

## Tehnologii deschise în programarea calculatoarelor. Schimbul dinamic de date între aplicații - o abordare Delphi

Prep. Cristian USCATU

Catedra de Informatică Economică, A.S.E. București

*În Windows, de obicei, rulează mai multe aplicații simultan, fiecare în fereastra ei. Aceste ferestre par complet izolate între ele. Chiar dacă din punct de vedere al aplicațiilor, afirmația este de obicei adevărată, din punct de vedere tehnic este falsă. Între aplicații independente pot exista schimburi de informații care se fac pe diferite căi. Windows oferă trei mari categorii de metode pentru comunicația între programe: clipboard, schimb dinamic de date (referit în continuare ca DDE) și OLE.*

**Cuvinte cheie:** schimb dinamic de date, server, client, conversație, serviciu, subiect (topic), obiect (item)

### 1. De ce DDE ?

Deși cele trei tehnici se folosesc împreună și deseori nu se poate face o delimitare clară între ele, din punct de vedere al utilizatorului și al programatorului diferențele sînt semnificative. Folosirea clipboard-ului este orientată către utilizator, care controlează toate operațiile (copiere, adăugare etc.) în mod explicit. Tehnica OLE (Object Linking and Embedding) este cea mai nouă, cea mai puternică și constituie viitorul tehnologiilor deschise, cel puțin în viziunea Microsoft. OLE se bazează, totuși, pe tehnica DDE pentru realizarea legăturilor între aplicații. În plus folosirea OLE necesită efort suplimentar de programare și este consumatoare mare de resurse ale sistemului de operare. Pe de altă parte, DDE este ușor de implementat prin funcții ale interfeței Windows API și oferă rezultate foarte bune cu un consum neglijabil de resurse.

Inițial, DDE a fost un protocol de comunicație între aplicații bazat pe un set de mesaje Windows. Protocolul a fost creat pentru aplicația MS-Excel. Între timp, folosirea DDE s-a impus ca standard pentru comunicația între programe și a fost integrată în Windows. De la versiunea Windows 3.11 a fost creată biblioteca DDEML (DDE Management Library) care gestionează toate comunicațiile tip DDE. Biblioteca memorază informații despre toate conversațiile active și se plasează ca un intermediar între cele două aplicații care comunică.

Prin tehnica DDE, o aplicație poate să primească date la cerere, să trimită date în mod voluntar, să trimită comenzi unei alte aplicații. Ambele aplicații implicate în schimbul de date trebuie să fie prevăzute cu facilități de comunicare DDE.

### 2. Conversația DDE

Un schimb de date/comenzi are loc întotdeauna între două aplicații: furnizorul de date (server) și beneficiarul datelor (clientul - solicitant). Schimbul de date se numește *conversație*. Atît serverul cît și clientul pot întreține simultan mai multe conversații, cu parteneri diferiți, iar o aplicație poate fi simultan server într-o conversație și client în alta. Rolul serverului este de a trimite date clientului la cererea celui din urmă. În plus, serverul poate anunța clientul asupra oricărei schimbări intervenite în setul de date trimis anterior. Notificarea se poate rezuma la un anunț asupra modificării sau poate fi însoțită de retrimiteră întregului set de date. Serverul poate primi comenzi din partea clientului pe care le execută sau le ignoră, în funcție de comportamentul definit la implementarea serverului.

Clientul este cel care inițializează conversația, cere date serverului, trimite date acestuia în mod voluntar și îi cere să execute comenzi.

Desigur aspectul principal al conversației este transmiterea de date. Aceasta se poate face în două feluri: clientul solicită date

## nic de

i. Aceste  
firmația  
lente pot  
gorii de  
ferit în

(topic),

ate să  
date în  
iei alte  
ate în  
zute cu

întot-  
ul de  
ientul  
mește  
il pot  
ii, cu  
simu-  
alta.  
lien-  
plus,  
cărei  
imis  
a un  
oșită  
Ser-  
ien-  
, în  
ple-

ver-  
es-  
ute

ției  
ate  
ate

serverului în mod direct (*request*) sau într-un ciclu numit *advise loop*. Într-un ciclu *advise loop*, clientul cere serverului să-i retrimite datele, dacă acestea suferă vreo modificare pe durata conversației; retrimițerea se face fără vreo altă intervenție a clientului.

În afara serverului și clientului conversația este determinată de alte trei elemente:

- serviciul: este numele aplicației server DDE. De obicei este numele programului executabil (fara extensie) sau un alt nume definit chiar de server.
- subiectul (topicul): este tema globală a conversației. Poate fi numele de fișierului de date, numele unei tabeli unei baze de date sau (de obicei) numele unei ferestre a programului server.
- obiectul (*item*-ul): este un identificator al unui element de date specific care se va transfera între cele două capete ale conversației. Poate fi un câmp al unei tabeli dintr-o bază de date, o celulă a unei foi de calcul etc. În cadrul aceleiași conversații se poate face schimb de date referitoare la mai multe obiecte diferite.

### 3. DDE în Delphi

Mediul Delphi oferă suport pentru tehnologia DDE prin patru componente incluse în pagina System din paleta de componente (imaginea grafică a conținutului bibliotecii de obiecte grafice - Visual Component Library). Deși sînt incluse în biblioteca grafică, componentele sînt non-vizuale: nu au un corespondent grafic în formă Delphi (la proiectarea aplicației prezența lor pe *form* va fi marcată printr-un buton). Cele patru componente sînt:

- **DdeServerConv**: conține descrierea conversației din punct de vedere al aplicației server și oferă suport pentru executarea comenzilor primite de la clienți.
- **DdeClientConv**: constituie portul prin care clientul comunică cu serverul. Are rolul de a lansa o conversație, specificînd serverul (serviciul) și topicul (subiectul).
- **DdeServerItem**: este obiectul (itemul) conversației la capătul dinspre server al conexiunii (fiecare din cei doi parteneri ma-

nipulează propriul său obiect în care stochează datele comunicate într-un sens sau altul); face parte din aplicația server.

- **DdeClientItem**: este obiectul conversației manipulat de client; face parte din aplicația client.

Prezentarea exhaustivă a celor patru componente ar duce la parcurgerea întregii ierarhii de obiecte Delphi, ceea ce nu constituie scopul acestui articol, de aceea se vor prezenta numai aspectele legate de conversația cu alte aplicații.

**DdeServerConv**: componenta nu este neapărat necesară (decît dacă aplicația server oferă suport pentru execuția comenzilor primite de la clienți).

**Proprietăți**: • **name** (Name), implicit DdeServerConv1; este folosit pentru stabilirea legăturii cu o componentă DdeServerItem.

**Metode**: • **Create**: crează o instanță a clasei TDdeServerItem (constructorul obiectului).

• **Destroy**: șterge o instanță a clasei TDdeServerItem și eliberează resursele ocupate (destructorul obiectului); • **ExecuteMacro**: copiază șirul de caractere care conține comanda primită de la client și apelează un eveniment OnExecuteMacro. Dacă nu există un handler pentru acest eveniment, comanda va fi ignorată, altfel handlerul se va ocupa de execuția ei. În ambele cazuri clientul va fi informat asupra succesului/eșecului tentativei de executare a comenzii.

**Evenimente**: • **OnOpen**: se declanșează la crearea/stabilirea unei conexiuni și permite execuția unor operațiuni specifice prin handler-ul atașat; • **OnClose**: asemenea, la terminarea conexiunii; • **OnExecuteMacro**: se apelează pentru executarea comenzilor primite de la client.

#### DdeServerItem

**Proprietăți**: • **Fmt**: este un cod numeric ce reprezintă formatul de transmitere a datelor. Implicit este folosit formatul CS\_TEXT. Sînt posibile toate formatele acceptate pentru clipboard, inclusiv cele definite de utilizator; • **Lines, Text**: proprietăți care memorează datele trimise sub formă de șir de caractere. Lines este de tip Tstrings și permite manipularea pe linii individuale a

datelor iar text este de tip String și permite manipularea datelor în întregime în aceeași operație; •ServerConv: este numele componente tip DdeServerConv care gestionează conversația (dacă există).

*Metode:* •Change: permite apelarea evenimentului OnChange; •CopyToClipboard: copiază în clipboard datele din proprietatea lines; •Create/Destroy: constructorul și destructorul instanței; •Pokedata: se execută atunci când s-au primit date trimise voluntar de client.

*Evenimente:* •OnChange: atunci când proprietatea Lines își schimbă valoarea, înainte de a fi trimise datele clientului se apelează acest eveniment pentru prelucrări suplimentare; •OnPokeData: se apelează atunci când clientul trimite voluntar date serverului, pentru tratarea acestor date.

#### DdeClientItem

*Metode:* •Create/Destroy: constructorul și destructorul instanței.

*Proprietăți:* •DdeConv: numele componente DdeClientConv care gestionează comunicația cu serverul (prin cine se desfășoară comunicația); •Ddelitem: numele obiectului din aplicația server cu care se desfășoară conversația (de tip DdeServerItem). Este cunoscut și poate fi setat după stabilirea conexiunii; •Lines, Text: au aceeași semnificație ca și la componenta anterioară.

*Evenimente:* •OnChange: se apelează atunci când serverul a trimis noi date (la modificarea valorii proprietății Lines/Text);

#### DdeClientConv

*Proprietăți:* •ConnectMode: controlează modul în care este realizată conexiunea. Dacă are valoarea ddeAutomatic atunci se stabilește conversația în momentul în care a fost definită prin metoda SetLink.; dacă valoarea este ddeManual atunci aplicația trebuie să execute explicit metoda OpenLink sau PasteLink pentru stabilirea conversației; •DdeService: definește serverul cu care se va purta conversația. De obicei este numele aplicației server. Tipul proprietății este string; •DdeTopic: proprietatea definește topicul conversației. În general este numele aplicației server, cu calea completă, dacă

este nevoie. Dacă serverul este o aplicație Delphi atunci topicul va fi numele ferestrei din aplicația server în care se află componenta DdeServerItem cu care se conversează. Dacă clientul DDE este legat de o componentă tip TDdeServerConv atunci topicul este numele acestei componente; •FormatChars: este o proprietate de tip boolean care indică dacă la primirea datelor de la server vor fi tratate și caracterele de formatare (CR/LF, spațiu, tab, backspace). Dacă proprietatea are valoarea true atunci vor fi tratate și aceste caractere, altfel vor fi ignorate; •Name: numele componente; •ServiceApplication: numele aplicației server. Dacă aplicația nu rulează în momentul în care se încearcă stabilirea unei conversații, metoda OpenLink va folosi această proprietate pentru a lansa aplicația server; •WaitStat: proprietate de tip boolean care indică dacă este posibilă începerea unei noi tranzacții cu serverul. Dacă serverul procesează încă ultima tranzacție (execută un macro etc.) orice tentativă de începere a unei noi tranzacții se va solda cu eșec.

*Metode:* •CloseLink: termină o conversație. După terminarea conversației nu se poate face schimb de date până la stabilirea unei noi legături cu serverul (OpenLink); •ExecuteMacro: este o funcție membru care se apelează pentru a trimite serverului o comandă. Parametrii de intrare sînt: comanda, de tip Pchar (șir terminat cu caracterul nul) și o valoare booleană care indică dacă clientul va aștepta ca serverul să termine execuția comenzii înainte de a face alte prelucrări. Returnează o valoare booleană care indică succesul tentativei de executare a comenzii; •ExecuteMacroLines: asemănător cu metoda anterioară, însă permite transmiterea simultană a mai multor comenzi. Parametrul *comanda* este de tip TStrings și va conține pe fiecare linie câte o comandă; •OpenLink: încearcă stabilirea unei conversații cu serverul. Dacă se reușește, returnează *true* (este o funcție) și apelează un eveniment OnOpen. În caz de eșec, returnează *false*. Dacă ConnectMode este ddeAutomatic, OpenLink este apelată automat de SetLink; •PasteLink: stabilește o conexiune DDE cu un obiect aflat în clipboard.

Proprietățile DdeService și DdeTopic sînt setate automat în funcție de obiectul din clipboard. Returnează o valoare booleană care indică succesul sau eșecul încercării de conectare; •PokeData: este folosită pentru a trimite date serverului. Parametrii sînt: numele componentei DdeServerItem asociate (care va primi datele) și un parametru de tip Pchar, textul care se trimite serverului; •Setlink: definește o conversație DDE. Parametrii sînt numele serviciului și al topicului. Dacă ConnectMode este ddeAutomatic, apelează OpenLink pentru deschiderea efectivă a legăturii; •RequestData: este o funcție prin care se cer date serverului. Primește ca parametru numele componentei de tip DdeServerItem care va furniza datele și returnează un șir terminat cu caracterul nul (tip Pchar) care conține datele cerute.

#### 4. Realizarea practică a comunicației

În form-ul principal al aplicației client se plasează o componentă DdeClientConv și una DdeClientItem. Se stabilește legătura între cele două componente prin setarea proprietății DdeConv a celei din urmă, în fereastra inspectorului de obiecte. Se definește metoda care tratează evenimentul OnChange al componentei DdeClientItem (va trata datele primite de la server și depuse în proprietatea text/lines). Stabilirea conexiunii se poate face la proiectare prin definirea proprietăților service și topic ale DdeClientConv sau la execuție, ca răspuns la apăsarea unui buton, de exemplu, astfel :

```
if
DdeClientConv1.SetLink('prog_server', 'Fereastra principala') then
begin ShowMessage('Conectare reusita');
DdeClientItem1.ddeitem:='dserveritem1'
end
```

```
else ShowMessage(' Eroare');
```

Pentru generalitate se pot citi numele serverului și al ferestrei sale principale de la tastatură înainte de definirea legăturii. Dacă proprietatea ConnectMode are valoarea ddeAutomatic, se poate începe deja o tranzacție, în caz contrar pe ramura then trebuie apelată explicit și metoda OpenLink:

```
DdeClientconv1.OpenLink;
```

Dacă serverul nu actualizează automat datele este necesar să fie apelată metoda RequestData, ori de cîte ori clientul are nevoie de date de la server:

```
with DdeClientConv1 do
```

```
RequestData('dserveritem1');
```

Deși este o funcție, limbajul permite apelul ca și procedură, datele primite fiind depuse în proprietatea lines/text a clientului. Atunci cînd se dorește terminarea conversației se apelează metoda CloseLink:

```
DdeClientConv1.CloseLink;
```

Am schițat mai sus pașii care trebuie făcuți de aplicația client pentru efectuarea schimbului dinamic de date. Pentru acest exemplu am presupus existența unei aplicații server realizate tot în Delphi care conține doar o componentă DdeServerItem. Dacă aplicația server actualizează proprietatea lines/text a acestei componente, periodic sau în cazul producerii anumitor evenimente, atunci clientul nu are nevoie să ceară explicit trimiterea datelor (RequestData); în acest caz avem o situație *advise loop* impicită.

#### 5. Concluzii

Folosirea clipboard-ului pentru comunicare este o metodă comună majorității aplicațiilor Windows. În programele dezvoltate în Delphi se poate introduce cu ușurință suport pentru acest gen de operații. Punctul slab al tehnicii este orientarea totală către utilizator, care trebuie să execute manual toate operațiile. Introducerea schimbului dinamic de date constituie o îmbunătățire semnificativă a aplicațiilor Delphi, oferind posibilitatea comunicației cu pachete de programe scrise de diferite case de software.

Implementarea tehnicii DDE în aplicațiile Delphi este foarte ușoară, majoritatea codului fiind încapsulat în cele patru componente prezentate; în acest fel conversația este controlată automat. În sarcina programatorului rămîne doar crearea/terminarea conversației și folosirea datelor aferente. Singurul impediment care apare este lipsa documentației referitoare la capacitățile pachetelor software de a folosi tehnica DDE, ceea ce limitează posibilitatea de comunicare cu acestea. Contra cost se pot obține

astfel de documentații de la cei care au scris pachetele respective. În general aceste aplicații vor fi solicitate ca server de către aplicații client dezvoltate în Delphi.

### **Bibliografie**

- Marco Cantú - Mastering Delphi 2.0 for Windows 95/NT, 1996

- Jonathan Matcho ș.a. - Special Edition Using Delphi 2, Que Corporation, Indianapolis, Indiana, U.S.A., 1996
- Dan Osier, Steve Grobman, Steve Batson - Teach yourself Delphi 2 in 21 days, Borland Press & SAMS Publishing, Indianapolis, Indiana, U.S.A., 1996
- Delphi - Help on-line