

ノータッチデプロイメント 序論

Webアプリケーションを超えられるか？

尾島 良司

OJIMA, Ryoji

日本ユニシス株式会社
.NETビジネスディベロップメント

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
 - Visual Studio .NET 2003

Level



Samples

はじめに

「あえて言おう、カスであると！」

冒頭から年齢がばれちゃうガンダムネタ^[注1]でごめんなさい。

最近、なんだかプログラミングの充実感が少なくなってきたと思いませんか？ あなたのユーザーは、業務が効率化したと喜んでいませんか？ より良いものを作るためではなく、必要最低限のことを実現するために徹夜作業をしていませんか？

そう、悪いのは“Webアプリケーション”です。使い勝手が悪い上に実装に必要なコストが高い。実際に利用する人が喜ばないものを苦勞して作って楽しいはずがありません。誤解を恐れずに言ってしまうでしょう。「Webアプリケーションはカス」なのだと。

昔作ったC/SS (Client/Server System) のアプリケーションみたいに、痒いところに手が届きまくったプログラ

注1) ギレン・ザビ総帥。ア・バオア・クー戦直前演説。

ムを作りたい。理想のユーザーインターフェイスについて徹夜で議論したい。みなさんもそう感じているではありませんか？

幸いなことに、.NET Frameworkには“ノータッチデプロイメント (No-Touch Deployment)”という技術があります。ノータッチデプロイメントを使えば、Webアプリケーションと同等のTCO (Total Cost of Ownership) でリッチなユーザーインターフェイスを持つアプリケーションを作ることができます。あの充実した日々をこの手に取り戻せるのです。

というわけで、本稿では、ノータッチデプロイメントについて述べさせていただきます。

Webアプリケーション って使いづらい

ある日の、私[👤]と新人君[👤]との会話。

👤 Webアプリケーションって使いづらいよね。

- そうですか？ 普通に使いやすいと思いますよ。
- 使いづらいよ。ドラッグ&ドロップとかできないし。
- そんな……。Wordじゃないんですから。
- WordもWebアプリケーションも、ユーザーにとってはどちらも同じアプリケーションなの！ どちらも作業を効率化する道具にすぎないの！

Webアプリケーションしか作ったことのない新人君は、Webアプリケーションの限界を「思考の限界」にしてみました。これではノータッチデプロイメントを解説しても、その良さをわかってもらえません……。

とういわけで、まずはWebアプリケーションの限界について考察しましょう。考察のポイントは3つ。「Webブラウザは画面を表示する端末に過ぎない」ことと「JavaScriptにはグルーしか書けない」こと、そして「ActiveXコントロールにはセキュリティ上の不安がある」ことです。

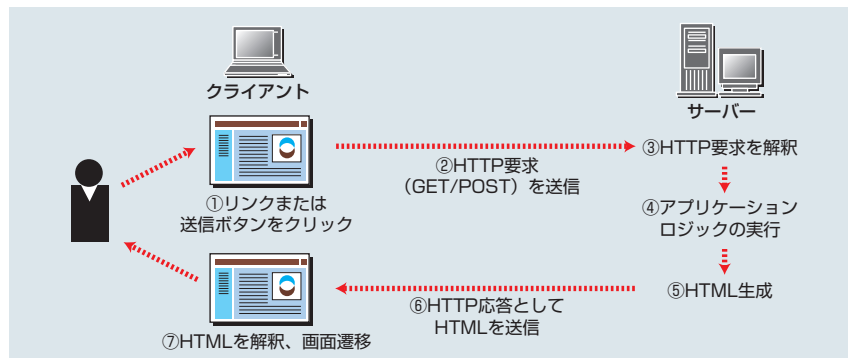
Webブラウザは画面を表示する端末に過ぎない

Webブラウザとは、HTTP (Hyper Text Transfer Protocol) でサーバーにリクエストを送信し、受け取ったHTML (Hyper Text Markup Language) をもとに画面を描画するソフトウェアに過ぎません。

Webアプリケーションの動きを概念的に表現すると、図1ようになります。

問題は2つあります。「HTML生成の部分にしか手を出せない」ことと「リ

図1：Webアプリケーションの動き



クエストが画面単位である」ことです。

「HTML生成の部分にしか手を出せない」というのは、我々プログラマがプログラミングできるのはリクエストを受けてどのようなHTMLを生成するかという部分だけということです。

「リクエストが画面単位であること」というのは、サーバーにリクエストを出すと現在表示している画面をすべて書き換えてしまうということです。

この2つの問題を合わせると、「プログラムを動かすためには画面をすべて書き換えなければならない」となります。つまり、何か処理をさせるためには、画面遷移をさせなければならないのです。

考えてください。メニューをクリックするたびに別のウィンドウが開いてしまうアプリケーションを。そして、その際に元のウィンドウが閉じてしまうアプリケーションを。こんなの、誰も使いたくないですよ。Webアプリケーションとは、まさに「このようなアプリケーションだった」のです。

JavaScriptにはグルーしか書けない

ひとつ前で「このようなアプリケー

ションだった」と書いたのは、先に挙げた制約がJavaScriptによって緩和されたためです。

Webサーバーからしかアクセスできない資源、たとえばRDBMSのデータなどにアクセスする場合は駄目ですけど、Webブラウザ内部で閉じる処理であればJavaScriptで実現できるのです^[注2]。

しかし、JavaScriptで可能になる範囲は限られています。一般にGUIはコンポーネント指向で作成します。Webアプリケーションでもこれは同じ。コンポーネント指向で作成されるプログラムは、コンポーネントとコンポーネントをつなげるための“グルー（糊）”で構成されます。コンポーネントがボタン、グルーがボタンに対するイベントハンドラです。さて、JavaScriptで作成可能なのは、このグルーの部分のみなのです。そして、HTMLは非常に低機能なコンポーネントを少量しか提供してくれないのです。

考えてください。もしもVisual Studio

注2) Internet Explorer 5.0のデータアイランドを使用すれば画面遷移しないでサーバーと通信できますし、framesetを使用して見えないフレームに通信させるという手もあります。が、これらの機能はいつでもどこでも使えるわけではないと考えます。