

Visual Studioで 構築エンタープライズ システム

Application
Architecture for .NET
の利用例

第2回 データベースを操作する

株式会社CSK
eソリューション技術部
中垣 健志
NAKAGAKI, Kenji

Level				
1	2	3	4	5

Technology Tools
<input checked="" type="checkbox"/> Visual Basic
<input checked="" type="checkbox"/> Visual C#
<input type="checkbox"/> Visual C++
<input checked="" type="checkbox"/> SQL Server
<input type="checkbox"/> Oracle
<input type="checkbox"/> Access
<input checked="" type="checkbox"/> ASP.NET
<input type="checkbox"/> Other:

Samples
<p>・この記事で取り上げたソースコードおよびサンプルプログラムは、 http://www.shoeisha.com/mag/windev/からダウンロード可能です。</p>

読みづらいプログラム

N君は今、社内のあるプロジェクトを支援するためにいつもより3階上のフロアに来ています。朝からよく晴れていて窓からは富士山がきれいに見えているのですが、目の前のディスプレイには複雑怪奇なSQL文を生成する、やはり複雑怪奇なVisual Basicのプログラムがあります。少し疲れたN君は、一緒に支援に来ているFさんに同意を求めます。

「データベースをC#やVB.NETで扱うのって面倒ですよね」

Fさんは、すかさず答えます。

「そうですねー。なんでこんな読みづらいプログラムになるんですかねー。ありえないですよ。“データベースを操作するプログラムはこうだ”って、決まりがあるべきですよ、ぜっ

たい」

たしかにそのとおりだとN君も思いました。でも、データベースを操作するプログラムの決まりってどういふものなのでしょうか？

データレイヤーで行なうこと

データレイヤーでは、データの永続化を行なうシステムへの統一したアクセス方法を上位レイヤー（今回はビジネスレイヤー）に提供します。

永続化されたデータとは、たとえばそのデータを扱うシステムが停止したとしても変わらず保持されるデータを指します。データが保持される期間は一般的に長く、目安として数時間からときには数年間におよぶこともあります。永続化の仕組みとしてはSQL ServerやOracleなどのリレーショナルデータベースが最も一般的に使われています。

そのほか、既存のホストやWebサービスなども一種の永続化の仕組みとしてみなすことができます。

このようなさまざまな永続化の仕組みがあるにもかかわらず、データレイヤーを参照する上位レイヤーからは違いを意識せずに統一した方法でデータの入出力を行なわせる必要があります。ただしすべての永続化の仕組みを統一しようとする、かえって複雑になり使いづらくなってしまいます。重要なのは“統一のバランス”です。そこで今回は、「データレイヤーの実装はリレーショナルデータベースで行なう」という前提を置いた上で、統一したアクセス方法を提供することにします。

データベースに用意されているオブジェクトは、一般的に以下のようなものが挙げられます。

- ・ テーブル
- ・ ビュー
- ・ ストアドプロシージャ

◆テーブルに対して提供する操作

リレーショナルデータベースでは永続化されるデータを、テーブルに保存します。永続化されたデータに対しては、次に示す4つの操作が実行できなければなりません。これらの操作はSQL (Structured Query Language) という統一された言語を用いて、データベースの種類を問わず行なうことができます。また、これら4つの操作は、その頭文字をとって一般的にCRUD (Create、Retrieve、Update、Delete) と呼ばれます。

① Create (作成) 操作

テーブル内に新しくデータを作成します。SQLではINSERT文がCreate操作を担当します。

② Retrieve (検索) 操作

テーブル内から条件に従ってデータを検索します。SQLではSELECT文がRetrieve操作を担当します。

③ Update (更新) 操作

すでにテーブル内に存在しているデータを更新しま

す。SQLではUPDATE文がUpdate操作を担当します。

④ Delete (削除) 操作

すでにテーブル内に存在しているデータを削除します。SQLではDELETE文がDelete操作を担当します。

Soarアプリケーションでは、利用するテーブルごとにCRUD操作を行なうためのメソッドを持つクラスを「テーブルデータクラス」としてデータ層に新しく定義します。テーブルデータクラスは上位レイヤーに公開されます。

テーブルデータクラスでは、テーブル自身の構造を表わす情報も提供する必要があります。テーブル自身の構造とは主に以下のようなものが挙げられます。

- ・ テーブル名
- ・ 列情報 (型、長さ)
- ・ 制約
- ・ テーブル間の関連

このクラスを設計する上で、Soarアプリケーションでは「データベースへのアクセスに必要な接続を呼び出し元から渡す」という方法を取りました。

テーブルデータクラスは、構造上各種テーブルに対する操作をVB.NET/C#レベルでライブラリ化したものと考えられます。つまり、ひとつのクラスがひとつのテーブルを操作する処理を担っているわけです。また、.NETでのトランザクションはデータベースへの接続を作成した後で開始され、接続を閉じる前に終了します^[注1]。そのため、テーブルデータクラスの中で接続を作成する実装方法は、クラスが操作の対象とするテーブルに対してしかトランザクション処理ができないことになってしまいます。これは通常のビジネスアプリケーションではありえない処理です。したがって、データベースへの接続はあらかじめテーブルデータクラスを利用するクラスで作成しておき、必要があればトランザクションも開始し

注1) Microsoft Transaction Server (MTS) を使わない場合。