

Implementarea unui sistem informatic bancar

Prof.dr. Ioan ODAGESCU
Catedra de Informatica Economica, A.S.E. Bucuresti
Carmen TRUFASU
Banca UNIREA Bucuresti

În 'era informatională' datele sunt cel mai pretios 'activ' al unei întreprinderi. O societate bancară, în condițiile actuale ale pieței privind concurența extraordinară a bancilor străine care și-au deschis sucursale și filiale în România, nu este viabilă dacă sistemul informatic care operează nu este viabil.

Un sistem informatic viabil este definit de coerența datelor stocate, de promptitudinea cu care sistemul răspunde la cererile utilizatorilor, de calitatea interfeței cu utilizatorii neinformaticieni – funcționari bancari și de performanța tehnologiei utilizate la implementarea sistemului. Din această pseudodefiniție rezultă necesitatea unei foarte mari atenții acordate etapei de proiectare a sistemului.

Software-ul propus în acest articol reprezintă doar o infimă parte a unui astfel de sistem rezolvând problemele legate de acordarea, urmărirea și rambursarea creditelor. Articolul prezintă de asemenea facilitățile oferite de mediul BORLAND Delphi pentru dezvoltarea unui astfel de sistem.

Cuvinte cheie: sistem informatic, software, activitate bancară, Delphi.

1. Introducere

Pentru înțelegerea soluției de implementare propuse în acest articol este necesară prezentarea în linii mari a procesului de acordare și rambursare a creditului. Orice bancă desfășoară activitățile enumerate mai jos, adăugând și altele conform politicii de gestiune și creditare adoptată de banca respectivă:

- a) analizarea cererii de creditare și evaluarea situației financiare a solicitantului de credit;
- b) decizia de acordare/respingere a creditului pe baza analizei efectuate;
- c) rambursarea datoriilor aferente creditului conform graficului stabilit în contractul de credit și urmărirea derulării acestuia;
- d) stingerea relației de credit ca urmare a rambursării acestuia în întregime – la termen sau înainte de termen, cu penalizări sau ca urmare a executării silite a garanțiilor creditului.

Ca metode de programare și componente mai 'speciale' ale mediului de dezvoltare utilizat se vor prezenta: interfata dintre pro-

gram și baza de date, generarea de rapoarte sugestive utilizând componente ale bibliotecii VCL, integrarea în aplicație a obiectelor de tip ActiveX și folosirea unor controale deosebite oferite de biblioteca vizuală.

2. Structura modulului de creditare

Modulul destinat creditării este în directă interdependentă cu modulele destinate gestionării conturilor clienților, a dobânzilor și comisioanelor percepute de bancă și cu modulul destinat evidenței clienților bancii. Interacțiunea între aceste module se face atât la nivelul bazei de date – se folosesc aceleași tabele cât și la nivelul aplicației unde se fac validări de date – pentru menținerea formei normalizate a bazei de date, așa cum a fost proiectată.

Prima etapă în procesul creditării este evaluarea situației financiare a clientului solicitant de credit (în cazul în care clientul este persoana juridică), a activității firmei și a istoricului relațiilor cu banca.

Pentru evaluarea punctajului – *Credit Scoring*, pe baza căruia se fundamentează pro-

punerea de acceptare/refuz la creditare, aplicata ofera o fereastra speciala - în limbajul

Delphi cunoscuta sub numele de Form, numita ScorForm:

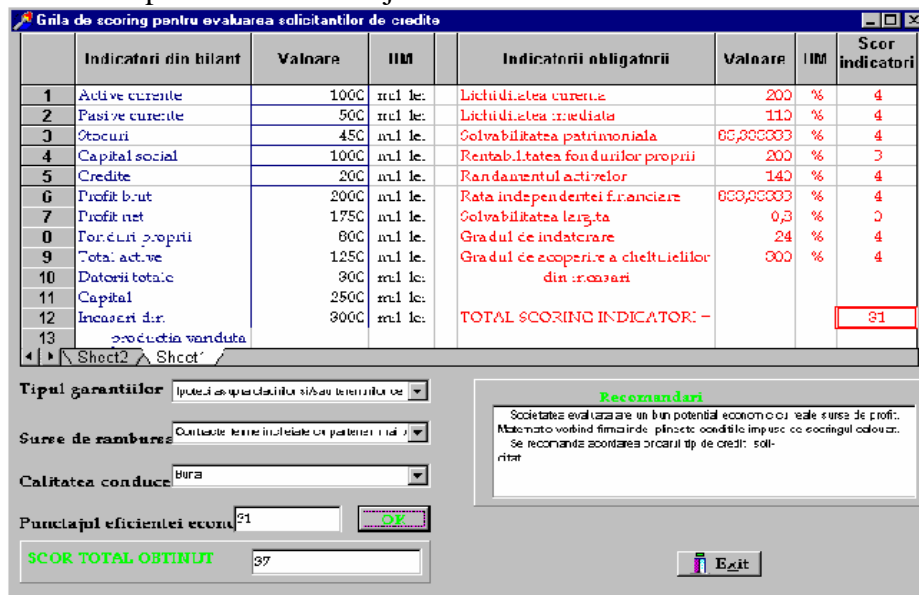


Fig. 1. Fereastra de calcul a scorului creditului

ScorForm este un obiect TForm ce reprezinta o fereastra aplicatie standard. Obiectul TForm se foloseste pentru a crea form-uri cum ar fi: ferestre principale, cutii de dialog, sau descendenti MDI(Multiple Document Interface). Obiectul TForm poate contine mai multe obiecte cum ar fi obiecte de tip buton (TButton), combo-box-uri (TcomboBox) si altele.

În Delphi obiectele se utilizeaza definind proprietatile lor, metodele implementate si evenimentele la care acestea raspund. Se prezinta în continuare setarile proprietatilor obiectului ScorForm :

object ScorForm: TScorForm

Left = 28

Top = 50

Width = 739

Height = 502

Caption = 'Grila de scoring pentru evaluarea solicitantilor de credite'

Font.Charset = ANSI_CHARSET

Font.Color = clWindowText

Font.Height = -13

Font.Name = 'Book Antiqua'

Font.Style = [fsBolds]

Position = poScreenCenter

OnActivate = DeschideDB

OnClose = InchideDB

PixelsPerInch = 96

TextHeight = 16

{definitia celorlate obiecte componente}
end

Celelalte obiecte componente ale form-ului ScorForm sunt obiecte de tipul :

- TComboBox – ConBox, GarBox, Ramb-Box;

- TDBMemo – RecMemo (element de interfata ce permite vizualizarea continutului unui câmp de tip Memo dintr-o tabela);

- TLabel – Total, ScorEc (control ce permite afisarea/citirea unui text dintr-o fereastra a aplicatiei);

- TPanel – ScorPan, RecPan (element ce permite gruparea controalelor si componentelor ferestrei atât din punct de vedere al mostenirii proprietatilor cât si din punct de vedere al aspectului interfetei ferestrei);

- TButton – OkButton, ExitButton;

- TFIBook – F1Book1- componenta mai speciala ce functioneaza ca o foaie de calcul cu formulele predefinite de program.

Aceste obiecte au atasate proceduri de prelucrare, ce se lanseaza ca urmare a producerii unui eveniment recunoscut de aceste controale. De exemplu, butoanele din clasa

TButton pot reactiona la urmatoarele evenimente: `OnClick`, `OnDragDrop`, `OnDragOver`, `OnEndDrag`, `OnEnter`, `OnExit`, `OnKeyDown`, `OnKeyPress`, `OnKeyUp`, `OnMouseDown`, `OnMouseMove`, `OnMouseUp`, `OnStartDrag`, lista ce se poate extinde prin definirea, de catre programator, a unor evenimente suplimentare.

Butonul `OkButton` serveste la calcularea scorului total al solicitantului de credit, raspunzând la evenimentul `OnClick`. Procedura de prelucrare este urmatoarea :

```

procedure
TScorForm.OkButonClick(Sender:
TObject);
begin
  if ScorOk and GarOk and SursOk and
CondOk then
  begin
    Scor:=scor+sc+sg+ss;
    Total.Text:=IntToStr(Scor);
    RecT.First;
    if (scor>=30) and (scor<=60) then
RecT.FindKey(['A']);
    if (scor>=23) and (scor<=29) then
RecT.FindKey(['B']);
    if (scor>=18) and (scor<=23) then
RecT.FindKey(['C']);
    if (scor>=11) and (scor<=17) then
RecT.FindKey(['D']);
    if (scor>=0) and (scor<=10) then
RecT.FindKey(['E']);
    RecPan.Visible:=True;
    ScorPan.Visible:=True;
  end;
end;

```

În aceasta procedura se face legatura între scorul total obtinut de client si tabela ce contine interpretarile si recomandările ca urmare a evaluării scoring-ului.

Componenta `TF1Book` este o componenta de tip `ActiveX`, incorporata în paleta standard de componente oferite de mediul de programare Delphi ce permite lucrul cu o foaie de calcul din interiorul unei aplicatii independente.

O componenta control `ActiveX` este o componenta obiect a unui program ce poate fi refolosit de mai multe programe de aplicatii, ce nu poate fi rulat singur ci numai 'plasat' în interiorul unui alt program numit 'host'. Un control de tip `ActiveX` poate fi asemanat, în linii mari, ca modalitatea de concepere si utilizare, cu un applet Java. Hostul unei componente `ActiveX` poate fi orice program ce suporta OLE (Object Linking and Embedding) si care utilizeaza interfetele COM (Component Object Model).

COM este o arhitectura software ce permite aplicatiilor sa fie construite din componente software binare (compile). COM este arhitectura ce sta la baza serviciilor software de nivel înalt, ca acelea oferite de OLE si `ActiveX`, ce permit interoperabilitatea între componente. Aceste componente pot fi realizate în limbaje diferite, de producatori diferiti, modul de comunicare între ele fiind oferit de COM – o arhitectura software ce :

- defineste un standard binar pentru interoperabilitatea între componente;
- este independent de limbajul de programare;
- este suportat de multiple platforme (Windows 95, NT, MacIntosh, Unix);
- ofera o evolutie robusta a aplicatiilor si sistemelor bazate pe componente;
- este foarte extensibil;
- foloseste un singur model de programare pentru comunicarea între componente în cadrul unui acelasi proces sau din procese sau retele diferite;
- permite utilizarea partajata a memoriei între componente;
- ofera un sistem robust de semnalare a exceptiilor si erorilor;
- permite încarcarea si descarcarea dinamica a componentelor.

Delphi permite ca orice obiect de tip `TForm` sa poata functiona ca un container `ActiveX`, în cazul de fata acest rol fiind jucat de fereastra `ScorForm`.

Scoring-ul rezultat din completarea celulelor foii de calcul este introdus în câmpul cu eticheta *Punctajul eficientei economice* si dupa

alegerea celorlalte elemente (*Tip garantii, Calitatea conducerii si Surse de rambursare*) cu ajutorul componentelor de tip ComboBox si apasarea butonului OK se va afisa scorul total împreuna cu recomandarile predefinite în functie de intervalul în care se gasesc scorul total. În functie de politica de creditare a bancii se va accepta/respinge cererea de creditare.

Modulul de creditare contine referinte catre urmatorul module :

- *Acordare* – pentru înregistrarea datelor despre un credit acordat;
- *Modificare* – pentru cazul în care trebuie modificate conditiile derularii relatiei de credit sau când au fost introduse eronat date despre acel credit;
- *Utilizare* – pentru înregistrarea utilizarii sumelor din contul de credit;
- *Rambursare* – pentru înregistrarea rambursarii sumelor în cadrul contractului de creditare, fie ca acestea reprezinta dobânzile aferente fie ca reprezinta transele stabilite a fi rambursate;
- *Restante* – pentru trecerea la restant a dobânzilor si transelor nerambursate la momentele de timp stabilite în cadrul contractului de creditare.

Aceste module sunt încarcate dinamic la apasarea butoanelor modulului de creditare si descarcate dinamic în momentul iesirii din fereastra.

Modulele enumerate se numesc: AcordareForm, ModifCrForm, RambForm, UtilizCrForm, RestantForm si sunt apelate din forma CrediteForm:

```
procedure
TCrediteForm.AcordareClick(Sender:
TObject);
begin
Application.CreateForm(TAcordareForm,A
cordareForm);
AcordareForm.ShowModal;
AcordareForm.Free;
end;
procedure
TCrediteForm.ModificareClick(Sender:
TObject);
```

```
begin
Application.CreateForm(TModifCrForm,M
odifCrForm);
ModifCrForm.ShowModal;
ModifCrForm.Free;
end;
```

```
procedure
TCrediteForm.RambursareClick(Sender:
TObject);
begin
Application.CreateForm(TRambForm,Ramb
Form);
RambForm.ShowModal;
RambForm.Free;
end;
```

```
procedure
TCrediteForm.UtilizareClick(Sender:
TObject);
begin
Application.CreateForm(TUtilizCrForm,
UtilizCrForm);
UtilizCrForm.ShowModal;
UtilizCrForm.Free;
end;
```

```
procedure
TCrediteForm.RestanteClick(Sender:
TObject);
begin
Application.CreateForm(TRestantForm
,RestantForm);
RestantForm.ShowModal;
RestantForm.Free;
end;
```

Se observa modul de lucru si referirea tipic obiectuala a componentelor, proprietatilor si metodelor unui obiect.

În cadrul modulului destinat creditarii au fost folosite cereri SQL fie pentru a selecta pentru utilizator doar acele înregistrari din baza de date care îi sunt utile la un moment dat fie pentru a-l informa în general cu privire la înregistrările baze de date în cadrul rapoartelor structurate:

```
Select CodC, NumeC, DataA, Val, UM,
Scadenta from clienti, credite
```

Where clienti.CodC=credite.CodClient

Exemplul precedent este un "join" realizat pe doua tabele *clienti* si *credite* ce sta la baza raportului ce desemneaza clientii Bancii ce au fost creditati.

```
select TipRest, SumaRest, UM, Scadenta
from restante
where CodCredit=:cod
```

Exemplul anterior este un model de cerere parametrizata ce returneaza din tabela *restante* datoriile restante ale clientului ce detine creditul cu codul *cod*, parametru ce va primi valoare la rularea programului, valoare ce va fi introdusa de utilizator într-un câmp de tip *TEdit*. Transferul parametrului dinspre aplicatie catre BDE pentru realizarea cererii se realizeaza datorita setarii la momentul executiei a atributului *Params* cu o valoare corespunzatoare ca tip cu cea declarata în atributul cu acelasi nume. Cererea se

numeste *UtilizQ* iar parametrul *cod* a fost declarat ca fiind de tip *string* si a primit numarul de ordine 0 în cadrul listei de parametrii din clauza SQL, el fiind si singurul de altfel .

```
UtilizQ.Params[0].AsString:=ContCr.Text;
```

O alta modalitate de instantiere a valorii parametrului este utilizând metoda *ParamByName* si specificând numele parametrului de instantiat.

```
UtilizQ.ParamByName('cod').AsString:=ContCr.Text
```

În figura 2 se prezinta un exemplu de raport cu privire la creditele mari acordate de banca, raport ce foloseste un obiect din clasa *TQRChart* pentru a figura structura creditelor mari acordate de banca clientilor sai.

Ponderele creditelor mari in totalul creditelor mari

(mai mari de 50 mld.) acordate de banca

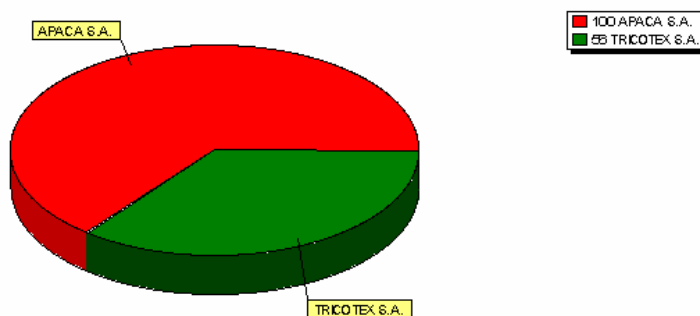


Fig. 2. Parte a unui raport cu privire la structura creditelor acordate de banca

Sistemul de raportare cu privire la creditele acordate, restante sau curente poate constitui deasemenea subiectul unui articol întreg, în articolul de fata scopul fiind acela de a trece doar în revista posibilitatile de implementare modulului de creditare – parte a unui sistem informatic bancar.

Bibliografie

[1] Basno Cezar, Dardac Nicolae, Floricel Constantin - **Moneda, Credit, Banci**, Editura Didactica si Pedagogica, Bucuresti, 1995

[2] Dardac Nicolae, Botea Emil - **Moneda si credit**, Universitatea independenta "Titu Maiorescu", 1994

[3] Dr. Dobbs Journal 1996

[4] Norton Peter - **Borland Delphi 2.0**, Editura Teora, Bucuresti, 1998

[5] Lopatan Ionut - **Borland prezice viitorul RAD: Delphi**, RAD în ajutorul bazelor de date, Revista PCWorld 1996

[6] **BORLAND Delphi 3.0 Client /Server Suite Help**