

制御オブジェクトの役割の解明*

4ZC-1

小林良介 吉田敦 磯田定宏†
豊橋技術科学大学 知識情報工学系‡

1 はじめに

Jacobson が開発した振舞駆動方法論の代表例である OOSE[1] は、ユースケースに基づき 3 種類のオブジェクトを識別し、これらを組み合わせてモデル構築を行うという一種の設計パターンを用いた分析手法を提案している。この設計パターンを本稿ではユースケース制御パターンと呼ぶ。

- 境界オブジェクト：システムとシステム外部に存在するアクタとのインタラクションを仲介する。
- 実体オブジェクト：システムが扱う情報とそれに対する操作をカプセル化する。
- 制御オブジェクト：実体や境界オブジェクトが持つべきでない振舞いを持つ。

境界・実体オブジェクトの役割は明確であるが、制御オブジェクトの役割は明示されておらず具体的にどのような役割を持つかわからない。そこで本稿では、様々な問題を OOSE で分析し、クラス図に現れる制御オブジェクトの役割を分析する。さらに、制御オブジェクトの役割がデータ駆動方法論の代表例である OMT[2] のクラス図でどのように扱われるかを検討することにより、ユースケース制御パターンを用いたモデル化の有効性を考察する。

2 実験

2.1 制御オブジェクトの 3 つの役割

OOSE を用いて、銀行 ATM 問題 [2]、酒屋倉庫問題 [3]、図書館問題 [4]、貸しポート屋問題 [5]、電子ファイリングシステム [6] をモデル化した。これらのモデルで識別された合計 21 個の制御オブジェクトを分析したところ、制御オブジェクトが次の 3 つの役割を持つことがわかった。

1. オブジェクト間の協調動作を制御する
2. 複数のオブジェクトに依存する制約条件を処理する
3. トランザクションを表す

それぞれの役割を具体例を用いて示す。

2.1.1 協調動作

貸しポート屋問題の「報告書作成」ユースケースは、ポート毎の貸出回数と全貸出平均時間を報告書として作

成する。制御オブジェクトは、「ポート毎の貸出回数を得るためにポートオブジェクトの貸出回数計算操作を呼び、つづいて貸出平均時間を得るために貸出群オブジェクトの平均貸出時間計算操作を呼び出して報告書の作成を行う」という協調動作を制御する。

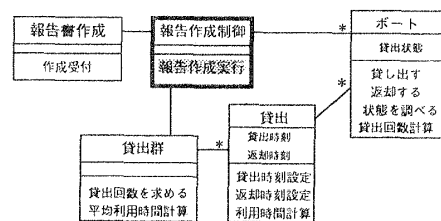


図 1. 「ユースケース報告書作成」の部分クラス図

2.1.2 制約条件

図書館問題の「図書貸出」ユースケースに、以下に示す貸出制約を加える。

『図書館には利用者として一般利用者および特別利用者がいる。図書には一般図書、限定図書、貸出禁止図書がある。利用者および図書の種類によって貸出可能日数が表 1 のように制約される。』

表 1. 図書と利用者の種類における貸出日数制約

	一般利用者	特別利用者
一般図書	7 日	14 日
限定図書	2 日	7 日
貸出禁止図書	0 日	1 日

この制約は、利用者オブジェクトと図書オブジェクトの 2 つのオブジェクトに依存する。このような制約処理は実体オブジェクトが持つべきでない振舞いであり、貸出制御オブジェクトに割り当てられる。

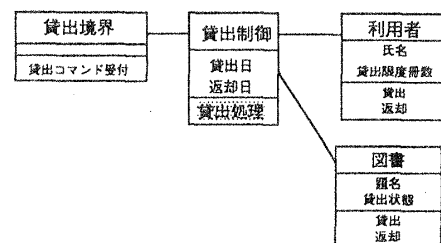


図 2. 「ユースケース図書貸出」の部分クラス図

2.1.3 トランザクション

図書館問題で個々の貸出の履歴を記録するには、「貸出」ユースケースで識別される貸出制御オブジェクトに貸出日や返却日等の属性を持たせ、さらに永続性を持た

*Elucidation the role of control objects

†Ryousuke KOBAYASHI, Atsushi YOSHIDA, Sadahiro ISODA

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

せればよい。このような性質を与えられた制御オブジェクトは処理の記録、すなわちトランザクション [7] を表す役割を担っている。この制御オブジェクトはユースケースが表す動的な概念を具現化したものに相当する。ここで、トランザクションとは、分散処理におけるデータベースのアクセスに限らず、一般的な「処理」や「取り引き」を表す。

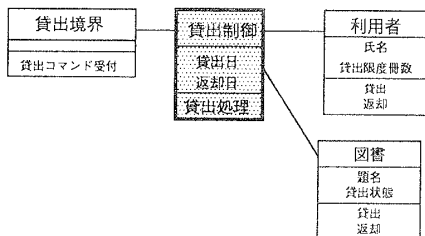


図3. 「ユースケース図書貸出」の部分クラス図

2.2 有効性の考察

2.2.1 トランザクション

OMT は基本的に現実世界をモデル化する手法であり現実世界に存在しない概念はオブジェクトとして識別しにくい。このため、図書館問題を OMT で分析すると図5に示す初期クラス図が得られ、その後の分析で貸出や返却処理等の履歴を取る必要があることに気づき「貸出」トランザクションを識別する。

一方、銀行 ATM 問題を OMT で分析すると初期クラス図として図4を得る。トランザクションを表すオブジェクトが識別されているが、これは問題文中に単語「預入トランザクション」が存在したためである。

これに対し、OOSE では制御オブジェクトが機械的に識別されるので、制御オブジェクトに与えられた機能の履歴を取る必要があるれば、機能に関するデータを制御オブジェクトに与え、永続的にすることで容易にトランザクションをモデル化できる。

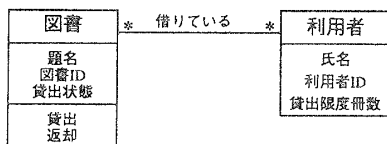


図4. OMT で分析した図書館問題

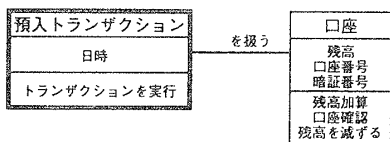


図5. OMT で分析した銀行 ATM 問題

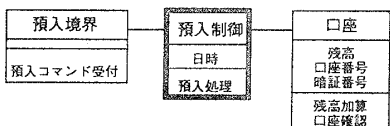


図6. OOSE で分析した銀行 ATM 問題

2.2.2 協調動作

OMT は基本的にデータ構造を表すモデルを構築する手法であり複数のオブジェクトによる協調動作はモデル化しにくい。

貸しポート屋問題を OMT で分析すると図7を得る。2.1.1 項で述べた報告書作成機能を実現するためには、ポートと貸出オブジェクトの協調動作を表現する必要があるが図7では実現できない。このため、新たにオブジェクトを追加する必要がある。

これに対し、OOSE では「報告書作成」ユースケースより機械的に識別される制御オブジェクトが報告書作成機能を持つので、複数のオブジェクトによる協調動作を容易にモデル化できる。

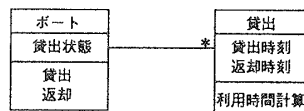


図7. OMT で分析した貸しポート屋問題

2.2.3 制約条件

図書館問題を OMT で分析すると図8を得る。このクラス図で 2.1.2 項で述べた貸出制約の実現を試みる。貸出制約は図書オブジェクトと利用者オブジェクトの両方に依存する。このため、どちらのオブジェクトに制約を持たせてもオブジェクトの独立性が阻害される。

これに対し、OOSE では「貸出」ユースケースより機械的に識別される貸出制御オブジェクトに複数のオブジェクトに依存する制約条件を処理する機能が与えられる。このため、実体オブジェクトの独立性が保たれ再利用性が向上する。

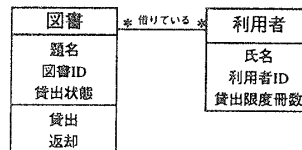


図8. OMT で分析した図書館問題

3 おわりに

OOSE が提案する制御オブジェクトは、システムが持つ機能の静的側面 (トランザクション)・動的側面 (協調動作、制約条件) を表す役割を持つことを既存のクラス図を分析することにより明らかにした。また、これらの役割が OMT でどのように扱われるか分析することにより、ユースケース制御パターンがシステムの持つ機能のモデル化を容易化することがわかった。

参考文献

- [1] I.Jacobson, “オブジェクト指向ソフトウェア工学 OOSE”, 監訳:西岡利博, 渡辺克宏, 梶原清彦, トッパン, 1995.
- [2] J.Rumbaugh, “オブジェクト指向方法論 OMT”, 監訳:羽生田栄一, トッパン, 1992.
- [3] 二村良彦, 雨宮真人, 山崎利治, “新しいプログラミング・パラダイムによる共通問題の設計”, 情報処理, 26(5), pp.458-459, 1985.
- [4] “Problem Set for the Fourth International Workshop on Software Specification and Design”, ACM, Software Engineering Notes, pp.94-96, 1986.
- [5] M.A.Jackson, “System Development”, Prentice Hall, 1983.
- [6] K.W.Derr, “Appling OMT”, SIGS, 1995.
- [7] P.Coad, “Object Models”, Prentice Hall, 1997.