



現行のC/Sモデルを .NETで再構築

.NETアプリケーションの開発：Windowsフォーム編

株式会社 ナルボ
醍醐 竜一 DAIGO, Ryuichi

Windowsフォーム という選択肢

これまでの企業アプリケーション構

築の進化の過程は、図1のようなものでした。

まず初期のモデルであるスタンドアロンアプリケーション、これは紙ペー

スで行っていた業務にコンピュータを導入することでコストの削減や情報の整理を実現しました。

そして次に、複数のコンピュータがネットワークで接続されることでC/S(クライアント/サーバー)モデルのアプリケーションが実現しました。これは、情報の共有/分散といった面でのメリットをもたらしました。

その後、OS非依存/インターネット汎用プロトコルの使用や、容易に広範なコンピュータへサービスが提供可能になるなど、エンドユーザーや開発者、システム管理者にとって多くのメリットをもたらすWebアプリケーションモ

本稿で前提となるもの

OS Windows 2000 Professional (SP2) 以降
Windows 2000 Server (SP2)

開発環境 Visual Studio.NET
.NET Framework 1.0.3705.288
Visual C# .NET Compiler 7.00.9466
Internet Explorer 6.0.2800.1106.xpsp1
Internet Information Services 5.0
SQL Server 2000

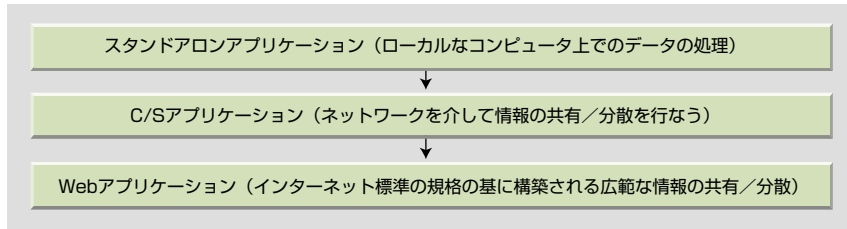


この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥CS2NETフォルダ以下に収録しています。

¥DOTNET_MAGAZINE.STOCKDBOBJECTS :
データベースアクセスオブジェクト
¥STOCKMANAGERSERVICE : XML Web サービス
¥STOCKMANAGERXMLCLIENT : XML Web サービスクライアント
¥WINDOWSFORMXMLSOL :
XML Web サービスサンプルのソリューションファイル
¥REMOTESTOCKOBJECT : リモートオブジェクト

¥STOCKMANAGERREMOTINGSERVER :
リモートオブジェクト登録用コンソール
¥STOCKMANAGERREMOTINGCLIENT :
リモートオブジェクトクライアント
¥WINDOWSFORMREMOTINGSOL :
.NETリモートリングサンプルのソリューションファイル
README.HTM : サンプルの構成、セットアップ手順などの注意書き

図1：アプリケーションモデルの変化



デルが時代のトレンドとなりました。

こういったアプリケーション構築モデルの変化の流れの中で、あえて「Windows フォーム」を選択するということは一見、時代の流れに逆行するように感じる方も多いことでしょう。はじめに「なぜ今、Windows フォームなのか？」ということについて考えてみましょう。

Windows フォームのメリット

Webアプリケーションモデルには多くのメリットがあります。しかし、同

時にデメリットがあるのも事実です。

まずは表1をご覧ください。表1には、「従来のC/S Windowsアプリケーション」「Webアプリケーション」「.NET Windows フォームアプリケーション」のそれぞれの特徴が理解できるよう、いくつかの比較項目をあげています。

Webアプリケーションの一番の欠点は、ユーザーインターフェイスの貧弱さにあります。

そして、リッチなユーザーインターフェイスを提供する従来のC/Sアプリケーションでは、必要とされるネットワーク環境や配布にかかるコストとい

った面での問題があります。

今回注目する.NET Windows フォームアプリケーションでは、リッチなユーザーインターフェイス、インターネット標準プロトコル、配布コスト削減といった特徴に見られるように、C/SモデルとWebモデルのそれぞれの良いところを取った（一部には中間を取った）設計がなされています。

エンドユーザーにとって最も重要な要素のひとつと考えられる「リッチなユーザーインターフェイス」を提供することができる点が“Windows フォーム”の一番の魅力でしょう。

.NETの分散オブジェクト技術

実際にアプリケーションを構築する際、N層での設計を行なうことが多いのではないかと思います。図2には、分

表1：C/S、Web フォーム、Windows フォームの比較

比較項目	従来のC/S Windowsアプリケーション	Webアプリケーション	.NET Windows フォームアプリケーション
動作プラットフォーム	Windowsに限定	HTML（一部にはJava Script、VBScriptも使用）が表示できればOSには非依存	.NET Frameworkが動作プラットフォームであるため、基本的にはOS非依存（現在の実情ではほとんどがWindowsだが、将来MacやLinuxでの動作も十分考えられる）
必要ネットワーク環境	多くの一般的な分散C/S環境では、DCOMやCORBAなどを利用していた。そのため、ファイアウォールが越えられない	HTTPでの通信のため、インターネット標準	XML Web サービス、.NETリモートティングを利用する。前者はHTTP標準。後者はTCP、HTTPが標準でサポートされているため、インターネット標準にも対応
セキュリティ	インターネットへ公開する場合は、専用ポートを用意する必要がある。ネットワークセキュリティの面での不安あり（ただし、ローカルネットワーク内での利用であれば問題ない）	HTTPのため通常のWeb公開と同様	HTTPが利用できるため、Webアプリケーションと同様
配布	クライアントアプリケーションおよび依存DLL（ランタイム、システム含む）を配布。また、レジストリ登録が必要な場合もある。バージョンアップ時には再度配布が必要	ブラウザさえあれば配布の必要はない（URLへのアクセスのみ）	クライアントには、.NET Framework（ランタイム）がインストールされていれば、アプリケーションフォルダのXCOPYで配布が可能。Visual BasicランタイムやMFCといった依存するシステムDLLの再配布は不要。また、バージョンアップ時には自動アップデート機能を利用可能
ユーザーインターフェイス	WindowsのリッチなUIを使用可能	HTMLのため、作成できるUIには限界がある	.NETのリッチなUIを使用可能（ほぼWindowsのUI機能と同様）

