

DEZVOLTAREA UNUI SISTEM DE TRANZACȚIONARE

GABRIEL STICLARU
Universitatea Spiru Haret*

Rezumat: Articolul explică cum putem dezvolta un sistem de tranzacționare care să permită configurare și personalizare conform nevoilor utilizatorului. Componenta principală este un soft de analiză tehnică de top, de exemplu Metastock, cu care dezvoltatorii experimentați pot să-și integreze propriile soluții de tranzacționare folosind interfața API. Se prezintă modul de decodificare și colectare a datelor de bursă, crearea de noi indicatori tehnici, filtre și biblioteci externe. Dezvoltarea sistemului are ca scop automatizarea tranzacțiilor prin obținerea de semnale (alerte) și fundamentarea deciziei de vânzare/cumpărare.

Cuvinte cheie: Metastock format, Indicatori tehnici, FTP Downloader, MDK (Metastock Developer Kit), tranzacții bursiere.

1. INTRODUCERE

Decizia de vânzare/cumpărare de acțiuni la bursă trebuie fundamentată științific, atât în cazul pieței spot, cât și al pieței instrumentelor financiare derivate (opțiuni și futures). Este foarte dificil, practic imposibil pentru o echipă mică de programatori, să realizeze un soft de tranzacționare care să rivalizeze cu Metastock, TradeStation, Omnitrader, Wall Street Analyzer sau Amibroker. În spatele acestor aplicații complexe stă munca a sute de programatori și cercetători, decenii de dezvoltare și actualizare. Cel mai bine este să realizăm aplicații externe care să comunice cu aceste sisteme de top și să le personalizeze, conform propriilor noastre cerințe.

Scopul acestei lucrări este să arate cum, plecând de la un astfel de soft de top, de mare generalitate, putem dezvolta un sistem integrat de aplicații, care să poată fi gestionat eficient de către un trader din România.

2. CUM PUTEM DEZVOLTA SISTEMUL DE TRANZACȚIONARE

Sistemele de tranzacționare gen Metastock, TradeStation sau OmniTrader oferă un SDK (*Software Development Kit*), pe baza căruia programatorii experimentați pot dezvolta sistemul, prin integrarea de aplicații proprii.

Metastock este cel mai vechi și mai sofisticat soft de tranzacționare, cu versiune Windows începând din 1995. Metastock este produs al firmei Equis International, companie care aparține de Reuters, fiind considerat liderul mondial în softul de analiză tehnică.

* Masterand

Kitul de dezvoltare oferit, MDK (*MetaStock Developers Kit*), conține librării API care oferă următoarele facilități:[1]

- programarea în C++ Builder, Delphi și Visual Basic;
- crearea de noi indicatori tehnici;
- prelucrări de date direct în formatul Metastock (citire/scriere, editare, sortare, ștergere, înserare);
- crearea de biblioteci externe folosind "MetaStock Formula Language", care dacă sunt depuse în directorul "External Function DLLs" al instalării, vor fi încărcate automat;
- export de formule și indicatori;
- schimbul de date cu aplicații compatibile DDE (*Dynamic Data Exchange*), cum ar fi Excel sau aplicații proprii, clienți DDE;
- personalizarea interfeței utilizator (butoane, opțiuni de meniu, legături la aplicații externe).

Un sistem de tranzacționare se poate dezvolta prin:

- configurarea sistemului pentru piețele selectate: BMFMS(Bursa Monetar Financiară și de Mărfuri Sibiu) sau BVB(Bursa de Valori București);
- posibilitatea selectării simbolurilor ce se doresc a fi urmărite;
- utilizarea unor indicatori tehnici care să corespundă teoriilor proprii, elaborate pentru piețele selectate;
- personalizarea meniului softului de analiză tehnică;
- preluarea datelor de bursă;
- crearea unei baze de date de istoric;
- analiza datelor de istoric (de exemplu prin sisteme exploratorii);
- crearea de sisteme de test, pentru verificarea strategiilor proprii;
- crearea de sisteme expert pentru obținerea de semnale (alerte)
- prelucrarea datelor de bursă;
- export și import de date cu alte aplicații;
- realizarea de aplicații proprii, care să comunice cu softul de tranzacționare.

2.1. FORMATUL STANDARD METASTOCK

Bursa pune la dispoziție date în format text (CSV) sau Metastock (care s-a impus ca standard internațional), arhivate la sfârșit de perioadă (zi, săptămână) sau date în timp real (arhivate de exemplu la 5 minute).

Datele în format text (CSV) nu sunt standardizate, fiind arhivate în diferite modalități. De exemplu, o structura minimală conține: Symbol, Date, Open, High, Low, Close, Volume. O structură mai complexă are câmpurile: Simbol, Data, Maxim, Minim, Pret, Volum, Număr tranzacții, Valoare tranzații, Best Ask, Best Bid, Volum Best Ask, Volum Best Bid.

Din analiza formatului Metastock am desprins următoarele tipuri de fișiere asociate(MASTER și Fn.DAT sunt obligatorii):

MASTER – este un fișier index binar, care conține câmpuri de identificare a tranzațiilor. Câmpurile care pot fi extrase din această structură sunt următoarele: Security (simbolul emitentului de acțiuni), Symbol (cod simbol), Periodicity (lunar, săptămânal, zilnic etc), First date, Last date, First time, Last time, Start time, End time, Collection (data arhivei)

Fn.DAT (n de la 1 la 255) – fișier care conține toate tranzacțiile unui anumit titlu financiar, pe perioada considerată, cu următoarele câmpuri: Date (data calendaristică a tranzacției), Time (oră, minut, secundă), Open (prețul de deschidere), High (prețul cel mai mare), Low (prețul cel mai mic), Close (prețul de închidere), Volume (cantitate tranzacționată), Open Interest (număr poziții deschise).

EMASTER (Extended Master file) este un fișier similar cu MASTER, adăugat de MetaStock ca o extensie la formatul original Computrac.

Optional, fișierul Fn.DOP, cu structura variabilă, în format ASCII, cu informații suplimentare.

Pentru cazul când există mai mult de 255 cotații, s-a creat un nou fișier index, XMASTER cu fișierele corespunzătoare Fn.MWD (n de la 256 la 6000), cu structura identică cu Fn.DAT.

Pentru a crea o aplicație care să folosească o arhivă Metastock, este nevoie să cunoaștem structura acestor fișiere binare.

Fișierul **MASTER** a fost introdus de firma ComPutrac (actualmente SmartTrader) și are o structură pe 2 înregistrări de câte 53 bytes.

Prima înregistrare:

Nume câmp	Format	Start	Mărime
Number of Files	UW	1	2
Next file	UW	3	2
Reserved5	UB	4	45
Unknown	MBF	49	4

A 2-a înregistrare:

Nume câmp	Format	Start	Mărime
File Number	UB	1	1
Type	UW	2	2
Length	UB	4	1
Fields	UB	5	1
Reserved	1	6	2
Security	A	8	16
Reserved	2	24	1
Vflag	A	25	1
First Date	MBF	26	4
Last Date	MBF	30	4
Period	A	34	1
Time	UW	35	2
Symbol	A	37	14
Reserved	3	51	1
AutoRun	A	52	1
Reserved	4	53	1

unde:

UB = unsigned byte “unit8”

UW = unsigned word “uint16”

A = ASCII

MBF = Microsoft Basic Floating point in 4 bytes

Fn.DAT are următoarea structură:

Nume câmp	Start	Mărime
Date	0	4
Open	4	4
High	8	4
Low	12	4
Close	16	4
Volume	20	4
Open Interest	24	4

Un acces la arhiva Metastock se desfășoară astfel:

1. Se deschide și se citește fișierul MASTER.
2. Se determină numărul de înregistrări din MASTER.
3. Se citește înregistrarea 2 din MASTER.
4. Pentru fiecare înregistrare de lungime 53 bytes din MASTER, se determină fișierul asociat Fn.DAT.
5. Se deschide Fx.DAT și se citesc datele.

EMASTER (Extended Master) a fost adăugat de firma MetaStock, devenită mai târziu Equis International. Are o structură mai simplă decât MASTER și folosește formatul IEEE pentru reprezentarea numerelor zecimale, tot într-o structură cu 2 înregistrări, dar de 192 bytes fiecare:

Prima înregistrare:

Nume câmp	Format	Start	Mărime
Number of Files	UW	1	2
Last file	UW	3	2
Reserved5	UB	4	188

A 2-a înregistrare:

Nume câmp	Format	Start	Mărime
ID code	A	1	2
File Number	UB	3	1
Filler1	A	4	3
Fields	UB	7	1
Filler2	A	8	2
AutoRun	A	10	1
Filler3	A	11	1
Symbol	A	12	14
Filler4	A	26	7
Name	A	33	16
Fill5	A	49	12
Time Frame	A	61	1
Fill6	A	62	3
First Date	CVS	65	4
Fill7	A	69	4

Last Date	CVS	73	4
Fill8	A	77	50
First Dt Long	CVL	127	4
Fill9		131	1
Dividend Date	CVL	132	4
Dividend Rate	CVS	136	4
Fill10		140	53

XMASTER este un fișier index, binar cu 150 bytes, cu structura următoare:

Nume câmp	Start	Mărime
Unknown	0	1
Stock symbol	1	15
Stock name	16	46
D-update type	62	1
n in Fn.MWD	65	2
Unknown	67	13
End Date	80	4
Unknown	84	20
Start Date	104	4
Start Date	108	4
Unknown	112	4
End Date	116	4
Unknown	120	30

2.2. PRELUAREA DATELOR DE BURSĂ

Datele cu cotațiile de pe piața americană și piața internațională se pot descărca gratuit de la Yahoo sau Google (finance.yahoo.com, finance.google.com)

Majoritatea pachetelor de analiză tehnică au programe cu care se pot descărca cotații pentru unele burse străine din diverse surse (de exemplu programul Downloader pentru Metastock sau AmiQuote pentru Amibroker).

Pentru a obține datele de pe bursele interne, putem apela la brokerul prin intermediul căruia realizăm operațiunile bursiere: plasarea, modificarea sau anularea de ordine, operațiuni financiare de plată sau încasare.

Dacă dorim însă să realizăm o bază de date proprie, este preferabil să realizăm un program propriu de preluare a datelor de la bursă.

O aplicație de tip "FTP downloader" poate fi realizată folosind ICS, Indy sau TurboPower, pachete de componente distribuite freeware, cu sursele incluse, pentru diverse limbaje de programare.

ICS (Internet Component Suite) este un set de componente native Delphi pentru FTP client și server, HTTP server, Ping, SMTP client, POP3 client, sockets, DNS, finger etc.

Indy (Internet Direct) este o librărie "open source" care suportă socheți pentru clienți și servere TCP, UDP, RAW precum și alte protocoale: SMTP, POP3, NNTP, HTTP etc. Pachetul este disponibil pentru: C#, C++, Delphi/Kylix și platformele .NET.

TurboPower Internet Professional este un set de componente VCL, freeware/open-source pentru conectarea la Internet: POP3, SMTP, NNTP, FTP, HTTP, Mesagerie, HTML.

O aplicație minimală de tip FTP trebuie să conțină în fereastra principală:

- URL server FTP (nume sau IP);
- port (uzual 21 sau 1080);
- nume utilizator (root, guest, anonymous etc);
- parola (opțional);
- conexiune de date pasivă (câmp boolean);
- directorul de unde se face preluarea fișierelor, care poate fi de exemplu:
www.tranzactiibursiere.ro/bvb/intraday_temp/;
- lista de fișiere de descărcat;
- directorul de pe stația locală, unde se depozitează fișierele, de exemplu:
C:\BVB\intraday_temp\;
- ora de început și de sfârșit pentru activitatea bursei (de exemplu între 10.00 și 16.00);
- perioada de reluare (unele burse arhivează datele la fiecare 5 minute)
- zilele fără activitate (zile nelucrătoare).

Iata o listă de fișiere de preluat, care poate fi menținută într-o componentă de tip ListBox:

Fișier	Simbol	Explicații
MASTER		Fișier index
EMASTER		Fișier index
F1.dat	BIO	BIOFARM S.A.
F2.dat	BRD	BRD - GROUPE SOCIETE GENERALE S.A.
F3.dat	BRK	S.S.I.F. BROKER S.A.
F4.dat	RRC	Rompetrol Rafinare
F5.dat	SIF1	SIF BANAT CRIȘANA S.A.
F6.dat	SIF2	SIF MOLDOVA S.A.
F7.dat	SIF3	SIF TRANSILVANIA S.A.
F8.dat	SIF4	SIF MUNTENIA S.A.
F9.dat	SIF5	SIF OLTENIA S.A.
F10.dat	SNP	S.N.P. Petrom S.A. Bucuresti
F11.dat	TEL	Transelectrica S.A.
F12.dat	TLV	Banca Transilvania
F13.dat	BET	Indicele BET(Bucharest Exchange Trading)
F14.dat	BETC	Indicele BET-C (Bucharest Exchange Trading - Composite)
F15.dat		Indicele BET-FI (Bucharest Exchange Trading-Fonduri de Investiții)

Fișierul MASTER se va descărca obligatoriu, pentru celelalte fișiere poate să existe un câmp boolean pentru selecție/deselecție.

Un astfel de program va putea descărca automat cotațiile day sau intraday în format MetaStock în directorul selectat de utilizator, pentru a putea fi utilizate de Metastock sau de alte aplicații proprii pentru prelucrări diverse(sortări, selecții, grupări, grafice etc).

Dacă am programat un astfel de "FTP Downloader", actualizarea bazei de date se poate face zilnic, în timp real sau la închiderea burselor RASDAQ, BVB sau BMFMS.

Prezentăm o implementare parțială a programului, folosind componenta FTP din TurboPower (codul sursă în Delphi).

```
// setare director sursă pentru preluare date
```

```

procedure TForm.BtnDirectorSursaClick(Sender: TObject);
begin
  FtpClient.ChangeDir(DirectorSursa.text);
end;
// secvența de login
procedure TForm.BtnLoginClick(Sender: TObject);
begin
  FtpClient.Login(URL.text,Utilizator.text, Parola.Text,"");
end;
// secvența de logout
procedure TForm.BtnLogoutClick(Sender: TObject);
begin
  FtpClient.Logout;
end;
// bucla de preluare fișiere aflate în ListBox
procedure TForm.BtnStartClick(Sender: TObject);
var
  I : Integer;
begin
  for I := 0 to ListBox.Items.Count - 1 do
  begin
    FTPClient.Retrieve(ListBox.Items[i], DirectorSursa.text+'\'+
      ListBox.Items[i],rmReplace ,0);
    while FTPClient.InProgress do
    begin
      Application.ProcessMessages;
      sleep(500);
      Application.ProcessMessages;
    end;
  end;
end;
end;

```

2.3. CREAREA DE NOI INDICATORI

Sistemele de tranzacționare de top oferă un limbaj de programare proprietar, pentru crearea de noi indicatori: Metastock Formula Language (Metastock), EasyLanguage(TradeStation), AmiBroker Formula Language(AmiBroker) etc.

În cele mai multe cazuri, o strategie de tranzacționare a unui simbol înseamnă alegerea unui set de indicatori tehnici care vor fi urmăriți. Unii indicatori sunt implementați de Metastock, alții trebuie modificați iar uneori, programați noi indicatori. Fiecare nou indicator are o interpretare și o formula de calcul. De exemplu, “Force Index”, creat de dr. Alexander Elder [2], este un indicator de tip oscilator, care combină 3 elemente ale informației de pe piața:

- prețul;
- direcția schimbării prețului;
- volumul tranzacționat.

Vom programa în continuare indicatorul original Force Index, folosind utilitarul “Indicator Builder” și limbajul MFL (Metastock Formula Language) [3].

MetaStock -> Tools -> Indicator Builder -> New.

La "Name" se scrie de exemplu Force Index.

La "Formula" se scrie textul de mai jos:

{ Se crează un set nou de date, multiplicând volumul (cantitatea tranzacționată) cu diferența dintre prețul de închidere (C=Close) între 2 momente consecutive }

FI:=V* (C-Ref(C,-1));

{ Această serie de date este ajustată cu mediile mobile (Moving Average), calculate pe 13 momente consecutive, lisaj exponențial (E) }

Mov(FI,13,E);

Pentru a permite utilizatorului să-și verifice propriile reguli (Elder recomandă și cazul mediilor mobile pe 2 momente consecutive), vom adăuga la codul precedent o fereastră în care să fie setat parametrul "număr momente".

{cerem valoarea minimă, maximă, implicită}

NRP:=Input("Număr perioade pentru <Force Index>", 2,30,13);

FI:=V* (C-Ref(C,-1));

Mov(FI,NRP,E);

Să presupunem că trader-ul (utilizatorul sistemului de tranzacționare) dorește să găsească toate tranzacțiile în care seria cotațiilor pe 2 momente consecutive, se intersectează cu seria cotațiilor pe 7 momente (comparații între mediile mobile simple). Vom crea un sistem de explorare (filtrare), care va afișa aceste înregistrări:

MetaStock -> Tools -> Explorer -> New.

{ introducem în câmpul Filter, formula de mai jos }

Cross(Mov(C,2,S) , Mov(c,7,S))

Avem de scris mai mult cod dacă dorim să analizăm cotațiile care au prețul de închidere (C=Close) peste mediană, în ultimele 4 momente:

MetaStock -> Tools -> The Explorer -> New.

{ completăm coloanele A, B, C, D și filtrul }

Col A = CLOSE - MP()

Col B= (Ref(CLOSE,-1))-(Ref(MP() ,-1))

Col C = (Ref(CLOSE,-2))-(Ref(MP() ,-2))

Col D = (Ref(CLOSE,-3))-(Ref(MP() ,-3))

FILTER = ColA>=0 AND ColB>=0 AND ColC>=0 AND ColD>=0

Metastock este echipat cu o serie de Experți, care se pot atașa unor grafice, dar putem să construim experți noi, pentru obținerea de alerte sau afișarea unor comentarii la producerea unor evenimente favorabile sau nefavorabile (MetaStock -> Tools -> Expert Advisor -> New) [4].

Putem simula o strategie proprie, dacă construim un sistem de testare (MetaStock -> Tools -> The Enhanced System Tester -> New) care va genera un raport financiar de profitabilitate sau pierdere: cât am fi câștigat sau pierdut dacă am fi aplicat aceasta strategie (mai multe exemple se pot găsi în [3], [4], [5])

2.4. BIBLIOTECI EXTERNE

Folosind interfața de programare MDK(Metastock Developer Kit), vom crea o librărie Win32(DLL), care trebuie copiată în directorul "External Function DLLs" al instalării. La fiecare lansare a aplicației, biblioteca va fi încărcată automat și vom găsi implementat un nou indicator, util în simularea strategiilor, care apelat va genera o serie aleatoare de tranzacții.

```
// codul sursă în Delphi
library Test_MDK;
uses SysUtils, Math;
{$I MSXStruc.inc} // furnizat de MDK
function MSXInfo (var a_psDLLDef: MSXDLLDef): LongBool; stdcall;
begin
  StrLCopy (a_psDLLDef.szCopyright, 'Biblioteca externa creata cu API',
MSX_MAXSTRING-1);
  a_psDLLDef.iNFuncs := 1; // număr de indicatori creați
  a_psDLLDef.iVersion := MSX_VERSION;
  MSXInfo := MSX_SUCCESS;
end;
function MSXNthFunction (a_iNthFunc: Integer; var a_psFuncDef:
MSXFuncDef): LongBool; stdcall;
var l_bRtrn : LongBool;
begin
  l_bRtrn := MSX_SUCCESS;
  case a_iNthFunc of
    0: begin
      StrCopy(a_psFuncDef.szFunctionName, 'Serie_Aleatoare');
      StrCopy(a_psFuncDef.szFunctionDescription, 'Functia generează o serie
aleatoare de numere');
      a_psFuncDef.iNArguments := 0;
    end;
    else l_bRtrn := MSX_ERROR;
  end;
  MSXNthFunction := l_bRtrn;
end;
function Serie_Aleatoare (const a_psDataRec: PMSXDataRec;
const a_psDataInfoArgs: PMSXDataInfoRecArgsArray;
const a_psNumericArgs: PMSXNumericArgsArray;
const a_psStringArgs: PMSXStringArgsArray;
const a_psCustomArgs: PMSXCustomArgsArray;
var a_psResultRec: MSXResultRec): LongBool; stdcall;
var l_iIndex : Integer;
    l_iMaxIndex: Integer;
    i : Integer;
    l_bRtrn : LongBool;
begin
  l_bRtrn := MSX_SUCCESS;
  l_iIndex := a_psDataRec.sOpen.iFirstValid;
```

```

l_iMaxIndex := a_psDataRec.sOpen.iLastValid;
randomize;
if l_iMaxIndex >= l_iIndex then
  for i:= l_iIndex to l_iMaxIndex do
    a_psResultRec.psResultArray.pfValue[i] :=random;
  Serie_Aleatoare := l_bRtrn;
end;
exports
MSXInfo, MSXNthFunction, MSXNthArg,
Serie_Aleatoare;
begin
end.

```

3. CONCLUZII

În jurul unui soft de tranzacționare care oferă o interfață de programare și dezvoltare (SDK), putem dezvolta componente specifice. Putem crea aplicații externe care să colecteze și să prelucreze datele de bursă, să administreze o bază de date proprie, cu istoricul tranzacțiilor și să comunice direct (prin librării externe) sau prin DDE cu componenta centrală a sistemului. Dacă utilizatorul și-a creat strategii proprii de tranzacționare, acestea pot fi simulate și apoi implementate în softul de analiză tehnică, cu comentarii dependente de context și alerte care se declanșează la producerea evenimentelor programate.

BIBLIOGRAFIE

1. <http://www.metastocktools.com>
2. Elder A., *Trading for a Living*, John Wiley & Sons, Inc, 1993.
3. Equis International, *Formula Primer*, 2002.
4. Equis International, *MetaStock Professional For Windows 98 and higher User's Manual*, Version 8, 2002
5. Achelis S. *Technical Analysis from A to Z*, Equis International, 2003

Abstract: Development of a trading system. This article explains how we could develop a trading system, which may allow us configuring and personalizing it, according to the user needs. The central component is a trading top soft like Metastock, with which experienced developers can integrate their own customized trading solutions using API. We present how to decode the data format, supply historical data, and create new technical indicators, explorations and external library. The purpose is to make the trading automatic through collecting the signals (alerts) and preparing the selling/buying decision.