



# 現行のクライアント／ サーバーモデルを .NETで再構築

## .NETのエンタープライズアプリケーションモデル

株式会社 ナルボ  
齋藤 友男 SAITO, Tomoo

### Microsoft.NETの混乱

Microsoft.NETの開発環境として Visual Studio.NETがリリースされてから1年が経とうとしています。みなさんの周りでは.NETはどのように見られ、また扱われているのでしょうか。

.NETはDOS、Windowsに続く、大きなパラダイムシフトであるにも関わらず、市場の反応は冷めたものであったように思われます。

ひとつには、ちょうどエンタープライズアプリケーション開発においてJavaが定着してきた時期と重なったこともありますが、それだけでもないようです。.NETは、いわゆるクライアン

ト／サーバーモデルからWebブラウザモデルまで含む統一プログラミングモデルをベースにXML Webサービスの話があり、さらにはパスポートなどの各種サービスをMicrosoftが提供しているなど、そのトピックには事欠きません。一方、DOSからWindowsへの移行時のようにビジュアルなインパクトがなく、エンドユーザーに対する直接的なメリットが見えにくいこともあって、これまでのWindowsの拡張の範囲と見られる面もあります。このため、さまざまなトピックの情報のみがかぎ交錯し、.NETの見られ方や扱われ方には混乱があったようにも思われます。

しかし、Microsoftの必死のマーケット活動もあり、2003年に入り状況も変

わりつつあるようです。いまだJavaだけに注目しているソフトウェア関連会社もあるようですが、積極的に生産性を上げようとする会社や部署においては、.NETは無視できない存在と認識され、少なくとも調査／検討に入り、そしてその多くが今後利用する方向で動き始めているのではないのでしょうか。

すでにお気づきの読者も多いと思いますが、ソリューションという視点から見れば、必ずしも.NETを導入する必要はなく、これまでどおりWindowsでの開発でよいのかもしれませんが、しかし、大切なのは、

「インターネットをプラットフォームとするソフトウェア開発の生産性において、.NETは圧倒的に有効な環境である」

ということです。

要するに.NETはソフトウェア開発者のためのパラダイムシフトなのです。そして、その成果は最終的にエンドユ

本稿で前提となるもの

初級

中級

上級

ユーザーへも反映されることとなります。ともあれ、今後Microsoftが提供するあらゆる環境において、.NET以外の選択の余地がないことは確かなようです。

## これまでのテクノロジーと .NET !

.NETはその一貫性とは裏腹に、これまでの連続性から見ると大変焦点が絞りづらいテクノロジーともいえます。つまり、

「Visual Basic (VB) で開発したクライアント/サーバーアプリケーションはどうなるの?」

「ASP (Active Server Pages) で開発したWebアプリケーションはどう変わるの?」

など、それぞれの既存テクノロジーの観点から見るとその全体像は見えにくいといえるでしょう。なぜなら、.NETは、WindowsというプラットフォームをベースとするこれらのMicrosoftテクノロジー、すなわちWindows DNA (Distributed interNet Applications Architecture) を、インターネットとXMLを軸に再整理/融合させたものだからです。

そのため、それぞれのテクノロジーの拡張としてとらえようとするとしても無理があります。ということで、.NETを理解するには、まずは.NET Framework、すなわち共通言語ランタイム (CLR: Common Language Runtime) を理解し、さらに.NET Frameworkクラスライブラリを知る必要があります。

とはいえ、これまで利用してきたVB

やASPなどのノウハウが無駄になることはありません。むしろ.NETにおいては、これらの経験を活かせるものとなっているため、これまでの経験がある技術者にとっては、無理なく.NETへ移行することができるはずです。すでに本誌の読者は、VS.NETが使われた方も多いと思いますが、その操作性やクラスライブラリのスマートさに驚かれた方も少なくないのではないのでしょうか。

結局、問題は.NETにおける設計、すなわち全体のデザインや処理方式、および実装方式ということになります。

## エンタープライズアプリケーションのコンポーネントモデル

企業におけるアプリケーション、すなわちエンタープライズアプリケーションの必要性は言うまでもありません。これまでもJavaにおけるJ2EE (Java2 Platform, Enterprise Edition)、WindowsにおいてはWindows DNAというまとまりで、エンタープライズアプリケーションの環境は用意されてきました。そこで、ここではエンタープライズアプリケーションのシステムモデルについて考えてみることにしましょう。

まず、ポイントとして、エンタープライズアプリケーションのシステムには、

- ・必ず対象となるデータがあり
- ・そのデータを扱うユーザーがいる

ということがあげられます。単純明快なポイントではありますが、これをコンポーネントモデルとして整理すると、

- ・データを管理する“データアクセスオブジェクト”
- ・ユーザーとのインターフェイスをとる“プレゼンテーションオブジェクト”
- ・その中間にアプリケーションに必要な処理を提供する“ビジネスロジックオブジェクト”

という構成となります。

これらのオブジェクトは相互に連絡し、そのためのインターフェイスをもちます。これらは、

- ・プレゼンテーションオブジェクトとビジネスロジックオブジェクト間の通信インターフェイス
- ・ビジネスロジックオブジェクトとデータアクセスオブジェクト間のデータアクセスインターフェイス

です (図1)。そして、このインターフェイスの選択がアプリケーションモデルを特徴づけます。

まず、データアクセスオブジェクトは、いかにデータを安全に保存し、高速に扱えるかがテーマとなります。いわゆるデータベース管理システム (DBMS) にあたるものです。

データベース管理システムというと、一般にパッケージのリレーショナルDBMSが連想されますが、必ずしもそれらのリレーショナルDBMSを指すものではありません。

次にプレゼンテーションオブジェクトですが「フォーム系」と「レポート系」があり、ユーザーに対し、いかにデータをわかりやすく提供するかがテーマとなります。フォームはWindows



現行のクライアント/サーバーモデルを .NETで再構築

Client/Server Model to .NET Model