

.NET フレームワークを活用した DM ソリューション

Development Methodology of DM Solution Taking Advantage of “.Net Framework”

柳 沢 博 一, 矢 野 玲, 内 村 友 亮

要 約 ShopMAX^{*1} は通信販売ビジネス向けのソリューションとして 20 ライセンス以上の導入実績を誇り、産業流通部門のソリューションの成功事例の一つとしてあげられているが、CSS 型のシステムであるが故に近年いくつかの課題が浮き彫りとなり、近い将来、市場競争力を失うことが予想された。ShopMAX での課題を解決し、通信販売ビジネス向けのソリューションビジネスをさらに拡大するためにコア技術として .Net Framework と XML WebServices を採用した IMPACT-DM R3 を企画し実装を行った。

本論文では、.NET Framework, XML WebServices を選択するに至った背景、製品の特徴、実装において工夫した点について解説しながら、今後の課題についても提言する。

Abstract In the commerce industry, ShopMAX is one of the successful solution software that has been served for more than 20 direct marketing companies. However, ShopMAX was expected to lose its competitiveness in immediate future caused by restrictions of CSS type system. In order to solve these issues and to expand our Solution business for direct marketing companies, we have planed and developed IMPACT-DM R3 adopting .Net Framework and XML WebServices as core technology.

This paper covers the reason of employing .NET Framework and XML WebServices the feature of the product, and inventions in the implementation stage, and also points some future challenges.

1. は じ め に

通信販売ビジネス向けのソリューションサービスは日本ユニシスの製造流通部門において、成功しているソリューションサービスのひとつであると言われている。この成功は、流通業界が不況の中においても通信販売ビジネスだけは成長し続けていることと、日本ユニシスが通信販売ビジネス向けのサービスを提供する部門を専任化した結果のノウハウの蓄積が強みとなっているからである。

しかし、近年通信販売ビジネス業界自体の環境変化により、我々が提供するソリューションサービスの形も変えていかなければならない局面をむかえている。我々は業界のビジネス環境の変化と IT の動向を分析し、.NET Framework と XML Web Services の技術の中核とした新たなソリューションを市場投入した。

2. 製品化の背景

2.1 通信販売ビジネスをとりまく環境の変化

日本通信販売協会の調査によると、2003 年度の通信販売業界全体の売上高は 2 兆 7900 億円となり前年度の 2 兆 6300 億円に比べ 6.1% の増加となり調査以来最高額を記録した。1996 年を境に一旦伸び率が頭打ちとなったが、1999 年以降再び年率 5% 程度の成長を続けている。小売業全体の売上が縮小傾向にあることを考えると健闘していると言える。

現在の通信販売ビジネスを取り巻く環境を整理してみると、四つの特徴が挙げられる。

1) インターネット利用の拡大

1 点目として「インターネット利用の拡大」があげられる。ひとりのインターネットブームは沈静化したが、これからも通信販売ビジネスの主要なメディアの一つであることには変わりはない。通信販売では単にサイト上に掲載した商品を販売するといった手法だけではなく、カタログに掲載した商品の詳細な情報や在庫状況をインターネットで提供したり、カタログ上の注文番号で注文を受け付けるといったメディアミックス型の使われ方にも注目しなければならない。

2) 異業種の市場参入

インターネット利用の拡大は、異業種の通信販売ビジネスへの参入を加速化させたとも言える。リアル店舗を持つ小売や商材のメーカーなどが次々と通信販売市場への参入をしている。迎え撃つ通信販売企業は商品、サービスなどで独自性を打ち出せなければ生き残れない環境になりつつある。

3) 利用者層の多様化

従来、通信販売ビジネスの主要な顧客層は 30～50 代の女性と言われてきたが、2) であげたインターネット利用の拡大や女性の社会進出などの社会環境変化により、通信販売の利用者層が以前に比べ拡大していると言える。こういった従来の通信販売の主要な顧客以外の利用層の動向についても今後無視できない存在となる。

4) 新たなメディアの台頭

4 点目に携帯電話やデジタルテレビの双方向サービスなどの新たなメディアの台頭があげられる。これらは現時点では通信販売ビジネスの主要なメディアとは言えないが、今後重要なメディアとなっていく可能性を秘めている。

これらの市場環境の変化に対応するために、通信販売企業側も生き残りをかけてビジネススピードを上げるとともに、新たなサービス・販売手法の模索やビジネスプロセスの改革を行っている。

我々もこれらの企業に対して高付加価値を提供するためのソリューション、サービスを開発していかなければならない。

2.2 ShopMAX での課題

IMPACT-DM R3 の製品化の前に通信販売ビジネス向けのソリューションとして ShopMAX が製品化されている。ShopMAX は CSS アーキテクチャによるオールインワンのパッケージシステムである。

ShopMAX は IMPACT-DM/FFII として 1999 年に製品化されて以来、20 ライセンス以上の導入実績があるが、近年いくつかの課題が浮き彫りになっていた。

1) 多様なビジネスモデルに対応することが困難

インターネットの普及や、異業種の通信販売ビジネスへの参入などにより、新たなビジネスモデルやアウトソーシング利用による新たなビジネスプロセスが創出されている。ShopMAX は元々、紙媒体を中心とした伝統的な通信販売ビジネスをモデルに設計されたシステムであり、インターネットを中核とした新たなビジネスモデルに対応しにくいシステムであった。また、オールインワンのシステムであるが故に、新たなビジネスプロセスに適用するためには膨大なカスタマイズコストと期間を要し、近い将来市場での競争力を

失うことが予測された。

2) 外部システムとの連携

通信販売ビジネスは元来アウトソーシングに依存するところが大きいビジネスである。商品の配送や代金の決済は、ほぼ外部の企業に委託しているし、ビジネスの中核であるコールセンターや物流拠点などもアウトソーシングしていることが少なくない。必然的に ShopMAX とアウトソーシング企業のシステムとの間でデータの授受が発生するが、業界としてインタフェース方法が標準化されているものは皆無であり、ShopMAX においても毎回この部分について手作りで対応している。

また、インタフェース方式もファイル形式のデータによるバッチ形式の送受信が中心で、ビジネススピードを上げることの妨げにもなっていた。

3) 保守性の低さ

2.1 節でも述べたとおり、通信販売ビジネスを取り巻く環境は劇的に変化しており、またその環境に置かれる企業も変化を強いられている。通信販売企業が新たな販売手法を構築したりビジネスプロセスを改変したりすれば、当然のことながらフルフィルメントを司る基幹システムにも手を入れなければならない。ShopMAX を導入した企業において我々は何度もこのような局面に遭遇している。ShopMAX は CSS のアーキテクチャを採用したシステムのため、保守性といった面において難があると言わざるを得ない。昨今のビジネススピードの速さに対応するためには保守性の向上が課題であった。

2.3 製品に求められること

通信販売ビジネスの環境の変化と、ShopMAX の課題に対応するために新しい製品の市場投入が求められており、三つの課題にフォーカスし開発を行う事とした。

2.3.1 多様化するビジネスモデルへの対応

従来の紙媒体を中心とした通信販売ビジネスのモデルだけではなく、ネット専業モデルやデジタルテレビなど新たなメディアへの対応が可能であることは必須要件である。単一のメディアだけでなく複数のメディアが混在する場合に、そのすべてに対して同一のプラットフォームから共通のサービスレベルを提供できることが理想である。

また、オールインワンのシステムではなく、対象とする企業のビジネスプロセスや共存すべきレガシーシステムに合わせて部分適用も可能であることも必要条件となる。

2.3.2 外部リソースの活用と標準化

商品の配送を委託する運送会社や代金決済を委託する金融機関などのシステムと連携することで、あたかも自社のシステムのように扱えることができれば、コスト、ビジネススピードの観点から企業にとって理想的である。これを実現するためには従来の EDI 型のデータインタフェースではなく、XML Web Services を利用したビジネスプロセスの再定義が必要であると考えた。また、ビジネスプロセスの定義において作成される XML スキーマを横展開していくことで業界標準のデータモデルを作成できるのではないだろうか。

2.3.3 ビジネススピードへの対応

ShopMAX の顧客適用では要件定義が終わって開発に着手してから本番稼動までに通常 6 か月から 10 か月を要していた。要件定義から 10 か月も経ってしまっているのは、ビジネス環境やその企業自身も大きく変化している可能性があり、その場合、出来上がったシステムはビジネスにアンマッチなものになってしまう。新たなソリューションの顧客適用では実装工程の期間短縮と、実装期間中のビジネス変化に柔軟に対応できるような開発手法の選択が必須である。

また、システム稼動後においても企業は刻々と変化しており、その変化に対応するためのシステムの柔軟性が必要である。

これらの要求に答えるソリューションの開発のために、アーキテクチャとして、.NET Framework と XML Web Services を採用することとした。

3. IMPACT-DM R3 のコンセプトと特徴

3.1 技術アーキテクチャ

3.1.1 .NET Framework と XML Web Services の採用

前章で挙げた要求仕様を踏まえて、.NET Framework による XML Web Services を採用したパイロットシステムで評価を行った。このパイロットシステムによって、新技術ゆえの様々な課題を事前に抽出したことにより、XML Web Services 適用のノウハウを蓄積できた。また、従来技術との差異を明らかにし、設計上の規約やコーディングルールなどの標準を定めることができた。本章では、パイロットシステムによって明らかになった課題から、IMPACT-DM R3 に課せられた技術要件を解説する。

1) 多様化するビジネスモデルへの対応

製品に求められることの一つである「多様化するビジネスモデルへの対応」については、1 提供サービスをサブシステムの単位として捉え、疎結合となったコンポーネント単位を販売するという設計モデルを採用した。パイロットシステムは XML Web Services 技術を利用して、各コンポーネントをサービス間連携とする、コンポーネント指向の考えにならって推進された。いわば XML Web Services 技術により、コンポーネント間のメッセージ伝達をネットワーク越しに交信できるようにしたが、トラフィック量の増加により予想していた処理効率を達成できなかった。語りつくされたことではあるが、XML 形式のデータに変換してやりとりする Web サービスでは、トランザクションが大量に発生するビジネスモデルには不向きであるとの結論に至った。

実行効率の観点から、同一システムのコンポーネント間通信は、異機種間や外部サービスのために必要なインタフェースを定義した XML Web Services 連携である必然性は薄いといえる。パイロットシステムからの教訓としては、イントラネットシステム内のコンポーネント間は XML Web Services を利用しないものとした。たとえば、OLTP(On-Line Transaction Processing) はサーバ内を参照するコンポーネントは直接参照としたり、バッチジョブ系でもコンポーネント間に直接参照を採用している。

2) 外部リソースの活用と標準化

上記で説明したサブシステムを一つのサービス単位としたコンポーネントは、そのコンポーネント自体が外部リソースの活用を想定しており、コンポーネント単位に別 Web サービスとし、データを格納するデータベースも別のデータベースに分割していた。従って、

環境設定によって疎結合にも密結合にもできる仕組みになっている。パイロットシステムの課題は、Web サービス間でトランザクションの信頼性をどのように保証するかにあった。トランザクションが成功した場合は呼び出された Web サービスすべてをコミットにし、トランザクション失敗の場合には呼び出された Web サービスすべてをロールバックして確実に元の状態に復旧する必要がある。

パイロットシステムでは、Web サービスの中でさらに Web サービスを呼び出すとき、トランスポートプロトコルに採用された HTTP だけでは、トランザクションの信頼性を保証することができないことが判明した。ここでの対策として、OLTP はサーバ側の処理を一つの Web サービスとして、その Web サービスの中でトランザクションを完結するような設計にした。基幹系システム連携に各システムを疎結合で設計させるのは当たり前だったように、今後の Web サービスの信頼性の確保という面では、トランスポートにメッセージングシステムを採用することになる。たとえば、Microsoft の分散トランザクションコーディネータや既製のトランザクションモニタ製品などの信頼性の高いトランスポートプロトコルの検討が必要になる。時間の関係上、これらの機能の検証に時間を費やすことができなかったが、外部リソースを活用する時点では検証を実施する事とした。

3) ビジネススピードへの対応

パイロットシステムの設計モデルは、オブジェクト指向分析/設計(以下、(OOA/OOD))を採用、データとプロセスを整理してアプリケーションを個々のサービスとして提供するブロック(すなわち、クラス)の組み合わせとして設計を進めた。これは整理方法に抽象概念を使うことで、工程間のスムーズなシフトチェンジによる納期短縮を図る狙いがある。あわせて ER 図、CRUD 表も作成、基本的にはデータは一事実一箇所とするデータ中心アーキテクチャ(以下、DOA)の手法も採用させることで、従来技術に慣れた設計者の理解を深めやすくしている。パイロットシステムで、クラス中心設計とも言い換えられる OOA/OOD にウェイトをおいて開発を進めた結果、設計者によってモジュール設計レベルの差異が生じていた。

上記の結果を踏まえて、IMPACT-DM/R3 システムの設計モデルでは、業務で取り扱われるデータを中心にシステムを捉えていく手法の DOA にウェイトをおいた。DOA にウェイトをおいたのは、IMPACT-DM/R3 設計基本方針に開発方法論の LUCINA^{*2} 準拠が謳われたためで、従来技術者が新技術への理解を深める副次的なメリットにもなっていた。OOA/OOD の主なドキュメントである「クラス図」「ユースケース図」「シーケンス図」は設計者間の意志疎通のビジュアルコンタクトツールとして留めた。DOA におけるエンティティと、OOA/OOD におけるエンティエィクラスの概念はほぼ同じのため、パイロットシステムのエンティエィクラスのモデリングを DOA のデータモデリングにほぼそのまま対応付けることができた。この新旧技術によるアプローチは、今後のインプリメンテーションにおいても、どの部品をどこに組み合わせればよいかを迅速に把握する一助になると期待できる。

また実装にあたり解決した課題を以下に補足説明する。

4) セキュリティ対策への取り組み

Web サービスを利用した製品として最低限確保しなければならない機能に、セキュリティ対策がある。情報漏えいや不正アクセスなどの脅威から、個人情報や取引情報をどの

ように守るかについて、次のように取り組んだ。

- ・アプリケーションのログイン認証は、LDAP 連携にする。
- ・SOAP メッセージ全てを SSL で暗号化する。
- ・クライアント、ウェブサーバ、外部リソースの Point-to-Point でお互いを承認、認証する。
- ・クライアントからデータベースサーバを直接のぞき見できないようにする。

5) イディオム

最後に、パイロットシステムで得られた、C#.NET の Tips 集について一部ご紹介する。未知の技術要素に対して、プログラム中で頻出するコーディングをパターン化したものを次のような Tips & Tricks にして技術情報として共有した。

文字列を大量につなげるときには StringBuilder クラスを使う。
コレクションに対するイテレータアクセスには Foreach を使う。
静的配列に要素を追加する場合は、いったん ArrayList に変換して追加し、そのあと静的配列に戻す。
インスタンス生成(new)はコストのかかる処理なので、極力減らすようにする。
データベース接続プールの利用を必須とし、接続再利用時のオーバーヘッドを低減する。

Tips & Tricks は、共通フレームワークで標準化できないような問題にも使用され、回避策としてのサンプルコードや設計指針を提供した。

3.1.2 フレームワーク

1) .NET Framework の採用

IMPACT-DM/R3 フレームワークは、基盤フレームワークに .NET Framework を利用している。 .NET Framework を利用するメリットとしては、設計、実装ガイドラインを MSDN で提供しており、クラスライブラリなどの部品が充実している点が挙げられる。開発環境は、Visual Studio .NET を利用して、容易なモジュール開発を実現した。さらにはオペレーティングシステム、データベースサーバ、アプリケーションサーバ、開発ツールなどの実行環境を単一社製品に統一することで、障害時の一貫解析が可能になる。

2) IMPACT-DM/R3 フレームワークの目的と設計方針

IMPACT-DM/R3 フレームワークは、LUCINA for .NET^{*3} に基づいて開発された Web アプリケーション用の WAT (Web Application Template for .NET) フレームワークを改良し、Web アプリケーションを迅速かつ円滑に開発でき、ユーザのニーズに柔軟に対応できるようにした基盤アプリケーションである。WAT フレームワークとは、DM 新ソリューション・パイロットシステム構築時に開発した Web アプリケーション専用のミドルウェアである。IMPACT-DM/R3 フレームワークの目的を以下にまとめた。

- アプリケーション開発者には基盤を意識させない
- 実装環境を提供することが期待できる
- 統一した設計コンセプトで、実装される

また、IMPACT-DM/R3 フレームワークで掲げた設計方針は、以下のとおりである。

- アプリケーションが持つべき機能とその配置を明確にする
(むやみに Web サービスを利用して実行効率を下げない)

- DOA+Layers に準拠しコンポーネント単位で開発する
 - WAT フレームワーク/LUCINA for .Net に基づき統一した開発モデルを設計する.
- 3) IMPACT-DM/R3 フレームワークのアプリケーションアーキテクチャ各層の要件

IMPACT-DM/R3 フレームワークの主な設計方針は、WAT フレームワーク及び LUCINA for .Net の設計方針に準拠しているが、Windows アプリケーションを考慮して、一部の修正および機能追加作業を行った。IMPACT-DM/R3 フレームワーク基本構造の各論理層の役割と概要を表1にまとめる。

表1 IMPACT-DM/R3 クラス概要説明

論理層	役割および概要
画面	リッチなスマート端末から Web サーバ内の Windows フォームへのアクセス手段を提供する
モデル	画面とビジネスロジックのインタフェースとして Model メソッドを提供し、それらの差異を吸収する
Web サービス	利用者の要求に対する各種処理を HTTP および SOAP で可能とする AP 層の XML Web Services を呼び出す Web サーバ上の XML データを統合して画面レンダリングする 冗長性/拡張性とともロードバランス による負荷分散で対応する
画面コントロール ビジネス	WEB サーバから呼び出される、ビジネスロジックが格納されたコンポーネント群および、WEB サーバで共通利用される各種データの基礎収集機能（バッチ処理）を提供する。AP 層からの XML Web Services 呼び出しによりビジネス処理を起動する 共通データベース アクセスツールでデータベース を利用できる。処理結果を Web 層へ戻す 冗長性/可用性ともロードバランス による負荷分散で対応する 拡張性、柔軟性は、ルールセット管理ツールでユーザ独自のビジネスの簡単な追加、またはカスタマイズ化をサポートする
データアクセス ビジネス	画面コントロールビジネスとデータアクセス間のパラメタ・インタフェース
データアクセス	データベースサーバのデータに対しての操作を実装。
データベース	AP サブシステムが利用するデータを提供する AP 層から必要によりストアドプロシージャを利用できる バッチ等(データ抽出)で直接、またはジョブ管理ツールを通ストアドプロシージャを利用できる。冗長性はクラスタリングにより対応する。拡張性はサーバ本体のリソース増強により対応する。柔軟性は共通データベース アクセスにより対応する

4) アーキテクチャ及びモジュール構成

本プロジェクトでは、アプリケーションが持つべき機能とその配置を明確している。詳細については割愛するが、クラス構造は図1のようになっている。

表2 開発プロセスにおけるチームと役割分担

チーム名	主な役割
分析チーム	業務モデルの詳細化、分割を行う分析フェーズを担当。データモデリングの精査。実装テクノロジーによって発生する制約をあまり考慮しない。
業務アーキテクチャチーム	経験者、有識者による過去の経験を元にしたアドバイスを業務モデルの観点から実施する。各業務サブシステムチームが、物理設計時に陥りやすいミスを事前に回避する。
技術アーキテクチャチーム (フレームワーク)	設計コンセプトを作成して、実装テクノロジーを踏まえたアプリケーション組み立ての指針を作成。フレームワーク開発、アプリケーションの共通基盤設計を担当。他に、指針となる実装ガイドラインの標準化も行う。プロトタイプ開発、技術検証、開発環境構築支援、構成管理。
各業務サブシステムチーム	分析フェーズのアプトブットをアプリケーションの設計モデルに落とし込む。テクノロジー制約を勘案しつつ、インプリメンテーション要員が実装できる形にする。
インフラストラクチャチーム	ハードウェア・ミドルウェアの設計を担当。他に、運用面の取りまとめ、開発時の構成管理。

3.2 ビジネスアライアンス（YSD の ASP サービス）

外部リソースの活用モデルを構築するための第一弾として、通信販売ビジネス企業の多くがアウトソーシングしている商品の配送業務について XML Web Service を利用したビジネスプロセスの再定義を行った。

この作業は我々 IT 企業だけでできるものではなく、運送業界の関連企業協力が不可欠である。我々は、多くの通信販売企業が配送を委託しており、かつ IT 化が最も進んでいると言われるヤマト運輸グループのヤマトシステム開発株式会社（以下、YSD）をパートナーとしてこの作業を進めることとした。

IMPACT-DM R3 では、YSD とのビジネスアライアンスにより、図2のようなサービスを製品版として提供している。

配達可能日時問合せ	… 発送日と届け先郵便番号から最短の配達日時が分かります。
配送状況問合せ	… 伝票番号、又は注文番号などのお好みのキーにより商品の配送状況を照会できます。
配達日時変更	… 出荷済の商品の配達指定日時を変更できます。
出荷確定／配達完了通知	… 出荷、及び配達が完了した荷物の一覧を入手できます。
お届け事前通知	… 商品のお届け予定日時をお客様に通知することができます。
産直出荷依頼	… 産直商品の伝票発行、集荷を依頼することができます。
商品引取依頼	… お客様から商品の返品などの際に、引取伝票発行、引取を依頼することができます。
運賃請求明細	… 運賃請求の明細をデータで入手できます。
投函状況問合せ	… メール便の投函状況が照会できます。
コレクト支払明細	… コレクトの支払明細をデータで入手できます。

図2 YSD 提供サービス一覧

3.3 機能概要

3.3.1 システム全体概要

外部リソース連携を確実に速く適用するという観点から、かつてのような画一的な方法だけでビジネスモデルを設計しているのではなく、システム手直しなどに迅速に対応できるような疎結合なコンポーネント指向となっている（図3）。

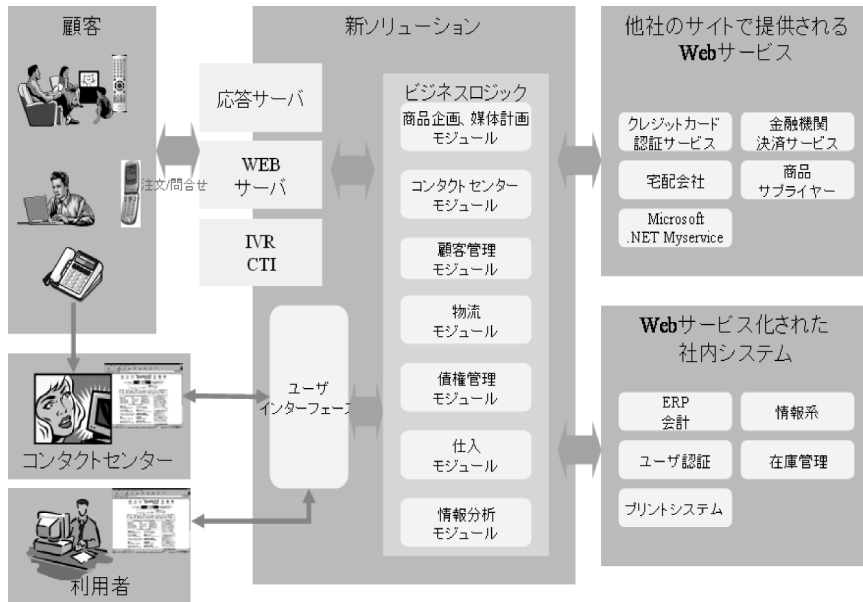


図3 システム全体概要

3.3.2 基本機能

基本機能に商品企画、媒体計画、仕入れ、コンタクトセンター、物流、顧客管理、債権管理、情報分析を用意して、さまざまな局面でのビジネス形態を幅広く支援している（図4）。

基本モジュール	
商品企画、媒体計画	商品企画、媒体計画業務を支援。 商品DBや需要予測、カタログ製作と連動した媒体計画の管理機能を提供。
仕入	販売する商品情報の管理業務の支援。 サプライヤーからの商品調達や納期管理機能を提供。
コンタクトセンター	電話でのリアルタイムな受注登録や、はがき/FAX/Web等からのバッチでの受注登録業務の支援。
物流	物流センターの在庫管理、入出庫業務の支援。
顧客管理	顧客の与信管理や基本属性管理のための支援機能を提供。
債権管理	顧客の売掛金の管理、未入金債権の督促業務を支援。 各種入金データの繰込み尾機能。
情報分析	ダイレクトマーケティングビジネスにおける基本的な情報分析機能を提供。

図4 基本機能

3.3.3 Web サービスによるインテグレーション

Web サービスによるインテグレーションでは、コンポーネント単位の販売が可能。コンポーネント単位に外部リソースのWeb サービスを利用することで、自社システムのように扱う

ことができる (図 5)。



図5 Web サービスによるインテグレーション

4. IMPACT-DM R3の課題と今後

4.1 インプリメンテーション要員の育成

IMPACT-DM R3 開発では、.Net C#を単一開発言語として選定した。VBやC++ではなくC#に決定した理由は、マイクロソフトがC#を.Net戦略におけるXML Web Servicesの中心的ツールとして位置づけている事による。また、単一言語選定による開発のねらいは、部品の共通化による保守性の向上にもあったが、それ以上に要員がC#の言語スキルのみで開発を行えるため育成が容易であると考えたことにある。C#はマイクロソフトが2000年に発表したばかりの歴史の浅い言語でありC#を使用可能な開発要員の数は十分とは言えなかった。にもかかわらず、C#開発要員の育成が容易と考えた根拠は、Javaとの類似点も多く、そもそもC言語系のコンパイラに似ている事や、構文自体がシンプルで短期育成が可能と判断したためである。C#単一言語開発は、JAVA要員やC言語要員からのシフトを比較的スムーズに行う事ができ、満足の得られる結果となった。IMPACT-DM R3の開発で採用した.NetやLUCINAは、コンポーネント指向や論理3階層モデルを推奨しているが、開発の初期段階でオブジェクト指向言語や論理3階層モデルでの開発経験者が少なかったため、前述の推奨方法について有効なTips & Tricksを持っていなかった。また、ほとんどの要員は、若く経験の浅いメンバーであったため、ビジネスルールや規約の実装、テスト方法など、システム構築の基礎を十分身につけていなかった。このため、IMPACT-DM R3フレームワーク(規約、標準化の実装、強制)、各種標準化規約書、共通ライブラリ、サンプルプログラム、Tips & Tricks集、ビジネスルール集などの整備、設計書及びコーディングレビューなどを通して要員育成を実施した。内工要員は経験3年以内の若いメンバーが中心であったが、今回の開発の経験を通して現在では顧客適用プロジェクトで重要なパートを担当できるほど、生産性が高く品質の良いシステム構築ができる要員として成長した。この実績からIMPACT-DM R3関連プロジェクト

では今後もこの育成方法をベースにする予定である。

4.2 分散拠点開発の課題

開発体制や開発環境にも課題がある。今回の開発は複数の拠点で開発を行ったが、分散開発は拠点が離れることが原因で開発者間のコミュニケーションが難しくなり、開発成果物の整合性を取る事が難しい。今回の開発では、この問題を解決する方法の一つとして、部品化開発を推進した。部品化にあたり、まずはロジックを可能な限り共通化し、それを再利用しつつ組み立てることとした。データアクセス層の部品は再利用が比較的容易なので、まずこの層の部品共通化を推進した。データアクセス層の共通部品化により、業務ロジック担当者はデータアクセスをあまり意識せず、業務ロジックのみに注力して開発が可能である。更に、業務ロジックも可能な限り共通化した。この方法によって分散拠点開発でも、各々が確実に部品を作成し、正しく組み立てれば、正常に動作するアプリケーション構築が可能と考えた。ただし、関連する部品間でのインタフェース・データベース仕様の変更や、一つの部品の遅れが開発全体の進捗に大きく影響するのでインタフェース・データベース仕様やスケジュール管理を厳格に行う事が要求される。また、経験から言って、コミュニケーションを密にとるために、設計段階においては主要な設計者を一箇所に集結することが必要と考える。尚、国内拠点はもとよりオフショア開発を行う上でも、この開発方法は部品単位の発注が可能であり適している事を加えて報告する。今後我々は、分散開発の距離によるコミュニケーションの困難さを解消するためにインターネット基盤によるメーリングリスト、コミュニケーションボード、テレビ会議システム、Web 会議システム、インターネット VPN による統合開発環境構築など、さまざまな技術の活用方法を検討し確立する必要がある。現在 IMPACT-DM R3 の顧客適用事例では、前述のインターネット活用例のうちコミュニケーションボードを除き実践している。

4.3 開発言語の課題 (パフォーマンス)

.Net で開発したアプリケーションは CPU パワーとメモリを大量に消費する。特にメモリについては、同様の画面プログラムの初期ロード時の値を例とすると VB の約 10 倍の量を消費している。Net の実装では期待するパフォーマンスを達成するためにメモリと CPU は十分搭載すべきである。開発環境では更に CPU やメモリの性能によりコードや画面、ヘルプの表示や、ビルドのパフォーマンスなど直接生産性の低下に影響するのでさらに増強すべきである。ADO.NET (SQLDataAdapter) を使用したアプリケーションは更に期待するパフォーマンスを達成する事が難しい。ネットワークへの多大な負荷が原因と考えられるが、ネットワークリソースは CPU やメモリに比べると進歩の速度が遅いと言われているので、この分野の進歩を期待する事は難しい。顧客のネットワーク環境を全て高速化できるわけではないので、我々は今後の ADO.NET の改善に期待している。その他 IMPACT-DM R3 開発では、C# による大量データのバッチ処理で期待するパフォーマンスが達成できないので、C# 単一言語による開発と言うポリシーを変更して SQLServer のストアドプロシージャを利用する事により、パフォーマンスチューニングを行った。

4.4 XML Web Service の技術的課題

IMPACT-DM R3 開発では、クライアントとして Web ブラウザではなくリッチクライアント

トを採用し、クライアントサーバ間の通信に XML Web Service を利用している。今後通信の標準技術になると予想される XML Web Service にもいくつかの課題がある。

4.4.1 セキュリティの課題

XML Web Service は、標準的に冗長性の高いテキスト (XML) を平文でを使用して通信を行うためネットワーク上でのセキュリティが甘い。一般には SSL の採用、.Net 間では WSE 2.0 の WS-Security を採用して解決できると考える。

4.4.2 ネットワーク負荷の課題

前項でも説明したとおり、XML Web Service は、標準的に冗長性の高いテキスト (XML) を平文でを使用して通信を行うためネットワークに負荷をかける。この課題の解決方法として、データの圧縮及び伸長が有効である。.Net ではフリーウェアの SoapExtension などを採用して解決できると考える。

4.4.3 初回起動時間の課題

XML Web Service は初回起動時間が数秒から 10 数秒と長い (初回起動時間とは、IIS の起動後最初に XML Web Service を起動するのにかかる時間)。この課題は、現在解決方法を模索中で、IMPACT-DM R3 では XML Web Service に最初にアクセスしたプログラムが犠牲になっている状態である。今後の解決策としては、IIS パラメタ、XML Web Service 記述のチューニング方法の確立と、毎朝の定期処理などで XML Web Service を立ち上げるプログラムの運用などを考えている。

4.4.4 タイムアウトの課題

XML Web Service は疎結合のため、これまでに IMPACT-DM R3 開発者が経験してきた、密結合の通信では問題にならなかった点がクローズアップされる事になった。問題はタイムアウトである。アプリケーションクライアントと DB サーバの 2 階層モデルでは、クライアントサーバの通信がタイムアウトを起こした場合、トランザクションはロールバックされるため、タイムアウトは他の一般的なエラーと何ら変わらなかった。しかし疎結合である XML Web Service は、タイムアウトがクライアント主導であり、タイムアウトが発生してもサーバプロセスは動作し続けるため、トランザクションの状態がクライアントからは保証できないのである。この課題は未だ完全には解決していない。IMPACT-DM R3 は機能として参照画面を多数用意し、これを利用してトランザクションが完結したかどうかオペレータが検査する事で解決できる。例えば受注登録でタイムアウトが発生した場合は、オペレータがお買い上げ照会など受注状態を参照できる画面で登録状況を確認し、未登録であれば再度登録を行うなどの対応が可能である。

4.4.5 ロングトランザクションの課題

XML Web Service は、ロングトランザクションが実装できない。IMPACT-DM R3 ではロングトランザクションを利用していないが、今後複数の XML Web Service をまたいでトランザクション保証が必要になった場合は、2 層コミットの採用などにより解決しなければなら

ない。

4.5 ビジネスアライアンスの拡大

XML Web Service を活用する事により、今後 IMPACT-DM R3 と他のシステムのビジネスアライアンスの拡大が見込める。例えば前章でも述べたように、YSD ではすでに通販支援の ASP を XML Web Service を利用して実装している。業界トップのビジネス方式に他の配送会社が追随する事は容易に予想される。また、クレジット与信、顧客信用調査、住所コード検索などの分野でもサービスが提供されれば利用する企業は少なくないと考えられる。外部で公開されているサービスを利用するのは逆に、通信販売企業は自社の XML Web Service を公開する事により、それを利用するデベロッパが購入サイトを構築してくれるなど、販売チャネルの拡大が期待できる。実際に書籍通信販売企業が XML Web Service の提供を開始している事例もある。

アライアンスの拡大は通信販売業界発展の大きな要因になると考えられるが、顧客情報保護が制度化され重要視されている昨今、セキュリティ面の強化は大きな課題として検討する必要がある。

5. お わ り に

IMPACT-DM R3 の開発には、コンポーネント指向、Web Service、.Net と全てが新しい、もしくは経験の無い要素技術が存在した。未経験の技術実装方法については誰も筋道を教えてくれない。ではどのように解決するか。それは早い段階でのリスク回避しか方法はない。パイロットシステム構築での技術検証、ドキュメントレベル/ドキュメント内容/運用の随時見直しなどである。また、アプリケーション技術については全社でナレッジを共有する仕組みの早期構築が望まれるところである。今後 IMPACT-DM R3 の部品を使った客先適用が始まるが、この開発での反省点を十分生かし、進化したソリューションを開発していきたいと考えている。

最後に、本稿の執筆にあたり、アプリケーション構築にご助力いただいた協力企業や社内の皆様にこの場を借りて心より感謝の意を表す。

-
- * 1 ShopMAX はクライアント/サーバー・システムアーキテクチャによる通信販売業務オールインワンパッケージである。日本ユニシス株式会社で開発。
 - * 2 LUCINA は、開発方法論とカスタマイズ可能なフレームワークを包含した、開発領域をトータルにサポートするサービスである。
 - * 3 LUCINA for .NET は、.NET プラットフォームをベースとしたコンポーネント指向の開発方法論である。

執筆者紹介 柳 沢 博 一 (Hirokazu Yanagisawa)

1988 年國學院大学経済学部経済学科。同年日本ユニシス(株)入社。通信販売企業向けのソリューション企画とその適用に従事。現在、日本ユニシス・ソリューション(株)産業流通第一サービス本部システム二部 DM 開発室所属。

矢 野 玲 (Akira Yano)

1991 年法政大学法学部法律学科卒業。同年日本ユニシス(株)入社。製造、流通の通販分野でアプリケーション開発、SE サービスに従事。現在、日本ユニシス・ソリューション(株)産業通流第一サービス本部システム二部システム一室所属。

内 村 友 亮 (Tomoaki Uchimura)

1997 年明治大学理工学部機械工学科卒業。1999 年日本ユニシス・エクセリューションズ(株)入社。製造、流通を中心とした SE サービスに従事。現在、日本ユニシス・ソリューション(株)産業通流第一サービス本部システム二部システム二室所属。