

## ビジネスプロセスの表現に BPMN を用いた MDA 開発の研究

小野 勇介<sup>†</sup> 片岡 信弘<sup>‡</sup>

<sup>† ‡</sup> 東海大学工学研究科情報理工学専攻

〒259-1292 神奈川県平塚市北金目 1117

E-mail: <sup>†</sup> 6adrm008@keyaki.cc.u-tokai.ac.jp <sup>‡</sup> kataoka9@keyaki.cc.u-tokai.ac.jp

あらまし：現在 MDA において CIM から PIM への変換ができていない。本研究は BPMN で CIM を作成し、BPMN からクラス図を作成するためのメタデータを作成する。そしてメタデータから PIM に相当する概念的なクラス図へ変換することを可能にしている。メタデータを用いることで、クラス図の生成に特定の UML ツールに依存することがなくなる。そして概念的なクラス図に手を加えることで設計レベルのクラス図を作成し、実装へと繋いでいく手順を提案するものである。BPMN を用いることで、現在のビジネスプロセスの全体を把握することができ、そこからプロセスの最適化を実現することが可能となる。

キーワード ; システム開発, ビジネスプロセス, BPMN, 業務改善, MDA

## A proposal of MDA development using BPMN for expression of Business Process Model

Yusuke ONO<sup>†</sup> and Nobuhiro KATAOKA<sup>‡</sup>

<sup>† ‡</sup> Tokai University

1117 Kitakaname, Hiratsuka-si, Kanagawa, 259-1292 Japan

E-mail: <sup>†</sup> 6adrm008@keyaki.cc.u-tokai.ac.jp <sup>‡</sup> kataoka9@keyaki.cc.u-tokai.ac.jp

**Abstract** :At this time MDA can't transform CIM into PIM. This research makes it possible to make CIM with BPMN. That is transformed into meta-data for making class diagram. And meta-data is transformed into conceptual class diagram corresponding to PIM. As meta-data is used, it doesn't depend on specific UML tool to generate class diagram. This is a proposal of development procedure for information system that it is added detailed elements to make specific class diagram. Using BPMN, it makes present Business Process cleared and understood. That helps to optimize Business Process.

**Keyword** ; System Development, Business Process, BPMN, Business Improvement , MDA

### 1. はじめに

情報システムの複雑化、顧客要求の曖昧さ、仕様変更の多さ、ビジネスプロセス自体の複雑化、など様々な理由により情報システムの構築が難しくなっている。また、ビジネスプロセスマネージメントの重要性が増し、業務の改善と共にシステムの変更が頻発するようになった。このような理由により困難になっている情報システムの構築を成功させるために顧客の基本的な要求を取り込みきる必要がある。かつビジネスプロセスの変更に対応するべく、変更し強い情報システムを構築する必要もあるといえる。変更し強いシステム構築法に MDA が挙げられる。モデルを中心にシステム構築を進めることで、実装技術や環境の変化に強くなることができる。

また、モデルからソースコードへの変換も可能であり、新しい環境への移行の負担を軽減することもできる。しかし、現在 MDA においてはビジネスプロセスからのシステム開発は確立していない。

そのような問題点を解決すべくビジネスプロセスからの MDA 開発の研究を行った。BPMN を用いて現状のビジネスプロセスの全体を把握することで、ビジネスプロセスの最適化をすることも可能となる。今回 CIM である BPMN から

PIM に相当する概念的なクラス図を生成し、それに実装に向けて詳細な情報を付け加えることで設計レベルのクラス図とする手法を提案する。これによりビジネスプロセスマネージメントによる業務の最適化が図れるとともに、それに伴

う変化の激しいビジネスプロセスやシステム構築の対応が容易になる。以下2章で研究の背景を述べ、3章で提案を述べ、4章で提案内容の試作について述べ、5章で検証と評価について述べ、6章でまとめを述べる。

## 2. 研究背景

### 2.1 ビジネスプロセスの重要性

ビジネスプロセスとは「ある目的のために、企業内あるいは企業間で行われる作業」と定義される。業務そのものであり、システム構築を行う際はこのビジネスプロセスの分析が必要になる。構築するシステムの要求分析レベルでヒアリングの失敗、また要求管理を失敗したことによるプロジェクトの崩壊は非常に多い。また、プロセスという性質上、目にすることができないため全員が同じ認識をしているとは限らない。また、システム構築の視点だけでなく、経営の視点からもビジネスプロセスは重要である。現在、ビジネスは多様化、複雑化してきている。そのため変化も激しく、対応に時間をかけていてはビジネスチャンスを逃してしまうことにも繋がる。現状のビジネスプロセスをしっかりと把握し、変化に対応し、ビジネスチャンスを逃さないようにすることが重要である。

そのため、近年はビジネスプロセスマネジメントの重要性が高まってきている。これを可視化するための技術にビジネスプロセスモデリングがある。

### 2.2 ビジネスプロセスモデリング

ビジネスプロセスを明確化することで、業務に必要な流れが明確となり、適切な流れであるかどうかの判断が可能となる。そして流れの変更の必要性や、他社とのより良い連携をとる手段、顧客からの要求を明確にできるようになる。ビジネスプロセスを明確化する手段にBPM(ビジネスプロセスモデリング)がある。

ビジネスプロセスを可視化することで、自分の回りだけの狭い範囲だけでなく、現状のビジネスプロセスの全体像をしっかりと把握することができるようになる。現状を把握することでビジネスプロセスのマネジメントやリエンジニアリング、つまり最適化及び再構築を可能にすることができる。そして把握するだけでなく、可視化することにより関係部門間とのコミュニケーションツールとしてビジネスプロセスモデルを利用することができるようになる。

現在、ビジネスプロセスを表現するツールとしてARIS、UML、BPMN等がある。その中で、今回ビジネスプロセスを表現するツールとしてBPMNを使用した。BPMNとはBusiness Process Modeling Notationの略であり、ビジネスプロセスを表記するOMG標準化仕様の表記法である。同じOMG標準であるUMLのアクティビティ図と似ているが、BPMNはその記述をXMLに出力するフォーマットが定義されているツールが存在するため、今回の提案ではBPMNを用いている。

## 2.3 MDA

モデルを中心としてシステム開発を進めていく手法としてOMG(Object Management Group)が提唱するMDA(Model Driven Architecture)がある。開発手順はFig.1のようになっている。システム自体から独立したビジネスプロセスレベルのモデルであるCIM(Computation Independent Model)の作成。作成したCIMを基にプラットフォームに依存しないモデルであるPIM(Platform Independent Model)の作成そして各種プラットフォームに特化したPSM(Platform Specific Model)の作成。PIMやPSMはそれぞれ自動変換され、PSMからソースコードを自動生成し、システムの実装へと繋げていく。

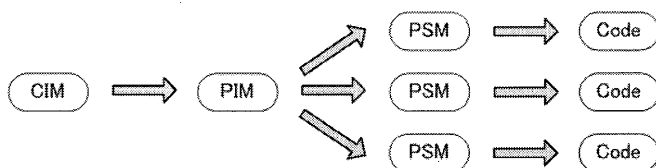


Fig.1 MDAのアーキテクチャ

現在PIMからPSM、そして実装コードまでの繋がり様々研究がされており、完全ではないが実際に生成できるツールも存在している。しかしCIMからPIMへの自動変換は存在していない。たとえばARISでは、ARISのビジネスプロセスモデルにクラス図の表記が可能であり、そのビジネスプロセスモデルからクラス図の名前、操作、属性等は生成することができるが、あくまでこれは手作業である。また、それぞれの関連まで生成することはできない。その為、クラス図を作るには生成されたものを手作業で関連付けを行う必要がある。

## 3. 提案内容









### 3.1 提案の骨子

研究背景で取り上げたように、現在のシステム開発ではビジネスプロセスが重要視されている。しかしMDAに基づいて開発を行おうとした際に、CIMからPIMへの移行ができていないという問題点がある。この問題を解決するために、CIMからPIMに変換し、概念的なクラス図の生成を行い、それに手を加えることによって設計レベルのクラス図の作成を行い、実装へと繋げていく手順の提案を行う。

### 3.2 変換方式

MDAの考え方に基づき、CIMにBPMNを用いてビジネスプロセスを表現し、PIMにUMLのクラス図を用いた概念的なクラス図の自動生成を行う。また、CIMとPIMの間にメタデータを使用することで、PIMであるクラス図を生成する特定のUMLツールに依存することをなくする。CIMからPIMへ変換を行う為にTable.1のようにBPMNの要素クラス図の要素対応付けをした。このように関連付けた理由を次に記す。

Table.1 BPMN とクラス図の対応付け

BPMN 要素	クラス図要素
 データオブジェクト	クラス及びクラス名
 アクティビティ	クラス図の操作名
 シーケンスフロー	クラス間の関連付け
 ゲートウェイ	クラス間の関連付け
 スタートイベント	特になし
 エンドイベント	特になし
 メッセージフロー	クラス間の関連付け
 関連	データオブジェクトと アクティビティの対応付け

#### ・データオブジェクト

BPMN でのデータオブジェクトの役割は「成果物」である。ビジネスを行うにあたり、必ず情報を扱う必要がでてくる。情報は名詞または名詞句である。構築するシステムに対して、クラスをどのように抽出するかを考える方法に「名詞抽出法」というものがある。文章から名詞または名詞句を取りあげてクラスにするというものである。今回はその方式を取り、ビジネスプロセスに出てくるデータオブジェクトをクラスに対応付けた。

#### ・アクティビティ

アクティビティは業務など作業の総称を示している。データオブジェクトに対して何か作業を行った場合、データオブジェクトに何かしらの変更を行っている。これをシステム化するとクラスに対しての操作となりえる。そのため、アクティビティはクラスの操作として対応付けている。

#### ・シーケンスフロー、メッセージフロー、ゲートウェイ

どれもアクティビティやプール等の要素を接続するものである。ビジネスプロセスモデルとしてはそれぞれ意味が違っているが、クラス図となればクラス間を接続する関連だと言える。ゲートウェイはビジネスプロセスモデルでは分岐や併合を行うが、クラス図においてはモデルの時点では全て関連付いてしまうため、インスタンスの状態であれば分ける必要がないと考えた。

#### ・関連

この要素でデータオブジェクトとアクティビティを繋いだ場合、アクティビティの作業を関連付けたデータオブジェクトに行うと考えられる。その為、関連付けたデータオブジェクトとアクティビティがそれぞれクラス図でのクラスとそのクラスの操作に対応付けている。

#### ・スタートイベント、エンドイベント

BPMN でのこの要素はクラス図には対応する要素がない

ため、対応付けていない。

## 4. 提案内容の試作

本研究の提案に沿った CIM から PIM への変換の手順は以下の Table.2 の通りである。Table.2 の 2 の ExtractProcess と 4 の BPMN2UML は今回試作した変換システムである。

Table.2 提案手順

		表現方法	使用したソフトウェア	入出力形式
1	CIM	BPMN	Process Modeler for Visio	出力: SimpleXML
2	↓	↓	ExtractProcess (今回作成)	入力: SimpleXML 出力: XML
3	↓	メタデータ		
4	↓	↓	BPMN2UML (今回作成)	入力: XML 出力: XMI
5	PIM	クラス図	各種 UML ツール	入力: XMI

### 4.1 ExtractProcess

ExtractProcess を使うために、BPMN で表記したビジネスプロセスモデルをシンプルな XML で作成しておく必要がある。本研究では、BPMN の表記に ITP Process Modeler for Microsoft Visio を使用しており simpleXML でエクスポートをすることができる。Fig.2 に示すように、エクスポートした simpleXML を ExtractProcess でメタデータに変換する。

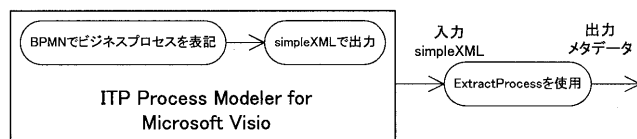


Fig.2 ExtractProcess 使用までの流れ

### 4.2 メタデータ

ITP Process Modeler for Microsoft Visio から出力した Simple XML を ExtractProcess で抽出した状態をメタデータと定義する。メタデータをおくことにより、Fig.3 のようにメタデータに追記するデータによりクラス図を再生する UML ツールを選ばなくなるという特徴ができる。

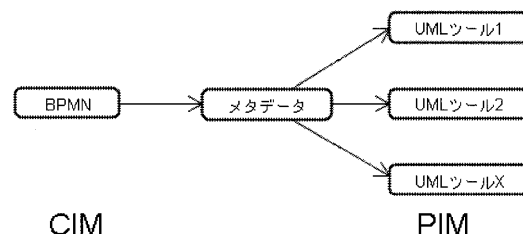


Fig.3 メタデータの位置づけ

今回のメタデータは Fig.4 の UML Metamodel を参考に作成した。メタデータは Fig.5 のような構成をしている。それぞれクラス図でのクラスのメタデータと操作のメタデータ、関連のメタデータである。メタデータ毎にクラス生成に必要なクラス ID とクラス名、操作生成に必要な操作 ID と操作名そしてその操作が所属しているクラス ID、関連 ID とクラス

間の接続に必要な接続元クラス ID と接続先クラス ID を記録している。また、実際のメタデータを Fig.6 に記す。

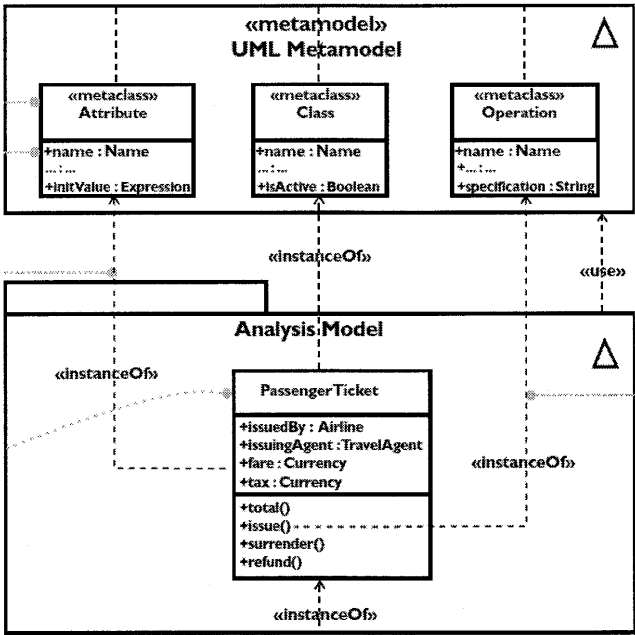


Fig.4 UML Metamodel

～出展 Cris Kobryn : 「A Standardization Odyssey」～

クラス	操作	関連
クラスID	操作ID	関連ID
クラス名	操作名	接続元クラスID
	クラスID	接続先クラスID

Fig.5 メタデータの構成

```
<element type="DataObject">
  <attribute name="Id" type="String"
    value="e2f51299-6d94-4917-a2b8-8ccf060bb50c" />
  <attribute name="Name" type="String" value="顧客情報" />
</element>
<element type="Association">
  <attribute name="Id" type="String"
    value="6e6c4d70-840c-4f45-95f1-26dd5a548ae1" />
  <attribute name="Source" type="Task"
    value="362d3d8e-b4f2-4433-88da-0bd2269267d3" />
  <attribute name="Target" type="DataObject"
    value="e2f51299-6d94-4917-a2b8-8ccf060bb50c" />
</element>
```

Fig6 メタデータの XML 形式事例

Table.1 で対応付けた通りに変換するため、Fig.6 での element type が DataObject のものはクラスへ、Association のものは関連へと変換する。Fig.5 のメタデータの構成と比較すると、element type が DataObject のタグ以下では attribute name に Id と Name が存在しており、それぞれ Fig.5 のクラス ID とクラス名に対応している。同じく element type が Association のタグ以下では、attribute name に Id と Source、Target が存在しており、それぞれ関

連 ID、接続元 ID、接続先 ID となっている。

4.3 BPMN2UML

ここではメタデータを XMI 表記のクラス図に変換し、更に特定の UML ツールに対応した XMI 表記を付け加えている。メタデータからクラス図を生成するにあたっては Fig.4 の UML Metamodel を参考にした。定義したメタデータをクラス図に変換するときの対応は次の Fig.7 のように関係付けている。

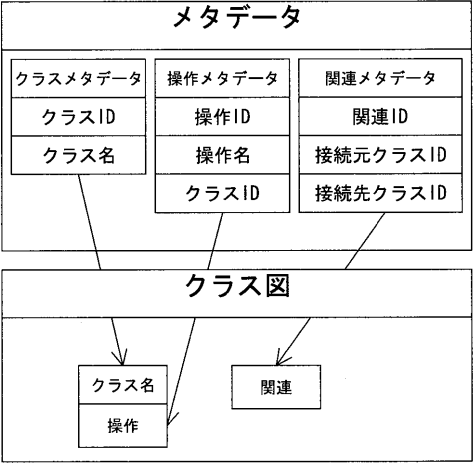


Fig.7 メタデータとクラス図の関係

XMI(XML Metadata Interchange)とは、モデルをツールやミドルウェア間でXMLを使って交換するためのOMGの標準規格である。しかしXMIでの交換も完全に互換性があるとはいえない。実際に ChangeVision 社の JUDE/Professional と Sparx Systems 社の Enterprise Architect にて XMI を使い相互にモデル交換をしたところ、どちらも正しく表示できる結果は得られなかった。本システムでは、メタデータからツールに合った XMI データを作成できるようにしてツールを選ばず PMI 相当のクラスの生成ができるようにした。

・クラスの変換

クラスは BPMN のデータオブジェクトから生成されている。メタデータにはデータオブジェクト時の ID と名前が保存されており、それをクラス情報に変換している。XMI でのクラス情報は ID、名前をそのまま移すだけで XMI 表記のクラスは生成が可能である。

・関連の変換

クラス図の関連は BPMN の関連とシーケンスフローから生成されている。メタデータにはそれぞれの要素の ID と何と何を繋いでいたかが記録されている。関連はクラスと同じように直接変換することができない。関連はクラスとクラスを繋げるものだが、BPMN でデータオブジェクトとデータオブジェクトを接続するようなことはしないからである。そこで、データオブジェクト同士がペアになるまで BPMN 上の関連やシーケンスフローを辿る方式を取っている。このように関連やシーケンスフローを辿ってデータオブジェクトを探すことで、ビジネスプロセス上で最も近いデータオブジェクト同士をつなげることができる。ビジネスプロセス上で

位置に近いデータは関連が深いと考え、この方法をとっている。

## 5. 検証と評価

### 5.1 検証項目

提案に基づいて開発したシステムとその過程で自動生成したモデルについて次の項目の検証を行う。

- (1) ビジネスプロセスから概念的なクラス図の自動生成の検証
- (2) 通信販売業務におけるシステム開発手順の検証

### 5.2 ビジネスプロセスから概念的なクラス図の自動生成の検証

自動生成したクラス図を異なる UML ツールを用いて生成を行った。する。その際に同じく表現されているか比較を行う。検証用に作成した通信販売業務のビジネスプロセスを BPMN で表現したものを Fig.8 とする。これは、顧客からの購入申込書を通信販売業者が受け取ることがトリガーとなりビジネスプロセスが開始する。通信販売業者には受付係、販売係、経理係、在庫管理の 4 つのレーンが存在している。通信販売業者が購入申込書を受け取ると、それは受付係にまわされる。購入申込書から受付係が顧客情報の登録と注文票の作成を行う。次に経理係が顧客からの入金を確認し、入金伝票を作成する。入金を確認されたら発送依頼票を作成する。発送依頼票を元に商品管理情報を用いて在庫の確認をし、在庫に不足がなければ商品の出庫処理を行う。最後に顧客宛に納品書を作成する。

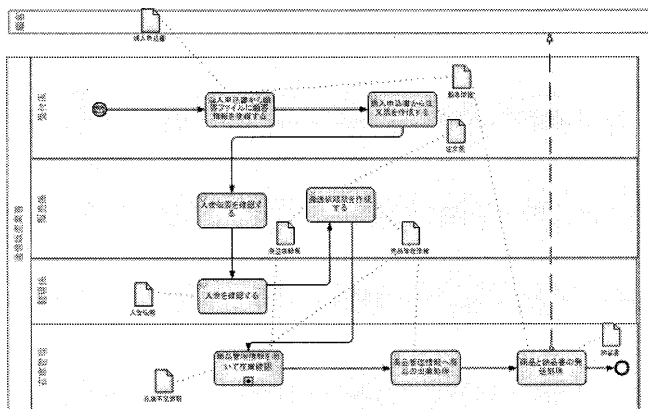


Fig.8 販売業務の BPMN

これを変換し、概念的なクラス図を自動生成して JUDE Professional で生成したところ、Fig.9 の通りになった。また、Enterprise Architect で生成したところ、Fig.10 のようになった。

どちらのツールでもクラス図を生成することができた。しかし、JUDE Professional と Enterprise Architect を比較したときに、納品書クラスと在庫不足クラス間で関連の有無に違いが生じた。まだ完全にはツール独自の XMI 表記の差が取りきれていないと考えられる。

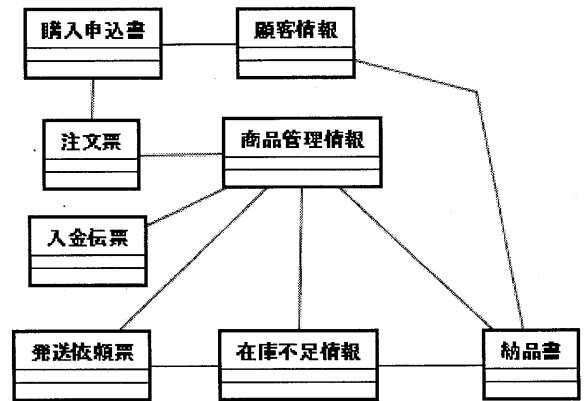


Fig.9 概念的なクラス図(JUDE Professional)

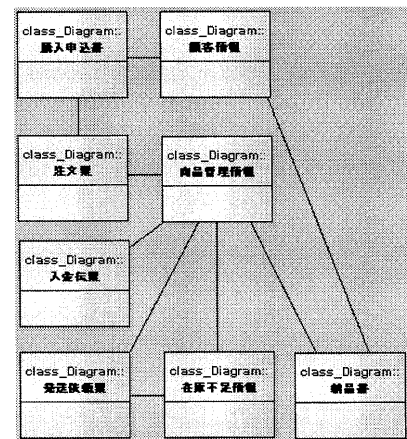


Fig.10 概念的なクラス図(Enterprise Architect)

### 5.3 通信販売業務におけるシステム開発手順の検証

自動生成された概念的なクラス図の正当性を検証すべく、実際に手動で設計レベルのクラス図へと変換を進めていった。Fig.8 から生成した Fig.9 のクラス図に BPMN の各レーンに合った画面を追加した図が次の Fig.11 である。

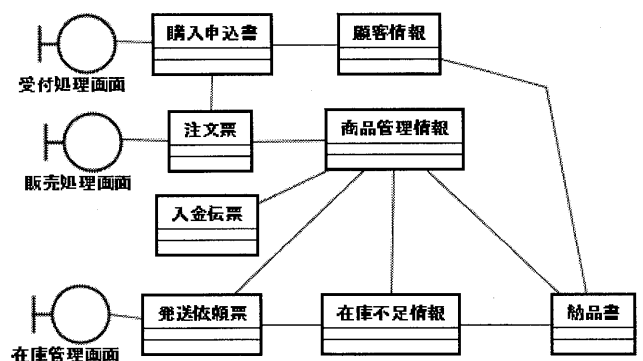


Fig.11 販売業務の画面追加クラス図

この図は各レーンにおけるシステムとのインターフェースを表している。受付係が購入申込書を作成する際には受付処理画面を使用するため、購入申込書のクラスに **Boundary** として受付処理画面を追加している。同じように販売係が注文票から発送依頼票を作成する際に販売処理画面を使用し、在庫管理が発送依頼書を用いて納品書を作成する際に在庫管理画面を使用している。これをよりシステムの実装に近づけるべく細かく記述していく。**Boundary** をクラス化し、各クラスの属性や操作も追記していくことで、概念レベルのクラス図から設計レベルのクラス図にする。この時の操作は、BPMN 図のプロセスに対応して追記すると共に不足分の追記を行う。この図に更にコントローラを付け加えた形が Fig.12 である。

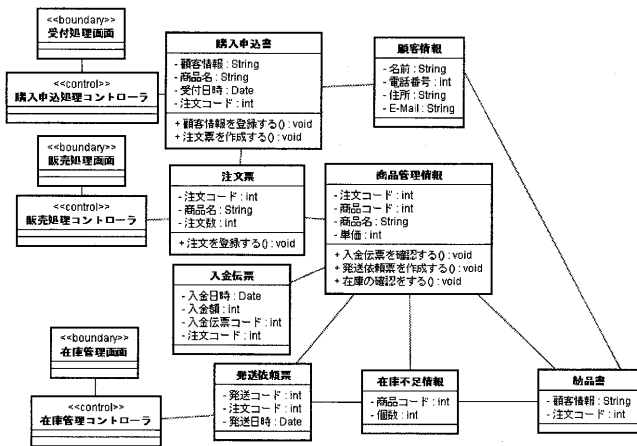


Fig.12 通信販売業務の設計レベルクラス図

この設計レベルのクラス図に更に手を加えることで、最終的には詳細レベルのクラス図へと落とし込んでいき、実装へ行き着くことが可能となる。

### 5.3 考察

検証の対象として挙げた項目について考察する。

#### (1) クラスの作成の正当性の評価

今回はビジネスストーリー上から名詞を抜き出してクラスを考える「名詞抽出法」を使って BPMN のデータオブジェクトを作成し、そのデータオブジェクトを変換、システムによりデータオブジェクトからクラスへの自動生成を行った。複数の BPMN からクラス図を生成したところ、全ての BPMN からクラス図を生成することができた。しかし Fig.9 と Fig.10 のように一部の関連に不具合が発生した。これはまだツール独自の XMI 表記の差が取りきれいていないと思われる。各ツールの独自表記を解析し、どのようなツールにも対応できるように修正していく必要があると考える。

#### (2) 作成されたクラスからシステム構築の手順の評価

BPMN からクラス図の自動生成を可能にしたことに加え、この概念的なクラス図を元に、手を加えることで設計レベルのクラス図の作成までを示した。これに更に手を加えること

で詳細レベルのクラス図とし、実装へと繋げていくことが可能となった。今回はこの設計レベルのクラス図から詳細レベルのクラス図を設計は手作業で行ったが、ここを自動化することが可能になればビジネスプロセスから実装までの一連の手順が自動化されることになる。この自動化の手順を確立することがこれからの課題である。

## 6. まとめ

日々複雑化しているビジネスプロセスを全体的に把握し、より良いものにマネージメントするためにはモデリングを行い、可視化することが効果的である。それにより、顧客の曖昧な要求や、顧客自身が見えていない潜在的な要求まで発見することが可能になる。そうすることで、情報システムの構築もより確かなものになる。

本研究による MDA を用いた情報システム構築の手順によって、ビジネスプロセスモデルからメタデータの作成、異なったツールでの概念的なレベルのクラス図まで自動変換することができた。これは CIM から PIM を自動生成し、システム実装へと繋げていく手順の 1 つである。また、自動生成したクラス図に手を加えることで設計レベル、詳細レベルのクラス図へと落としこんでいき、実装へと繋げることができた。

今後の課題として、現在の概念的なクラス図をより良く自動生成できるようにし、実装まで考えた具体的な手順を確立することである。

## 文 献

- [1] 加藤正人, BPMN によるビジネスプロセスモデリング入門, ソフト・リサーチ・センター, 2006
- [2] 小泉寿男, 辻秀一, 吉田幸二, 中島毅, ソフトウェア開発, オーム社, 2003
- [3] ハインリヒ ザイドルマイヤー, ARIS によるビジネスプロセス・モデリング, BNN 新社, 2004
- [4] Cris Kobryn, 安藤進, Journal of Information Processing Society of Japan, Vol.41, No.3, pp.238-266,
- [5] 猪股順二, 藤川泰之, 前田隆之, 松田友隆, 吉田裕之, UML2.0 のすべて～仕様書徹底解剖と MDA 関連技術～, ソフト・リサーチ・センター, 2006
- [6] オージス総研, かんたん UML, 翔泳社, 1999
- [7] 若林隆久, 改訂 UML 演習 UML2.0 対応, 株式会社 SCC, 2005
- [8] 大澤貴彦, 片岡信弘, 連携するシステムを構築するためのビジネスプロセスモデルを用いた開発方式の提案と検証, SWIM 研究会第 4 回研究会, Jan.2006