教材用注文処理システムの MS-Accessによる開発とその利用

片　山　益　男

要約

既発表の業務処理をMS-Accessを用いて再開発し，その内容と利用方法について説明した。

1．はじめに

このテーマに取り組み始めてから20年近くになる。当初は汎用大型機によるTSSシステムの環境下で利用することを前提とした教材であった（COBOL使用)。その後教育環境がパソコン利用の情報演習室という形態に変化し，それに合わせて，教材もLevel Ⅱ COBOLを用いたものに再開発した。

パソコンのOSの進化とともに，パソコン用COBOLのバージョンもアップし，それまで無料で利用できていたCOBOLのruntime routinesが有料でしか利用できないことになった。そのための追加費用は，多数のパソコン台数では百万円のオーダーとなり，とても容易に実現できるものではないので，追加費用を必要としない（他の目的で導入されているMS-OFFICEに含まれる）MS-ACCESSを用いたシステムの開発に着手した。

この報告では，イベント駆動型ソフトを用いたシステム開発の体験の紹介と，そのシステム仕様をどのように記述すればよいのかという問題，さらに教育への利用方法などを説明する。

2．システムの概要

開発したシステムの機能はこれまでのものと全く同一である［1,2]。すなわち図１に示すように，常備品については，受注，出荷，売上計上，請求，入金，および補充発注と受入検収の各業務ステップがボタンのクリックで実行できる。注文品については仕入先への発注，受入検収業務が追加される。

業務に用いる共用ファイル（データベース）の設計も基本的にはこれまでと同一である。変更したのは，１件の受注という記録と，そこに含まれる複数商品の明細を分離したことである。MS-ACCESSではファイルの代わりにテーブルという用語を用いているので，以降ではテーブル

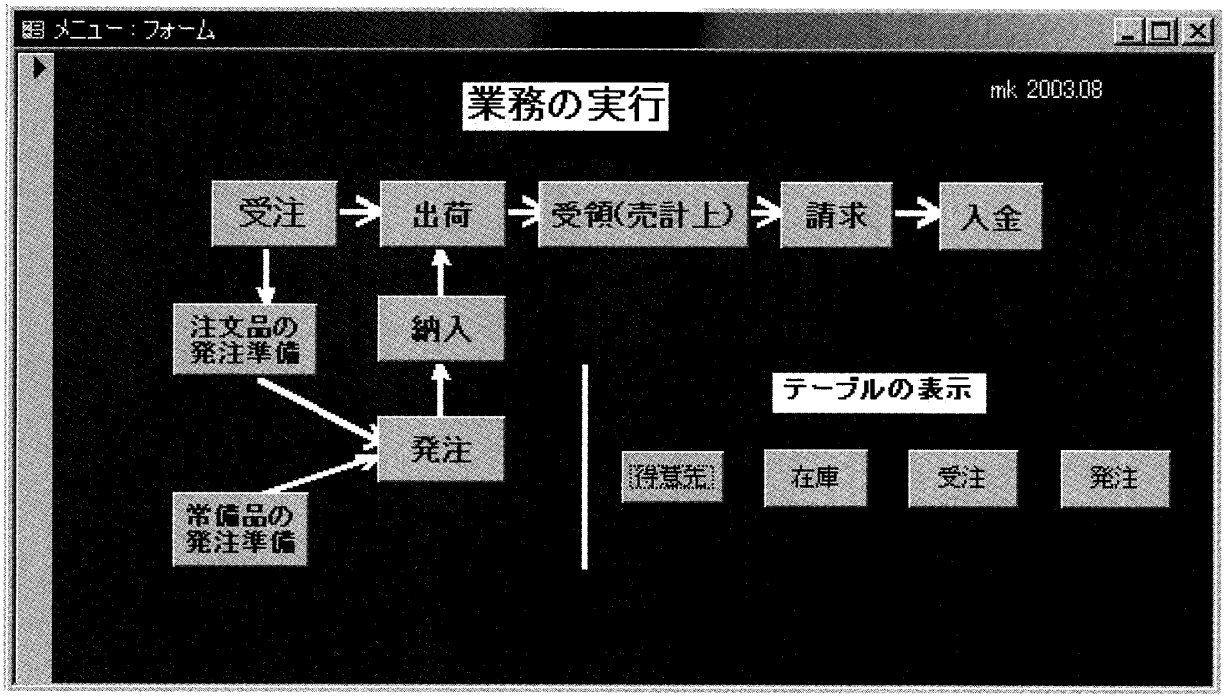


図1

という用語を用いることにする。使用したテーブルと，そのリレーションシップ設定を図2に示す（太字の項目は主キー設定した項目である）。テーブル名は左から順に，得意先t，受注t，受注明細t，在庫t，品名t，発注t，仕入先tとなっている。

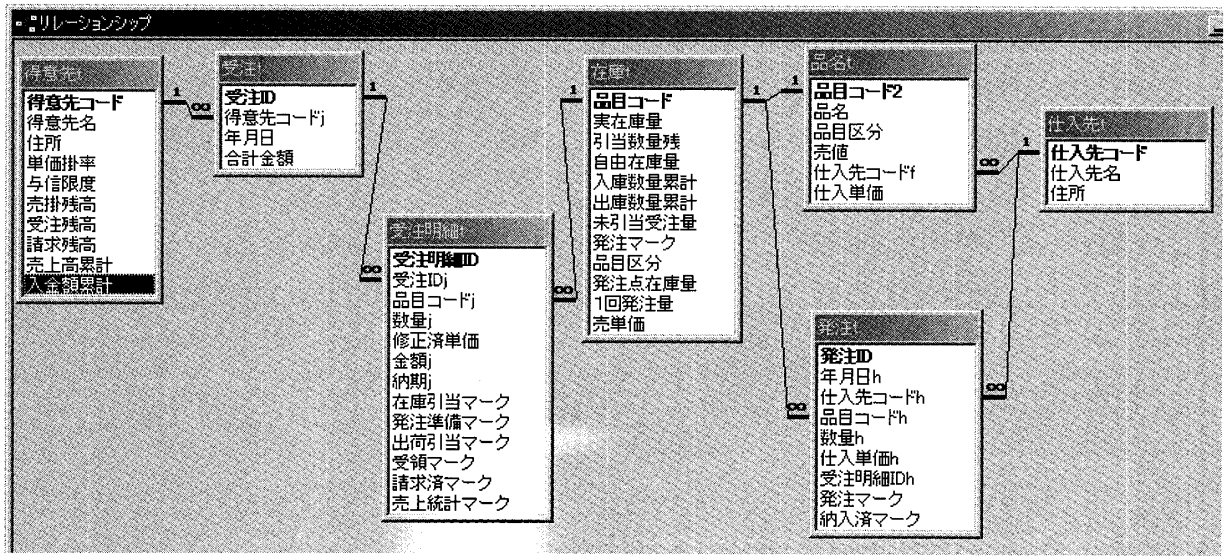
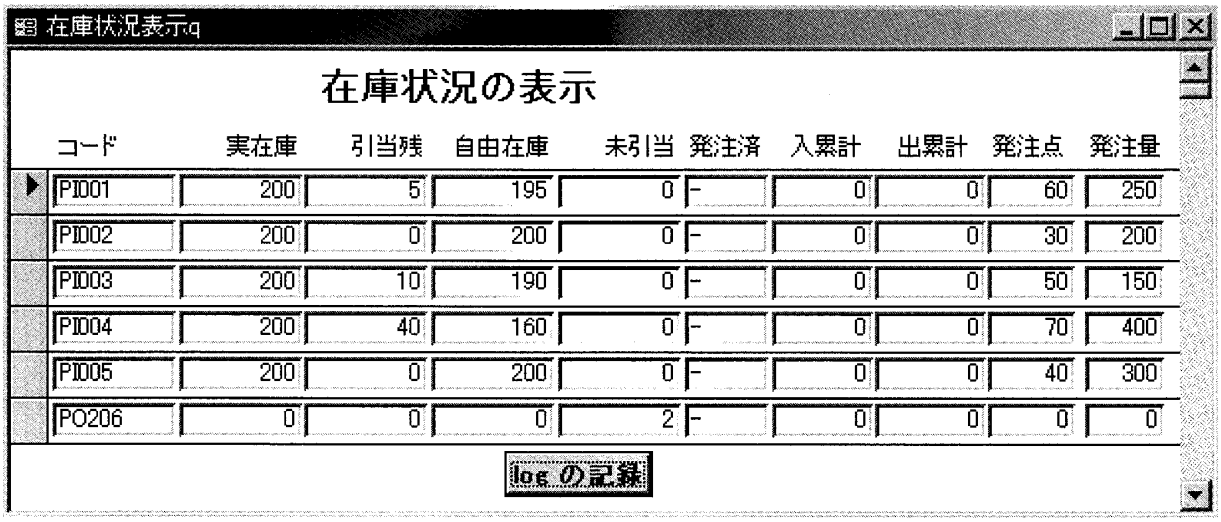


図2

各テーブルの主要項目の値を一覧リストとして，いつでも表示できるように，図1の右下部分に表示ボタンを配置した。一覧リストの一例を図3に示す。

システムの利用経過は，従来通りすべてログに記録する方式とした。学生のレポート提出にログのプリントアウトを利用できる。ただしテーブル内容の一覧リストを表示させたものをすべて



コード	実在庫	引当残	自由在庫	未引当	発注済	入累計	出累計	発注点	発注量
PI001	200	5	195	0	-	0	0	60	250
PI002	200	0	200	0	-	0	0	30	200
PI003	200	10	190	0	-	0	0	50	150
PI004	200	40	160	0	-	0	0	70	400
PI005	200	0	200	0	-	0	0	40	300
P0206	0	0	0	2	-	0	0	0	0

図3

記録するとログの分量が大きくなる可能性があるので，「Logの記録」ボタン（図3）を押したときだけ記録することとした。ログの例示をリスト1に示す。

開発したシステムの内容については後半で説明することとし，次章では本システムの利用例を説明する。

リスト1

```
***2003-09-28     10:52:33

***  受注処理開始
明細ID  1          C111          受注ID  1          品目コード PI001    数量  10
明細ID  2          C111          受注ID  1          品目コード P0206    数量  5

***  得意先状況
コード   受注残高     売掛残高      請求残      売上累計      入金累計
C111        8,740            0           0           0           0
C112            0            0           0           0           0
C113            0            0           0           0           0

***  在庫状況
コード   実在庫   引当残   自由在庫   未引当   発注済   入累計   出累計
PI001      200       10        190        0   -          0        0
PI002      200        0        200        0   -          0        0
PI003      200        0        200        0   -          0        0
PI004      200        0        200        0   -          0        0
PI005      200        0        200        0   -          0        0
P0206        0        0          0        5   -          0        0
```

*** 受注状況

明細ID	受注ID	得意コード	品目コード	数量	発注	在庫	出荷	受領	請求
1	1	C111	PI001	10		H			
2	1	C111	P0206	5					

*** 出荷処理開始
処理件数 1

*** 在庫状況

コード	実在庫	引当残	自由在庫	未引当	発注済	入累計	出累計
PI001	190	0	190	0	-	0	10
PI002	200	0	200	0	-	0	0
PI003	200	0	200	0	-	0	0
PI004	200	0	200	0	-	0	0
PI005	200	0	200	0	-	0	0
P0206	0	0	0	5	-	0	0

*** 受領処理開始
受注明細ID 1 受領処理

*** 得意先状況

コード	受注残高	売掛残高	請求残	売上累計	入金累計
C111	7,600	1,140	0	1,140	0
C112	0	0	0	0	0

*** 請求書作成処理開始
処理件数 1

*** 受注状況

明細ID	受注ID	得意コード	品目コード	数量	発注	在庫	出荷	受領	請求
1	1	C111	PI001	10		H	S	A	X
2	1	C111	P0206	5					

*** 入金処理開始
C111 500 入金処理

*** 得意先状況

コード	受注残高	売掛残高	請求残	売上累計	入金累計
C111	7,600	640	640	1,140	500
C112	0	0	0	0	0

*** 注文品発注準備処理開始
処理件数 1

*** 発注状況
発注ID　品目コード　数量　発注　納入　受注明細ID
1　　P0206　　5　　2

*** 発注処理開始
処理件数　1

*** 納入処理開始
発注ID　1　納入処理

*** 在庫状況

コード	実在庫	引当残	自由在庫	未引当	発注済	入累計	出累計
PI001	190	0	190	0	-	0	10
PI002	200	0	200	0	-	0	0
PI003	200	0	200	0	-	0	0
PI004	200	0	200	0	-	0	0
PI005	200	0	200	0	-	0	0
P0206	5	5	0	0	-	5	0

*** 受注状況

明細ID	受注ID	得意コード	品目コード	数量	発注	在庫	出荷	受領	請求
1	1	C111	PI001	10		H	S	A	X
2	1	C111	P0206	5	X	H			

3．教育への利用の一例

筆者は本システムを授業で使用しているが，その目的はブラックボックスとなっているコンピュータ化システムの内部を少しでも理解させ，コンピュータにあやつられるワーカーに止まらないために，どのように取り組めばよいかの手がかりを与えることである。

具体的には，コンピュータ化システムの主要な構成要素であるデータベースとプログラムを対象とし，各業務ステップの実行により，どのようなデータ項目が変化していくのか，プログラムの中にどのような業務規則が記述されているのかを演習を通じて確認させていくという方法を取っている。

そのために次のようなレポート課題を出題している（内容は毎年多少変化)。

3.1　課題1　業務によるデータ内容の変化

コンピュータ利用注文処理の事例システムを用いて常備品の注文処理を実行するとき，各業務ステップの実行により4種の共用ファイル（得意先，在庫，受注，発注テーブル）の内容（各データ項目の値)がどのように変化していくかを調べ，その結果を次のような形式の表にまとめよ。

業務＼テーブル	得意先	在庫	受注	発注
受注	変化部分を箇条書きで説明。複数行になることが多い。	同左	同左	同左
出荷	同上			
売上計上	同上			
請求	同上			
入金	同上			
常備品発注準備	同上			
発注	同上			
納入	同上			

記入例

業務＼テーブル	得意先	在庫	受注	発注
受注	・該当得意先の「受注残額」が受注金額だけ増加	・「自由在庫」が受注数量だけ減少 ・該当品目の「引当数量」が受注数量だけ増加	・新レコードが追加される	
出荷				

レポート課題の目的：

・コンピュータ利用システムの操作方法に慣れること。

・コンピュータ利用システムにおける「データ」の役割を理解すること。

(手作業システムにおける台帳・伝票の綴りと同じ内容の記録＋注文の処理進行段階を示す情報の記録が必要となる。)

3.2　課題2　プログラムに記述されている業務規則の確認

顧客が常備品を注文してきた場合，与信残額の不足，自由在庫の不足などから，すぐに出荷できない場合が発生する。それらの場合，受注や出荷が可能な状態にするには，どのような業務処理が必要かについて，具体的にどのような規則がプログラムされているのかを演習を通じて確認し，各々の要点を箇条書きで説明せよ。

1）どのような場合に「与信限度超過」と判定されるのか。その状態は，どのような時に解消されるのか。

2）「常備品の発注準備」はどのようなときに実行可能となるのか。その時の発注量は，どのような規則で決められているのか。

3）発注した商品が「納入」されたとき，どのようなデータ項目の値が変化するか，各々のデータ項目の変更はどのような意味を持っているのか（なぜ変更が必要になるのか)。

4）ある商品の自由在庫が50個のとき，その商品への注文数量が100個の注文が来ると出荷できないが，その後で，数量5個の注文が新しく入ったとき，後の注文の出荷はどうなるのか。

5）自由在庫より大きな注文数量の注文がきて．「常備品の発注準備」処理をすませたが，まだ納入されていないときに，新しく自由在庫を超える注文を受けたとき，再度「常備品の発注準備」処理を行うことができるか。

3.3 効果，その他

情報処理により果たすべき機能（業務内容）は基本的に同一だが，処理方法が「データ」と「プログラム」の2要素を中心とする方法になるというコンピュータ化システムの特徴を理解してもらうのが目的なので，筆者は数回の授業でよいのではないかと考えているが，実際には10回程度の時間を費やしている。もっとスピードを上げようとすると，学生から早すぎて付いて行けないという苦情が非常に多くなる。

この教材を使うことについての学生からの反応は，この教材のために多くの時間を割くことについての苦情は全くなく，「注文処理業務というものがよく理解でき非常に役立った。コンピュータ利用の必要性についても再認識できた。ただし業務内容については，前期に取り組んだ，手作業による注文処理手続の課題がなければほとんど理解できなかっただろう。」というのが代表的な意見である。

今回開発したシステムをすでに3年間使用してきたが，それまで使用していたCOBOLベースのシステムと比べてGUI（graphical user interface）が大きく向上したため，操作方法に関する苦情はほとんど無くなった。

開発にあたり，データベースの初期化をどうすればよいかということで頭を悩ましたが，コロンブスの卵の喩えのように，本システムを起動させるのにバッチファイルを用い，まず別の場所に保管している新規ファイルを所定の場所に上書きコピーし，コピーしたファイルを起動させることで，起動するたびに初期状態からシステムを使用できるようにした。

ちなみに担当科目（情報システム論）の受講者数は10年余りの間，毎年200～300人の範囲で，この教材を使う間160台のパソコンを設置した教室を使用している。教材使用のための特別なアシスタントは置いていない。

4．イベント駆動型システムの設計と記述

COBOLのような手続型言語を用いるときは，データベースの定義とプログラム・リストでシステムを記述することができる。しかしMS-ACCESSでFORMを利用するような，GUI重視のシステムを設計し，それを記述することは非常に煩雑な事柄となる。

システム開発学習者の参考に資することを意図して．まず今回のシステムで最も複雑な「受注」業務ステップを取り上げ，筆者が開発過程で考えたことを説明する。

4.1　「受注」業務ステップの設計プロセス

受注業務では，ある得意先からの注文を受付け，記録することが骨子となるが，1回の注文には複数の商品が含まれ，受付過程において在庫の有無確認，与信限度の確認を行い，誤入力データの修正にも対応できること，という要件を想定した。

業務処理用の画面としては，多少の試行錯誤の上，図4の様式を用いることにした。これはMS ACCESSを多少学習した人なら容易に考えつく，フォームにサブフォームを入れ子にした様式である。入力データはテーブルに記録しなければならないが，作業の途中でデータ内容の改変が生じることがあるので，作業用のテーブルに入力データを記録し，1件の受注確定後に正規のテーブル（受注テーブルと受注明細テーブル）に追加記録するという方法を取ることにした。さらに，正式記録の時点では得意先テーブル，在庫テーブルの更新（受注残額の追加，引当数量の追加など）も必要になる。

画面設計にあたっては，どのようなデータをどのような順序で入力していくかということを想定しなければならない。日付データは自動入力とする（手で修正可）。

次は得意先の指定というのが自然な順序となる。そこで，コンボボックスを用い，得意先コードを選択すれば得意先名が自動的に表示されるという使い方が最も自然だと思われる。それを実

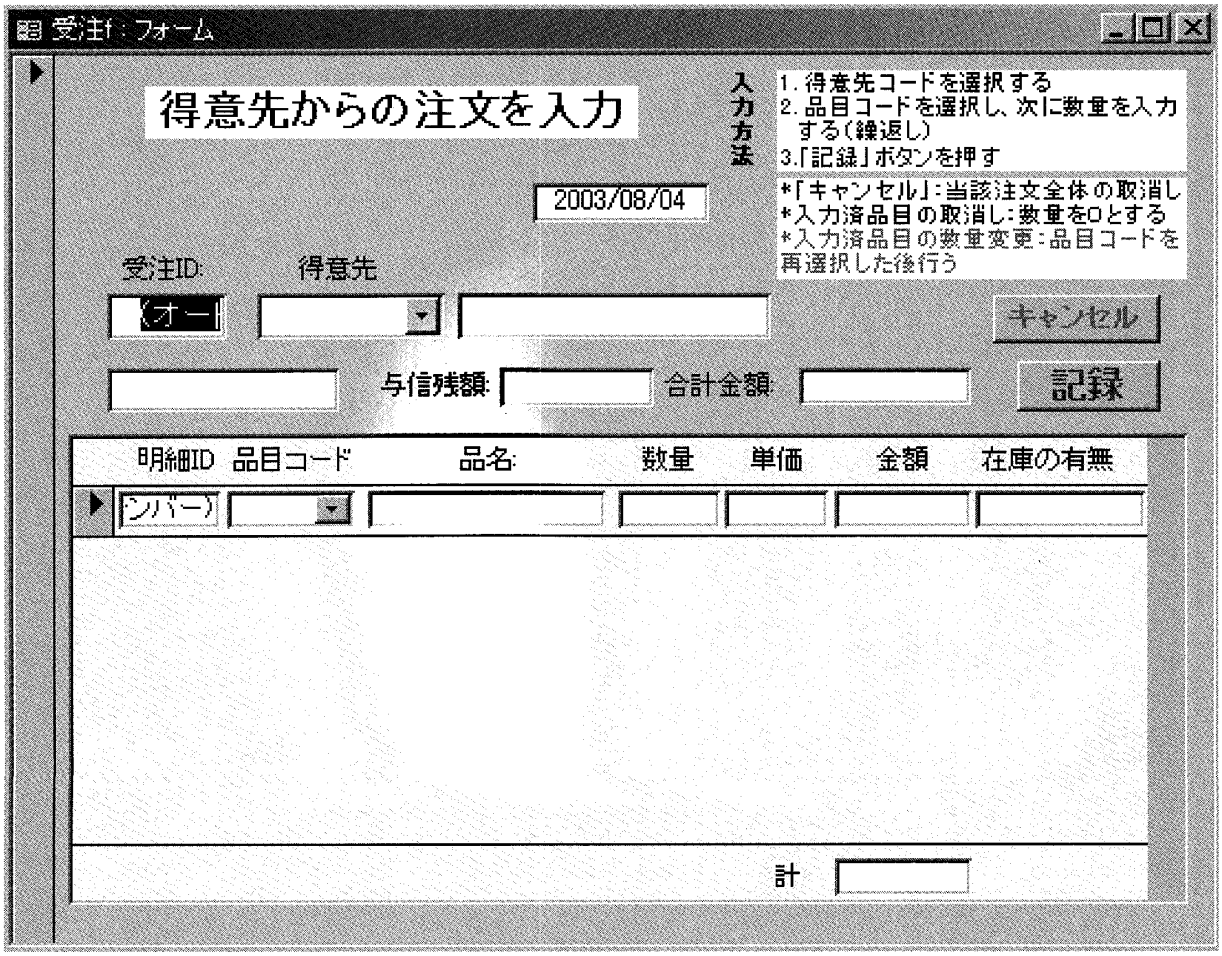


図4

現するには，「得意先コード」の「データ更新後」イベントというプロパティの場所に，選択したコードに対応した得意先名を得意先テーブルから検索し，名前を「得意先名」ボックスに表示するというスクリプトを記入しなければならない。後で，与信限度の確認という作業も必要になるので，得意先テーブルから該当レコードを検索したとき，ついでに与信残高のデータを作業変数に記録しておくのが効率的だと思い付く。

次は注文明細として，商品と数量の入力という順序になる。これは複数行にわたることが前提となる。得意先の場合と同様に，商品コードを指定（選択）すれば商品名が自動表示されるという対応が望ましい。そこで，「商品コード」の「データ更新後」イベントというプロパティの場所に，選択したコードに対応した商品名を品名テーブルから検索し，名前をテキストボックス「品名」に表示するというスクリプトを記入する。商品を指定すれば，次は数量を入力するというのが自然な順序であるが，「数量」欄にいちいちマウスを移動させてクリックし，それから数量を入力するというのはおっくうな作業なので，商品コードを指定したとき，商品名表示の後，マウスカーソルを「数量」欄に移動させるということもスクリプトに書き加えるのが親切である。

さて，数量を入力すると，その金額を計算して表示する必要がある。それは「数量」の「データ更新後」イベントというプロパティの場所に，単価×数量を計算し，計算結果を「金額」欄に表示するというスクリプトを記入する。この事例では，単価は商品毎の基準単価に得意先毎の単価掛率を掛けて計算する仕組みになっているので，単価計算に必要なデータを，得意先テーブル検索時と品名テーブル検索時に予め取得し，作業変数に記録しておくのが効率的である。

金額が計算されると，それにより与信限度超過という状態にならないかどうかを検討することができる。その検討と結果の表示も数量データ更新後イベントのスクリプトに追加しておくのが効率的と思い至る。

なお「数量のデータ更新後イベント」はどのような時点で発生するのかという疑問が生じ，実験の結果，「数量」欄にデータを入力し，マウスカーソルが他の場所に移ったときに発生することを知った（１文字のデータを入力する都度発生するのではない）。

このようにして，必要なだけの商品と数量を指定し，終了すれば，それらのデータを正式のテーブルに追加し，得意先テーブルなどの関連テーブルのデータ更新を行う。その作業を開始するために「記録」というボタンを設けている。記録ボタンに対応づけられているスクリプトには，記録追加の作業の前に，与信限度超過のチェック，作業テーブルのクリアなどを行う部分も含ませておかねばならない。

当初は受注画面を「閉じる」ボタンを作っており，そのボタンを押すことにより画面を閉じると同時に，オープンしているテーブルを閉じる命令も同時に実行するようにしていたが，初めて授業で使用したところ，Windowsで基本的な操作になっている画面右上の「X」印をクリックして画面を閉じる学生が非常に多いという事実に直面し，テーブルのクローズ命令を「X」印クリック・イベントの場所に記述するように変更した。このように，試行により設計変更した個所もいくつかあった。

受注業務ステップで必要な処理と，そのための画面設計を，以上のような試行錯誤的思考・実験過程を経て具体化し，図4の画面を作成した。

入力操作過程の例示を図5に示す。

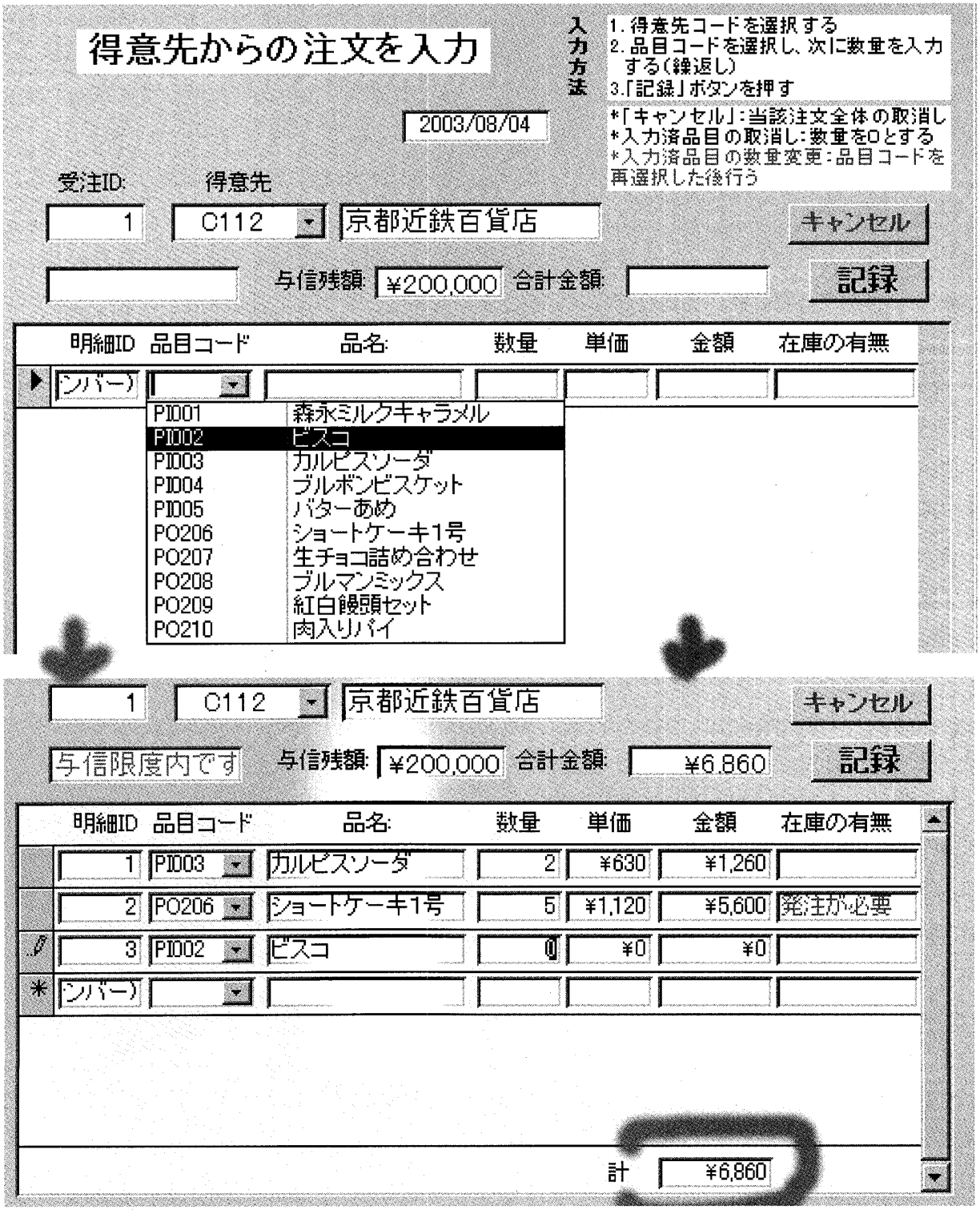


図5

4.2　システム仕様の記述様式

このような過程を経て（一応）完成させた画面と，処理内容をシステム仕様として，どのような方法で記述すればよいのか。記述の主目的を．後日システムの更新が必要となったとき，当該システムの内容を理解しやすい形で更新担当者に伝えることに置く。このような目的で作成された内容は，システムを学習させたい場合にも役立つ。他の目的，たとえばシステム操作を説明するための記述内容はかなり異なるものとなるが，ここでは取り上げない。

記録がないとシステム更新のとき，非常に困難な事態が生じることはすでに常識となっている。しかしイベント駆動型システムの仕様をどのように記述すればよいのか，まだ標準的な記述様式が確立されていないのではないかと思われる（勉強不足なので，誤りなら指摘いただきたい）。ここでは1つの試作を説明する。

イベント駆動型システムにおいて，設計された画面は容易に見ることができる。処理プログラムとしてのスクリプトも，スクリプト・リストをプリントアウトすれば，どのイベント発生のとき，どのような処理をするのかを知ることができる。しかしそれら2種類の記録だけでは，業務処理目的を達成するため，それらの要素がどのように関連づけられているのか，すなわち各々の時間的順序や存在理由の説明が欠如している。(MS-Accessを用いたシステムでは，さらにどのようなクェリーを作成しているのか，フォームはどのようなテーブル，クェリーと対応づけられているのかなどの説明も必要となる。）これらの点がイベント駆動型システムを説明する場合の鍵になるのではないかと考える。

イベント駆動型システムの（各ステップ／モジュールの）記述には少なくとも次のような項目を含めるべきであろう。

1）その業務ステップで処理すべき業務内容（処理目的）
2）処理のための道具立て（データベースの構造，使用する画面とその構成）
3）GUIで用いる各部品（画面の各構成要素）の使用順序と，それらに連結されているイベント処理の内容
4）エラー発生時の処理内容

システム仕様記述の内容は従来に増して複雑になり，その労力も膨大になるが，次の担当者への連絡として，必要最小限の内容を文書化しておかないと，後日多大な労力を費やさなければならなくなる。使用したソフト固有の条件から発生する事項（例えば使用するテーブルやクェリーの種類など）に関する説明は，次に使用するソフトが変更されれば不要となるので，開発システムの寿命と勘案して省略してもよいが，業務の処理規則の詳細（例えば与信限度のチェックの仕方，納入時に該当品目を選択する方法など）や利用時に必要となる工夫（例えば入力ミスへの対処方法としてワークファイルの利用，画面を閉じるときの留意点など）はできるだけ具体的に記述しておく必要がある。

目下，普及が進展しつつあるUMLを用いた記述方法については，十分に咀嚼していないので，

適否についての断定的な言及は差し控えるが，少なくとも今回対象としているシステムの記述方法として，最適な方法とは言えないのではないかと思われる。例えば「選択した品目の数量を入力したとき，これこれの内容を実行する」というような詳細記述にシークエンス図や状態チャート図を用いることは，いたずらに記録の作業量を増加させるだけのように感じる。

5．開発システムの内容例示

紙面の制約もあり，ここでは今回開発したシステムの，イベント駆動型の特徴が見られる個所を中心に，いくつかの部分を紹介する。それらの前提として，図1，2で示した以外の(MS-Access

図6　ワークテーブル

での）使用部品（ワークテーブル，クェリー，フォーム，レポート）を図6，7に示す。図6のテーブル名は左上から順に，jmwork，出荷t，jworkt，発注wk，請求tである。

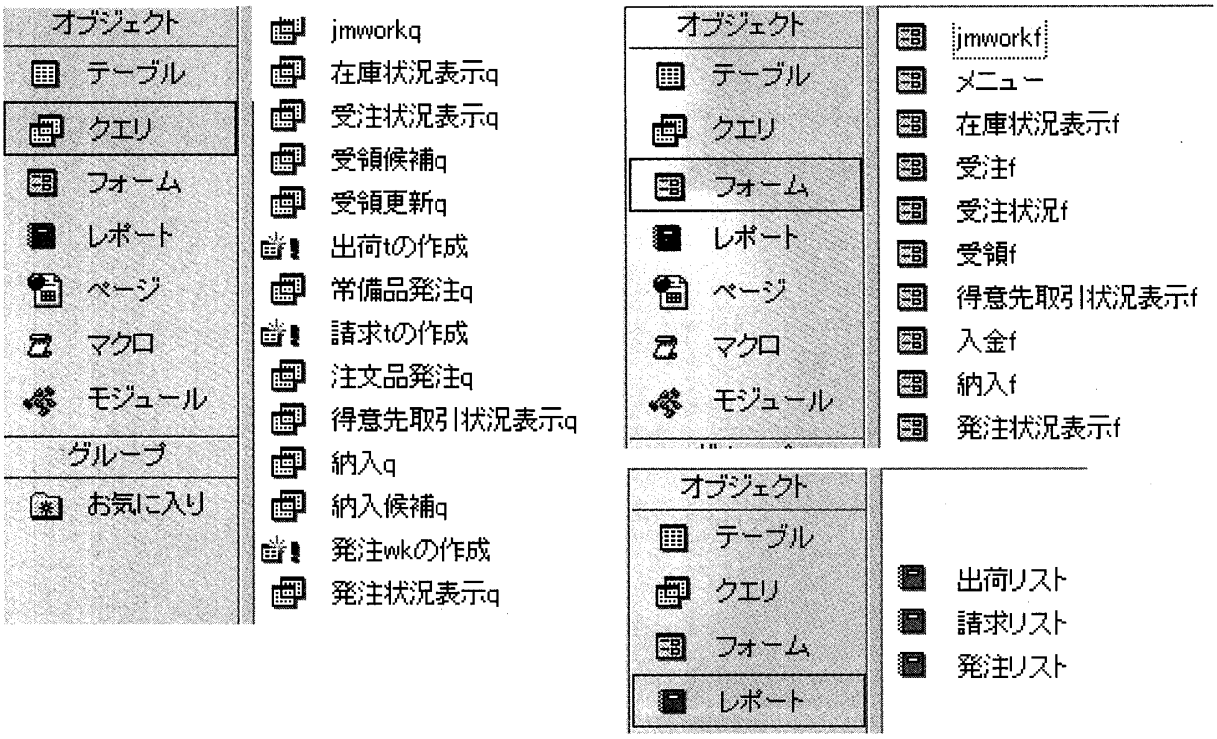


図7

5.1　受注

この業務処理は受注内容を記録することであるが，そのGUIにはフォーム「受注f」を用いている。主フォームはテーブル「得意先t」と対応づけている。品目一覧を示すサブフォームはテーブル「jmwork」などをもとに作成したクェリー「jmworkq」と対応づけている。

使用法の大部分はすでに4.1で説明したので，ここでは補足的な説明をする。

受注品目の数量を入力し終えたとき（マウスを他の場所でクリックしたとき）その金額計算，在庫の有無検討，与信残額の有無などのチェックを行うスクリプトが実行される。それは「数量」の「データ更新後」イベントに連結されている。

1組の受注内容を入力した後，その内容が正しければ「記録」ボタンをクリックするが，それによりリスト2のスクリプトが実行される。処理内容の詳細については既発表の文献に説明されているので，ここでは省略する。ファイル処理はすべてSQL文を書くだけなので非常に効率的である。

リスト2

```
'**記録ボタンを押したときの処理内容

Private Sub kirokucmd_Click()
    DoCmd.SetWarnings False
    Me.SetFocus
'サブフォーム：jmworkfの合計金額をメイン・フォーム：計金額の欄にコピーする。
    Me![計金額] = [jmworkf].[Form]![stotal]
    Me.Recalc
    Dim a As String
'今回の受注金額合計が与信残高を上回る場合，受注できないメッセージを表示し，
'  ワークテーブル：jmwork，jworktの内容をクリアする。
'  受注可能な場合，記録のためのモジュールを呼び出す。
    If Me![計金額] > Me![yoshinzan] Then
            a = MsgBox("与信限度を超過。記録できません。")
            Me.Requery
            DoCmd.RunSQL ("delete * from jmwork;")
            DoCmd.RunSQL ("delete * from jworkt;")
        Else
            Call kiroku
    End If
'
'受注フォームのメッセージ表示内容をクリアする。
    Me![msg2] = ""
End Sub

'**記録モジュール（サブルーティン）

Private Sub kiroku()
'
'受注金額が正の場合，受注テーブルに新しい注文のレコード
'（jworktの内容）を追加する。
    Dim a, str1, str2 As String
    Dim outv As Long
        str1 = "insert into 受注t select * from jworkt where 合計金額 > 0;"
        DoCmd.RunSQL (str1)
    '
'得意先テーブルの該当得意先コードのレコードについて，受注残額を更新する。
'
        Dim db1 As DAO.Database, rec1, rec2 As DAO.Recordset
        Set db1 = CurrentDb
        Set rec1 = db1.OpenRecordset("得意先t", dbOpenDynaset)
        rec1.FindFirst "得意先コード = '" & Me![tcodej] & "'"
        rec1.Edit
        rec1![受注残高] = rec1![受注残高] + Me![計金額]
```

```
        rec1.Update
        rec1.Close
    '
'
'受注明細のワークテーブルを用い，在庫テーブルの該当品目の内容を更新する。
'
        str1 = "update jmwork set 在庫引当マーク = ' ';"
        DoCmd.RunSQL (str1)
        Set rec1 = db1.OpenRecordset("jmwork", dbOpenDynaset)
        Set rec2 = db1.OpenRecordset("在庫t", dbOpenDynaset)
        rec1.MoveFirst
        Do Until rec1.EOF
            outv = rec1![数量j]
'
' ログ・ファイルに受注内容を書き出す。(次の２行)
'
            Print #1, "明細ID "; rec1![受注明細ID], Me![tcodej], "受注ID "; rec1![受注IDj], _
                    "品目ｺｰﾄﾞ "; rec1![品目コードj], "数量 "; outv
'
            rec2.FindFirst "品目コード = '" & rec1![品目コードj] & "'"
            rec2.Edit
            If outv > rec2![自由在庫量] Then
                    rec2![未引当受注量] = rec2![未引当受注量] + outv
                Else
                    rec2![引当数量残] = rec2![引当数量残] + outv
                    rec2![自由在庫量] = rec2![自由在庫量] - outv
                    rec1.Edit
                    rec1![在庫引当マーク] = "H"
                    rec1.Update
            End If
            rec2.Update
            rec1.MoveNext
        Loop
        rec1.Close
        rec2.Close
'
'受注明細ワークテーブルの内容を受注明細テーブルに追加し，その後
' ワークテーブルの内容をクリアする。
        str2 = "insert into 受注明細t select * from jmwork" _
        & " where 数量j <> 0;"
        DoCmd.RunSQL (str2)
        Me.Requery
        DoCmd.RunSQL ("delete * from jmwork;")
        DoCmd.RunSQL ("delete * from jworkt;")
End Sub
```

5.2　出荷，請求，常備品の発注準備，注文品の発注準備，発注

すべて自動処理されるが，処理内容については既文献にゆずる。プリントアウト用の帳票は，レポート・オブジェクトとして予め作成しておいた「出荷リスト」などをスクリプトから実行指示するだけなので，帳票様式制御のためのコーディングが不要となる。

5.3　売上計上

図8のごとく，「受注明細f」の該当レコード・リストから1つを選択するという操作だけで済むようにした。処理内容はコンボボックスの「選択後」イベントと関連づけられたスクリプトに記述されている。

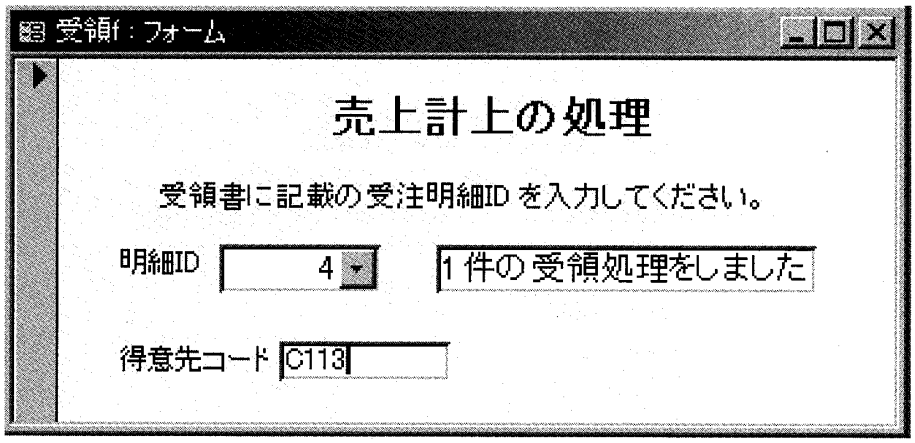


図8

5.4　（発注品の）納入

売上計上と同様に，該当レコードを選択するだけでよい（図9，10）。しかしその時の処理内容はかなり複雑である。そのスクリプトをリスト3に示す。

リスト3

```
Option Compare Database

Private Sub combl_AfterUpdate()
 On Error GoTo Err_combl_AfterUpdate

    Me![txtl] = ""
    '
    Dim strl, str2, str3, icode, a As String
    Dim kubun, m_id As Single
    Dim vol, cumvol As Long
    '
```

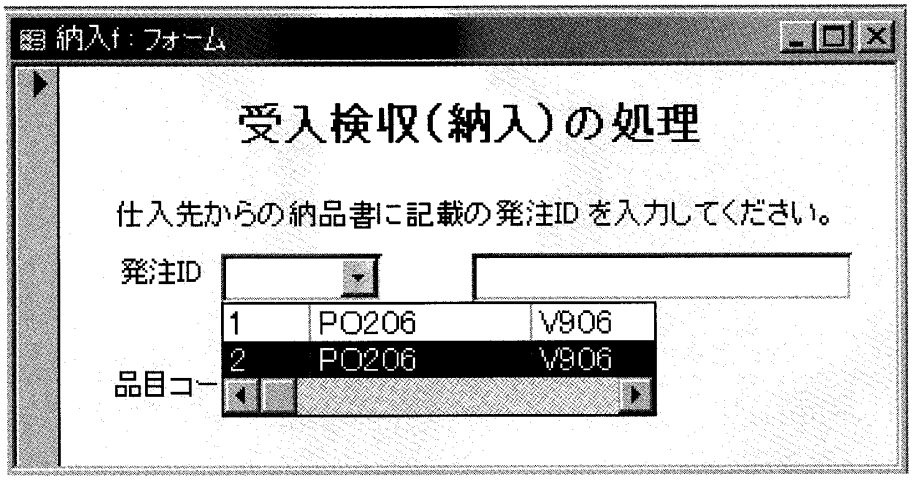


図 9

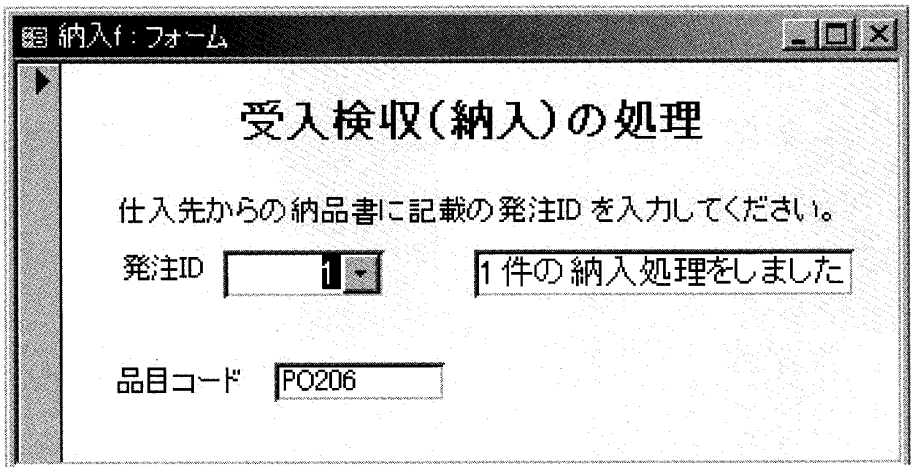


図10

```
Dim db1 As DAO.Database, rec1, rec2, rec3 As DAO.Recordset
Set db1 = CurrentDb
str1 = "select * from 納入q where 発注ID = " & Me![comb1] & " ;"
Set rec1 = db1.OpenRecordset(str1)
If rec1.EOF Then
        a = MsgBox("「発注ID」が条件に合いません。")
        Print #1, "発注ID "; Me![comb1]; " 条件に合わない"
        Close #1
    Exit Sub
End If
icode = rec1![品目コードh]
Me![icodef] = icode
kubun = rec1![品目区分]
m_id = rec1![受注明細IDh]
vol = rec1![数量h]
```

```
Print #1, "発注ID "; Me![combl]; " 納入処理"
'
rec1.Edit
rec1![納入済マーク] = "U"
rec1.Update
rec1.Close
'
If kubun = 1 Then
    str2 = "select * from 受注明細t where 受注明細ID = " _
           & m_id & ";"
    Set rec2 = db1.OpenRecordset(str2, dbOpenDynaset)
    rec2.Edit
    rec2![在庫引当マーク] = "H"
    rec2.Update
    rec2.Close
    '
    str3 = "select * from 在庫t where 品目コード = '" _
           & icode & "' ; "
    Set rec3 = db1.OpenRecordset(str3, dbOpenDynaset)
    rec3.Edit
    rec3![実在庫量] = rec3![実在庫量] + vol
    rec3![引当数量残] = rec3![引当数量残] + vol
    rec3![未引当受注量] = rec3![未引当受注量] - vol
    rec3![入庫数量累計] = rec3![入庫数量累計] + vol
    rec3.Update
    rec3.Close
  Else
    str2 = "select * from 受注明細t where 在庫引当マーク <> 'H' " _
      & " and 品目ｺｰﾄﾞj = '" & icode & "' order by 受注明細ID; "
    Set rec2 = db1.OpenRecordset(str2, dbOpenDynaset)
    cumvol = 0
    If rec2.EOF Then
      Else
        rec2.MoveFirst
        Do Until rec2.EOF
            If (rec2![数量j] + cumvol) <= vol Then
                cumvol = cumvol + rec2![数量j]
                rec2.Edit
                rec2![在庫引当マーク] = "H"
                rec2.Update
                rec2.MoveNext
              Else
                Exit Do
            End If
        Loop
    End If
```

```
        rec2.Close
        '
        str3 = "select * from 在庫t where 品目コード = '" _
               & icode & "' ; "
        Set rec3 = db1.OpenRecordset(str3, dbOpenDynaset)
        rec3.Edit
        rec3![実在庫量] = rec3![実在庫量] + vol
        rec3![引当数量残] = rec3![引当数量残] + cumvol
        rec3![未引当受注量] = rec3![未引当受注量] - cumvol
        rec3![自由在庫量] = rec3![自由在庫量] + vol - cumvol
        rec3![入庫数量累計] = rec3![入庫数量累計] + vol
        rec3![発注マーク] = "-"
        rec3.Update
        rec3.Close
    End If
    '
    Me![txt1] = "1件の納入処理をしました。"

Exit_combl_AfterUpdate:
    Exit Sub

Err_combl_AfterUpdate:
    MsgBox Err.Description
    a = MsgBox("error!")
    Resume Exit_combl_AfterUpdate

End Sub

Private Sub Form_Close()
    Close #1
End Sub
```

5.5　入金

図11のフォームを用いた。「得意先t」と対応づけている。得意先コードを選択すると得意先名と請求残額が表示される。入金額をキー入力し「処理の実行」ボタンを押すと実行される。

6．おわりに

継続したニーズがあるので，情報処理演習設備の更新に合わせるために今回のシステム開発を行い，ここでその内容を報告した。

GUIの向上から利用者には使いやすいシステムに改良されたが，開発側からは新しいソフト

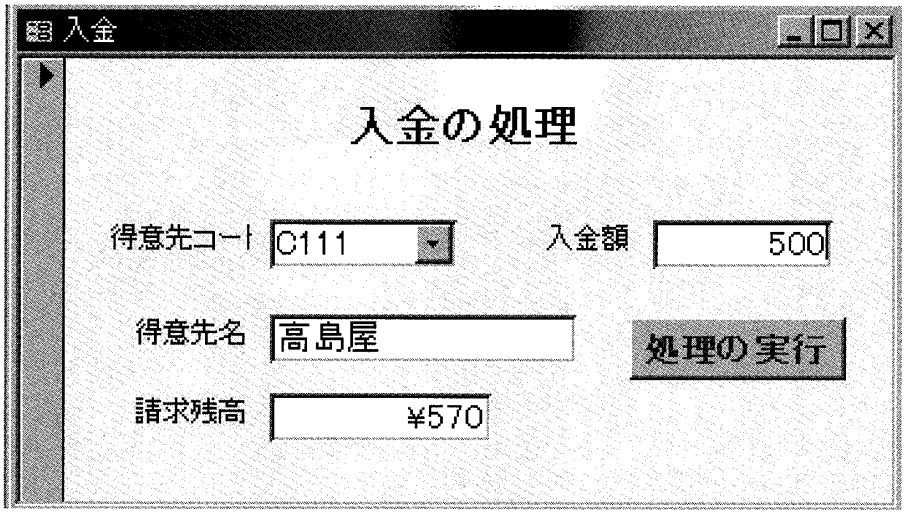


図11

(MS-Access) の使用法を新たに学習する必要があり，かなりの労力が必要になる。

報告にあたり，イベント駆動型ソフトの内容をどのように記述すればよいのか，1つの試行錯誤を試みた。すなわち，4.1で例示したように（ここでの記述は冗長度がすぎるが），処理目的を含んだ内容説明が重要だとする立場からの一主張である。

業務処理は，今日ではネットワークに接続されたデータベースを用いて行われることが多い。そこで教育においてもネットワーク経由のデータベース利用を体験させるべきではないかということを，開発段階では検討したが，学生各人が自己用のデータベースを操作する必要があり，ネットワーク制御の煩雑さ，開発工数の増大，利用時における応答時間の長さなどの短所を考えると，単純で操作性の高いスタンドアロン・システムの存在価値を改めて確認した。

ただし次の開発においては，webページをGUIとして用いるネットワーク利用が更に一般化するので，JavaScriptやFlash技術を採用したGUI画面の作成が必要になると思われる。

参照文献

1）拙稿，「教育用経営情報システム・モデルの開発」，大阪産業大学論集　社会科学編，84号，pp.37-68，1991

2）拙著，「経営システムと情報システム」付章3，中央経済社，2000