

## オブジェクト識別についての一考察とその効果

加藤 貞行

オーロラシステム設計事務所 DATARUN シニアコンサルタント

〒113-0033 東京都文京区本郷 4-2-5 MA ビル

電話：03(5804)6621, E-mail：[datarun@aurorasd.co.jp](mailto:datarun@aurorasd.co.jp)

オブジェクト指向パラダイムの問題はオブジェクトの識別が難しいことである。理由は発見的手法によっているためである。筆者らは、オブジェクトを基本データの容器と考える方法論：DATARUN を導入することで、この問題を解決出来た。しかし、習熟度の高い技術者には基本データを生成するビジネス・プロセスの識別が困難という新たな問題に直面した。筆者は問題解決のためにビジネス・プロセス識別のガイドラインを作成したが、ガイドラインは満足な結果を得ることが出来た。ビジネス・プロセスはビジネス遂行のための過程として捉え、ビジネス・プロセスの数は 12 個を限度にモデリングすべきである。ビジネス・プロセスは、「What, Who, Where, When または Why」と表現可能な発生要因を持ち、その名称は引き起こされる具体的な過程とすべきである。

## The study about definition of object, and inspection of the effect

Sadayuki Kato

Senior Consultant of DATARUN  
Aurora System Design office Co., Ltd.

MA Building 4-2-5 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan  
Phone: +81-03-5804-6621, Email: [datarun@aurorasd.co.jp](mailto:datarun@aurorasd.co.jp)

Object Oriented Paradigm has the problem that to define object is very difficult, because it depends on heuristic way. We could solve the problem using DATARUN that define object as container of primary data. However, I was in face of another problem that to define business process that generates primary data is difficult for engineer who isn't practiced in DATARUN. I had designed the guideline about definition of business process, and applied to some projects. Then I have been able to quite satisfy with the results of applications of the guideline. I have proposed it business process should recognize as activity of business, and the number of business process should be limit to 12. In addition business process should have trigger that is able to represent by "What", "Who(m)", "Where", "When" or "Why", and name of business process should be activity that is generated by trigger.

## 1. はじめに

オブジェクト指向パラダイムはプログラミング言語をその起源として発達してきた。そして、従来メインフレーム一色であった基幹システムの開発においても、オブジェクト指向プログラミング環境なくして実装工程は語れない状況にある。このような経緯から、実装に関する領域に限定すれば、オブジェクト指向パラダイムに関する定義は、多くの専門家や技術者の間で一致している。しかし、オブジェクト指向設計からオブジェクト指向分析へと抽象的な領域に移るに従い、オブジェクト指向パラダイムに関する定義は、不明確かつ正確性を欠く内容になり、合意を得ている意見や概念は少なくなる。だが、皮肉なことに「オブジェクトの識別が困難である」という点では多くの専門家や技術者の認識が一致している。

筆者らは「オブジェクトはビジネス・プロセスで生成される基本データの容器である」という方法論：DATARUN の導入により、この問題を解決することが出来た。しかし、筆者が社外ユーザのソフトウェア開発をコンサルティングした時に新たな問題に遭遇した。

本稿ではオブジェクト識別に関する問題と解決策について考察する。以下、2章ではオブジェクト識別の難しさと要因について論じる。3章では筆者らの導入した方法論でのオブジェクト識別方法を説明する。4章では新たに生じた問題点と筆者の考案した解決策を述べる。5章では解決策の有効性について USE-CASE との比較なども含め評価する。

## 2. オブジェクト識別の困難さ

筆者は「オブジェクトの識別が難しい要因は、オブジェクト指向パラダイムが本来持つ問題ではなく、オブジェクトの識別方法自体に誤りがあるため」と考えている。問題視するオブジェクトの識別方法とは大凡以下の通りである。

- 1) まず業務内容やユーザの要求をよく理解する
- 2) 次にその要求を満たすための機能をよく検討して定義しオブジェクトを識別する
- 3) オブジェクトに必要なデータをよく考えてオブジェクトの関係を定義しながら詳細化する

文中の「よく」という副詞は個人の経験や勘に依存の形容であり、この説明文は発見的手法による「機能」の識別を表現している。

「機能をソフトウェア開発のパラダイムとするこ

と」が機能中心パラダイムの最大の問題点であった。

「機能」は曖昧で認識に個人差があり且つ安定性に欠けるからである。この問題が、システムの大規模化や複雑化に伴い、一層深刻になったことがオブジェクト指向へのパラダイムシフトの要因と言って良い。

「オブジェクト」の識別が困難とされるのは、機能中心パラダイムと同様、「機能」の識別に注目しているからに他ならない。ところが、それを認識している専門家は殆どいないのではないか。オブジェクト指向パラダイムに多大な影響力を持つ J. Rumbaugh しかり。曰く「問題記述から候補となるオブジェクトクラスを並べ上げることから始めよう。最初から正しいものだけを選ぼうとせず、心の中に浮かぶすべてのクラスを書き下ろす」。彼の言葉は、発見的手法を追認しオブジェクト識別の困難さや属人性を容認する以外、何ら有益な示唆も与えていない。

## 3. DATARUN にみるオブジェクトの識別方法

DATARUN は Laval 大学の D. Pascot 教授の提唱する方法論である。筆者らは 1994 年以来 DATARUN を導入しソフトウェア開発を行なっている。導入の理由は、全ての局面において発見的手法と対局にある方法論であることと、筆者の件案であったオブジェクトの識別における明確な指標「オブジェクトは基本データの容器である」に注目したためである。

本稿は DATARUN の紹介を目的とするものではないが、オブジェクト識別に関する筆者のポジションを明確にするには、DATARUN に関する言及が不可欠であると考えられる。ここでは、本稿のテーマと関連するビジネス・プロセス・モデルと概念データ・モデルの二つのモデルについて説明する。表記法自体の説明は割愛する。

### 3.1 ビジネス・プロセス・モデル(図-1)

使用する表記法は構造化設計で用いられたデータフロー図 (Data Flow Diagram) とほぼ同じである。ビジネス・プロセス・モデルの目的は「ビジネス・プロセスの理解」、「基本データ発生源 (Primary Data Generator, 以下 PDG) の発見」、「システム化対象範囲の決定」そして「解決すべき問題点の発見」の四つである。

ビジネス・プロセス・モデリングにおける留意点は、現在の業務プロセスをそのまま記述するのではなく、本来あるべき姿としてのプロセス・モデルを記述することである。何故なら、本来あるべきプロセスを省略したり変更したりしている場合 (実際に多く見受けら

れる)には、現在の業務プロセスをそのまま記述しても、ビジネス・プロセスを正しく理解したことにならないからである。

ビジネス・プロセス・モデルで発見したいビジネス・プロセスとは、機能を詳細に説明するプロセスではなく、ビジネス遂行に不可欠な基本データ (Primary Data) を発生させるプロセスである。ビジネス・プロセスを引き起こすトリガでもある PDG は、(システムに) 記録されるべき必要最小限のデータである、基本データを作り出す。つまり、PDG によりビジネス・プロセスが引き起こされ、ビジネス・プロセスの中で基本データが生成される。

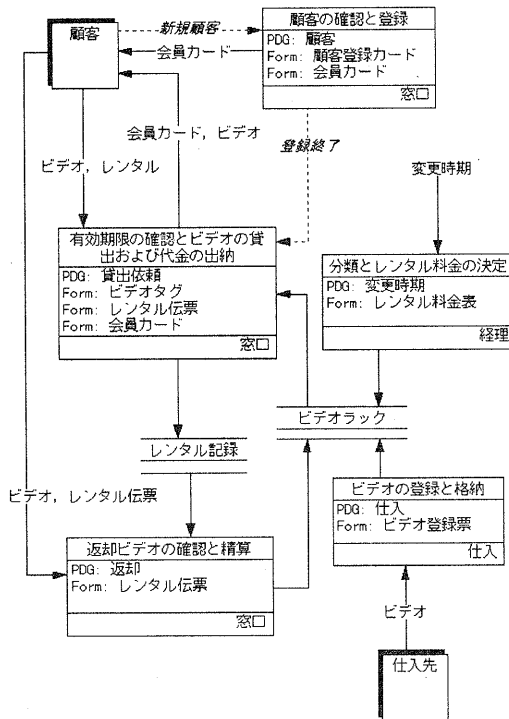


図-1 ビジネス・プロセス・モデルの例

ビジネス・プロセスには PDG (による基本データの生成) が必要であり、単なる報告書作成など PDG (による基本データの生成) のないプロセスは、ビジネス・プロセスではなく、ビジネス・プロセス・モデルには表現しない。インタフェース (スクロール・リストかグラフ表示か) や処理の仕掛け (バッチかオンデマンドか) でデータの発生の仕方が変わるわけではない。データが「どのように発生するか」が重要であり、「ど

のように使われるか」が重要なのではない。

### 3.2 概念データ・モデル (図-2)

概念データ・モデルの目的は、コンピュータが操作するデータを理解することであるが、インタフェースや処理の仕掛けあるいはデータストアを用意することではない。概念データ・モデルでは、コンピュータ処理を意識しない純粋なビジネス・ルールの意味を、エンティティ・リレーションシップ図 (Entity Relationship Diagram) の拡張表記法により表現する (DATARUN ではデータ・モデルの表記法自体はそれ程重要とは考えていない)。

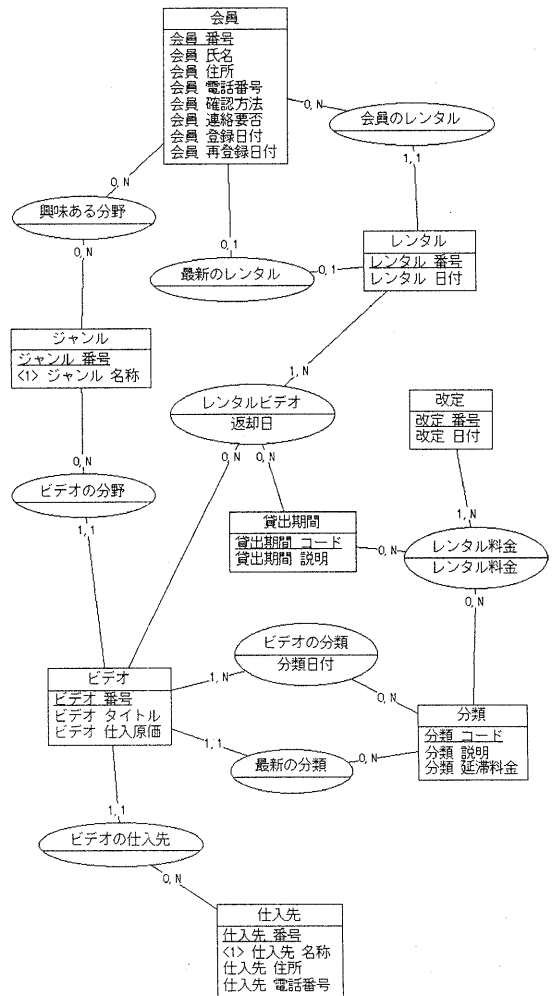


図-2 概念データ・モデルの例

概念データ・モデルの構築は、基本データの識別に始まり、基本データの容器としてのオブジェクトの決定、最後に基本データと基本データ相互の関連を表現する。オブジェクトの識別には単純且つ明確なルール：「オブジェクトは基本データの容器である」を適用する。オブジェクトは単独で存在することはなく、基本データ（属性）を格納するために存在し、基本データを操作するための機能（メソッド）を併せて持つ。故に、「基本データはオブジェクトの存在を決定する」と言えるが、その逆は成立しない。

オブジェクト識別のルールでは、PDG が同じ基本データは同一のオブジェクトに格納され、PDG が異なれば別のオブジェクトに格納される。基本データは PDG によって生成される。そのため「PDG が同じ」は「同じ時点で生成される」ことを意味し、「PDG が異なる」は「異なる時点で生成される」を意味するからである。

概念データ・モデルが正規化理論を満足させるモデルであることは言うまでもない。しかし、ビジネス・プロセス・モデルで発見された PDG が適切であり、オブジェクトの識別に問題がなければ、概念データ・モデルは自ずと正規形モデルになる。正規形モデルはビジネス・ルールと無関係には存在しえない。一般的な受注モデルでは、基本データ：注文日付、基本データ：顧客氏名および基本データ：製品名は、それぞれ異なるオブジェクトに格納されるであろう。しかし、ビジネス・ルール：一見顧客の注文生産を明確に表したモデルは異なる。基本データ：顧客氏名および基本データ：製品名は注文時に発生する。即ち、基本データ：顧客氏名および基本データ製品名の PDG は基本データ：注文日付と同じく注文であり、同じオブジェクト：注文に格納される。

以上のことから、概念データ・モデルの品質（モデルに表現される基本データやオブジェクトの正当性）はビジネス・プロセス・モデル（特に PDG）の品質に依存すると言える。

DATARUN のモデルには、この他に、情報システム・アーキテクチャ、システム・プロセス・モデル、共通仕様モデルおよびインタフェース仕様モデルがあるが、本稿のテーマである「オブジェクトの識別」とは直接関連がないので割愛する。

#### 4. オブジェクトの識別方法と新たな問題

オブジェクト指向パラダイムでは「オブジェクトを識別し次にその属性を定義する」という識別方法と

が一般的である。しかし、2章で述べたように、この識別方法を実践することは非常に難しい。

筆者らは「オブジェクトは基本データの容器である」という DATARUN のコンセプトを導入することにより、オブジェクト識別の困難さから開放され、データ・モデリングの生産性と精度の向上に成功した。しかし、筆者がソフトウェア開発のコンサルティングをするに至り、オブジェクトの識別に関して新たな問題点に直面した。訓練を受けた開発チームには基本データの発見やモデリングは容易な作業であったが、そうでない技術者では基本データ識別の困難さからモデリングの停滞を招くのである。

オブジェクトは基本データが決定し、基本データはビジネス・プロセスで生成される。故に「オブジェクト識別の成否はビジネス・プロセスの適切性に依存する」と言える。この視点に立てば、「オブジェクトの識別に失敗しているプロジェクトでは、ビジネス・プロセスの識別に問題がある」と結論づけられる。調査の結果、彼らの作成したビジネス・プロセスモデルは夥しい数のビジネス・プロセスによって構成されていることが判明した。ここで表現されているのはビジネス・プロセスではなくコンピュータ・プロセスである。

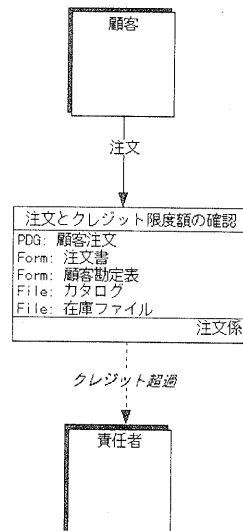


図-3 正しいビジネス・プロセス・モデル

ビジネス・プロセスとは、機能を詳細に説明するプロセスではなく、基本データを発生させるプロセスである。そのためデータに注目したモデルは定義されているプロセスの数が少ない（図-3）。基本データを発

生させている業務の活動や過程をプロセスと考えれば、その数が多くないことが容易に理解出来る。

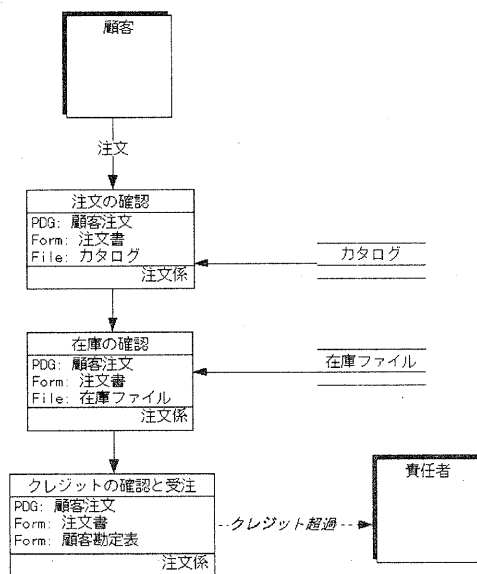


図-4 間違ったビジネス・プロセス・モデル

これに対して機能を説明するモデルはプロセスをコンピュータ処理と解釈して表現されている。このアプローチでは、コンピュータ処理一つひとつの詳細な機能により業務を表現しなければならず、プロセス数は多くなる(図-4)。技術者の多くは過去に経験したソフトウェア開発を通じて「ソフトウェアにおけるプロセスとはコンピュータ処理である」との刷込みを受けている。そのため、ビジネス・プロセスの概念を理解し識別する手段に習熟していないと、表現すべき基本データ発生モデリングではなく、コンピュータ処理機能のモデリングという過誤を犯す危険性が高い。

コンピュータ処理で基本データの生成の意味を説明することは出来ない。当然の帰結として基本データの容器としてのオブジェクトは識別不可能である。

## 5. 解決方法とその効果

筆者は「(習熟していない)技術者のためのビジネス・プロセス識別」のガイドラインを作成した。この章ではガイドラインの概要とその効果について述べる。

### 5.1 ビジネス・プロセス識別のガイドラインの概要

筆者の提案するガイドラインは次の四つからなる。

1) ビジネス・プロセスを表わす言葉として活動、作業、過程あるいは手続きを使用する

ビジネス・プロセスを表現する場合に、常に活動、作業、過程あるいは手続きなどのコンピュータ処理を連想しない言葉を用いることで、技術者にビジネス・プロセスがコンピュータ処理ではないことを強く認識させる。

2) ビジネス・プロセスの起動要因である PDG を 5W で説明する

ビジネス・プロセスは業務遂行に必要な過程である。識別したビジネス・プロセスの起動理由が 5W : 「What (何), Who (誰), Where (何処), When (何時) あるいは Why (何故) なのか」で記述可能であり、因果関係がビジネス・ルールとして認識できれば適切なモデルである。

3) ビジネス・プロセスの名称を PDG による具体的な過程として説明とする

ビジネス・プロセスは基本データ発生過程であり、ビジネス・プロセスは PDG によって引き起こされる。つまり、ビジネス・プロセスは PDG によって引き起こされる基本データ発生過程である。「PDG : A によって引き起こされる過程は B である」と表現できれば B はビジネス・プロセスである。

4) A4 サイズ用紙 1 ページにモデリング

4 章で説明したようにデータを発生させるプロセスは決して多くはない。「ビジネス・プロセス・モデルはプロセス数 12 個を限度とし A4 サイズ用紙 1 ページに表現する」制約を加えることで表現できるプロセス数は少なくなる。コンピュータ処理による機能の詳細な表現を防止できる。

## 5.2 ガイドラインの有効性の検証

ここでは、筆者がコンサルティングした、M社の受発注システムのモデリングでのガイドライン適用を説明し、有効性について検証し考察する。モデリングは受発注に関連する事象に限定した。

コンピュータ処理に注目したプロセス・モデル(図-5)は、技術者にとって馴染みがあり理解し易いように思えるが、プロセスとプロセスの起動要因の関係が表現されていない。

次に概念データ・モデルを図-6 に示す。先のプロセ

ス・モデルでは、基本データと PDG の関係が表現されていない。そのため、オブジェクトの識別は困難であり、データ・モデリングは（技術者の経験に依存した）発見的手法にならざるを得ない。図-6 のデータ・モデルには「発注の時入庫すべき倉庫，納入先の顧客および納入の業者は決まる場合もそうでない場合もある」というルールを表わされている。しかし、このルールに意味はなくビジネス・ルールと呼ぶには曖昧すぎる。このような場合、ビジネス・ルールはプログラムの手続きで表現することになる。正当性や精度はデータ・モデルの品質ではなくプログラマのスキルに依存する。

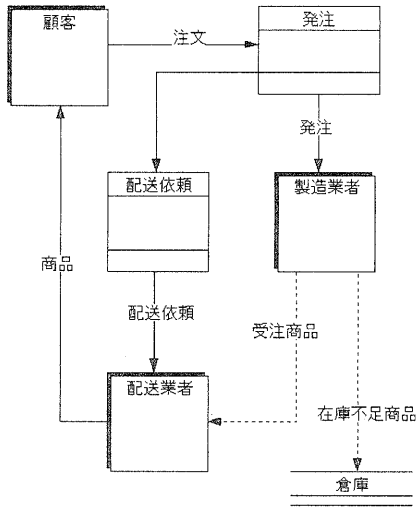


図-5 コンピュータ処理に注目したプロセス・モデル

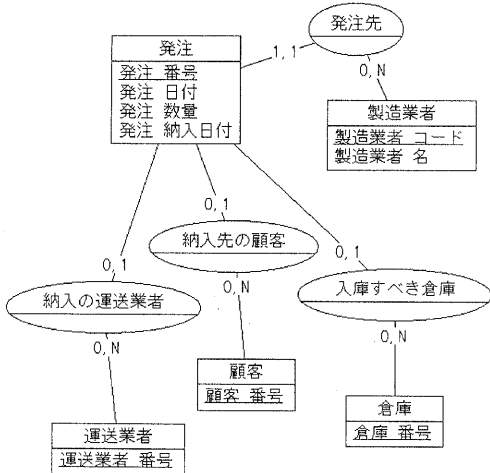


図-6 ビジネス・ルールの曖昧な概念データ・モデル

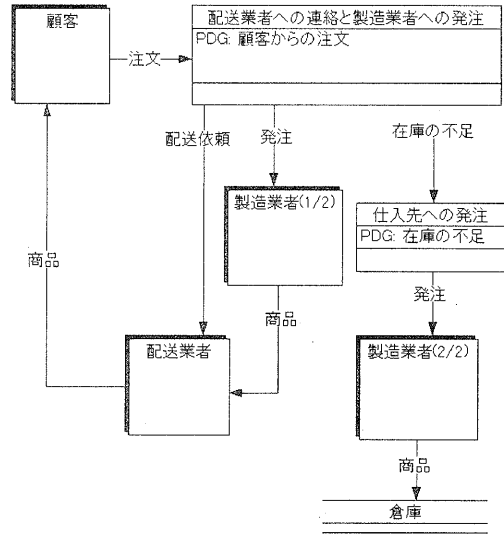


図-7 ガイドラインによるビジネス・プロセス・モデル

ガイドラインに従って作成し直したビジネス・プロセス・モデルを図-7 に示す。その時の具体的なモデリング手順は次の通りである。

- 1) 受発注業務における活動，作業，過程，手続きには何があるかをインタビューする
- 2) インタビューを基にプロセスを定義し起動要因である PDG を 5W のどれかで表現する
  - ・プロセス：製造業者への発注が見つかる。しかし、その起動要因：5W は顧客からの注文と在庫の不足である。
- 3) PDG で引起こされるプロセスを説明する
  - ・プロセス：製造業者への発注を PDG の違いにより二つに分ける。
  - ・PDG：顧客からの注文によって引起こされるのはプロセス：配送業者への連絡と製造業者への発注である（配送業者は製造業者の商品を顧客に配送する）。
  - ・PDG：在庫の不足によって引起こされるのはプロセス：製造業者への発注である。

これでプロセスとその起動要因である PDG の関係がビジネス・プロセス・モデルに表わせた。

次に、このビジネス・プロセス・モデルがオブジェクト識別を容易し適切なデータ・モデリングを支援することを示す。主なオブジェクトの識別とモデリングについて順を追って述べる（図-8）。

- 1) 基本データ：発注番号，発注日付，発注数量の PDG は製造業者への発注（顧客からの受注，在庫の不足に関係なく）であり格納オブジェクトはエンティティ：発注とする
- 2) 基本データ：入庫予定日付の PDG は在庫不足のための発注であり格納オブジェクトはエンティティ：入庫とする
- 3) 基本データ：納入予定日付の PDG は受注のための発注であり格納オブジェクトはエンティティ：納入とする
- 4) 基本データ：入庫すべき倉庫の PDG は在庫不足のための発注でありリレーションシップ：入庫すべき倉庫で表現する
- 5) 基本データ：納入先の顧客の PDG は顧客からの受注でありリレーションシップ：納入先の顧客で表現する
- 6) 基本データ：納入の運送業者の PDG は顧客からの受注でありリレーションシップ：納入の運送業者で表現する

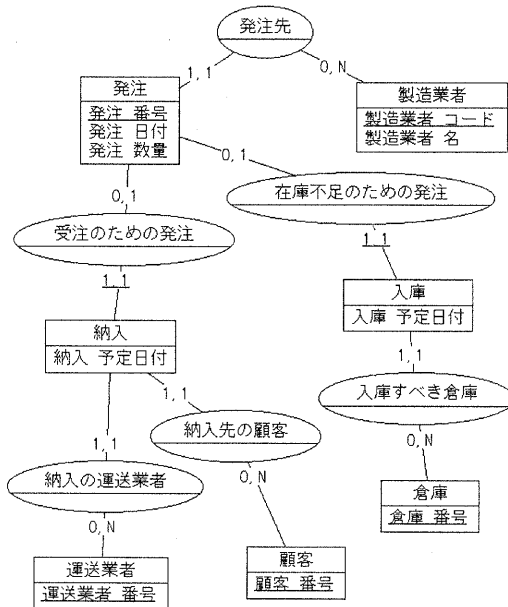


図-8 ビジネス・ルールの明確な概念データ・モデル

このように作成された概念データ・モデルからは以下のビジネス・ルールが明確に読み取れる。

- 1) 発注には発生要因の違いから受注による発注と在庫不足による発注の二つがある
- 2) 発注番号，発注日付，発注数量および発注

先の製造業者は全ての発注に必ず存在する

- 3) 受注による発注には納入の配送業者と納入先顧客が必ず存在する
- 4) 在庫不足による発注には入庫すべき倉庫が必ず存在する
- 5) 製造業者，倉庫，顧客および運送業者は発注の有無に関係なく存在する

データ・モデルが適正であれば，それを構成するオブジェクトは適正である。故に，適正なオブジェクトを識別できたガイドラインは有効である。

### 5.3 use-case によるアプローチとの比較

ここでは，I. Jacobson の提唱する use-case によるアプローチ (A Use-Case Driven Approach, 以下 USE-CASE) の事例をガイドラインによりモデリングし，ガイドラインの有効性を USE-CASE との比較から考察する。尚，事例の use-case モデルは，ここに提示しないので巻末に示す文献を参照のこと。

#### 5.3.1 事例：ACME 倉庫管理会社の業務概要

ACME は全国にある倉庫を顧客に提供する事業を取り扱っている。顧客は販売する商品の保管場所を保有しない会社である。ACME では事業の拡張を計画しており，そのための情報システムを必要としている。

対象業務には品物の倉庫内での移動と倉庫間での移動の両方が含まれており，これらは全て顧客からの指令に基づいて行われる。保管する品物はあらゆる種類のものが対象となるが，このことは特定の品物を区別することが重要になることを意味する。例えば，(工業製品と食品のように) 互いに接触してはならないような組み合わせが存在する。

#### 5.3.2 DATARUN+ガイドラインによるモデリング

use-case モデルには「倉庫主任の承認なしで保管品が引出される」など問題となると思われる個所や不明瞭な記述が幾つか存在したが，基本的な変更はしなかった。また，顧客登録などは対象から外した。

業務概要からのビジネス・プロセス・モデル (図-9) に続いて，概念データ・モデル (図-10) を以下の要領でモデリングする。

- 1) 基本データ：保管番号，保管日付の PDG は保管依頼であり格納オブジェクトはエンティティ：保管とする
- 2) 基本データ：保管明細番号，保管明細品名，保管明細数量，保管明細期限の PDG は保管依

頼であり格納オブジェクトはエンティティ：  
保管明細とする（第1正規形の制約）

- 3) 基本データ：配送番号，配送日付の PDG は配送であり格納オブジェクトはエンティティ：配送とする
- 4) 基本データ：配送先倉庫，配送保管明細の PDG は配送であり格納オブジェクトはリレーションシップ：倉庫間の配送で表現する
- 5) 基本データ：引出し番号，引出し日付の PDG は引出しであり格納オブジェクトはエンティティ：引出しとする
- 6) 基本データ：引出し明細，引出し明細数量の PDG は引出しであり格納オブジェクトはリレーションシップ：引出し明細で表現する（第1正規形の制約）
- 7) 基本データ：禁止される配置の PDG は配置ルールであり格納オブジェクトはリレーションシップ：禁止される配置とする

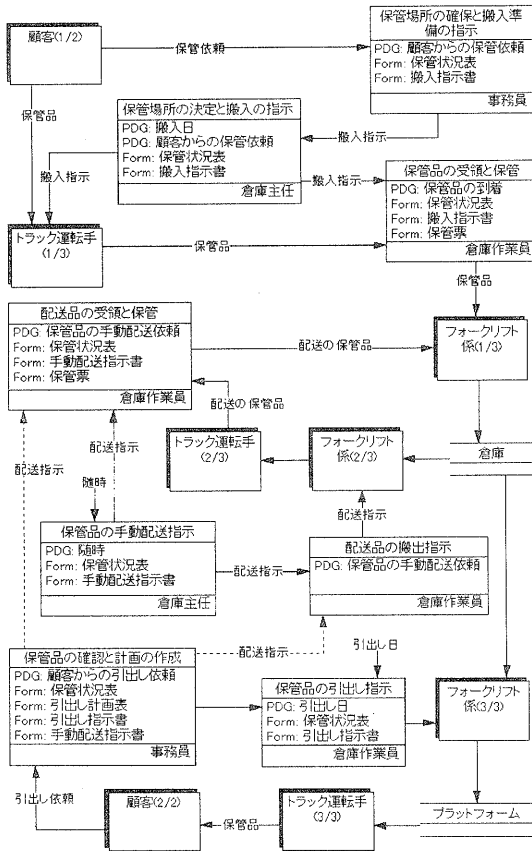


図-9 ACME のビジネス・プロセス・モデル

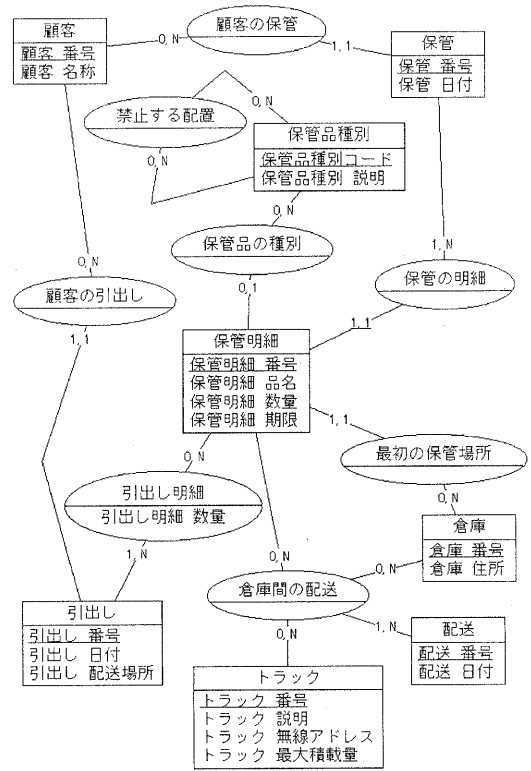


図-10 ACME の概念データ・モデル

### 5.3.3 比較と考察

筆者がモデリングに要した時間は、業務内容の理解も含めビジネス・プロセス・モデルに約2時間、概念データ・モデルに約30分であり、ガイドライン適用の有効性は確認できたと考える。

ここでは、本稿のテーマであるオブジェクトの識別性について比較し考察する。論点を明確にするために、オブジェクトの識別方法（「オブジェクト識別」）、システムに必要とされる機能（即ちプロセス）である use-case の定義（「use-case 定義」）および use-case の決定方法（「use-case 決定」）について、USE-CASE と筆者との違いを表-1に示す。

表-1 オブジェクト識別方法の違い

	USE-CASE	DATARUN+ガイドライン
オブジェクト識別	use-case	基本データ
use-case 定義	全プロセス	データ発生プロセス
use-case 決定	ユーザ要求	ビジネスルール



両者とも「オブジェクト識別に use-case が必要である」点ではほぼ一致しているが、筆者が use-case はデータ発生プロセスと明確なのに対し、USE-CASE は全てのプロセスと曖昧である。当然「use-case の決定方法」についても、USE-CASE は「ユーザ要求（インタフェースなどデータの使われ方）」とならざるを得ないが、筆者は「ビジネス・ルール（データの発生の仕方）」と考える。両者の違いを要約すれば、USE-CASE は「ユーザの要求がシステムの機能を決定しオブジェクトを識別する」と考え、筆者は「ビジネス・ルールが基本データを決定しオブジェクトを識別する」と言う立場を採る。

しかし、USE-CASE の方に無理があることは ACME 社の事例に置き換えれば容易に理解できよう。ユーザからの「プロセス：保管場所照会の実装」要求の有無に関係なく顧客より預けられるオブジェクト：保管品は存在する。また、プロセス：保管場所照会の検索仕様の変更でオブジェクト：倉庫間の配送の方法が変わるわけでもない。顧客から預かったオブジェクト：保管品を倉庫主任はビジネス・プロセス：倉庫間の配送の一環として保管場所照会し、オブジェクト：倉庫間の転送はビジネス・プロセス：倉庫間の転送で発生した基本データを格納する。この一連の use-case は、ユーザ要求により決定されるのではなく、ビジネス・ルールにより決定されることは自明であろう。

use-case モデルに表わされた情報の多くは、ソフトウェア開発に不可欠ではあるが、データ・モデリングには必要ない。ユーザ要求の有無に拘らずウィンドウ：保管依頼やウィンドウ：倉庫間の配送は必要だし、データ・モデルが適切であれば、ユーザの要求する倉庫間の転送に関する照会ウィンドウは実装できる。業務に必要なインタフェースや処理の仕掛けは、ユーザ要求ではなく、ビジネス・ルールを表わしたデータ・モデルから導き出せる。データ・モデリング時点でのインタフェースや処理方法への言及は論点を拡散させオブジェクト識別を困難にする危険性すら含んでいる。

USE-CASE によるオブジェクト識別やモデリングの工数が少なくないことはモデルのページ数や I. Jacobson の記述から窺い知ることが出来る。果してこれほどの工数がオブジェクトの識別やモデリングに必要な疑問が残る。また USE-CASE のソフトウェア開発への適用にも大きな障害となるう。

USE-CASE の幾つかの問題点について述べたが、最大の問題は「ユーザの要求する機能がオブジェクトを決定する」点であろう。「ユーザ仕様が凍結出来ない」、

「仕様変更が工数増大を招く」あるいは「簡単な機能変更がデータベースの大幅変更を引起す」は現実のソフトウェア開発において常に語られる問題である。ソフトウェア工学の第一人者とされる I. Jacobson が、何故このようなプリミティブな問題を看過しているか、筆者は理解に苦しむ。

## 6. おわりに

DATARUN の提唱者である D. Pascot は「データ・モデリングが簡単である」の意味で「自転車に乗るように」と形容することがある。自転車は誰でも乗れるからであろうが、筆者は「自転車に乗るにも訓練による習熟が必要である」と考えている。自転車に例を取れば筆者のガイドラインは「自転車の乗り方」に相当するだろう。しかし、「自転車の乗り方」を読むだけでは自転車に乗れないように、ガイドラインだけでモデリングは出来ない。モデリングにも習熟が必要なのである。筆者のガイドラインは、指標としては有効であるが、習熟に代替できる存在ではない。

筆者は、系統的な訓練により習熟度を高め、ガイドラインを指標にすることで、「自転車に乗るように」（簡単に）データ・モデリングが出来ると信ずる。

尚、3章の二つモデル図の事例は、日本工業大学の太木幹雄助教の厚意により提供戴いたが、モデリングに関する過誤は全て筆者の責任である。

最後になるが、不肖の徒である筆者を常に暖かく指導戴いた D. Pascot 教授、ビジネス・プロセスに関する有意義な助言を戴いた木ノ下勝郎氏に深甚なる謝意を表するものである。

## 参考文献

- [1] D. Pascot; DATARUN Method; CSAR, 1995
- [2] D. Pascot; IS Engineering with DATARUN; CSAR, 1995
- [3] D. Pascot; Conceptual Data Modeling; CSAR, 1995
- [4] D. Pascot; C/S データベースシステム入門; pp. 123-208 日経 BP, 1996
- [5] J. Rumbaugh 他; オブジェクト指向方法論 OMT; pp. 4-7, 167-185; プリンティスホールトッパン, 1992
- [6] I. Jacobson; 他 オブジェクト指向ソフトウェア工学 OOSE; pp. 69-77, 111-120, 315-360; トッパン, 1995
- [7] 加藤貞行; 日経ソフトウェア「良いプログラムは良い設計から」; 日経 BP 1998