

独立エンティティと関連クラスによるドメイン・モデル

Domain models by independent-entities and association classes

河合 昭男[†]

Akio Kawai[†]

[†](有)オブジェクトデザイン研究所

[†] Object Design Laboratory, Inc.

要旨

オブジェクト指向の特徴は何でもオブジェクトとできることであるが、それは同時にモデルの作成および理解を困難にしている問題点でもある。オブジェクトをモノとコトに分けるとモデルの作成および理解はより容易となる。筆者の提唱する手法は、それ自体で存在し得る「独立エンティティ」をまず発見し、次にイベントにより発生する独立エンティティ間の「関連クラス」を発見する手法である。この手法により理解し易いドメイン・モデルが自然に作成できることを例を挙げて示す。

1. はじめに

オブジェクト指向によるドメイン・モデル作成方法は、対象領域から名詞に注目してクラス抽出を行いそれらの意味的関係を発見して関連で表すのが伝統的方法である。この方法は必ずしも分かり易い方法とは言えず、特に初心者にとってオブジェクト指向モデリングの阻害要因のひとつとなっている。

ドメイン・モデルの対象となるオブジェクトにはモノとコトがある。例えば「顧客」や「商品」などそれ自体で存在可能ないわゆるものと、例えば「注文する」などのイベントの結果として生成される「注文」のように事象を表すものがある。これらを同列にすべてオブジェクトという統一概念で表現できること、言い換えれば「何でもオブジェクト」がオブジェクト指向の特徴であり同時にオブジェクト指向を分かり難くしている問題点でもある。

本稿ではこの事象を表すオブジェクトをイベントにより発生する関連クラス型オブジェクトとして捉え、ドメインモデルの対象となるオブジェクトを独立型と事象型に分けることによるドメインモデルの作成と理解の容易性改善を例を挙げて述べる。

2. 独立エンティティと事象エンティティ

ここではそれ自体で存在し得るオブジェクトをステレオタイプ《independent entity》で表し、独立エンティティ型のオブジェクトと呼ぶことにする。二つ以上のオブジェクトの存在を前提として、それらのリンクのプロパティとして存在する関連クラス型オブジェクトをステレオタイプ《event entity》で表し、事象エンティティと呼ぶ。事象エンティティは必ずしも2項関連クラスとは限らず、一般的にはN項関連クラスがあり得る（後述）。

ドメインモデルにはこれら以外にコンポジット関係でオブジェクトの属性を表す従属的オブジェクトも存在し得るが本稿では特に触れない。

ここではそれぞれを図1のようなアイコンで表すことにする。独立エンティティのアイコンはヤコブソンのエンティティと同じアイコンを用いているが、本稿ではこのアイコンは独立型エンティティとしている。

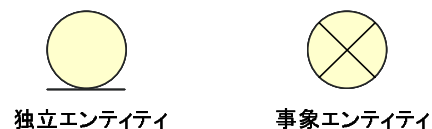


図1 独立エンティティと事象エンティティ

3. 関連とイベント

イベントはなんらかのオブジェクトの状態変化をもたらす。オブジェクトの状態は属性値または他オブジェクトとのリンクの有無で表すことができる。リンクはその有無のみならずそれ自体のプロパティをもつ場合がある。そのプロパティは関連クラス型オブジェクトとして表すことができる。従ってこの

ような場合の関連クラスは事象を表している。

4. 注文の例

よく知られている注文-明細のモデル（図2）は次の手順でモデリングすることができる。

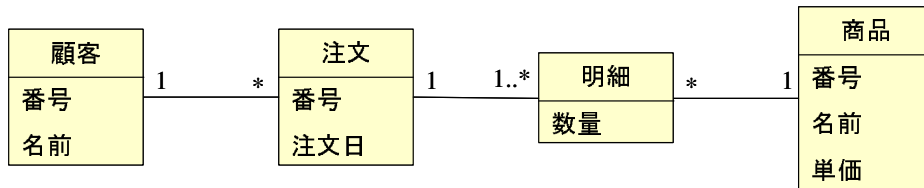


図2 注文-明細モデル

独立エンティティは顧客と商品の二つで、注文と明細は、「注文する」というイベントにより生成される関連クラス型の事象エンティティとしてモデリングできる。図3にそのモデリングの手順を示す。

step 1: 独立エンティティ顧客と商品を作成し、その間に「注文する」という関連を結ぶ。

step 2: 注文の属性（番号、注文日 etc.）を設定するために関連クラス「注文」を作成する。

step 3: 関連クラス「注文」を通常の2つの2項関連で表す。

step 4: 注文と商品の間には数量を保持するために関連クラス「明細」を作成する。

step 5: 関連クラス「明細」を通常の2つの2項関連で表すと図2のクラス図となる。

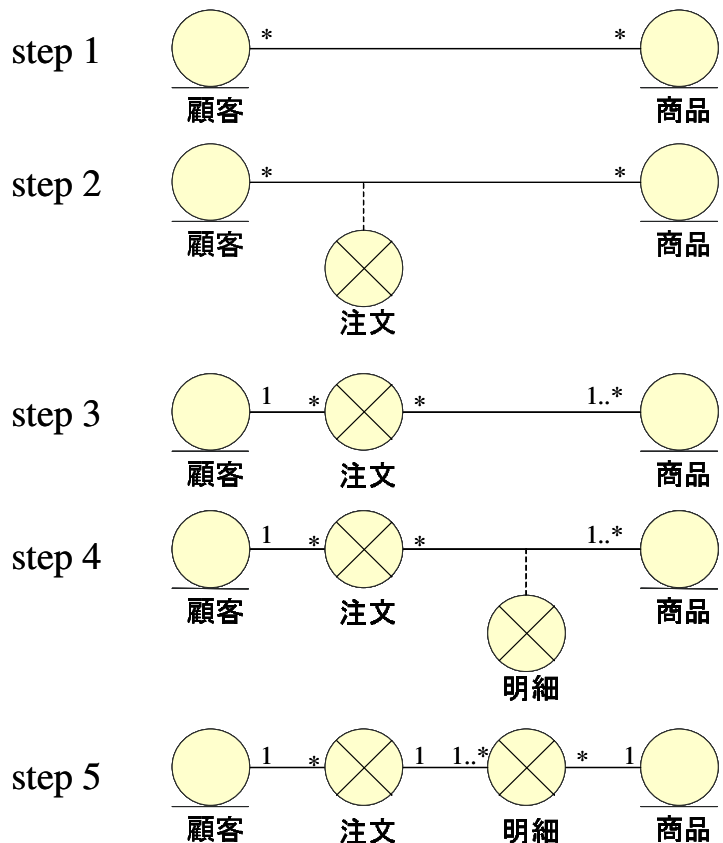


図3 注文-明細モデル作成手順

5. 状態とイベント

次に受注から納品・請求、回収までの業務フローと注文オブジェクトの状態遷移およびオブジェクト図を見てみよう。オブジェクトの状態は属性の値または他オブジェクトとのリンクの有無で表現できる。ここでは説明を簡単にするため明細は省略する。

① 「受注」の状態

「注文する」というイベントが発生すると、ある顧客オブジェクトとある商品オブジェクトの間にリンクができ、そのリンク属性となる「受注」オブジェクトが生成される。

② 「在庫引き当て」の状態

「在庫引き当てる」というイベントが発生すると、その受注オブジェクトとある在庫品目オブジェクトの間にリンクができる。

③ 「納品」の状態

「納品する」というイベントが発生すると、納品と同時に請求をおこなうために顧客オブジェクトと注文オブジェクトの間に請求するというリンクができ、そのリンク属性となる請求オブジェクトが生成される。

④「回収」の状態

「回収」というイベントが発生すると、例えば請求オブジェクトの回収日という属性に値が設定される。

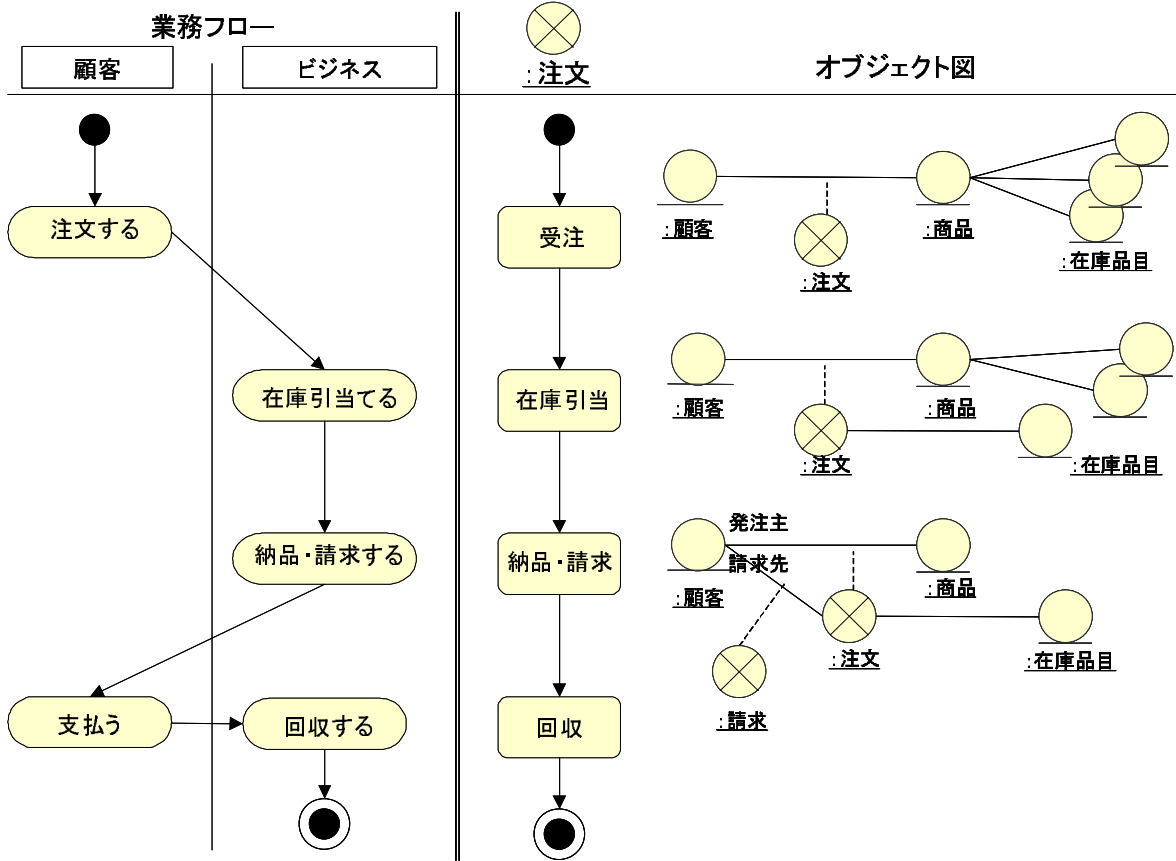


図4 業務フローと注文オブジェクトの状態遷移およびオブジェクト図

6. アナリシス・パターンの例

M.ファウラーのアナリシス・パターン[1]からいくつかの例を関連クラスを用いて表して見よう。理解困難な同著を独立エンティティと事象エンティティを用いて読み解くことができる。

5.1. トレーディング

トレーディングの基本パターンは二つのパーティがある商品の契約を行うという形で表される(図5)。これは図6のオブジェクト図のように買い手と売り手という二つのパーティおよび両者が取引契約を行う商品との間の3項関連と考えることができる。

図5ではすべてのクラスが同等に扱われていてやや見辛いですが、図6のように独立エンティティと事象エンティティを区別すると理解し易い。図6左のオブジェクト図では、契約は買い手と売り手および商品との間の3項関連のリンク属性であるとしている。中央のクラス図(1)は買い手と売り手をひとつのパーティという

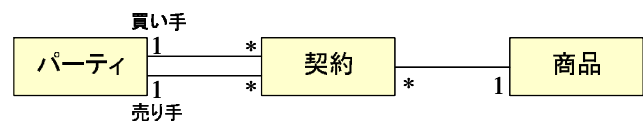


図5 トレーディング

クラスに統合したクラス図である。右のクラス図(2)は関連クラスを通常のクラスとして表現したものである。

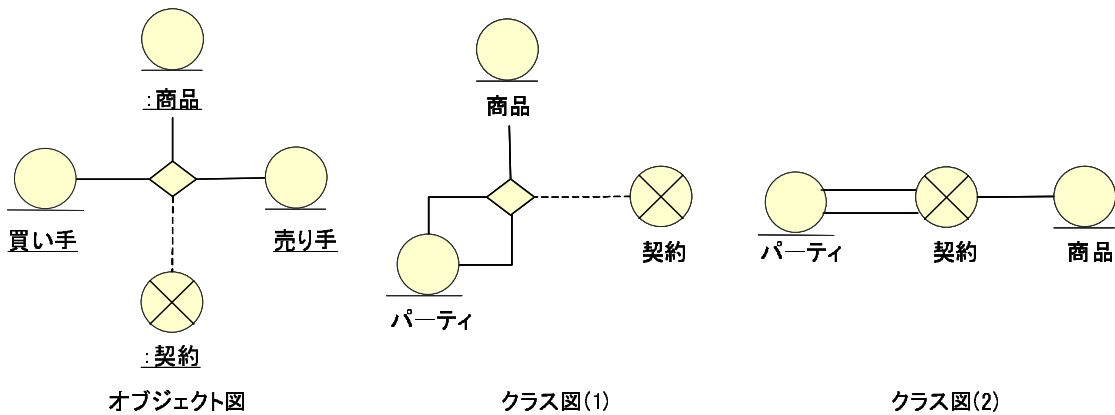


図6 トレーディングのオブジェクト図とクラス図

5.2. 責任関係

責任関係の基本パターン（図7左）は、独立エンティティ委任者および責任者のパーティ間の関連クラスが責任関係であると解釈するとモデルが理解し易い。責任関係は「委任する」というイベントにより生成される事象型クラスと考えることができる（図7右）。

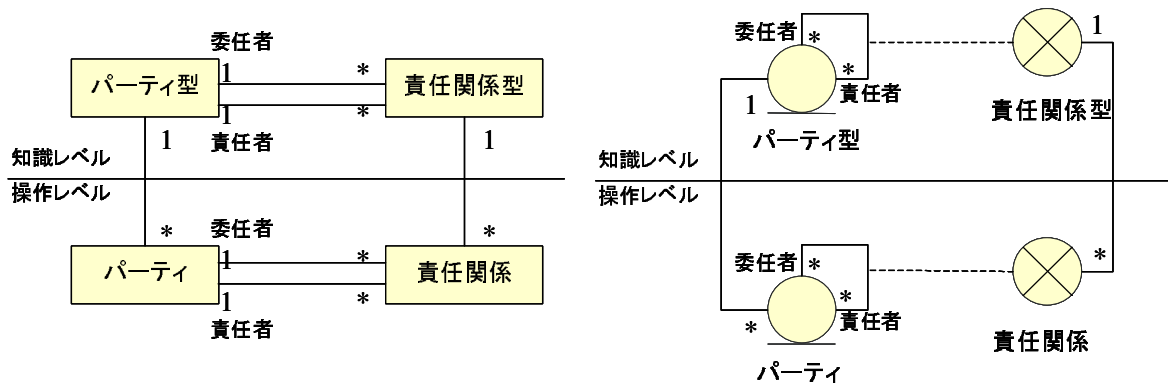


図7 責任関係

7. まとめ

オブジェクト指向の特徴は何でもオブジェクトと考えることができることである。それは便利である反面モデルが理解し辛い原因にもなる。本稿のようにまず独立エンティティを問題領域から発見し、次には業務フローからイベントを抽出し、イベントによる状態遷移でリンク属性がある場合それを事象エンティティとすることができる。事象エンティティを関連クラスとして捉えて独立エンティティと区別するとモデルはより作成し易くかつ理解し易くなる。

参考文献

[1] M.ファウラー[著], 堀内一[監訳], "アナリシス パターン", アジソン・ウェスレイ, 1998