

業務の可視化を行う BPM 基盤の試作

加藤光幾[†] 野村佳秀[†] 金井 剛[†] 原 裕貴[†]

企業では事業拡大やシステムのリプレースなどに伴い、業務システムがますます複雑化していく傾向がある。このような業務システムでは業務の流れが複雑なため、業務上どこに問題点や弱点があるかが把握しにくく、IT 投資効果も明らかでない。

業務の流れを可視化し、絶えず問題点を把握し改善を繰り返すことにより、業務システムの全体最適化を支援するビジネスプロセスマネジメント(BPM: Business Process Management)基盤を開発した。本稿ではその方式について述べる。

A Prototype of Business Process Management Infrastructure

KATO Koki[†], NOMURA Yoshihide[†], KANAI Tsuyoshi[†], and HARA Hiroataka[†]

Business systems in an enterprise tend to get more and more complex with business expansions or system replacements. Business processes on such a system also become too complicated to point out problems and weak points on it.

We developed a prototype of Business Process Management Infrastructure that enables visualization and total optimization of business processes.

1. はじめに

1.1 複雑化する業務システム

企業ではその成長に伴い業務システムを段階的に導入し、またビジネスユニット単位で個別業務システムの導入やリプレースを行うなどにより、企業内に複数の業務システムが存在することが多い。

これらの業務システムは互いに複雑に関係し合うため、次のような課題が発生している。

- 業務処理全体の進捗を把握しにくい

顧客からの受注やサービスが、現在どのような状況にあるか把握するのが困難である。特に複数の部署にまたがる場合、どこまで処理が進んでおり、あとどのくらいかかるか見当がつかない。

- 業務プロセスの問題点がわからない
業務プロセス(処理手順、業務の流れ)のどこかに問題があることは分かるが、具体的な問題箇所が特定できない。または担当者は自分が担当している業務の範囲では問題点が分かっているが、それらを業務プロセス全体との関係として把握できない。
- IT 投資に対する効果がわからない
業務システムを変更する場合に、期待する

[†](株)富士通研究所 ソフトウェアイノベーション研究部
Software Innovation Lab, Fujitsu Laboratories Ltd.

効果が得られるか評価できず、結論を出せないため業務システムの変更が遅れ、結果的に後手に回ることになる。

このような状況は業務プロセスを全体として可視化し把握・評価する仕組みがないため生じると考えられる。

1.2 BPM

業務プロセスの現状やあるべき姿をモデル化し、業務全体を可視化、改善していく技術として **BPM (Business Process Management)**^{1,2} が注目されている。

BPM には以下のような狙いがある。

- 企業全体の業務プロセスを視覚的に明確化する

業務全体の流れをグラフィカルに描き、それを基に後述の業務システム間の連携や業務の進行状態モニタリングを行う。

複数の部門が連携した業務では他部門での進行状況を把握するのが困難である。また個々の部門では問題が分かっているが全体との関係が不明である。そのような状況では限られた範囲での最適化しかできないが、企業全体から見ると効果がないか逆効果である場合もあり得る。そのため業務全体を俯瞰したうえで問題点を捉える必要がある。

- 業務プロセスの追加や変更を容易にする

EAI (Enterprise Application Integration) やワークフローシステムと連携することにより、業務プロセスの追加や変更を、すばやく効率的に実装できるようにする。

EAI は企業内で使われる異種システム間を柔軟に結合・連携するためのハブとして機能するミドルウェアであり、ワークフローと同様なフロー定義を持つ。フロー定義を変更することで業務を比較的容易に変更することができる。

- 業務プロセスの実行状況をモニタリングし問題点を特定する

定義した業務プロセスに沿って流れる業務の進行状態をモニタリングすることで、業務の流れ全体を可視化する。

また各アクティビティ(業務プロセス内の処理単位)の処理時間情報を蓄積し、アクティビティごとの処理時間を求めて統計的に処理することで、業務上の問題点を特定しやすくなる。

以下では BPM が持つ特徴のうち、特に業務プロセスの可視化に注目する。

2. BPM 基盤

2.1 従来の BPM の問題点

現状の BPM 製品の多くは EAI やワークフロー製品の一機能として組み込まれており、導入のためにはまず EAI で業務システムを結合することが前提となる。EAI を導入していない企業では EAI を導入すること自体が大きな負担となる。そのため、まず自社の業務の流れを可視化したい企業に対し、EAI を用いなくても簡易に可視化できる方式が求められている。

また、すでに EAI が導入されている企業でも、EAI で把握できるのは EAI が呼び出す単位であり、EAI が呼び出した業務システム内部での動作を把握できない。特に EAI の結合先が業務パッケージである場合、業務パッケージ自体が何らかのフローを実行しており、その中での業務の流れは把握できない。

一方利用者の観点から見ると、利用者が意識する業務プロセス定義と、それを EAI などで実装する際に業務システムで用いるプロセス定義とは異なる場合が多い。そのため利用者に対し EAI のプロセス定義に基づき BPM のモニタリング情報を示しても、知りたい情報が足りない、ほかの情報と混在している、必要以上に細かい情報が提

示されるなど、分かりにくい可能性がある。

利用者が見たいのは利用者が定義した業務プロセス上での流れであり、それに即した情報を提示する仕組みが必要である。

2.2 プロトタイプの特徴

EAI やワークフローなど特定のミドルウェアに依存せず、既存業務システムに対して少ない変更で業務の可視化を行うことができる、BPM 基盤のプロトタイプを開発した。

対象業務プロセスを定義(Plan)→業務システムから実行状況をリアルタイムに収集(Do)→現状を把握(Check)→問題点を発見・解決(Action)→また改善された業務プロセスに対してモニタリングを行うという PDCA サイクルをまわしつづけることで、継続した業務システムの全体最適化を支援する。

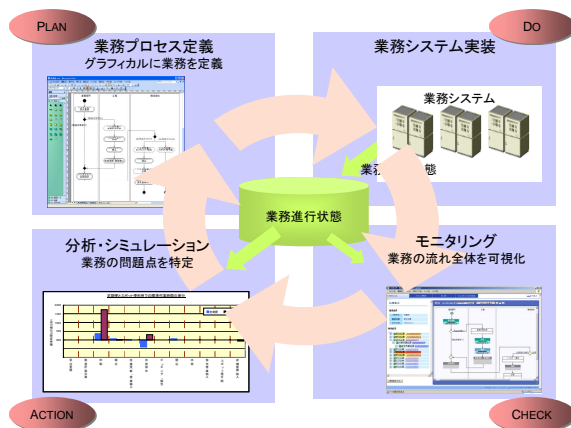


図 1 PDCA サイクル

業務システムから実行状況を収集するフェーズで業務システムへの変更をなるべく少なくすることを目指している。これにより業務プロセス可視化の導入が容易になると期待する。

また、BPM 導入の初期で業務プロセス定義に長期間費やすと、最初の可視化が得られるまでの期間が長くなり利用者の改善意欲が減少することにつながりかねない。そのため最初から完全な業務モデルを作り業務の全てをモニタリングするのではなく、最初は業務モデルを簡単に粗い

状態で定義しモニタリングして問題点を絞り込み、次の PDCA サイクルでより細かく見たいところにあわせて業務プロセスも細かく記述し、より詳細なモニタリングを行うことで、利用者に対して早く結果を出し改善意欲を維持する。

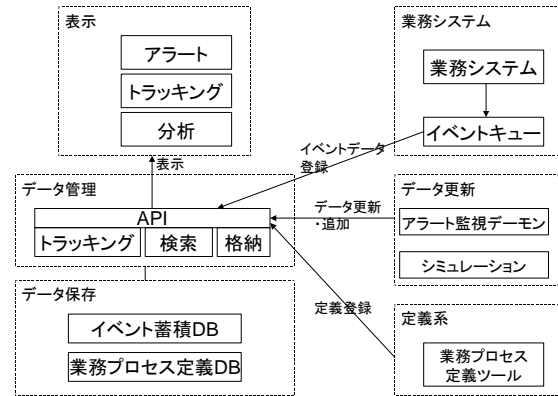


図 2 全体構成

以下に BPM 基盤技術の特徴を挙げる。

2.2.1 業務プロセスを UML で記述

BPM 基盤の利用者はシステム管理部門よりも、担当者やリーダ・管理職が使うことを念頭においている。そのため業務プロセスの可視化ではリーダなどが分かる業務プロセスに対応して表示することとし、定義も業務プロセスを定義する。業務システムのレベルでの流れ記述では実装に依存するため必ずしも業務プロセスと同じとは限らない。

業務システムの実装ではなく業務本来の流れを記述することで、業務的な観点でプロセス全体の監視・分析を行う。

業務プロセスの表記には国際標準である UML のアクティビティ図を用い、意味のあいまい性をなくす。

業務の進行状況は後述のイベントデータを収集することで行う。

業務プロセス定義にあわせ、業務プロセス内のアクティビティ(業務プロセスでの最小業務処理単位)とイベントデータの対応を定義する。

業務プロセス内の問題箇所を絞り込むために、アクティビティの追加や、サブアクティビティを用いて業務プロセスを詳細化していくことが可能である。

アクティビティや遷移に標準時間を設定することで、後述のアラートや分析に用いることができる。

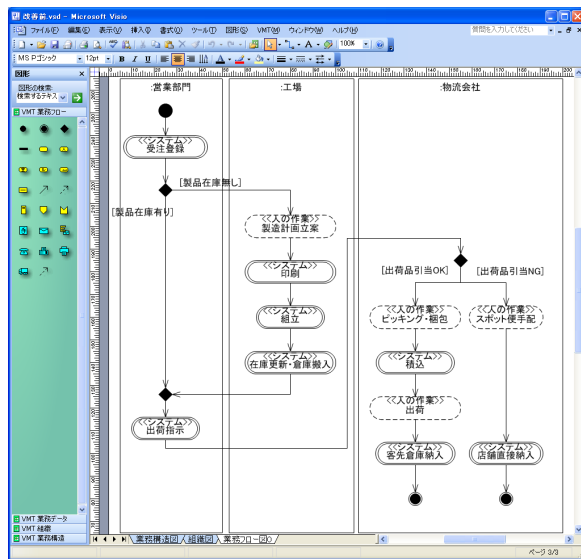


図 3 業務プロセスの記述例

2.2.2 イベントデータ抽出

業務プロセスの進行を把握するため、アクティビティの開始・終了を示す「イベントデータ」を各業務システムから抽出する。

イベントデータは業務プロセス名、アクティビティ名、開始ないし終了の時刻、アクティビティで扱う業務データの ID、関連する業務データの ID (任意個)、プロパティ(担当者名、製品番号、コスト情報など)を含むデータである。XML で表現しており、分析項目を増やすためプロパティを追加するなどが容易に行える。

イベントデータは業務システムから BPM 基盤サーバのデータベースに転送され蓄積される。

イベントデータの例を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<bpm:event xmlns:bpm="urn:fj-bpm-event">
  <bpm:modelName>受注出荷業務</bpm:modelName>
  <bpm:primaryKey type="受注">JU00011</bpm:primaryKey>
  <bpm:relativeKey type="見積り">QU00002</bpm:relativeKey>
  <bpm:property name="担当者">加藤</bpm:property>
  <bpm:property name="納品希望日">2004/08/23</bpm:property>
  <bpm:eventData>
    <bpm:eventName>受注登録</bpm:eventName>
    <bpm:finishTime>2004-08-19T21:21:22</bpm:finishTime>
  </bpm:eventData>
</bpm:event>
```

図 4 イベントデータの例

イベントデータを抽出するため、アクティビティの開始や終了時に実行される動作(伝票入力画面終了時や、RDB へのデータ追加など)に対応してイベントデータをリアルタイムに出力するコードを業務システムに追加する。

イベントデータの出力方法は業務システムの構成に依存する。以下に例を挙げる。

- 業務パッケージソフトにユーザ出口が設けられている場合

業務パッケージの機能を使用したときに通過する(例えば伝票入力画面終了時)ユーザ出口にイベントデータ出力を設定する。

- RDB を使っている業務システムの場合

RDB の TRIGGER 機能を用い、特定のテーブルにレコードの挿入やアップデートが発生したときイベントデータを出力する。

- 業務アプリケーションを作成・改造するタイミングの場合

業務アプリケーションにイベントデータを出力する機能を追加する。

- EAI を使用している場合

EAI がアプリケーションを起動ないし終了した時点でイベントデータを出力する (EAI に BPM 基盤対応が必要)。

特定の業務システムに対しては業務プロセス定義からイベントデータ抽出用コードの雛形を生成する機能を設けている。SE はこの雛形を必

要に応じて修正し業務システムに埋め込む。

イベントデータは各業務システム内に設ける「イベントキュー」に一旦蓄積される。イベントキューに蓄積されたイベントデータは JDBC など BPM 基盤サーバに転送・蓄積される。

なお実際には上記のイベントデータを業務システムから出力する時点では XML 形式ではなくイベントデータの中身に相当するデータの並びとして出力することができる。業務システムへの負担をなるべく低減するためである。その場合はイベントキューから転送するときに XML に変換する。

2.2.3 イベントデータの関係付け

各業務システムから出力されたイベントデータは ID を用いて関係付け、データベースに蓄積する。

通常の業務では受注処理のように 1 つの伝票に複数の明細が含まれ、それぞれが個別の処理に分かれるとか、配送処理のように複数の品物をまとめて発送するなどの処理が発生するが、このような 1:n や m:1 の関係も上記の ID 関係を用いることで扱える。例えば図 5 (a)のように受注伝票の明細(A と B)ごとに在庫伝票が発生する場合、在庫伝票には在庫伝票の ID のほかに何が元になり発生したかを示す関連 ID が埋め込まれている。これを用い受注伝票と在庫伝票の 1:n 関係を扱う。

このように業務で用いる ID をイベントデータとして取得し、それらに関係付けることにより、業務プロセスのインスタンスとして把握することができる。

なお前述のプロパティなどを増やしてもデータベースのスキーマを変更する必要はない。

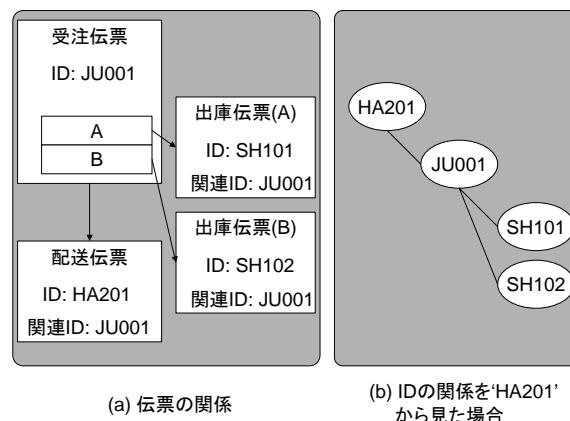


図 5 ID の関係

2.2.4 トラッキング

業務の全体を可視化する基本的な機能として、業務プロセスのインスタンスごとに進行状況を把握する機能が必要である。

関係付けられたイベントデータを業務プロセス定義の図に重ね合わせて表示する、業務のトラッキング機能を設けた。

関係はツリーとしても表現しており、逆引き (ID の参照関係を逆にたどる) トラッキングも行える (図 6 の左側部分)。逆引きを用いて、例えば図 5 (b)のようにある配送で問題が発生したとき、配送処理に関係する受注処理や在庫処理を検索し対処することができる。

ID が 1:n や m:1 の関係になっており、アクティビティが実際には複数同時に実行される場合は、その個数も図中に表示する。

実行中のアクティビティやすでに終了したアクティビティは業務プロセスで定義した図と同様の図に色別に表示することで、残りの作業(アクティビティ)がひと目で把握できる(図 6 の右側部分)。そのため例えば顧客からの問合せがあったとき、どの部署に問合せすればいいかが分かりやすくなり、また問題があるときにまだ実行していないアクティビティの部分で回復可能ななどの検討をしやすくなるのが期待できる。

業務プロセス図の表示には SVG を用いており、

表示部分の拡大・縮小が容易に行える。また業務プロセス定義でサブアクティビティを用いて定義している場合は、トラッキング表示時でサブアクティビティをクリックするとサブアクティビティの表示が変わる。

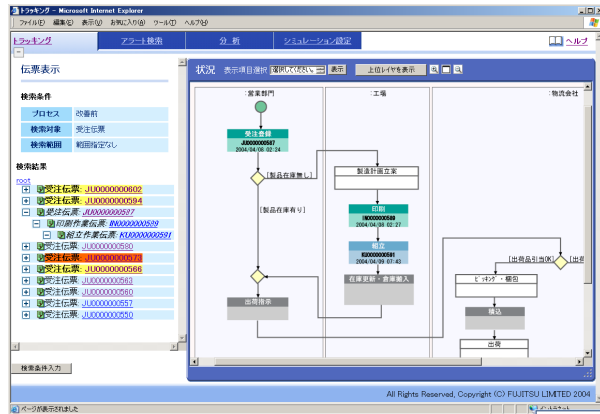


図 6 トラッキングの表示例

2.2.5 アラート

トラッキング機能で全ての業務プロセスの進行状況を把握するのは困難である。そのため、あらかじめ業務上クリティカルな状態にあるかをアラート条件として設定し、定期的のアラート条件を満たしている業務が存在するかをチェックするデーモンを用意している。

基本的なアラート条件は標準で用意するとともに、より複雑な条件はユーザがプログラムにより定義することが可能である。

たとえば受注した品物の納品が顧客に約束した納期日より遅れることがないように、業務プロセスの進行状況を元に問題の発生を事前に発見し、対応を行うことができる。

業務プロセスを定義するときあらかじめアクティビティや遷移の標準時間を設定することで、あるアクティビティからほかのアクティビティに至るのに要する標準処理時間が求められる。

トラッキング機能により現在どのアクティビティに存在するかが分かるので、アラートの定義で着目するアクティビティまでに要する残りの

時間が求められる。これらより今後標準時間で業務が進んだときにアラート条件をクリアできるか否かを計算を行う。

またアラートが発生したとき担当者にメールを送信し、対処を促すという応用も可能である。

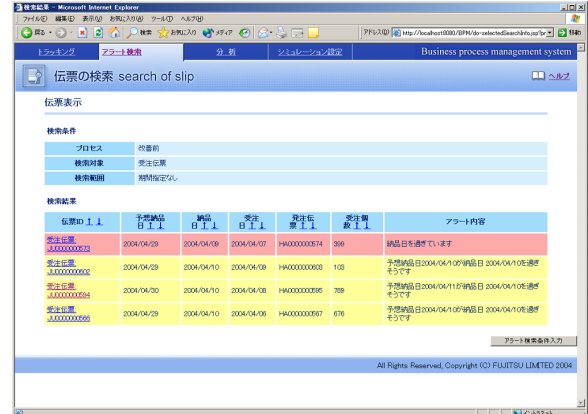


図 7 アラート一覧の例

2.2.6 分析

BPM 基盤では業務プロセスのインスタンスごとにリードタイム、アクティビティの処理時間、資源待ちなどで発生するアクティビティ開始待ち時間、プロパティに設定したコストなど各種情報を求められる。これらと業務プロセス定義で設定した標準時間などを元に、期間内の平均値、標準値との乖離など統計量を計算し業務プロセス図に重ねて表示する機能を設けた。これにより業務プロセス上の問題箇所を特定しやすくなる。

独自の分析をしたいときは計算ロジックを Java で記述しプラグインとして組み込む。また蓄積したデータを CSV 形式に変換し出力する機能も設けてあるため、表計算ソフトで読み込み処理することも可能である。

2.2.7 シミュレーション

上記の各種機能が提供する情報に基づき、問題点を改善する案を作成する。

その際、部門単位ではなく企業全体から見て業務改善が最適で妥当かを、稼動している業務を変更する前に評価することが重要である。さらには

IT 投資効果に見合っているかも判断しやすくなり改善を積極的に行う材料となる。

業務プロセス定義の振る舞いを評価するためのシミュレーション機能を開発、改良中である。

業務プロセス分析・モデリングツールにも業務プロセスで設定した標準実行時間などのパラメータを用いて乱数で業務の流れをシミュレートする機能を持つものがあるが、設定する値はコンサルタントなどが設定した値であり、設定した値が現実と食い違っていれば結果も不確かなものになる。

一方、BPM 基盤は現実の業務で発生した業務プロセスのインスタンス単位の情報を持っているので、実際の値を用いて業務プロセスを変更した場合をシミュレートすることが可能である。

複雑なロジックを要する場合に対応するため、シミュレーション用のプラグインを Java で記述し追加可能とした。

3. 考察と課題

現在実際の業務へ適用し検証を行おうとしている段階であり、評価はまだであるが、いくつかの点に関し、考察と課題をあげる。

3.1 EAI ベースの BPM との違い

2.1 で述べたように従来の BPM は EAI の一機能として実装されている場合が多い。業務プロセスの可視化の観点から BPM 基盤を EAI ベースの BPM と比較する。

● 業務プロセス記述

BPM 基盤で用いる業務プロセス記述は業務の流れ自体を記述しその実装には依存しない。一方 EAI ベースの BPM では EAI で構築する業務の実装レベルの処理フローを記述する。製品によっては業務プロセス(業務の流れ)を記述し、それに関係付けて実装レベルの処理フローを定義するものもあるが、業務のモニタリングやシミュレーションを行う際には実装レベルの処理フローに

沿っており、業務プロセスとして結果を見るのが困難である。

● 導入の簡易性

すでに全社が EAI で結合されている企業では EAI ベースの BPM で即座に業務の可視化が行える。一方 EAI をまだ導入していないが部門にまたがる可視化を行いたい企業では、可視化に先立ち EAI を導入するという方法もあるが業務システム全般を変更するなど敷居が高い。一方 BPM 基盤のようにイベントデータだけを収集する仕組みは負担が少ない。

● プロセスの改善

EAI ベースの BPM では EAI の処理フローの変更などで問題点を比較的容易に変更することが可能である。一方 BPM 基盤では改善方法の具体的なものは持っておらず、実装依存である。ただし改善方法の検討の結果 EAI 導入となった場合でも、業務プロセスなど BPM 基盤で検討した結果は EAI にも役立つ。

表 1 EAI ベースとの比較

	EAI ベース	BPM 基盤
業務プロセス記述	実装寄り	業務寄り
導入の容易性	敷居が高い	比較的容易
プロセス改善	容易	実装依存

3.2 課題

● 導入のしやすさ

BPM 基盤ではイベントデータ抽出用コードの雛形を生成する機能を設けているが、適用できる業務システムが限られているため、より多く適用できるように拡充する必要がある。また分析やシミュレーションで複雑な処理を行うためにはプラグイン開発を要するが、Java の知識が必要となり現場の担当者が開発するのは困難であるため、なるべくプラグイン開発なしでカバーする範囲を広げようとしている。

● BI ツールとの連携

企業の活動を経営の視点から可視化し管理する手段としては従来から CPM (Corporate Performance Management)があり、主に Data Warehouse を用いた BI (Business Intelligence) ツールが使われている。

CPM は 1 日とか四半期ごとの集計により経営指標を算出するという性格上、データの集計がリアルタイムでない場合が多く、BI ツールの結果を見てから業務に対して行動を起こすのでは時機を逸する場合がある。一方 BPM を、企業の活動を業務の視点から可視化し管理する手段と捉えれば、CPM とは相補的な役割を持つため、両者の連携が重要になると考える。

4. まとめ

業務プロセスの可視化に重点を置いた BPM 基盤を試作した。可視化導入を容易にしたことで、業務システムの全体最適化に向けた PDCA サイクルを短期で実現することが期待できる。

今後は本基盤の機能充実を行い、製品化およびソリューションへの適用を推進していく予定である。

参考文献

1. Smith, H. and Fingar, P., Business Process Management: The Third Wave, Meghan-Kiffer Press, 2002.
2. 吉田洋一, 竹村司, “企業アプリケーション統合とビジネス統合のための開発ツールの動向”, 情報処理学会誌 Vol. 45, No. 1, 2004.