

オブジェクトクラスに基づく最適な制御構造の作成規則*

4 Z C - 6

西澤和正 吉田敦 磯田定宏†

豊橋技術科学大学 工学部 知識情報工学系‡

1 はじめに

ヤコブソンのOOSE[1]では、ユースケースに対応して、制御クラスを中心にその配下に境界と実体を置いたパターンを作成し、それらを組み合わせてモデルを構築する。しかし、この方法では機能とデータを分離した機能指向的なモデルに陥りやすい。そこで、オブジェクトクラスに基づく最適な制御構造の作成方法を提案する。ここで、オブジェクトクラスとは集約により結ばれた実体クラス群のことである。なお、表記法はUML[2]に準じる。

2 二種類の制御構造

制御構造には図1のように制御クラスが制御の中心となる集中構造と、実体クラスも制御を担う分散構造とがある。各構造について、ヤコブソンは以下の適用の指針を示している。

集中構造 操作の順序が変更されやすいとき、または新しい操作が導入される可能性があるときに選択する。

分散構造 オブジェクトが強く結合しており、常に同じ順序で操作が実行されるときに選択する。



図1: 集中構造と分散構造

一般に、集中構造は操作を呼び出す順序を制御クラスにカプセル化するため再利用性が高い。一方、分散構造は実行効率が良いが再利用性が低い。しかし、ヤコブソンの指針はどのようなときにその適用条件が満たされるかを明示していない。このため、実際に適用するには具体性に乏しく、分析者により多様性が生じる余地がある。そこで、本稿では制御構造を選択する規則を提案する。

3 関連の分類

関連には静的な役割と動的な役割がある。まず、関連の静的な役割は次の二種類に分けられる。

- 集約 (aggregation, composition)
全体と部分の包含関係や情報の階層を表現する。またここでは、集約に準ずる関連も含まれるとする。
- 見知り関連 (acquaintance)
オブジェクトが互いに知っている関係を表現する。

次に、動的な役割としてコミュニケーション関連がある。

*An optimum design of control structures based on object clusters

†Kazumasa NISHIZAWA, Atsushi YOSHIDA, Sadahiro ISODA

‡Department of Knowledge-based Information Engineering, Toyohashi University of Technology

- コミュニケーション関連 (communication)
メッセージ送信の経路となる。

なお、関連は静的と動的の両方の役割を果たすことがある。また、上記の定義からわかるようにコミュニケーション関連を識別することにより制御構造を作成できる。

4 制御構造作成規則

以下に従い、オブジェクトモデルを作成する。

Step 1 ユースケースに対応して一つの制御クラスを識別する。ユースケースに関与するアクタごとに境界クラスを識別する。

Step 2 ユースケースの個々の振る舞いの中で設定・更新・参照されるデータをカプセル化する実体クラスを識別する。次に、分野知識に基づき、ここで識別したクラスと集約関係にある(既識別または新たな)実体クラスを識別することで、オブジェクトクラスを作成する。同様に、分野知識に基づき、実体クラス間の見知り関連を識別する。

Step 3 ユースケースの個々の振る舞いを操作を起動するメッセージ送信として捉え、以下の場合に分けて、メッセージ経路を設定する。

- 3-1 制御クラスが実体クラスにメッセージを送るとき、制御クラスと実体クラス間にコミュニケーション関連を識別する(図2)。
- 3-2 同一のオブジェクトクラス内の実体クラスにメッセージを送ることでユースケースが規定する振る舞いを実現できるとき、集約にコミュニケーション関連としての役割を与える(図3)。
- 3-3 複数のオブジェクトクラスが連動してユースケースが規定する振る舞いを実現するとき、オブジェクトクラス間に見知り関連が存在し、かつ、この関連がメッセージの受け手となるオブジェクトを限定していれば、見知り関連にコミュニケーション関連としての役割を与える(図4)。

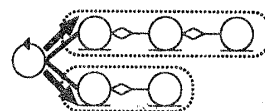


図2: 制御・実体間のコミュニケーション関連



図3: 単一クラス内のコミュニケーション関連

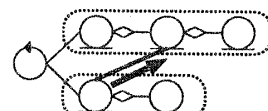


図4: 複数クラス間のコミュニケーション関連

3-4 複数のオブジェクトクラスが連動してユースケースが規定する振る舞いを実現するとき、実体クラス間に暗黙の見知り関連があれば、コミュニケーション関連を識別する。ここで、暗黙の見知り関連とは、メッセージの受け手となるクラスのインスタンスがただ一つであるために、省略されている見知り関連のことである。3-3 の特殊ケースにあたる。

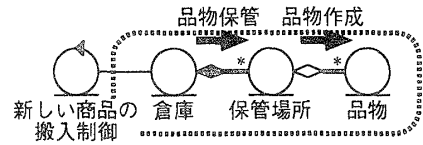


図 7: 単一クラスタ内のメッセージ経路

5 制御構造作成規則の適用例

4 節で述べた規則を倉庫管理システム [1] に適用する。倉庫管理システムとは“顧客から預かった品物を保管し、品物を別の倉庫へ輸送したり、顧客の要求に応じて任意の受取り場所で返却するサービスを提供するシステム”である。ここではユースケース“倉庫間品物運搬”、“倉庫内品物運搬”、“新しい品物の搬入”を対象にする。

Step 1 ユースケースに対応して制御クラス“倉庫間品物運搬制御”、“倉庫内品物運搬制御”、“新しい品物の搬入制御”を識別する。また、ユースケースに関与するアクタ“倉庫主任”、“倉庫内作業員”、“フォークリフト係”、“トラック運転手”、“事務員”に対応してそれぞれ境界クラスが識別されるが、制御構造の作成には関与しないのでここでは省略する。

3-3 倉庫間品物運搬ユースケースの振る舞い“システムは、輸送する品物を待機中に設定する”を実現するためには、図 8 の 2 個のクラスタが連動する必要がある。クラスタ間に見知り関連“輸送する”が存在し、かつ、この関連がメッセージ“待機中に設定”の受け手となるオブジェクトを限定しているため、図 8 の太線で示す見知り関連にコミュニケーション関連としての役割を与える。

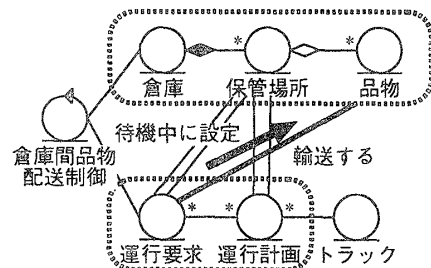


図 8: 複数クラスタ間のメッセージ経路

Step 2 “倉庫間品物運搬”、“倉庫内品物運搬”、“新しい品物の搬入”の各ユースケース内で設定・参照・更新されるデータを発見し、これをカプセル化する実体クラスとして、倉庫、保管場所、品物、運行要求、運行計画、トラック、顧客を識別する。次に、分野知識“品物は倉庫に保管され、倉庫内は区分されている”、“運行要求は複数の運行計画を組み合わせる”から図 5 の 2 個のクラスタと 6 本の見知り関連を識別する。

3-1~3-3 と同様の手順を繰り返し、さらに各クラスに属性を追加することで、図 9 の初期クラス図が得られる。

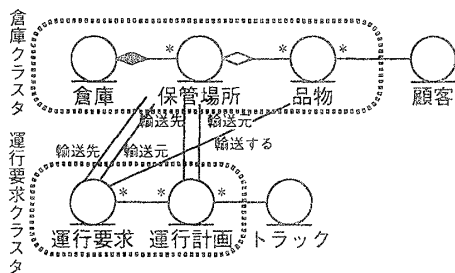


図 5: 倉庫クラスタと運行要求クラスタ

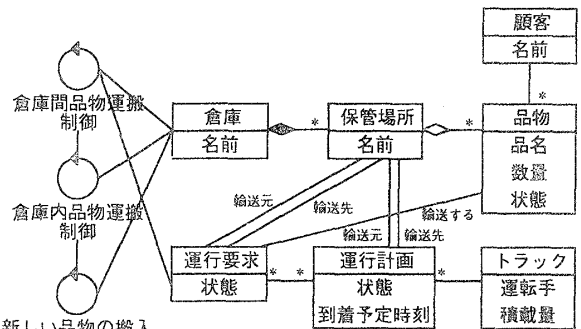


図 9: 初期クラス図 (境界クラスは省略)

Step 3 以下の場合に分けて、メッセージ経路を設定する。

3-1 新しい品物の搬入ユースケースの振る舞い“システムは保管した品物とその数量を登録する”から“新しい品物搬入制御”が倉庫クラスタにメッセージ“品物追加”を送ることがわかり、図 6 の太線で示すコミュニケーション関連を識別する。

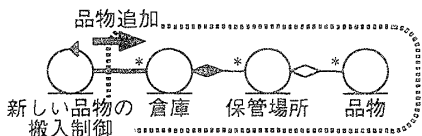


図 6: 制御・実体間のメッセージ経路

3-2 前項のユースケースの振る舞い“システムは保管した品物とその数量を登録する”は、倉庫クラスタ内の保管場所、ついで品物にメッセージを送ることで実現できるため、図 7 の太線で示す集約にコミュニケーション関連としての役割を与える。

6 おわりに

本稿ではユースケース駆動のオブジェクト指向方法論における制御構造の作成規則を提案した。この規則を用いて作成されたモデルでは、オブジェクトが強く結合しているクラスタ内および見知り関連で結合されたクラスタ間では分散構造が選択され、それ以外については制御クラスによる集中構造が選択される。これによりオブジェクトクラスタの独立性と実行効率に関して最適な制御構造が得られる。

参考文献

[1] I. Jacobson, 監訳 西岡利博, “オブジェクト指向ソフトウェア工学 OOSE”, トッパン (1995).
 [2] J. Rumbaugh, I. Jacobson, G. Booch, “The Unified Modeling Language Reference Manual”, Addison Wesley (1999).