特集2

Windows でもAjax!

その1

Asynchronous JAvascript+XML

ASP.NET 1.1でAjax

吉松 史彰
YOSHIMATSU, Fumiaki

Level

1 2 3 4 5

Technology Tools

- ◯ Visual Basic
- ✓ Visual C#
- ◯ Visual C++
- ✓ SQL Server
- ◯ Oracle
- ◯ Access
- ✓ ASP.NET
- ✓ Other:
  ↓
  Visual Studio .NET 2003
  JavaScript

Samples

この記事で取り上げたソースコードおよびサンプルプログラムは、http://www.shoeisha.com/mag/windev/からダウンロード可能です。

Ajaxを知ってるかい?

今ここで初めて「Ajax(Asynchronous JAvascript+XML)」という言葉を見たという読者は、よほどWebアプリケーションの開発に興味がないか、世間の事情に疎いにちがいない。Ajaxという言葉はそう断言できるほどすでにWebアプリケーションの開発者に浸透している。

Webとデスクトップ

従来のWebアプリケーションのユーザーインターフェイスの動作は、図1のようにユーザー(ブラウザ)とWebサーバーが互いに呼吸をあわせ、同期を取って画面遷移を繰り返す仕組みだった。

たとえばGoogleでWeb検索を行なう場合、http://www.google.co.jp/に行き、検索キーワードを入力して[Google検索]ボタンをクリックすると、GoogleのWebサーバーへのアクセスが行なわれる。Webサーバーから返事が届くと、ブラウザは画面を描画するために内容を一度すべて消去し、Webサーバーから返されたHTMLをブラウザ画面に描画する。これで画面遷移が完了し、Webアプリケーションは“検索キーワード入力”状態から“検索結果閲覧”状態へと移行する。

デスクトップアプリケーションにも、このような動作をするものがある。いわゆる「ウィザード形式」のアプリケーションでは、ユーザーが画面上で選択した内容に基づいて状態が移行し、そのたびに画面のほぼ全体が(選択内容に応じて)再描画される。

ただし、デスクトップアプリケーションにおいては、ウィザード形式ではないアプリケーションのほうに、この仕組みがよく使われている。Excelでは、あるセルに入力された数値を変更すると、そのセルに関連付けられたセルだけが計算し直され、再表示される。Wordで文書を印刷すると、画面下のステータスバーに印刷状態を通知するアイコンが表示され、そこだけがせわしなく更新される。印刷を行なったとたんに別の画面に移動し、真っ白な画面に一から描画が始まるような動作はしない。

デスクトップアプリケーションでは、Webアプリケーションよりも細かな制御ができるため、「リッチなユーザーインターフェイスを実現できる」といわれる。リッチなユーザーインターフェイスでは人間への負担が少ないため、Webアプリケーションでもできるだけリッチ

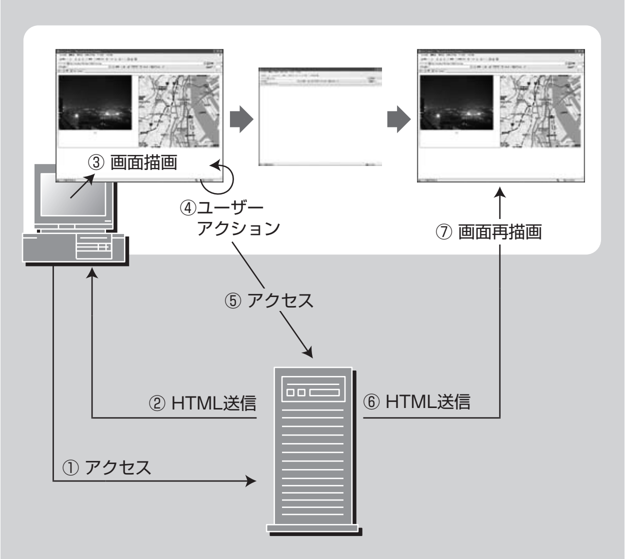

図1：従来のWebアプリケーションの動作

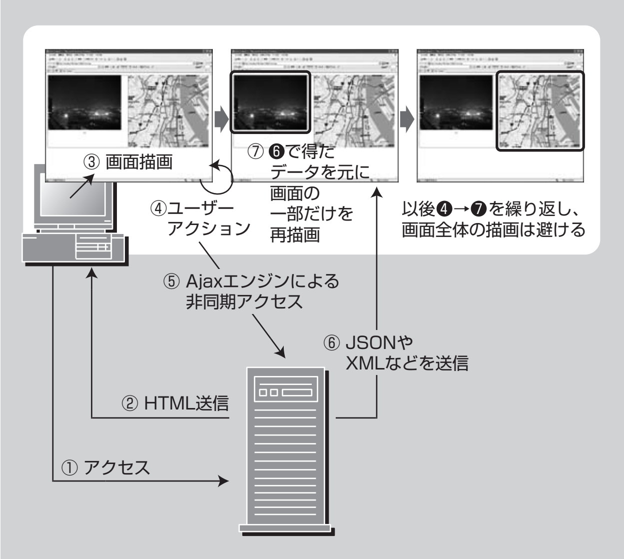

図2：Ajaxの動作

なユーザーインターフェイスが求められるようになるのは当然で、JavaアプレットやActiveXコントロールに始まり、Windowsフォームのアプリケーションをインターネットからダウンロードして起動できるようにしたり、Flashのように動画を表示するエンジンを利用してユーザーインターフェイスを作ったりと、Webブラウザのプラグインとなる何らかのエンジンを別途ダウンロードさせて、そのエンジンでユーザーインターフェイスを動かす方法が盛んに提案されている。

Ajaxは、リッチなユーザーインターフェイスを持つWebアプリケーションを実現するこれらの方法の中でも今もっとも勢いがあり、ブラウザに余計なエンジンをインストールしなくてもよいので動作させやすく、何よりも先進的で“クールな”印象を与えることができる技術だ（図2）。

Ajaxの仕組み

Ajaxの仕組みを利用するには、ブラウザ上でプログラムを実行する必要がある。多くのブラウザにすでにインストールされている実行エンジンから考えて、プログラミング言語の選択肢はJavaScript以外にない。したがって、Ajaxという技術を最大限活用するには、JavaScriptの知識が必須となる。もうひとつAjaxの仕組みに不可欠なのは、ブラウザ上のJavaScriptから起動できるHTTPプロトコルのエンジンだ。これがないと、画面の更新に必要な情報をWebサーバーにリクエストできない。

実際には、これら2つの必須要素が多くのブラウザで整備されたのが先で、それを利用するアプリケーションが出始め、それにAjaxという名前が最後につけられたというのが歴史的経緯である。Ajaxと呼ばれる技術利用法は、あくまで実用本位で、いまある技術だけでできることを積み上げているもので、大仰なアーキテクチャ論などとは無縁である。以前から存在した技術に“Ajax”というそれっぽい名前をつけてみたらブームになったということだ。

Ajaxの最後のXはXMLということになっているが、XMLはAjax的アプリケーション開発において必須要素ではない。ブラウザ上で動くプログラムとWebサーバー側で動くプログラムの両方を自分で開発する場合、お互いのプログラムにとって最適な形式でデータをやり取りすればよいわけで、何も冗長なXML形式に限定する必要はない。事実、A9.comなどAjaxで作られているといわれている多くのWebアプリケーションでは、XMLではなく「JSON（JavaScript Object Notation）」という形式でデータをやり取りしている。

Ajaxと呼ばれるために必要なのは、

・ブラウザで動作するJavaScriptで書かれたプログラムが画面遷移とは別に（非同期的に）Webサーバーと通信して動作すること

である。XMLかどうかは重要ではない。