

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

第2回

クライアントアプリケーションの問題点

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:

Level



Samples

はじめに

今回は、Webアプリケーションを例に挙げながら、「クライアントのアプリケーションの実装形態によって、求めている仕様と現実がマッチしない」現状を紹介しました。そこでは、Webアプリケーションを否定するのではなく、「開発者は、アプリケーションが有する可能性と限界を認識する必要がある」ことを指摘しました。

もちろん、Windowsクライアントで実装すればすべてが解決するというものではありません。必要なときに必要なものを選択してゆくことで、実際の利用者であるエンドユーザーの視点に立ったクライアントアプリケーションが実現されるということです。

本稿では、前回とは逆にリッチなクライアント、つまり「Windowsアプリケーション」を中心にしてユーザーにとってのクライアントの在り方とその方向性を検討してゆくことにしましょう。

リッチクライアントとは？

リッチクライアント^{注1)}は、Webアプリケーション全盛以前に、多くの場面で使用されていました。Visual Basicなどを使用したWindowsアプリケーションは、開発生産性も高く、また、カスタマイズされた高い操作性を持つクライアント環境を提供することができたからです。業務に関連したアプリケーションを例に挙げると、[Enter] キーによるコントロールのフォーカス移動や、テンキーのみの入力を可能にするなど、特別なソリューションに特化したアプリケーションの開発／導入には非常に有効な手段でした。

つまり、前回紹介したWebアプリケーションがもっている、以下のような問題は、実はWindowsアプリケーションのようなリッチクライアントを使用することで、回避できていたのです。

・ Webアプリケーションでは、思いど

注1) ファットクライアント (=Fat Client) とも呼ばれる

おりの操作を実現する際に、クライアント環境に制約を与える、もしくは操作性が犠牲になる

- アプリケーションそのものの表現力を高めてゆくことが難しい。それゆえ、ユーザーの視点に立ったアプリケーションといえないケースもある

これらは、リッチクライアントを使用していたとしたら、いったいどうなっていたのでしょうか。

ここで、Visual Studio .NET 2003を利用したリッチクライアントの開発を例にとって検証してみましょう。前回、ASP.NETを利用したWebアプリケーションで実現したのと同じ機能を持つアプリケーションを、リッチクライアントによって作成してみます。主な機能としては、「データベースから書籍のカテゴリ情報をルックアップし、詳細の情報をグリッド上に表示する」というものです。

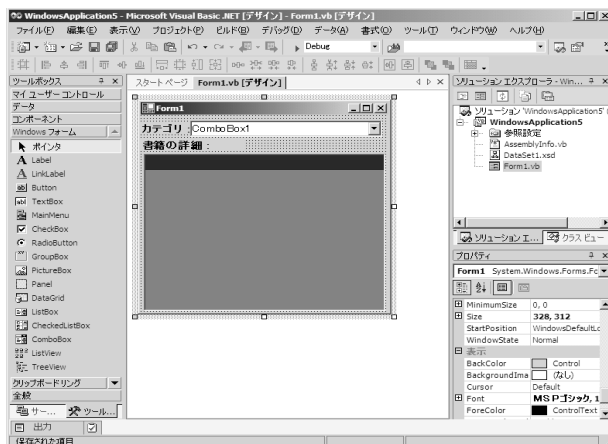
実装してみる

Visual Studio .NET 2003を起動し、新規にWindowsアプリケーションを作成します。ここでは、Visual Basic .NETで作成することにしましょう。

アプリケーションを作成する際は、Webアプリケーションを実装するときと同じで、必要なコントロールをドラッグ&ドロップし、画面上に配置してゆきます。

前回作成したWebアプリケーションと基本的な仕様はま

図1：Windowsアプリケーションのデザイン



ったく同じものにします。ここでは、ComboBoxとDataGridの2つのコントロールを配置し、各々の役割がわかるようにLabelで説明を表示できるようにしました(図1)。あとは、データアクセスを簡単にしてくれるコントロールの配置と最低限のコードを記述するだけで基本的な実装は完了です。

まずは、SqlDataAdapterコントロールをドラッグ&ドロップし、データベースとの接続情報の入力と、アクセスしようとするテーブルへのクエリなどを記述しておきます。カテゴリ情報を取得できるように、データアダプタ構成ウィザードで「SQLステートメントの使用」を使い、クエリビルダで図2のように設定します。

ここでは、もうひとつSqlDataAdapterコントロールを配置します。このコントロールで、ストアードプロシージャを呼び出し、コンボボックスで選択されたカテゴリに応じた書籍の詳細情報を取得します。

コントロールをドラッグ&ドロップしたら、データベースとの接続情報を選択し、「データアダプタ構成ウィザード」の「クエリの選択」では「既存のストアードプロシージャを使用」を選択します(図3)。次に、カテゴリ情報から書籍の情報を抽出するストアードプロシージャとして、「GetBooksByCategoryId」がすでにあるので、「SELECT」コンボボックスから選んで(図4)終了です。

その後、先の作業と同じように、画面上に表示されたSqlDataAdapterコントロールを右クリックし、データセットの生成を行ないます。ここでは、既存のデータセットに

図2：SQL文の生成

