

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

する

第9回

データ&ビジネスレイヤーの フレームワーク実装

株式会社CSKシステムズ
IT生産技術部
中垣 健志
NAKAGAKI, Kenji

はじめに

N君の所属する部署は、.NETを使っているプロジェクトに対して技術的な

Level

1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

Samples

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。

手助けを行なっています。近頃にはわかに.NETを使ったプロジェクトが増えているために、今までにも増して忙しくなっています。そんな中で感じるのは、どのプロジェクトでも「どうやって作ればよいかわからない」という点で悩んでいることが多いということです。

.NETと一口に言っても、ASP.NETやADO.NETなど「.NET」と名のつくさまざまな副次的技術が混在しています。それらをどのように組み立てればよいのかは難しい問題です。そもそも数学の問題のように答えがひとつだけではなく、プロジェクトの規模、期間、開発メンバーのスキルなどさまざまな要因からいくつもの解決策が導き出されます。

「フレームワークが提供するアーキテクチャは、その中でも満点ではないが及第点が取れるもの。それもできるだけ多くのプロジェクトで」。そんなフレームワークを作りたい。こんな思いを込めながらN君は、過去の.NETプロジェクトの成果を見直し

ています。

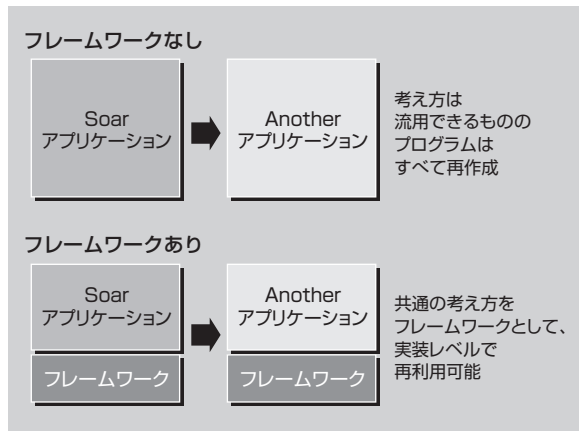
フレームワークを 設計する

Application Architecture for .NET (以下AAfN) では、アプリケーションを3つのレイヤーとひとつの共通コンポーネントに分割して実装します。これまでの連載では、Soarアプリケーションを用いて具体的な設計/実装を行なってきました。このSoarアプリケーションには、アプリケーション固有の実装と、他のアプリケーションでも共有することが可能な実装が混在しています。フレームワークは、このうち後者の部分を抜き出して独立させたものになります。うまく独立させることができれば、図1に示すような再利用が可能となります。

◆テーブルデータクラスの汎化

Soarアプリケーションでは、“テーブルデータクラス”というクラスを新たに定義しました^[註1]。テーブルデー

図1：フレームワークの位置づけ



タクラスが果たすべき役割は、以下のように定義されています。

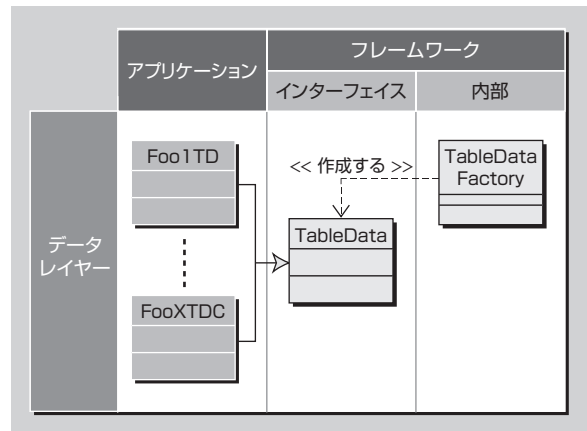
- ・データベースの各テーブルごとに用意する
- ・テーブルへのCRUD操作を行なうメソッドを持つ
- ・テーブル構造を定義している

Soarアプリケーションでは、この定義に沿ってテーブルごとにテーブルデータクラスを作成しています。しかしこのクラスは設計レベルでは共通性があるものの、実装レベルではまったく別物のクラスです。そこでこれらのクラスを、同じ性質を持ったクラスとしてまとめて扱えるようにするために、“汎化”という作業を行ないます。

汎化とは、複数のクラスが持つ共通的な機能をまとめたクラスを新たに作成することです。このとき「汎化されたクラス」はフレームワークで定義されたクラス、「各テーブルを実際に操作する特化されたクラス」は各アプリケーションで定義されるクラスになります。

各テーブルごとのテーブルデータクラスはアプリケーションで定義されるクラスになりますが、テーブルデータクラスをインスタンス化する処理はフレームワーク側に用意する必要があります。なぜならば、フレームワークはアプリケーション内の処理の流れを規定するものであり、その中にはクラスのインスタンス化や破棄といった作業を含んでいるからです。以前、これを実現するテ

図2：テーブルデータクラスを汎化する



クニックとして“リフレクション”という機能を紹介しました^[注2]。

リフレクションを使うと、クラスの名前や型情報を元に間接的にクラスをインスタンス化することができます。そこで、リフレクションを用いてテーブルデータクラスをインスタンス化するクラスを、フレームワークの内部に用意します。このクラスは「テーブルデータを作成する場所」という意味を込めて、“TableDataFactory”^[注3]という名前にします（図2）。

◆ビジネスロジッククラスの汎化

次に、ビジネスレイヤーで登録されているビジネスロジッククラスについて検討していきます。Soarアプリケーションでは、ビジネスレイヤーに属するクラスとして、

- ①ビジネスエンティティクラス
- ②ビジネスコンポーネントクラス

の2つを定義しました^[注4]。

最初のビジネスエンティティクラスは、DataSetクラスやDataTableクラスをラッピングするクラスです。この2つのクラスが持つ基本的な機能を活かしながら、ア

注1) 本誌2005年6月号の本連載（データベースを操作する）を参照。
 注2) 本誌2005年11月号の本連載（フレームワーク化を始める）を参照。
 注3) デザインパターンのひとつ「Factory Method」を意識しています。
 注4) 本誌2005年7月号の本連載（ビジネスロジックを実装する）を参照。