

オブジェクト指向開発手法における「関連の設計」

4R-3

～実装言語をC++とした場合～

坂本英樹 大森 晃

東京理科大学工学部経営工学科

1. はじめに

ソフトウェア開発のためのよりよい手法が望まれるなか、オブジェクトを中心に分析・設計・実装を行う手法として「オブジェクト指向開発手法」が注目されている。これは多くの利点を持つ一方で技術的な問題も抱えている。その一つとして、関連の設計に関して体系的な手法が整備されていないという問題点を挙げる事ができる。よく用いられるオブジェクト間の諸関係のうち関連は、OOP言語がそれに対応する機構を備えていないために実装（コーディング）上の技術的ボトルネックとなり、実装できるように関連を設計することを余儀なくされるというのが技術的現状である。

文献 [1]と[2]によれば15種類のオブジェクト指向開発手法（Booch,CCM,CoadYourdon,OMTなど）の設計フェーズにおいて関連がオブジェクト間の関係のタイプとして用いられているが、関連の設計に明示的に言及している手法はOMT（Object Modeling Technique）法だけである。しかし、OMT法はごく簡単な技術的示唆しか与えていない。このように、関連の設計は現在のオブジェクト指向開発手法においてかなり共通した技術的課題であるにもかかわらず、そのための体系的な手法は未整備であるというのが現状である。

そこで本研究では、関連の設計に関する体系的な手法の整備に向けてポインタ関連図、メッセージフロー図などを提案し検討を行う。オブジェクト指向開発手法としては関連に関してかなり豊富な概念を有するOMT法を主な対象としているが、オブジェクト間の関係のタイプとして関連を用いる他のオブジェクト指向開発手法にも我々の提案は役立つものである。

2. 「関連の設計」に関する体系的な手法の整備に向けて

2.1 関連の区分

OMT法で用いられるオブジェクト間の関連を区分すると表1のようになる。今回検討した関連は、制約なしの2項関連で、多重度がそれぞれ1対1, 1対多, 多対多のものである。

2.2 関連の方向性の吟味

多くの場合、関連は実装においてメッセージパッシングに置き変わる。メッセージの流れには方向性が存在することを考えると、関連の方向性を明確にする必要がある。OMT法においてはこれに関する示唆を行っているが具体的ではない。我々は、関連し合うそれぞれのクラスの属性と操作から関連の方向性を吟味することを提案する。

2.3 ポインタ関連図

プログラムの実行時において、クラスで定義された属性および操作は全てインスタンスが受け継ぎ、コンピュータ上でのインスタンスはメモリ領域に対応する。あるクラスのインスタンスが、別のあるクラスのインスタンスにメッセージを送るためには、そのインスタンスの所在（アドレス）が必要になる。C++による実装において関連は、ポインタを用いたメッセージパッシングによって実現されることを考えると、メッセージの送信元インスタンス（センド）が、メッセージの送信先インスタンス（レシーバ）へのポインタを持つことが必要となる。異なるクラスに属するインスタンス間で、メッセージの送受信を実現するために行なうC++のコーディングとしては、あるクラスのインタフェース部に、別のあるクラスへのポインタ変数を定義し、そのポインタ変数に具体的なメッセージの送信先インスタンス（レシーバ）のアドレスを格納するといったことが必要となる。ここでは、インスタンスに持たせるこのようなポインタ情報を記述するためにポインタ関連図（図1）を提案する。

表1 OMT法で用いられる関連の区分

項番	関連の区分	制約	多重度
1	2項関連		1対1 1対多 多対多
		(条件付き)	1対多 多対多
		(限定付き)	1対1 1対多 多対多
		(リンク属性)	1対1 1対多 多対多
2	3項関連		1対1対1
		(リンク属性)	1対1対1

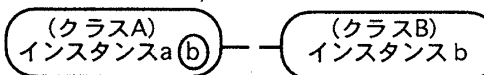


図1 ポインタ関連図

図1は、クラスAに属するインスタンスaがクラスBへのポインタ（○印）を持ち、そのポインタ（○印）が具体的に、クラスBに属するインスタンスbのアドレスを持つこと示している。なお、角の丸い四角（インスタンス）間の点線は、どのクラスへのポインタを持つかといった情報を表わすものであり、ここではクラスAのインスタンスaが、クラスBのインスタンスbと点線で結ばれていることから、クラスAのインスタンスaが持つポインタ（○印）はクラスBへのものであることを示している。

2.4 メッセージフロー図

関連をメッセージパッシングに置き換える上でさらに、どのインスタンスからどのインスタンスへどんなメッセージが送信されるのか、またそのメッセージの送信元となる操作は何かを明確にする必要がある。これを行なうために、図2に示すようなメッセージフロー図を提案する。図2では、クラスAのあるインスタンスからクラスBのあるインスタンスへメッセージが送信されることを矢印によって示している。矢印上の操作名（操作a）は、そのメッセージの送信元が、クラスAで定義された当該インスタンスの有する操作aであることを意味している。また「操作a（操作b）」は、操作aの中から操作bをcall（つまり操作bというメッセージを送信）することを意味している。なお、矢印上に操作名を記述するのは、多くの操作が定義されている場合に、メッセージの送信元となる操作を識別しやすいようにするためである。

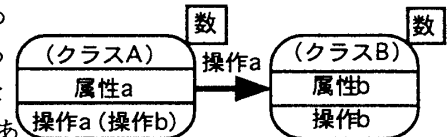


図2 メッセージフロー図

インスタンスの右肩にある四角内の数は、設計時の多重度をそのまま表すためのものであり、これによって実装の際にポインタをどれくらい用意すればよいかといった指針を与える。数の書式はOMT法に準じるものとし、多重度が1の場合には省略するものとする。

3. 「関連の設計」－具体例

ここではこれまで提案してきたことをもとに、「利用者」クラスと「引き出し専用ATM」クラスの簡単なやり取りを例にして、多重度が1対1である2項関連の設計を行う。分析での「利用者」クラスの属性と操作はそれぞれ「利用金額」と「残高を調べる」であり、「引き出し専用ATM」クラスの属性と操作はそれぞれ「残高」と「利用金額を調べる」である。これをもとにOMT法のオブジェクト図で分析・設計を行い（図3）、さらに提案したポインタ関連図とメッセージフロー図を用いて「関連の設計」を行った（図4）。なお、図3の設計モデルに定義されている属性と操作より、関連の方向性は双方向であることが分かる。

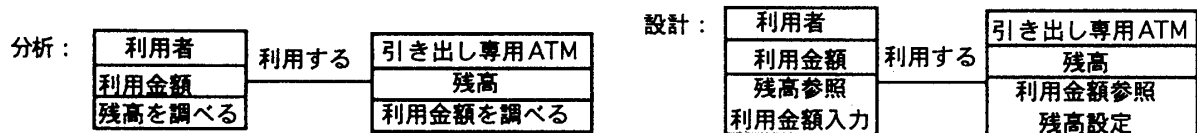


図3 分析・設計モデル

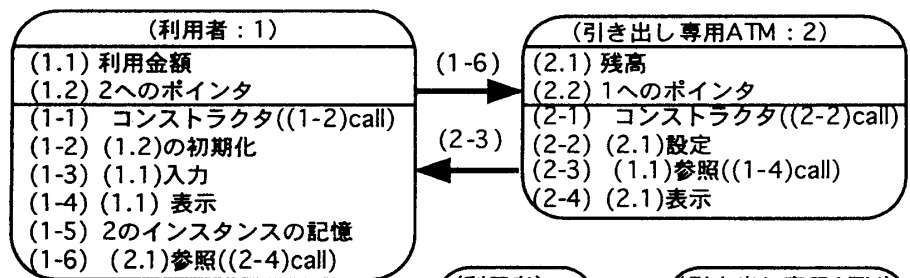
4. おわりに

本研究では、関連の設計に関する体系的手法の整備に向けて、関連の方向性の吟味、ポインタ関連図、メッセージフロー図を提案し、それらを用いて関連の設計の具体例を与えた。今後の課題は、ランタイム中におけるインスタンスの動的生成を考慮した関連の設計を行っていくことである。

【参考・引用文献】

[1] アンドリュウ・T・F・ハット 編, オブジェクト・マネジメント研究会訳: オブジェクト分析と設計～オブジェクト指向21手法の解説と徹底比較～, トッパン (1995)
 [2] J. ランポー, M. ブラハ, W. プレメラニ, F. エディ, W. ローレンセン 著, 羽生田栄一 監訳: オブジェクト指向方法論OMT, トッパン (1992)

メッセージフロー図:



ポインタ関連図:

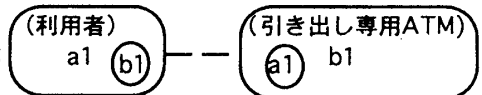


図4 関連の設計モデル