

真の“使いやすさ”を目指して

.NET Architecture Forum

ユーザーインターフェイス論 序説

新連載

現在のコンピューティング環境を考える

マイクロソフト株式会社
デベロッパーマーケティング本部 デベロッパーエバンジェリスト
西谷 亮 NISHIYA, Ryo

Technology Tools

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

Duwamish7vb
サンプルアプリケーション

Level



Samples

はじめに

経理／在庫管理／販売管理といったさまざまなシステムは、もはやデフォルトといってもよいでしょう。読者のみなさんも、利用はもちろんのこと実装経験のある方もいるのではないのでしょうか。

システムを構築する段階においては、UMLによるモデリングやOO (=Object Oriented: オブジェクト指向) による設計、デザインパターンの活用やSOA (=Service Oriented Architecture: サービス指向アーキテクチャ) の実践といった、さまざまな設計手法が取り入れられ、より柔軟で変化に強いシステムへの考慮がなされているはずです。

しかし、バックエンドに存在する各種コンポーネントの実装方法は論じられても、システムを利用する側の立場に立った議論はあまり見かけることがありません。つまり、システムの入り口となるユーザーインターフェイスに対するアプローチが見落とされているケースが多く見受けられます。

それはいったいなぜなのでしょう。

本連載では、これから数回にわたって現状の問題点などを探り、アプリケーションにおけるユーザーインターフェイスの意味やそのあり方を問いながら、エンドユーザーの立場に立ったシステムの構成やユーザーインターフェイスデザインの指針を検討してゆくことにします。

現状を正しく認識する

現在のコンピューティング環境は「Webアプリケーションの活用」という流れの中にあります。これは、これまでのシステムの懸案事項だった多くの問題点を解決してくれるソリューションとして期待され、さまざまなアプリケーションで採用されています。もちろん、マイクロソフトもそのためのプラットフォームなどを随時提供してきました。

しかし、Webアプリケーションを採用したからといって、すべての問題点が解決されるわけではありません。

Webアプリケーションに潜在している問題点も認識しておかなければならないのです。ここでは、具体的な例を元に、問題点の洗い出しとエンドユーザーが求めるコンピューティング環境とはいかなるものかを考えてゆくことにしましょう。

メリットは何だったのか？

Webアプリケーションを採用する決め手となったのは、さまざまな理由があったはずですが。

たとえば、

- Webサーバーの集中管理によるアプリケーションのメンテナンスが実現できる
- ブラウザさえあれば、どんなクライアント環境からもシステムの利用ができる
- クライアント環境に対する直接的なメンテナンス頻度を低下させることができる
- 統合開発環境の利用で開発効率の向上と早期導入が可能
- 流行に乗り遅れない

などなどです。一番最後に挙げた理由はおいておくとして、これ以外にもさまざまな理由が検討されることでしょう。このようなメリットを感じているからこそ、Webアプリケーションが採用され、システムとして導入されてきたわけです。ある意味では、今までのクライアント／サーバー型のソリューションから脱却し、理想的なシステムがこれで実現できるようにも思えたはずですが。「どんなクライアントからも利用できて、管理コストはこれまでに比べ低減可能。そのうえ開発費用も時間も削減することができて、新しいビジネスをこれによって推進してゆくことができる……」と。

では、ここに問題点は存在しないのでしょうか？

問題点はどこか

メリットとして挙げられた項目に問題はないように思えます。しかし、実は大きな落とし穴があります。それは、

Webアプリケーションは、システムそのもの、または仕様上の制約によって問題点が発生してくるということです。

たとえば、以下のような要件の単純なWebアプリケーションがあったとします。

- データベースからカテゴリの情報を取得し、ドロップダウンボックス内に表示
- ドロップダウンボックスからカテゴリを選択すると対応した商品情報がグリッド上に表示

たったこれだけです。

ASP.NETとVisual Studio .NETを利用すれば、あっという間に作成できるようなものです（図1）。

■実装例-1

では、実際の実装例を見てみましょう。

Visual Studio .NET 2003（2002でも同様の作業で検証できます）で新規にASP.NET Webアプリケーションプロジェクトを作成し、図1のように画面上へ、DataGridコントロールとDropDownListBoxコントロールを配置します。各々の役割がわかるように、ここではLabelコントロールを使用して、説明（Category、Detail）を表示するようにしています。これらのコントロールには、SQL Server上にあるデータベーステーブルの情報を表示するためにSqlDataAdapterコントロールを利用して、データベースへの接続やDataSetの取得を行なうように設定してあります。ここでは、Duwa

図1：単純な実装例

