

Aspecte comparative între bazele de date relaționale și orientate obiect

Asist. Cristina IONIȚĂ, prep. Petrișor OPREA
Catedra de Informatică Economică, A.S.E. București

În ultimul deceniu, abordarea orientată obiect a înregistrat o extindere deosebită nu numai în domeniul programării sau al analizei și proiectării sistemelor ci și în domeniul bazelor de date (BD). Facilitatea de a defini și manipula ușor obiecte complexe, modelarea mai naturală a lumii reale, posibilitatea de reutilizare a codului și nu în ultimul rând creșterea performanțelor aplicațiilor dezvoltate au stârnit interesul specialiștilor în baze de date, rezultatele neîntârziind să se concretizeze în domenii ca multimedia, sisteme informatice geografice, medii CASE, CAD/CAM. În domeniul gestiunii economice însă bazele de date orientate obiect (BDOO) au pătruns mai puțin, supremația continuând să fie deținută de bazele de date relaționale (BDR), deși limitele acestora sunt unanim recunoscute. În acest domeniu se concretizează tendința de a promova soluțiile hibride, relațional – orientate pe obiecte, care să îmbine avantajele oferite de ambele abordări. Ne propunem să analizăm comparativ BDR și BDOO și să punem în evidență caracteristicile soluțiilor hibride. Vom supune analizei aspectele calitative ale abordării relaționale, respectiv OO în realizarea BD. Comparația se face între modele de referință pentru cele două abordări, respectiv Codd și Ullman pentru BDR și Date și Kim pentru BDOO, făcând abstracție, pentru început, de extensiile acestor modele implementate de o serie de sisteme comerciale. Setul de metrici utilizat cuprinde: structura modelului, implementarea restricțiilor de integritate, semantica modelului, coerența între etapele de realizare a bazei de date, funcționalitatea limbajelor. Specificul abordării OO este dat de specializare/generalizare, moștenire și încapsulare, proprietăți care sunt caracteristice întregii tehnologii OO: analiză, proiectare, programare, inclusiv baze de date orientate obiect (BDOO). În cadrul analizei comparative ne vom axa mai mult pe caracteristicile care sunt strâns legate de tehnologia BD și care stau la baza setului de metrici enunțat.

Cuvinte cheie: model de date, baze de date, relațional, orientat obiect, soluții hibride.

1. Structura modelului

Din punct de vedere structural, relațiile, atributele, tuplurile și tipurile elementare de date sunt singurele construcții ale modelului relațional. Simplitatea modelului relațional bazat pe relații de forma tabelor ce conțin rânduri și coloane ale căror elemente sunt restricționate la valori elementare permite caracterizarea acestui model ca fiind de tip *bidimensional* sau plat. Identificarea tuplurilor se realizează pe baza valorii unuia sau mai multor atribute care formează cheia relației. În plus, realizarea legăturilor între relații, respectiv asigurarea referințelor între tupluri se face tot prin valoare. De aceea, modelul relațional este caracterizat ca fiind *bazat pe valoare*.

În cadrul modelului OO construcția de bază o constituie clasa. La nivel de tip, o clasă poa-

te avea unul sau mai multe atribute. La nivel de instanță, o clasă poate avea unul sau mai multe obiecte. Până la acest punct, aparent, nu există mari diferențe între cele două modele de date, întrucât relațiile, tuplurile și atributele din modelul relațional corespund clasei, obiectelor și respectiv atributelor din modelul OO. Totuși, există diferențe semnificative în ceea ce privește modul de identificare a obiectelor și de realizare a referințelor între obiecte. Astfel, în modelul OO sistemul asigură automat fiecărui obiect un identificator. Orice obiect va fi referit prin identificatorul său și de aceea modelul OO este caracterizat ca fiind *bazat pe identitate*. În plus, atributele din modelul OO pot fi de orice tip, atât tipurile de date asigurate de sistem cât și tipurile de date abstracte definite de utilizatori. Un tip abstract de date poate conține ca atribute alte tipuri abstracte

de date. Ca urmare, putem caracteriza modelul OO ca fiind *multidimensional*.

Diferența între modul de stabilire a legăturii între tupluri este dată de faptul că în modelul OO legătura este *fizică* și este disponibilă în orice moment, în timp ce în modelul relațional legătura este *logică* și stabilită în momentul exploatării. Deși sunt mai costisitoare din punct de vedere al exploatării, legăturile logice între tupluri conferă o flexibilitate sporită modelului relațional pentru că pot fi oricând reconstruite fără să fie necesară reorganizarea datelor din BD. De aceea, modelul relațional este mai potrivit pentru

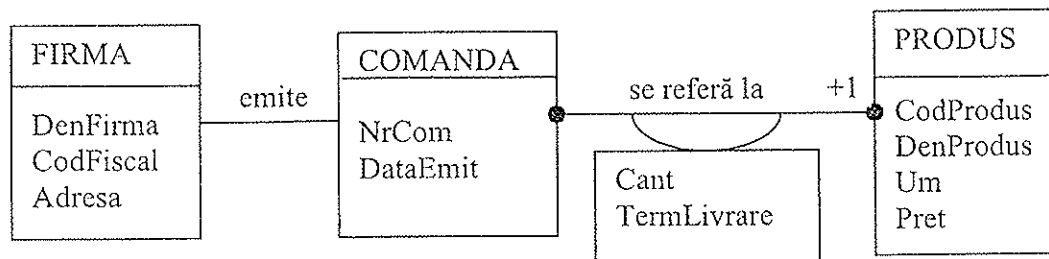
modelarea activității unei întreprinderi care înregistrează frecvente schimbări organizatorice ce nu pot fi prevăzute.

Modelul relațional este un model logic, fundamentat teoretic, care permite modelarea și testarea BD înainte ca aceasta să fie implementată, spre deosebire de modelul OO care este un model fizic, realizat din considerente practice.

În figura 1 este prezentată comparativ schema conceptuală a BDR și respectiv BDOO pentru activitatea de gestiune a comenzilor de desfacere.

FIRMA(CodFirma, DenFirma, CodFiscal, Adresa)
 COMANDA(Nrcom, DataEmit, CodFirma)
 RAND_COMANDA(NrCom, CodProduce, Cant, TermLivrare)
 PRODUS(CodProduce, DenProduce, Um, Pret)

a) Schema conceptuală a BDR



b) Schema conceptuală a BDOO

Fig. 1. Reprezentarea comparativă a schemei BD

2. Implementarea restricțiilor de integritate

Restricțiile de integritate semantice sunt legate de comportamentul (regulile) sistemului modelat și țin cont de semnificația valorilor din BD. Ele sunt mult prea generale pentru a putea fi prevăzute și incluse în definiția bazei de date și sunt, de regulă, gestionate în afara modelului de date, fie în manieră procedurală, fie în manieră declarativă. De aceea, vom supune analizei numai restricțiile structurale de integritate care sunt generice și sunt strâns legate de modelul datelor.

Modelul relațional nu asigură intern, respectarea restricțiilor structurale și de aceea ele vor fi asigurate extern, în manieră declara-

tivă sau procedurală.

În cadrul modelului OO necesitatea restricțiilor structurale este eliminată întrucât sistemul asigură fiecărui obiect un identificator unic iar utilizarea identificatorului de obiect pentru implementarea legăturilor ne asigură că obiectul referit există, fie referința este nulă.

În concluzie, referitor la respectarea restricțiilor de referențialitate modelul OO este superior modelului relațional pentru că respectarea restricțiilor este asigurată automat intern, prin structura modelului și nu prin specificări externe, în manieră declarativă sau procedurală, așa cum se procedează în cazul modelului relațional.

3. Semantica modelului

Normalizarea ascunde și distruge semantica datelor. Relațiile normalizate nu corespund, de regulă, unui obiect din lumea reală pentru că ele sunt determinate de cunoștințele privind modul de manipulare a datelor, cunoștințe reprezentate de dependențele funcționale între atribute.

Caracteristicile structurale ale modelului OO, identificatorul obiectelor și tipurile abstracte de date, îmbogățesc semantica modelului și permit o modelare naturală a obiectelor din lumea reală. Sunt posibile următoarele construcții semantice: referințe simple, referințe multiple, referințe în dublu sens între obiecte, obiecte compuse și construcții de tip colecții.

În lumea reală multe obiecte sunt complexe fiind compuse din alte obiecte care la rândul lor pot fi compuse. Legătura între sub-obiect și obiect este o cuplare strânsă spre deosebire de cuplarea slabă modelată prin referințele simple, referințele multiple și dubla referință. Această legătură este implementată în modelul OO prin înglobarea directă a sub-obiectelor în obiect. Legătura de compunere îmbogățește semantica modelului OO, permițând reprezentarea naturală a obiectelor din lumea reală. Totodată, legătura de compunere între obiecte asigură automat respectarea restricției de existență a sub-obiectelor întrucât, ori de câte ori obiectul este creat, copiat sau șters, sub-obiectele componente vor fi și ele create, copiate respectiv șterse.

Caracteristica de model plat, bidimensional a modelului relațional exclude posibilitatea de a defini tupluri (entități) compuse. Întrucât forma normală unu (FN1) interzice existența atributelor compuse în cadrul unei relații, sau altfel spus posibilitatea de a defini relații în cadrul unei relații, entitățile componente pot exista numai ca relații separate. Unul din dezavantajele dispersării entităților componente îl constituie dificultatea de a vizualiza (reconstitui) complet entitatea complexă. Putem aprecia că semantica obiectelor complexe este artificial reprezentată în modelul relațional pentru că acesta nu permite înglobarea directă a unui tuplu în alt tuplu. Un alt dezavantaj este dat

de efortul suplimentar cerut pentru a asigura respectarea restricției de existență a entităților componente care să permită crearea, copierea și ștergerea entităților componente ori de câte ori este creată, copiată și respectiv ștearsă entitatea complexă. Asigurarea dependenței de existență se realizează în manieră declarativă, utilizând instrucțiunile limbajului de manipulare a bazei de date.

Prin gruparea obiectelor de același tip putem forma alte tipuri de date numite colecții. Majoritatea BDOO oferă astfel de construcții cum ar fi masivul, lista, setul. Diferențele între acești constructori sunt date de posibilitatea și modul de ordonare a obiectelor. Într-un masiv obiectele sunt ordonate prin indecși în cadrul unui interval. Într-o listă obiectele sunt ordonate prin indecși fără interval. În cadrul unui set, obiectele nu sunt ordonate.

Modelul relațional implementează numai construcția de tip set (relația) în care obiectele nu sunt ordonate. Dacă o aplicație are nevoie de obiecte într-o anumită ordine, atunci fie sistemul de gestiune a bazei de date va furniza aplicației obiectele sortate, fie aplicația va trebui să sorteze mai întâi obiectele.

În concluzie, modelul OO este mai bogat din punct de vedere semantic și asigură o modelare mai naturală a lumii reale, în timp ce modelul relațional încorporează o semantică săracă și realizează o modelare artificială a lumii reale.

4. Coerența între etapele de realizare a bazei de date

Vom analiza coerența pe care fiecare model de date o asigură pentru trecerea de la o etapă la alta în procesul de realizare a bazei de date.

În cazul modelului OO, în etapa de analiză sistemul țintă este analizat și modelat sub formă de clase, obiecte, asocieri, ierarhii de clase etc. În etapa de proiectare, clasele și obiectele identificate la nivel conceptual sunt utilizate pentru specificarea detaliată, la nivel logic și fizic al claselor și obiectelor, în concordanță cu restricțiile impuse de soluția de implementare. În cadrul etapei de imple-

mentare distingem două activități importante și anume implementarea bazei de date și implementarea aplicațiilor. Implementarea BD presupune codificarea definițiilor de clase și obiecte. De remarcat faptul că o parte din funcțiile sistemului sunt implementate deja în cadrul claselor sub forma metodelor și pot fi utilizate de către aplicații. Putem aprecia că întregul proces de realizare a BDOO se bazează pe paradigma orientată obiect și deci se asigură o bună coerență între etape.

În cazul modelului relațional apar o serie de probleme legate de asigurarea coerenței. Întrucât puterea de modelare semantică a modelului relațional este slabă, în etapa de analiză a sistemului existent se utilizează, de obicei, modelul entitate asociere (ER). În etapa de proiectare, însă, modelul utilizat este cel relațional. Ca urmare, este necesară convertirea entităților, a asocierilor, a atributelor și a restricțiilor structurale din modelul ER în relații, asocieri, attribute și restricții ale modelului relațional. Relațiile rezultate în urma conversiei sunt supuse mai departe procesului de normalizare pentru eliminarea anomaliilor de actualizare generate de redundanța datelor.

În concluzie, utilizarea modelului OO asigură o coerență mai bună între etapele de realizare a BD, în timp ce în cazul modelului relațional coerența este mai slabă pentru că se utilizează diferite modele între care trebuie realizată conversia.

5. Funcționalitatea limbajelor

Limbajele utilizate în gestiunea bazelor de date cuprind limbajul de definire a datelor (LDD) și limbajul de manipulare a datelor (LMD).

BDR utilizează, în general, atât ca LDD cât și ca LDM limbajul SQL care face parte din categoria limbajelor declarative. Limbajul SQL și, în general, limbajele de manipulare a BDR au drept caracteristică posibilitatea de prelucrare a unui set de date la un moment dat și de realizare a cererilor ad-hoc, caracteristici care au asigurat în mare parte succesul BDR. Întrucât LMD pentru BDR nu oferă suport pentru realizarea unor prelucrări complexe asupra datelor, se aplează adesea la limbaje de programare de nivel înalt care, însă, prelucrează o singură înregistrare la un moment dat. Apare astfel o nepotrivire de impedanță care crează dificultăți în asigurarea interfeței între limbaje.

BDOO utilizează, de regulă, atât ca LDD cât și ca LMD un limbaj de programare OO (C++, Smalltalk, Gemstone etc.). În aparență, descrierea bazei de date este mai dificil de realizat, acest lucru însă se datorează în mare parte faptului că modelul OO este mai complex, mai bogat din punct de vedere semantic și nu limbajului utilizat. În cazul BDOO limbajele OO utilizate pentru manipularea BD oferă suport pentru realizarea prelucrărilor complexe. În tabelul care urmează sunt sintetizate rezultatele comparației între BDR și BDOO.

Criteriu de comparație	BDR	BDOO
Structura modelului	<ul style="list-style-type: none"> • Simplă și flexibilă • Model bidimensional bazat pe valoare • Model logic, fundamentat teoretic 	<ul style="list-style-type: none"> • Complexă și mai dificil de modificat • Model multidimensional bazat pe identitate • Model fizic, realizat din considerente practice
Implementarea restricțiilor structurale de integritate	<ul style="list-style-type: none"> • Codificate extern în cadrul aplicațiilor 	<ul style="list-style-type: none"> • Asigurate intern prin structura modelului
Semantica modelului	<ul style="list-style-type: none"> • Slabă • Realizează o modelare artificială a lumii reale 	<ul style="list-style-type: none"> • Bogată • Asigură o modelare naturală a lumii reale
Coerența între etapele de realizare a BD	<ul style="list-style-type: none"> • Slabă • Se utilizează modele diferite ce necesită conversii 	<ul style="list-style-type: none"> • Bună • Se utilizează numai paradigma OO
Funcționalitatea limbajelor	<ul style="list-style-type: none"> • Prelucrări simple • Cereri ad-hoc • Prelucrări pe seturi de tupluri 	<ul style="list-style-type: none"> • Prelucrări complexe • Cereri pe criterii predefinite • Prelucrări tuplu cu tuplu

Concluzii

Se poate, putem aprecia că, în ansamblu, modelul OO este superior modelului relațional, în special în privința bogăției semantice a modelului și a performanțelor oferite în exploatarea BD. Principalele dezavantaje ale modelului OO sunt legate de lipsa unei fundamentări teoretice care să permită studierea proprietăților și comportamentului BD înainte ca aceasta să fie implementată. De asemenea, accesul navigațional la obiectele din cadrul modelului realizat prin pointeri (identificatorul obiectului) și nu prin valoare exclude posibilitatea realizării interogărilor ad-hoc în BD, regăsirea obiectelor putându-se realiza numai pe căile prestabilite. Maturizarea tehnologiei OO în domeniul BD va oferi, probabil, în viitor soluții acestor probleme, dar se conturează tendința ca cele două tehnologii, relațională și OO, să fuzioneze într-o singură tehnologie care să îmbine funcționalitățile și performanțele fiecăreia.

Bibliografie

- ☐ Stefano Ceri, Piero Fraternali, Designing Database Applications with Objects and Rules, Addison-Wesley, 1997
- ☐ C.J.Date, An Introduction to Database Systems, Addison-Wesley, 1995
- ☐ Ian Graham, Object Oriented Methods, Addison-Wesley, 1994
- ☐ Won Kim, Modern Database Systems. The Object Model, Interoperability, and Beyond, ACM Press, 1995
- ☐ Dah-Jie Liou, An Object-oriented Database Approach to Manufacturing Information Systems with Emphasis on Production Management, Am Arbor, 1996
- ☐ I.Lungu, C.Bodea, G.Bădescu, C.Ioniță, Baze de date. Organizare, proiectare și implementare, Ed. ALL, București, 1995
- ☐ J.D.Ullman, Principles of Database Systems, Computer Science Press, 1985