L'interfacciamento

http://www.vbsimple.net/database/db_005.htm

Difficoltà: 2 / 5

L'utilizzo di una fonte dati suppone sempre un accesso diretto o indiretto al database da cui recuperare i dati. L'accesso diretto consiste nel leggere direttamente i files fisici con le istruzioni per la manipolazione di dati binari e lavorare *alla cieca*; si tratta di un'operazione nella maggior parte dei casi inapplicabile, complessa ed in generale insulsa. Per evitare che il programmatore maneggi complessi archivi, delle più disparate tipologie, si utilizza l'accesso indiretto alla fonte dati.

L'accesso indiretto si basa sull'utilizzo di una o più librerie esterne, che si occuperanno autonomamente di effettuare l'accesso diretto alla fonte dati. Il programmatore Visual Basic dovrà limitarsi a stabilire una connessione tra il programma in sviluppo e la fonte dati mediante un'interfaccia dati.

Si definisce **interfaccia dati** un meccanismo, solitamente una libreria, che consente di stabilire una connessione ad una fonte dati esterna e di maneggiare i dati in essa esistenti. Spesso si utilizza interporre un ulteriore livello tra il programma utilizzatore dei dati e l'interfaccia dati; questo ulteriore livello è detto **modello dati**.

Il diagramma mostrato nella Figura 1 rappresenta le quattro forme più semplici di interfacciamento con una fonte dati.

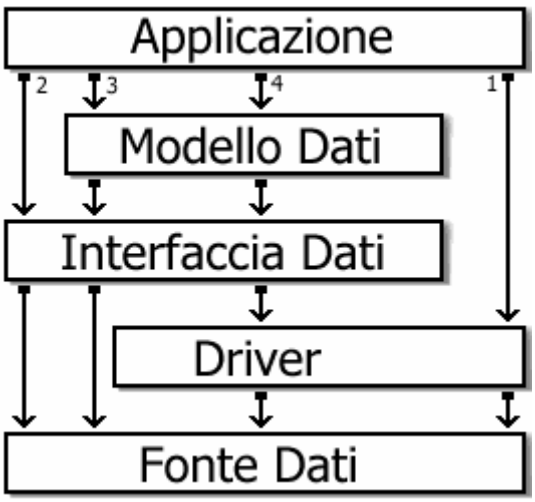

Figura 1

1. **Applicazione - Driver - Fonte Dati**
   La modalità più semplice e leggera consiste nel far utilizzare all'applicazione finale, un driver progettato appositamente per la fonte dati cui si tenta di accedere. Si tratta di una soluzione difficilmente applicabile in Visual Basic e richiede la conoscenza del funzionamento del driver che accederà al database.

2. **Applicazione - Interfaccia Dati - Fonte Dati**
   Per molti versi simile alla modalità precedente ma l'accesso alla fonte dati è effettuato da un'interfaccia dati, molto più generale e flessibile di un driver progettato per il database di destinazione. Anche questa soluzione è generalmente complessa da utilizzare all'interno di Visual Basic.

3. **Applicazione - Modello Dati - Interfaccia Dati - Fonte Dati**
   Il modello di accesso più funzionale, si basa sull'uso di un modello dati in grado di comunicare con un'interfaccia dati risolvendo quindi le difficoltà date dalla soluzione precedente. In ambito Visal Basic si tratta quasi sempre della migliore soluzione, combinando semplicità d'uso e velocità.

4. **Applicazione - Modello Dati - Interfaccia Dati - Driver - Fonte Dati**
   La forma primitiva più complessa, basata sull'uso di tre differenti strati per accedere alla fonte dati di destinazione. Utilizzata per lo più quando non è possibile utizzare il modello precedente oppure è necessario mantenere un elevato grado di scalabilità del prodotto.

Modalità di interfacciamento più complesse implicano l'uso di differenti sistemi e quindi ulteriori strati di trasporto e conversione di dati. Si tratta generalmente di combinazioni delle precedenti modalità ma sono utilizzate in progetti di elevata importanza e non saranno quindi trattate in quest'articolo.

I Modelli di Dati

L'utilizzo di una fonte dati esterna in progetti Visual Basic implica quasi sempre l'utilizzo di un **modello di dati** che consenta di dare una conformazione ad oggetti, basata su proprietà, metodi ed eventi conosciuti e quanto più possibilmente indipendenti dalla fonte dati cui si accede.

I modelli di dati consentono quindi di interrogare la fonte dati, senza dover conoscere i meccanismi interni che ne regolano lettura e scrittura; sarà il modello di dati ad esporre una serie di membri per effettuare tutte le operazioni possibili.

I modelli di dati più conosciuti in ambito Visual Basic sono tre:

1. **Data Access Object** (*DAO*)
   Il modello di dati più vecchio e quasi del tutto abbandonato. Nonostante questo continua ad essere il modello di dati preferito da principanti e nei piccoli progetti.

2. **Remote Data Object** (*RDO*)
   Leggermente più avanzato di DAO, consente anche l'accesso a fonti dati remote, quali database in Internet. In ragione del basso interesse, questo modello di dati non sarà trattato all'interno di questo corso.

3. **ActiveX Data Objects** (*ADO*)
   Il più moderno dei modelli di dati, così piatto da sembrare inutile, si rivela invece il modello più avanzato. Il suo elevato grado di generalizzazione consente l'accesso a

fonti dati sia locali che remote, di qualsiasi tipologia.

La scelta del modello di dati dipende gran parte dalla velocità necessaria o dal tipo di database da gestire. Nella maggior parte dei casi si raccomanda l'uso di *ADO* perché sempre idoneo a qualsiasi fonte dati, veloce ed aggiornato. L'uso dei modelli di dati *DAO* ed *ADO* sarà trattato più avanti in questa sezione.

Le Interfacce Dati

I modelli dati si basano sempre su un'**interfaccia dati** che consenta il recupero dei dati dal database, al fine di modellarli secondo la struttura del modello scelto.

Il modello ***DAO*** è in grado di utilizzare due differenti interfacce dati:

1. **Microsoft Jet**
   Progettato l'accesso a database di tipo Microsoft Access, è in grado di accedere a tante altre fonti dati basate su files ISAM quali dBase, Paradox, FoxPro ed altre.

2. **ODBCDirect**
   Consente l'accesso a fonti dati mediante ODBC (Open DataBase Connectivity).

Il modello ***ADO*** basa invece tutta la sua attività su un'interfaccia dati chiamata **Provider OLE DB**; tali Provider possono essere suddivisi in due tipologie:

1. **Provider OLE DB** normali
   Si defiisconono così tutte quelle interfacce dati che effettuano l'accesso alla fonte dati finale senza passare per ulteriori livelli.

2. **Provider ODBC**
   Un Provider OLE DB particolare, che non effettua l'accesso alla fonte dati direttamente ma richiama i Driver ODBC, i quali svolgono le operazioni sulla fonte dati e restituiscono i risultati al suddetto Provider, il quale li restituisce a sua volta al modello dati ADO.

In linea generale l'accesso mediante provider OLE DB è più veloce dell'accesso tramite ODBC per via di un livello di comunicazione in meno. Tuttavia a causa di alcune problematiche di certi provider OLE DB, la scelta di ODBC a volte si rivela la scelta migliore. In sintesi è bene non fidarsi della pubblicità intorno ad un certo prodotto ma si raccomanda di provare con mano la velocità di entrambi gli accessi e scegliere quello che si rivela più veloce. ODBC tuttavia manca di tantissime caratteristiche presenti nell'interfaccia dati OLE DB.

È anche possibile accedere direttamente all'interfaccia dati senza passare da un modello specifico; si tratta però di un'operazione relativamente complessa e che necessita l'uso dell'API oppure di una Type Library in grado di comunicare con l'interfaccia dati.

I Driver

Il livello di comunicazione con la fonte dati più basso e diretto. In ambiente Visual Basic i driver vengono impiegati nelle comunicazioni *ODBC*. L'interfaccia ODBC infatti non consente l'accesso a nessuna fonte dati senza l'utilizzo di una libreria che effettui le comunicazioni necessarie.

ODBC infatti espone soltanto una serie di funzioni e tipi di dati standard per tutti i database; i singoli driver si occuperanno quindi di effettuare le dovute conversioni e gli accessi alla fonte dati in maniera corretta.

Naturalmente queste operazioni avvengono in maniera invisibile all'utente e spesso in maniera altrettanto invisbile allo sviluppatore del prodotto.

Fonti Dati

Un piccolo appunto prima di concludere: sono stati utilizzati indifferente i termini fonte dati e database. In linea generale i due concetti si equivalgono ma il primo dei due può indicare qualsiasi gruppo di dati come ad esempio matrici di numeri, caselle di posta elettronica e tanto altro ancora.

Il modello di dati ADO si basa proprio su questo concetto; la fonte dati non deve necessariamente essere un database relazionale, ovvero un gruppo di dati relazionati, ma può avere qualsiasi forma e caratteristica; sarà compito del Provider OLE DB riportare queste informazioni in maniera tabellare.

La strada ai database si apre proprio in questo punto.
La migliore soluzione per un progetto non è data soltanto dal database scelto ma dal modello di dati utilizzato, dal tipo di interfaccia scelta e dal numero di strati che si sovrappongono per effettuare l'interfacciamento dell'applicazione con la fonte dati.

Nei capitoli successivi saranno trattati i due modelli più utilizzati (DAO ed ADO) e le succitate interfacce dati per lo sviluppo di un semplice programma di catalogazione di CD-ROM.

Fibia FBI
1 Dicembre 2002

Torna all'indice della sezione Database
