

# 実践! 次世代



# C/S

クライアント/サーバー

# システム

スマートクライアント&Webサービスの実際

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

第4回

## サーバー側の実装を行なう



### スマートクライアントシステムのサーバーサイドアーキテクチャ

本連載で取り上げているスマートクライアントシステムのアーキテクチャでは、VB6時代のクライアント/サーバー型のシステムの形はそのままに、二者の間をつなぐネットワークがLANではなく「インターネット」や「イントラネット」という構成をとっています。そして、先月号で解説したクライアント側のプレゼンテーションレイヤーの実装

では、過去のクライアント/サーバー型システムのノウハウを有効に活用できることを紹介しました。

しかし、サーバー側のビジネスレイヤーやデータレイヤーの実装については、クライアント/サーバー型のシステムに比べて気をつけなければならない点がいくつか出てきます。

そこで今回は、スマートクライアントシステムにおけるサーバー側の実装方法について説明していきます。



### サーバー側で実現するレイヤーの役割と実装方法

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 (プレゼンテーション、ビジネス、データ) を組み合わせて構築することを推奨しています。スマートクライアントアプリケーションではこのうち、画面表示や情報の入出力を実現するプレゼンテーションレイヤーと、ビジネスロジックやデータベース操作を実現するレイヤーが、ネットワークを越えた別のコンピュータ上で動作するのが大きな特徴です (図1)。

サーバー側の2つのレイヤーは、役割に応じてさらに複数のコンポーネントに分割されます。ここでの役割とは、生産管理や販売管理といった業務的な役割ではなく、トランザクションの管理方法やデータベースの操作方法といった技術的な機能から分類

レベル >>> Level

1 2 3 4 5

ツール >>> Tool

- Visual Studio 2005 Professional
- SQL Server 2005

言語 >>> Language

- Visual Basic
- C#

サンプル >>> Sample

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.shoeshisha.com/mag/windev/> からダウンロード可能です。

される役割のことです。そして分類された役割について、.NET Frameworkで用意されているどの技術を使えばよいのかを検討していく必要があります。そこで今回は、各レイヤー内で必要となる役割とそれを実装する技術について説明します。

### ※サービスインターフェイス

ビジネスレイヤーのうち、もっともプレゼンテーションレイヤーに近い場所に「サービスインターフェイス」が用意されています。このサービスインターフェイスは、AAfNでは、

「ビジネスロジックをサービスとして公開するために、さまざまなコンシューマにより要求される通信規約（メッセージベースの通信、データ形式、プロトコル、セキュリティ、例外など）をサポートする場所」

と定義されています。この内容を簡単に言い換えると、

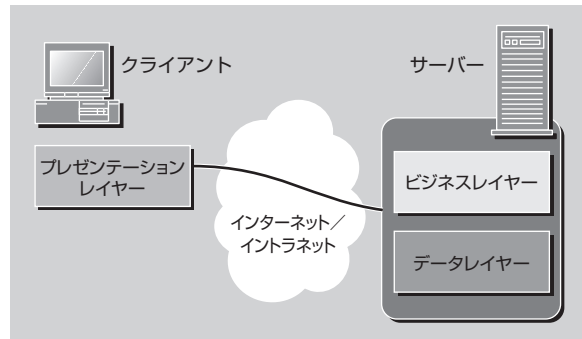
プレゼンテーションレイヤーからビジネスレイヤーを呼び出すときに、どのような呼び出し方法を利用するのかを規定する場所

となります。

スマートクライアント型のアプリケーションにおいては、プレゼンテーションレイヤーとビジネスレイヤーの間はインターネットまたはイントラネットで隔てられています。このネットワークはHTTPによる通信しか認められていません。Microsoftではこの間の通信を、XML Webサービスを用いて行なうよう推奨しています。したがってスマートクライアント型のシステムでは、サービスインターフェイスはWebサービスを用いて実現します。

サービスインターフェイスの定義を読み直すと、ビジネスロジックやデータベース操作を行なう場所とは定義されていません。つまりサービスインターフェイスにはこれらの処理を記述せず、実際のビジネスロジックやデータベース操作を行なう処理を呼び出す“単なる窓口”として振る舞います。これは、デザインパターンに登場

図1：クライアントとサーバーは別のコンピュータ上で動作する



する「Facadeパターン」のわかりやすい実装例となっています (図2)。

窓口としてのもうひとつの役割は、セキュリティと例外です。

アプリケーションで定義したXML Webサービスを呼び出すのは、用意したクライアントアプリケーションだけでは限りません。.NET Frameworkの場合、XML WebサービスはURLによって公開されるため、ブラウザ上で直接URLを入力してXML Webサービスを不正に呼び出す可能性がないとはいえません。そのようなリクエストに対してサーバー内部のビジネスロジックを実行できないようにするために、セキュリティ処理を実装する必要があります。

また、サーバー側で発生した例外に対して、クライアントは適切な処理を行なう必要があります。しかし、クライアントとサーバーの処理は別のコンピュータ上で動作しているため、.NET Frameworkの例外を伝播させることはできません。代わりにクライアント/サーバーの両システムでSoapExceptionを使います。SoapExceptionを使うと、クライアントがサーバー側の例外状況を取得するためのデータ構造は文字ベースとなります。

この処理はシステム全体のセキュリティや例外の扱い方と関係してくるので、次回説明する予定です。

ここでは簡単なサービスインターフェイスの実装例を示します (リスト1)。

### ※ビジネスコンポーネント

サービスインターフェイスは、ビジネスコンポーネン