

# .NET 対応アプリケーションサーバ “ CrossMission<sub>TM</sub> ” CrossMission<sub>TM</sub> .NET-Compatible Application Server

平野 和順      多田隈 雄二

HIRANO Kazunori

TADAKUMA Yuji

東芝テック(株)は永年にわたり、POS(Point Of Sales)ターミナルや店舗サーバの分野において Windows<sup>®</sup>(注1)プラットフォームを採用し、自社の流通業向けアプリケーションソフトウェア(以下、アプリケーションと略記)開発のために独自のフレームワークを育成してきた。このフレームワークを基に、マイクロソフト社の.NET技術に対応した、アプリケーションサーバ機能を含む開発・実行環境を、商品名“CrossMission<sub>TM</sub>”として商品化した。CrossMission<sub>TM</sub>は、Visual Studio.NET(注2)及び.NET Framework(注3)上で、ソフトウェア開発者が直接プログラム構造やAPI(Application Programming Interface)を意識せず、純粋にアプリケーションの論理コード設計に専念できる、オブジェクト指向の開発環境を提供する。

CrossMission<sub>TM</sub> is the first Microsoft<sup>®</sup> .NET-based application server in the world. It offers a development environment and application execution environment for system engineers to build various systems on the Internet platform. Toshiba TEC Corporation has been developing its own framework for Microsoft<sup>®</sup> Windows<sup>®</sup> to develop point of sales (POS) terminals and store server applications. All CrossMission<sub>TM</sub> development tools are compatible with Microsoft Visual Studio .NET, while the middleware components coincide with the Microsoft .NET Framework and the Common Language Runtime (CLR). CrossMission<sub>TM</sub> combines extensive class libraries, standardized background services, and standardized graphical user interface components.

## 1 まえがき

専用ハードウェア、独自基本ソフトウェア(OS)を採用していたPOSターミナルや店舗のコントローラの世界にも、1990年代中盤に入るところからパソコン(PC)技術の波が押し寄せてきた。24時間365日営業するコンビニエンス店舗など、PCにとって過酷な動作環境であるが、ハードウェア及びソフトウェアのコストが圧倒的に低くGUI(Graphical User Interface)の優れたPC技術の採用は必然の流れであった。東芝テック(株)もWindows NT<sup>®</sup>(注4)の登場とともに本格的な取組みを開始した。

Windows<sup>®</sup>上で開発経験のないソフトウェア開発技術者を早急に戦力化するために、アプリケーション開発者が直接OSにアクセスせずに済み、作成したロジックが直ちにOS上で動作可能となるフレームワークを構築した。現在では当社の流通小売業向けシステムの80%以上に採用されている。

当社は、世界初のマイコン採用電子レジスター発売以来、流通業向けシステムのトップ企業の位置を占めている。マイ

クロソフト社(以下、MS社と呼ぶ)とも、OLE POS<sup>(注5)</sup>、ActiveStore<sup>(注6)</sup>などMS社の提唱する流通分野向け標準化技術の推進に関し従来から協力関係にある。2000年夏ころからMS社が“.NET技術戦略”を表明し、従来のOS技術をインターネット基盤上へ大きく方向転換する戦略を打ち出した。

当社は、豊富なWindows<sup>®</sup>向けシステム開発の経験から、OSとアプリケーションを接合する自社ミドルウェアの信頼性と有効性を認識し、従来社内向けであったフレームワークを積極的に外販することとした。.NET戦略協力企業の開発促進を狙ったMS社の.NET Early Adoption Programに参加し、MS社自身の.NET開発に並行して.NET対応開発を進めてきた。こうして、永年店舗システム開発で培ったWindows<sup>®</sup>プラットフォーム上でのクライアント/サーバアプリケーション開発のノウハウを、業界初の.NET Framework完全対応アプリケーションサーバ“CrossMission<sub>TM</sub>”として結実させ、MS社の開発ツールVisual Studio.NETとほぼ時を同じくして、2002年5月から市場へ投入した。

ここでは、CrossMission<sub>TM</sub>の技術的な特長を紹介する。

## 2 CrossMission<sub>TM</sub>の特長

CrossMission<sub>TM</sub>は分散オブジェクトを実現するために、

(注1)、(注2)、(注3)、(注4)、(注5)、(注6) Windows, Visual Studio.NET, .NET Framework, Windows NT, OLE POS, ActiveStoreは、米国Microsoft Corporationの米国及びその他の国における商標又は登録商標。

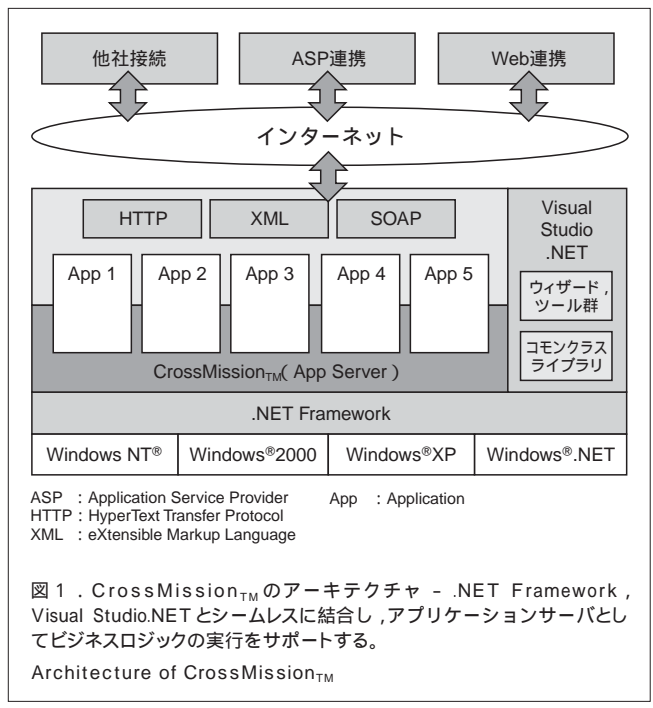
SOAP(Simple Object Access Protocol)を利用したメッセージングミドルウェアをアプリケーションサーバの中核に配置している。これにより、Windows®が持つマルチスレッドとイベントドリブンのパワーを最大限に生かした、高速かつ高効率なアプリケーションサービスの実装が可能となる。CrossMission™は、同期/非同期、RPC(Remote Procedure Call)、ブロードキャストなどの豊富な形態のメッセージングコミュニケーションと、優先度管理、負荷制御、排他制御、エラーリカバリといったアプリケーションサービス間のメッセージ連携を強力にサポートするコアサービスを提供している。

CrossMission™は、Visual Studio .NETと完全に統合されたウィザードベースによる開発環境を提供する。これまでWindows®プラットフォームでは、メッセージングミドルウェアをアプリケーション開発者が直接APIを意識せずに作ることはほとんど不可能であった。CrossMission™が提供するフレームワークでは、各アプリケーションをスレッドとし、業務ロジックの設計に集中した開発ができる。

またCrossMission™は、業務アプリケーションに要求されるタスクコントロールを中核とするユーザーインタフェースを簡単に設計・開発できるように、当社のPOSターミナル開発ノウハウから厳選したユーザーインタフェースコントロールを提供している。

### 3 CrossMission™の構成

CrossMission™は、次の三つの要素から構成されている(図1)。



- (1) アプリケーションサーバ
- (2) アプリケーションサービスフレームワーク
- (3) 開発環境    ウィザードとコントロール

#### 3.1 アプリケーションサーバ

CrossMission™のアプリケーションサーバ(アプリケーションの実行エンジン)の中核は、Message Exchanger, Flow Controller, Result Managerの3本のメッセージングミドルウェアで構成される。

また、そのほかの各種サービスを提供する次の7本のサービスが用意されており、機能概要は以下のとおりである。

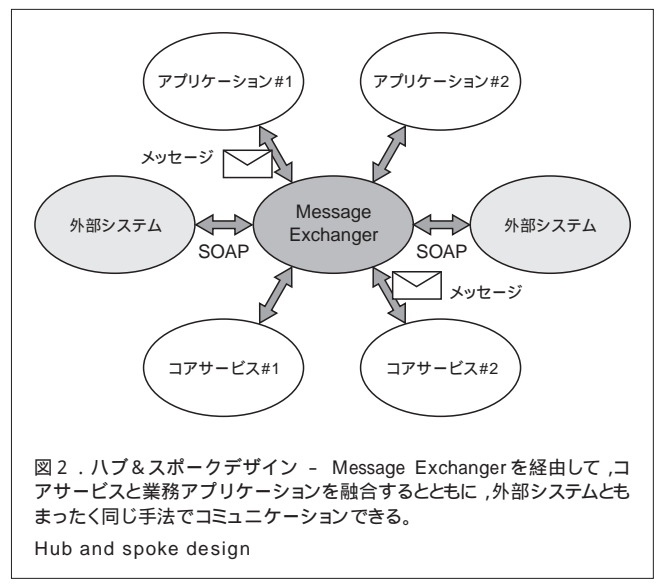
- (1) State Manager    複数プロセス共有の状態を管理
- (2) Schedule Manager    時刻ベースのスケジュール制御
- (3) System Controller    サービスの起動・停止・監視など
- (4) Process Manager    非.NET スクリプト言語を実行
- (5) XML Server    HTTPフロントエンド動作軽量 Webサーバ
- (6) XML Client    軽量 Web サービスクライアント
- (7) Data Converter    XML/RDB( Relational Data Base )間のマッピングサービス

これら10本のサービスは、ターゲットとする業務システムの必要性に応じて自由に組み合わせで使用できる。

##### 3.1.1 Message Exchanger

Message Exchangerは、CrossMission™が提供するメッセージングサービスの中心に位置し、すべてのメッセージの配送を受け持っている。CrossMission™では、アプリケーション間で送受信されるすべてのメッセージが、同一コンピュータ内のMessage Exchangerを必ず経由する“ハブ&スポークデザイン”を採用している(図2)。このため個々のプロセスがメッセージのあて先の場所を意識する必要はまったくない。

Message Exchangerが提供するメッセージ通信には、次



の三つのパターンがある。

- (1) ワンウェイ 送信元が送信先に対してメッセージを一方的に送付する。
- (2) リクエスト/レスポンス RPCと同様に,送信元のメッセージに対して,送信先からのレスポンスが必ず発生する。
- (3) 自由形式 送信元と送信先で自由にリクエストとレスポンスの送信を行う方式。前項と異なり,必ずしも送信先から送信元へのレスポンスメッセージで手順を終結する必要はない。

アプリケーションは,これらの3種類の通信手順を自由に組み合わせてサービス間の連携を作成する。リクエスト/レスポンスパターンの場合でも,Message Exchangerがメッセージングの中間に必ず介在するため,アプリケーションサービス自身は障害の影響を考慮する必要がない。特に送信先がリモートコンピュータに存在する場合に有効である。また,必要に応じてメッセージのキューイングが行われるため,メッセージの欠落や損失などに対し耐障害性の高いシステムが構築される。また,トラブルシューティングや開発時の動作確認をサポートするために,Message Exchangerと連動するプロセス間通信モニタアプリケーション(SOAPモニタ)も提供されている。

3.1.2 Flow Controller Flow Controllerは,CrossMission<sub>TM</sub>のメッセージングサービスに対し,簡単なスクリプト言語によってメッセージの送受信を制御できるようにする機能を提供する。

Flow Controllerの役割は,複数のアプリケーションサービスによるメッセージ連携が必要な場合に,経路指定,メッセージ配信方法の指定(同期,非同期),同時実行を実現するための排他制御によって,アプリケーションサービス間のメッセージ連携を力強くサポートすることである。Flow Controllerによるアプリケーションサービスのオーケストレーションが提供されるため,個々のサービスの関係を,疎結合でかつきめ細かなサービスレベルに保つことができる。

3.1.3 Result Manager Result Managerは,Flow Controllerとともに,サーバアプリケーションのオーケストレーションに重要な役割を果たしている。Result Managerは,Flow Controllerによるメッセージングサービスの実行結果を一元管理し,スクリプト単位での実行結果とエラー発生時のリカバリ処理を行っている。あらかじめ予測されるエラーについては,リカバリ動作を指定しておくことにより,自動的なりカバリ処理を実行させることができる。

### 3.2 アプリケーションサービスフレームワーク

CrossMission<sub>TM</sub>は,サーバアプリケーションをWindows<sup>®</sup>サービスとして登録,実行する。サーバアプリケーションの実行単位はスレッドとなる。スレッドはその生成方法から,

次の3種類のいずれかのモデルで表される。

- (1) ダイナミックスレッディングモデル このモデルでは,スレッドはメッセージ受信のつど,CrossMission<sub>TM</sub>により生成,実行される。もっとも標準的なスレッディングモデルである。このモデルはメモリ効率と実行効率の双方にバランスが取れたモデルである。
- (2) レジデンススレッディングモデル このモデルでは,CrossMission<sub>TM</sub>により,サーバアプリケーションの起動時に,あらかじめ設定した数のスレッドプールが生成される。メッセージ発生のつどスレッドを生成する必要がない反面,常時一定数のスレッドによりメモリが使用されるため,発生頻度が高く,並行動作が必要となるメッセージを扱うサーバアプリケーションに向けたスレッディングモデルである。
- (3) シングルトンスレッディングモデル このモデルではスレッドは1クラスにつき1インスタンスのみ作成され,メッセージはこのスレッドの実行権によって順番に処理される。その結果,アプリケーションは排他制御を意識する必要はなくなるが,そのトレードオフとしてメッセージの並列実行ができなくなるため処理効率は低下する。

CrossMission<sub>TM</sub>は,サーバアプリケーションのメッセージキューを監視し,実行を制御する。メッセージキューは,サーバアプリケーションの処理対象となるメッセージの種類別に設定してもよいし,サーバアプリケーション単位に一つのキューを共有させてもよい。キューは,キュー単位に処理の優先度を10段階に設定することができる。これにより,複数のキューにメッセージがキューイングされている状態で,どのメッセージを先に処理するかが決定される。例えば,応答が不要なメッセージであれば,優先度を低く,逆に応答が必要なメッセージであれば優先度を高く設定するといった制御をすることができる。なお,キューに格納された未処理メッセージは,プロセス終了時に順番に配列され,再起動後にキューに再スケジューリングするように設定することもできる。

### 3.3 開発環境/ウィザード

CrossMission<sub>TM</sub>は,Visual Studio .NETに統合された3種のフレームワークウィザードを提供している。

- (1) プロジェクトウィザード プロジェクトウィザードは,Visual Basic.NET<sup>(注7)</sup>,Visual C#.NET<sup>(注8)</sup>それぞれのアプリケーションサービスのパラメータを設定していくことで,プロジェクトを自動生成する。
- (2) メッセージウィザード メッセージウィザードは,サーバアプリケーションの実行単位であるワークスレッドのメソッド名と実行条件を入力することで,メッセージ

(注7),(注8) Visual Basic.NET, Visual C#.NETは,米国 Microsoft Corporationの米国及びその他の国における商標又は登録商標。

の受信によるワークスレッドの起動を制御するマップと、ワークスレッドクラスのソースコードスケルトンを自動生成する。

- (3) フォームウィザード フォームウィザードは、あらかじめ登録しておいたフォームのテンプレートから派生させたウィンドウアプリケーションのスケルトンフォームを自動生成する。通常、業務アプリケーションでは、プロジェクトや業務の種別により統一されたレイアウト、サイズ、コントロールを配置したフォームを複数使用する。フォームウィザードにそのような業務アプリケーションのテンプレートとなるフォームを登録することにより、速やかにアプリケーションを開発することができる。

これらの3種のウィザードにより、CrossMission<sub>TM</sub>に対応したサーバアプリケーションやクライアントアプリケーションの開発を容易に開始することができる。

### 3.4 開発環境 / コントロール

CrossMission<sub>TM</sub>は、業務用Windows<sup>®</sup>アプリケーションのユーザーインターフェース作成に有用な各種コントロールを提供する。これらは、当社がPOS端末や店舗サーバでその有用性を確認し磨き上げてきたコントロールコンポーネント群を基にしている。これらのコントロールのうちボタンを持つものは、あらかじめビットマップをはり付けられるように作成されている。ここで紹介したウィザードやコントロールのほかに、CrossMission<sub>TM</sub>が提供する開発支援ツールとして、設定ファイルを直観的に作成できる各種設定ファイルエディタが用意されている。

## 4 CrossMission<sub>TM</sub>導入の効果

開発基盤としてCrossMission<sub>TM</sub>を採用することにより、自動的に次のようなシステム特性を得ることができる。

### 4.1 システム性能の高効率化

CrossMission<sub>TM</sub>のメッセージングミドルウェアは、すべてを同期処理で実行する単純なWebベースのシステムと異なり、快適な処理レスポンスをもたらす。CrossMission<sub>TM</sub>の非同期アーキテクチャは、コンピュータ資源を遊ばせないでフルに活用することにより、顧客のハードウェア投資を最小に抑えることができる。

### 4.2 開発リソースの効率化

CrossMission<sub>TM</sub>を使用することにより、アプリケーションのビジネスロジックとユーザーインターフェースを分離して別々に開発することができる。したがって、アプリケーションの根幹であるビジネスロジックの作成に、より多くの開発パワーを集中することができる。

### 4.3 開発容易性

CrossMission<sub>TM</sub>が提供する各種ツールやVisual Studio

.NETに組み込まれる開発環境、オブジェクト指向技術によってもたらされたサービスロジックの差分プログラミングなどの仕組みにより、緩やかな学習曲線で最大の効果を上げることができる。

CrossMission<sub>TM</sub>ではメッセージングサービスを中核として配置し、ビジネスロジックの作成は受信したリクエストメッセージの内容によって問題領域の処理を行い(必要であれば)レスポンスメッセージを作成し応答する、というように極めて単純化されている。このことは、オブジェクト指向開発に不慣れなプログラマであっても、フレームワークの仕様(作成すべきサーバアプリケーションの動作)が理解しやすいため、すぐにCrossMission<sub>TM</sub>上の開発を始められる。

### 4.4 保守性

開発したビジネスロジック(サーバアプリケーション)は、Flow Controllerによってオーケストレーション可能なビルディングブロックとなる。このことは、新規サービスの追加や変更に対して個々のビジネスロジックのレベルでの修正が少なく済み、より容易に、Flow Controllerに対するスクリプトのレベルで保守していくことができる。

## 5 あとがき

CrossMission<sub>TM</sub>は、東芝テック(株)のWindows<sup>®</sup>開発技術をMS社.NET Framework上で実現した、アプリケーション開発のためのフレームワークである。NETがインターネットWebサービスの基盤技術を指向していると思われるので、CrossMission<sub>TM</sub>もWebアプリケーションサーバの視点から語られることが多いが、本来の目的としては、リッチクライアントからサーバまで広く利用していただけるアプリケーション開発のための基盤である。

一般的にはWebアプリケーションサーバとしての活用事例が増えるものと予想され、今後クラスタ対応、ロードバランス機能の強化を図っていく。またコンパクトフレームワークにも対応し、広く開発者の皆さまに愛される開発環境とすべく改善を続けていく所存である。

なお、CrossMission<sub>TM</sub>に関する詳細情報はURL <http://www.wxfw.net> から入手できます。



平野 和順 HIRANO Kazunori

東芝テック(株)流通情報システムカンパニー 営業推進統括部参事。流通小売業向け店舗システムのマーケティング企画に従事。

Toshiba TEC Corp.



多田隈 雄二 TADAKUMA Yuji

東芝テック(株)流通情報システムカンパニー 営業推進統括部参事。POSターミナル・店舗サーバ開発に従事。情報処理学会会員。

Toshiba TEC Corp.