

ДО ПИТАННЯ ПРОБЛЕМ ПРОЕКТУВАННЯ СУЧАСНИХ WEB-БАЗОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

За минулі десять років бізнес змінився більш ніж за попередні п'ятдесят. Жодна сучасна компанія не може розраховувати на успішність власного бізнесу, якщо вона ігнорує інформаційні технології. Осяг інформації, породжуваної компанією в процесі своєї діяльності і використовуваної нею при ухваленні рішень, нераціонально обробляти вручну. Як результат, інформаційні технології лягли в основу виконання бізнес-процесів практично всіх відділів юридичної компанії, посівши значне місце в її життєдіяльності.

З погляду архітектури, сучасні інтернет-орієнтовані інформаційні системи (далі – ІС) є частиною розподіленої системи. Сильною стороною web-систем є парадигма «все навколо – це документи». Ця концепція розглядає всю сукупність web-елементів як деяку розподілену інформаційну систему з архітектурою «клієнт-сервер». Клієнт відправляє запит документа на сервер і абсолютно не переймається тим, де зберігаються документи і як вони обробляються. Крім того, додатки, створені для web-середовища, мають глобальною доступністю для користувачів. Оскільки web-архітектура була спроектована як незалежна, вона має значний потенціал щодо істотного скорочення витрат на розгортання додатків і навчання персоналу. В даний час багато організацій швидкими темпами створюють нові або удосконалюють старі додатки з метою використання переваг web-технологій.

Успіх web-технологій пов'язаний в основному з їх простотою. Вони дозволяють без істотних обмежень отримувати, використовувати і посилатися на географічно розподілену інформацію. Крім того, ці технології гарантують користувачам можливість одноманітності під

час роботи з документами, незалежно від використовуваного апаратного забезпечення.

Сьогодні багато web-додатків побудовано на основі файлових систем, у яких кожен документ зберігається в окремому файлі. Для невеликих додатків такий підхід цілком виправданий, але для великих web-додатків подібне рішення істотно ускладнює процеси управління даними. Одними з основних проблем великих програмних додатків є своєчасне оновлення версій усіх документів і підтримка актуальності зв'язків між ними.

Вирішення вказаних проблем можливе шляхом інтеграції в web-середовище технологій баз даних (далі – БД) як основних сховищ інформації (варіант збереження інформації у файлах може повністю не виключатися). Незважаючи на те, що web-середовище є розподіленим, для організації зберігання даних використовуються як розподілені, так і локальні БД. Це пов'язано з тим, що завдання побудови корпоративних ІС перебувають іншого підходу й інших методів. Підходи, які добре себе зарекомендували під час вирішення завдань малої автоматизації, можуть бути абсолютно неприйнятними при перенесенні на розподілене середовище. Особливо це актуально при організації роботи з БД як джерелом інформації для користувачів.

У статті буде розглянуто підхід до побудови розподіленої бази даних у рамках створення корпоративної ІС юридичної компанії.

Аналіз проблем створення інформаційного забезпечення web-базованих ІС. Під час розробки інформаційних систем перед розробником постають проблеми [1], які можна умовно розділити на дві групи: проблеми, пов'язані з технологією, і проблеми, пов'язані

з організаційними, законодавчими та іншими особливостями, які властиві замовникові. При побудові корпоративних інформаційних мереж, як правило, використовується одна з двох базових архітектур: клієнт-сервер або інтернет/інтранет. Однією з найпоширеніших на сьогодні архітектур побудови корпоративних інформаційних систем є архітектура «клієнт-сервер». Існує два варіанти цієї технології: дво- і триланкова, яка є розвитком дволанкової і застосовується при створенні складних ІС. У реалізованій за даною архітектурою (клієнт-сервер) інформаційній мережі клієнтам надається широкий спектр додатків та інструментів розробки, які орієнтовані на максимальне використання обчислювальних можливостей клієнтських робочих місць, які використовують ресурси сервера в основному для зберігання й обміну документами, а також для виходу в зовнішнє середовище. Для тих програмних систем, які мають розподілення на клієнтську і серверну частини, використання даної архітектури дозволяє краще захистити серверну частину додатків, при цьому надаючи можливість додаткам або безпосередньо звертатися до інших серверних програмних продуктів, або маршрутизувати запити до них. Засобом (інструментарієм) для реалізації клієнтських модулів, як правило, є різні середовища RAD (Rapid Application Development).

Розподілений характер побудови системи обумовлює її налаштування і супровід. Чим складніша структура мережі, яка побудована за архітектурою «клієнт-сервер», тим вища ймовірність відмови будь-якого з її компонентів.

Останнім часом дедалі більший розвиток отримує архітектура інтернет/інтранет. В основі реалізації корпоративних інформаційних систем на базі даної архітектури лежить принцип «відкритої архітектури», що багато в чому визначає незалежність реалізації корпоративної системи від конкретного виробника. Все програмне забезпечення таких систем реалізується у вигляді аплетів або сервлетів (програм, написаних на java-подібній мові) або у вигляді cgi-модулів.

У загальному випадку ІС, реалізована з використанням даної архітектури, включає web-вузли з інтерактивним інформаційним наповненням, яке реалізовано, наприклад, за допомогою технологій Java, JavaBeans і JavaScript. Це інформаційне наповнення взаємодіє з предметною базою даних, з одного боку, і з клієнтським місцем – з іншого. База даних, у свою чергу, є джерелом інформації для інтерактивних додатків реального часу.

При сучасному розвитку інформаційних технологій, відносній дешевизні обслуговування ІС і низькій вартості передачі одиниці інформації стала можливою побудова ІС, що використовують розподілені бази даних замість локальних.

Аналіз інструментальних засобів, які використовуються під час проектування web-базованих ІС. Життєвий цикл інформаційної системи в загальному випадку [2] починається в момент ухвалення рішення про її створення і закінчується в момент виведення її з експлуатації. Основними його етапами (якщо опустити деталі) зазвичай є:

- проведення передпроектного обстеження;
- проектування даних;
- розробка додатків;
- впровадження створеної інформаційної системи і навчання користувачів;
- експлуатація і супровід;
- виведення з експлуатації і утилізація.

На етапі передпроектного обстеження здійснюються аналіз і моделювання бізнес-процесів, що підлягають автоматизації (іноді цей процес називається структурним моделюванням), а також формулюються вимоги до майбутнього продукту. Нерідко на цьому ж етапі проводиться вибір СУБД та інструментальних засобів. Звичайне подібне обстеження проводиться за участю потенційних користувачів.

Інструментальні засоби, призначені для моделювання інформаційних систем, можуть бути віднесені до однієї з наступних категорій [3]:

- локальні, такі, що підтримують один-два типи моделей і методів (Design/IDEF, ProCap, S-Designer, «CASE. Аналітик»);
- малі інтегровані засоби моделювання, які підтримують декілька типів моделей і методів (ERwin, BPwin);
- середні інтегровані засоби моделювання, що підтримують від 4 до 10-15 типів моделей і методів (Rational Rose, Paradigm Plus, Designer/2000);
- великі інтегровані засоби моделювання, що підтримують більше 15 типів моделей і методів (ARIS Toolset).

Серед локальних і малих інструментальних засобів дуже популярними залишаються програми, які базуються на реалізації структурного підходу до аналізу й проектування систем та методологій Icam DEFinition або Integrated Definition (IDEF). Це сукупність методик, розроблених в США за програмою комп'ютеризації промисловості ICAM (Integrated Computer-Aided Manufacturing). Незважаючи

на відносно великий вік, напрямок IDEF розвивається й сьогодні, правда, в основному, в США. На сайті Knowledge Based Systems, Inc. (www.kbsi.com) міститься інформація про методології IDEF0, IDEF1, IDEF1X, IDEF3, IDEF4, IDEF5, IDEF6, IDEF8, IDEF9, IDEF14 та інструментальних засобах їх підтримки (AIO WIN, SMARTER, ProCap тощо). Усі вони належать до категорії локальних інструментальних засобів.

Процес проектування даних можна умовно розділити на два етапи: логічне моделювання та фізичне проектування [4]. Результатом першого з них є так звана логічна (або концептуальна) модель даних, що виражається зазвичай діаграмою «сутність-зв'язок», або ER (Entity-Relationship)-діаграмою, яка представлена в одній зі стандартних нотацій, прийнятих для відображення подібних діаграм. Результатом другого етапу є готова база даних або DDL-скрипт для її створення.

Логічна модель даних описує факти й об'єкти, що підлягають реєстрації в майбутній базі даних. Основними компонентами такої моделі є сутність, їх атрибути і зв'язки між ними. Як правило, фізичним аналогом сутності в майбутній базі даних є таблиця, а фізичним аналогом атрибута – поле цієї таблиці.

Із логічної точки зору сутністю є сукупність однотипних об'єктів або фактів. Фізичним аналогом екземпляра зазвичай є запис у таблиці бази даних. Як і записи в таблиці реляційної СУБД, екземпляри сутності повинні бути унікальними, тобто повний набір значень їх атрибутів не повинен дублюватися. Так само, як і поля в таблиці, атрибути можуть бути ключовими та неключовими.

Переважна більшість засобів проектування даних дозволяє створювати ER-діаграми візуально, зображаючи сутність і сполучаючи їх зв'язками за допомогою миші. Інтерфейс таких засобів нерідко настільки простий, що дозволяє освоїти логічне проектування даних не тільки розробникові, але й користувачеві-непрограмістові, якщо він бере участь у проектуванні даних як експерт у предметній галузі.

Як правило, сучасні засоби проектування даних підтримують декілька типів СУБД (наприклад, ERwin фірми Computer Associates підтримує більше 20 різних СУБД). Рівень підтримки тієї або іншої платформи в різних засобах проектування даних може відрізнятися. Наприклад, конкретний засіб може підтримувати або не підтримувати для даної СУБД такі властивості, як створення процедур, генерація об'єктів фізичної пам'яті (таб-

личних просторів, сегментів відкату тощо), завдання місцезнаходження об'єктів бази даних у фізичних об'єктах тощо. Тому, обираючи засіб проектування даних для вирішення конкретного завдання, варто звернути увагу на його можливості з погляду підтримки особливостей тієї або іншої платформи. У разі вдалого вибору можна заощадити чимало часу на «ручному» доведенні створеної бази даних (або DDL-скрипта для її генерації) до необхідного стану. При цьому, природно, чим більше можливостей і платформ підтримує конкретний засіб проектування даних, тим дорожче він коштує (слід зазначити, що CASE-засоби взагалі належать до достатньо дорогих програмних продуктів: ціни на них становлять від однієї до декількох десятків тисяч доларів).

Модифікована технологія побудови канонічної моделі розподіленої бази даних. У ході проведеного дослідження було встановлено, що представлена запропонована модель і технологія побудови РБД не враховує такої важливої в сучасних умовах інформації, як рівень критичності даних, рівень використання даних в ієрархії управління й актуальності інформації, що зберігається.

Пропонуємо додатково внести таку інформацію до моделі РБД:

– критичність даних – відносна значущість представлених у базі даних для кінцевих споживачів. Шкала значень формується проектувальником РБД за сукупністю таких чинників, як рівні адміністративної ієрархії користувачів БД, загальна кількість користувачів, кількість вузлів РБД і користувачів цих вузлів;

– рівень використання даних в ієрархії управління – інформація про те, на якому рівні в ієрархії управління компанією дані є найбільш затребуваними. Кількість рівнів використання даних не може бути більшою за кількість рівнів управління, які виділені на стадії аналізу предметної області. Можна допустити об'єднання декількох рівнів ієрархії в один логічний, якщо це не суперечить характеру використання даних;

Таким чином, пропонуємо модифікувати технологію побудови канонічної структури РБД таким чином:

– додати вектор оцінок критичності інформаційних елементів D_k k -го користувача

$$C = \{c_l | l = \overline{1, L}\};$$

– додати вектор, що визначає s -й рівень організаційної ієрархії k -го користувача до інформаційних елементів безлічі D_k ;

Нижче наводяться елементи модифікованої технології побудови канонічної структури РБД.

Етап аналізу предметних областей користувачів і побудова зовнішніх моделей:

1) безлічі D_k ставиться у відповідність до вектора $C_k = \{c_{ki}\}$ і $S_k = \{s_{ki}\}$;

2) оперативність отримання інформаційного елемента за запитом користувача задається в зовнішній моделі у вигляді набору векторів для кожного елемента безлічі S_k , де τ_{ki}^{ξ} визначає мінімально необхідний час (у сек., хв., год. тощо), протягом якого після ініціації запиту до РБД система повинна надати користувачеві s -го рівня запитувані ним дані.

У матриці суміжності M по черзі фіксується рядок (або стовпчик) для кожного i -го ненульового інформаційного елемента векторів $\xi_k^{\xi} = \{\xi_{ki}^{\xi}\}$ та $\xi_k^k = \{\xi_{ki}^k\}$. Потім для елемента m_{ij} зафіксованого стовпчика (рядка) матриці M визначаються значення відповідних йому елементів векторів $C_k = \{c_{ki}\}$ та $S_k = \{s_{ki}\}$. Якщо значення c_{ki} менше за деяке c_0 , а значення s_{ki} менше за деяке s_0 , то в рядку (стовпчику) матриці M знаходять усі елементи, з якими елемент m_{ij} має взаємозв'язок, і якщо значення c_{ki} і s_{ki} для цих елементів менше за деякі c_1 і s_1 , то у векторах ξ_{ki}^{ξ} і ξ_{ki}^k елементи,

відповідні m_{ij} набувають ненульового значення.

Висновки. Таким чином, сучасний бізнес висуває підвищені вимоги до якості роботи інформаційних систем. Особливо це стосується обробки, зберігання і доступу до інформації. Сучасна динаміка бізнесу вимагає швидкого реагування на зміни ситуації на ринку, ухвалення точних і максимально обґрунтованих рішень. Крім того, особливістю є необхідність створення віддалених робочих місць, доступу до територіально розподіленої інформації та управління версіями великої кількості документів. Отже, для вирішення описаних завдань доцільно застосовувати розподілені бази даних.

На сучасному етапі розвитку інформаційних технологій класичні методи проектування БД не дозволяють спроектувати БД, яка б повною мірою могла задовольнити постійно зростаючі вимоги безлічі користувачів, що працюють віддалено одночасно з величезними обсягами зберігаємої інформації. Даний підхід не враховує ряд важливих аспектів зберігання й використання розподілених даних кінцевими користувачами, характерних для web-базованих інформаційних систем, які використовуються в компаніях із розвиненою багаторівневою ієрархічною оргструктурою.

Література

1. Проблемы и перспективы создания крупных корпоративных информационных систем с точки зрения ИТ-службы заказчика [Электронный ресурс]. – Режим доступа : <http://www.pmpofy.ru/content/rus/145/1455-article.asp>.
2. Введение в базы данных: Средства проектирования данных [Электронный ресурс]. – Режим доступа : <http://mf.grsu.by/other/lib/db/part8.html>.
3. Моделирование бизнеса: средства и методы [Электронный ресурс]. – Режим доступа : http://www.iteam.ru/publications/it/section_51/article_1133.
4. Средства проектирования данных [Электронный ресурс]. – Режим доступа : <http://www.interface.ru/misc/sredprdn1.htm>.

Надійшла до редколегії 20.02.2010

Анотації

Розглянуто методи, підходи, моделі та інструментальні засоби, що застосовуються під час розробки баз даних web-базованих інформаційних систем. Розглянуто загальну процедуру створення побудови розподіленої бази даних (РБД) та деталізована процедура побудови канонічної структури РБД.

Рассмотрены методы, подходы, модели и инструментальные средства, которые применяются при разработке баз данных web-ориентированных информационных систем (ИС). Рассмотрены общая процедура создания распределенной базы данных (РБД) и детализированная процедура построения канонической структуры РБД.

Methods, approaches, models and tools which are used at elaboration of databases web-oriented informational systems (IS) are considered. Common procedure of the distributed database (DDB) creation and detailed procedure of the construction canonical structure of DDB are researched.