オブジェクト指向設計法に関する一考察

4S－1

小尾 俊之　櫻庭 秀次　松村 一夫
(株式会社 東芝　システム・ソフトウェア技術研究所)

1．はじめに

オブジェクト指向言語を用いたアプリケーション開発が数多く行われているが，その一方で，「何を，どのようにしてオブジェクトとするか」，「良いオブジェクトとはどういうものか」，「それを表すときの表し方は」等といった問題もある．要するにこれは，オブジェクト指向言語を用いて開発する際の設計法が必ずしも明確になっていないことによる．オブジェクト指向の利点を最大限に活かすには，その利点をもたらす概念，機能をどのように適用していったらよいかの指針を与える必要がある．

我々は，ソフトウェア生産の工業化を行うIMAP の一環として，オブジェクト指向に基づいた設計法の構築を目標としている．本稿では Booch の提案したオブジェクト指向設計法[1]に従って実際のアプリケーションを開発し，その設計法について，オブジェクト指向の特徴を考慮しているかという観点から評価し，課題を明確にした．

2．オブジェクト指向設計方法論の要件

一般に設計方法論とは，次の三つについて何らかの指針が揃っている必要がある．

設計方法論 ┬ 設計の手順（発見法）
　　　　　├ 設計の記述法（表現法）
　　　　　└ 良否の判断基準（評価法）

一方，従来の手続き型に比べて，オブジェクト指向の特徴的な概念，機能，及びそれによる利点には次のものがあると考えられる．

1)実世界にそったモデル化：問題解決のための機能を計算機上に写像し易い．
2)抽象化（モジュール化）：データ抽象化や機能を局所化することにより，機能変更時の他モジュールへの影響の減少が図れる．再利用し易い．
3)クラス／インスタンス：同性質のオブジェクトを生成し易い．
4)継承（インヘリタンス）：差分プログラムによる記述の容易性．クラスを系統的にまとめることにより管理，再利用が容易におこなえる．
5)並列性：自立的に活性化するオブジェクトをモデルに考えるため，動作を自然に表現できる．

従って，オブジェクト指向設計法としての理想像は，これらの概念や機能を活かした設計法，すなわち，例えば継承をうまく利用していく為にはどのような手順で設計を行い，それをどのように設計結果として表現し，また，その設計の良否を判断をする何らかの基準が存在することである．

3．事例へのオブジェクト指向設計の適用

今回，Booch の提案したオブジェクト指向設計法を実際のプログラム開発に適用した．事例は，木構造チャートを基本とした設計記述からアセンブラのソースコードを生成するプリプロセッサ（一種のコンパイラ）TCGEN/ASM とした（図3.1）．

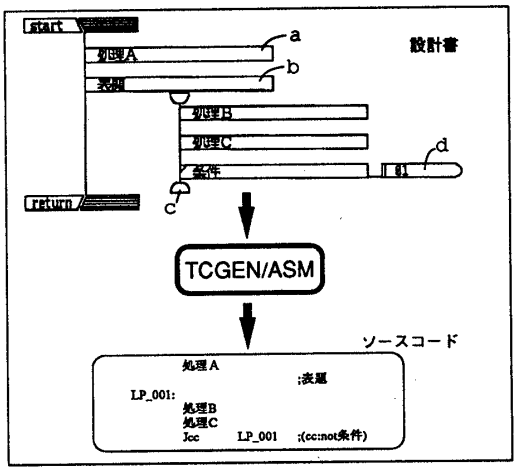

図3.1 TCGEN/ASMの位置づけ

以下に問題の informal-strategy （上位レベル）を示す．

> 設計データを読み込み、対応するアセンブリ言語のソースコードを出力する。
> 設計データには、ボックス記号、展開記号、修飾ボックス記号があり、ボックス記号には、展開記号や、修飾ボックス記号が付加される場合がある。

これをもとに次のオブジェクトを選定した．

ボックス記号：図3.1 の a，b のようなボックスの場合，展開記号オブジェクトより起動される．設計データオブジェクトへメッセージを送信することによりボックス内の設計データを受信し，対応するソースコードをソースコードオブジェクトへ送信する．
展開記号：構造化言語の3要素である ⧖(反復:c)，⧗（分岐）或いは ⊐（順次）の場合に起動される．設計データオブジェクトからデータを読み込み対応するボックス記号オブジェクトを起動する．
修飾ボックス記号：d に示すような修飾ボックスの場合，ボックス記号より起動される．対応するソースコードをソースコードオブジェクトへ送信する．
設計データ：記述された設計データを保持しており，メッセージによる要求により設計データを返信する．
ソースコード：受信したソースコードのメッセージを特定のファイルへ格納する．

モデル化の結果の概要（上位レベルのオブジェクト）を

A study for Object-Oriented Design.
Toshiyuki OBI, Shuji SAKURABA, Kazuo MATSUMURA
TOSHIBA COPORATION.

図3.2 に，クラス階層を図3.3 に示す．

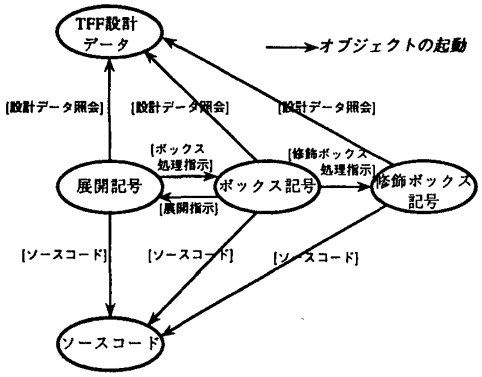

図3.2 TCGEN/ASMのモデル化

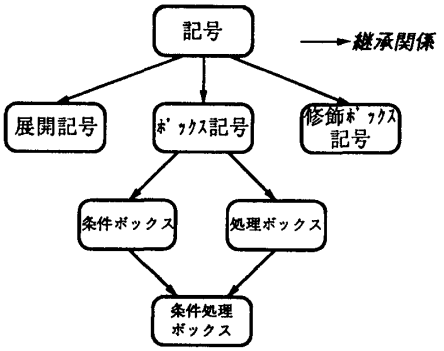

図3.3 TCGEN/ASMのクラス階層

4．評価と検討

この適用を通して，2章で挙げられた概念，機能が，方法論の中に活かされているか（考慮されているか）を評価した．その結果を表4に示し，以下にその概要を述べる．

	手順	記述法	基準
モデル化	○	△	△
抽象化	○	○	△
クラス/インスタンス	△	△	△
継承(インヘリタンス)	×	△	×
並列性	△	×	×

○:利用可能な指針がある
△:一部考慮している
×:考慮していない

表4 Boochのオブジェクト指向設計の評価

1) 実世界にそったモデル化について

事例のようなデータ変換では，一部実世界にそったモデル化が難しいところもあるが，全体的には比較的容易にモデル化（オブジェクトと意識する）することができた．表記法や基準については特に定められていないが，基準については実世界との対比が行えるので良否の予想は可能である．しかしいずれにしろ何をオブジェクトとするかといったことに関する明確な指針はまだない．

2) 抽象化について

手順的にも記述的にも比較的容易に行えた．また，基準についても表記法によって書かれた内容から判断できた（例えば，パッケージ間の線の本数等）．

3) クラス／インスタンスについて

クラスをオブジェクトとみなすか，あるいはオブジェクトの型とみなすかによってクラスの意味が異なるため明確な指針はない．手順的には各オブジェクトの名称や属性を選定することによって，クラス／インスタンス，あるいはクラス階層を意識していくことができた．表記法については，クラスをオブジェクトの型とみなす場合は，オブジェクトの関連図にクラス階層を入れると複雑性が増加してしまったので区別して表現したほうがよい．

4) 継承（インヘリタンス）について

Booch の場合は，Ada での開発を対象としているため継承については考慮されていない．図3.3 の結果は Booch の手順からは得られなかった．これを見つける手順については5章で検討する．ただし表記法については，Booch のタスク間の関係を表す図を用いて，継承関係を表現することが可能である．

5) 並列性について

1) によって実際の動きを考えると並行性を考えるのは比較的容易であるが，その表記法については同期，非同期の情報が得にくく，基準についても明確な指針はない．

全体的にみると，手順的としては基本的には指針が示されている．ただしその詳細については，経験に頼るところが多い．また記述法は，表現力や簡潔性に於いてまだ改良の可能性がある．さらに基準については，今のところ明確な指針がないと言える．

5．継承についての検討

本章では，特に Booch では考慮されていないと思われる継承について考える．

オブジェクト指向設計において，継承の概念を活かすためにはクラス階層を決める段階でより明確な手順が必要である．つまり Booch の方法においてオブジェクトの属性を選定する段階で次の2項目を追加する．

1)属性を抽出する際には，名詞（オブジェクト）に付随する形容詞やその名詞が本来意味的に備えている基本要素等に注目し，同じ言葉（意味的に）については共有化を図る．

2)その属性をオブジェクトとして定義し，最初に選定されたオブジェクトと継承関係を結ぶ．

具体的には，3章の informal-strategy に於いて，XX記号という名詞（オブジェクト）には特定の規則に基づいた文字列（記号）という属性が含まれている．これをオブジェクトとする．つまり各XX記号内部には記号というオブジェクトが存在しており継承関係が存在するといえる．ボックス記号についても同様に考える．これによって得られた継承関係を図3.3 に示す．

6．おわりに

オブジェクト指向設計法では，オブジェクトの選定が最も難しく，かつ重要な部分である．Booch の設計法ではオブジェクトの選定段階で明確な指針が与えられておらず，オブジェクトの再利用についても触れられていない．そのため，オブジェクト指向設計としてはまだ改良の余地がある．なお，再利用については，オブジェクトの選定において着目した属性が既存のクラス（ライブラリ）に含まれていればそれを再利用すべきである．これを効率よく行うためには既存のクラスの検索機能など設計をサポートする環境も重要であると考えられる．

今後は，これらの課題について検討を行い，設計方法論を含めたオブジェクト指向設計環境の構築を目指す．

参考文献

G.Booch:Object-Oriented Development, IEEE, trans.on SE VOL SE-12 No.2

等々力：オブジェクト指向設計言語による在庫管理システムの記述，情報処理学会研究報告，Vol.88,No.7,1988