

УДК 004.415(045)

**ОПТИМІЗАЦІЯ ПРОЦЕСІВ ПЕРЕДАВАННЯ ДАНИХ МІЖ СИСТЕМАМИ,
РОЗРОБЛЕНИМИ НА ПЛАТФОРМІ ІС : ПІДПРИЄМСТВО 8*****Литвинов В.В., **Дмитраш О.В., **Хоменко А.В.**

*Інститут проблем математичних машин і систем НАНУ

**Національний авіаційний університет

litvinov@mmsp.kiev.ua

У статті розглянуто існуючі методи передавання даних між системами, розробленими на платформі ІС: Підприємство 8(система ІС). Сьогодні кожне підприємство має вести податковий, управлінський, бухгалтерський обліки (багато підприємств переважає система ІС, які чудово адаптовані під нашу податкову систему). З бурхливим розвитком Інтернет власний сайт — це не розкіш, а необхідність. Дуже зручно мати змогу виконувати обмін інформацією між системою ІС та web-сайтом. Оптимізація передавання даних є дуже важливою справою.

In this article existent methods are described transmissions given between the systems developed on the platform of IC: Enterprise 8(system of IC). Today every enterprise must conduct a tax, administrative, book-keeping accounts (a lot of enterprises give advantage the systems of IC, which are remarkably adapted under our tax system). With stormy development the Internet an own site it a not luxury, but necessity. Very comfortably to be in a position to execute an exchange information between the system of IC and web-site. Optimization transmissions given is the matter of great importance.

Вступ

З усе більшим поширенням систем ІС та web-сайтів виникає необхідність оптимізації передавання даних між ними. Потрібно забезпечити синхронізацію масивів даних у реальному часі. Масивами даних є довідник «Номенклатура» та журнал документів «Замовлення покупців». Обмін даними повинен здійснюватися без файлів посередників (*.xls, *.xml, *.dbf). Для оптимізації передавання довідника «Номенклатура» необхідно реалізувати підсистему реєстрації змін (ПРЗ), щоб передавати лише нові або змінені елементи.

Проблема передавання інформації між системами ІС розглядалася такими вченими, як М. Радченко, А. Михайлов. Також основні методи передавання даних описуються для ІС: Підприємство 7.7. та ІС: Підприємство 8.0 [1].

У цій статті запропоновано передавання даних з використанням технології COM і ACTIVEX, OLE Automation, ADO, SQL-DMO, електронну пошту, об'єкт FTPСоединение, http з'єднання та ін.

Мета — забезпечити оптимальну передавання даних між системою розробленою на платформі ІС: Підприємство 8 та web-сайтом (інформація зберігається на MySQL) у реальному часі.

Оптимізація передавання даних один із найпоширеніших напрямів розвитку сучасної теорії інформації, пов'язаний із розвитком інформаційних технологій. За будь-якого передавання даних необхідно враховувати специфіку обміну: секретність інформації, її актуальність в

програми будуються з компонентів, які, у свою чергу, складаються з об'єктів. Ці компоненти і об'єкти можуть бути безпосередньо виконуваним двійковим кодом або файлами (DLL, EXE), які ніяк не треба «пов'язувати» з проектом, для якого вони побудовані. Їх досить зареєструвати в операційній системі, і вони будуть доступні будь-якій програмі, що виконується на даній машині. Таким чином, використання їх у програмі проводиться без застосування операцій збірки модуля. Більше того, модель COM дає змогу викликати об'єкти якого-небудь компоненту зі своєї програми без вказівки того, де вони розташовані. Тут достатньо знати тільки ім'я об'єкта.

Технологія ACTIVEX побудована на основі компонентів COM. ACTIVEX — це технологія, розроблена фірмою Microsoft з метою стандартизації програмних компонент. Це системна технологія, що являє собою сукупність засобів, за допомогою яких об'єкти, розроблені різними розробниками на різних мовах програмування і які працюють у різних середовищах, можуть взаємодіяти один з одним без будь-якої модифікації їх виконуваних модулів (двійкових код).

ACTIVEX дає програмістам набори стандартних бібліотек, що значно полегшують процес кодування. Якщо раніше при написанні програм використовувалися механізми OLE (OLE Auto-mation, OLE Documents, OLE Controls), засновані на компонентній об'єктній моделі (COM — Component Object Model), то тепер бібліотеки OLE переписані так, щоб забезпечувати функціональність, достатню для написання мережних застосувань.

Таким чином, тепер при написанні програм використовується DCOM (Distributed Component Object Model) — розподілена компонентна об'єктна модель, а реалізують її бібліотеки ACTIVEX, які за об'ємом стали значно меншими, ніж бібліотеки OLE, а за швидкістю — швидшими. Збереглася і сумісність — будь-який

© В.В. Литвинов, О.В. Дмитраш, А.В. Хоменко, 2009

разі затримки надходження. Розглянемо детальніше основні методи передавання даних. Абревіатура COM розшифровується досить просто. Component Object Model (компонентна об'єктна модель). Іноді говорять — «модель COM».

Побудова компонентної об'єктної моделі здійснюється за допомогою відповідної інформаційної технології. Суть її полягає в тому, що

програмний компонент OLE працюватиме з бібліотеками ACTIVEX.

OLE — це абревіатура назви технології *Object Linking and Embedding* (скріплення і впровадження об'єктів). Вона вказує на здатність працювати з складеними об'єктами, створеними в інших додатках (наприклад, малюнками, документами і т. ін.). Основні терміни, з якими оперує дана технологія, — це OLE-об'єкт, сервер додатка і контейнер додатка.

OLE-об'єктом називають об'єкт, що створений в іншому додатку і зберіг зв'язок із цим додатком. Точковий малюнок, створений у редакторі *Paint*, електронні таблиці у форматі *Excel* або діаграма з *MS Graph* — усі вони можуть бути OLE-об'єктами, якщо будуть вставлені в документ відповідним чином. Якщо не вставляти їх як OLE-об'єкт, то зв'язку з оригінальним додатком не буде.

Контейнером додатку OLE називають додаток, у якому створюється складений документ, що містить OLE-об'єкт, дозволяючи обробляти його в початковому додатку (наприклад, такому як *Paint* або *Excel*), яке використовувалося для створення цього об'єкта.

Сервером додатку OLE (*OLE Server Application*) називають додаток, що створює об'єкти, які можна помістити в документ контейнер. Програми — «1С: Підприємство», *Microsoft Word* і *Excel* є додатками, які можуть виступати і як OLE-сервер, і як OLE-контейнер. Іншими словами, ці додатки можуть створювати нові OLE-об'єкти, а також зберігати OLE-об'єкти, створені в інших додатках. Спочатку механізм OLE був задуманий як технологія інтеграції програмних продуктів, що входять у комплект *Microsoft Office*. Перше втілення OLE (OLE 1) було механізмом розробки і роботи зі складеними документами (*compound documents*).

З погляду користувача складений документ виглядає єдиним набором інформації, але фактично містить елементи, створені двома або кількома різними додатками. За допомогою OLE 1 користувач міг, наприклад, об'єднати електронну таблицю, створену *Microsoft Excel* з текстовим документом *Microsoft Word*. Ідея полягала в тому, щоб документноорієнтована (*document-centric*) модель роботи з комп'ютером дозволила б користувачеві більш думати про інформацію і менш про додатки, що її обробляють. Як впливає зі слів «скріплення і впровадження», складені документи можна створити, або зв'язавши їх, або повністю упровадивши один документ в інший.

ADO (*Microsoft ACTIVEX Data Objects*) — це набір бібліотек, що містять COM-ОБ'ЄКТИ. Основне їх призначення — реалізація прикладного програмного інтерфейсу для доступу до баз даних і використання в клієнтських додатках. ADO використовує бібліотеки OLE DB, що надають низькорівневий інтерфейс для доступу до даних. OLE DB, зі свого боку, забезпечують доступ до даних за допомогою COM-інтерфейсів.

OLE DB і ADO є частиною універсального механізму доступу до даних *Microsoft (Microsoft*

Universal Data Access), який забезпечує високопродуктивний доступ до різних джерел інформації (включаючи реляційні і нереляційні бази даних), зокрема до даних, що зберігаються на мейнфреймах, даним електронної пошти і файлової системи, текстових, графічних. Для багатьох сучасних додатків, які використовують різну інформацію, характерна подібна різноманітність її джерел. Більш того, цілком очевидно, що можуть з'явитися нові формати даних і способи їх зберігання. Тому розумною вимогою до універсального механізму доступу до них могла б стати можливість підтримки не тільки форматів, що існують у даний час, а й форматів даних, які будуть створені в майбутньому. Призначення універсального механізму доступу до даних фірми *Microsoft* — надати доступ до перерахованих джерел за допомогою єдиної моделі доступу.

Розглянемо основні компоненти архітектури універсального механізму доступу до даних *Microsoft* і обговоримо їх детальніше.

Microsoft ACTIVEX Data Objects (ADO) — програмний інтерфейс для доступу до даних з додатків. З погляду програмування, ADO і його розширення є спрощеним високорівневим об'єктноорієнтованим інтерфейсом до OLE DB.

OLE DB — це низькорівневий інтерфейс для доступу до даних. ADO застосовує OLE DB, але можна використовувати OLE DB і безпосередньо, минувши ADO.

Open Database Connectivity (ODBC) — стандартний спосіб доступу до реляційних баз даних. Цей компонент універсального механізму доступу залишений з метою забезпечення сумісності з колишніми версіями програмного забезпечення. У сучасних додатках застосуванню ODBC-драйверів віддають перевагу над використанням OLE DB-провайдерів. Для доступу до джерела даних за допомогою OLE DB потрібно, щоб на комп'ютері, де використовується клієнтський додаток, був встановлений OLE DB-провайдер для даної СУБД. OLE DB-провайдер є динамічною бібліотекою DLL, що завантажується в адресний простір клієнтського додатка і використовується для доступу до джерела даних. Для кожного типу СУБД потрібний власний DB-провайдер OLE, оскільки провайдери базуються на клієнтських API (*Application Programming Interface*, програмний інтерфейс додатка) функціях, різних для різних СУБД.

ADO є високорівневим програмним інтерфейсом для доступу до OLE DB-інтерфейсам. Він дає змогу маніпулювати даними за допомогою будь-яких OLE DB-провайдерів, що входять як до складу *Microsoft Data Access Components*, так і до складу програмних продуктів, розроблених сторонніми виробниками. ADO містить набір об'єктів, що використовуються для з'єднання з джерелом інформації з метою читання, додавання, видалення або модифікації даних, що їм надаються.

Об'єкт *connection* застосовується для установавлення зв'язку з джерелом даних. Він являє

собою єдину сесію. Цей об'єкт дає можливість змінити параметри з'єднання з базою даних, а також почати або завершити транзакцію. Використовуючи об'єкт *connection*, можна виконувати команди (наприклад, SQL-запити) за допомогою методу *Execute*. Якщо команда повертає набір даних, то автоматично створюється об'єкт *Recordset*, який повертається в результаті виконання цього методу.

Об'єкт *Error* використовується для отримання відомостей про помилки, що виникають у процесі виконання додатка.

Об'єкт *command* є командою, яку можна виконати в джерелі даних. Команда може містити SQL-речення або виклик процедури, що зберігається. В останньому випадку для визначення параметрів процедури може бути використана колекція *Parameters* об'єкта *Command*.

Об'єкт *Recordset* — це набір записів, отриманих з джерела даних, і може бути використаний для додавання, видалення, зміни, переглядання записів. Даний об'єкт може бути відкритий безпосередньо або створений за допомогою об'єктів *Connection* або *Command*.

Об'єкт *Field* — це колонка в наборі даних, представлених об'єктом *Recordset*. Об'єкт може бути використаний для набуття значень конкретного поля таблиці, його модифікації, витягання метаданих, таких як ім'я колонки і тип даних.

Об'єкт *Record* — представляє один запис усередині об'єкта *Recordset* і може бути використаний для роботи з гетерогенними і ієрархічними даними (доданий у бібліотеку ADO 2.5, що є складовою частиною операційної системи Windows 2000).

Об'єкт *stream* представляє двійкові дані, пов'язані з об'єктом *Record*. Наприклад, якщо об'єктом *Record* є файл, то його об'єкт *stream* повинен містити дані усередині цього файла (доданий в ADO 2.5).

Connection — найважливіший об'єкт ADO, що відповідає за зв'язок між додатком і базою даних. У нього багато методів і властивостей, і одна з них — *connectionstring*. Вона визначає власне рядок ініціалізації ODBC-з'єднання.

SQL-DMO (SQL *Distributed Management Objects*) — це забезпечувана Microsoft SQL Server об'єктна модель, яка заснована на COM-технології. SQL-DMO приховує деталі структури мови TRANSACT-SQL і використовується для написання адміністративних додатків і сценаріїв для *Microsoft SQL Server*. Забезпечувані *Microsoft SQL Server* графічні засоби адміністрування написані із застосуванням іменного SQL-DMO, яка не є моделлю інтерфейсу даних і не застосовується для написання стандартних додатків баз даних. SQL-DMO дає змогу клієнтському додатку маніпулювати такими об'єктами бази даних, як таблиці, процедури і властивості сервера через інтерфейс COM. Модель надає додатку такі можливості:

- управління таблицями (перегляд, створення, видалення, модифікація);
- управління списком доступних серверів;
- управління правами доступу до серверів;
- управління списком баз даних на кожному сервері;
- управління сценаріями таблиць бази даних;
- створення SQL-скриптів (таблиць, уявлень, процедур, користувачів та їхніх ролей);
- виконання SQL-запитів.

Для того, щоб об'єкти SQL-DMO можна було використовувати на конкретному комп'ютері, необхідно встановити на ньому клієнтську частину *Microsoft SQL Server*, який встановлює бібліотеку *Microsoft SQL OLE Object Library*. Дана бібліотека є ActiveX-інтерфейс до об'єктів SQL-DMO. За допомогою SQL-DMO можна отримувати дані з таблиць *SQL Server*.

Особливо ефективно дану модель можуть застосовувати розробники, використовуючи *Microsoft SQL Server Desktop Engine (MSDE)*, який постачається разом з Access 2003. Цим сервером баз даних є проміжний варіант *SQL Server* і був створений спеціально для сполучення з ранішими версіями *SQL Server*. Хоча MSDE володіє меншими можливостями, ніж *SQL Server*, він постачається безкоштовно у складі *Microsoft Office 2003*. Проте версія MSDE, яка постачається разом з *Microsoft Office 2003*, не включає *Enterprise Manager*. У результаті користувачі MSDE не можуть оцінити всіх переваг графічного інтерфейсу при управлінні об'єктами бази даних. У зв'язку з бурхливим розвитком інтернет-програмування підтримка сервісу електронної пошти стала вельми поширеним завданням для програмістів.

Щоб ефективно розв'язати задачу доступу до електронної пошти, користувача, в своїй програмі, необхідно зрозуміти основи її будови. Електронна пошта використовує сукупність протоколів, що регламентують обмін даними по мережі, а також програмних засобів, що реалізують ці протоколи і що надають користувачеві відповідні інтерфейси. Основними протоколами є протоколи передавання (відсилання) і читання листів, які працюють над TCP/IP (*Transmission Control Protocol / Internet Protocol* — протокол управління передаванням даних Інтернет). Для передавання повідомлень по TCP-з'єднанню переважна більшість програмних засобів користуються протоколом SMTP (*Simple Mail Transfer Protocol*). Для прийому поштових повідомлень в Інтернеті зазвичай використовується протокол POP3 (*Post Office Protocol 3*).

Безпосереднє використання в конфігураціях системи «ІС: Підприємство» команд протоколів SMTP і POP3 неможливе. У зв'язку з цим для роботи з електронною поштою використовуватимуться засоби більш високого рівня (зовнішні компоненти, додатки і стандартні об'єкти системи).

Абревіатура FTP розшифровується як *File Transfer Protocol* (протокол передавання файлів).

Протокол FTP оптимізований для передавання файлів, це і стало причиною того, що програми, які використовують його, стали частиною окремого сервісу Інтернету. Сервіс, що забезпечує доступ до файлів інших комп'ютерів мережі по протоколу FTP, носить однойменну назву.

Щоб скористатися протоколом FTP, знадобиться спеціальна програма, так званий FTP-клієнт. FTP-клієнтом може служити браузер (*Microsoft Internet Explorer, Netscape Navigator*), програма управління файлами FAR, спеціалізована програма CUTEFTP та ін. Для забезпечення програмного доступу до протоколу FTP можна використовувати або виклик програми ftp.exe, або — об'єкт FTP, що є частиною об'єктної моделі системи «1С: Підприємство». Для роботи з видаленими комп'ютерами через протокол FTP призначений об'єкт FTPСоединение. За допомогою даного об'єкта можна виконувати типові дії з файлами, наприклад, такі як: викачувати і закачувати файли, перейменовувати їх, звертатися до їх властивостей, створювати каталоги. Для встановлення з'єднання з FTP-сервером найпростіше скористатися конструктором об'єкта ftpСоединение, який має такі російськомовні параметри:

- сервер–сервер, з яким здійснюється з'єднання;
- порт–порт сервера, з яким здійснюється з'єднання;
- ім'я користувача–ім'я користувача на вказаному сервері;
- пароль користувача—пароль користувача на вказаному сервері;
- прокси–значення типу Інтернетпрокси (проксі, використовуваний для з'єднання з сервером).

Поле установки з'єднання за допомогою методу «Установить Текущий Каталог» можна вказати поточний каталог сервера, з яким відбуватиметься робота надалі. Далі можна працювати з файлами і каталогами віддаленого комп'ютера. На базі вище розглянутих методів було реалізовано імітаційну модель. ПРЗ реалізована в системі 1С з використанням об'єкта типу «Планы обмена» [2]. Підсистема відслідковує та зберігає посилання на елементи системи, які було змінено, створено або видалено після останнього обміну даними. Такий фільтр підвищує швидкість обміну, зменшуючи кількість інформації, яку необхідно передавати. В даному випадку швидкість збільшується прямо пропорційно до кількості елементів у довіднику «Номенклатура» (товарів).

Обмін у реальному часі спрощено до синхронізації, що відбувається щохвилини. Через указаний проміжок часу база 1С встановлює за допомогою ODBC з'єднання з базою даних MySQL, де зберігається інформація сайту. Підключившись, можна виконувати запити, викликати функції та процедури MySQL з бази 1С.

У базі 1С виконується безпосереднє редагування таблиць MySQL (в цьому разі це таблиці, де зберігаються товари).

Відбувається пошук товару в MySQL, зареєстрований ПРЗ. Якщо такий на знайдено, до таблиці додається запис. Запис можна здійснити прямо з бази 1С, виконавши відповідний запит або викликати PHP функцію і передати їй необхідні параметри. Якщо запис із товаром знайдено, то можна виконати запит на його редагування або викликати PHP функцію, яка це зробить.

Товари мають реквізит «Основное Изображение» й можуть мати необмежену кількість будь-яких файлів. Це можуть бути презентації продукту, галерея з картинок та інше.

Щоб передати такі відносно великі обсяги даних використовується, об'єкт системи 1С FTPСоединение. Встановивши FTP з'єднання, можна передати всі необхідні файли або видалити непотрібні (для товарів, які були видалені з бази системи 1С). Після передавання даних уся інформація з ПРЗ видаляється. Після синхронізації довідника «Номенклатура» відбувається обмін новими замовленнями, створеними на сайті. Оскільки

всі нові замовлення (створені після останнього обміну) зберігаються у відповідних таблицях MySQL, то після виконання одного запиту всі дані про документи можуть опинитися в базі 1С. Там вони будуть опрацьовані належно.

На даній моделі був проведений ряд імітаційних експериментів, направлених на вибір кращого методу передавання даних.

Висновки. У результаті проведення модельних експериментів вдалося отримати оптимальну систему передавання даних між системою 1С та web-сайтом (інформація зберігається на MySQL). Обмін у реальному часі було спрощено до щохвилинного обміну. Модель є актуальною за умови, що в системі 1С працює до 50 користувачів із середньою частотою редагування товарів $\leq 0,1$. Важливим елементом оптимізації є ПРЗ.

З її використанням швидкість збільшується прямо пропорційно до кількості товарів. У різних підприємств кількість позицій «Номенклатури» в базі системи 1С варіює від 500 до 100 000 і навіть більше. При використанні даної моделі, якщо інтенсивність редагування товарів зменшується, то можна частіше виконувати обмін, що наблизить до синхронізації в реальному часі. І навпаки, якщо частота редагування збільшується — необхідно обмін даними робити рідше.

ЛІТЕРАТУРА

1. Михайлов А. В. 1С: Предприятие 7.7/8.0: системное программирование / А. В. Михайлов. — Питер, 2005. — 336 с.

2. Радченко М. Г. 1С: Предприятие 8.1. Практическое пособие разработчика. Примеры и типовые приемы / М. Г. Радченко. — Питер, 2007. — 368 с.