オブジェクト指向によるモデリング――例題 2

川 南 理 恵

1. はじめに

与えられた問題文からシステム化の対象領域を切り出し，それをオブジェクト指向によってモデル化すると言うのが与えられた例題であった．

本例題のモデリングには BOAD 法（詳細は本特集号解説編の「ビジネス・アプリケーション向けオブジェクト指向モデリング技法：BOAD 法」参照）を使用した．BOAD 法では，問題領域に関する分析モデルとアプリケーション・システムのための設計モデルを作成する．設計モデルの作成にあったては，アプリケーション・アーキテクチャとして BOAD 法の参照モデルを使用している．

2. 分　　析

2.1 問題領域の設定

問題文は，「通販ビジネスのプロセスであるカタログ送付→注文受け付け→商品引き当て→配送→決済」のうち，「このシステムでは，商品引き当て処理，配送処理や決済処理は含まない」と指定してあるので，まず，これらの処理をシステム化の対象から外した．また，カタログ送付処理に関しては，「名簿の会員数が 200 万人になるまで続けられ」，「獲得した顧客は，林通販の宣伝部の次年度予算に反映させる」とあるが，これらの処理は顧客管理システムとして本システムとは独立したシステムとした方が良いとの判断で，これもまたモデル化の対象外とした．

「出来るだけ簡単に」とのことから，システムのメインとなるユースケースに着目し，システムのシナリオを絞ることから始めた．

問題文は，「顧客名簿を充実するために，カタログに新規顧客紹介ページを設けた」ことを中心に，システムの説明をしている．また，「紹介」のインセンティブとして「クーポン券」を導入している．そこでまず，ボトムアップ的に「紹介」ユースケースと，顧客（既存顧客，新規顧客）アクターを導き出した．この「紹介」が例題 2 のメインユースケースである．さらに「紹介」から「クーポン券」までを埋める，「注文」と「クーポン券発行」という二つのユースケースを追加した．これによって，既存顧客が新規顧客を紹介し，新規顧客が注文をした結果，クーポン券が発行されるという，一連の流れ，シナリオが見えてきた．これらは，注文受け付け処理に該当する機能である．

システムのアクタとユースケースの関係を図示したのが図 1 のユースケース図である．

2.2 クラスの識別

各ユースケース毎に，問題文の中から，問題領域を構成しているクラスを抽出した．

ユースケース「紹介」というのは，「新規顧客」を紹介してくれた「既存顧客」にインセンティブを与えることであり，この関係をシステムで管理しておく必要がある

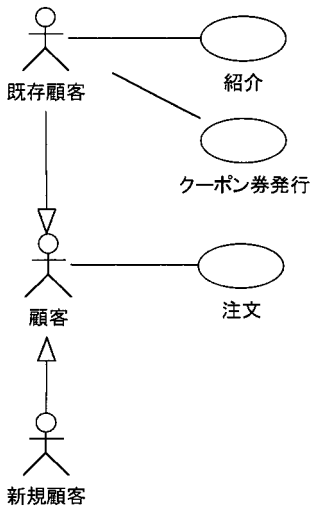

図 1　ユースケース図（1）

ので，アクターとは別に，それぞれクラスとして識別した．すなわち，ある「既存顧客」と「新規顧客」の間には「紹介」という関係があって，その関係に基づいて「クーポン券発行」が行われると考えた．

既存顧客　→　紹介　→　クーポン券発行

これならば，既存顧客と新規顧客の 1 対 1 の関係上で，一定期間の後クーポン券を発行するかどうか決定すればよい．ただし，実装では，定期的に新規顧客の購買情報を調べることになるのかもしれない．「新規顧客」ごとにタイマーをセットして，タイムアウトになった「新規顧客」から「既存顧客」を参照することになる．

もう一つの方法は，紹介とは無関係にバッチが走るという方式で，この場合は，紹介があるなしに関わらず，クーポン券を発行できるかどうか調べなくてはならない．

既存顧客　→　紹介

既存顧客　←　クーポン券発行

ここで，「インセンティブ」の拡張性を考えると，「紹介」のみでなく，既存顧客自身の購買額にも「インセンティブ」が与えられる等の拡張性を考慮して，ビジネスを分けることにした．そして，クーポン券の発行処理を行うためのタイマーをイベント・アクターとして追加した．イベント・アクターは，タイマーなどのイベントによって発生するプロセスをモデル化するためのものである．これにより，ユースケース「クーポン券発行」が実際行っていることは，「顧客」の新旧のつけかえと「クーポン券」の発行の二つとなる．図 2 にイベント・アクターを追加したユースケース図を示す．

ユースケース「注文」とは，購買申し込みのあった人を「顧客」とし，以後定期的に顧客に「カタログ」を送り，顧客はカタログをみて注文するという一連のビジネス活動である．ここで「顧客」とは，上記「既存顧客」と「新規顧客」のことである．

また，顧客が申し込みをできるのはカタログに記載されている商品に限定されるので，これを「カタログ商品」とした．この会社が扱っているすべての商品を本システ

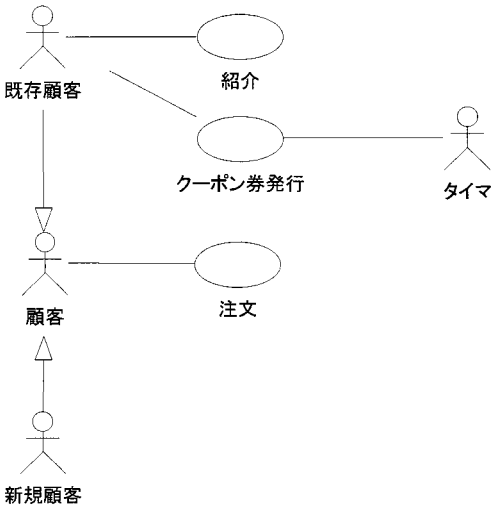

図 2　ユースケース図（2）

ムで管理することはしない．

問題文の中には，どの顧客がどの商品を購入したかの管理方法は記述されていないが，システムではこの関係を何らかの形で管理しておく必要がある．そのためのクラスとして「購買」を設定した．「購買」で管理している情報は，実際には何らかの伝票で管理されている．本モデリングでは，伝票の形式を，一枚の伝票には，その顧客が一回の注文で指示したすべての商品が記載されているものと想定した．すなわち，購買した複数の商品は，その伝票の明細として記入されている．この明細は，それぞれの「購買」毎に変化するので，「購買商品」として独立したクラスとした．

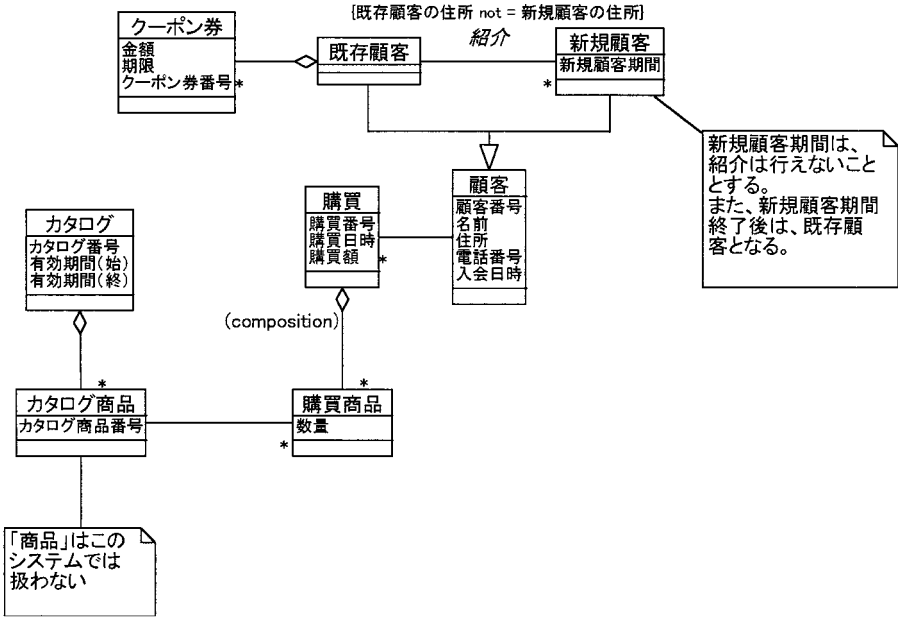

図 3　問題領域のクラス図

また，家族関係については，紹介された新規顧客が，既存顧客の家族かそうでないかを調べる必要がある．そのため「顧客」同士の間に再帰的な関連を設定すると，ある顧客は他の顧客の家族である，ということを表していることになる．問題は「既存顧客と同じ住所の顧客は，その顧客の新規顧客にはならない」と言うことなので，関連「紹介」に｛既存顧客の住所 not＝　新規顧客の住所｝という制約条件を付けた．

問題領域の中から識別された以上のクラスをクラス図で表現したものが図 3 の概念モデルである．

2.3　シナリオの作成

各ユースケース毎に，シナリオを記述した．

各シナリオは，以下の通りである．

1）　シナリオ：紹介

1．既存顧客は，新規顧客情報（名前，住所，電話番号，同意署名）をカタログの紹介ページに記入する．

新規顧客が既存顧客の場合や既存顧客の家族である場合は（1.1）へとぶ．

2．システムは新規顧客の登録を行う．

1.1　システムは，顧客にクーポン券を発行できない旨のレターを送る．

2）　シナリオ：クーポン券発行

1．イベントアクタは，一定期間毎に，登録されているすべての新規顧客の新規顧客期間を調べる．

2．新規顧客期間が終了した新規顧客に対して，その期間内の購買金額を調べる．

3．購買金額を調べ終わった新規顧客は，新規顧客登録から削除し，既存顧客登録を行う．

購買がなかった場合は（3.1）へとぶ．

4．紹介してくれた既存顧客に，新規顧客の購買金額に応じてクーポン券を発行する．

3.1　システムは，顧客にクーポン券を発行できない旨のレターを送る．

3）シナリオ：注文

1．顧客は，顧客情報（顧客番号があれば顧客番号，なければ名前，住所，電話番号）と，必要なだけのカタログ番号，商品番号，数量を記入する．

顧客番号やカタログ番号，商品番号等が間違っている場合は（1.1）へとぶ．

2．システムは，該当商品すべての在庫を調べる．

商品の在庫がない場合は（2.1）へとぶ．

3．在庫があれば注文を受け付ける．

1.1　システムは，何が間違っているのかを顧客へ伝え，（1）へ戻る．

2.1　システムは，

・その商品をキャンセルするか，

・キャンセルして替わりのものを注文するか，

・取り寄せるか

を顧客に選択させ，2）へ戻る．

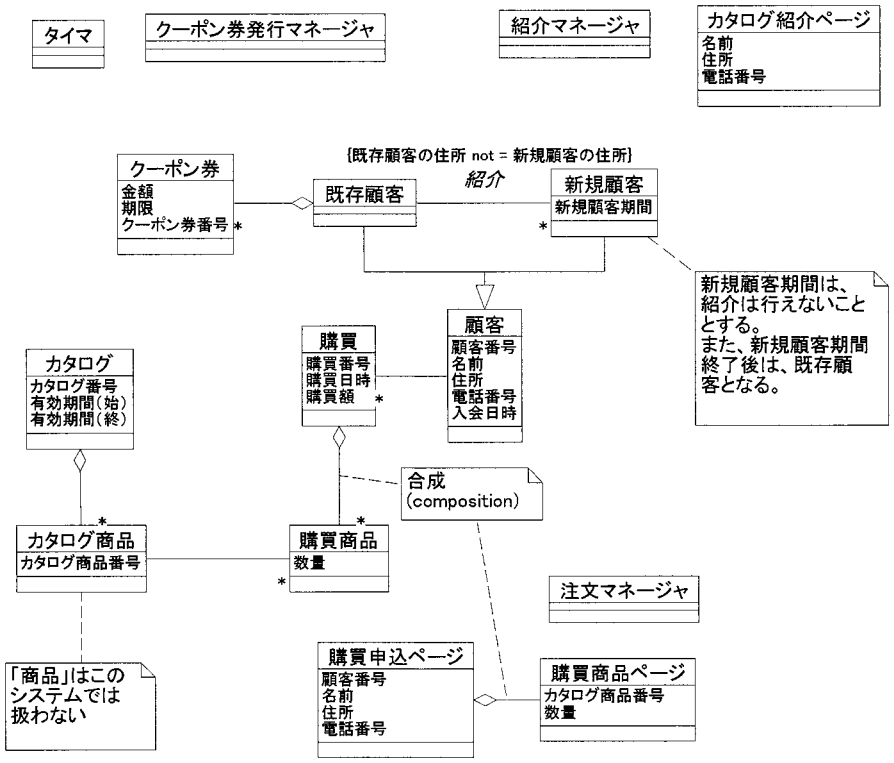

図 4　システムのクラス図

3. 設　　計

3.1 アプリケーション・オブジェクトの設定

相互作用モデルの作成に先立って，アプリケーション・オブジェクトであるインタフェース・オブジェクトとコントロール・オブジェクトを設定した．

インタフェース・オブジェクトは，ウィンドウイメージで作成した．

・カタログ紹介ページ

・購買商品ページ

・購買申し込みページ

また，各ユースケース毎に下記のコントロール・オブジェクトを設定した．

・紹介マネージャ

・クーポン券発行マネージャ

・注文マネージャ

これらのオブジェクトとドメイン・オブジェクトの関係は，相互作用であって，関連ではない．したがって，インタフェース・オブジェクトやコントロール・オブジェクトは，それ自身独立したパッケージとして表現することもできるが，今回の事例は，あまり複雑ではないので，ドメイン・オブジェクトの横に，どのクラスとも関連を持たない孤立したクラスとして置いておくことにした．これを，図 4 に示した．

3.2 相互作用モデルの作成

次に，このクラス図とシナリオによって，各オブジェクト間の相互作用をシーケン

ス図によって表記する．

1）　ユースケース「紹介」の相互作用（図 5）

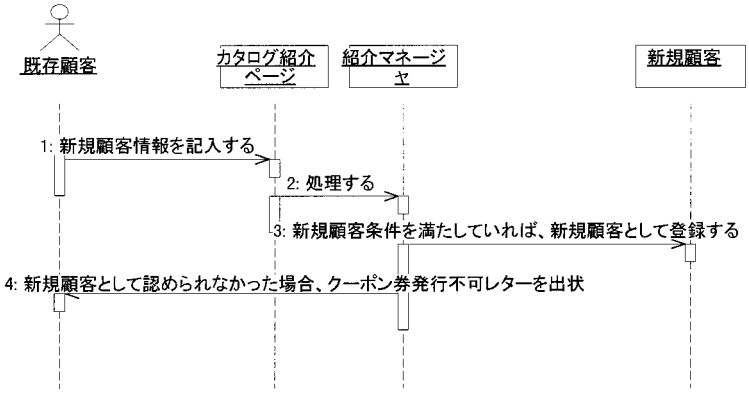

図 5　ユースケース「紹介」のシーケンス図

問題文では明確に記述されていないが，新規顧客が既存顧客に推移した時点で，その新規顧客と紹介関係にあった既存顧客は，紹介既存顧客からただの既存顧客に戻ってしまうことになると考えられる．これをシーケンス図で表現しようすると，インスタンス「新規顧客」を導入する必要がある．実装を考えると，クラスを分ける必要はないと思うので，既存顧客一つにしたいのだが，それだと「既存顧客」には紹介を行った顧客と未だ紹介をしていない顧客が居るということが明示的に示されないため，混乱する．紹介⟷既存の推移は，結局は関連で表現できる域を出ないと思い，推移は表現しなかった．

2）　ユースケース「クーポン券発行」の相互作用（図 6）

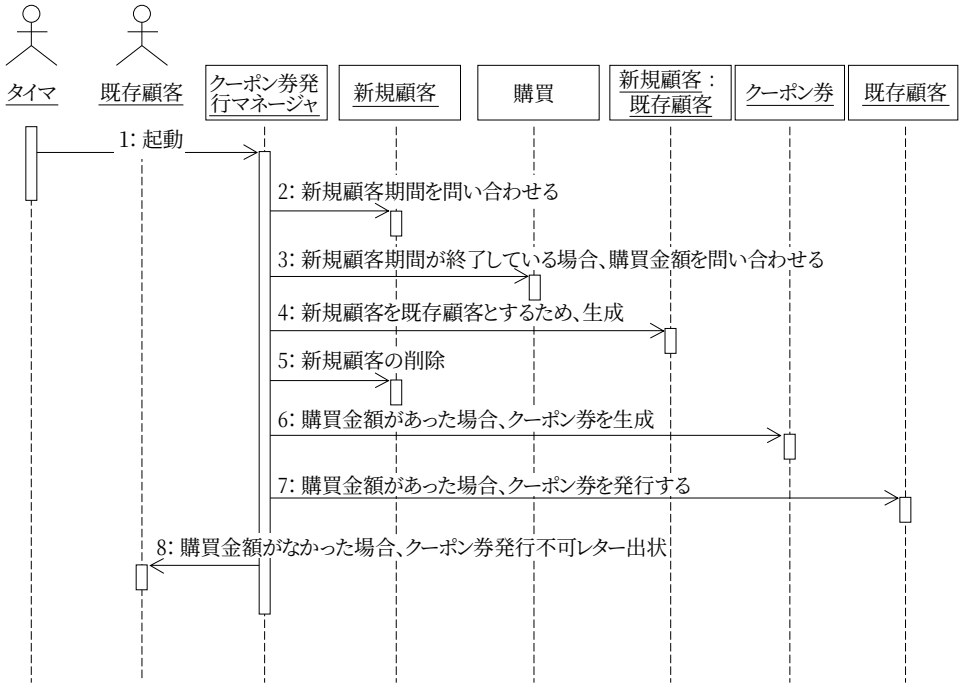

図 6　ユースケース「クーポン券発行」のシーケンス図

「クーポン券発行」のシーケンスでは，イベント・アクターも普通のアクターと同じように，入力刺激はコントロール・オブジェクトに渡されるが，システムの内部アクターなのでインタフェース・オブジェクトは経由しない．また，「既存顧客」へのクーポン券発行不可通知も，郵便などを使って行われるのでインタフェース・オブジェクトは経由しないものとした．

3） ユースケース「注文」の相互作用（図 7）

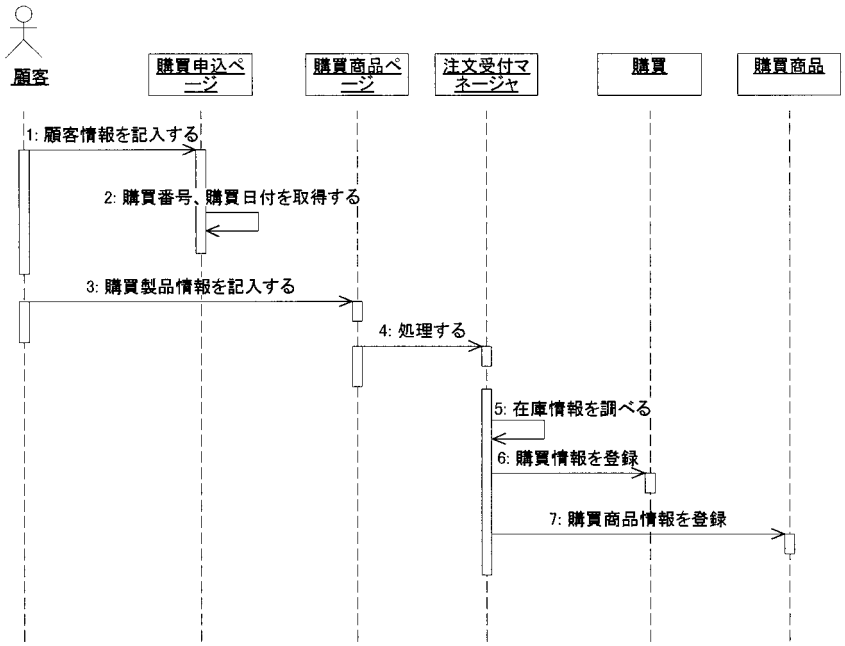

図 7　ユースケース「注文」のシーケンス図

4. おわりに

BOAD 法の設計モデルの作成では，本来，これらのシーケンス図を元に各クラスの操作を定義して行くことになるが，今回は，システムの実装において，どれだけの情報が必要かという検証実験の意味も含めて論理モデルの作成は行なわなかった．その結果，インスタンスの検索や状態の推移などに関するアルゴリズムの設計も含めて実装段階での設計に任せることとなった．オブジェクト指向システム開発の実装段階で必要とされる情報の種類についての何らかの指針が得られれば幸いである．

執筆者紹介　川　南　理　恵（Rie Kawaminami）
1967 年生．1990 年日本大学文理学部応用数学科卒業．同年日本ユニシス(株)入社．社内システムエンジニアの教育・研修・カリキュラム開発を担当し，今日に至る．現在総合教育部 IT 教育推進室に所属．