



Web アプリケーションの効率的な開発

田中 和明¹

1 はじめに

インターネットを介して、さまざまなサービスを提供するためのアプリケーションとして、Web アプリケーションが広く採用されています。利用者の立場からは、ブラウザを使ってアクセスするだけで、静的なホームページを閲覧するのと同じ感覚で、さまざまなサービスが利用できるというメリットがあります。

Web アプリケーションでは、サービスを提供する Web サーバ側でほとんどすべての処理が行われ、ブラウザは処理に対する入出力を行うためのインターフェースという位置づけになります。従って、スタンドアロンアプリケーションにおけるソフトウェアのインストールやデータベースなどの準備が不要となります。このメリットのおかげで、誰でも簡単にさまざまなサービスを受けることができるわけです。

本稿では、これから Web アプリケーションを使ってサービスを提供しようとする開発者に役立つ技術を紹介し、Web アプリケーションの実装方法は、本稿で紹介する手法以外にも提案されていますが、ここではできるだけ少ない手順で汎用的な Web アプリケーションを実装することを最大の目標としています。そのため、必ずしも既存のソフトウェア資産を継承することは考えていません。この点にご注意ください。

2 ASP.NET と .NET Framework

Web アプリケーションの開発として、いくつかの開発手法が提案されています。ここでは、Web アプリケーション開発手法の一つとして、ASP.NET について説明します。ASP.NET は、Microsoft 社の提供する、Web アプリケーションおよび Web サービス向けのクラスライブラリで構成されるソフトウェアです。

2.1 .NET Framework

.NET Framework は、Microsoft 社が提供するソフトウェアプラットフォームで、OS 機能の上位に OS に依存しないインターフェースを設けることで、さまざまなハードウェアプラットフォームに対応できるアプリケーション開発を可能にする仕組みです。

.NET Framework は、ランタイム (CLR=Common Language Runtime) とクラスライブラリで構成されています (図 1)。この図の構成は、ちょうど JAVA の VM とクラスの関係に相当しています。

¹情報工学部 機械情報工学科 講師

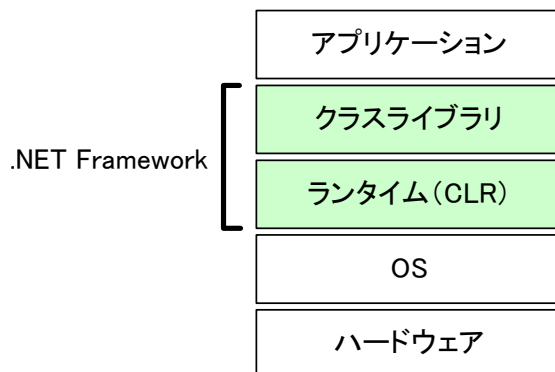


図 1: .NET Framework

.NET Framework そのものは、ソフトウェアプラットフォームであるため、アプリケーションを記述するプログラム言語を限定していません。実際、.NET Framework に対応したプログラム言語として、Microsoft 社から C++, Visual Basic, C# が提供されていますし、他のベンダからさまざまな言語に対応したコンパイラが提供されています。Pascal や COBOL, FORTRAN, Prolog, Lisp などでも利用できるようです²。

作成したアプリケーションは、コンパイラにより実行可能な形式に翻訳されますが、.NET Framework では、MSIL(Microsoft Intermediate Language) と呼ばれる中間コードとして生成されます。この MSIL が、ランタイム内の JIT コンパイラによりネイティブコードとして実行されます。

.NET Framework によって、言語に依存しない開発環境が提供されることとなりますが、もう一つの特徴として、言語間でクラスを継承できるという点があります。これは、例えばプログラムロジックを C++ で、ユーザインターフェースを Visual Basic で記述しておき、それらをクラス継承を使うことで組み合わせ、1つのアプリケーションとして完成させるということが可能です。

このような面で、実行環境(クラスライブラリ、ランタイム)は JAVA に似た仕組みを持って言いますが、言語から中間コードとクラスライブラリを分離していることで、開発手法における自由度が高くなっています。

2.2 ASP.NET

ASP(Active Server Pages) とは、Microsoft 社の Web サーバである IIS(Internet Information Services) で利用できる機能の一つで、Web サーバで動作するアプリケーション(Web アプリケーション)を開発できます。ASP では、クライアントからの Web ページのアクセスに対して、Web ページを動的に作成して応答します。この機能は、CGI や PHP などでも利用することができますが、IIS においては ASP が標準の機能として組み込まれています。

ASP.NET は、ASP が提供していた Web ページの動的作成の仕組みを、.NET Framework で提供するものです。ASP.NET は、.NET Framework のクラスライブラリの一部として提供されることから、.NET Framework の特徴と同じく、言語に依存せずに Web アプリケーションを開発できます(図 2)。

²JAVA や SmallTalk, APL まで存在すると聞いたことがあります。

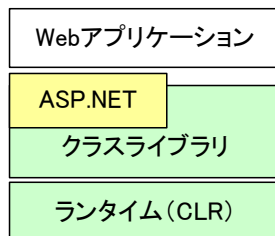


図 2: ASP.NET

ASP.NET で Web アプリケーションを作成するには、Web ページの構成 (HTML で記述するファイル) と動的にページを作成するためのコードを記述します。ASP.NET でのコード記述は、インラインコードかコードビハインドと呼ばれる手法で行います。

インラインコードとは、PHP のように、Web ページの構成の中にプログラムを埋め込む手法です。この手法では、HTML ファイルを記述するのと同時にプログラムのコードを記述できます。

ブラウザから hello.aspx のページに対する要求があった時に、IIS がコードをコンパイルします。IIS は、プログラムを実行して得られた結果を、<asp:Label ... \>の箇所に埋め込んで、ブラウザへ返します。なお、一度コンパイルされたコードは、自動的にキャッシュされ、それ以降の実行は短時間で済みます。

インラインコードの例 "hello.aspx"

```

<%@ Page language="c#" %>
<script runat="server">
public void Page_Load(Object sender, EventArgs e) {
    Label1.Text = "Hello, World!";
}
</script>
<html>
<body>
<asp:Label id="Label1" runat="server" />
</body>
</html>
  
```

コードビハインドとは、Web ページの構成とは別に、プログラムの記述を行います。そのため、1 つの Web ページは、少なくとも 2 つのファイルで構成されることになります。コードビハインドを使うことで、Web ページの出力に関する記述と、処理のロジックに関する記述を分離できます。

ブラウザから hello.aspx のページに対する要求があった時に、IIS は hello.cs をコンパイルします。

コードビハイン드의例 "hello.aspx"

```
<%@ Page Src="hello.cs" Inherits="HelloWorld"%>
<html>
<body>
<asp:Label id="Label1" runat="server" />
</body>
</html>
```

コードビハイン드의例 "hello.cs"

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;

public class HelloWorld : Page
{
    public Label Label1;

    public void Page_Load(Object sender, EventArgs e) {
        Label1.Text = "Hello, World!";
    }
}
```

コードビハインドにより，Web ページのデザインとロジックを分離できることから，開発者の負担を軽減できます。

3 データベース

.NET Framework では，データベースに関連したクラスライブラリも標準で提供されます。多くの開発環境では，データベース機能を付加的な機能として位置づけていました。このことは，データベースを利用するために必要となる実行環境（データベースアクセス用のライブラリなど）が必要になっていました。

.NET Framework がデータベースのクラスライブラリを持っているので，ASP.NET では特別な環境を用意しなくてもデータベースの機能を使えます（図 3）。

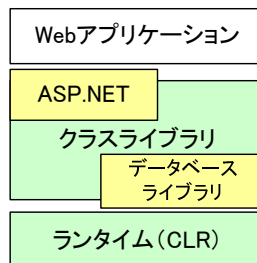


図 3: ASP.NET とデータベース

.NET Framework で提供されているデータベースクラスライブラリは、ADO.NET と呼ばれるデータベースアクセス手法を基本としています。Microsoft 社だけではなく、他社から提供されるデータベースクラスライブラリについても、ADO.NET に準拠しているため、標準のクラスライブラリと同じ手順で利用できます。

Microsoft 社からは、SQL Server、OLE DB(Access など)、ODBC、Oracle に対するクラスライブラリが標準で提供されています。他のデータベースベンダは、ADO.NET に対応したクラスライブラリを公開していますので、多くのデータベースが利用できます³。

別な観点で言えば、データベースの移行が容易であるともいえます。データベースの種類によらず、データベースへのアクセス手法は ADO.NET に準拠していますので、アクセス対象となっているデータベースのクラスライブラリの呼び出し部分を変更するだけで、データベースの変更が可能となります。

4 Web アプリケーションの例

ASP.NET を使った具体的な Web アプリケーションを紹介します。

ここでは、入力ボックスに入力された数値に対して四則演算を行うプログラムを示します。開発環境として、Microsoft 社からフリーで提供されている、「Microsoft Visual Web Developer 2005 Express Edition⁴」を使っています。

Web アプリケーションの開発は、Web ページの設計とコードの入力の 2 段階に分けられます。

Web ページの設計では、図 4 のように、画面上に文字やコントロール (Web ページを構成する部品) を配置していきます。この設計を、HTML エディタを使ってファイルを直接編集して行うこともできます。

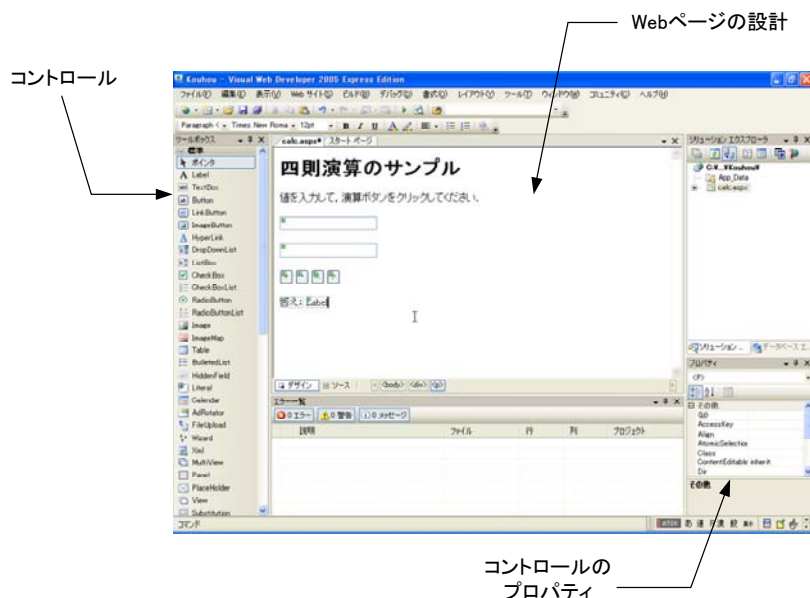


図 4: 開発環境 (Web ページの設計)

³DB2, PostgreSQL, MySQL など

⁴<http://www.microsoft.com/japan/msdn/vstudio/express/vwd/>

解説

次に、ボタンがクリックされたときの処理を記述します。Web ページの設計画面でボタンをダブルクリックすると、図 5 のようにコードを入力する画面へ切り替わります。

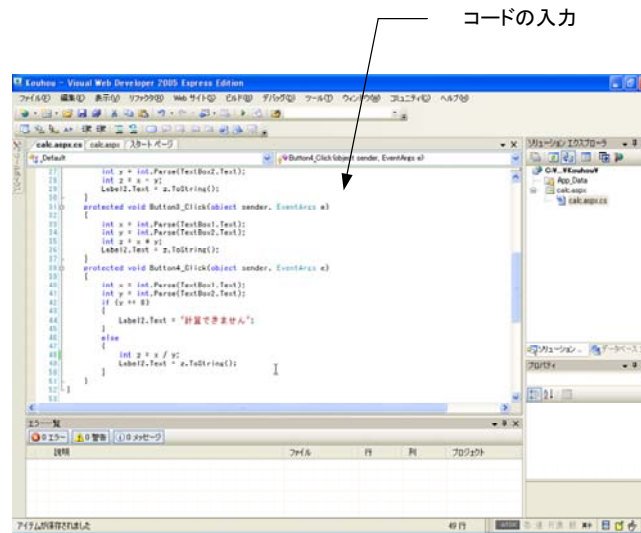


図 5: 開発環境 (コードの入力)

例えば「÷」ボタンをクリックしたときのコードを示します。

```
「÷」ボタンをクリックしたときのコード
protected void Button4_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    if (y == 0)
    {
        Label2.Text = "計算できません";
    }
    else
    {
        int z = x / y;
        Label2.Text = z.ToString();
    }
}
```

Web ページの設計とコードの入力が完了したら、Web サーバで公開します⁵

⁵なお、開発環境も Web サーバを持っているので、ファイル転送などを行わなくても、そのままブラウザで動作を確認できます。

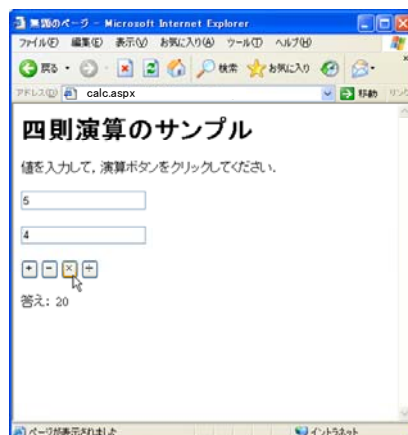


図 6: 実行結果

5 おわりに

本稿では、ASP.NET の利用について簡単に解説しました。 .NET Framework によるアプリケーション開発は歴史が浅く、参考資料も豊富とはいえません。しかし、言語に依存しないクラスライブラリや、言語相互のクラス継承などの特徴を持ちます。これらの機能を使うことで、ソフトウェア資産の活用や、開発の負担を軽減が見込めます。

また、Web アプリケーションは、ページデザインとロジックとの切り分けが難しいことから、従来は開発が難しいとされてきました。ASP.NET のコードビハインドを使うことで、これらの分担も可能となるでしょう。

.NET Framework、および、その応答である ASP.NET は、幅広いアプリケーション開発の基盤となる技術ではないかと期待しています。