

業務システムの最適解を目指す

Windowsシステム 拡充プロジェクト

株式会社NTTデータ

TERASOLUNA Server Framework for .NET

“低コスト”“高品質”は、アプリケーション開発者ならいつも耳にしている合言葉だろう。しかし、これを実現するのは難しい。特に大規模なアプリケーションでは、大人数による開発となるためさらに困難を極める。今回は、アプリケーション構造を定型化することで品質の均質化を図る、ASP.NET対応の拡張フレームワークについて紹介しよう。

社内のASP.NETアプリケーションの開発を支援

インターネットの登場以来、多くの企業がWebアプリケーションを導入し、さまざまなビジネスシーンで活用している。今回紹介する「TERASOLUNA Server Framework for .NET」(以下TERASOLUNA FW for .NET)は、NTTデータのSI案件で利用されているASP.NET対応の拡張フレームワークだ。

TERASOLUNA FW for .NETが完成したのは、.NET Frameworkの登場から2年後の2004年。製品担当の横宏晃

氏は、当時を次のように振り返る。

「NTTデータでは、大規模システムを中心にWebアプリケーションの受注開発を行っていますが、.NET以前のWindows開発ではWebアプリケーションの構築にActive Server Pagesを使い、スクリプトの記述など非常に手間がかかり、デバッグも大変でした。

そのような状況において、マイクロソフトから.NET Frameworkが登場しました。この.NET Frameworkでは、新たにASP.NETというWebアプリケーションの開発機能群が提供されました。そこで、この.NET Frameworkと自社

の開発ノウハウをもとにWebアプリケーション用のフレームワークを作れば、より生産性を上げることができるのではないかと考えました。このようにして誕生したのが、TERASOLUNA FW for .NETです。

現在NTTデータ社内では、この開発フレームワークと、開発プロセス、プロジェクト管理の3つで構成される、Webシステム開発の総合ソリューション“TERASOLUNA”を展開しています」

なお、ソリューション名の「TERASOLUNA」(テラソルナ)は、開発プロセス(TERA:地球)、プロジェクト管理(SOLARの“SOL”:太陽)、フレームワーク(LUNA:月)が三位一体となって提供されることから由来している。

フレームワークの構造

では、TERASOLUNA FW for .NETには、どのような機能が用意されているのだろうか。製品担当の崎山直洋氏は、まずキモとなるフレームワークの構造から話してくれた。

「TERASOLUNA FW for .NETのフレームワークは、UIの処理を行なう

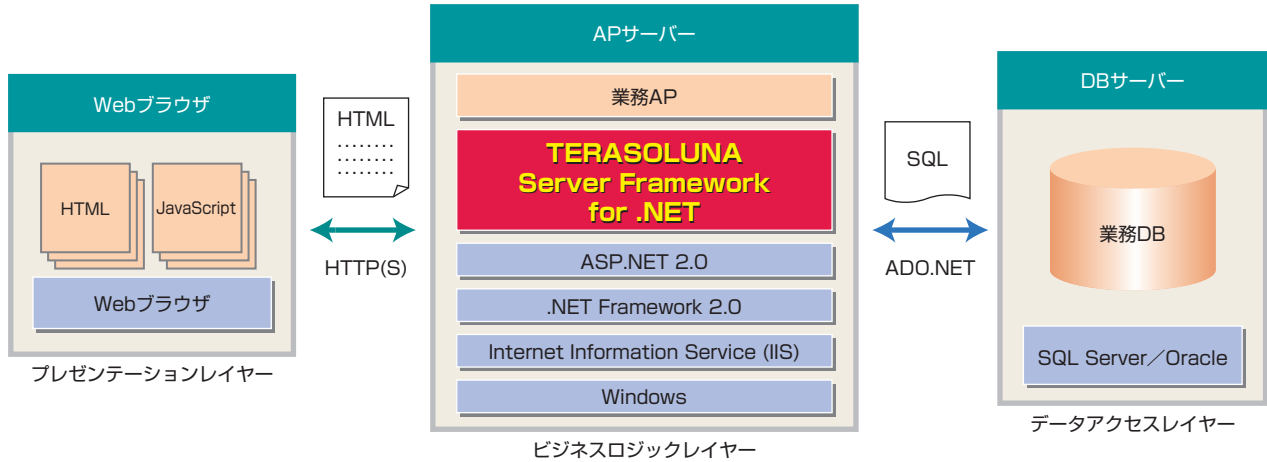


株式会社NTTデータ
基盤システム事業本部
システム方式技術ビジネスユニット
第一技術統括部 第一システム方式技術担当
横 宏晃



株式会社NTTデータ
基盤システム事業本部
システム方式技術ビジネスユニット
第一技術統括部 第一システム方式技術担当
崎山 直洋

図1：TERASOLUNA Server Framework for .NETのシステム構成



「プレゼンテーションレイヤー」、ビジネスロジックの処理を行なう「ビジネスロジックレイヤー」、データベースへのアクセス処理を行なう「データアクセスレイヤー」の3階層構造のアーキテクチャに規定しています (図1)。このようにレイヤー化することにより、容易に機能の追加/修正を行なうことができますし、単体テストがしやすいというメリットもあります。

また、各階層ごとにWebアプリケーションでよく利用される機能群が用意されています (次ページの図2を参照)。開発者はこれら機能を利用してWebアプリケーションの開発を行なうことにより、生産性は向上します」

さらに図2をみると、業務依存部分として設定ファイルが多いことがわか

る。これは一体何のためにあるのだろう。

「TERASOLUNA FW for .NETの開発ガイドラインには設計書 (Excelファイル) の書き方が示してあり、これに沿って記述したビジネスロジック、画面遷移、SQL文といった各種設計情報からツールを利用してXML形式の設定ファイルが生成されます。設定ファイルにはアプリケーションの挙動が定義されており、フレームワークに用意されている機能 (ビジネスロジック起動/画面遷移保証機能/SQL文管理機能) で利用します。設定を外部ファイルとしてフレームワークに紐付けておくことにより、ファイルの内容を変更するだけで、コンパイルすることなく機能の追加/修正が柔軟に行なえるという

メリットがあります」

一番苦労したのは……

最後に、TERASOLUNA FW for .NET開発時に苦労した点について聞いた。

「一番苦労したのは、テストです。品質の高いWebアプリケーションを開発するために利用されるわけですから、テストを繰り返しフレームワークの品質を高めました。

また、ASP.NETにはコンストラクタが公開されていないクラスがいくつかありますが、これらはインスタンスを作成できないので単体テストを行なうことができません。そのため、テスト用のアプリケーションを作成し、正常に処理を行なっているか試しましたね」

🎯 拡充のポイント

- ・ **アーキテクチャを3階層構造に規定。**
これにより、メンテナンス/テストを容易に行なうことができる
- ・ **Webアプリケーションの開発に必須の機能群を提供。**
これらを利用して開発を行なうことによりコスト削減