

特集2

dotNET 事例ファイル

WebMuonに見る .NETアプリケーション 事例

NTTデータによる.NETを ベースにした大規模システム

中野 智加良 *NAKANO, Chikara*

株式会社NTTデータ
ビジネス開発事業本部
基盤システム事業部
第三ソリューション技術担当

はじめに

NTTデータビジネス開発事業本部の私のチームでは.NET Framework上にWebシステム開発用フレームワーク「WebMuon (ウェブミュオン)」を開発しています。このWebMuonフレームワーク (以下WebMuon) は、スケーラブルでハイパフォーマンス、高開発効率、高品質、高安全性なWebベースのデータエントリシステム/ワークフローシステムを開発するフレームワークです。実装のみならず、設計から試験までトータルにカバーしています。

以下、Webシステム開発の現状、.NET技術導入後に想定したメリット、WebMuonの概要などを交えながら、WebMuonの開発技術に採用した.NET技術について記述してゆくことにします。

Webシステム開発の現状

Webシステム開発では、以下の6点への考慮が非常に重要となります。

- ・ユーザーインターフェイス
- ・セッション管理

- ・文字コード/外字
- ・通信量/通信頻度
- ・印刷
- ・セキュリティ

また、Webシステム開発に限らず、システム開発においては次のような点も重要です。

- ・設計から実装/試験がスムーズに行なえる方法論とアプリケーションアーキテクチャとフレームワーククラスライブラリ

そこで、これ以降、クライアント/サーバーシステム (以下C/Sシステム) との比較を交えながら、上記各点についての詳細を検討してゆくことにしましょう。



ユーザーインターフェイス (UI)

WebシステムにおけるUIは、言うまでもなくWebブラウザです。Webブラウザは元々コンテンツのブラウジングを目的としたものであるため、C/SシステムのクライアントUIと比較すると操作性の面で貧弱な点や問題となる点が多いといえます。主だったものを表1と表2にまとめました。

ここで、「ナビゲーション関連」で挙げた点は、そもそもC/Sシステムでは考慮する必要がなかった点です。これはWebブラウザというコンテンツブラウジングを目的として開発された汎用のUIを利用したため発生した新たな問題であるといえるでしょう。

操作性関連については、C/Sシステムでも考慮してきた点ですが、専用のUIを用いていたため、比較的容易に解決できていました。Webシステムでは、Webブラウザという汎用のUIを使うことにより、その対応が技術的に困難になったのです。

っている業務における一時的な情報が必ず存在します。この一時的な情報がセッションです。

C/Sシステムでは、セッションはすべてクライアントアプリケーションに保持されており、さらにクライアントアプリケーションの挙動は、すべてそのアプリケーション自体によって制御されているので、セッションに対する特別な配慮は不要でした。しかし、Webブラウザではページ遷移が発生するとページ内に保持されていたメモリ情報は破棄されてしまうため、セッションを保持し続けることができません。そこで考案されたのが、サーバーサイドセッション管理です。しかし、このサーバーサイドセッション管理には次のような問題があります。



セッション管理

問題① 一貫性の問題

WebシステムでもC/Sシステムでも、ユーザーが行な

[戻る] ボタンなどサーバーへの通信を伴わないブラウザ

表1：ナビゲーション関連（Webシステムの問題点）

フォームの二重送信／リロードボタン	ユーザーはいつでもツールバー／メニュー／[F5] キーなどでフォームの再送信が可能。これによって、情報の二重送信や二重更新の危険が生じる
[戻る] / [進む] ボタン、お気に入りやショートカットなど	これらの機能はコンテンツブラウジングには適しているが、操作自体はWebサーバーには送信されない。そのため、情報の投入や更新機能があるWebシステムでは、想定していない操作シーケンスが生まれてしまい、ブラウザとWebサーバーおよびDBサーバーで一貫性を失う危険性が生じる
二重ナビゲーション	ユーザーはいつでも新たなウィンドウでブラウジングを開始することができるため、二重送信や二重更新、あるいは想定しないシーケンスの危険性だけでなく、“カンニング”のようなことができてしまう危険性もある

表2：操作性関連

	C/Sシステム	Webシステム
キーボード制御	ファンクションキーやショートカットキー、あるいは「入力文字のフィルタリング／自動変換」「挿入／上書き」などの制御を行なえるため、ユーザーの利便性が高い	Webブラウザの機能だけでC/Sシステムと同様な機能を実現するのは困難か不可能
IME制御	フィールド単位での「IMEのオン／オフ」「文字種設定」などのIME制御を行なえる	WebブラウザやHTMLなどの仕様は海外で開発／策定されたものであるため、IME制御への考慮が少ない。サポートされていないものが大半
フォーカス制御	情報投入系のUIでは、桁数の決まったフィールドへの入力で、その桁数に達すると自動的に次のフィールドへフォーカスを遷移するようにしているものが多い	キー入力をリアルタイムで取得できないWebブラウザではC/Sシステムと同様な機能を実装するのは極めて困難
ウィンドウサイズやレイアウト	一覧性や操作性向上のため、ウィンドウサイズを指定（場合によっては固定化）し、さらにスクロールしなくてもよいようなUIが好まれる	ウィンドウサイズはユーザーが変更可能。内容のレイアウトもスクロールを前提とするコンテンツブラウジングに最適化されている
クリップボードやドラッグ&ドロップ制御	ファイルアップロードを行なう場合、フォルダ内のファイルをドラッグ&ドロップで対応可能	Webブラウザの機能だけでは、フォルダ内のファイルをドラッグ&ドロップだけでアップロードすることは困難