

ベールを脱いだWindows Server 2003 との関係を探る

特別企画

現実解を加速する Visual Studio .NET 2003 登場

都筑 董 TSUZUKI, Sumire

本誌が書店に並ぶ7月上旬には、マイクロソフトの新しい開発環境「Visual Studio .NET 2003」がリリースされているはず。このプロダクトは.NETを基盤とするアプリケーション開発ツールの第2世代となります。

この新しいVisual Studio .NET 2003を使った開発にはどのようなメリットがあるのでしょうか。そのメリットや、同時にリリースされるWindows Server 2003との関係を解説しつつ、これから私たち開発者は何を目標さなければならないのかを考えてゆくことにします。

なぜ Visual Studio .NET 2003?

Visual Studio .NET 2002が出荷されてから1年有余。にもかかわらず、次のバージョンである2003の提供が開始されました。今までにはあまり例を見ない早いペースで開発環境の更新がなされたわけです。

このタイミングで新しい開発環境を利用する、あるいはアップグレードする。まずはその意味について考えてみましょう。

新しいビジネスの展開

当たり前の話ですが、私たち開発者がVisual Studio .NET 2003を使って開発するアプリケーションは、最終的にはシステムを利用するエンドユーザーが活用するものです。

これまでのVisual Studio .NET 2002でも、WindowsアプリケーションやWebアプリケーション、コンポーネントやWebサービスといったものを開発することができました。したがって、前のバージョンさえあれば今必要とされるシステムは十分に構築できるに思うかもしれませんが。

一方でビジネスの環境は刻一刻と変化しています。多くのユーザーによって利用しやすい環境に変化に追随しながら迅速に提供することは、開発者としての責務でも

あり、システムそのものの価値も高めます。

たとえば、モバイル向けアプリケーション開発です。これまで、携帯電話に対応するWebアプリケーションの拡張方法として「Mobile Internet Toolkit」がVisual Studio .NET 2002および.NET Framework 1.0向けに提供されていました。

この機能は、.NET Framework 1.1において完全に統合され、ASP.NET モバイルコントロールとして実装されています。そのため、ソースコードはもちろんプログラミングモデルにおいても一貫性が保たれており、これまで構築してきたコンテンツは、基本的にそのまま.NET Framework 1.1上で動作させることができます。

また、Visual Studio .NET 2003では、リッチクライアントのソリューションを実現する手段として.NET Compact Frameworkが搭載されています。Pocket PCやWindows CE .NET 端末に対してリッチな操作感を実現するアプリケーションを展開できることは、現在のシステム環境下では非常に重要なポイントです。

たとえばこの機能は、外出先での利用が多いことを想定した営業支援アプリケーションなどに向いているはずです。これまで、わざわざノートPCなどを持ち出していたようなケースも、PDAなどのデバイスへの変更によって、作業の効率化や負担を軽減することができます。そして、オフラインでのデータアクセスやデータベースとの同期を実現することで、必要なデータに必要なときに

アクセスできる操作性の高いクライアントソフトウェアを提供することが可能となります。

このようなソフトウェア環境は、既存のシステムをXML Web サービスやSQL Server for Windows CEなどを利用して拡張すれば簡単に実現できるはずで、つまり、今あるシステムに新たに付加価値を創出し、そして、利用しているエンドユーザーにおけるシステムへの満足度を高めてゆくことができるわけです。

□ 現状のシステム、今後のシステム

システムは、クライアントごとに単独で存在しているは効率が良いとは決まっていせん(図1)。さまざまなクライアントを結び付けるための技術やクライアントそのものを実現する技術が必要です。また、クライアントの裏で動作しているシステム同士を結び付ける技術など、数え切れないくらい多くの要素によってユビキタスネットワーク的なソフトウェア界が実現されます(図2)。

そして、ここで取り上げたような各種Web/Windowsアプリケーションやモバイルの上で動作するリッチクライアントはもちろん、それらのバックエンドで動作するコンポーネントやデータアクセスまで、一貫したプログラミングモデルによって実現される点がVisual Studio .NET 2003の一番大きな特色といえるでしょう(図3)。

既存資産の活用

ここまでで紹介したメリットを享受するためには、今あるアプリケーションが.NET Framework上で動作していなければ意味がありません。もちろん、データのみを共有化して共存するという手法もありますが、できることならシステム全体を.NETに対応したシステムとして構築しておくことがベストです。

そこで注目したいのが、既存資産となるVisual Basic 6.0(以下VB6)のコードやプロジェクトをどのように活用できるのかという点です。

これまでVisual Studio .NET 2002では、“アップグレードウィザード”を提供し、一般的なVB6プロジェクトを移行することが可能でした。しかし、このバージョンで移行できるのは、Windowsアプリケーションとそれら

図1：インターネット環境とソフトウェアの関係

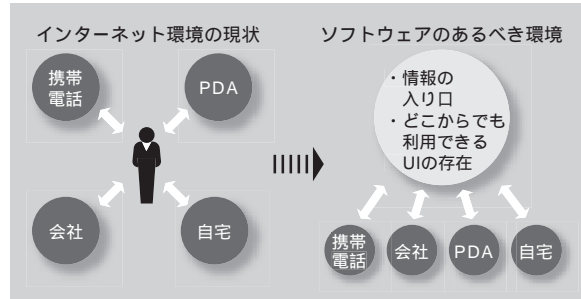


図2：ソフトウェアとしてのユビキタスネットワークの実現

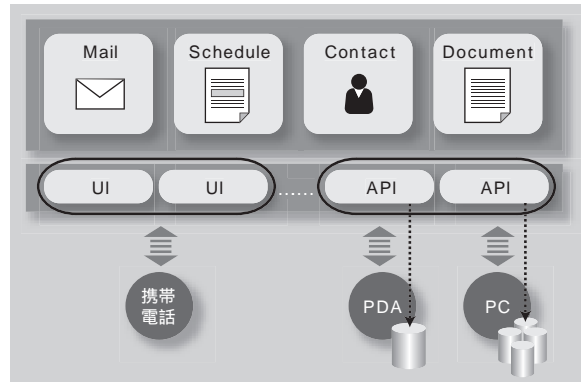
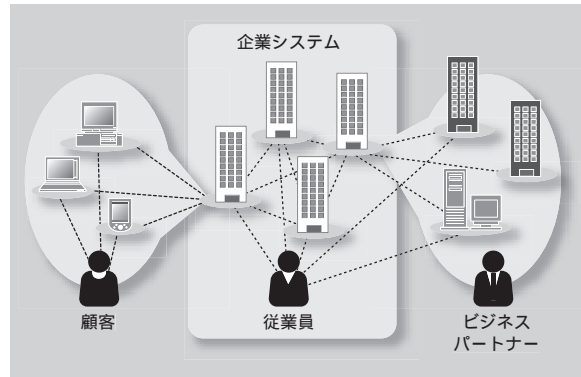


図3：“つながる”企業システム



に関連するものであり、Webを利用したプロジェクトに対しては移行方法がありません。

これに対して、Visual Studio .NET 2003では、これまで提供されていたアップグレードウィザードをより発展させ、ユーザーコントロールやWebClassなどのプロジェクトもVisual Studio .NET 2003のプロジェクトへ(条件付きながら)変換できます。これにより、これまで以上に多くの資産を.NETの世界へと移行することができ