

オブジェクト指向設計工程に関する一考察

3G-2

— C++をターゲットとして —

笑輪行真* 鈴木賢** 稲葉稔智* 石原平太郎*

*沖電気工業株式会社

**沖通信システム株式会社

1. はじめに

オブジェクト指向によってソフトウェアを開発するフェーズは大きく2つに分けられる。1つは分析フェーズであり、もう1つは設計フェーズである。分析フェーズでは、問題を解決するために必要となるオブジェクトを抽出する。¹⁾ 設計フェーズでは、分析フェーズで得られたオブジェクトを実際に設計する。設計フェーズはプログラミング言語に大きく依存する部分であり、その工程を明確に記したものは過去にはあまり例がない。そこで今回は、オブジェクト指向言語C++をターゲットとして、オブジェクトの設計工程、および設計工程の生産物であるドキュメントのフォーマットについて報告する。

2. 設計工程の明確化の必要性

オブジェクト指向によるプログラミングには、探索型が適しているということは過去に何度か報告されている。^{2) 3)} これは最終的なオブジェクトの仕様を事前に予測することが不可能であること、オブジェクトはそれ自身で1つの動作可能なモノであることによる。しかし、大規模なシステムを設計する場合、多人数で開発を行なうため、オブジェクトのインタフェースを決めたり、オブジェクトの仕様がある程度は事前に決めることが必要になる。これはオブジェクト指向による設計にも、その工程を明確にする必要があることを示している。またオブジェクト指向による設計にはつきものといわれる設計変更の頻発も、設計工程を明確にすることで能率が上がると考える。

3. C++のための設計工程

設計フェーズはオブジェクトを実際に設計するところであり、プログラミング言語に大きく依存するフェーズである。ここでは、オブジェクト指向言語C++をターゲットとして、そこで必要となる工程、およびコーディング時のファイル構成について述べる。はじめに、他のプログラミング言語にも適用可能な概略設計工程について述べた後、C++だけに適用できる詳細設計工程について述べ、最後にファイル構成について述べる。

3. 1. 概略設計工程

ここでは、オブジェクト指向分析¹⁾の出力であるオブジェクトフロー図をもとに全体の流れの再確認(以下の(1)、(2))と、各オブジェクトを生成するクラスの基本設計(以下の(3)から(7))を行なう。

(1) メッセージの送信に対するエラー種別、およびそれらに対する処理を決定する。

(2) メッセージを使って渡すデータ、あるいはそのメッセージを使うことによって得られるデータ、およびそれらのデータ構造を決定する。

(3) クラス名の決定

(4) クラスのデータ構造の決定(機能編)

ここでは、(2)までに決定したクラス機能を実現するために必要となるデータ構造を決定する。

(5) インタフェースの決定(メッセージ編)

ここでは、クラスに対してパッシングするメッセージを決定する。詳細を以下に示す。

(a) 外部関数の名称

(b) 外部関数の引数の数、型

この時引数の型として構造体を用いるならば、その型も決める。

(c) 外部関数の戻り値の型

(d) オブジェクトの生成方法の決定

初期化の種類、引数の数、型を決める。

(6) インタフェースの決定(マクロ編)

クラスへのパッシング、クラスからの返答に使用されるパブリックなマクロを決定する。

(7) クラス宣言ファイルの作成

ここでは以下のことを行なう。

(a) 継承の種類

オブジェクト指向分析の出力を参考に、継承の種類を決める。

(b) フレンド関係の確立(C++特有)

(1)、(2)で変更されたオブジェクトフロー図から、フレンド関係を持たすべきオブジェクトを選出する。

(c) インクルードファイルの作成

そのクラスで使用する他のクラスのクラス宣言ファイルをインクルードするように作成する。このファイルの作成によって、クラス構造が複雑になってもクラス宣言ファイルのインクルードが簡易化できる。

(d) クラス宣言ファイルの記述

前項までに明らかになったクラス名、クラスの内部変数、クラスの外部関数、およびインクルードファイルを使って記述する。ただし、ここで記述されたファイルは完成されたものではない。あくまでも詳細設計工程を各自で行えるようにするためのものである。

3. 2. 詳細設計工程

ここでは、概略設計工程の出力を参考に実際にクラスを設計する。この工程は、各クラスごとに各自で行える。

(1) マクロ定義ファイルの作成

概略設計工程で決定したパブリックなマクロ、およびクラス宣言ファイル中で使用するローカルなマクロを記入する。

(2) 関数定義

クラスの外部関数、および外部関数定義にともなう必要性が生じる内部関数を定義する。さらにそのような内部関数をクラス宣言ファイルに追加する。

(3) クラスのデータ構造の決定(内部関数編)

関数定義の最中に内部変数とすべきものが生じた場合それをクラス宣言ファイルに追加する。

3. 3. ファイル構成

ここでは、実際にオブジェクトをコーディングする際のファイル構成を示す。1オブジェクトは4つのファイルからなり、それぞれのファイルは2重のインクルードに耐え得るようにしてある。以下にクラス名がMaleの場合の例を示す。

インクルードファイル	マクロ定義ファイル
//File name: Male.inc	//File name: Male.def
#ifndef MaleINC	#ifndef MaleDEF
#define MaleINC	#define MaleDEF
#include "..."	#define xxx ...
#include "..."	#define yyy ...
#endif	#endif

A study of Object Oriented Design - for C++ -

Yukimasa MINOWA*, Ken SUZUKI**, Toshinori INABA*, Heitaro ISHIHARA*

*Oki Electric Industry Co., Ltd., **Oki Telecommunication Systems Co., Ltd.

```

クラス宣言ファイル          クラス定義ファイル
//File name: Male.hh       //File name: Male.cc
#ifndef MaleHH              #include "Male.hh"
#define MaleHH              #include "Male.def"
#include "Male.inc"         #include <...>
class Male:... {          void Male::run(int ...)
...
public:
...
};
#endif
    
```

4. C++のための設計ドキュメント

手法によらずソフトウェアを開発した場合、ソフトウェアの保守や再利用のためにドキュメントを残さなくてはならない。ここではオブジェクト指向で開発されたソフトウェアのためのドキュメントを、C++をターゲットとして述べる。

(1) パスチャート (図1)

これは、分析フェーズの出力であるオブジェクトフロー図に、以下の項目を追加し、問題の解法の全容を具体的に把握することができるようにしたものである。

- ・メッセージパッシング時のエラー処理
- ・メッセージパッシングの際に付加するデータ、およびそれによって得られるデータ
- ・オブジェクトの生存期間

(2) マクロ定義表 (表1)

(3) オブジェクトインタフェース仕様 (図2)

これは、オブジェクトを使用する際に必要となる各種のインタフェースを示したものである。

(4) 拡張オブジェクト階層図 (図3)

これは、分析フェーズの出力であるオブジェクト階層図に、以下の項目を追加し、クラス継承の把握性を向上させたものである。

- ・継承の種別
- ・フレンド関係の明示
- ・分析フェーズで新たに出現したオブジェクト

(5) オブジェクト仕様書

これは、3. 3項で述べた4つのファイルからなる。

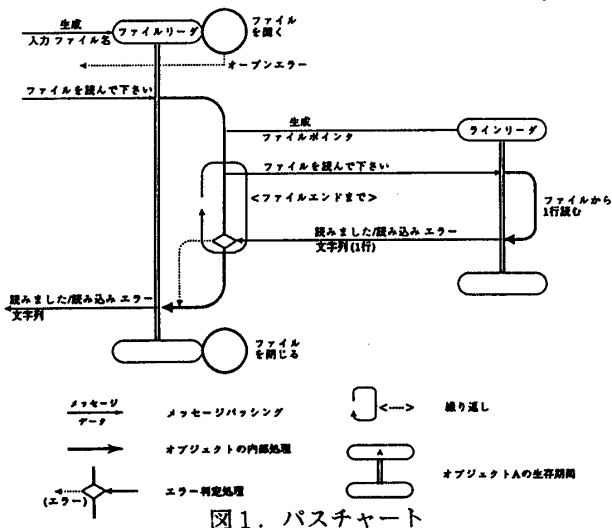
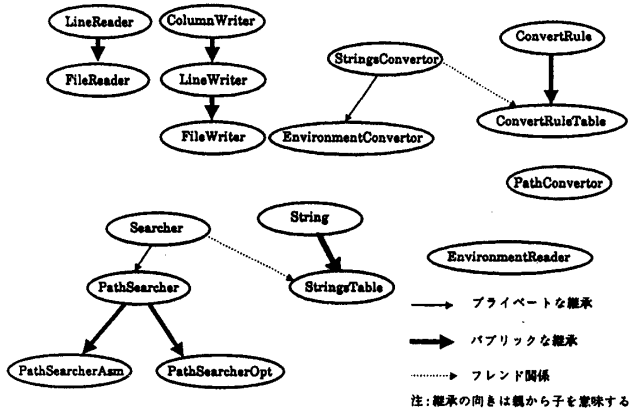


表1. マクロ定義表

マクロ	定義	使用オブジェクト
NG	0	LineReader FileReader ...
OK	1	LineReader

オブジェクトインタフェース仕様	
クラス名	Searcher
受諾メッセージ	out
メッセージパッシングパターン	out(char**)
機能	
このメッセージは、オブジェクトSearcherが探し出した文字列をすべて出力する。出力は文字列のポインタの配列で達成され、配列の終端部分には(char*)0が挿入される。また、文字列、及び文字列のポインタの配列はnew演算子で確保されるため、このメッセージの使用は責任をもってそのエリアの解放を行わなくてはならない。	
エラー通告	
(1) OK : 正常終了	
(2) NG : メモリ容量等の理由で探索した部分を出力できない	
(3) NON : 探索で見つかった部分がなかった	
関連受諾メッセージ	
search(void)	

図2. オブジェクトインタフェース仕様



5. まとめ

本報告では、オブジェクト指向によるソフトウェア開発フェーズのうち、設計フェーズについて、その工程およびそのドキュメントについて述べた。本報告は、CTRONⁱⁱ⁾仕様OS (沖電気製)の初期設定ツール (システムコンフィギュレータ)の設計経験をもとにまとめたものである。また、VAX/ULTRIXⁱⁱⁱ⁾、VAX/VMSⁱⁱⁱ⁾間のパス変換ツールの試作を通して本工程を試験した。その結果、概念的な評価であるが、設計手順が示されたことによってオブジェクトの設計が行い易くなったと評価する。

参考文献

- 1) 稲葉, 他: 構文解析を用いたオブジェクト指向分析手法, 第41回情報処理学会論文誌, 1990
- 2) 福永, 沼尾: オブジェクト指向言語のプログラミング環境, 情報処理, Vol.29, No.4, pp.334-343(1988)
- 3) 横手: オブジェクト指向言語のプログラミング環境 - SmallTalk-80における環境を例に -, 情報処理, Vol.30, No.4, pp.334-346(1989)

ⁱ⁾CTRONはCommunication & Central TRONの略です。
ⁱⁱ⁾TRONはThe Real-time Operating system Nucleusの略です。
ⁱⁱⁱ⁾VAX/ULTRIX, ⁱⁱⁱ⁾VAX/VMSはDEC社の商標です。