

CONSIDERAȚII PRIVIND UTILIZAREA MEDIILOR RAD ÎN DEZVOLTAREA APLICAȚIILOR ECONOMICE

Autor: Andreea Bianca URCAN
Universitatea de Vest din Timișoara, România
Facultatea de Economie și Administrare a Afacerilor
Master SIA, anul I
Coordonator științific: prof. dr. Mihaela Muntean

Abstract

Dezvoltarea de aplicații în medii RAD (Rapid Application Development) a cunoscut un real succes încă de la prima apariție a acestei abordări. Într-o primă parte a lucrării ne propunem o incursiune în filozofia mediilor de dezvoltare rapidă a aplicațiilor, punând accentul pe elementele definitorii ale unei aplicații dezvoltate într-un mediu RAD. Elementele teoretice ce vizează proiectarea interfeței grafice, conectarea la surse de date externe, precum și conceperea unor șabloane de interogare vor fi exemplificate practic cu ajutorul mini-prototipului AIF. Acesta, în baza informațiilor preluate din situațiile financiare, calculează indicatorii economici, pe principalele criterii de analiză (structura patrimonială, structura financiară, bonitatea financiară, echilibrul financiar și gestiunea resurselor financiare) și formulează concluzii referitoare la poziția financiară a întreprinderii. Pentru analize complexe, detaliate privind diversele aspecte ce pot influența dinamica indicatorilor și, în consecință, evoluția generală a întreprinderii, se recomandă extinderea prelucrărilor prin facilități OLAP.

Cuvinte cheie: aplicații, business, analiză financiară, RAD, Borland Delphi.

Clasificație JEL: D80, M15

INTRODUCERE

Obiectivul principal al acestei lucrări constă în studierea principalelor caracteristici ale mediilor de dezvoltare RAD (Rapid Application Development) și realizarea unui sistem informatic suport pentru analiza economico-financiară a unei întreprinderi, ca implementare a fundamentelor dezvoltate în prima parte a lucrării.

Necesitatea elaborării acestei lucrări este reprezentată de răspândirea tot mai largă a conceptului de RAD în contextul dezvoltării de aplicații pentru business. Unul din principalele avantaje ale acestora este reprezentat de un timp mai redus de elaborare și implementare a sistemului, care duce la scăderea considerabilă a costurilor angrenate în acest proces. Este foarte important, ca viitori economiști, să stăpânim facilitățile oferite de mediile RAD, întrucât putem realiza aplicații destinate diversilor agenți economici, în mod eficient și rapid.

Lăsând la o parte aspectele teoretice, cea mai importantă parte a lucrării o reprezintă, desigur, aplicația practică, sistemul informatic realizat în termenii filozofiei RAD.

Aplicația se dorește a fi un suport pentru deciziile manageriale, în ceea ce privește analiza economico-financiară. Sistemul informatic, prin cele patru etape logice ale sale, își propune să prelucreze informațiile din situațiile financiare ale unei întreprinderi și, pe baza indicatorilor calculați, să obțină interpretări succinte privitoare la însemnătatea acelor valori pentru viitorul firmei. În acest scop, se vor obține atât situații sub formă de raport, cât și reprezentări grafice, mult mai sugestive.

1. CONSIDERAȚII GENERALE PRIVIND MEDIILE RAD (RAPID APPLICATION DEVELOPMENT)

Conceptul de Rapid Application Development (RAD) a fost definit de către James Martin și reprezintă un proces de dezvoltare a software-ului care permite construirea unui sistem în 60-90 de zile, adesea cu acceptarea unor compromisuri. Din punct de vedere al evoluției, RAD se bazează pe modelul spirală, pe utilizarea prototipurilor și pe dezvoltarea incrementală. De asemenea, există și o influență a metodelor orientate obiect, ce asigură o eficiență sporită pentru RAD, prin utilizarea de componente „pre-fabricate”.

Modelul RAD, în varianta James Martin, are ca principal domeniu de aplicare dezvoltarea de sisteme informaționale și nu este potrivit pentru domenii precum dezvoltarea de sisteme de operare, dezvoltarea de instrumente software (4GL precum Excel) sau crearea de jocuri.

- **Avantajele mediilor RAD**

În modelul RAD, organizarea procesului de elaborare a aplicațiilor se bazează pe următoarele considerente:

- Utilizatorii sunt implicați în proiectarea și validarea sistemului;
- Etapele de proiectare și realizare nu sunt separate în mod real;
- Durata de dezvoltare a sistemului este foarte scurtă, echipele de lucru sunt destul de mici și implicit, costurile de dezvoltare sunt mici.

Un bun motiv pentru alegerea modelului RAD este convergența spre un proiect acceptabil pentru client și fezabil pentru echipa de dezvoltatori. Cu toate acestea, prin reducerea timpului de dezvoltare a aplicației folosind această metodologie, următoarele obiective nu pot fi întotdeauna atinse:

- Cât mai puține defecte posibil;
- Cel mai ridicat nivel posibil pentru satisfacția clientului;
- Cele mai scăzute costuri de dezvoltare.

- **Caracteristici și componente**

Un mediu de dezvoltare RAD este o îmbinare reușită între programarea tradițională (reprezentată în primul rând prin limbajul textual de programare) și dezvoltarea vizuală, care se regăsește în special în fază de proiectare a interfeței. Concret, componentele unui mediu RAD se pot prezenta astfel:

- **Mediul de lucru integrat (IDE – Integrated Development Environment)** asigură dezvoltarea aplicației și permite accesul la toate componentele aplicației.
- **Instrumentele vizuale de proiectare a interfeței** sunt folosite pentru a construi formuri, ferestre de dialog, machete de rapoarte.
- **Limbajul de programare** este în continuare necesar, în ciuda tuturor facilităților vizuale oferite de instrumentele RAD.
- **Supportul pentru conectarea la surse diverse de date (conectivitatea)** este astăzi asigurat de toate instrumentele RAD.
- **Componentele** vizează obținerea unui spor de productivitate în dezvoltarea aplicațiilor și se referă la două aspecte: utilizarea de componente prefabricate și reutilizarea codului de la o aplicație la alta.
- **Distribuția** vizează aici activitățile de după dezvoltare, până ce aplicația este funcțională și instalată la client. Cel mai important aspect este compilarea modulelor componente și realizarea unui set de discuri de instalare.

- **Metode de implementare RAD**

În practica firmelor producătoare de software, RAD este implementat în diverse moduri. Spre exemplu, unele firme implementează RAD într-un ciclu de dezvoltare în care prototipurile asigură obținerea rezultatelor așteptate de utilizator. În alte implementări, RAD se regăsește sub forma

unei metodologii care comprimă fazele de analiză, proiectare, construire și testare într-o serie de cicluri de dezvoltare scurte, iterative.

În prima variantă, apare „prototipizare”, în a doua apare „dezvoltare iterativă”. Prototipizarea se referă la construirea rapidă a unei versiuni a sistemului propus, versiune ce este apoi revizuită, în mod repetat pe baza reacțiilor utilizatorilor, care „probează” fiecare versiune livrată, până când se obține versiunea dorită. Între avantajele metodei se înscriu: obținerea unui produs în concordanță cu așteptările utilizatorilor, datorită comunicării permanente între analist și utilizatori, și cu o interfață utilizator foarte bună. Prototipizarea este indicată mai ales în cazurile în care utilizatorii nu știu cum vor să arate produsul final. În plus, practica a confirmat faptul că prototipizarea asigură un timp de realizare mai mic decât în varianta tradițională.

- **Principalele medii RAD existente**

Mediile de dezvoltare vizuală existente pe piață sunt orientate spre un anumit limbaj de programare.

Câteva exemple:

- **C/C++:** Visual C++ (Microsoft), C++Builder (Borland-Inprise);
- **Java:** Visual J# (Microsoft), JBuilder (Borland-Inprise), JDeveloper (Oracle);
- **Pascal/Object Pascal:** Delphi (Borland-Inprise);
- **Basic:** Visual Basic (Microsoft);

În concluzie, acest capitol reprezintă o introducere în filosofia mediilor de dezvoltare rapidă a aplicațiilor, prezentând o scurtă evoluție a acestora, elementele definitorii ale unei aplicații dezvoltate într-un mediu RAD, caracteristicile și componentele utilizate în construirea unei astfel de aplicații, prezentarea succintă a câtorva medii de dezvoltare RAD cu succes pe piață.

2. PROIECTAREA INTERFEȚEI GRAFICE

Interfața unei aplicații este acea componentă soft caracterizată prin complexitate foarte mare și prin implementare, depanare și modificare dificile. Pentru utilizatori, interfața este cea mai importantă – utilizatorii finali consideră că interfața este de fapt aplicația, pentru că interfața este ceea ce văd ei pe ecran.

Interfața care permite o manipulare vizuală directă se numește **interfață grafică utilizator** (*GUI – Graphical User Interface*). Acest tip de interfață a devenit aproape universal.

Studiul interacțiunii om-calculator a scos în evidență următoarele aspecte generale ce trebuie avute în vedere la proiectarea oricărei interfețe:

- Prezentarea informațiilor trebuie să fie „croită” după nevoile și profitul utilizatorului (ideal este ca acesta să poată aborda aplicația fără să aibă obligatoriu cunoștințe informatice);
- Tratarea (abordarea) acțiunilor dintr-o aplicație în mod similar raționamentului uman;
- Feedback vizual continuu (ca răspuns la acțiunile utilizatorului);
- Prevenirea (evitarea) selectării de opțiuni incompatibile (invalide);
- Includerea unui help contextual și a unor mecanisme de verificare a intrărilor și de restaurare în cazul erorilor;
- Includerea de shortcut-uri pentru utilizatorii avansați;
- Minimizarea nivelului de cunoștințe a priori cerut utilizatorilor pentru exploatarea aplicației.

- **Caracteristicile interfețelor grafice**

În general o interfață grafică se obține prin integrarea mai multor elemente prin care se asigură reprezentarea grafică a programelor, datelor și, în general, a obiectelor de pe ecran, ca și interacțiunea dintre ele. O interfață grafică încearcă să ofere utilizatorilor aplicațiile și mijloacele necesare pentru ca ei să facă ceea ce vor și nu să le arate o listă cu ce poate calculatorul să facă.

Iată o listă a caracteristicilor pe care trebuie să le regăsim la o IGU:

- Folosește un afișaj „hartă de biți” de înaltă rezoluție;
- Utilizează un echipament de indicare (de regulă mouse);
- Vizualizează textele și graficele așa cum apar la tipărire (WYSIWYG);
- Respectă paradigma de interacțiune obiect-acțiune;
- Permite transferul de informații între programe;
- Permite manipularea directă a obiectelor și informațiilor de pe ecran;
- Oferă elemente de interfață standard, cum ar fi meniurile sau dialogurile;
- Asigură reprezentarea vizuală a obiectelor pe ecran (ferestre și icon-uri);
- Asigură feedback vizual la acțiunile utilizatorului;
- Reprezintă vizual acțiunile și operațiunile posibile (în meniuri, linii de instrumente);
- Asigură preluarea opțiunilor și intrărilor utilizatorului prin controale grafice;
- Permite utilizatorului să personalizeze interfața și interacțiunile;
- Asigură utilizatorului libertatea de opțiune de a utiliza tastatura sau alte echipamente de intrare, după dorință și obișnuință.

O interfață grafică bine proiectată trebuie să respecte următoarele **principii**:

1. Controlul aplicației de către utilizator.
2. Manipularea directă a informațiilor.
3. Consistență.
4. Toleranță la greșelile utilizatorului.
5. Asocierea de „replici” pentru comenzile unei aplicații.
6. Estetică.
7. Claritate.
8. Simplitate.

- **Interfața grafică în Delphi**

În Delphi, crearea interfeței cuprinde: proiectarea ferestrelor aplicației, poziționarea lor, introducerea controalelor, dimensionarea și poziționarea lor în fereastră. Delphi ușurează considerabil munca de proiectare a interfeței. Pe tot parcursul procesului de proiectare, programatorul vede pe ecran componentele interfeței și poate adăuga / modifica / șterge controale la oricare din formurile aplicației prin operații extrem de simple. Așa cum reiese, o aplicație poate conține unul sau mai multe formuri. Un form conține mai multe controale, definite cu instrumentele din *Toolbox* și poate avea și o linie de meniuri, proiectată cu fereastra *Menu Design*.

În concluzie, proiectarea unei interfețe eficiente nu este simplă. Desigur, majoritatea oamenilor care au la dispoziție instrumentele potrivite pot să pună laolaltă casete de validare, butoane, casete de intrare și meniuri pentru a concepe un monstru care arată ca o aplicație Windows. Dar este nevoie de multă muncă și o planificare atentă pentru construirea unor interfețe elegante, care să îi atragă pe utilizatori în loc să îi sperie.

Acest capitol tratează tocmai acele elemente și principii, tehnici și metode de care trebuie ținut cont pentru asamblarea unor interfețe prietenoase și practice.

3. CONECTAREA LA SURSE DE DATE EXTERNE. INTEROGĂRI SQL ȘI OLAP

Pentru a dezvolta o aplicație de baze de date în Delphi avem nevoie de un motor special pentru accesarea sursei (*database engine*). Acesta ne ajută să ne concentrăm asupra datelor pe care vrem să le accesăm (Ce vrem să folosim?), și să lăsăm modul de accesare a acestora pe seama motorului (Cum obținem datele?). Încă de la prima versiune, Delphi ne ușurează accesul prin intermediul motorului BDE (*Borland Database Engine*). Începând cu versiunea Delphi 5, acesta suportă și interfața Microsoft ADO.

- **Componente utilizate în gestiunea bazelor de date**

Componentele ADO (Boldea M., 2006) permit realizarea aplicațiilor cu baze de date ADO Delphi. Se utilizează baze de date și obiecte ActiveX din Microsoft, sistemul dispunând de obiecte DB (bază de date) care pot furniza conexiunile la sursa de date și accesul la acele date. ADO provider reprezintă o variantă și un tip de acces direct la obiectele bazelor de date de pe diferite drivere. Dacă aplicațiile utilizează SQL database sau au fost realizate în MS SQL Server sau Oracle, se solicită conectarea lor la sursa lor de date prin intermediul componentei ADO Connection. Logarea unei componente la sursa de date se realizează prin proprietatea Connection String. Sistemul realizează:

- Logarea la server;
- Managementul tranzacțiilor;
- Lucrul cu baze de date asociate;
- Obținerea de metadate.

Principalele componente din pagina ADO, cu funcțiile lor, sunt următoarele:

- *ADO Connection* – realizează controlul atributelor de conectare la sursa de date;
- *ADO Command* – dirijează accesul la sursele de date accesate prin ActiveX;
- *ADO Dataset* – realizează accesul direct la tabelul căutat utilizând Data Manipulation Language (DML) sau prin intermediul instrucțiunilor de tip SQL;
- *ADO Table* – accesează datele dintr-un singur tabel al bazei de date conectate instrucțiunilor de tip SQL;
- *ADO Query* – realizează accesul direct la una sau mai multe tabele din baza de date activă prin instrucțiuni SQL;
- *ADO StoredProc* – procedură utilizată în aplicațiile de tip client, cuprinzând un grup de instrucțiuni pentru metadate (legături între tabele din baze de date diferite, indexări, etc.);
- *RDS Connection* – îndeplinește rolul unui manager ce coordonează multi-applications utilizând baze de date de tip business (Application Servers).

Toate aceste componente au o proprietate comună, și anume DataSource, care este folosită pentru a face legătura între o componentă de date din Delphi și o sursă de date, care este ea înăși legată de un șir de date.

În procesul de gestiune a bazelor de date, sistemul Delphi utilizează două categorii de componente:

- Componente non-vizuale (ADO Table, ADO Connection, ADO Query, etc.) .
- Componente vizuale - care permit afișarea, editarea și actualizarea articolelor unei baze de date, de exemplu: DBGrid, DBNavigator, DBText, DBMemo, DBComboBox, etc.

- **Interogări SQL**

SQL (Structured Query Language) este un limbaj standardizat de definire și manipulare a datelor aflate într-o bază de date relațională. În conformitate cu modelul relațional, baza de date este percepută ca un set de tabele, relațiile dintre acestea sunt reprezentate de valorile din aceste tabele, iar datele sunt obținute prin crearea unei interogări pe baza datelor din unul sau mai multe tabele de bază.

Dacă dorim să folosim interogări SQL în aplicații, va trebui să ne familiarizăm cu componenta *Query*. Delphi permite aplicațiilor noastre să opereze cu interogări asupra bazei de date prin intermediul acestei componente. *Query* poate încapsula una sau mai multe comenzi SQL, le execută și ne dă posibilitatea de manipula rezultatul prin diverse metode. Comenzile SQL pot fi împărțite în două categorii: cele prin care se obțin seturi de rezultate (cum e comanda SELECT) și cele prin care nu se obține un rezultat afișat, ci se produce o modificare în baza de date (un bun exemplu sunt comenzile UPDATE sau INSERT).

O interogare SQL se comportă în cele mai multe cazuri, ca un filtru aplicat asupra unei table de date. Câteodată, este mai puternică decât un filtru, deoarece poate obține date din mai multe table (prin joncțiune) sau poate obține subseturi de câmpuri sau înregistrări din anumite table.

- **Interogări OLAP**

OLAP (On-Line Analytical Processing) este o tehnologie foarte răspândită pentru analiza multidimensională a afacerii. Aceasta se bazează pe modelul multidimensional al datelor, care va fi dezvoltat în cele ce urmează. Conceptul de OLAP a fost descris prima dată în 1993 de către Dr. E.F. Codd, cunoscutul cercetător în domeniul bazelor de date și inventatorul modelului relațional al bazelor de date. În prezent, suportul pentru OLAP este implementat în diverse baze de date și instrumente.

OLAP este o componentă cheie pentru crearea depozitelor de date. *Data warehousing* este procesul prin care se colectează date din sisteme informaționale diferite și se obțin agregări disponibile pentru utilizatorii finali și pentru rapoarte. Depozitele de date pot fi folosite pentru descrierea acestor colecții de date și a informațiilor agregate care sunt puse la dispoziția utilizatorilor.

Cât despre Delphi, versiunea Enterprise vine cu un set de componente OLAP pentru aplicațiile client, care se pot găsi în cadrul grupului *Decision Cube* din paleta de componente.

4. DEZVOLTAREA ÎN TERMENII FILOZOFIEI RAD A UNUI SISTEM INFORMATIC PENTRU ASISTAREA PROCESULUI DE ANALIZĂ ECONOMICO-FINANCIARĂ

Sistemul informatic a fost conceput ca suport al deciziilor managementului. În atingerea acestui țel, aplicația este creată pentru a urmări patru etape principale:

Etapa 1: Preluarea informațiilor din situațiile financiare ale întreprinderii (bilanț financiar și note explicative – câteva date din Contul de profit și pierdere sau Fișele de cont care nu se regăsesc în Bilanțul financiar, dar care servesc la calcularea unor indicatori ai poziției financiare a întreprinderii).

Etapa 2: Prelucrarea datelor introduse în vederea obținerii valorilor indicatorilor poziției financiare. Indicatorii analizați sunt: structura patrimonială, structura financiară, bonitatea financiară, echilibrul financiar și gestiunea resurselor financiare.

Etapa 3: După prelucrarea datelor și obținerea indicatorilor, urmează etapa de interpretare a valorilor, pe baza unor valori de referință. Această etapă poate fi denumită generic “analiza statică a indicatorilor”.

Etapa 4: După analiza statică urmează analiza în dinamică a indicatorilor, în care se compară valorile indicatorilor din 2 ani diferiți, se obțin indicii de creștere și se interpretează evoluția întreprinderii de la un exercițiu financiar la altul. În această etapă, pentru o mai bună reprezentare a analizei, dinamica indicatorilor se prezintă și grafic.

- **Structura bazei de date**

Programul folosește o bază de date relațională, creată în Microsoft Access 2007 și este structurată în 10 table, după cum urmează:

- **Bilanț financiar** – conține datele din bilanțul financiar;
- **Note explicative** – conține informații suplimentare din Contul de Profit și Pierdere și din Balanță necesare calculării unora din indicatori;
- **Structura financiară pe termene** – conține valoarea indicatorilor de structură financiară pe termen;
- **Structura financiară după proveniență** – conține valoarea indicatorilor de structură financiară după proveniență;
- **Structura patrimonială** – conține valoarea indicatorilor de analiză a structurii patrimoniale (a activului);

- **Lichiditatea** – conține valoarea indicatorilor de lichiditate;
- **Solvabilitatea** – conține valoarea indicatorilor de solvabilitate;
- **Echilibrul financiar pe termen lung** – conține valoarea indicatorilor pentru analiza echilibrului financiar pe termen lung;
- **Echilibrul financiar global** – conține valoarea indicatorilor pentru analiza echilibrului financiar global;
- **Gestiunea resurselor financiare** – conține valoarea indicatorilor pentru analiza gestiunii resurselor financiare de care dispune întreprinderea.

➤ **Structura interfeței**

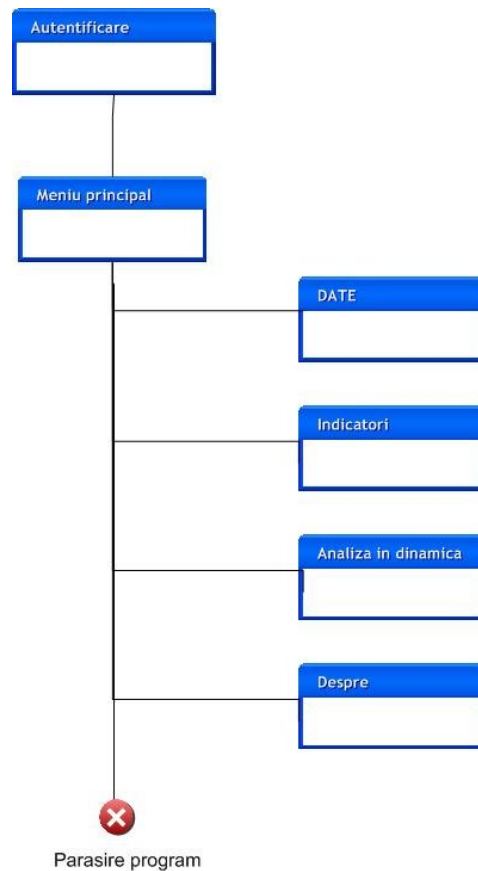


Figura 1. Interfața aplicației

Interfața programului este compusă din 23 de forme. Așa cum se poate observa în figura de mai sus (figura 1), structura interfeței este formată din:

- O formă de autentificare a utilizatorilor;
- O formă principală care conține o imagine și meniul principal al programului.

Forma de autentificare:



Figura 2. Forma de autentificare

Forma principală:



Figura 3. Forma principală

Meniul Date cuprinde următoarele submeniuri: *Adăugare bilanț*, *Adăugare note explicative* și *Modificare/ Ștergere date*.

Meniul Indicatori cuprinde : *Structura patrimonială*, *Structura financiară*, *Bonitatea financiară*, *Echilibrul financiar* și *Gestiunea resurselor financiare*.

Meniul Analiza în dinamică cuprinde: *Structura patrimonială*, *Structura financiară*, *Bonitatea financiară*, *Echilibrul financiar* și *Gestiunea resurselor financiare*.

Meniul Despre face legătura la forma ce prezintă informații succinte referitoare la program.

➤ **Preluarea informațiilor din situațiile financiare**

Acest lucru presupune:

- Adăugarea datelor din Bilanțul financiar

- Adăugarea datelor din Notele explicative
- Modificarea datelor din cele două situații (modificare, ștergere)
- Vizualizarea datelor din Bilanțul financiar

Adăugarea datelor din Bilanțul financiar se realizează prin forma de la figura 4.

Figura 4. Introducerea datelor din bilanțul financiar

Vizualizarea datelor din bilanț se realizează accesând butonul “Vizualizează bilanț” care ne conduce la forma:

Figura 5. Vizualizarea bilanțului

După selectarea anului, se obține bilanțul financiar într-o formă care poate fi vizualizată sau imprimată:

		ANUL: 2005
1	Imobilizari necorporale	1000
2	Imobilizari corporale	345556
3	Imobilizari financiare	0
4	ACTIVE IMOBILIZATE - TOTAL	346556
5	Stocuri	111111
6	Creante	142222
7	Disponibilitati si investii fn. pe termen scurt	50000
	a) Investitii financiare pe termen scurt	35000
	b) Disponibilitati	15000
8	ACTIVE CIRCULANTE - TOTAL	303333

Figura 6. Vizualizarea bilanțului financiar

Adăugarea datelor din Notele explicative se realizează cu ajutorul formei din figura 7:

Figura 7. Adăugarea datelor din notele explicative

Modificarea sau ștergerea datelor din bilanț sau notele explicative se realizează prin accesarea formei următoare, care are 2 butoane prin care se jonglează între cele două situații, în funcție de preferințele utilizatorului (figura 8):

Figura 8. Modificarea datelor din bilanț și notele explicative

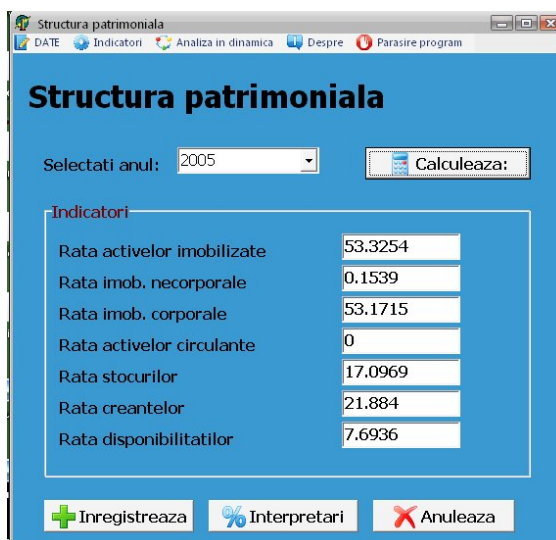
➤ Prelucrarea datelor. Obținerea indicatorilor

Această etapă cuprinde următoarele operații:

- Calcularea valorii indicatorilor aferenți analizei poziției financiare a întreprinderii;
- Înregistrarea indicatorilor calculați în baza de date.

Indicatorii analizați sunt: structura patrimonială, structura financiară, bonitatea financiară, echilibrul financiar și gestiunea resurselor financiare.

Ca exemplificare, calcularea valorii indicatorilor pentru structura patrimonială se realizează cu ajutorul formei de la figura 9:



Indicatori	
Rata activelor imobilizate	53.3254
Rata imob. necorporale	0.1539
Rata imob. corporale	53.1715
Rata activelor circulante	0
Rata stocurilor	17.0969
Rata creantelor	21.884
Rata disponibilitatilor	7.6936

Figura 9. Indicatorii de structură patrimonială

➤ Analiza statică a indicatorilor

Prin accesarea butonului „Interpretări”, se afișează interpretările indicatorilor, în funcție de niște niveluri de referință, care se încarcă sub formă de raport ce poate fi imprimat.

➤ Analiza în dinamică a indicatorilor

Această ultimă etapă presupune:

- Preluarea indicatorilor deja calculați, din doi ani diferiți, și calcularea indicilor de creștere (cuantificarea modificării lor în procente);
- Realizarea interpretărilor;
- Centralizarea datelor întrun raport și vizualizarea dinamicii indicatorilor prin intermediul unui grafic.

Calcularea indicilor de creștere pentru indicatorii structurii financiare se realizează ca în următoarea formă:

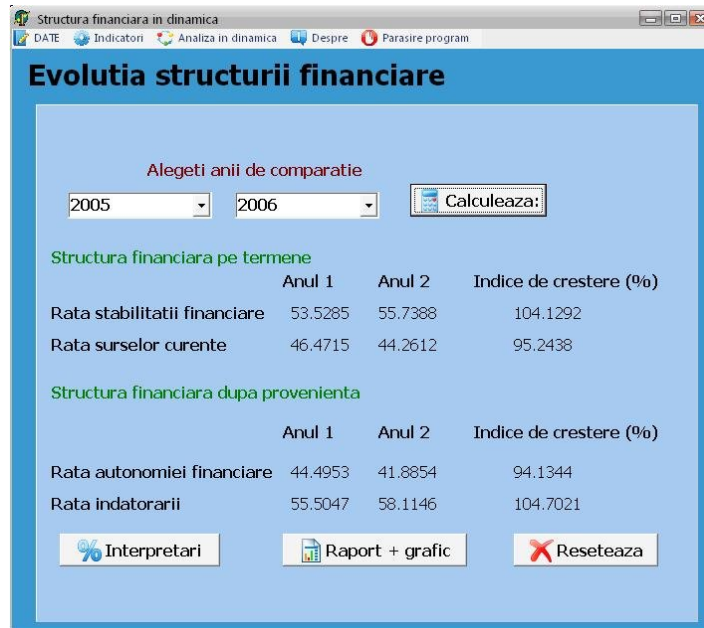


Figura 10. Analiza în dinamică a structurii financiare

Vizualizarea prelucrărilor se realizează prin apelarea butonului “Raport + grafic” din forma de la figura 11, care ne conduce la următoarea fereastră:



Figura 11. Raport comparativ

După selectarea anilor de comparație, butonul “Generare” ne va conduce la raportul propriu-zis (figura 12), care conține în partea de jos și graficul aferent evoluției indicatorului de la o perioadă la alta:

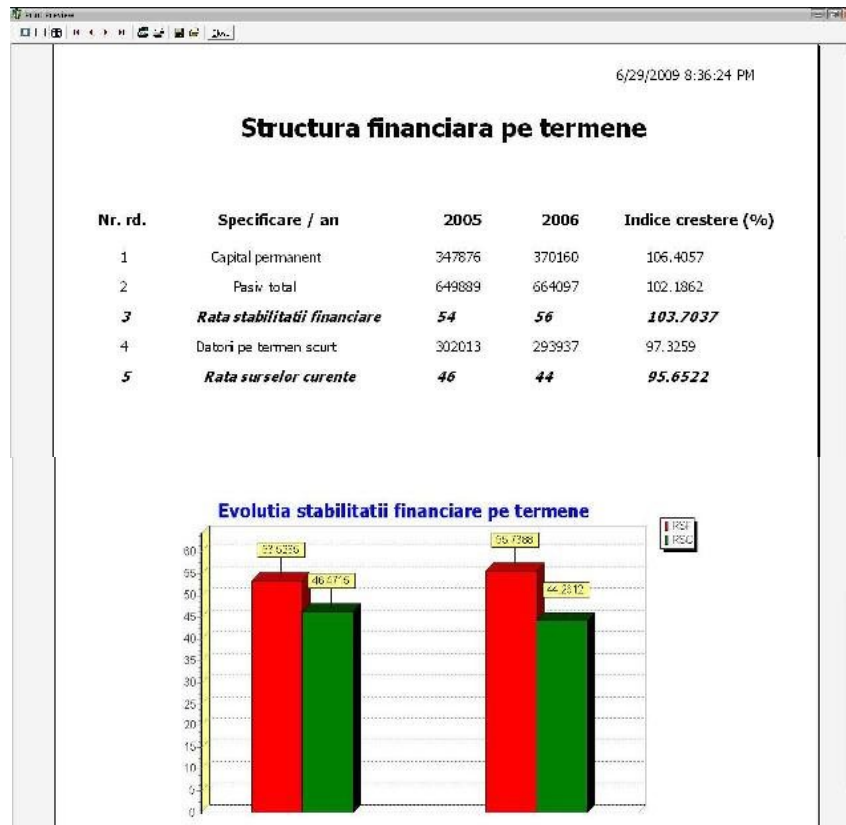


Figura 12. Vizualizare raport structura financiară pe termene

CONCLUZII

Dezvoltarea de aplicații în medii RAD (Rapid Application Development) a cunoscut un real succes încă de la prima apariție. Putând fi realizate în două etape separate una de alta dar care se desfășoară în paralel, proiectarea interfeței și dezvoltarea aplicației propriu-zise, aplicațiile dezvoltate în acest mediu sunt de cele mai multe ori de mică anvergură, realizate cu echipe de dezvoltatori restrânse, uneori chiar la o singură persoană.

Sistemul informatic a fost conceput pentru a fi un **suport decizional**. Astfel, aplicația preia informații din situațiile financiare, calculează indicatorii economici, pe principalele criterii de analiză (structura patrimonială, structura financiară, bonitatea financiară, echilibrul financiar și gestiunea resurselor financiare), apoi realizează interpretări atât la nivel static (întrun singur an) cât și la nivel dinamic (comparația între doi ani diferiți). Așa cum se poate observa, acești indicatori, per ansamblu, ne ajută să formulăm concluzii referitoare la poziția financiară a întreprinderii. În realizarea acestui scop, am apelat la componentele vizuale și non-vizuale ale mediului Delphi, cu precădere a celor de manipulare a bazei de date, precum și la interogările SQL pentru selectarea datelor necesare prelucrărilor realizate.

Ca o idee de îmbunătățire a acestei aplicații, de continuare a studiului realizat asupra mediilor RAD, se pot include noi indicatori în cadrul programului, analiza extinzându-se și asupra contului de profit și pierdere, în întregime sa. De asemenea, pentru situații mai detaliate privind diverse aspecte ce pot influența dinamica indicatorilor și, în consecință, evoluția generală a întreprinderii, se pot utiliza **interogări OLAP**, ca o completare a interogărilor SQL deja folosite.

BIBLIOGRAFIE

1. Boldea, M. (2006), *Delphi – programarea vizuală*, Editura Mirton, Timisoara
2. Buglea, Al. (2005), *Analiza financiară – Concepte și studiu de caz*, Editura Mirton, Timișoara
3. Connelly, Th., Begg, C., Strachan, A. (2001), *Baze de date – proiectare, implementare, gestionare*, Editura Teora, București
4. Elmanova, N. (2000) „*Borland Conference 2000: Implementing OLAP in Delphi Applications*”, publicat în Wordware Publishing Inc., 2000
5. Oprea D., Fotache M, Airinei D. (2002), *Sisteme informaționale pentru afaceri*, Editura Polirom, Iași
6. *** (2001), *Borland Delphi 6 for Windows: Developer's Guide. Delphi*, Borland Software Corporation
7. <http://www.olapcouncil.org>