

.NET流、 業務アプリケーションの 作り方

C/S、Webアプリケーション、データ処理と ビジネスロジックをどのように実装すべきか

大澤 文孝

OSAWA, Fumitaka

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level



Samples

はじめに

.NET Frameworkは、技術的な資料も出揃い、いよいよ本格的な業務アプリケーションを構築する時期に来ていると言えます。

業務アプリケーションは、スタンドアロン、クライアント／サーバー（以下C/S）、Webアプリケーションと、その形態はさまざまです。

また、求められる性能や同時アクセスするユーザー数など、規模によっては、n階層アプリケーションとして構成することも視野に入れる必要があるでしょう。

Win32アプリケーションでは、Windows DNAとして提唱されるアーキテクチャを採用し、ビジネスロジックをCOMコンポーネントとして実装してきました。.NET Frameworkでも同様に、コンポーネント（クラス）を利用して、n階層アプリケーションを構築できます。

.NET Frameworkでは、ビジネスロジックをXML Webサービスとして実装

する方法もとれます。また、すでにCOMコンポーネントとして実装したビジネスロジックを再利用することもできます。

.NET FrameworkアプリケーションでもWin32アプリケーションでも、アプリケーション構築の基本はほとんど同じです。しかし、.NET Frameworkならではの方法や気をつけるべき点もいくつかあります。

そこで本稿では、Win32アプリケーションとの違いや連携を中心に、.NET Frameworkで業務アプリケーションを構築するときに、どのような構成の選択肢があり、どのような点に配慮して設計すべきかを説明します。

Webアプリケーションか Windowsアプリケーションか

近年は、業務アプリケーションをWebアプリケーションとして構築するのが主流になりつつあります。

しかしWebアプリケーションが、常に最良のソリューションとは限らず、クライアント側のアプリケーションを

Windowsアプリケーションとして構築したほうがよい場合もあります。

そこでまず、WebアプリケーションとWindowsアプリケーションのそれぞれの特徴をまとめてみましょう。

Webアプリケーションの場合

Webアプリケーションの最大のメリットは、クライアント側にWebブラウザさえあればよく、クライアントのOSを問わないという点です。

またプログラムは、ASP.NETによってサーバー側で実行されるため、クライアントのCPUパワーを必要としません。

そしてアプリケーションを改良する際には、サーバー側に配置したプログラムを修正するだけで良いので、保守や改良が容易という点もメリットとして挙げられます。

反面、Webアプリケーションでは、ユーザーインターフェイスがHTML（場合によってはJavaScriptを含む）で構成されるため、提供できるユーザーインターフェイスは、HTMLで表現できるものに限定されます。すなわち、複雑なユーザーインターフェイスを表現でき

ません。

また、Webアプリケーションでは、クライアントが明示的にサーバーを呼び出さない限り、何かプログラムが実行されることはありません（図1）。

Webアプリケーションにおいてユーザーが見ている画面は、URLを入力したり、リンクをクリックしたり、フォームの [SUBMIT] ボタンを押したときなどのタイミングでサーバー側のプログラムが実行された静的な結果に過ぎず、リアルタイムなデータをユーザーに表示することはできません。

そのためデータの即時性が求められる場面では、Webアプリケーションは向かないと言えます^[注1]。

Windowsアプリケーションの場合

Windowsアプリケーションの場合には、図2のようにクライアントOS上でプログラムが動き、データベース処理をする構成となります^[注2]。

.NET Frameworkでは、Windowsアプリケーションを、Windowsフォームとして構成することになります。そのためWindowsフォームで提供されてい

る多様なコントロールを使った豊かなユーザーインターフェイスを提供できます。

しかしクライアントOS上で、.NET Frameworkのアプリケーションが動作することになるため、次の点が問題となります。

問題 1 .NET Frameworkが必要

クライアントには、.NET Framework自身を事前にインストールしておかなければなりません。

.NET Frameworkをインストールするには、.NET Framework再配布パッケージをクライアントOS上で実行します。Windows XPの場合には、Windows Updateを使って.NET Frameworkをインストールする方法もとれます。

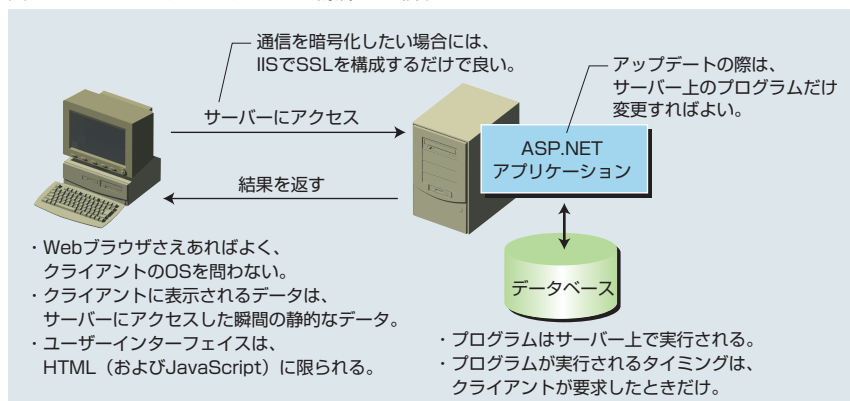
また、アプリケーションのインストーラを提供するのであれば、インストーラに.NET Framework再配布パッケージを含めてしまう方法もとれます。

.NET Framework再配布パッケージを含むインストーラを作りたいのであれば、マイクロソフトの「.NET Framework 1.1配布ガイド」で提供されている「Setup.exe Bootstrapperサンプル」を使うとよいでしょう。

.NET Framework 1.1 配布ガイド

▶ http://www.microsoft.com/japan/msdn/net/deployment/dotnetframedepguid1_1.asp

図1：Webアプリケーションとして構成する場合



注1) METAタグで「HTTP-EQUIV」を使うと、最新のデータを刻々とクライアントに表示できますが、これはリロードにより定期的にサーバーに接続し直して最新の情報をユーザーに表示しているに過ぎません。

注2) 図2は、2階層アプリケーションの話です。3階層以上のアプリケーションについては後述します。