ソフトウェア工学 58－8
（1988. 2. 4）

オブジェクト指向言語による在庫管理システムの記述

等々力 正文
（財）鉄道総合技術研究所

本報告ではオブジェクト指向言語の一つであるSmalltalk-80を用いて、酒類販売会社の在庫管理システムの作成の共通問題[6]を解く。先ず、オブジェクト指向の考え方と特長を要約し、オブジェクト指向設計法について述べ、次に、この設計法に従い在庫管理システムを設計し、それのSmalltalk-80による記述を試みる。その結果、作成されたプロトタイプの設計と実装について次の結果を得た。

① 共通問題のような事務処理では、帳票、ファイル、伝票等のデータの集まりとその処理機能とをオブジェクトとすることが分かり易い設計／ソフトウェアである。

② 在庫管理システムでは本質的に有効なクラス階層は無かったが、Smalltalk-80がもつ is-a 関係と AKO関係を基本構造として含むアプリケーションで有効である。

③ Smalltalk-80と他のオブジェクト指向言語による記述では、オブジェクトの選定では差異があり、ステップ数ではSmalltalk-80が大きく、Concurrent Prolog が最小であった。

Description of one part of an inventory control system
for alcohol dealer by Smalltalk-80

Masafumi TODORIKI

Railway Technical Research Institute
Hikari-cho 2-8-38, Kokubunji-si, Tokyo 185, JAPAN

In this paper,using Smalltalk-80 that is one of the object-oriented programming languages,the common problem of specifying the inventory control system for an alcohol dealer is solved.First,the basic concept and specific characteristics are summarised and the object-oriented design method is discussed.Then,according to the method the inventory control system is designed and coded with Smalltalk-80.The design and its prototype are discussed with the following results obtained.

(1) In business applications, it is well understandable for design and software to take data collection such as a slip and a file,and its processing functions as objects.

(2) In this solution the class hierarchy is not used.Smalltalk-80 is suitable for applications that contain fundamental structures such as "is-a"and"AKO" relationships.

(3)Comparing Smallatalk-80 with the other object-oriented programming languages,there is no difference in identification of objects,and the steps of programing by Smalltalk-80 are more numerous than in the other three languges.

1．まえがき

オブジェクト指向と云う考え方がプログラミング言語[1][2][3]ばかりでなく、マルチ・メディア・データベース、エキスパート・システム、自然言語理解、コンピュータ・グラフィックス等広い分野で研究開発されている。1970年代は構造化プログラミングが中心的課題であったように1980年代のそれはオブジェクト指向プログラミングであると云う人もいる[4]。オブジェクト指向言語は離散型シミュレーションを対象としたSIMULA67[5]に始まり、その影響を強く受けた[6] Smalltalk-80が、ビット・マップ・ディスプレイとマウスを用い、マルチ・ウインドウを駆使したワークステーション上に実現した。

本報告ではSmalltalk-80を用いて、酒類販売会社の在庫管理システムの作成の共通問題[7]を解く。先ず、オブジェクト指向の考え方と特長を要約し、オブジェクト指向設計法について述べる。次に、この設計法に従い在庫管理システムを設計し、それのSmalltalk-80による記述を説明する。最後に、作成されたプロトタイプの設計と実装の評価について述べ、他言語による記述との比較を行う。

2．オブジェクト指向設計法

プログラミング言語とソフトウェア設計法とは相互に関連がある。従来のCOBOL,FORTRAN 等の手続き型の言語にはツールとしての流れ図や、サブルーチンを単位とする機能分割（設計）法が合致していたと云える。オブジェクト指向プログラミングとオブジェクト指向設計法についてその考え方の要点を述べる。

（1）オブジェクト指向プログラミングの考え方

基本的な事項の要約をSmalltalk-80の考え方に従って述べる[8]。

①オブジェクト

対象とする世界のもの・ことのモデルであって、それは内部状態と操作手段をもつ。

②クラス

同一種類のオブジェクトの定義体。逆に、個々のオブジェクトはこのクラスを原型として生成される。このオブジェクトをまたクラスのインスタンスとも呼ぶ。クラスは階層を成すことができる。このとき、上位のものをスーパー・クラス、下位のものをサブクラスと云う。下位のクラスでは上位で定義された状態と操作を継承(inherit) されるし、上位で定義された操作を下位でのそれで書き替え(over write)も可能である。

③メッセージとメッセージ・パッシング

あるオブジェクトから別のオブジェクトにメッセージを送ることが従来の意味での原始的な処理スッテプに相当する。メッセージを受信したオブジェクトは該当する操作手段を起動する。この操作は受信オブジェクトのクラスでメソッドとして定義されている。これまでの処理・計算はオブジェクト間で次々とメッセージを交信して行くこと、つまり、それがメッセージ・パッシングである。

（2）オブジェクト指向プログラミングの特長

オブジェクト指向プログラミングの特長は、①カプセル化（encapsulation)と②継承(inheritance)である[9]。

①カプセル化（encapsulation)

メッセージの存在はオブジェクト間で周知されているが、そのメッセージによって起動される操作の仕様・実装は受信側のオブジェクト内に封じ込められている。また、オブジェクトの状態は外からは不可視であり、変更も不可である。オブジェクトのもつ情報は隠蔽されている。以上のことは一つの機能が一つのモジュールに凝集されていることであり、その機能の変更の影響を当該オブジェクトに止めており、システムの保守性の向上をもたらしている。

②継承（inheritance)

クラスは階層を成し、下位のクラスでは上位で定義されている操作と状態を利用できる。このことはソフトウェアを上位概念の部品から下位概念のそれへと段階的に作成し、目的に応じて組み合

わせることにより、システムを組み上げて行くことを可能にする。このことからオブジェクト指向プログラミングは差分プログラミングであるとも云われる。また、各種ソフトウェア部品を組み合わせてプロトタイピングする具体的な方法にもなっている。

（３）オブジェクト指向設計法の手順

オブジェクト指向設計法は次の手順に従う[4]。

①オブジェクトとその属性の選定

対象とする世界（問題空間）におけるオブジェクトを選定し、それらの役割を理解することが重要である。選定の基準の一つは、問題空間に存在する『名詞』がオブジェクトの候補となる。在庫管理システムのような事務処理では、帳票、台帳、ファイル等のデータの集まりとそれらのアクセッス・処理機能をオブジェクトとすることが分かり易いモデル化となる。

このステップは次の②と③と一緒にして従来のモジュール分割設計法[10]におけるサブシステム、機能階層への分割、つまり、モジュールへの分割に相当する。オブジェクトの選定の良否はこのスッテプだけでは決まらず、以降に続くスッテプの結果によって判定される。

②オブジェクトの操作の抽出

このスッテプでは分割された各々のオブジェクト又はオブジェクトのクラスの操作を決める。つまり、オブジェクトがもつメソッドの名前を決める。一つのオブジェクトに注目して、それ自身が成す操作と他のオブジェクトに成す操作を明確にする。この操作の抽出を、対象とするオブジェクト全体について行う。操作の抽出の基準は問題空間での『動詞』が候補となる。またこのステップでは操作間の時間的順序関係の適性化を図ることも必要である。

③オブジェクト間の可視性の確立

このステップではスコープの観点から各オブジェクト間での可視性／不可視性を確立する。これは対象とする世界のモデルと、その実現としてのオブジェクト間の関係を確認することである。また、クラス階層も決める。

④インタフェースの確立

このステップでは使用言語を踏まえて、モジュールの外部仕様を作成する。これは前ステップでのオブジェクトによるモデルの具体化である。

⑤オブジェクトの実装

このステップでは各オブジェクトに相応しい表現を選択し、前ステップでのインタフェースを実装する。この過程では分解と合成が伴う。

３．在庫管理システムの設計

２章で述べたオブジェクト指向設計法に従い在庫管理システムを設計する。

（１）オブジェクトの選定

オブジェクトの候補は、台帳、リスト、ファイル等のデータとそれらに付随する処理機能を単位として抽出・選定する。図１に選定したオブジェクトとメッセージを示す。長方形がオブジェクトを、〔　〕内にデータを示す。

①　受付管理オブジェクト

受付情報を管理するもので、外部機能仕様上は受付係とデータの授受をするユーザ・インタフェースの部分である。

②　在庫管理オブジェクト

酒の銘柄別に在庫数量を管理するもので、顧客からの注文に対して在庫の有無を回答する。

③　在庫不足管理オブジェクト

顧客からの注文に対して在庫無しとなったものを記録しておき、後日の入荷時に配送手配の指示を行うものである。

④　コンテナ在庫管理オブジェクト

コンテナ単位に在庫を管理し、顧客の注文に対してどのコンテナに入っている酒を配送するかを

決定する。また、空になったコンテナの搬出指示も行う。

（２）オブジェクトの操作の抽出

各オブジェクト間の操作（メッセージ）を図１の矢印で示している。

（３）オブジェクト間の可視性の確立

この段階では各オブジェクトがもつ変数を広域的／局所的か、クラス間に階層を設けるか否かを判断する。原則的には変数は局所的で、クラス間はフラットである。

（４）インタフェースの確立

（１）から（３）までのステップでモジュールへの分割は終わったので、個々のオブジェクトの操作の外部機能仕様を作成するステップである。操作はクラスのメソッドとして実現されるが、Smalltalk-80でのメソッド（原始的メソッド）と、ある機能を一連のメソッドで表現するメソッド・チェインとを区別する。図２に例として受付管理オブジェクトの操作の仕様記述（メソッド・チェイン）を示す。

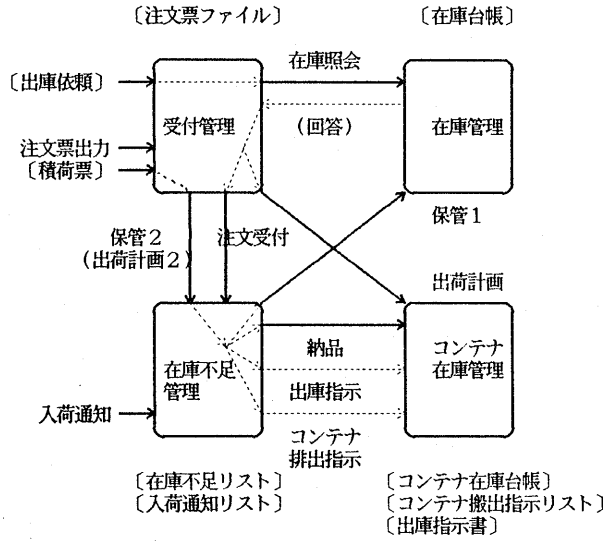

図　１　在庫管理システムのオブジェクト

［出庫依頼］
① 出庫依頼票（注文票）を読み込む。
② 注文品名と数量に対して、在庫を照会する（在庫照会）。
③ 在庫有りの場合には、出庫計画を決定する（出荷計画）。
④ 回答メッセージを出力する。

［積荷票］
① 積荷票を読み込む。
② コンテナ在庫台帳に記帳する（納品）。
③ 積荷票により、在庫台帳を更新する（保管１）。
④ 積荷票により、在庫不足リストを更新する（保管２）。

図　２　受付管理オブジェクトの操作の仕様

（５）オブジェクトの実装

オブジェクトの実装方については次の章に示す。

４，在庫管理システムの記述

（１）クラスの記述

Smalltalk-80によるクラス宣言は次の３つから構成されている。

① クラス名
② インスタンスが利用可能な変数
③ メソッドに応答するためにインスタンスが使用するメソッド

（２）ＭＶＣユーザ・インタフェース

Smalltalk-80はモデル（Model）－ビュー（View）－コントローラ（Controller）と云う３つのシステム・クラス（モデルはアプリケーション）から成るユーザ・インタフェースを提供している。在庫管理システムでユーザ・インタフェースと密接に関係するのは受付管理オブジェクトであり、それを図３に示す。ビューがユーザへの表示を、コントローラが入力を司る。モデル（受付管理オブジェクト）では出庫依頼（注文票）と積荷票が入力された後の処理を記述する。

（３）データ構造

台帳、ファイル等のデータ構造はSmalltalk-80が提供するものを利用した。具体的にはスタックを実現できるクラスOrderedCollection のインスタンスとした。

（４）クラス階層

プロトタイプのクラス階層を図４に示す。クラス　在庫管理システムはシステム・クラスとアプリケーション・クラスを分離し、それを束ねたものであり、プロトタイプに共通的なオブジェクト

の生成、ファイルの生成等を行っているだけで、実質的なメソッドはもたない。

5．プロトタイプの評価

（1）オブジェクトの選択

オブジェクト指向では問題空間での何をオブジェクトとするかが中心的な事柄である。共通問題は事務処理の典型である。その場合、帳票、ファイル、伝票等のデータ又はデータの集まりをオブジェクトとすることが、問題空間からの連続性から分かり易い設計／ソフトウェアと云える。分かり易さの条件として、

① 全体に対する部分（モジュール）の位置付けが明確であること。

② 機能が部分に閉じていて、完結していること。

③ 他人に理解し易いこと、

などが挙げられる。

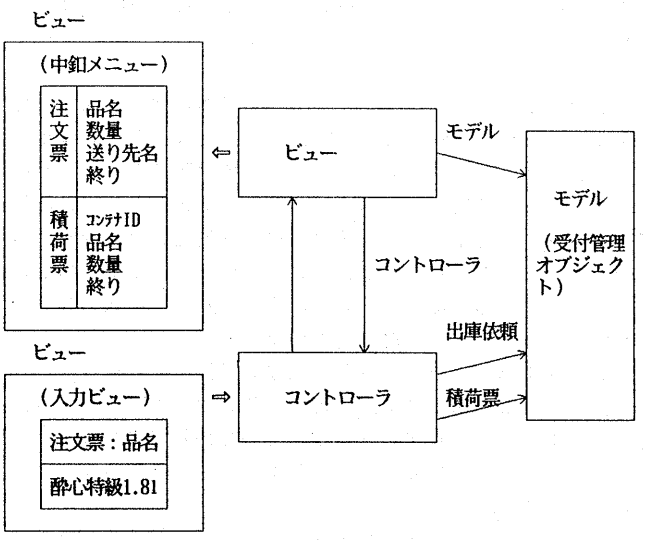

図 3 ＭＶＣユーザ・インタフェース

（2）クラス階層の選択

本報告では図４に示すクラス階層を選択した。クラス 在庫管理システムではオブジェクトとファイル類の初期条件設定を行っているが、これらは各オブジェクトを生成するところで個別に行ってもよいことである。ＭＶＣユーザ・インタフェースは Smalltalk-80の提供するものを利用している。従って本プロトタイプには本質的なクラス階層はないと云える。これは共通問題が簡単なシステムであって、クラス階層を利用するまでのメリットを含んでいないことである。スーパー・クラスとサブクラスの関係は is-a 関係を、クラスとオブジェクト間の関係は AKO(A Kind Of)関係を提供する。問題の基本的な構造にこのような関係を含む場合にクラス階層が適合する。フレーム・モデルに基づく知識をクラス階層で表現することは周知の適用例である。また、Smalltalk-80のシステム・クラスでは随所にクラス階層を活用している。

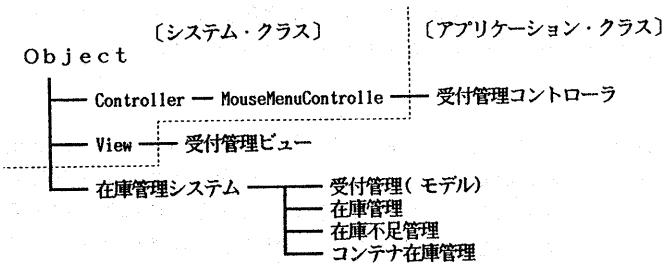

図 4 在庫管理システムのクラス階層

(3) カプセル化

Smalltalk-80はデータ抽象化を包含し、強化したオブジェクト指向言語である。必要なもの以外他のオブジェクトから見せないようにすることはモジュール間の独立性を高め、モジュールの変更を局所化している。本稿のプロトタイプでは、ファイル類はオブジェクト内のクラス変数とし、オブジェクト名を広域変数とし、他は各オブジェクト内の局所変数としている。ＭＶＣインタフェースのビュー(View)とコントローラ(Controller)はシステム・クラスを利用しているので、アプリケーション・モデルを１つのクラスにまとめることはできない。システム・クラスとアプリケーション・クラスを区別すると云う考え方がとりにくい。従来の言語では基本命令とそれによるモジュールの区別が明確であるのに対し、Smalltalk-80にはそれが無く、システム・クラスは全てソフトウ

ェア部品であって、ワークステーション上のSmalltalk-80の環境は１つのアプリケーションの全空間である。

（４）タイプ

データ抽象型言語ではタイプ・チェック機能が重視されていて、コンパイル時にそれが行われるものが多い。しかし、Smalltalk-80はタイプと云う考え方は無く、コンパイル時にチェックがされないが、実行時にキーワード・メッセージの引数間のチェックはされる。Smalltalk-80でのコーディング／デバッグでは、コンパイルは容易にできても、実行時に見つかるバグは多く、そのデバッグがし難い嫌いがある。Ａｄａ等の言語では安全性を重視し、Smalltalk-80は柔軟性を重視していると云える。このこととシステム・クラスのソフトウェア部品の活用により、Smalltalk-80はプロトタイピングを目指したシステムであることが容易に窺われる。

（５）他のオブジェクト指向言語との比較

オブジェクト指向言語による共通問題の記述にはABCL[1]、Concurrent Prolog[2]とClu[3]がある。これらと本稿での結果を比較すると次のとおりである。

①　オブジェクトを選定した後は、その外部機能の仕様決定に専念でき、情報と思考の分散化が実現できることは共通的である。これはオブジェクトのカプセル化による直接的な結果であると云える。

②　何をオブジェクトと考えるかにより、選定結果に差異が生じる。問題空間に存在するものそのものをオブジェクトに選定するもの[3]から、物やデータとその管理人を結合した複合体と捉えるもの[1]まで若干の幅はある。

③　プログラムの記述は言語によって当然違ってくる。CPでは92と143 ステップ、Clu では約300 行と報告されている。Smalltalk-80では約 400行であった。

６．あとがき

本報告ではオブジェクト指向言語の一つであるSmalltalk-80を用いて、酒類販売会社の在庫管理システムの作成の共通問題[6]を解いた。先ず、オブジェクト指向の考え方と特長を要約し、オブジェクト指向設計法について述べ、次に、この設計法に従い在庫管理システムを設計し、それのSmalltalk-80による記述を試みた。その結果、作成されたプロトタイプの設計と実装について次の結果を得た。

①　共通問題のような事務処理では、帳票、ファイル、伝票等のデータの集まりとその処理機能とをオブジェクトとすることが分かり易い設計／ソフトウェアである。

②　在庫管理システムでは本質的に有効なクラス階層は無かったが、Smalltalk-80がもつ is-a 関係と AKO関係を基本構造として含むアプリケーションで有効である。

③　Smalltalk-80と他のオブジェクト指向言語による記述では、オブジェクトの選定では差異があり、ステップ数ではSmalltalk-80が大きくCPが最小であった。

最後に、本報告を纏めるに際して（財）鉄道総合技術研究所　野末尚次氏には終始有益な討論をして頂いた。ここに記して深甚なる感謝の意を表明します。

参考文献

(1) 柴山悦哉ほか:"並列オブジェクト指向言語ABCLによる在庫管理システムの記述",情報処理 Vol.26 No.5,pp.460/468,(1985.5).
(2) 大木優ほか:"論理型並列プログラミング言語Concurrent Prolog による在庫管理システムの記述",情報処理 Vol.26 No.5,pp.469/477,(1985.5).
(3) 久野靖:"データ抽象向けプログラム設計技法",情報処理 Vol.26 No.11,pp.1310/1318,(1985.11).
(4) G.Booch:"Object-Oriented Development",IEEE Trans.on SE VOL.SE-12 NO.2,pp.211/221,(1986.2).
(5) M.Todoriki et al.:"A Hierachical Model Structuring of Computer Systems by SIMULA 67", 1977 SCSC(Summer Computer Simulation Conference),pp.795/797,(1977).
(6) A.Goldberg et al.:" Smalltalk-80 The Language and its Implementation", Addison-Wesely,(1983).
(7) 山崎利治:"共通問題によるプログラム設計技法解説",情報処理 Vol.25 No.9,p.734,(1984.9).
(8) A.Goldbergほか(相磯秀夫監訳):" Smalltalk-80 ―言語詳説―",オーム社,(1987.8).
(9) B.J.Cox:"Object Oriented Programming An Evolutionary Approach",Addison-Wesely,(1986).
(10)花田収悦編:"日科技連ソフトウェア品質管理シリーズ 第２巻 ソフトウェア仕様化と設計",日科技連,(1986.8).