

# .NET Framework

## なにを使うか、どう使えるのか アイデアノート

第17回

秋月巖ソリューション事務所  
秋月 巖 AKIZUKI, Iwao  
<http://www.akizuki.co.jp>

### Webアプリケーションで 双方向リアルタイム通信を 実現する顧客対応システム

#### Technology Tools

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

#### Level



#### 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、付録CD-ROMの¥DOTNET¥NOTEディレクトリに収録しています。

##### ¥WEBRT

顧客対応Webシステムのデモサンプル

##### ¥BIN

サンプルの実行に必要なUDAクラスのアセンブリ

#### コンポーネントを使わずに リアルタイムで通信

図1はWebブラウザで顧客とオペレータがリアルタイムで対話する顧客対応Webアプリケーションのサンプルである。このWebアプリケーションは、Webサイトに訪れた人が、その場ですぐにWebサイトの担当者に質問して回答を得ることができる機能を提供する。たとえば銀行のWebサイトで振り込み手続きをしている途中、操作方法がよくわからないことがあってすぐに質問したいというようなことはないだろうか。その場合、通常はメールによる問い合わせになるわけだが、そうすると返事がいつくるかはわからないし、少なくとも今の操作を中断しなければならない。電話してもいいわけだが、担当につないでもらうだけで一苦労しそうである。これがインターネットでなく、実際の銀行ならば近くに

る従業員に聞けばいいだろう。しかし、インターネットの場合、そうもいかない。

そのようなときに、このWebアプリケーションのような機能があれば、Webページ上のボタンをクリックすることで質問をするためのウィンドウが起動する(図1)。通常、このようなリアルタイム処理を行なうには、Javaアプレット、ActiveX、Flashといったクライアントモジュールを必要とするが、インターネットでは顧客の使用しているWebブラウザがそれらに対応しているかどうかはわからないという問題点があった。今回、紹介するサンプルアプリケーションでは、WebRT (Web RealTime) 技術を用いることで標準的なWebブラウザの機能だけでリアルタイム通信を実現している。クライアントのWebブラウザに要求されるのはフレームとクライアントスクリプト (JavaScript) に対応していることだけである。DHTML

図1：サンプルのWebリアルタイム応対システム



に対応している必要もないので、パーソナルコンピュータ用のほとんどのブラウザで使えるはずである。

ような技術的な限界をクリアするものである。決してトリッキーなものではないし、複雑なものでもなくシンプルな解決方法でこの問題をクリアしている。

## 通常のWebチャットと何が違う

図1のデモは付録CD-ROMにすべてのプログラムとソースコードを収録しているのので、読者はIISが稼動するサーバーコンピュータで動作を確認することができる。本連載の第2回でIISをWebサーバーとして利用し、Windowsアプリケーションで通常のチャットを行なうプログラムを紹介したことを覚えているだろうか。原理的にはあれと同じで、単にWindowsプログラムがWebブラウザに置き換わっただけである。

Webアプリケーションが利用しているHTTPプロトコルはステートレスのプロトコルだから、何かコンポーネントを利用しない限り、チャットのようなリアルタイム処理はできないはずだと考えるのが普通だろう。実際、Webブラウザを使うチャットプログラムは、定期的にWebサーバーから新しいデータを取得する非リアルタイムのものか、あるいは前述のようなクライアントモジュールを使うものがほとんどである。WebRT技術はこの

## WebRT技術のメリット

WebサーバーはステートレスのHTTPプロトコルを用いているので、Webブラウザがリクエストを送ってきた直後だけはデータを返すことができるが、その後切断してしまうので、任意のタイミングでWebブラウザに送り返すことができない。これがWebブラウザを使ってリアルタイム通信ができない理由のすべてである。だから、任意のタイミングでWebサーバーからWebブラウザへデータを送信する仕組みさえ用意すればリアルタイムで双方向通信ができることになる。

では、それをどうやって実現するかというと、その秘密はWebブラウザに非表示で読み込まれた隠しフレームにある。そのためWebRT技術を利用するにはフレーム対応のWebブラウザが必要なのである。これだけで勘のいい人は気づいたかもしれない。ステートレスのプロトコルでステートを維持するために、WebRT技術では非