеної інформації, або існування та розвитку співтовариства користувачів. Важливо підкреслити також, що відбувається розвиток бізнесу, зокрема зростання ефективності партнерських взаємовідносин, що пов'язане з доступом до великої бази різноманітних мережевих даних, використанням ефективного сервісного програмного забезпечення чи так званого соціального програмного забезпечення. Також звертається увага та коротко розглядаються найважливіші бізнес-моделі, які отримали свій розвиток у Мережі або були перенесені в Мережу з фізичної реальності. У цьому контексті варто ці бізнес-моделі конкретизувати та показати їх розвиток в новому середовищі Web 2.0. Також сьогодні спостерігаємо, як технічні можливості сучасної Мережі дають змогу ефективно комбінувати бізнес-моделі, що було неможливим ще декілька років тому.

Аналіз останніх досліджень

Типові структури для розроблення мережевих додатків

Програмне середовище, призначене ефективно допомогти Web-розробникам проектувати і створювати сайти з новими функціональними можливостями, сьогодні розвивається доволі успішно. Стисло розглянемо методи та принципи розроблення, що лежать в основі Web-програмування. Хоча програмне середовище для проектування і створення сайтів є стандартним та використовується кожним, хто має із цим справу, проте у цій галузі існують й певні особливості.

Кількість програмних засобів, що дають змогу ефективно розробляти Інтернет-додатки (Rich Internet Applications, RIA), за останній час значно зросла. Серед найвідоміших прикладних мов програмування для RIA сьогодні Ajax та багато інших, які власне ґрунтуються на цьому

програмному середовищі. Інструментарій цієї групи можна поділити на клієнтські (client-side) та серверні (server-side) додатки, де до перших зазвичай належать JavaScript-файли, які користувач може завантажити з Інтернету і на які посилаються Web-додатки. Серед іншого тут варто зазначити як відносно нескладний XMLHttpRequest (XHR) аж до потужного об'єктно-орієнтованого середовища JavaScript бібліотек, що задовольняє більшість потреб програмування. Server-side-середовище програмування надає Ajax-підтримку різним мовам програмування, зокрема Java, DotNet, C++, PHP, Ruby.

Як приклад client-side-середовища розглянемо Kabuki Ajax Toolkit. Цей пакет пропонує такий сервіс, як оболонка XHR, підтримка подій, мережових комунікацій, SOAP та візуальні інтерфейси на основі «drag and drop». Прикладом client-side середовища може бути Ruby на базі Rails. Зацікавленість до останнього зросла відтоді, як у цьому середовищі стало можливим швидке макетування та позбавлення від багатьох проблем розроблення Web-додатків.

Одна із причин популярності Ajax – його повна інтеграція у сучасні Інтернет-браузери. Немає жодної необхідності інсталиувати додаткове програмне забезпечення, оскільки все, потрібне для розроблення, є в інсталяційних пакетах всіх віртуальних операційних систем. Проте ще дотепер використовуються браузери, які не підтримують JavaScript; зокрема, це текстовий браузер Lynx. Деякі інші браузери мають вимкнений JavaScript або їх захист блокує його. Але треба наголосити, що за статистичними даними понад 94% користувачів мають можливість використовувати JavaScript доволі успішно. До недоліків Ajax належить те, що він не підтримує такого медіа-контенту, як аудіо та відео. Тому додатки, у яких виникає потреба використання аудіо чи відео, не можуть розроблятися на основі використання Ajax і повинні використовувати інші технології для обробки медіа-контенту.

Популярність Ajax настільки велика, що може скластися враження, ніби не існує інших альтернатив. Але це не так. Протягом декількох років використовується Adobe Flash, більш відомий як Macromedia Flash, який дає відмінну підтримку медіа та анімації; Java Web Start дозволяє виконання Java-додатків у браузері. Але останні зазначені засоби мають певний недолік, а саме: вони потребують інсталяції у браузері розроблених додатків для того, щоби їх можна було виконати. Це обмежує не лише їх використання групами користувачів (оскільки такі додатки можуть не підтримуватися деякими операційними системами), але й значно обмежує ефективність використання Web-додатків (неможливість використання гіперпосилань, порушення загальної системи адресації тощо). Ще одним прикладом може бути penLaszlo. Цей пакет був створений на основі використання Open-Source ліцензії. Він цікавий тим, що ефективно підтримує використання додатків, реалізованих у різних платформах. Початки реалізації цього пакета пов'язані із Flash-платформою, а сьогодні він підтримує Ajax та працює з переважною більшістю цільових розробок.

Традиційні технології розроблення програмного забезпечення

Загальна відмінність у технологіях розроблення (наприклад, проектування баз даних, інформаційних систем, моделювання процесів) полягає між *моделлю* (наприклад, модель взаємовідносин об'єктів), такою як уніфікована мова проектування (*Unified Modeling Language, UML*), та *методологією* розробки додатків. Щодо останньої, це загалом відмінність між проектуванням «згори донизу» (тобто від загального рівня і йдучи донизу, наскільки це є необхідним) та проектуванням «знизу догори» (тобто починаючи від компонентів і поступово узагальнюючи в міру потреби). Технології переважно використовують одну, або як у випадку UML – кілька моделей, з метою структурування результатів різних етапів розроблення. Певною мірою технології для конкретної мети визначаються власне їх перевагами (корисністю) для поставленого завдання порівняно з іншими технологіями. Для прикладу, уніфіковане проектування використовує UML, коли об'єднуються декілька підходів (дії, компоненти, класи, процеси). Найдавнішим підходом до розроблення програмного забезпечення є метод *Waterfall*, який був запропонований Ройсом в 1970. Він ґрунтується на проектуванні «згори донизу» і починається з *аналізу вимог*. Цей крок (аналіз вимог) визначає побажання і вимоги клієнта, які мають велику вагу, оскільки окреслюють наступні дії та кроки. Внаслідок того, що відомі конкретні вимоги, подальше

розроблення програмного забезпечення та докладний опис того, що планується реалізувати, набуває конкретних вимірів. Коли програмна система взаємодіє з іншими системами, особливо важливим є те, що повністю задовольняються вимоги до інтерфейсу і він буде стабільним, тобто забезпечується нормальна взаємодія системних компонент та, у випадку необхідності, ефективні відповідні зміни. Беручи до уваги те, як специфікація трансформується у викональну систему, *архітектурою* програмного забезпечення визначають, як приблизно виглядатиме проєктована система. До цього процесу належать і визначення рівнів системи, і технологічні сценарії. *Впровадження системи* означає, що архітектура системи трансформується у викональний код, внаслідок чого деталізуються та реалізуються такі системні подробиці, які не описуються архітектурою системи. Після написання коду відбувається *тестування* системи. Починаючи від тестування окремих компонент системи (таких, як окремі класи, реалізовані конкретним програмістом), тестування проходить весь шлях аж до перевірки загальної роботи всієї системи в реальному середовищі. У такому процесі останній крок розглядається як завершення *розроблення* системи і подальші кроки гарантують ефективну роботу системи в реальному середовищі.

Треба зауважити, що Ройс ніколи не пропонував лінійний поступовий рух за вищезазначеними кроками, а був впевнений, що у цьому процесі будуть певні цикли та можливість повернутися до попередніх кроків у випадку помилок, суперечностей чи недоліків, які неодмінно виникають у такому складному процесі. Це призводить до ітераційного процесу розроблення програмного забезпечення, де кроки, подібні до поданих вище, повторюються. Відомим ітераційним процесом розроблення є *Unified Process* (розроблений Якобсоном в 1999 році), який, між іншим, порушує послідовність кроків процесу, пропонуючи їхнє паралельне виконання. Ключові моменти виконують роль цих кроків, і таким способом ефективно розробляється програмне забезпечення.

Динамічна технологія створення програмного забезпечення

Методи та модифікації технології створення програмного забезпечення, подані вище, часто критикувалися за надто високу вартість проєктів. Для всіх них загальним є те, що обов'язковим є документування, у якому подається результат виконання різних кроків (специфікації, діаграми тощо), що є одночасно вхідними даними для наступних кроків. Більше того, критика стосувалася також і деякої наблизеності до зовнішніх обставин. Проєкт розроблення програмного забезпечення часто не враховує того, що загальна програма може у значному ступені визначатися вимогами та побажаннями замовника і замовник може значно впливати на розробку.

Технології створення програмного забезпечення мають передбачати повне задоволення потреб майбутніх користувачів. Але ці технології насправді розвиваються так, що спрощують результати того, що мало бути на початку проєкту і отримується у процесі розроблення. У зв'язку із сказаним виникло так зване *екстремальне програмування (Extreme Programming, XP)* – методологія, яка ґрунтується на принципах простоти, комунікативності, зворотного зв'язку, рішучості. Простота означає найпростіше можливе вирішення проблеми; комунікативність та зворотний зв'язок – це постійні контакти всіх осіб, що беруть участь у проєкті, зокрема і майбутнього користувача. Розробники повинні бути *рішучими* у своїх діях, часто та інтенсивно спілкуватися між собою і становити добрий колектив.

Існують й інші підходи, тісно пов'язані з XP-програмуванням, які можна об'єднати одним терміном: *динамічна технологія програмування*. Ця назва була впроваджена групою відомих суспільних особистостей, які опублікували так званий Маніфест динамічної технології розроблення програмного забезпечення (*Manifesto for Agile Software Development*), основними принципами якого є:

- *Індивідууми та взаємодія* стоять вище від процесів та інструментарію;
- *Працююче програмне забезпечення* є вищим від об'ємної документації;
- *Співпраця з клієнтом* є важливішою від договорів та перемовин;
- *Реакція на зміни* важливіша від виконання планів.

Як бачимо, елементи, що знаходяться праворуч у вищеподаних принципах, можна неодноразово виявити у традиційних технологіях створення програмного забезпечення, про які згадувалося вище. Проте Маніфест зовсім не відкидає ці положення, що виявляється у такій фразі Маніфесту: «мають сенс твердження правої сторони речень, проте ми надаємо перевагу лівій частині».

Цілі статті

Від часу свого виникнення на початку 90-х років XX ст. Web революційно змінив світ та наше життя істотніше, ніж інші технологічні інновації у сучасній історії розвитку людства. Протягом останніх 10–15 років світова спільнота отримала можливість доступу до величезного різноманіття інформації за допомогою пошукових машин та інтернет-порталів; виникла та інтенсивно розвивається Web-комерція, що дало змогу компаніям розширити сфери застосування та підвищити ефективність бізнесу. Такій ситуації значною мірою сприяв революційний стрибок ефективності новітніх технологій у сфері застосування Web. Це стосується розвитку як апаратного, так і програмного забезпечення.

Ціллю статті є стислий огляд ситуації, що склалася сьогодні у сфері розвитку різноманітних напрямків використання інтернет-технологій, їх застосування та перспектив (зокрема, відносно бізнесу), сучасного стану новітніх Web-технологій. Цей огляд не претендує на розгляд всіх можливих інновацій у галузі, оскільки таких існує сьогодні велике різноманіття, окрім того, щоденно виникають нові напрямки, удосконалюються раніше впроваджені технології, змінюються сфери їх використання. Тому ціллю статті є насамперед розгляд технічного і програмного аспектів Web-технологій, використання Web-додатків у бізнес-проектах як таких, що найбільш відчули на собі вплив інновацій.

Основний матеріал

Сучасні Web-технології

Загальні технології розроблення програмного забезпечення зокрема можуть використовуватись і під час розроблення Web-додатків. RIA якраз і є тим окремим випадком Web-додатків, де можна повною мірою використовувати технології та їх моделі, такі як діаграми подій, класи та діаграми взаємодії. Слід зазначити, що серед розробників Web 2.0 додатків дуже популярною є *динамічна технологія програмування*. У статті «Що є Web 2.0» Тім О'Реллі подав приклади того, як має виглядати звичайний проект для Web 2.0. Спільно з Массером О'Реллі назвав вісім принципів, які використовуються у проектах розроблення Web 2.0 додатків.

Перший з них формулюється як «якомога більш ранні та часті релізи», «залучення користувачів як співрозробників та тестувальників у реальному часі» та «удосконалення вашого продукту» і є відображенням третього принципу: «співпраця користувачів». Для більшості Web 2.0 додатків на початку проекту користувачі відсутні, отже, використання цього принципу має свою специфіку. На практиці це означає проведення такої роботи, щоб створити користувацьку базу, яка буде задіяна у розроблюваному продукті і який буде реалізований на основі цієї бази. Клієнти (користувачі) згодом встановлюють зворотний зв'язок у питаннях розробленого сервісу, що дасть змогу розробникам удосконалити продукт. Зворотний зв'язок може бути у вигляді коментарів від користувачів, але не обмежується цим, оскільки користування сервісом може моніторитися безпосередньо в реальному часі на сервері. Якщо надана користувачами інформація використовується для «створення набору користувацьких дій», то його можна застосувати для подальшого вдосконалення продукту. Типовим прикладом може бути Flickr, провідний розробник якого Кол Гендерсон (Cal Handerson) якось повідомив, що кожна нова версія подається щопівгодини. На основі цього прикладу спостерігаємо на практиці, як може змінюватися продукт внаслідок моніторингу змін у його використанні.

Як вже було сказано, впровадження першого принципу О'Реллі дає змогу реально розвивати продукт у напрямку його вдосконалення і водночас запобігає, так би мовити, його перепроектуванню: ті властивості, які нікому не потрібні (або не вимагаються) не дають змоги «сирому» продукту надто швидко досягнути ринку продажів і отже, мінімізують ризик повної невдачі проекту. Зауважимо, що, можливо, існує різка суперечність між «не перепроектувати» і використанням традиційної технології розроблення.

Отже, з одного боку, маємо фіксацію вимог на початку проектування і спроби певним чином реагувати на зміни, або марні зусилля, коли зміна вимог виявляється запізно. Важливо зазначити,

що запобігання перепроєктуванню не повинно зводити нанівець творчі зусилля, адже у протилежному випадку жодна з існуючих сьогодні компаній, що використовують Web 2.0, ніколи не змогла б ефективно працювати.

Другий принцип О'Реллі («Прості моделі» і «Рентабельна масштабованість») тісно пов'язаний із динамічним принципом «реагування на зміни». Деякі з кращих методів, пов'язаних з цим принципом, ґрунтуються на тезах «реакція на вимогу», «визначення кошторису і граничних моделей», «корпоративні бізнес-моделі». Ці принципи на практиці означають зміни у власному сервісі як відповідь на вимогу; зміни у можливості перетворення сервісу на гроші; зміни у сервісі відповідно до корпоративних змін.

Розробляючи проекти на основі Web 2.0, традиційні технології та моделі, на нашу думку, є менш придатними і ефективними, ніж динамічні технології програмування, тому останні застосовуються значно частіше і більше відповідають вимогам програмування.

Бізнес-моделі в Інтернеті та Web-додатки

Раніше єдиною можливістю заробляти гроші, окрім продажу певних товарів за допомогою Інтернету, була реклама. Сьогодні такі можливості значно зросли, а також виникли принципово нові підходи.

Класифікація бізнес-моделей, яку подав професор Майкл Репп, Університет Північної Кароліни (США) така:

- засноване на комісії маклерство та продавці;
- реклама;
- інформаційні посередники;
- модель співтовариства;
- підписки.

Скорочено подамо опис кожної з п'яти категорій.

Засноване на комісії маклерство та продавці

Один з найвідоміших Мережевих прикладів – Amazon.com, який починав свою діяльність як електронна книжкова крамниця і сьогодні продає продукцію більш ніж тридцяти різних категорій та дає змогу іншим продавцям використовувати свою платформу. Amazon.com розвиває свій сервіс, пропонуючи різноманітне програмне забезпечення й апаратні *послуги*. Іншим яскравим прикладом є Інтернет-аукціон eBay.

Варіантом моделей, заснованих на комісіях, є так звана *приєднана* модель, яка також стала популярною завдяки Amazon.com. Модель пропонує стимули у формі відсотка прибутку для партнерів, з якими встановлені зв'язки. Встановлюються зв'язки звичайним «кліком» мишкою на сайті партнерів, що безпосередньо провадять до продавця. Комісія заснована на моделі «оплата за роботу» (тобто тільки тоді, коли продажі виконані, проводиться оплата).

Маклерство може мати різні форми. Amazon.com є фактично комбінацією віртуальної *крамниці*, де клієнти можуть переглянути каталог і заповнити кошик закупівель, і віртуального *ринку* або місця для зустрічей, яке можуть «відвідувати» сторонні торговці в режимі онлайн, провести транзакції, отримати список продажів, здійснити комерційні операції, які виконуються від їх імені. Прикладами цієї категорії є, зокрема, Alibaba і mySimon; останній може бути класифікований як *пошуковий агент*, оскільки забезпечує персоналізовану купівлю або інформаційні послуги через інтелектуального агента. Інша форма маклерства – купівля/продаж, виконання замовлення, яка полягає у прийманні замовлень клієнтів щодо купівлі чи продажу продукту або послуги; прикладом може бути CarsDirect. Такі сервіси, як Travelocity і Orbitz спеціалізуються на подорожах і спеціалізуються не лише на звичайному виконанні замовлень, а пропонують оцінку і переговори.

Треба згадати й *комерційну модель*, де, згідно з Майклом Реппом, гуртові продавці і роздрібні продавці товарів і послуг контактують між собою (на відміну від бізнес-структур і потенційних «кінцевих» клієнтів). Продажі можуть бути виконані за цінами прейскуранту або через аукціони вищезазначених типів. Продавці можуть бути повністю віртуальні або здійснювати бізнес-замовлення поштою із використанням Інтернет-каталогів.

Реклама

Реклама сьогодні найбільш розвинута та прибуткова бізнес-модель в Інтернеті; витрати, пов'язані з пошуком оголошення, значно зросли протягом останніх п'яти років. Мережева реклама стала головною бізнес-моделлю з моменту появи Google AdSense і Google AdWords.

Власне реклама – давній засіб залучення споживачів або потенційних клієнтів, зацікавлених у продукції або послугах, зокрема у класичних ЗМІ (газети, радіо, телебачення, кіно, спортивні арени) і в інших засобах інформації. Традиційна реклама завжди характеризувалася двома особливостями:

1. Рекламодавець ніколи не міг бути впевненим, що залучив потенційних клієнтів або відповідну аудиторію. Наприклад, читач газети, можливо, на даний момент не цікавиться ринком нових автомобілів і просто нехтуватиме відповідними оголошеннями щодо нових автомобілів, за які виробник автомобілів заплатив велику суму грошей.

2. Рекламодавцеві доведеться оплатити рекламу незалежно від того, чи існує хоча б найменша можливість отримання прибутку від його реклами.

Інакше кажучи, реклама, можливо, не досягне своєї мети, тобто призведе до марної витрати грошей. У Мережі цей вид реклами був і все ще присутній у формі *банерних оголошень*, які є зазвичай зображеннями, що розміщуються в межах веб-сторінки і які пов'язані з певним постачальником товарів або послуг. Банери часто є випадковими оголошеннями в межах веб-сторінки і цим не значно відрізняються від традиційних оголошень.

Оплата за «клікання»

Як було встановлено Білом Гроссом, засновником GoTo.com (перейменована восени 2001 р. Overture, яка зазнала краху і яку придбав її найбільший клієнт Yahoo! у 2003 р.), та іншими, Всесвітня Мережа може повністю змінити ситуацію: Мережа дає змогу рекламу розмістити так, що цільові споживачі та рекламодавці можуть отримати саме те, у чому вони зацікавлені, у тому випадку, коли їх реклама сприяє торгівлі. Ключем до цього є пошук: висновок, зроблений Білом Гроссом, був таким: якщо люди *шукають* специфічний елемент, це, можливо, непогана ідея для розміщення нашої реклами для елементів тієї самої категорії поряд з результатами пошуку. Отже, розміщення реклами поряд з результатами пошуку має значно більшу ефективність, ніж розміщення її у будь-якому іншому місці, і це є суттю того, що називають *пошуковим маркетингом*.

Треба зауважити, що недостатньо лише дозволяти рекламодавцям вибирати ключові слова і терміни, у яких вони зацікавлені, оскільки це створило б конфлікт інтересів. Тому ідея GoTo полягала в тому, щоби рекламодавці платили за вибір місця в рейтингу під час пошуку і примушувати їх заплатити один цент за клік, якщо б користувач через цю рекламу приходив до них. Термін *оплачених клікань* або *плати за клік* (PPC) відомий також як *вартість одного кліку* (CPC, *cost per click*).

Google AdWords

У Google AdWords рекламодавець не платить за те, щоб бути внесеним до списку результатів пошуку, а може купити “рекламні слова.” Постачальник певних товарів або послуг формує оголошення, яке обмежене декількома лініями тексту, і вибирає одне або більше ключових слів, які стосуються реклами. Якщо користувач шукає одне з цих ключових слів, реклама, можливо, буде розміщена поряд з пошуковими результатами таким чином, щоби вона була скерована до осіб, зацікавлених даним товаром. Користувачі можуть клікнути на цьому оголошенні для того, щоб бути перенаправленими у рекламовану крамницю, і, в ідеалі, здійснити там закупівлю.

Знову ж, рекламодавець лише тоді платить Google, коли особа, що здійснює пошук, клікає на сторінці результату і переходить на сайт рекламодавця. Сума, яка сплачується при цьому, визначається на аукціоні, у якому всі зацікавлені, зокрема, результатами пошуку чи ключовими словами. Google застосовував модель CPM, у якій рейтинг рекламодавця визначається кількістю звертань до сторінки з оголошенням.

Відтоді, як Google почав використовувати стратегію включення до списку пошукових результатів, засновану, окрім інших параметрів, на PageRank сайтів, не дивно, що ранжування також було впроваджене у рекламу на Google. Тобто Google інтегрував популярність оголошення або його “рейтинг клікання” у його повне ранжування.

Google пішов на крок вперед у бізнес-рекламі, впровадивши *AdSense* у 2003 р., де використовується підтримка реклами дещо іншим способом. AdSense надає доступ власникам Web-вузла до мережі рекламодавців Google таким способом, що вони мають можливість розмістити свою рекламу шляхом використання Google на своїх власних сайтах. Після реєстрації AdSense проглядає сайт нового клієнта і розміщує контекстуально доречні оголошення на цьому сайті або розміщує їх поряд з результатами пошуків на цьому сайті. Власник Web-вузла надалі отримує відсоток, якщо хтось клікає на таке оголошення і далі – через Google – на рекламодавця. Google AdSense дозволив навіть маленьким Web-сторінкам створювати прибуткові потоки без потреби у великих адміністративних силах.

Варто наголосити, що мережева реклама змогла набути характеристик повноцінного бізнесу, створити потужні ринкові відносини, потужність яких ніхто не міг навіть уявити і передбачити.

Інформаційні посередники

Наступна категорія бізнес-моделей – *інформаційний посередник* (англ. *infomediary*). Властиво, інформаційний посередник є еволюцією моделі інформаційного брокера. *Інформаційний брокер* збирає дані ринку, дані щодо розвитку ринку, інформацію щодо ринкових змін як усередині, так і за межами країни, типові дані про те, скільки витрачається на рекламу і чим є загальні канали розподілу, проводить реальну оцінку тенденцій споживчого попиту. Оскільки все більше людей використовує Інтернет і оскільки електронна комерція стала великим бізнесом, відомості про споживачів, їх звички, їх «потоки кліків» стали важливою інформацією для бізнес-структур. Раніше неопрацьовані дані отримувалися з файла реєстрацій Web-сервера, і це була не дуже цікава інформація, але вона стає дуже доречною, коли вона належним чином аналізується на основі сучасних комп'ютерних технологій аналізу і може надалі використовуватися для рекомендацій, оптимізації сторінки, цільового маркетингу тощо.

Найвідомішими у цій галузі є Nielsen/NetRatings – компанія, яка досліджує різноманітність аспектів Інтернет-аудиторії і є важливим джерелом для аналізу оперативних рекламних ефектів; компанія Doubleclick, яка збирає і поширює інформацію щодо реклами і зводить разом агентства, виробників, продавців, фахівців з маркетингу з метою цифрового маркетингу. Як приклад того, яку інформацію можуть надавати такі інформаційні посередники (повна кількість відвідувань або кількість відвідувань за тематичними категоріями), на рис. 1 подано вигляд сторінки, створеної Google Analytics. Серед ефективного інструментарію, що належить до цієї категорії, відзначимо AWStats, Webalizer.

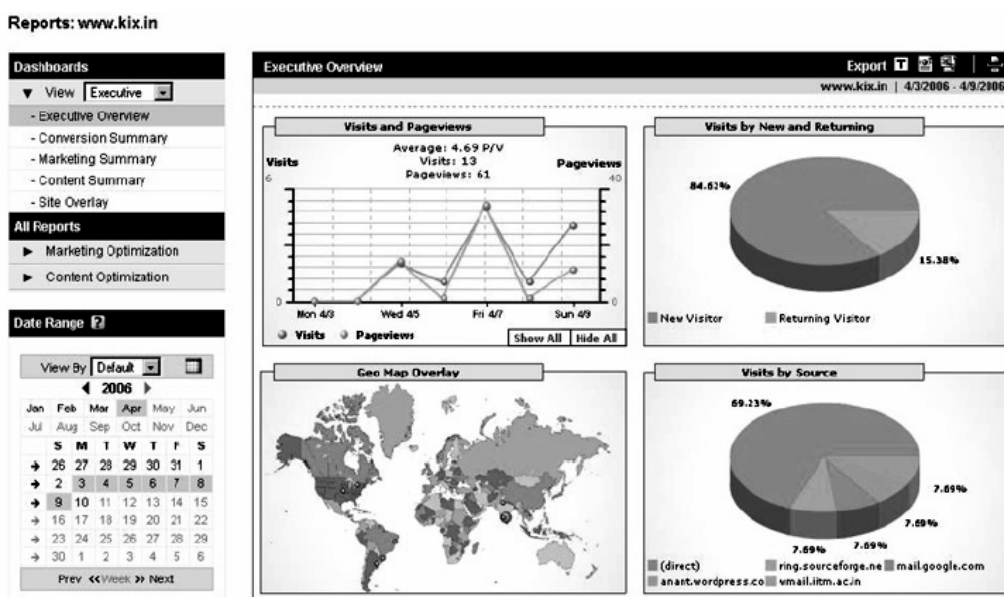


Рис. 1. Загальний вигляд Google Analytics

Інша мета збирання даних, окрім власне трафіка – виробники та їх продукція, відповідно підготовлена інформація про яких буде важливою для споживачів, які бажають щось придбати.

Суспільство і моделі підписок

Суспільна модель є наслідком виникнення такого явища, як розроблення програмного забезпечення з відкритим кодом, де програмісти спільними зусиллями розробляють і розвивають програмний код, зокрема без грошових зборів за ліцензії, і відкритого погодженого сумісного використання у мережах як, наприклад, Wikipedia. Відкрите програмне забезпечення є безкоштовним, хоча деякий прибуток може бути отриманий за рахунок підтримки або супроводу продукту, або інших послуг, які мають відношення до програмного забезпечення. Інші форми суспільних моделей існують в Інтернет-радіо та у Інтернет-телебаченні, оскільки вони мають переважно статус некомерційних сайтів і підтримка яких відбувається за рахунок добровільних внесків (наприклад, SHOUTcast та інші).

Тому суспільна модель складається із співтовариства розробників, користувачів або їх сукупності і забезпечує учасників певним безкоштовним продуктом або послугою та ґрунтується на лояльності користувачів, оскільки вони зазвичай інвестують доволі небагато свого часу та досвіду. Прибуток може отримуватися від об'єднання суспільної моделі з іншими моделями, такими як, наприклад, реклама або підписка на додаткове обслуговування. Саме така комбінація перетворює суспільство на справжню бізнес-модель у сенсі, визначеному вище. Отже, за такого підходу до соціалізації впливає, що Інтернет став місцем для суспільних бізнес-моделей, оскільки суспільні об'єднання зазвичай формуються навколо специфічної теми або мети, тобто можуть бути чудовими потенційними клієнтами. Тому не треба дивуватися, що суспільна модель є дуже цінною і важливою, вона постійно розвивається і розширюється. Можна констатувати, що ця модель буде головною моделлю для майбутнього Глобальної Мережі, яка на віддалену перспективу може стати навіть кращою за рекламу.

Остання модель, яку зараз розглянемо, є *підписна модель*. У цій моделі користувачі залучаються до підписки на певну послугу (наприклад, виняткове право використання доменного імені; орендування простору пам'яті у постачальника Інтернет-послуг для Web-вузла підприємства; орендування DSL-доступу до Інтернету, який забезпечується місцевою телекомунікаційною компанією; плата за телепередачі, що забезпечуються каналом, який використовує Інтернет).

Власники інформації

Інформація і потоки інформації сьогодні є однією з головних особливостей Всесвітньої Мережі. Дійсно, інформація не тільки надходить до Мережі від користувачів чи то у вигляді «живих журналів» (блоги), чи різноманітних статей, оглядів або рекомендацій. Інформація (дані) автоматично створюється у багатьох програмних додатках (наприклад, заснованій на транзакціях електронній комерції, де використовується певна з моделей, описаних вище; в цифровій телефонії; у файлах реєстрації Web-сервера тощо); інформацію збирають пошукові сервери або супутники, іноді навіть урядові організації. Внаслідок сучасних технологій та можливості виробництва дешевого дискового устаткування величезні набори даних сьогодні є реальністю, і великі центри збирання даних встановлюються інтернет-провайдерами постійно.

У цих областях інформації чи даних, що продукуються як людиною, так і комп'ютерами, можемо спостерігати різні технології та методи управління накопиченням даних. Зрозуміло, що створювані людиною дані повинні кінцем кінцем залишитися під контролем людини або суспільства, які виробляють ці дані, проте це не завжди так. Наприклад, каталог продукції, створений для Amazon.com, вже не є власністю автора розробки, оскільки розробник надав ці права сайту. Блог, можливо, “належить” блоггеру і ця персона відповідальна за інформацію, яка існує в межах блогу. Хоча, якщо блог є на деякому хості, контроль над блогом фактично здійснюється кимось іншим. Якщо коментар блогу неприйнятний або невідповідний, власник блогу може знищити його. Якщо індивідуальна інформація блогу стає занадто великою чи давно не змінювалася, власник може редагувати цю інформацію або скорочувати її. Але суспільство

фактично не контролює особливості та вміст наборів даних, тому виникає проблема явного чи опосередкованого контролю за певними визначеними правилами.

Щодо інформації, яка «генерується» комп'ютером, приналежність даних часто встановлюється «сама собою», оскільки значна кількість даних просто зібрана і записана на серверах компанії або постачальника. Типовий приклад – дані транзакцій, які є результатом ділових транзакцій в електронній комерції (тобто дані, які пов'язують покупців, платежі, замовлення, відвантаження, постачальників і які повинні бути прораховані та збережені). Часто створювана комп'ютером інформація характеризується надлишковістю і взагалі кількість утворених даних є такою значною, що такі методики, як інтеграція даних, складування даних (data mining), пошук даних, використовуються для перетворення необроблених даних в інформативніші результати або для створення на цій основі нової інформації.

Збір даних та управління збором даних не є новими проблемами, які виникли у зв'язку з появою Всесвітньої Павутини. Дійсно, телефонні довідники та різноманітні каталоги використовувалися ще в кінці XIX ст. і завжди були важливі для телефонної компанії, яка створювала їх та підтримувала в актуальному стані.

Річ у тому, що існування наборів даних, таких як, наприклад, телефонний довідник в електронній формі, дає змогу інтегруватися в інші додатки. При цьому виникає можливість створення на цій базі нових послуг, зокрема «оберненого» телефонного довідника (тобто, поєднання телефонних номерів з іменами та адресами замість звичайного способу), відслідковування змін телефонних номерів, створення каталогів, не поданих у списку телефонних номерів тощо.

Телефонний довідник – добрий приклад того, що можна часто спостерігати у Мережі, а саме: він розповсюджується безкоштовно, але існують шляхи отримання прибутку за допомогою телефонного довідника, зокрема продаючи рекламу для введення її до Жовтих Сторінок. Аналогічний принцип у своєму сервісі використовує Google: пошуковик з його величезним індексом може використовуватися безкоштовно, а основним джерелом прибутку є реклама.

Програмне забезпечення як сервіс (SAAS)

Реальною можливістю для компанії значно скоротити видатки на ІТ є застосування сучасних Інтернет-технологій і послуг, використання яких сьогодні стає значно менш вартісним; така ситуація незабаром стосуватиметься і програмного забезпечення, а сьогодні є актуальною для програмного забезпечення з відкритим кодом. Одним з ключів до цього розвитку є *програмне забезпечення як сервіс (SaaS)*.

SaaS використовується сьогодні у двох головних формах: для розробників та для кінцевих користувачів (компанії і приватні особи) у формі так званих «feature-rich» послуг або наборів віджетів, які є безкоштовними або які можна придбати чи орендувати.

Погляд назад: Модель ASP

Як вже згадувалося вище, Інтернет-додатки (RIA) сьогодні забезпечують певну функціональність та узгодженість, яка раніше існувала лише для додатків для персональних комп'ютерів. RIA може відтепер розглядатися як матеріалізація мережевого комп'ютера, появу якого передбачалося деякими компаніями, зокрема Sun Microsystems і Oracle, в середині 90-х років. Мотивація для таких припущень на той час виникла на основі концепції *Забезпечення сервісу додатків (Application Service Provisioning, ASP)*, яка ґрунтувалася на ідеї забезпечення функціональності програмного забезпечення через Інтернет. У моделі ASP постачальник не продає програмне забезпечення як пакет, який клієнтові потрібно встановити на власному сервері.

Програмне забезпечення або додаток знаходяться на комп'ютері, який належить провайдеру і використовується кожного разу, коли це потрібно, за допомогою інтернет-з'єднання. У середині 90-х років модель ASP розглядалася як метод зробити програмне забезпечення дешевшим, оскільки в той час вартість програмного забезпечення зазвичай зростала, що призводило до зростання вартості технічних засобів. Окрім того, особливо для малих і середніх бізнес-структур, у яких зазвичай був відсутній

власний відділ системного адміністрування, інсталяція та супровід програмного забезпечення були доволі складною справою. Тому, замість купівлі дорогого програмного забезпечення та неможливості ефективної роботи внаслідок різноманітних проблем із ним, виникла ідея ASP – надавати програмні послуги за вимогою. Тоді відпадає потреба у встановленні та підтриманні, і системна служба постачальника послуг може оперативно модернізувати програмне забезпечення або усувати помилки програмного забезпечення, не залучаючи до цього складного процесу клієнта.

Програмне забезпечення надається в користування як шляхом простого доступу, так і у вигляді спеціалізованої програми, як, наприклад, конвертер валют, або певне програмне забезпечення для роботи з банківськими рахунками через віртуальні конференції і аж до складних пакетів прикладних програм. Зазвичай єдиний провайдер має на своєму хості низку найрізноманітніших додатків, що ілюструється на рис. 2. Коли додатки надаються у користування віддалено, технічний інструментарій, який при цьому використовується, може бути віддаленим методом доступу (RMI), Java-аплетами певного рівня, та іншим програмним інтерфейсом.

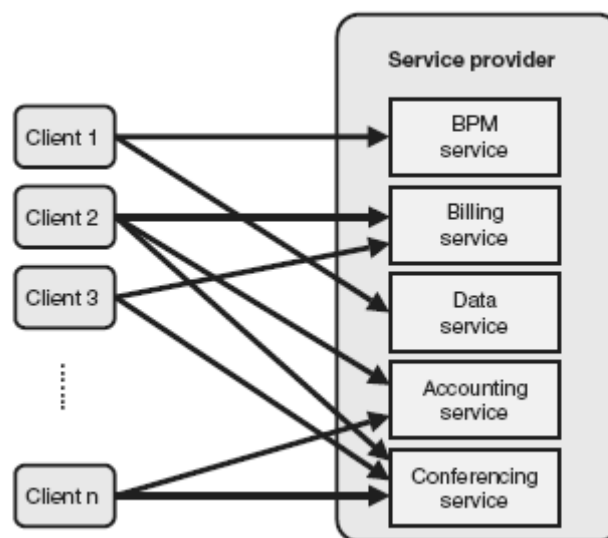


Рис. 2. Приклад архітектури провайдера сервісів

У моделі ASP велика відповідальність залишається за провайдером, оскільки йому потрібно надавати *сервісні гарантії* щодо придатності, безпеки, резервувань, технічної підтримки, всього того, що раніше доводилося реалізовувати клієнту. Коли модель ASP була впроваджена у 90-ті роки, виникла надія, що врешті-решт *все* програмне забезпечення на комп'ютері може перебувати десь в Інтернеті, зокрема і операційна система – все це реалізовувало ідею вищезазначеного мережевого комп'ютера.

Хоча для низки додатків модель ASP успішно використовувалася (зокрема найефективніше у США бізнес-структурами, урядовими організаціями, некомерційними організаціями тощо) модель ніколи не досягла рівня успіху, який передбачався спочатку. Головна причина цього полягала у технічних проблемах, зокрема у тому, що інтернет-з'єднання у кінці 90-х років були надто далекі від тієї надійності і швидкості, які існують сьогодні.

Технічна проблема обмеженої пропускної здатності і стабільності здебільшого сьогодні вже подолана, хоча завжди існуватимуть запити щодо вдосконалення та вищих норм. Проте зводити все лише до технічних проблем (кабелі, зв'язок), які дадуть змогу моделі ASP еволюціонувати, було б не зовсім коректно.

Погляд, орієнтований на провайдера

Коли програмні послуги надаються через Інтернет (наприклад, Google Maps API), провайдер встановлює шляхи, якими може відбуватися обслуговування. API з погляду сервісу зазвичай

складається з низки процедур, до яких можна звернутися або які можна ввести, та все ж ці процедури фіксовані і їх доводиться використовувати такими, які вони є. Тут треба виділити обслуговування додатків і обслуговування інфраструктури.

Обслуговування додатків

Бізнес-модель обслуговування додатків, де певна функціональність програмного забезпечення існує за умови, що клієнт використовує його як додаток, є зазвичай моделлю підписки. Обслуговування могло би бути безкоштовним, але більш загальний підхід – підписка або оплата за принципом «pay-as-you-go», тобто «оплата за користування». Отже, клієнтам доведеться платити, але зникає потреба у початкових інвестиціях в технічні засоби або програмне забезпечення (див. знову рис. 3) за винятком випадків, коли наданий додаток інтегрується в інші додатки на боці клієнта.

Як приклад можемо розглянути *Google App* (www.google.com/a). Служби Google – інструментарій, який містить Gmail, Google Talk для чатів і розсилки миттєвих повідомлень, Google Calendar, інструментарій для планування подій і зустрічей і для сумісного використання, Page Creator для проектування веб-сторінок, Start Page для створення персональної головної сторінки браузера, Google Docs & Spreadsheet для роботи з текстом і електронними таблицями, яка дає змогу спільно декільком людям працювати з документами або проводити обчислення за допомогою електронних таблиць. Всі ці компоненти, як і зберігання даних, виконуються на серверах Google. Визначена вартість становить п'ятдесят американських доларів на рік за обліковий запис користувача, тоді як основний сервіс з простором дискової пам'яті для поштової скриньки залишається безкоштовним.

Існує велике різноманіття сервісів, схожих до сервісу Google, які впроваджуються майже щодня, у діапазоні від індивідуальних сервісних дзвінків до повних складних програмних комплексів. Повніше уявлення щодо існуючих послуг і програмного сервісного забезпечення дає таблиця.

Програмне сервісне забезпечення

Application	Company	URL
Organizer and Calendar	Kiko	www.kiko.com
	37signals Backpack	www.backpackit.com
	30Boxes	www.30boxes.com
Planning	PlanHQ	www.planhq.com
Word Processing and Spreadsheets	EditGrid	www.editgrid.com
	ThinkFree	www.thinkfree.com
	Google Docs & Spreadsheets	docs.google.com
	Ajax 13 ajaxWrite, ajaxXLS	www.ajaxwrite.com
	Num Sum	www.numsum.com
	Zoho	www.zoho.com
	Peepel	www.peepel.com
	ShareOffice	www.sharemethods.com/products/shareoffice.html
Distributed Database Applications	Smallthought Systems	www.dabbledb.com
Invoicing and Time Tracking	Dabble DB	
Project Management, Collaboration, Shared Workspaces, Conferencing	FreshBooks	www.freshbooks.com
	37signals Basecamp	www.basecamp.com
	CentralDesktop	www.centraldesktop.com
	WebEx	www.webex.com
	GoToMeeting	www.gotomeeting.com
	activeCollab	www.activecollab.com
	TeamWork Live	www.teamworklive.com
Group Chat	LiveOnTheNet	www.liveonthenet.com
	Zimbra Collaboration Suite	www.zimbra.com
	37signals Campfire	www.campfirenow.com
Accounting and Payroll	Xero	www.xero.com
	Intacct	www.intacct.com
	KeepMore	www.keepmore.com
Customer Relationship Management	Etelos CRMforGoogle	www.crmforgoogle.com
	salesforce.com	www.salesforce.com
	RightNow	www.rightnow.com

Сьогодні намагаються зробити додатки, які раніше виконувались виключно на персональному комп'ютері, доступними як сервіс у Мережі, зокрема з мережевою операційною системою, та з деякими іншими проектами, що знаходяться на стадії реалізації.

Обслуговування інфраструктури

Вище розглядався сервіс, який є *додатком* з певною функціональністю, проте існує інша категорія, а саме *обслуговування інфраструктури*, яка також пропонується у Мережі. Послуги цієї категорії використовують принцип провайдерів в наданні дискового простору та збільшення потужності обчислень, або і те, і інше разом взяте, хоча також існують й інші види функціональності.

Найбільш вдалим прикладом у цій категорії є Amazon Web Services. Amazon.com сьогодні пропонує десять різних мережевих послуг, які є справжніми послугами інфраструктури у тому сенсі, що вирішують питання розвитку продуктів і не призначаються для індивідуальних користувачів.

У процесі розвитку продукту або під час старту компанії виникають такі проблеми, як хостинг сервера, здійснення переговорів, управління пропускнуою здатністю, прийняття рішень, масштабування та управління фізичним зростанням, різноманітними технічними засобами, координація роботи великих колективів. Amazon Web Services якраз і працює із цими проблемами; ця служба виникла як побічний продукт величезної обчислювальної інфраструктури, яка належить Інтернет-середовищу. Для прикладу розглянемо дві з них.

Amazon S3 (*Simple Storage Service*), про яку вже згадувалося у контексті мережевого сервісу, забезпечує ресурси дискової пам'яті в Інтернеті, що коштує п'ятнадцять центів за гігабайт за місяць зберігання і двадцять центів за гігабайт переданих даних, без податкових грошових зборів чи витрат на інсталяцію.

Amazon EC2 (*Elastic Compute Cloud*) – обчислювальне обслуговування, яке дає клієнтам змогу збільшувати потужність обчислень відповідно до їх завдань, що коштує десять центів за використану комп'ютерну годину. Сервіс забезпечує “віртуальне обчислювальне середовище, дозволяючи використовувати мережеві сервісні інтерфейси з метою отримання доступу до віртуальних комп'ютерів для використання, завантаження їх клієнтським середовищем додатків, управлінням дозволами доступу до локальної мережі, використовувати мережеве середовище та стільки операційних систем, скільки є необхідно”. За потребою обслуговування масштабується, так що користувач може використовувати, наприклад, єдину машину протягом повного місяця (тридцять днів двадцять чотири години на добу) за ціною сімдесят два долари або, якщо необхідно, 720 серверів протягом однієї години за тією самою ціною. Отже, EC2 – комерційна форма grid-обчислень.

Орієнтований на споживача погляд та налаштування сервісу

Існує й інший підхід щодо SaaS, а саме з погляду споживчої перспективи. Під цією перспективою споживач або клієнт сервісу розуміє таке обслуговування, при якому він може встановити технічні вимоги, які включають бажану функціональність, ціну, необхідну ефективність сервісу, швидкість (час) виконання та інші параметри. Клієнт тоді шукав би такого постачальника послуг, який задовольняв би умови цих технічних вимог.

Прикладом сервісного ринку може бути *Amazon Mechanical Turk*, який є однією з десяти мережевих послуг, що пропонуються Amazon.com. *Amazon Mechanical Turk* не є ринком автоматизованих послуг, але забезпечує API, через який програмісти або *замовники* можуть представити завдання, які мають бути виконаними; вони можуть також схвалити завершені завдання, а результати виконання ввести до своїх власних додатків програмного забезпечення. З пункту погляду програміста забезпечення такого сервісного запиту виконується через WPC у програмний інтерфейс Turk API. Проте, важливо мати на увазі, що запит на обслуговування фактично забезпечується особами, які використовують “НІТ” (Human Intelligence Task), виконують завдання і отримують за це платню.

Типове завдання або опис “НІТ” встановлює дату закінчення виконання для “НІТ” та винагороду для кожного завершеного завдання “НІТ”, яка становить зазвичай декілька центів. Завдання, подане замовником, може, наприклад, бути таким: у певний програмний додаток для роботи з GIS-форматом додати 50 малюнків і вибрати на кожному з них контури доріг; *виконавець* (тобто, людина, що приймає завдання до виконання) отримує п'ятдесят зображень і працює з кожним, позначаючи потрібні контури. Кожен завершений “НІТ” винагороджується вісьмома центами. Справа тут у тому, що завдання, пропоновані як “НІТ”, переважно є такими, що не можуть

бути зазвичай автоматично виконані комп'ютером і тому виконуються людьми. Проте результати, отримані людиною, як, наприклад, зображення у вищенаведеному прикладі, переносяться надалі у програмний додаток.

Спостерігається тенденція збільшення кількості компаній, які пропонують різноманітні підходи для створення індивідуальних програмних додатків, так що на сьогодні користувачеві стає досить просто встановити у себе, наприклад, електронну крамницю. Більшість із пропонованого програмного забезпечення ґрунтується на віджетах (widgets) (тобто, невеликих повторно використовуваних компонентах, які забезпечують просту інтеграцію багаторазових послуг). Для електронних крамниць, які зараз створюються надзвичайно легко, термін *MeCommerce* використовується сьогодні як синонім для “моєї власної крамниці”. Можна стверджувати, що сьогодні маємо чітку тенденцію: можна легко заснувати власний бізнес у будь-якій невеликій ніші без особливих зусиль.

Соціалізація

Тепер розглянемо третій головний напрямок поточного розвитку Мережі, який зазвичай називають “соціалізацією”. Соціалізація стосується аспекту вироблюваного користувачем вмісту, оскільки відбувається у блогах, виявляється у маркуванні, також як і в соціальному створенні закладок. У контексті цього огляду соціалізація не означає процесу вивчення соціальних відносин. Тут розглядаємо додаткові аспекти соціалізації, зокрема ті, що стосуються програмного забезпечення. Іншими словами, “соціальне” програмне забезпечення – це програмне забезпечення, яке стає все досконалішим внаслідок того, що ним користується велика частина суспільства. Якщо висловлюватися точніше, це, власне, не означає розвиток програмного забезпечення внаслідок інтенсивного використання, але виникнення різноманітних додатків на основі цього програмного забезпечення. Нижче розглянемо, як соціалізація і перетворення вмісту у Мережі взаємодіють з бізнес-моделями, поданими вище.

Соціальний пошук

Відома дилема з технологіями Мережевого пошуку – це, з одного боку, аналіз запиту та індексування все ще не є настільки досконалими, щоб забезпечити автоматизований процес ухвалення рішення щодо оцінки інформації, яку містить Web-вузол. З іншого боку, кожна особа здатна вирішувати, яка інформація є корисною для вирішення поставленої мети, але вона не має можливості чи часу, щоб проглянути мільйони відповідей, які надає пошукова машина у відповідь на запит. Хоча ця проблема має декілька аспектів, все ж тут розглянемо лише аспект соціалізації у зв'язку з пошуком.

Для визначення наміру користувача, коли запит формулюється до пошукового сервера, є простий і очевидний елемент: попередня (в минулому) поведінка. Це вимагає лише, щоби користувач зареєструвався на пошуковому сервері і, отже, його попередні запити тоді будуть відстежуватися і зберігатися для майбутнього використання. Оцінка призначених для користувача відгуків (враховуючи попередні результати пошуків) може використовуватися, щоб передбачити справжні зацікавлення користувача для майбутніх запитів. Наприклад, довга історія пошуків на сайтах виробників автомобілів вказує на значне зацікавлення цією темою. Якщо далі користувач ініціює пошук, скажімо, автомобіля «Ягуар», це, скоріш за все, означає, що він має намір оновлювати чи отримувати інформацію щодо автомобіля цієї марки, а не цікавиться твариною-ягуаром. З іншого боку, активіст організації захисту тварин, що подає той самий запит, ймовірно, зацікавлений саме твариною та оточенням, у якому вона перебуває.

Звідси виникає ідея введення історії пошуків до самого пошуку, тому провайдери пошукових серверів вже почали пропонувати таку послугу, як *персоналізований пошук*. Є два головні підходи щодо персоналізації у пошуковій машині: перший – використання *cookies*, інший ґрунтується на явній реєстрації. Останній підхід показує добрий результат в сенсі покращання пошукових результатів, оскільки улюблені сайти користувача мають вищу релевантність для персоналізованого користувача, ніж для анонімного при користуванні стандартними методами пошуку.

Соціальні аспекти програмного забезпечення

Соціалізація у формі внесків користувачів у вміст Мережі і спільне використання цього вмісту співтовариствами не тільки застосовується у пошукових машинах та впливає на їх основний бізнес; вона також на сьогодні є явищем, яке має далекосяжні наслідки навіть для інших видів програмного забезпечення. Як згадувалося раніше, соціальне програмне забезпечення – програмне забезпечення, яке стає кращим від того, чим більше людей використовують його. Власне протягом тривалого часу програмне забезпечення не змінюється (як програмна система), але якщо його застосовують багато користувачів з певною частотою, то воно називається *додатком*, який є складовою частиною програмного забезпечення і фактично є удосконаленням програмного забезпечення.

Соціальне програмне забезпечення або, точніше, соціальні аспекти програмного забезпечення не тільки стають важливими в персональних мережевих додатках чи електронній комерції – очікується, що соціальне програмне забезпечення матиме великий вплив і у таких випадках, коли люди встановлюють з'єднання з колегами, партнерами, клієнтами, впливатиме на ведення бізнесу компаніями та контакти з їх клієнтами. Компанії не тільки очікують кращого управління знаннями персоналу, але й збільшення продуктивності і кращого управління зв'язками з клієнтами. Як наслідок, великі ІТ компанії, що створюють програмне забезпечення, почали ставитись до цієї сфери як до специфічної сфери ринкових відносин, у якій використовується традиційний канал розподілу; прикладами нових програмних розробок у цій категорії соціального програмного забезпечення можуть бути IBM-Lotus Connections, пакет Oracle WebCenter, Intel's Suite.

Он-лайніві впливи соціальних мереж

Попередньо вже висловлювалася теза, що он-лайніві соціальні мережі змінюють стосунки у Мережі, які перестають бути лише звичайними зв'язками між Web-сторінками: вони стають зв'язками між людьми і співтовариствами. У такій мережі прямі посилання зазвичай вказують на найближчих друзів і приятелів, опосередковані посилання провадять до друзів наших друзів і так далі. Вже наголошувалося, що *соціальна мережа* у Всесвітній Павутині переважно є результатом використання програмного забезпечення, яке зосереджене на побудові онлайн-співтовариства з визначеною метою. Соціальні мережі поєднують людей із спільними інтересами і можуть бути як звичайним блогом, так і складнішим утворенням, скажімо, як згадуваний MySpace. Вище розглядалася низка соціальних мережевих сайтів, які роблять наголос на професійному використанні (наприклад, LinkedIn або openBC), або які дають можливість сумісного використання будь-якого виду ЗМІ, сьогодні переважно відео (наприклад, YouTube). Соціальна мережа може бути засобом для контактів працівників та експертів різного профілю департаментів та філій компанії, допомагати їм створювати деякі профілі значно легшим шляхом, набагато дешевше і гнучкіше, ніж традиційні адміністративні системи знань. Як тільки профіль був встановлений і опублікований у Мережі, будь-хто зацікавлений може відшукати персоналії із відповідним специфічним знанням і з'єднатися з ними. Зокрема, такі платформи як, наприклад, LinkedIn, openBC або Ryze, надають можливість онлайн-спілкування.

Дуже важливим є те, що розвиток Мережі у цьому аспекті призводить до того, що контакти окремих людей та співтовариств стають простішими та ефективнішими і відтепер не важко підтримувати професійну або особисту мережу приятелів у всьому світі. Інший аспект є той, що соціальна мережа відкриває нові джерела прибутку. Дійсно, деякі з бізнес-моделей, які розглядалися вище, можуть легко бути застосованими й до соціальної мережі, особливо у сфері рекламування. Наприклад, MySpace має “ систематизовану ” секцію, яка охоплює рекламні оголошення у таких категоріях, як вакансії робочих місць, послуги, нерухомість, рентні прибутки, автомобілі, продажі, тощо і локалізує їх у низці міст США та поза ними.

Як визнала більшість великих компаній (і що є безпосереднім продовженням зосередженої реклами, яку застосовують провайдери пошукових машин), соціальні мережі, особливо у Всесвітній Павутині, можуть зробити значно більше для бренду або популярності продукту, ніж традиційні маркетингові підходи.

Оскільки врешті-решт метою бізнесу, зокрема і в Інтернеті, є отримання прибутку, у цьому контексті виник новий термін – «соціальні закупівлі» як ідея створення електронних закупівель із соціальним вмістом внаслідок інтенсивного сумісного використання рекомендацій або оглядів

продукції. Прикладами є Chitika ShopCloud\$, eMiniMalls, Chitika ShopLinc, Loomia, PowerReviews, Kaboodle. PowerReviews, наприклад, надає свою послугу безкоштовно, а їх бізнес-модель заснована на отриманні прибутку від огляду вмісту через рекламу і механізм PPC їх порталу закупівель.

Створений користувачем рекламний вміст

Соціальні мережі, блогінг, спільне використання вмісту сайтами завжди ідентифікувалися як добре місце для цілеспрямованих реклам та отримання великих прибутків завдяки величезній кількості звертань, проте є й інший бік цієї бізнес-моделі, що називається контекстним рекламуванням. До цього часу рекламні оголошення здебільшого подавалися *поряд із* вироблюваним користувачем вмістом. Подальшим еволюційним кроком стала *інтеграція* рекламних оголошень у цей вміст – підхід, який використали виробники онлайн-ігор (наприклад, Massive Inc. та інші). Наступним кроком буде розміщення реклами безпосередньо на створюваних користувачем додатках; іншими словами, користувачі стають *творцями* реклами.

Яскравим прикладом інтеграційного підходу є такі компанії як Podcaster Ads, Visible World, Audioads, які інтегрують рекламу в аудіо- та відеокліпи. Використовуючи таку платформу, користувач, що створює кліп, може заробити гроші, вміщуючи рекламу на початку або кінці кліпу; кожного разу, коли кліп завантажений, власник кліпу отримує гроші.

«Second Life»

Завершальний приклад додатку для Web 2.0, популярність якого швидко росте – «Second Life». «Second Life» є онлайн-рольовим сайтом (сценарії різноманітних життєвих ролей), створений у Сан-Франциско лабораторією Linden Lab. Він забезпечує тривимірний віртуальний світ, який будується і є власністю “мешканців.” Для того, щоб стати таким мешканцем, користувачам потрібно зареєструватися в одній з декількох категорій і вибрати *аватара* (тобто, визначене віртуальне зображення, яке може бути змінене, якщо *аватар* не задовольняє користувача). «Second Life» може розглядатися як 3D варіант Інтернету: замість звичайної домашньої сторінки користувачі можуть створити будинок, свій бізнес або навіть острів у «Second Life»; замість e-mail адреси та визначеного користувачем імені він має персональне *аватар*, чії рухи і дії (активність) керуються клавіатурою комп'ютера користувача.

Сайт «Second Life» був започаткований у 2003 р. і з того часу «вибухово» розвивався, залучивши більш ніж чотири мільйони зареєстрованих користувачів у кінці лютого 2007 р. У межах свого віртуального світу користувачі можуть впливати на нього і змінювати цей віртуальний світ та його середовище. Користувачі можуть побудувати будинок або започаткувати бізнес, можуть купувати, продавати, спілкуватися з іншими мешканцями віртуального світу. На віртуальному *Ринку* сайту підтримуються щомісячні операції в мільйонах американських доларів, але фактичною валютою є *Linden доллар*, який конвертується в американські долари в онлайн-режимі. Обмінний курс змінюється, але був досить стійкий на початку 2007 р. і становив близько 270 *Linden*-доларів за один американський долар. Орієнтація в межах віртуального світу відбувається за допомогою карт, меню або контактів з іншими віртуальними персонажами.

Важливо зазначити, що не тільки приватні користувачі використовують з різних причин Second Life. Зростає кількість компаній, які позиціонують Second Life як засіб реклами, маркетингу, презентацій, контактів, ділової платформи. У середині 2006 р. Adidas Reebok відкрив у Second Life крамницю, Toyota почала використовувати маркетинг-модель Scion xB насамперед для молоді аудиторії. Можна перелічити й інші компанії, які почали використовувати Second Life для представлення продукту, маркетингу або взаємодії з клієнтами, а саме: BMW, Nissan, Mazda, General Motors, Dell, Sun Microsystems, IBM, Circuit City, німецька Bild.T-Online, яка пропонує віртуальну газету під назвою *AvaStar*; цей список щоденно зростає. DaimlerChrysler розмістив у Second Life віртуальний демонстраційний зал, випробувальний стенд, на якому їх автомобілі можуть взяти участь у випробувальних поїздках. Circuit City спільно з IBM відкрили крамницю у грудні 2006 р.; обидві компанії займаються дослідженнями, як цей віртуальний світ і 3D-Інтернет можуть бути об'єднані в майбутніх багатоканальних концепціях удосконалення продукції і чи це може бути перспективним комерційним каналом. Нарешті, ця крамниця демонструє віртуальні макети продукції, яку Circuit City також пропонує в його реальних крамницях у натуральну

величину. Зрозуміло, продукція може бути замовлена безпосередньо у цій віртуальній крамниці, яка властиво замінює онлайн-каталог.

Деякі компанії, зокрема Sears, експериментують з *віртуальною торгівлею*: коли новий продукт подається на платформі, подібній до Second Life, компанія може перевірити свій успіх на ринку, вивчаючи, як люди коментують продукт на блогах. Якщо продукт переважно критикується, можливо, немає сенсу просувати цей продукт на реальному ринку. В іншому разі його варто виробляти. Підхід, подібний до цього, називають *маркетингом, що перебуває під контролем споживача*, оскільки популярність продукту або його потенційний успіх визначається заздалегідь циклом споживчого зворотного зв'язку у великому масштабі. Окрім того, потенційні клієнти можуть перевірити, як продукт компанії виглядатиме в їхніх будинках, наприклад, змінюючи кольори меблів.

Висновки

Розглянуто різноманіття проблем, які є сьогодні реалізацією величезних можливостей Всесвітньої Павутини та різноманіття напрямків, за якими триває розвиток, прогрес. Основна увага стосувалася різних бізнес-моделей для Всесвітньої Павутини, зокрема реклами, підписки, інших засобів та їх взаємодії з такими парадигмами, як «програмне забезпечення як сервіс», власність даних і багато аспектів соціалізації.

За попередні роки бізнес-моделі, які розглядалися вище, одна за однією були перенесені або застосовані у Всесвітній Павутині. Сьогодні в Інтернеті спостерігається загальне використання різних *комбінацій* бізнес-моделей. Наприклад, у блогах можна поєднувати співтовариство і брокерські моделі, оскільки таке поєднання дає зацікавленим співтовариствам змогу обмінюватися своїми міркуваннями і здійснювати операції купівлі чи продажу товарів або надавати послуги. Парадигма SaaS дає можливість поєднання співтовариств і підписних моделей, соціальне програмне забезпечення може кваліфікуватися як злиття операцій, реклами та моделі співтовариства.

Через різні механізми та моделі розвитку, які розглядалися вище, соціальні *мережі* перетворюються на великомасштабні мережі *соціального значення*, і наші традиційні економіки стають *економіками взаємин*. Як зазначають аналітики, ЗМІ, блоггери, лише сьогодні починаються наукові дослідження взаємодії соціального та технічного вимірів Всесвітньої Павутини.

Тоді як важливість традиційних ЗМІ зменшується, кількість блогів у світі подвоюється кожні півроку; Всесвітня Павутина стала місцем, де користувачі ефективно контролюють вміст, бізнес-структури шукають нові шляхи впливу та намагаються скористатися цим явищем у своїх бізнес-інтересах. Очевидно, що тут виникають і юридичні проблеми, які необхідно досліджувати, зокрема, управління та контроль над вмістом, конфіденційність інформації, проблеми приватного життя і авторського права, але з'являється значно більше можливостей для ефективного розвитку. Зрозуміло, що крім позитивних моментів, є й зворотний бік медалі – негативні моменти, такі як ненавмисні чи несанкціоновані втручання. У цьому контексті варто згадати повідомлення, зроблене журналом *the New Scientist*, що Агентство національної безпеки Пентагону, яке спеціалізується на підслуховуванні та *кодовому зламі*, вкладає значні кошти у дослідження масового обміну інформацією, яку люди подають про себе у соціальних мережах". У Мережі поширюються різноманітні протиправні дії, такі як спам, фішинг тощо, отже, користувачам доводиться захищати себе від цього.

1. Michael Rappa. *Business Models On The Web*. – <http://www.digitalenterprise.org/models/models.html>. 2. Allan Afuah, Christopher Tucci. *Internet Business Models And Strategies: Text And Cases*. – New York University. – 2003. – McGraw-Hill Higher Education. 3. John Musser, Tim O'Reilly. *Web 2.0 Report*.- ISBN: 0596527691. 4. Paul Sloan, Paul Kahl. *Blogging For Big Bucks*. – *Business 2.0 Magazine*. <http://money.cnn.com/2006/08/21/Technology/Bloggingdollars0822.Biz2/index.htm>. 5. Alex Iskold. *The Attention Economy: An Overview*. – *Technology Blog*. – <http://Alexiskold.Wordpress.com/2007/03/02/The-Attention-Economy-An-Overview/>. 6. Richard Macmanus. *Read/Write Web*. – www.ReadWriteWeb.Com. 7. Tim O'Reilly. *What Is Web 2.0: Design Patterns And Business Models For The Next Generation Of Software*. -<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-Web-20.html>