

# CRCカードによる 業務とITの同時設計

2003/5/10

ITコーディネータ： 小林 正和

CRC: Class Responsibility Collaboration

# 目次

はじめに

序章：経営におけるIT活用について

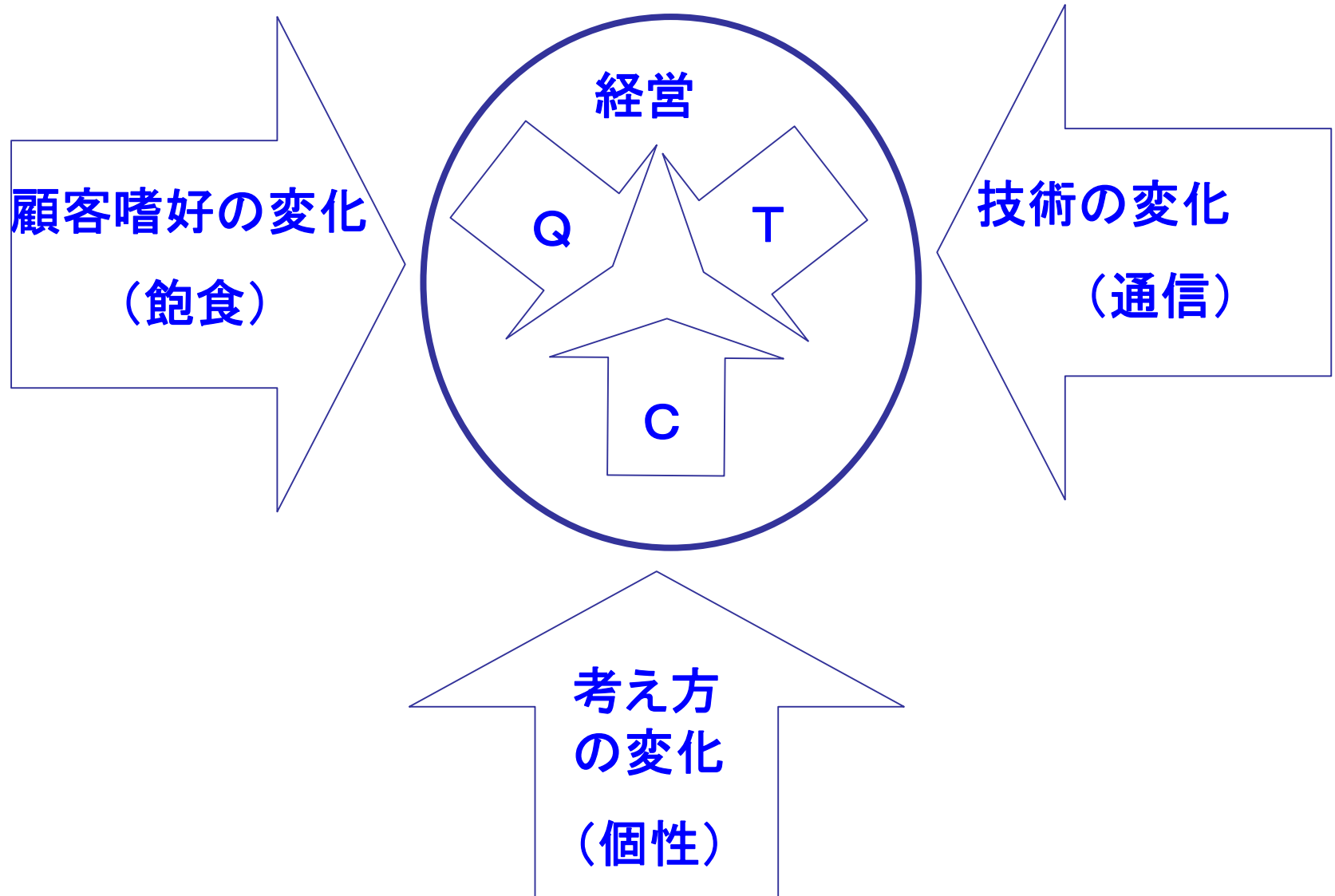
第1章：ビジネスプロセスモデル化の手法

第2章：CRCカード法の進め方

まとめ

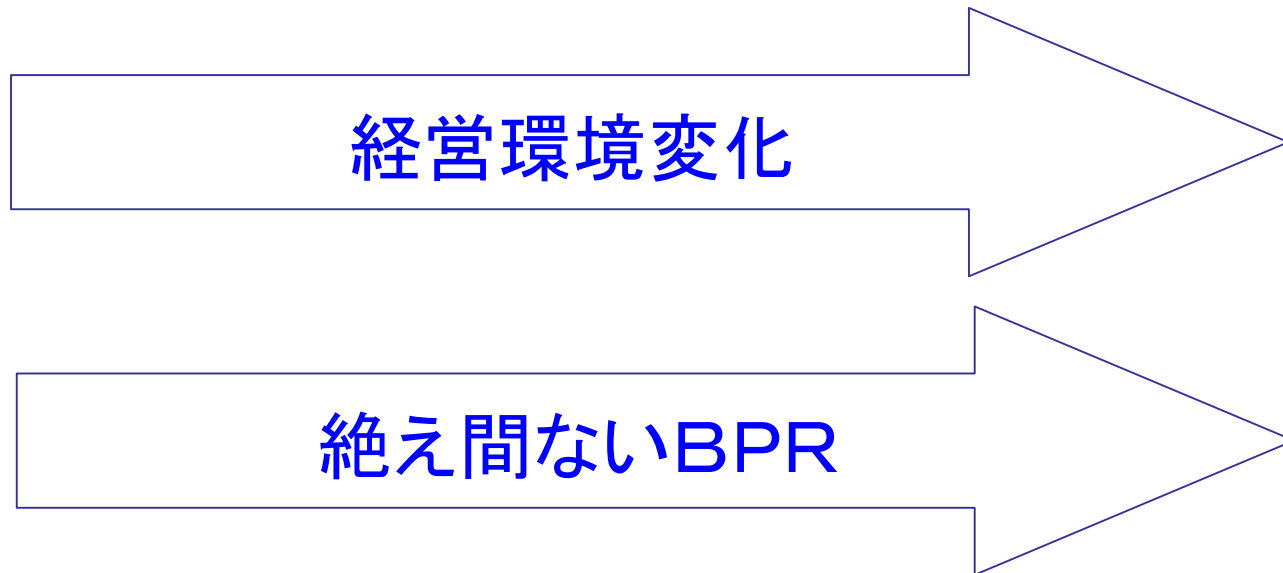
# 序章 : 経営におけるIT活用について

# 1. 現在の経営環境



## 2. 経営環境変化への即応

時々刻々変化する経営環境変化に即時に対応し続けられる(持続性のある)経営の仕組み(システム)を確立しなければ、企業は存続し得ない。

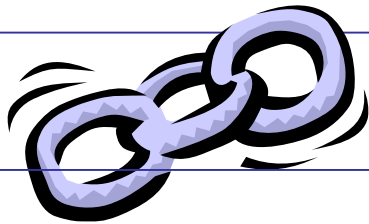


BPR: Business Process Reengineering

### 3. 足枷を取り除く

経営環境変化

絶え間ないBPR



重石

(保守的経営、組織の壁、情報システム、...)

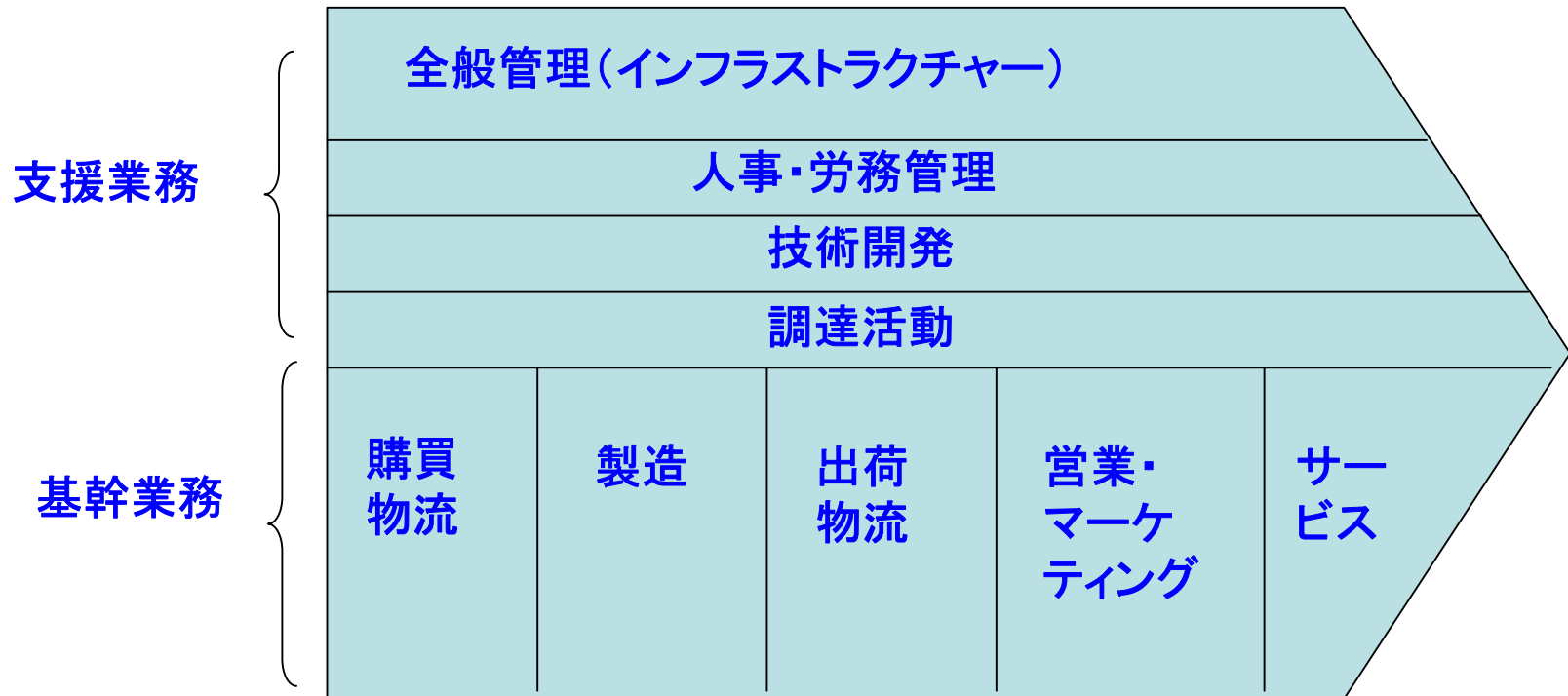
## 4. 新しい考え方とアプローチの必要性

- (1) まづ、BPR、
  - (2) 戦略情報化企画、
- では、間に合わない。

BPRと情報化企画を同時に行なう新しい考え方と方法が必要である。

# 5. ビジネスプロセス

## ビジネスプロセス(BP)





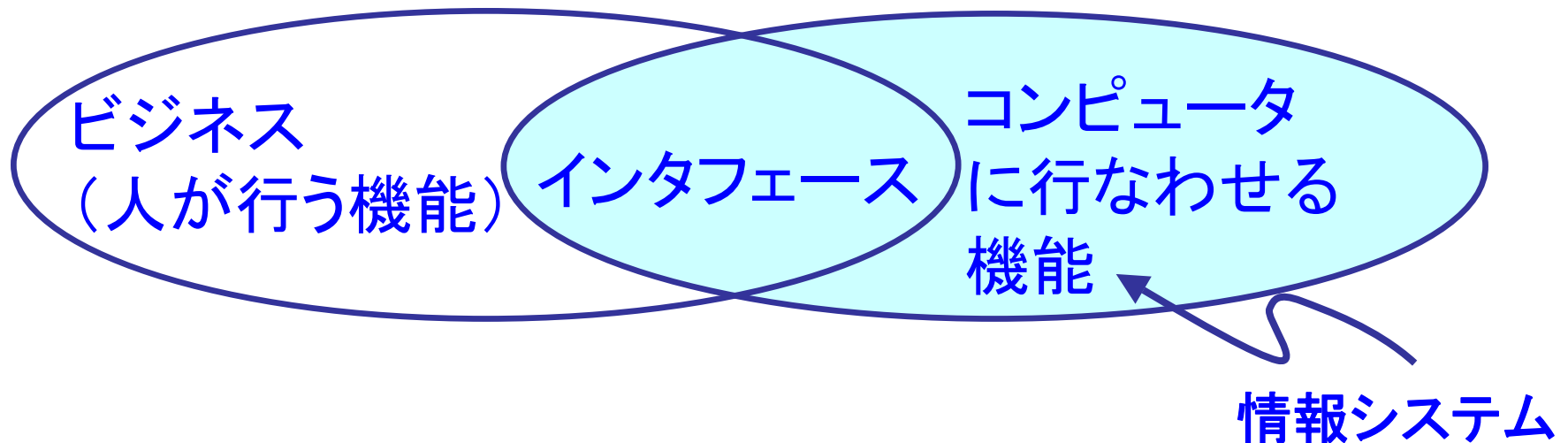
## 6. ビジネスプロセスの機能の捉え方

ビジネスプロセスの機能には2種類ある。

(1) 人が行なう機能、

(2) マシン(コンピュータ等)が行なう機能

この2機能の差は、実施するのが、人かマシンかだけである。



## <補足>

ビジネスプロセスを構成する機能で、人が実施する機能とマシンが実施する機能との間に本質的な差はない。

この感覚を掴むには、電子メールの返信応答を考えると良い。マシンによる自動応答か、人による応答かほとんど区別がつかない。利用者としては、マシン応答か、人による応答かは、目的が達成できればどちらでも良い事である。

情報・サービス提供側が、マシンで行なえる事を、人手で行なうのでは、おそらく採算が取れず、競争に勝てないだけのことである。

# 7. 従来の問題点とその解決策

経営企画部門: 経営戦略の企画・立案

経営者: 経営の遂行

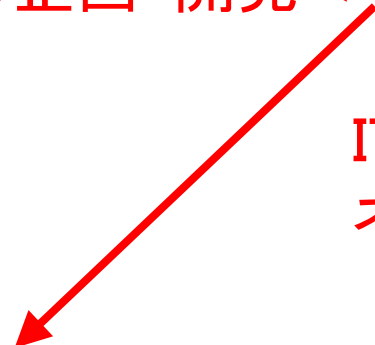
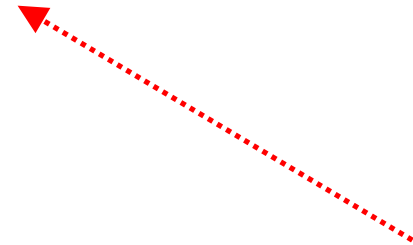
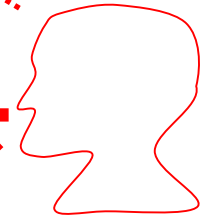
経営システム部門: 経営システム構築の企画・開発

Missing Linkを埋める

ITコーディネータ

情報システム部門: 情報システム構築の企画・開発/調達・運用

ライン: ライン業務の遂行



## 8. エンタープライズ・アーキテクト

ITコーディネータは経営システム構築の企画・開発の支援を行なう「エンタープライズ・アーキテクト」としての能力を持つ必要がある。

そのためには、技法・ツールを身につけなければならない。

# 第1章:ビジネスプロセスモデル化の手法

# 1. ダイレクトマッピング

ビジネスプロセスを構成する機能を、その実現手段(人／マシン)の区別なく、モデル化し、モデルの構成を変えないで、そのままコンピュータが実行できるソフトウェアのモデルに変換して、コンピュータの内部に実世界のビジネスプロセスモデルを構築する技法。

その基盤となるのが、オブジェクト指向技術

## 2. ダイレクトマッピングの手順

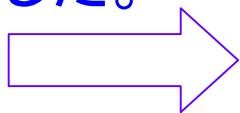
- (1) ビジネスプロセスのすべての機能を人が行なうものとしてモデル化する
- (2) ビジネスプロセスの機能のうち、手順が決まっており、コンピュータに実行させた方が正確かつ効率的である仕事と、
- (3) 手順をあらかじめ決めることが出来ず、人がその時々  
状況に応じて判断を行わなければならない仕事とを区別する
- (4) コンピュータに実行させる仕事は、そっくりそのままソフトウェア化する
- (5) 人が行う機能とソフトウェア化した機能との間のインタフェースを定める。

### 3. オブジェクト指向技術の本質

初心忘れるべからず(少し歴史を振り返って)

最初に、オブジェクト指向技術が登場したのは、1967年、SIMULA67というシミュレーション用の言語である。実世界の忠実なシミュレーションを行なうには、実世界の忠実なモデルをコンピュータの内部に作り、それを模倣的に稼動させるのが最も良い(ダイレクトシミュレーションという考え方)。

さらに、1970年ころ、人工知能の研究の一環として、MITのC.Hewittが、メッセージパッシングを主体とした計算モデルであるACTORモデルを提案している。さらに、1972年に、CMUのD.L.Parnusが、情報隠蔽(カプセル化)の考え方を提唱し、1974年にMITのB.Liskovが、抽象データ型の機能を持ったCLUという計算機言語を開発した。



計算機言語モデルに端を発し、分析、設計技法に発展した



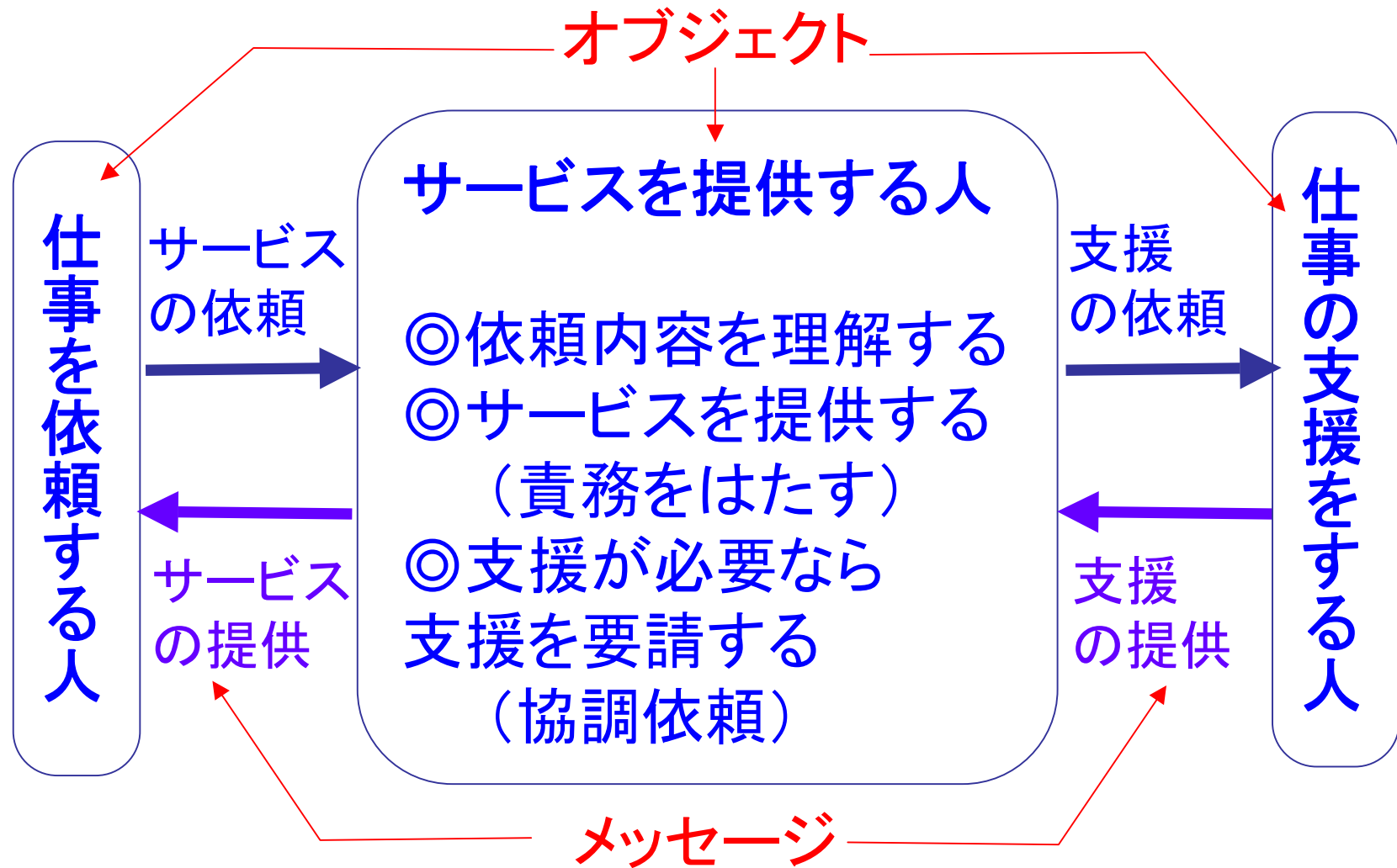
## 4. オブジェクト指向技術の本質

(1) メッセージパッシング、

(2) カプセル化が、

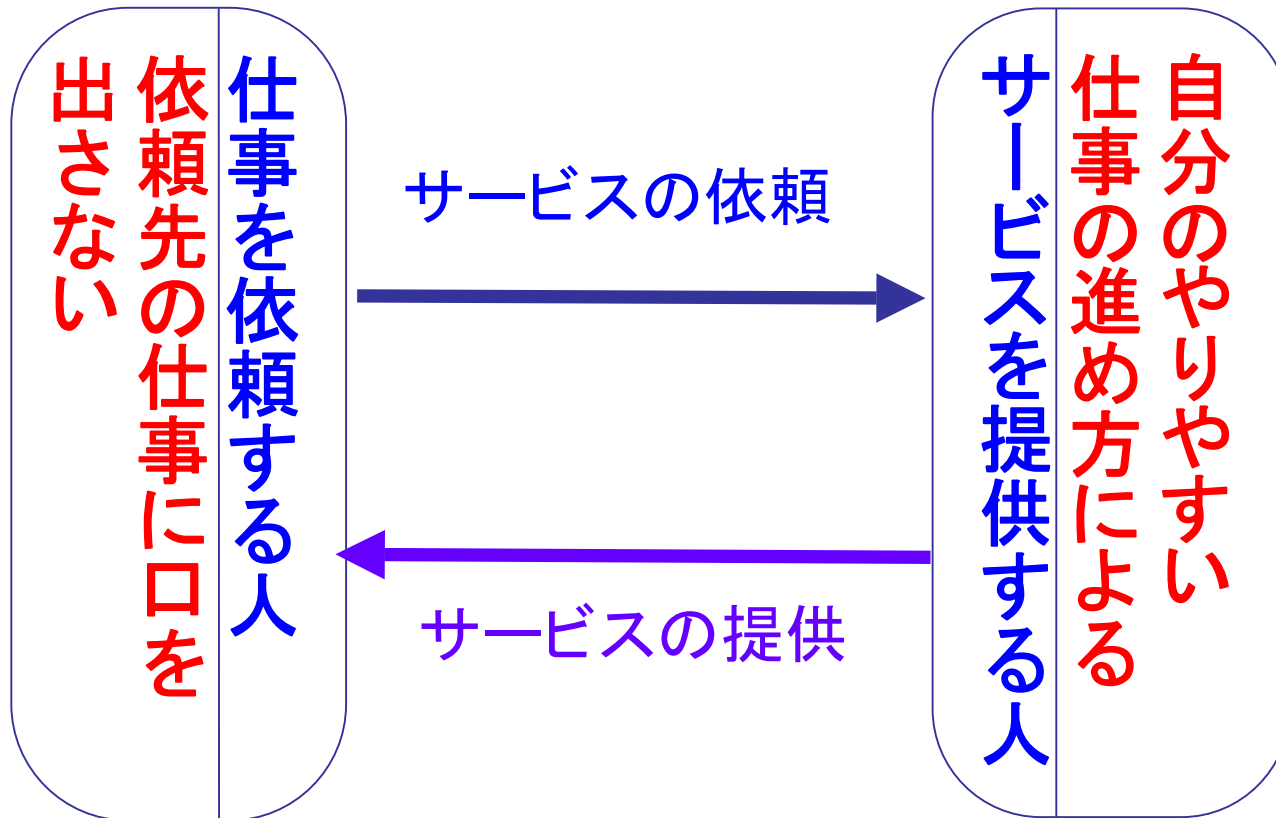
オブジェクト指向モデル化の基本概念である。

# 5. オブジェクトとメッセージパッシング



我々が仕事をする時のやりかたと同じ

## 6. カプセル化の本質



人に仕事を頼むときの原則

# 7. オブジェクト指向技法は何故難しいか

## (1) 導入教育が間違っている

最初に、汎化とか、抽象化から理解させようとする(オブジェクト指向の本質を教えない)

## (2) 用語が難しすぎる

クラス、インヘリタンス、ポリモルフィズム...  
頭がいたくなる

## (3) オブジェクト指向技法をマスターしたものが、 初心者を教えられない

## <補足>

ただし、オブジェクト指向技法を完全にマスターして、ビジネスアーキテクチャー設計(ビジネスプロセスの中にあるさまざまな活動の要素間の関係を適切なものに)することにはたいへん難しい側面がある。

人によっては、オブジェクト指向技法を理解できない人もいる。これが、「オブジェクト指向技法をマスターしたものが、初心者を教えられない。」という事に繋がっている。

しかし、ビジネスプロセスのモデル化(分析)にオブジェクト指向技法使うだけであれば、導入さえ間違わなければ、ほとんど誰にでも容易にできる。

## 8. モデル化の前にやって置くべきこと

(1) 経営のビジョン、ミッションから、SWOT分析等により、コアコンピタンス、コア技術、コア商品などを抽出し、強みを機会にぶつける戦略と、価値を生まない業務や弱みには他からの支援により補強する戦略を立案する(ビジネスモデルの作成)。

(2) ビジネス遂行の概略のシナリオ(ビジネスシナリオ)をいくつかつくる(ビジネスモデルのシナリオによる検証)。

(3) ビジネスシナリオを実現するビジネスプロセスのラフスケッチを作り、自社で行う機能と、外部からの支援を得る機能のおおまかな切り分けを行なう。

## <注記1>

これから生き延びる企業は、他社にない独自の能力（コアコンピタンス）、コア技術、コア商品などを持つところであると言われている。

すなわち、他社に追従を許さない価値（何らかのサービスといえる）を提供する事が、企業の「責務（Responsibility）」の中心であると見なされる。

これに対して、他の支援を得なければならない機能は、「協調（Collaboration）」を行うのが望ましい。

この「責務」を持つ機能と、「協調」を行う機能とを抽出して行くのが本日お話する技法の基本的な点である。

## <注記2>

本日お話しする技法の本質は、何らかのサービスを提供するもの(サービスを提供する「責務」をもつ主体)が、互いに、「協調」しあって仕事を行なって行く事にある。

サービスを提供するものが、人であってもよいし、マシンであっても良い。

このように考えれば、ビジネスプロセスモデルを作成し、その構成要素である機能の実現手段を人か、マシンか決めれば、戦略情報化企画が完了してしまう。

あとは、効率の良い、実装を考えれば良い。



## 9. ビジネスプロセスのモデル化の手順(1)

(1)モデル化の対象になる仕事(機能)がどのように行なわれるか書き出してみる。手順的な記述でよい。例外処理・異常処理は、さしあたり記述しない。

<記述手段は問わない>

・日常語で列挙、業務フロー、DMM、DFD等

(2)上記(1)の記述の中に出てくる名詞をリストアップする。これらの名詞の中から、

- ・自社にとって大切なもの(管理が必要)、
- ・自社にとって大切な仕事(なんらかのサービスを提供する仕事)を表すものを抽出する。

(3)上記(2)の「仕事」については、さらに、上記(1)→(2)を繰り返す。

## <補足>

一般的に行なわれているオブジェクト指向の教育では、ビジネスプロセスの分析に、人間がもっとも慣れている手順的思考を最初から行なわないようにするため、最初から敷居が高くなる。

そもそも、ビジネスプロセスは、手順的要素が強いもので、DFD(情報の変換系列)で表す事ですら、訓練を要する。

まして、最初から、オブジェクト指向モデルを作ろうとするのは無理がある。

入りやすい方法で入って、結果がオブジェクト指向によるモデルになれば良い。

## 10. ビジネスプロセスのモデル化の手順(2)

(4) 以上により、企業として大切な管理の対象(名詞)や処理の対象(名詞)が抽出される。次に、管理するとか処理するとかが、対象に対してどのような具体的な仕事を行なうのか明らかにする(オブジェクト指向の用語を使えば、メソッドを明らかにすることになる)。手順的な記述でよい。

上記により「管理・処理の対象とその対象に対する管理・処理の具体的な内容」が明らかになる。

これを**オブジェクト(あるいは、ケース)**という。このオブジェクトは見方を変えれば、他に対して、ある対象に対する管理・処理を行なう「責務」を持つ主体であり、他からみれば、「責務」をはたすサービスを提供する主体と考えられる。

# 11. ビジネスプロセスのモデル化の手順(3)

(5) (4)により抽出された、オブジェクト(あるいはケース)のステークホルダーを明らかにする。ステークホルダーとは、オブジェクトと何らかの関係を持つ主体である。この場合、ステークホルダーは、組織名や人名のようなものではなく、オブジェクトとかかわる「役割」で記述することが重要である。このステークホルダーは、UMLでは、**アクター**と呼ばれている。

(6) 上記、(5)で抽出した、ステークホルダーのうち、自社内か、社外かを区別する。自社内のステークホルダーについては、どのような役割をはたすか、上記(1)に戻って、分析を繰り返す。

## 12. ビジネスプロセスのモデル化の手順(4)

(7) (4)により抽出された、オブジェクト(あるいはケース)が果たす「責務」が、他者の支援を受ける必要がある場合、その支援機能については、誰(正確にはどのオブジェクト)に、どのような支援(「協調」)を受けるかを明らかにする。

ここで、支援機能サービスを提供するオブジェクトが既に抽出されている場合と、そうでない場合とがある。抽出されていない場合は、支援機能サービスを提供するステークホルダーが抽出されたことになるので、上記(6)を繰り返すことになる。

## 13. ビジネスプロセスのモデル化の手順(5)

(8) ビジネスの逐次的シナリオを作成する。

- ・まづ、最も普通のビジネスのシナリオ
- ・良く起こりがちな異常・例外処理のシナリオ
- ・まれに起こる異常・例外処理のシナリオ

の順に作成する。

(9) ビジネス逐次的シナリオによるモデルシミュレーションモデルが上手く機能するかどうかを、ビジネスの各種シナリオによってシミュレートすることにより検証する。

## 14. CRCカード法

(Class Responsibility Collaboration)

以上に述べた一連の手順を

- (1) CRCカードというカードを使用し、
- (2) ビジネスシナリオによるシミュレーションを  
ロールプレイングゲームによって行ない、  
ビジネスプロセスのモデルを構築する方法が、  
CRCカード法である。

## 第2章：CRCカード法の進め方



# 1. CRCカード法の手順

1. プロジェクトの目標を明確にする
2. チームメンバーの選出
3. チームメンバーを5日間の専従とする
4. 専用ルームを設置し、機材を整える
5. 目標と範囲(スコープ)の合意をとる
6. 逐次的ビジネスシナリオの作成
7. シナリオを動かす上で必要なオブジェクトの候補をリストアップ
8. ユースケース図を作る
9. オブジェクト(クラス)の候補を選出する
10. 各クラス毎の責務(Responsibility)を明らかにする
11. 各クラス毎の協調関係Collaborationを明らかにする
12. チームの協議検討(Talk-Through)によりクラスに漏れなくす
13. ユースケース記述を完成させる
14. ビジネスプロセスのシナリオに沿って  
    ロールプレイリハーサル(Walk-Through)を行なう
15. ビジネスプロセスのシナリオに沿って  
    ロールプレイ(Run-Through)を行ない完成を確認する

# 1. プロジェクトの目標を明確にする

- (1) 経営のビジョン、ミッションを実現する  
ビジネスモデルを描くまでは完了している  
ものとする。
- (2) BPRの方向性(ビジョン、ミッションとの整合性  
及びビジネスモデルとの整合性等)を確認する。
- (3) BPRの範囲を確認する。

## 2. チームメンバーの選出

(1) メンバー構成は、

- リーダー、進行役(1名か、それぞれ1名)
- BPR対象業務の精通者(1~2名程度)
- オブジェクト指向モデル化専門家(1から2名程度)
- 記録係(1名)

(2) メンバー総数は5~7名程度

(3) リーダーは、

- グループ作業指導技能保有、
- 業務改革経験
- オブジェクト指向モデル化知識保有

(4) 記録係は、

- オブジェクト指向モデル化訓練を受けている。

### 3. チームメンバーを5日間の専従とする

下記スケジュールをこなすため専従とする。

第1日目	シナリオを作る	オブジェクトを探す
第2日目	責務を割り当てる	関係を定義する
第3日目	Talk through	モデルの拡張
第4日目	Walk through	モデルの改良
第5日目	Run through	モデルの最適化

## 4. 専用ルームを設置し、機材を整える

- (1) くつろいで作業に集中できる会議室の確保
  - ・・・外部との電話、呼び出し、メール等一切遮断
- (2) チーム全員が一ヶ所で協同作業できるテーブル用意
  - ・・・あまり広すぎないこと
- (3) テーブルの周りを楽に動き回れるスペースと、  
テーブルを離れてロール・プレイングが出来るスペース  
確保
- (4) ホワイトボードと、十分なマーカー、模造紙、  
ポストイット等

## 5.目標と範囲(スコープ)の合意をとる

- (1) チーム編成に先立って検討したプロジェクトの目標を再確認し、チームメンバー全員の合意を得る。
- (2) チームで検討する業務範囲(スコープ)を再確認し、チームメンバー全員の合意を得る。

## 6. 逐次的ビジネスシナリオの作成

(1) DMMなどを用いてビジネス機能を抽出する。

(2) 逐次的シナリオを作成する。

下記を別々に作成する。

- 定常的、正常なシナリオ（主シナリオ）
- しばしば起こる例外的なシナリオ
- まれにしか起こらない例外的なシナリオ

## 例題：酒屋倉庫問題(二村他：情報処理 vol.26 No.5 pp.458-459)

ある酒販売会社の倉庫では、毎日数個のコンテナが搬入されてきます。その内容はビン詰め酒で、一つのコンテナには10銘柄まで混載できます。扱い銘柄は約200種類あります。倉庫係は、コンテナを受け取り、そのまま倉庫に保管し積荷票を受付係へ渡します。内蔵品は別のコンテナに詰め替えたり、別の場所に保管することはありません。空になったコンテナはすぐに搬出されます。

<積荷票：コンテナ番号(5桁)、搬入年月、日時

内蔵品名、数量(の繰り返し)>

受付係は毎日数10件の出庫依頼を受け、その都度、倉庫係に出庫指示書を出すことになっています。出庫依頼は出庫指示票または電話によるものとし、1件の依頼では、1銘柄のみに限られます。在庫がないか数量不足の場合には、その旨、依頼者に電話連絡し、同時に在庫不足リストに記入します。そして、当該品の在庫が必要になった時点で、不足品の出庫指示をします。また、空になる予定のコンテナを倉庫係に知らせることになっています。



< 続き >

< 出庫依頼: 品名 数量 送り先名 >

< 出庫指示書: 注文番号 送り先名

コンテナ番号、品名、数量、空コンテナ搬出マーク(の繰り返し) >

< 在庫不足リスト: 送り先名 品名 数量 >

このような受付係の仕事(在庫なしの連絡、出庫指示書および在庫不足リスト作成)のための計算機プログラムを作成しなさい。

補足: 移送や倉庫保管中の酒類の損失は生じません。この課題は現実的でない部分もあるので、入力データのエラー処理などは簡略に扱ってかまいません。以上の曖昧な点は、適当に解釈してください。

< この例題は、多くの手法の評価するときに使われてきた共通問題です。 >

## 7. シナリオを動かす上で必要な オブジェクトの候補をリストアップ

- (1) シナリオに出てくる名詞を中心にオブジェクトの候補になりそうな名詞をリストアップする。(チームで順繰り(round robins)に全員が候補を挙げられなくなるまで続ける。)
- (2) シナリオに出てこなくとも、出てきて良い様な語はすべてリストアップする。

## 例題：酒屋倉庫問題（二村他：情報処理 vol.26 No.5 pp.458-459）

ある酒販売会社の倉庫では、毎日数個のコンテナが搬入されてきます。その内容はビン詰めの酒で、一つのコンテナには10銘柄まで混載できます。扱い銘柄は約200種類あります。倉庫係は、コンテナを受け取り、そのまま倉庫に保管し積荷票を受付係へ渡します。内蔵品は別のコンテナに詰め替えたり、別の場所に保管することはありません。空になったコンテナはすぐに搬出されます。

<積荷票：コンテナ番号（5桁）、搬入年月、日時

内蔵品名、数量（の繰り返し）>

受付係は毎日数10件の出庫依頼を受け、その都度、倉庫係に出庫指示書を出すことになっています。出庫依頼は出庫指示票または電話によるものとし、1件の依頼では、1銘柄のみに限られます。在庫がないか数量不足の場合には、その旨、依頼者に電話連絡し、同時に在庫不足リストに記入します。そして、当該品の入庫が必要量あった時点で、不足品の出庫指示をします。また、空になる予定のコンテナを倉庫係に知らせることになっています。

<続き>

<出庫依頼:品名 数量 送り先名>

<出庫指示書:注文番号 送り先名

コンテナ番号、品名、数量、空コンテナ搬出マーク(の繰り返し)>

<在庫不足リスト:送り先名 品名 数量>

このような受付係の仕事(在庫なしの連絡、出庫指示書および在庫不足リスト作成)のための計算機プログラムを作成しなさい。

補足:移送や倉庫保管中の酒類の損失は生じません。この課題は現実的でない部分もあるので、入力データのエラー処理などは簡略に扱ってかまいません。以上の曖昧な点は、適当に解釈してください。

<この例題は、多くの手法の評価するときに使われてきた共通問題です。>

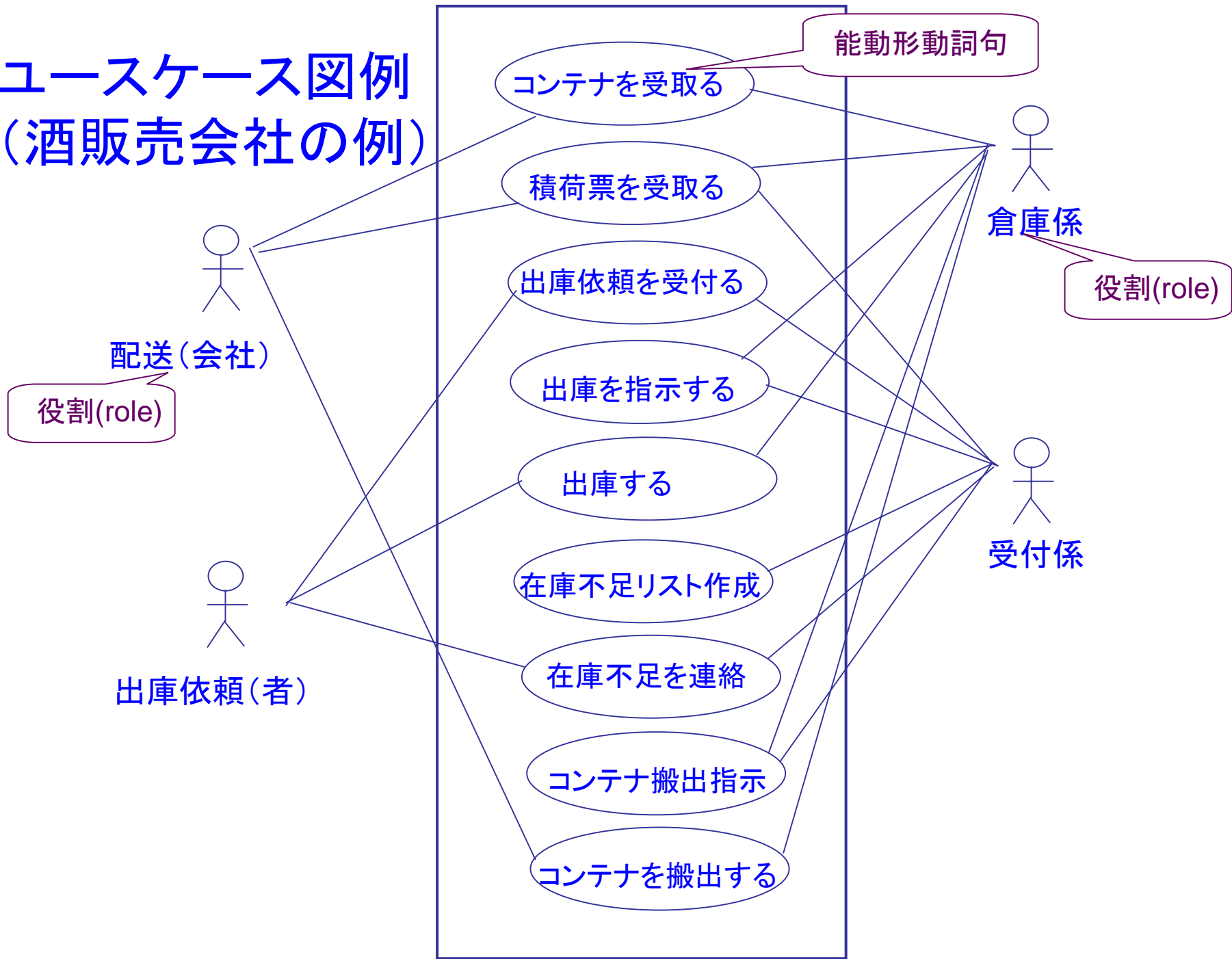
酒販売会社  
倉庫  
コンテナ  
ビン  
酒  
銘柄  
倉庫係  
積荷票  
受付係

出庫依頼  
出庫指示書  
空コンテナ搬出マーク  
出庫指示票  
電話  
依頼者  
在庫不足リスト  
出庫指示  
計算機プログラム

## 8. ユースケース図を作る

- (1) ユースケースは、能動型動詞句で表現する。
- (2) ステークホルダー(アクター、登場人物)は、役割(ロール)で表す。
- (3) 出入りのような関係は良くチェックする。

# ユースケース図例 (酒販売会社の例)



## 9. オブジェクト(クラス)の候補を選出する

- (1) シナリオに出てくる名詞を中心にリストアップされたオブジェクト(クラス)の候補の中からオブジェクト(クラス)を抽出する。
- (2) オブジェクトの中で、能動的なもの、受動的であってもそのオブジェクトのために責任者、管理者等を置く事が望ましいような重要なものを選択する。
- (3) CRCカードにオブジェクトを代表する名前をクラス名として記入する。



コンテナ(搬入、搬出)・・・管理者(倉庫係)  
酒(銘柄)(入荷、在庫、空)・・・管理者(受付係)  
ビン(在庫、出庫)・・・管理者(倉庫係)  
出庫依頼(出庫依頼、出庫、在庫不足連絡)・・・管理者(受付係)  
積荷票・・・管理者(倉庫係、受付係)  
在庫不足リスト・・・管理者(受付係)  
出庫指示書・・・管理者(受付係)

出庫指示票・・・出庫依頼の一部

空コンテナ搬出マーク・・・出庫指示書の一部

酒販売会社、倉庫・・・この例の場合、管理されているわけではないので対象外。

# 10. 各クラス毎の責務 (Responsibility) を明らかにする

- (1) クラスの責任者、管理者存在するものとして  
その果たすべき責務をあきらかにする。
- (2) CRCカードに責務を記述する。
- (3) 責務は多すぎないようにする。
- (4) 責務は多すぎるクラスは分割する。
  - ・・・クラス全般にわたり、責務の数はバランスするよう  
にする。

## 責務の記述例 (酒販売会社の例)

クラス名:コンテナ(管理者は倉庫係)	
責務: 受取る(搬入) 内蔵品目別数量を チェックする 搬出する	

クラス名:酒(銘柄 管理者は受付係)	
責務: 取り扱いを決める 在庫を計算する 出庫を指示する	

## 責務の記述例 (酒販売会社の例)

クラス名: 出庫依頼、出庫依頼票(管理者は受付係)	
責務: 受付ける 出庫指示する 不足品連絡をする	

クラス名: ビン(管理者は倉庫係)	
責務: 出庫する	

## 責務の記述例 (酒販売会社の例)

クラス名：積荷票(管理者は倉庫係、受付係)	
責務： 受付ける コンテナ内蔵品と突 き合せる(チェック) 入荷量を記録する	

クラス名：不足品リスト(管理者は受付係)	
責務： 不足品名と数量を 記録する 入荷量と突合せる	

# 責務の記述例 (酒販売会社の例)

クラス名 : 出庫指示書 (管理者は受付係)	
責務 : 出庫品名と数量を 記入する 空コンテナ搬出マ ークを記入する	

# 11. 各クラス毎の協調関係Collaborationを明らかにする

- (1) クラスが責務を果たすにあたり、他のクラスの協力を得る必要があれば、協力をしてくれるクラス名と協力の内容をCRCカードの協調の欄に記入する。
- (2) 責務を果たすクラスが、一部の機能を協調クラスにアウトソーシングしたとも考えられる。

## 協調の記述例 (酒販売会社の例)

クラス名:コンテナ(管理者は倉庫係)	
責務: 受取る(搬入) 内蔵品目別数量を  チェックする 搬出する	協調: 配送 積荷票、出庫指示書  配送

クラス名:酒(銘柄 管理者は受付係)	
責務: 取り扱いを決める 在庫量を計算する  出庫指示する	協調: 積荷票、出庫指示書 不足品リスト 倉庫係



## 協調の記述例 (酒販売会社の例)

クラス名: 出庫依頼、出庫依頼票 (管理者は受付係)	
責務: 受付ける 出庫指示する 不足品連絡をする	協調: 依頼者 倉庫係 不足品連絡をする

クラス名: ビン (管理者は倉庫係)	
責務: 出庫する	協調: 依頼者

## 協調の記述例 (酒販売会社の例)

クラス名：積荷票(管理者は倉庫係、受付係)	
責務： 受付ける コンテナ内蔵品と突 き合せる(チェック) 入荷量を記録する	協調： 配送、倉庫係、受付係 倉庫係  受付係

クラス名：不足品リスト(管理者は受付係)	
責務： 不足品名と数量を 記録する 入荷量と突合せる	協調： 受付係  受付係

## 協調の記述例 (酒販売会社の例)

クラス名：出庫指示書(管理者は受付係)	
責務： 出庫品名と数量を 記入する 空コンテナ搬出マ ークを記入する	協調：受付係  受付係

## 12. チームの協議検討 (Talk-Through) によりクラスに漏れなくす

- (1) クラスの漏れ、責務の漏れ、責務の表現の修正を行なう。
- (2) 責務の多すぎるクラスや少なすぎるクラスを分析し、クラス編成を改善する。

# 13. ユースケース記述を完成させる

(1) 下記を記述する。

- ・ユースケース名、
- ・スコープ(位置付け、目的、ゴールなど)、
- ・トリガー、
- ・アクター、
- ・基本系列(成功のシナリオ)、
- ・代替系列(例外、異常時のシナリオ)、
- ・契約(事前条件、成功時の事後条件、代替系列の事後条件)など。

# ユースケース記述の例

ユースケース名: 出庫指示書

スコープ: 出庫依頼に対する処置

トリガー: 出庫依頼の都度作成

実施者(主アクター): 受付係

協力者(副アクター): 倉庫係

事前条件: 在庫数量が出庫依頼数量を上回っていること  
          コンテナ別在庫数量が把握できていること。

基本系列: (1)在庫確認、(2)倉庫係に出庫指示、  
          (3)空コンテナ搬出指示

代替系列: 倉庫係から現品がない旨連絡があった時の処理など。

成功時の事後条件: 倉庫係に出庫指示書発行、  
          空になるコンテナがある場合は搬出指示。

## 14. ビジネスプロセスのシナリオに沿ってロールプレイのリハーサル(Walk-Through)を行なう

- (1)シナリオを動かす上で必要なクラスに漏れや矛盾がないかどうかチェックする。
- (2)クラスに漏れや矛盾を改善する。

## 15. ビジネスプロセスのシナリオに沿ってロールプレイ(Run-Through)を行ない完成を確認する

## 参考文献：

- 1) **David A. Taylor** :  
**Business Engineering with Object Technology**  
**John Wiley & Sons; (January 1995)**
- 2) **青山幹雄、中谷多哉子** :  
**オブジェクト指向に強くなる(技術評論社 2003)**  
**第2章 pp.107-156**
- 3) **David Bellin et.al. (デビッド・ベリン他)** :  
**実践CRCカード(ピアソン・エデュケーション 2002訳)**

**CRC法ケース研修** : **JUAS**で実施、問合せ先

**TEL: 03-3249-4102、e-mail: seminar@juas.or.jp**



# まとめ

- ・ 激変する経営環境に適応して、企業が発展的に存続するためには、経営の組織、プロセスが俊敏に変わる必要がある。
- ・ 経営の組織、プロセスの変化と、情報システムが同時に変化出来なければならない。
- ・ そのためには、経営をオブジェクトモデル化し、ビジネスのシナリオを描き、ビジネスオブジェクトモデルが機能するかどうか、常にシミュレートし、検証する必要がある。
- ・ 業務精通者と、ビジネスオブジェクトモデル化技術者とが、一同に会して実施するCRCカード法は、関係者間でコンセンサスの得られたビジネスオブジェクトモデルを効率的に作成するために有効と思われる。