

1P-04 Riel の containment と Jacobson の制御オブジェクトの関係*

大黒 敦 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

I. Jacobson はシステムの機能を記述したユースケースから、制御、実体、境界オブジェクトを識別し、それらのオブジェクトで構成されるパターン(図1)を組み合わせることでモデルを構成する手法を提案している [1][2]。この手法による効果として、Jacobson は保守性の向上を挙げている。

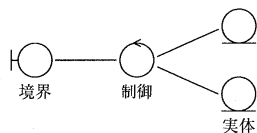


図 1: 境界・制御・実体パターン

Jacobson は制御オブジェクトの役割として「実体や境界オブジェクトに割り当てられない振舞いをもつ」ことを挙げている。しかし、「割り当てられない」という否定的な表現では、制御オブジェクトが担うべき役割が不明確である。また、Jacobson は [1] でいくつかの制御オブジェクトの少数の事例を提示しているが、それらから制御オブジェクトの具体的な役割を認識するのは困難である。このため、制御オブジェクトが様々に解釈されており、混乱が生じている。この一例として Riel [3] が挙げられる。

Riel は Jacobson の手法について以下のように主張している。

1. 初心者は機能指向的にデータと操作を分離した構造、すなわち、機能をもつ “god” オブジェクトを中心とし、配下のオブジェクトにデータとそれに対する get, set 操作を割り当てたモデルを作成しがちである。この構造では、“god” オブジェクトの配下のオブジェクトは意味のある振舞いをもたない。
2. (Riel の解釈では) 図 1 に示すパターンにおいて制御オブジェクトは、実体オブジェクトが本来もつべき意味のある振舞いを抽出し、それらをオブジェクト化したものである。したがって、実体オブジェクトの操作は get, set のみとなる。

ここで、Riel は制御オブジェクトの操作を「実体オブジェクトが本来もつべき意味のある振舞いを抽

*A relation between Riel's containment and Jacobson's control object

[†]Osamu DAIKOKU, Atsushi YOSHIDA, Sadahiro ISODA

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

出したもの」と解釈している。しかし [4] によると、制御オブジェクトの役割は協調制御、制約条件、トランザクションの三つである。したがって、Riel の解釈は [4] と矛盾する。以下ではこの矛盾を解明する。

2 containment の適用事例

Riel はオブジェクト間の関係の一つとして包含 (containment) 関係を挙げている。これは、UML [5] における集約と同等である。Riel の containment の適用事例 [3] を以下に示す。

2.1 銀行 ATM システム

2.1.1 ATM

現実世界における ATM 端末の機能をモデル化する。ATM 端末はトランザクションを処理する前に、以下の手順に従って顧客の認証を行なう。

1. Card Reader はカードが挿入されたことを検知すると、カードに記録されている PIN (Personal Identification Number) を読みとる。
2. Card Reader は顧客が入力する PIN を Keypad から得る。
3. 顧客が入力した PIN をカードから読みとった PIN と照合し、その結果を Display に表示する。

図 2 に示すクラス図は、全体 (containing) オブジェクト ATM が要素 (contained) オブジェクトとして Display, Card Reader, Keypad を含むことを表す。

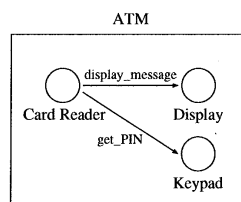


図 2: ATM 端末のクラス図 (1)

また、図 2 で実線の矢印により、Card Reader と Display, Keypad 間に利用関係があることを表している。Riel はこの関係を、図 3 に示すように ATM と Card Reader, Display, Keypad 間の利用関係に置き換えている。これは、Riel の「contained オブジェクト間の利用関係は最小限にすべき」という主張に基づく。

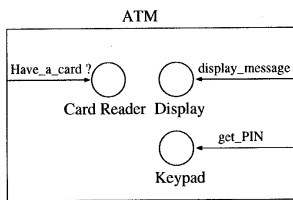


図 3: ATM 端末のクラス図 (2)

2.1.2 Super Keypad

顧客の認証処理をさらに詳細化する。顧客が PIN を入力する間、ATM 内部では以下の処理を行う。

1. 顧客がキーを押すと、Keypad はキーに対応する文字を検出する。
2. Display 上に “*” を表示し、次の入力を促す “_” を表示する。

以上の処理は図 4 に示す Keypad と Display 間の利用関係で実現できる。

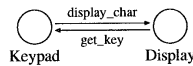


図 4: PIN の入力処理

しかし、Riel は ATM の場合と同様の理由により、containing オブジェクトとして Super Keypad を導入することで、Keypad と Display 間の利用関係を避けている (図 5)。

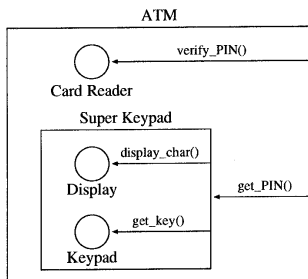


図 5: Super Keypad の導入

3 containment と制御オブジェクトの関係

図 6 に示す containment を考えよう。containing オブジェクト A は以下に示す役割をもつ。

- (1) オブジェクト B, C のグループ化
- (2) 外界とのインターフェース
- (3) オブジェクト B, C の制御

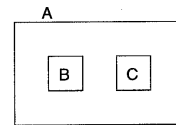


図 6: containment

そこで、この三つの役割を分離しよう。図 7 に示すモデルで、オブジェクト A はオブジェクト B, C をグループ化する役割のみをもち、オブジェクト Y は上記 (2) の役割をもち、オブジェクト X は上記 (3) の役割をもつ。すなわち、オブジェクト Y は図 1 に示す境界オブジェクトに対応し、オブジェクト X は制御オブジェクトに対応する。

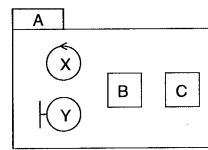


図 7: 制御の同列化

以上より、containing オブジェクトの三つの役割を分離することにより、containment が暗黙に含んでいた制御オブジェクトの役割が明示された。しかし、Riel はこの三つの役割を分離することを考えていなかったため、containing オブジェクトが制御オブジェクトの役割を含むことに気付かなかったと推察される。すなわち、制御オブジェクトは配下の実体オブジェクトの操作をオブジェクト化したものではなく、[4] に示す役割を担うものである。

4 まとめ

本稿では、Riel の制御オブジェクトに関する誤解を解明した。Riel は制御オブジェクトを、「実体オブジェクトの操作をオブジェクト化した “god” オブジェクトである」と批判しているが、これは彼が提唱する containment なる構造に三つの役割が含まれていることに気付かなかったことによる。

参考文献

- [1] Ivar Jacobson 他, “オブジェクト指向ソフトウェア工学 OOSE: Use-Case によるアプローチ”, トッパン, 1995.
- [2] Ivar Jacobson, et al., “The Unified Software Development Process”, Addison-Wesley, 1998.
- [3] Arthur J. Riel, “Object-Oriented Design Heuristics”, Addison-Wesley, 1998.
- [4] 小林良介 他, “制御オブジェクトの役割の解明”, 情報処理学会第 59 回全国大会講演論文集, 1999.
- [5] Rational Software Corporation, “Unified Modeling Language”, <http://www.rational.com/>.