

1

ビジネス・プロセス・マネジメント(BPM)への挑戦

Taking on the challenge of Business Process Management (BPM)



日本アイ・ビー・エム株式会社
ソフトウェア事業部
ソフトウェア・テクニカルサポート
エンタープライズ・ソフトウェア技術部
コンサルティングITアーキテクト

尾崎 久栄

Hisae Ozaki

Consulting IT Architect
Enterprise Technical Sales
Software Sales Support
IBM Software Group
IBM Japan, Ltd.

企業が今まで開発して資産として持っているビジネス・オブジェクト(プログラム資産)は多数ありますが、通常それらは個別開発されているため、個別最適を目指したシステムとなっています。しかし全社的な観点でビジネス・プロセスを考えると、個別のシステムを横ぐしに結び付けることが重要です。個々のシステムを有機的に結び付けることによって、ビジネス・モデルの Turn Around Timeを改善し、お客様へのサービスを向上させ、プロセスの自動化によってコストを削減することも考えられます。このようにビジネス・オブジェクトを互いに結び付ける技術をビジネス・プロセス・マネジメント(BPM)と総称します。BPMのコンセプトは分かりやすく、お客様にも広くお勧めできるものですが、個々のお客様案件を考えると、従来はメリットの面から適用を躊躇するケースがありました。本論文では、このBPM適用上の問題点や、その代表的な適用分野と見られるB to Bについて考察を加えます。

Companies have hitherto developed many kinds of business objects (program assets) which they now hold as their assets, but since these are usually developed individually, they constitute systems each with optimum features in its own right. But from the standpoint of business processes viewed from the perspective of a company as a whole, it's important to link individual systems on a horizontal axis. Organic linkage between individual systems will bring about improvements in the turn-around time of business models, will make it possible to provide customers with better services, and will reduce cost through the automation of processes. Methods for creating mutual links between business objects are known generically as Business Process Management (BPM).

The BPM concept is easy to understand and can be widely recommended to customers. But despite this, consideration of individual customer projects often results in indecision as to whether to go ahead from the standpoint of the advantages involved. In this paper we take a look at the problems involved in application of BPM and at the "B to B" area in which these methods are most commonly applied.

1.はじめに

e-ビジネス、Java™、XML、Webサービスなど、ソフトウェアの世界では、インターネットを利用した技術・製品が次々にその覇権を狙って登場しています。これらの技術をお客様が検討中の案件に結び付け、提案し、アイデアを凝らして夢を売り、それをソリューションとしてお客様と一緒に具現化していくのが、われわれITエンジニアの仕事です。

筆者は日本アイ・ピー・エムのソフトウェアのセリングを担当し、特にビジネス・プロセス・マネジメント(以下、BPM)の動向を、B to Bへの展開と一緒に考えています。このBPMの波は、海外ではガートナー・グループを中心としたコンサルタント会社が、アクセス・ハブ・コンセプトに続く第2の基幹コンセプトとして推し進めているもので、既に30を超えるベンダーからその製品が発表されています。

しかし、国内におけるBPMへの取り組みを考えると、その実施例はまだ数少なく、特定のワークフロー業務でお使いのお客様が散見される程度であり、本格的に組織を超えて企業間/企業内のプロセスで結び付けている事例を知りません。ビジネス・オブジェクトとビジネス・フローを分けることにより、今後の企業統合/統合解消、サプライヤーの変更・追加といったビジネス・フローの変化に容易に追従できるBPMのコンセプトは、お客様には理解いただけるものの、個別のプロジェクトを考えると、ビジネス・フローを独立させて開発することが得策なのかどうか判断に迷うケースが少なくありません。

本論文では、今後の展開が予想されるBPM、B to Bについての適用の可否、現状の問題点、弊社の製品の特質などを整理します。

2. BPMその価値と役割

2.1. BPMとは

BPMの考え方は明確です。プログラムを作成する場合を考えてみるとその構成要素は

- アプリケーション層(プログラム・ロジック部分)
- データ層
- プレゼンテーション層(ディスプレイ画面に表示する部分)

の3層に分けて設計することが常識です。従来のホスト・システムであれば、CICS®のBMSマップや、IMS™のMFSがプレゼンテーション層に相当し、Web開発で考えれば、JSPがその役目を担っています。「データ層」にはデータベースがあり、データ構造の持ち方をER図などで分析しながらデータベース設計を進め

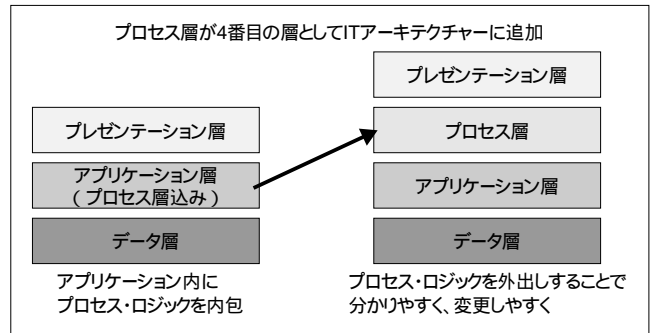


図1. プロセス層の独立

表1. プロセス層とアプリケーション層の分離例

プロセス層	アプリケーション層
If ローン金額 > 10万円 Then	取引状況チェック
Else	承認処理

るやり方が定着しています。

BPMは、残された「アプリケーション層」をさらに分割することでメリットを得られるのではないかという発想です。アプリケーション層を、プロセス層と(より機能面に特化した)アプリケーション層に分けて考えるのです(図1)。

例を挙げて説明してみましょう。銀行で、お客様からローンの相談があった場合、その金額が10万円以内であれば無条件に承認し、それ以上であれば取引状況をチェックするプログラムを作成するとします。今までは、このIf ~ Then ~ Elseのロジックをプログラム内でコーディングして提供してきました。これを表1のようにプロセス層とアプリケーション層に分けることで、アプリケーション層はより機能に特化したプログラムとなり、ほかのプログラムから利用できるようにクラス化することが可能となります。

では、プロセス層から見たメリットは何でしょうか。今まで一つだったアプリケーションがプロセス層とアプリケーション層に分かれるため、管理の手間が増えて困るという側面もあるのですが、その効用はプログラム保守の軽減にあります。例えば上記の例で、管理基準が変わって、無条件貸し出しの金額が10万円から50万円に変更する場合を考えてください。通常、このような値はテーブル化させ、プログラムから独立させて設計するのですが、ロジック内にコーディングされている想定とします。プログラマーはプログラム・ソースをにらみながら、どこにそのロジックがあるのかを見つけ出して修正し、テストを行います。プログラムの開発手法の汎用化が進んでいても、個人の素養による特性がまだまだあり、他人が作成したプログラムの保守には非常に時間がかかります。かつ大きなプログラムの場合、その修正の影響がほかのプログラムに影響しないかどうかの判断が非常に難しく、かなり気を使います。上記は簡単な例ですが、金額の変更が幾つのプログラムに影響するかどうかの分析も大変です。

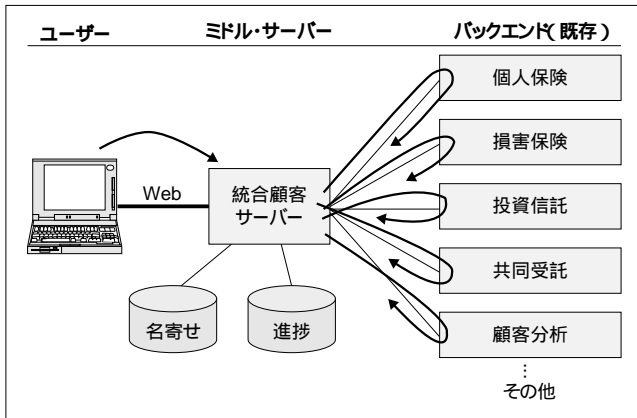


図2. 統合顧客システム略図

表2. ワークフロー化する場合のお客様のクレーム例

	お客様	IBM
スキル	ワークフロー製品の知識を新規に付けなくてはならない。	ワークフロー定義はユーザー・フレンドリーにGUIで簡単に定義できます。
開発生産性	従来の開発方法で進めた方が効率は上がる。	開発生産性は最初慣れるまでは、それほど期待できないかもしれませんが、保守の生産性はワークフロー製品を利用した場合、非常に高くなります。
必要性	わざわざワークフローを分けるようなシステム要件はないのでしょうか。	本当にそうなのでしょうか。業務事例を考えて検討しましょう。
既存システム・インターフェース	既存システムをワークフローと連携させて稼働させるには、既存システムに変更を加える必要はないのでしょうか。	既存システムとのインターフェースは標準的なアダプターとして開発するのが普通です。ベンダー開発のアダプター製品がそのまま利用できる場合もあります。この部分の開発は、ワークフローによる開発でなくても必要です。
パフォーマンス	毎秒数十トランザクションの処理をさせたいがワークフロー製品で対応できるのか。	現状は毎秒数件程度がワークフロー製品の取り扱いができる範囲ですが、これはシステム設計とハードウェア/ソフトウェアの進展により、数年で解決できるレベルになると確信しています。
実績	国内でのBPMの実績を聞かない。わが社の基幹業務を実績のない製品で構築する危険を冒すことはできない。	海外での実績は数多くあります。開発部門との直接インターフェースや設計レビューを実施しますので、ご安心ください。
価格	BPM製品は高い(最低2,000万~5,000万円程度かかるのが一般的)	SAP、i2、AribaなどERP/SCM製品も同様な値段です。システム開発総額に占める比率はそれほど大きくありません。今後のシステム開発でも利用できます。将来性を買ってください。

プロセス層を独立させれば、アプリケーション層には何ら手を入れずに済みますし、プロセス・ロジックは個性に依存しない分かりやすい形で表記できるので、初心者でも簡単に保守できます。

2.2. BPM適用上の問題点

問題は、BPMを実際にプロジェクト案件に適用しようとする、数多くの障壁があることです。

例えば「統合顧客サービス」を検討中の金融のお客様があったとしましょう。統合顧客とは、生命保険・養老保険・自動車保険などの複数の商品を購入されている一人の顧客が、それぞれ異なる口座番号で別々に管理されているために全体像をつかみきれず、顧客のライフ・サイクルに適合した提案をできない状況を解消するためのシステムです。この場合、個々のシステムは既存のものを尊重し、新たに作成する統合顧客システムは別と考えていました。

システム・イメージとしては図2のような形になります。

お客様が統合顧客システムを構築する際に、BPMのワークフローを独立させて開発するためのIBMの言い分とお客様の要求を対比してみましょう(表2)。これを見て納得がいく部分と、まったく的外れな回答だと合点がいかない部分があるのは、お客様の事情により異なるのでやむを得ません。ただ、共通して言えるのは

- BPMを今から実現する必要性があるかどうか。
 - BPM製品への投資の対価が報われるかどうか。
- の2点になります。

3. 企業間のビジネス連携

3.1. B to Bの現状認識

B to Bの企業間ビジネス連携の現状はどのようになっているのでしょうか。「B to B」という新しい言葉の印象から、これまでは企業間のビジネスがなかったかのように思えるかもしれません

表3. 従来のB to Bデータ交換のやり方

(1) 自社システムを提供	アプリケーションが入ったPCと専用ネットワーク込みで他企業に提供。このPCから自社システムに情報入力してもらう。企業内設置ATMや証券端末がその一例。
(2) VANサービスの利用(ファイル転送)	通信会社、コンピューター・メーカーが提供したファイル転送サービス。VAN経由で24時間365日データの蓄積交換が可能。
(3) 業界標準プロトコル	全銀手順などデータ交換プロトコルを業界ごとに規定し、これに合致したソフトを開発提供し、中立機関として協会を設立して統合管理する方式。

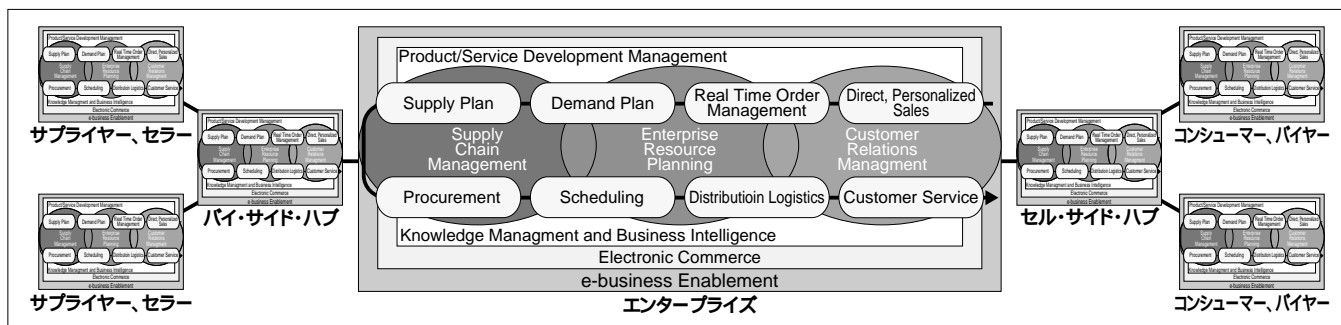


図3. eマーケットプレースの構図

が、それは間違いです。ビジネス連携とまではいえないものの、数多くのお客様で、系列企業を含めた他企業とのデータ交換システムを運用しています。まとめると表3のようになります。

これらの方式は、インターネットの登場でその様相をまったく変えてきました。従来のシステムには、業界独自のインターフェースしか持たなかったり、加入制限があり、利用代が高価で、実施までにある程度の準備期間が必要という欠点がありました。一方、インターネットは中小企業でも簡単かつ安価に利用できるという特徴を備えていることから、それを利用したB to Bは世の中の主流となってきました。そして現在、多くの企業(グループ)で実施しているのが自社製品のWebポータル・サイトと、e-マーケットプレースによるB to Bです(図3)。

3.2. BPMから見たe-マーケットプレースの問題点

B to Bとは、企業間の商取引を電子的手段で実現することですが、その目的は「オンライン・トランザクション処理で二つ以上のビジネス・プロセス間をプログラムなどにより連携/統合処理させる」ことにあります。

ところがe-マーケットプレースはコマースの部分に焦点を当てており、ユーザーはあくまでも人間が操作するWebブラウザであることを前提としています。ブラウザ画面でいろいろな検索条件を入れながら、数量・納期・価格・支払い条件をクリアした取引先企業にオーダーを飛ばすことを想定しているために、このマーケットプレースは自社システム(アプリケーション・プログラム)から直接利用できる形にはなっていないのです。人間経由のシステムでは、オーダーに必要な情報を入手し、目で確認し、マーケットプレースで注文し、その結果を自社システムに入力という処理を人間が行わなくてはなりません。担当者が急病で休んだ場合や、ほかの業務で忙しくてマーケットプレースへの入力が遅れてしまうと、この企業間連携のシステムのTurn Around Timeは人間に依存して遅れることになります。

B to Bのプロセスの連携では、人間の処理を極力避ける設計が一般的であり、通常処理はすべて自動でプロセスが流れ、何か途中で異常が発生した場合だけ人間系に通知して、後のリカ

バリー処理を任せると設計が好ましいでしょう。

ここで、B to Bのシステム連携において、自動化がいかに大事かということを簡単に紹介しましょう。

表4は銀行業務の1トランザクション単価の調査結果ですが、人間を仲介したシ

ステムと、人間の仲介を必要としないシステムには大きな差異があります。インターネットを利用した単価は、銀行側の設備負担が不要なため、支店窓口処理で17倍、コール・センターで24倍のコスト差となります。人間の処理を必要としない音声応答でも5倍、ATMを利用した場合でも7倍になります。近年、ホワイトカラーの生産性を上げることが国内の労働界の課題といわれていますが、企業間のシステムの自動連携が大きなコスト削減効果を生むことが理解できます。プロセスの連携はできるだけ自動化させて、システムの効率を上げ、Turn Around Timeを短縮させ、お客様満足度を向上させ、結果的に競争他企業と差別化するのがBPMの目的です。

表4. 銀行業務のトランザクション単価

チャネル	単価
インターネット	\$0.09
ATM	\$0.26
音声応答	\$0.32
支店 窓口	\$1.56
コールセンター	\$2.12
Eメール	\$4.78
支店 - ベース	\$5.30

3.3. BPM企業連携では何を規定しなければいけないのか
企業間でビジネスを協業する場合、お互いに気を付けるべきことは何でしょうか。国内では当たり前であった同属企業や関連会社間の連携という色彩を超えて、今まで知らなかったような新興企業や海外企業との連携も、その対象になることです。

e-マーケットプレースのように、途中でWebブラウザを経由したB to Bであれば、企業間のシステム連携は、ブラウザでいったん途切れるため、互いの企業のビジネス・プロセスの中身について意識する必要はほとんどありません。

しかし、企業間のビジネス・プロセスが連結して自動化されるBPMの世界になると、次の3点が重要となります(表5、図4参照)。

- プライベート・プロセスとパブリック・プロセスの管理

- データ形式の規定
- データ・フローの規定

表5. 企業間BPMのキー項目

<p>プライベート・プロセスとパブリック・プロセス</p>	<p>図4を参照してください。 企業間のデータ・フローの定義をパブリック・プロセス、一企業内のビジネス・プロセスをプライベート・プロセスと呼びます。プライベート・プロセスは企業独自のプロセスなので、自社のプライベート・プロセスは自由に設計・管理ができます。他社のプライベート・プロセスは、そのプロセスがどのように処理されているのかはまったく分かりません。そのプロセスの入力/出力データがパブリック・プロセス定義として分かるだけです。パブリック・プロセスは企業間のデータ・フローを規定します。 このように、企業間BPMでは、プロセスを連携するにも、互いのプロセスがブラック・ボックス化されて、自社のプライベート・プロセスだけが加工できるような機能が必要になります。</p>
<p>データ形式</p>	<p>業界ごとにデータ形式の標準が設定されつつあります。金融業界のFIXやRosettaNetがそれに相当します。Web環境を意識してXML形式のデータ形式を標準とするようになっていきます。 AribaのcXML、CommerceOneのxCBLのほか、BizTalk、Framework、ebXML、OBIなどのXML規定があります。データ形式の設定をどうするかについては、業界標準の設定を待つか、業界標準を参考にした独自フォーマットで取引を行うかどうかの検討が必要になります。</p>
<p>データ・フロー</p>	<p>企業間のデータ・フローを規定します。FIXやRosettaNetではデータ形式だけでなく、データ・フローのレベルまでの規定をしています。例えば、</p> <ul style="list-style-type: none"> • 注文処理を出す • 注文受け取り(ack)を返す • 3時間以内にキャンセルしない場合は自動発注される。 • 出荷情報を送る。 <p>などのデータ・フローを標準として規定します。データ形式と同様、業界標準の設定を待つか、業界標準を参考にした独自フォーマットで取引を行うかどうかの検討が必要になります。</p>

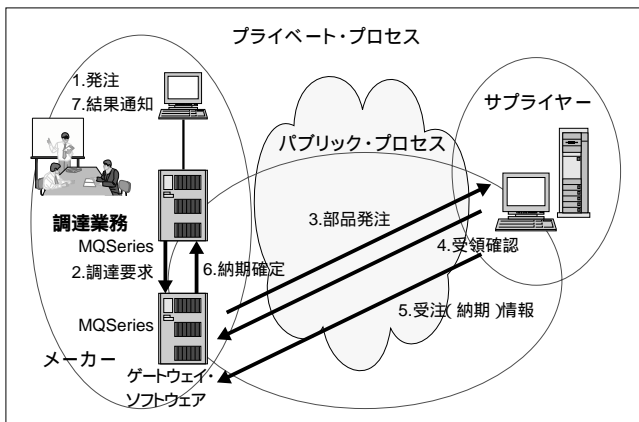


図4. パブリック・プロセスとプライベート・プロセス

4. BPM関連製品とその特質

ここではBPMをサポートするソフトウェア製品について、どのような用途にどのような製品が適用できるのかという観点で論じます。

4.1. ドキュメント中心型とプロセス中心型

BPM、ワークフロー製品は、ドキュメント中心型とプロセス中心型の二つに大別できます。前者がロータスドミノワークフロー(以下、ドミノワークフロー)、後者がMQSeries® Workflowになります。ドミノワークフローは文書を起票して、それを必要とする承認者に回しながら業務を遂行する方式のワークフローで、一般にはPeople Oriented Workflow(人間主導ワークフロー)といわれています。身近な例では、交通費の申請承認業務や、休暇取得・出張申請承認業務などが該当します。人間を経由した文書は最終承認された時点で検索可能な形態で保管されます。

MQSeries Workflowはプロセス間の自動化、すなわちアプ

表6. MQSeries Workflowとドミノワークフローの比較

比較項目	MQSeries Workflow	ドミノワークフロー
タイプ	プロセス中心型 アプリケーション連携ワークフロー(処理の自動化)	ドキュメント中心型 人間主導ワークフロー(意思決定プロセスの明確化)
競業製品	Vitra、Tibco / InConnect、... HP、Oracle	MS、Collaborative、Add-Ons Keyfile、Action、JetForm、...
特徴	スパンをまたがる複数のアプリケーションを連携させて処理を自動化させる。 トランザクショナルな処理が可能。	人・組織をまたがる個人の流れを制御。ドミノを利用したフォーム文書・セキュリティ設定が容易。 トランザクショナルな処理が可能。
技術	MQSeries (情報の連携) DB2® / UDB (情報の管理)	ロータス ノーツドミノ
代表事例	社員採用 • 採用者を給与/人事システム登録・苦情処理 • 顧客データベース、苦情データ(イメージ)連携 B to B • トランザクション処理連携 • Java業務開発連携	社員採用 • 承知 • 採用試験案内、面接結果報告・苦情処理 • 文書管理、トランザクション業務 B to B • ドミノ文書をWebで公開
人の介入	少(無)	多
ユーザー	一般事務職向き(自動処理)	技術職向き(意思決定)
フロー	複雑	単純
処理性能	進捗情報はUDBに逐次登録される仕組みなので、ここがネックとなる。AIX® S80-24way RS/6000®で毎秒180アクティビティーが実績値。1処理が6アクティビティーと仮定すると毎秒30トランザクション程度となる。毎秒数百トランザクションには不向き。	1処理1文書処理になるので、毎秒数百トランザクション程度が目安。

リケーション(プログラム)間のデータ渡しと自動起動、そしてその状況モニターを目的としています。Application Integration Workflow(アプリケーション連携ワークフロー)とも呼ばれます。アプリケーションの中には人間の処理も含まれると考えられ、MQSeries Workflowの処理途中に人間系の処理が入ったワークフローを設計することも可能です。しかし、ロータス ノーツドミノ(以下、ノーツドミノ)のような、きれいで見栄えの良い文書を作成するには手間が掛かり、一つの文書を複数の観点(View)から表示したり、フィールド単位にセキュリティをかけることも不得手で、それが必要な場合は、別途開発が必要になります。表6にその特徴をまとめます。

処理性能面を見ると、現状では、どちらの製品も毎秒数百といったトランザクション系のシステムに対応できていません。人間中心のノーツドミノ系業務では、毎秒数百のトランザクション処理を考慮する可能性は少ないのですが、システム間の自動連携を目指しているMQSeries Workflowでは問題になるケースもあります。また、この値は、ハードウェア性能とソフトウェアのバージョン・アップによって、今後改善されていきますから、常に最新情報を入手して判断してください。

4.2. 企業間連携(B to B)への対応

昨今の企業間連携の波の大きさに対応して、B to Bに特化したBPM製品が登場してきました。4.1節で紹介した製品をB to B環境でも使用できるのは事実ですが、その管理範囲は自社の企業内のビジネス・モデルに限定されます。他社のビジネス・プロセスを自社のビジネス・プロセスと結び付けて管理するBPM製品として、WebSphere® Partner Agreement Manager(以下、PAM)とWebSphere Business Integrator(以下、WSBI)があります。

表7. WSBI主要コンポーネント機能

コンポーネント	機能	ベース製品
TAM (Trust & ACCESS Manager)	アクセス制御	Tivoli® Secureway Policy Director
PAM (Partner Agreement Manager)	B to B企業間接続 B to BのEDI接続	Peregrine社のPAM D(Data Interchange)
IM (Interaction Manager)	ユーザー環境(PC、携帯電話、PDA)に合わせた個別画面表示	WebSphere Server WebSphere Personalization
BFM (Business Flow Manager)	ビジネス・フロー制御	MQWF, MQAK
IDM (Information Delivery Manager)	プロセスの実行指示とそれに必要なデータ加工	MQAQ, MQSI
SM (Solution Manager)	全体管理・監視(console) 開発環境の提供	MQAQ, WebSphere Studio

表8. PAMとWSBI比較

比較項目	PAM	WSBI
特徴	B to Bを初めから意識したBPM製品 現在V4で数多くの実績を持つ。	PAMをB to Bインターフェースで 採用した総合プロセス管理・開発 パッケージ製品。
開発元	Extricity社(Peregrine社がその後 買収)製品をOEM供給。	IBM
出荷実績	2000年12月(IBMからOEM販売) 1997年より出荷(Extricity社) 300件以上の導入実績。	2001年6月正式出荷 (2000年4Qから限定出荷)
稼働環境	Windows® / NT®(UNIX®版は開発予定)	Windows / NT(UNIX版は開発予定)
製品アダプター	MQ, FTPが標準 SAP, Oracle, BAAN, Peoplesoft, i2など	同左(PAM利用)
接続チャネル	PAM接続、ブラウザのみの接続 XML, RosettaNet, EDI channel可能	同左(PAM利用)
開発方法	PAMプロセス定義ツール	Solution Studio + 左記PAMツール

WSBIは昨年から注目を浴び、出荷が待たれていた製品であり、プロセスの開始から終了に至るすべてのプロセスを管理統合する機能を提供します。このプロセスの範囲は一企業を超えることが可能です。製品が提供している機能とその範囲は非常に大きく、その主要コンポーネントは、表7に示すように多岐にわたります。このコンポーネントの一つとして採用されているのがPAMであり、製品の関係としては、WSBIがPAMを含む包含関係となります。製品の評価はまだできない状況ですが、プロセスの開始から終了までを一貫した開発手法で構成していくことが可能であり、その意義は今後さらに大きくなると考えます。

PAMは3.3節で述べたように、自社企業内のプライベート・プロセスと企業間のデータとデータ・フローを定義するパブリック・プロセスを分けて定義できる製品です。IBMは、今まで企業間の連携には非同期通信の基本ミドルウェアMQを中心に考えてきましたが、簡単にMQに対応できないB to B環境では、PAMのようにプロセス自体は何ら変更せずに、対外接続方法の選択肢の多い製品が適切という判断があったと理解しています。

(注) 2002年BPM製品体系は、CrossWords製品の登場、WebサービスのBPM適用への動きと非常に多様化してきました。この状況についても今後着目してください。

5. BPMその設計方法を探る

BPMに関連したプロジェクトを始める場合、その設計手法や開発体制、保守/運用体制について意見を求められる場合

があります。BPMのプロジェクトは、複数部門ないしは複数企業から構成されるケースが多く、企業の一部門だけで何でも決定できる小さなプロジェクトとは違って、各部門(企業)間の調整作業と、仕事の範囲の取り決めが大事な作業になります。ここでは、これらの設計手法について論述します。

5.1. BPM設計手順

設計手順は次のようになります。

- (1) 製品導入とパイロット検証
- (2) 現状フロー分析
- (3) ビジネス・オブジェクト設計
- (4) データ・マッピング
- (5) パブリック・プロセスの設計
- (6) 配布と認証
- (7) プライベート・プロセスの設計
- (8) プロセス開始

この手順に従って説明を加えます。

5.1.1. 製品導入と試行

BPM製品を導入し、パイロット業務を選定して機能検証を行います。期間は1カ月程度が目安です。長すぎるとプロジェクトの進展に影響を及ぼしますし、短かすぎると機能テストが不十分に終わってしまう可能性があります。この段階で測定し、体得した開発実績値を基に、今後の開発/要員計画を見直します。

5.1.2. 現状フロー分析

パートナー間のワークフローが現在どのように流れているのかを、人の行動(アクション)とその間の情報(インフォメーション)について分けて図5のように記述します。

5.1.3. ビジネス・オブジェクト設計

パートナー間でやり取りする情報(メッセージ)を設計・作成します。一つのメッセージは、メッセージ属性と複数のビジネス・オブジェクトから構成され、一つのビジネス・オブジェクトはさらにオブジェクト属性と複数エレメントに分かれることとなります。エレメントは、標準テキスト、数値データ、図面データ、バイナリー・データなどのデータ・フィールドから成り立ちます。“ビジネス・オブジェクト”という表現は抽象的で分かりにくいのですが、発注情報・納期情報・受領確認などの一つひとつの情報がそれに相当します。パートナー間の情報では、これらのオブジェクトがまとめて送られる場合もあるので、1メッセージ複数オブジェクト構成で設計します。部品発注依頼に対して受領通知・納期回答・請求などの情報をまとめて1メッセージで返す形態

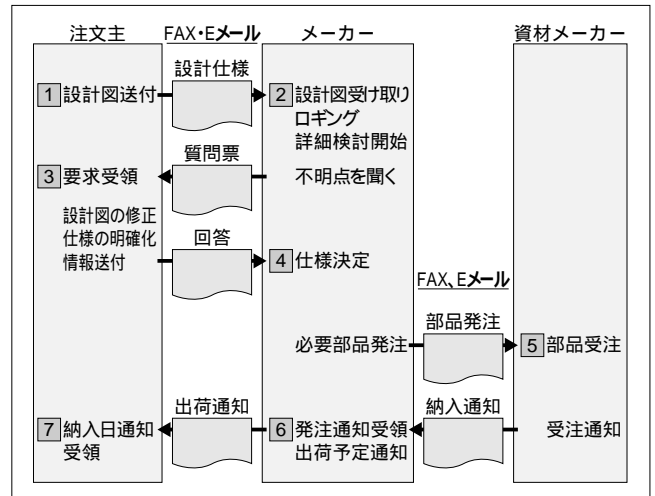


図5. 現状フロー分析事例

を想定しています。

メッセージ属性(オブジェクト属性も同じ)とは、同期/非同期の通信タイプと、同期タイプの場合のタイムアウト値(時間内に応答がない場合には取引不成立と判断)を指定するサービス・タイプがあります。オブジェクト属性では、そのオブジェクト内のキー項目を決定します(請求書であれば請求書番号・顧客番号・請求日など)。

ビジネス・オブジェクト定義は、業界標準が各業界で設定されています。RosettaNetやFIX、E2openなどです。業界標準は、あらゆるビジネス・モデルを考えているために、データ構造が一般に冗長で、全部のオブジェクト構成が審議中で未決定のケースもまだまだあります。しかし、BPM実現の対象範囲が海外に広がる可能性がある場合は国際標準を参考にしながら、ビジネス・オブジェクトを設計する必要があります。

5.1.4. データ・マッピング

ビジネス・オブジェクトのデータがどのシステムから得られるものか(MAP-IN)、どのシステムに渡すものか(MAP-OUT)のマッピングを定義します。パートナー間のビジネス・オブジェクトは、プログラムないし人間の判断を経由して変化していきます。ビジネス・オブジェクトのデータを、プログラム稼働に必要なデータにマッピングする作業と、逆にプログラムから出された情報をベースにパートナー宛のビジネス・オブジェクトにマッピングする作業です。この作業により、どこから入力されるのか分からないデータ・エレメントが発見されます。

このようなデータ・マッピング・ツールは、BPM製品の標準機能として提供されています。

5.1.5. パブリック・プロセスの設計

パートナー間のプロセスの流れと、その間で渡すデータにつ

いてフロー図を作成します。

最初は図5のような形の概要図で、正常処理系を記述します。ビジネス・プロセス間を流れる情報の内容とビジネス・プロセスの流れる順番を明確にするのは当たり前ですが、ビジネス・プロセス間の連携に時間の尺度を入れておくことが大事です。

パブリック・プロセスは、パートナー間のビジネス上の取り決め、すなわちビジネス憲章に相当するものですから、

- 注文後、翌日までにキャンセルしなかった場合は注文確定となる。
- 注文後、注文受領通知が当日内に戻らなければ、未注文と同じ扱いとする。

などのように、ビジネス・フロー間に一定期間以上の時間が経過した場合の設定を入れておかなければいけません。パブリック・プロセスは、相手となるビジネス・パートナーに対して、できるだけ共通なプロセス・フローとして設計すべきであり、個々のビジネス・パートナーに対して個別のビジネス・フローを設定することはまったく意味がありません。フローの管理・設計・開発のワークロードが多くなるだけです。通常は、一般パートナー、プレミアム・パートナー程度のパブリック・プロセスを設計することになります。

5.1.6. 配布と認証

作成されたパブリック・プロセスをパートナーに配布し、認証をお願いする作業です。パートナーと協議して合意したパブリック・プロセスは文書化され、IT媒体を経由して互いに承認作業を完了した後、実行できる環境になります。通常、このような作業は、担当者間の約束事で進んでいきますが、BPM製品としてこのような配布・認証の管理機能を持っている製品もあります。

あるパブリック・プロセスを、どのパートナーに対して、いつから実行可能にさせるか、そのパートナーからは実行承認はもらっているかという情報管理や、パブリック・プロセスのバージョン管理(どのバージョンをどのパートナーで使っているか)の機能を提供します。

5.1.7. プライベート・プロセスの設計

自社内のビジネス・プロセスを設計します。あるパブリック・プロセスの出力が、プライベート・プロセスの入力になり、その出力は別のパブリック・プロセスにつながります。

プライベート・プロセスは基本的にプログラムだと考えればいいのですが、以下のタイプに分類可能です。

- イベント通知機能
Eメール通知、FAX送信、携帯電話発信。
- 承認要請
OK/NGのいずれかの承認を仰ぎ、次のプロセスを決定。

• データ・マッピング

パブリック・プロセスからのオブジェクト・データをプログラム稼働に必要な形式にマッピングする機能。

• タイマー

一定期間、処理を中断する機能(相手からの応答を待つ場合などに利用できる)。

• 外部プログラム起動

既開発の外部プログラムをBPMプロセスの一環として起動。

• 別サブプロセス起動

別のビジネス・プロセスを起動。

• 停止

ビジネス・プロセスの終了。

この中でシステム設計上は、外部プログラム起動の機能が大切です。外部プログラムに相当するものが、自社の基幹システムやERP/CRM/SCMパッケージ製品になるため、このインターフェース部分(一般的にアダプターと呼ばれる機能)を提供するBPM製品があれば、開発ワークロードの短縮が可能です。以下のプロセス設計を進めていくことになります。

- IBMホスト接続であれば、IMS / CICS / 3270インターフェース
- MQ / MQSI / MQSeries Workflowインターフェース
- FTPインターフェース
- Eメール・インターフェース
- パッケージ製品インターフェース(Ariba、BroadVision、Siebel、i2、SAP、PeopleSoft、BAAN、Oracleなど)

5.1.8. プロセス開始

ここまでの七つのステップを経て、パートナー間のビジネス・プロセスを開始することが可能になります。ここでビジネス・プロセスを開始すること自体に問題はないのですが、その前に運用監視の仕組みを構築しておく必要があります。

状況の問い合わせ窓口として、ヘルプ・デスク、コール・センターで実施する体制を敷き、以下の運用監視システムを整備しておく必要があります。

- 運用BPMのシステムが正常に稼働しているかどうかなどの監視機能。
- BPMで取り扱っているトランザクション量の傾向はどのように変化しているかなどの統計機能。
- BPM資源・トランザクション記録のバックアップ・リストア機能。

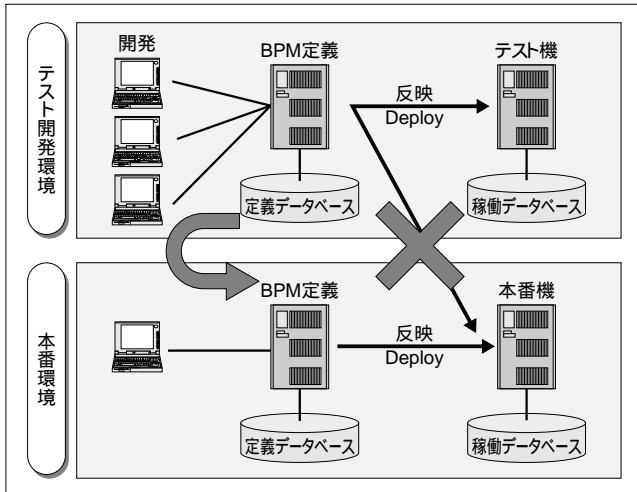
5.2. BPM設計ツール

BPMのシステム開発方法を説明してきましたが、開発ツールは、従来のDFD(Data Flow Diagram)、ER図(Entity/Relationship model)などで事足りるでしょうか。IBMでは、以

表9. BPM主要作業とその担当

作業	主担当
パブリック・プロセス定義	BPM担当(アプリケーション開発担当)
プライベート・プロセス定義	個別企業もしくは各部門
BPM運用標準	BPM担当(システム担当)
BPM開発標準	BPM担当(システム担当)

図6. BPM開発から本番移行方法例



前からADSG手法を利用して要件分析・基本設計・詳細設計...の各段階で出力するアウトプットを決める開発が行われています。このような手法で開発されたシステムは、規模が大きくなると設計書だけでもダンボール箱数個分の大きさとなり、経験の少ないシステム担当者には保守を任せられない状況になってしまいます。

BPMでは、別の設計ツールによるシステム設計が進んでいます。企業組織の役割、ビジネス・ルールの記述から始まり、企業データ構造からプロセス、サブプロセスを定義して実際のビジネス・モデルを構築します。

UML(Unified Modeling Language)ダイアグラムを定義することにより、クラス図、シーケンス図、ユース・ケース、コラボレーション図を作成していきます。

これらの手法(Methodology)は国内では、まだ定着しているとはいえませんが、BPMのシステム設計者が、上流から下流までをツールを駆使しながら連携して開発できるのは大きなメリットです。開発担当者の力量に依存しない均質な成果物となり、プログラム保守の軽減・影響度分析、プログラム関連の容易な把握が実現するからです。今後、ビジネス・プロセスをトップダウン手法で実現させる場合は、有効なツールとなるでしょう。その代表製品がHolosofx社やRational社などから提供されています。

5.3. BPMその開発体制

BPMの開発に当たって、特別な開発体制を考える必要はあ

りません。

BPMでは、企業パートナー間のビジネスの取り決めは、企業間の強弱関係で主導権を発揮できる企業、もしくは中立な立場の第三者機関が決定します。BPMの決定部門(機関)に付随して、BPMの開発保守体制を整備することになります。

BPMの主要作業と、その主な担当を表9にまとめます。

BPMは、業務フローを決めるというアプリケーション業務寄りの作業と、BPMツールを利用した開発・運用・保守をどのように行うのかというシステム寄りの作業という二つの側面があります。従って、組織的にはアプリケーション開発部門に所属するケースと、システム・インフラストラクチャー担当に所属するケースに分かれますが、BPMを最初に導入する場合には併用するといでしょう。BPMツールの使い方を、すべての開発担当者に身に付けてもらうには研修コストや製品コストが割高であり、現状ではBPM定義を行う専門チームをアプリケーション開発部門内に設置するのが妥当であることと、BPMツールによる開発はシステム部門で標準化を行い、運用方法も規定する作業が大事になるからです。BPMツールを利用した開発から本番機への移行方法例は、一般に図6のようになります。

BPMの開発は、BPM開発担当者ごとに開発用PCを提供し、そこから定義サーバーに向かってBPMの定義を行います。テスト機で機能テストを行うには、Deployと呼ばれる反映作業が必要です。ここで作動確認がOKになると、本番環境への移行作業が始まります。テスト開発機から本番機に直接Deployさせる環境も作成可能ですが、テスト機に登録しようとして本番機にうっかり登録してしまうというミスを防ぎたいため、大規模な開発環境では避けるべきです。本番環境では、あらかじめ本番登録担当者が作業するか、テスト機のBPM定義データベースから定義体を抽出して、本番環境にパラメーターを変更してDeployする作業を自動化する運用手順が必要になります。

6. 今後のBPMについて

ガートナー・グループでは、2003年には、e-ビジネスの90%はプロセス・オートメーションの技術を利用するようになるだろうという明るい予測を立てています。しかし昨今のIT不況を見ても、この予測がそのまま実現するという甘い考え方はできません。厳しい経済環境下では、将来におけるBPMの価値が実証できない限り、将来性を考えた投資という理由だけでBPMを採用することは決していないからです。

では、BPMの今後どのような状況になるでしょうか。BPMにとって、肯定的な流れも否定的な流れも考えられますが、筆者

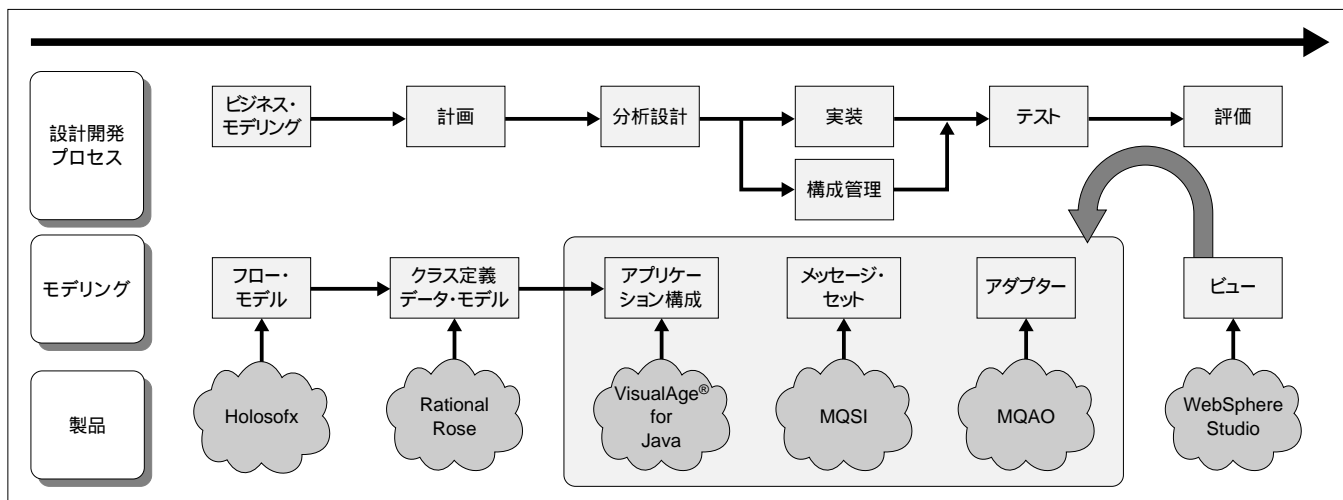


図7. BPMを想定したシステム開発の設計の流れ

は肯定的な流れを予想しています。これは、e-ビジネス・システムの構築手法そのものが、プロセス・マネジメントを前提としたメソドロジーに基づく開発手法に置き換わり、その中では、当然のようにBPMツールがシステム開発の上流から下流までを体系的に管理することが期待されるからです。

BPMは一企業内の部門間システムから始まり、B to Bの企業連携に展開すると考えられています。やがてB to Bや大きな部門間システムの適用などに限定された手法ではなくなると思います。一般的なシステム開発手法として、日常業務のシステム化を考えても十分適用できる手法になってくるのです。図7にBPMを想定したシステム開発の設計の流れと、モデリングのアウトプット、現在サポートしている製品をマップした構成を作成してみました。システム開発の上流から下流までを一元的に管理できるメソドロジーと管理ツールの重要度がますます高まると思われます。

7. おわりに

本論文を執筆しながら、国内と外国の企業風土の違いを強く意識するようになりました。それはメソドロジーの文化です。メソドロジーは、欧米の大学ではソフトウェア工学の基本講義として定着していますが、国内では、だれもが独自のスタイルでシステム構築方法を体得しており、システム工学的にIN/OUTを押さえた手法を理解している方は少ないのが実情と理解しています。筆者もメソドロジーの話には、今まではまったく興味はありませんでした。しかし、ますます国際化が進み、日本・米国・欧州・アジア各国での共同開発や開発結果資料の共有などの必要性が増加してきます。そのためには、BPMをきっかけとしたメソドロジー手法の理解と実践を進めなくてはいい

ません。メソドロジーをベースとしたプロジェクト開発は、それを理解している新人をすぐにプロジェクトに参加させ、即戦力として仕事をさせることができ、その成果物を共通利用できる形でまとめることが可能です。

IBMでも、SSM、TeAMethodなどの万国共通のセリング・メソドロジーを標準として採り入れ、新人の早期実践化と、ホワイトカラー技術職の生産性向上を期待しているように感じます。国内でも早急の対応が求められています。ソフトウェア/ハードウェアの技術そのものを追いかけることもITエンジニアとして大事な仕事ですが、メソドロジーについての勉強をこれからの課題としてぜひ取り組んでみてください。否、他人事ではなく、自身のテーマとしても考えていきたいと思っています。

(ページ数および表記上の観点から、著者の了解を得て編集部にて手を入れてあります)

[参考文献]

- [1] 一歩先行くB to Bシステムをつくる『日経インターネットテクノロジー』2000年9月号
- [2] 尾崎 久栄『WebSphere MQによる企業間連携(B to B) - BPM戦略』WebSphere2001資料
- [3] White Paper - Business Process Management, 2001年1月 Butler Group
- [4] WebSphere B to B Partner Agreement Manager Users Guide V1.1, 2000年10月