

# Sisteme de interogare folosite în bazele de date pentru prelucrări grafice

Sef lucrări dr. ing. Marius Dorian Zaharia,  
Catedra de Calculatoare, Universitatea POLITEHNICA București

*Lucrarea prezintă câteva categorii de cereri specifice bazelor de date pentru prelucrări grafice. Sunt date exemple din limbajele de definire a datelor sau limbajele de specificare a cererilor asociate unor sisteme de gestiune a bazelor de date utilizate cu precădere în domeniul aplicațiilor grafice.*

**Cuvinte cheie:** bază de date, bază de date spațială, sistem de interogare, model de date,

## 1. Categoriile de cereri specifice bazelor de date pentru prelucrări grafice

Sistemele de interogare ale bazelor de date folosite în prelucrări grafice sunt caracterizate prin următoarele elemente: specificarea cererilor poate conține predicate de căutare vagi pentru identificarea unor entități cu caracteristici similare altora; există cereri ce se referă la atributele tipurilor grafice (culoare, textură), cereri spațiale, cereri vizuale bazate pe exemple sau liste de selecție interactivă (pick list), cereri în care se asociază ponderi (priorități) predicatelor componente. În prelucrarea cererilor, mai ales în cazul celor dependente de context sau al celor de similitudine, faza de optimizare a execuției este deosebit de complexă. Rezultatele unei cereri (dacă sunt mai multe) sunt ordonate în funcție de ponderile asociate termenilor cererii sau valorilor atributelor obiectelor din baza de date. Spre deosebire de sistemele de baze de date convenționale în care după obținerea rezultatelor unei cereri procesul de interogare ia practic sfârșit, în unele sisteme avansate de regăsire a informației apare conceptul de "feed-back de relevanță" (pentru utilizator), reacția de acceptare sau nu a rezultatelor cererii de către utilizator este folosită pentru a recompune cererea și a o supune din nou spre rezolvare sistemului de interogare.

Cererile în sistemele de baze de date

spațiale au în general nivele de abstractizare diferite. La nivelele superioare, interogările cele mai frecvente se referă la detectarea unor șabloane în datele de analizat sau la anticiparea caracteristicilor datelor situate într-o anumită regiune sau la un anumit moment de timp. La nivele inferioare de abstractizare cererile specifice ar putea realiza: conversii între diferite formate de date, schimbarea caracteristicilor de vizualizare, îmbunătățirea unor parametri ai imaginilor.

Cererile situate la un nivel de abstractizare intermediar pot fi rezolvate cel mai bine folosind sisteme de gestiune a bazelor de date. Aceste cereri consideră datele ca fiind organizate ierarhic (în straturi). Fiecare strat este partiționat în zone (colecții de locații cu valori comune ale atributelor). De exemplu în cazul sistemelor de proiectare asistată de calculator a plăcilor cu circuite imprimate, un strat ar corespunde traseelor iar altul pastilelor sau găurilor de trecere iar într-un sistem de informație geografică straturile ar putea descrie modul de utilizare al parcelelor de teren, tipul de sol, rețeaua de drumuri etc.

În astfel de sisteme în care datele sunt organizate pe nivele, cererile pot fi clasificate ca: *cereri locale* (acestea se referă la caracteristici aflate la aceleași locații pe nivele diferite) *cereri zonale* (se referă la grupuri de locații situate pe același

nivel care au aceeași valoare de atribut) *cereri de focalizare* (se referă la vecinătățile unei locații situate pe un anumit nivel, vecinătatea poate fi delimitată restricții referitoare la distanță, direcție sau timp). Această ultimă categorie de cereri este echivalentă cu *cererile de domeniu* (range queries) din sistemele de baze de date convenționale (la aceste cereri domeniul în care trebuie să se afle punctul de date este un hiperdreptunghi).

O categorie importantă de cereri specifice bazelor de date spațiale sunt cele de *suprapunere poligonală*; în această situație sunt suprapuse două straturi caracterizate prin atribute spațiale diferite  $\zeta_i$ , în fiecare locație spațială, este aplicată o funcție asupra valorilor atributelor corespunzătoare. Dacă se face o analogie între "straturile cu informații" dintr-o bază de date spațială  $\zeta_i$  relațiile unei baze de date obișnuite, rezultatul unei cereri de tip suprapunere de poligoane este analog cu cel obținut prin efectuarea produsului cartezian al mulțimilor având ca elemente regiunile celor două straturi cu informații.

Cererile zonale pot fi folosite pentru a implementa cazuri particulare de cereri de suprapunere poligonală. De obicei o cerere zonală lucrează cu trei straturi de informație spațială, primele două constituie operandii cererii iar al treilea va conține rezultatul. Informația din primul strat (operand) acționează ca o mască ce va determina partiționarea celui de-al doilea strat în zone. Stratul al treilea va conține rezultatul aplicării cererii zonale tuturor locațiilor din al doilea strat care corespund fiecărei zone de interes. De exemplu, în cazul unei baze de date spațiale ce reține date referitoare la culturile agricole dintr-o anumită zonă geografică, o astfel de cerere ar putea fi formulată astfel: "Să se determine cantitatea de apă pluvială medie anuală în zonele unde crește grâu Triticum Durum". Ea ar corespunde operației de intersecție între stratul de informație ce se referă la cantitatea de apă/m<sup>2</sup>/an  $\zeta_i$  stratul de

informație referitoare la zonele în care crește grâu Triticum Durum. În final se va efectua un calcul de mediere de forma  $\sum_i f_i A_i / \sum_i A_i$ , unde sumarea se face peste toate zonele unde crește grâu T.D.  $\zeta_i$  care au o singură valoare a indicatorului "precipitație medie anuală" egală cu  $f_i$ ;  $A_i$  reprezintă aria unei astfel de zone.

## 2. Limbaje de interogare specifice bazelor de date pentru prelucrări grafice

În proiectarea și dezvoltarea unui limbaj de interogare al bazelor de date spațiale au fost utilizate abordări bazate pe extinderea unor limbaje ca SQL (în care structura fundamentală de interogare este organizată sub forma unui șablon SELECT ... FROM ... WHERE, clauze care corespund respectiv operațiilor de proiecție, produs cartezian și selecție, caracteristice algebrilor relaționale) QUEL (limbaj bazat pe calculul relațional cu tuple în care clauzele folosite sunt RANGE, FROM, WHERE) sau QBE (limbaj de interogare bazat pe exemple).

În limbajele de interogare specifice bazelor de date spațiale, conceptele și reprezentările sub formă tabulară a datelor, caracteristice bazelor de date relaționale nu sunt preponderente, un rol mai important îl joacă (datorită naturii geometrice a unei părți a datelor) dezvoltarea unor metode simbolice de reprezentare (iconițe) sau a altor metode de interacțiune. De aceea, în contextul bazelor de date spațiale, limbajul de interogare trebuie să permită nu numai regăsirea unor informații ci și descrierea unui mod adecvat de prezentare a datelor spațiale.

Un sistem de interogare pentru o bază de date spațială trebuie ca, în afara caracteristicilor specifice limbajelor clasice de interogare, să satisfacă următoarele condiții:

- posibilitatea de a defini un *tip abstract de date spațial* pentru ca utilizatorii să

poată referi datele spațiale la un nivel independent de codificarea internă a acestora

- rezultatul unei cereri trebuie să poată fi afișat în formă grafică (folosind atribute de culoare, intensitate, simboluri, șabloane de hașură) care este adecvată pentru analiza datelor spațiale și vizualizarea caracteristicilor obiectelor spațiale ale bazei de date care sunt reprezentate prin atribute nespaciale
- prezentarea grafică a informației trebuie să permită și afișarea *contextului spațial* (i.e. o informație a cărei afișare nu a fost cerută în mod special dar care este necesară pentru a putea interpreta corect rezultatul cererii)
- trebuie să existe mecanisme de control al utilizatorului asupra conținutului unui desen de prezentare a datelor spațiale
- corespondența între clasele obiectelor spațiale și reprezentările lor grafice este realizată prin intermediul unei "legende" grafice și a unor etichete asociate obiectelor spațiale
- mecanismele de vizualizare a datelor spațiale trebuie să permită determinarea dimensiunii reale a obiectelor desenate precum și restrângerea zonei de căutare a unei informații.

### 2.1. Extinderea limbajelor de interogare cu facilități orientate spre obiecte

După modelul unor limbaje de programare consacrate C, LISP, PASCAL care au primit extensii orientate spre obiecte (C++, CLOS sau Borland PASCAL), limbajul de interogare/manipulare a datelor SQL a primit extensii orientate spre obiecte. Astfel standardul SQL3 introduce caracteristici specifice bazelor de date avansate cum ar fi: extensii OO, extensii pentru specificarea restricțiilor de integritate, suport pentru recursivitate, puncte de salvare intermediare.

Alături de tipurile de date predefinite

(INTEGER, FLOAT, CHARACTER, DATE, TIMESTAMP) au apărut construcții de limbaj pentru specificarea tipurilor abstracte de date definite de utilizator. De exemplu, construcția

```
CREATE TYPE imagine
WITH OID
UNDER grafica
(PRIVATE
size INTEGER,
resolution FLOAT,
content PCX,
PUBLIC
CONSTRUCTOR PROCEDURE
imag(I imagine, cale VARCHAR);
DESTRUCTOR PROCEDURE
d_imag(I imagine);
ACTOR FUNCTION
get_size(I imagine) returns integer;
...
);
```

specifică numele tipului abstract de date, faptul că fiecare obiect instanță al acestui tip va avea un identificator unic generat de sistem; obiectul poate fi referit prin acest identificator. Clauza PUBLIC este analoagă celei din C++ (atributele unei instanțe TAD pot fi PUBLIC, PRIVATE sau PROTECTED). Clauza UNDER specifică ierarhia de tipuri (faptul că TAD-ul "imagine" moțtenezte de la TAD-ul "grafică"). Este posibilă moțtenirea multiplă i.e. un tip poate prelua proprietăți de la mai multe alte tipuri.

Standardul definește trei categorii de rutine care pot fi asociate unui TAD: constructorii (inițializează instanțe ale tipului abstract de date), destructorii (eliberează resurse alocate unor instanțe ale TAD-ului) și actorii (care îndeplinesc orice altă operație asupra instanțelor TAD-ului). Odată definit, un tip abstract de date poate fi folosit în declarații de tabele CREATE TABLE specifice limbajului de definire a datelor (DDL). În SQL3 baza de date conține la nivelul superior numai tabele (relații), instanțele TAD-urilor nu pot fi folosite

decât pentru definirea câmpurilor unor relații sau a argumentelor unei rutine.

Standardul ISO/IEC/WG3 definește tipuri de date abstracte pentru a extinde sistemul SQL în domeniul bazelor de date multimedia (SQL/MM) [Khos 96]. Astfel definițiile TAD-urilor formează o ierarhie de clase având ca rădăcină tipul GEO\_GeometricObject; unele dintre subclasele importante sunt GEO\_PointTypes, GEO\_LineTypes, GEO\_AreaTypes, GEO\_VolumeTypes, GEO\_RasterTypes, GEO\_GraphTypes sau GEO\_TextTypes.

De exemplu în SQL3 o celulă 3D se definește ca o subclasă a clasei GEO\_VolumeClass:

```
CREATE TYPE GEO_Celula_3D
  UNDER GEO_VolumeClass,
  (forma GEO_Cell_3DDefinition, __ poate
   __ fi paralelipipedică
   pos GEO_Cell_3DReference NOT
  NULL,
   __lista vecinilor celulei
   xDim REAL,
   yDim REAL,
   zDim REAL
 )
```

O altă abordare pentru a realiza integrarea mecanismelor de baze de date și a celor de programare orientată spre obiecte este *extinderea limbajelor de programare OO cu mecanisme specifice bazelor de date*, aceste extensii pot include facilități ca: interogarea unor baze de date, suport pentru tranzacții sau persistență. De exemplu limbajul OPAL (sistemul GemStone) a extins Smalltalk cu clase referitoare la gestiunea bazelor de date, alte sisteme dezvoltate în aceeași manieră sunt Versant (SGBD orientat spre obiecte al firmei Versant Object Technology) și Object Store. Unii producători (Microsoft, SUN, Objectivity, ServioLogic) au extins și limbajul C++ cu concepte caracteristice bazelor de date orientate spre obiecte.

Alături de cunoscutele caracteristici: tipuri de date abstracte (clase), moștenire (simplă și, în versiuni recente, multiplă), polimorfism și legare dinamică (prin posibilitatea supraîncărcării numelor de funcții sau operatorilor) sau mai recent suportarea de tipuri parametrizate, au fost introduse noi facilități ca:

- posibilitatea definirii de clase persistente prin moștenirea caracteristicii de persistență de la o clasă rădăcină care are metode specifice care să permită crearea unor noi instanțe și stocarea obiectelor persistente (new), distrugerea unui obiect persistent, citirea câmpurilor unui obiect din baza de date și scrierea stării curente a obiectului în baza de date, zăvorărea accesului la obiect pentru a realiza controlul explicit al concurenței de către utilizator.
- definirea de extensii (o extensie este mulțimea tuturor obiectelor instanțe ale unei clase) și specificarea explicită de către programatorul de aplicație dacă sistemul OODB va menține sau nu această extensie. În caz afirmativ, componentele extensiei vor putea fi vizitate și manipulate folosind programe C++.
- definirea de colecții de obiecte persistente (tablouri, mulțimi, liste și dicționare de obiecte) și relații. Instanțele acestor "obiecte colecție" vor fi valori ale unor variabile instanță dintr-o clasă persistentă. Există mecanisme de inserare, ștergere sau actualizare a unui obiect dintr-o colecție de obiecte. De asemeni, obiectele colecției au asociați "iteratori" i.e. funcții pentru a parcurge elementele colecției.

După mai mulți ani de dezvoltare independentă de multe ori fără să acorde atenție interacțiunii și interoperabilității, o serie de producători de baze de date orientate spre obiecte au elaborat standardul ODMG (Object Database Standard). Companiile participante la dezvoltarea

acestui standard sunt Ontos, Versant, O2, Objectivity, Hewlett Packard, POET Software, Intellitic, DEC, Servio, Texas Instruments. Componentele fundamentale ale acestui standard sunt: modelul de date ODMG-93, limbajul de definire a obiectelor (ODL), limbajul de interogare a obiectelor (OQL), legături cu limbaje de programare (C++).

**Modelul de date ODMG** este bazat pe obiecte și permite specificarea identificatorilor de obiect. Categoriile de obiecte sunt grupate în tipuri (obiectele unui tip au o comportare comună și un domeniu comun de stări), comportarea obiectelor este definită prin operații iar starea obiectelor este definită printr-o mulțime de proprietăți (atribute ale obiectului sau relații între un obiect și unul sau mai multe alte obiecte). Obiectelor li se pot da nume (un obiect poate avea mai multe nume dar un singur identificator).

În figura 1 este prezentată arhitectura unei baze de date realizată în conformitate cu standardul ODMG. Aplicația rezultă prin combinarea descrierilor ODL și OML cu rutinele puse la dispoziție de SGBD, rutine ce permit exploatarea bazei de date. Programul sursă este scris într-un limbaj de programare (de obicei C++) extins cu primitive OML inclusiv cele ce permit utilizarea tranzacțiilor sau specificarea de cereri. Declarațiile ce alcătuiesc schema bazei de date pot fi scrise în ODL sau într-o extensie a limbajului de programare în care este realizată aplicația.

Există tipuri de date colecție: mulțimi, liste și tablouri. O colecție poate conține un număr arbitrar de elemente care sunt toate instanțe ale aceluiași tip. Instanțele (cu

**Fig. 1.** Arhitectura unei Baze de Date ODMG

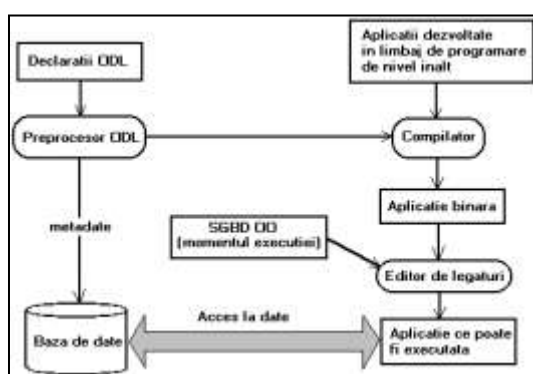
nume) acestor tipuri pot fi utilizate pentru a grupa obiecte. Tipurile obiect sunt legate într-un graf subtip/supertip. Pot fi declarate extensii care să conțină toate instanțele unui tip de date. În anumite cazuri, instanțele individuale ale unui tip de date pot fi identificate în mod unic prin chei formate din una sau mai multe proprietăți ale obiectului. Colecțiile și structurile pot fi compuse în mod arbitrar. Modelul ODMG suportă mulțimi de structuri, structuri de mulțimi, tablouri de structuri etc.

**Limbajul de definire a obiectelor (ODL)** cuprinde construcțiile sintactice necesare specificării caracteristicilor menționate anterior ale modelului de date. Acest limbaj permite ca o aplicație să fie portată într-un nou limbaj de programare fără să se rescrie schema de date. El este destinat definirii unor tipuri obiect care pot fi implementate în diferite limbaje de programare. Sintaxa ODL se bazează pe limbajul de definiție de interfață (IDL) elaborat de grupul CORBA. O schemă specificată în ODL poate fi implementată în orice SGBD orientat spre obiecte care suportă standardul ODMG. ODL-ul poate fi folosit în descrierea datelor și operațiilor unei aplicații, independent de limbajul de programare.

### 3. Exemple de sisteme de interogare în bazele de date pentru prelucrări grafice

#### 3.1. Limbajul Spatial SQL

Limbajele din această categorie se bazează pe SQL, ele extind gramatica SQL cu o mulțime de operatori spațiali și relaționali care se referă la date spațiale. Unele limbaje cum ar fi PSQL (dezvoltat de Roussopoulos) nu permit adăugarea de noi



operatori la momentul formulării cererilor.

O altă abordare permite adăugarea unor funcții (operatori) definite de utilizatori care sunt făcute cunoscute sistemului de interogare la momentul execuției cererilor. Astfel de funcții pot avea valori scalare (de exemplu aria sau perimetrul unei regiuni) sau pot fi dependențe multivaloare (care au ca rezultat relații). Unii autori numesc aceste din urmă dependențe, funcții tabelare. Spre exemplu în limbajul de interogare *Spatial SQL*, Egenhofer propune ca datele spațiale să fie abstractizate prin extinderea domeniului calculului relațional cu alte patru domenii spațiale numite *spatial0*, ..., *spatial3* în funcție de dimensiunea obiectelor spațiale reprezentate. Un atribut al domeniului spațial se numește atribut spațial iar o relație ce se referă la cel puțin un atribut spațial se numește relație spațială. Astfel de relații au proprietăți diferite față de cele ale relațiilor obișnuite (cu câmpuri de tip numeric sau țir de caractere) ele se referă la concepte topologice sau metrice și sunt folosite similar cu relațiile convenționale.

*Operatorii spațiali* pot fi *unari*, de exemplu: dimensiune (determină dimensiunea topologică a unui obiect spațial), frontieră (determină "fețele" frontierei unui obiect n-dimensional), interior (determină părțile unui obiect care nu sunt situate pe frontiera sa). Tot din categoria operatorilor spațiali unari fac parte operatorii: lungime, arie și volum ce exprimă așa numitele proprietăți aritmetice ale unor obiecte spațiale, precum și operatori ce combină aspectele aritmetice cu cele topologice: perimetru, muchii frontieră, înfășurătoare convexă, coordonate extreme.

*Operatorii spațiali binari* calculează o valoare pornind de la două tuple ale unor relații spațiale, de exemplu operatorii

distanță: atribut\_spațial x atribut\_spațial →  
real

direcție: atribut\_spațial x atribut\_spațial →  
0 .. 359° 59' 59"

*Operatorii relaționali spațiali binari* descriu relații între două atribute spațiale și produc rezultate booleene; de exemplu operatorul "conține" (o linie poate conține sau nu un punct). Relațiile binare topologice se bazează pe determinarea intersecțiilor între frontierele sau interioarele a două regiuni. Operatori de acest tip se pot aplica oricăror două obiecte spațiale de dimensiuni arbitrare. Ca exemple de astfel de operatori se pot menționa: *e\_disjunct*, *intersectează*, *se\_suprapune*, *conține*, *acoperă*, *e\_egal*. Se pot specifica și relații de ordine spațială de tipul: stânga/dreapta, nord/sud sau sus/jos.

O altă caracteristică a bazelor de date relaționale care trebuie adaptată, extinsă cu noi facilități în cazul bazelor de date spațiale este limbajul de definire al datelor (DDL). Majoritatea sistemelor de baze de date spațiale bazate pe SQL permit specificarea de atribute spațiale în clauza *CREATE TABLE*. Astfel Egenhofer folosește numele de atribut *geometry* având tipul *spatial\_n* ( $n \in \{0,1,2,3\}$ ) și consideră că o relație spațială poate avea în general un singur atribut spațial care să definească geometria unui obiect. Există posibilitatea ca un obiect să aibă mai multe geometrii ceea ce ar putea fi util în cazul prezentării la diferite nivele de detaliu a respectivului obiect.

Samet propune ca tipuri de date disponibile pentru atributele spațiale: *LINE\_SEGMENT*, *REGION*, *POINT*, *POLYGON*, *BOX*. Atributele spațiale și cele nespațiale sunt plasate la același nivel conceptual, valoarea unui atribut spațial fiind comună tuturor atributelor nespațiale ale unui tuplu al unei relații.

Operațiile standard de manipulare a datelor: proiecția, selecția, joncțiunea caracteristice domeniului nespațial sunt adaptate și pentru domeniul spațial. Samet caracterizează un

predicat (specificat în clauza WHERE) ca fiind spațial dacă se referă la cel puțin un atribut spațial sau folosește un operator spațial; în caz contrar predicatul este relațional.

Dintre operațiile de interogare specifice bazelor de date spațiale se remarcă selecția grafică interactivă, specificată printr-o cerere de forma:

```
SELECT nume
FROM ora_
WHERE geometry=PICK;
```

Semantica cererii de selecție grafică interactivă depinde de dimensiunea topologică a obiectului indicat. În procesul de selecție pot apărea ambiguități dacă cel puțin două obiecte din clasa care trebuie identificată sunt situate la aceeași distanță de poziția cursorului selector, în astfel de cazuri utilizatorului trebuie să i se ofere mai multe variante de alegere pentru a identifica obiectul dorit.

### 3.2. Limbajul de cereri și modelul de date în sistemul VIMSYS

Modelul de date utilizat în sistemul VIMSYS [Gupta 91] (Visual Information Management System) (model stratificat) este orientat spre obiecte; el grupează entitățile de informație în 4 plane diferite, acestea sunt denumite DO (Domain Objects and Relations) DE (Domain Events and Relations) IO (Image Objects and Relations) IR (Image Representations and Relations). Toate obiectele au asociate atribute și metode, atributele sunt legate într-o ierarhie. Relațiile pot fi spațiale, structurale sau semantice.

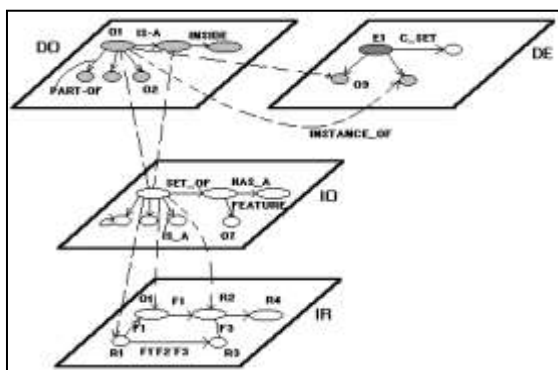


Fig. 2. Modelul de date VIMSYS

Datele constau în imagini, fiecare având asociate un antet precum și o înregistrare cu câmpuri alfanumerice. Elementele constitutive ale modelului de date sunt: reprezentările (i.e. tipurile de date pe baza cărora sunt construite obiectele) char, int, float, bool, string; constructorii set\_of, tuple\_of, vector\_of, matrix\_of, graph\_of, sequence\_of. O reprezentare de imagine este un tip abstract de date; pentru fiecare din obiectele imagine din planul IO, pot exista mai multe reprezentări asociate aceluiași obiect. În cursul procesului de interogare utilizatorul folosește o interfață grafică și un limbaj natural cu restricții pentru a specifica semantica unor caracteristici ale sistemului.

Sistemul dezvoltă o metodologie de organizare și indexare a imaginilor astfel încât să fie posibil accesul complet sau parțial la matricea unei imagini. Caracteristicile imaginilor sunt ordonate într-o ierarhie ceea ce permite construirea unor caracteristici mai complexe pornind de la unele mai simple. Modelul de date trebuie să permită utilizarea datelor spațiale și a fișierelor.

În planul IO există trei clase principale de obiecte: imagini, caracteristici de imagini, organizarea caracteristicilor; aceste obiecte sunt legate prin relațiile set\_of, is\_a și feature\_of. Localizarea regiunilor în imagini este realizată prin indexare folosind arborii  $R^+$ . Dintre caracteristicile pe care le poate avea o imagine sunt implementate textura (se referă la densitatea și orientarea muchiilor), culoarea, intensitatea și caracteristici geometrice (puncte, linii).

Planul DO conține o descriere la nivel semantic bazată pe informații conținute în planele IO și IR. Obiectele din planul DO sunt conectate printr-un graf (Object Relation Graph). Un obiect DO poate fi o

subclasă a unui obiect (sau mai multor obiecte) din planul IO. De exemplu, dacă obiectele  $O_1$ ,  $O_2$  situate în planul DO sunt subclase (sau obiecte prototip) ale obiectelor  $\omega_1$ ,  $\omega_2$  situate în planul IO, perechea  $(O_1, O_2)$  va reflecta toate relațiile între  $\omega_1$  și  $\omega_2$ . Un arc (sursă, destinație) între două obiecte situate în planul DO semnifică faptul că obiectul sursă are printre metodele sale o funcție care poate genera obiectul destinație. Pentru aceasta este suficient ca obiectul destinație să trimită un "mesaj cerere" cu parametri corespunzători.

Planul DE este inclus în modelul de date pentru a permite specificarea unor evenimente ce pot apărea în timpul derulării unor secvențe de imagini. Un eveniment (în acest context) este un ansamblu de caracteristici colectate asupra unei secvențe de imagini. Evenimentul "rotăție de imagini" va fi reprezentat printr-o mulțime de regiuni de imagini, un atribut (unghiul de rotație), o mulțime de restricții, și de metode care să găsească regiunile care satisfac respectivele restricții.

Sistemul de interogare permite accesul la o instanță a unei imagini prin specificarea unei cereri referitoare la orice mulțime de obiecte (care pot fi situate în mai multe planuri). În formularea unei cereri sunt posibile unele ambiguități astfel: utilizatorul poate folosi numele unui obiect semantic care nu este definit în sistem (va trebui să îl definească mai târziu pe baza obiectelor din sistem), sunt acceptate cereri de similitudine. Interfața utilizator este bazată pe meniuri, forme, butoane de selecție și ferestre multiple text sau imagine. O fereastră imagine permite executarea unor operații specializate ca: selectarea unei porțiuni de imagine, a unei regiuni, linie sau punct precum și a unor operații interne ca: panoramare, scalare, modificarea unor atribute de afișare.

În sistemul VIMSYS se propune un model de interogare în care utilizatorul nu specifică explicit clauza SELECT ci

incremental. Sistemul utilizează o interfață grafică pentru realizarea interogării, o bază de cunoștințe pentru a asista utilizatorul în procesul de detaliere a unei cereri și un dicționar de date pentru a stoca statistici referitoare la baza de date. Pentru a formula o cerere, utilizatorul specifică domeniul de interes urmat de aspectul de interes din cadrul domeniului (un aspect este o vedere a schemei OAR - Object Attribute Relation - având un nume unic). Modulul de cunoaștere primește la intrare o combinație (domeniu, aspect) și determină un subgraf al structurii OAR în planele DO și DE. Utilizatorul va alege obiectele (relațiile) care sunt necesare în cererea sa, modulul de cunoaștere va analiza obiectele în raport cu contextul (domeniu, aspect) dorit și va afișa mulțimea atributelor relevante ale fiecărei activități selectate. Utilizatorul va alege o submulțime a atributelor asociind pentru fiecare din acestea un predicat.

### 3.3. Limbajul de interogare Gral (algebre multisort)

Una din opțiunile de proiectare a unui Sistem de Gestiune a Bazelor de Date care să poată gestiona obiecte complexe este dacă structura acestor obiecte să fie sau nu vizibilă la nivelul modelului de date. Un obiect complex ar putea fi descris structural ca o colecție de tuple ale mai multor relații sau comportamental ca un simplu atribut care ia valori într-un domeniu specific respectivului obiect. În construirea SGBD-urilor extensibile au fost urmate două direcții: definirea cu precizie a unui model de date pentru sistem, ca și a interfeței cu utilizatorul asociată (POSTGRES) sau punerea la dispoziția utilizatorului a unei colecții de unelte de programare ce ar putea fi folosite în construirea unui SGBD, în acest din urmă caz utilizatorii bazei de date pot realiza direct adaptarea sau chiar implementarea unor porțiuni ale sistemului.

Un sistem extensibil din a doua categorie Gral [Güt 89] permite utilizatorului să



adauge tipuri de date sau operații noi la limbajul de interogare, să definească reprezentarea unui tip sau să adauge noi structuri de indexare care să fie adecvate noilor tipuri de date sau reprezentări de relații. Este posibilă și adăugarea unor operații specifice noilor tipuri de date sau specificarea unor reguli pentru a realiza optimizarea implementării operațiilor de interogare adăugate de utilizator.

Limbajul de interogare al sistemului Gral face parte din clasa limbajelor algebrice. Flexibilitatea deosebită a limbajului de cereri se datorează utilizării *algebrelor multisort* care permit atât specificarea cererilor cât și descrierea planelor de acces la informațiile din baza de date. Față de limbajele de cereri de tip SQL sau QUEL care permit introducerea cu ușurință a unor noi tipuri de date atomice și a unor funcții (predicate) specifice acestora, limbajele algebrice facilitează și extinderea limbajului cu noi operatori relaționali.

O algebră multisort este alcătuită dintr-o colecție de mulțimi (sorturi) și funcții (dependențe funcționale) între aceste mulțimi. În SGBD-ul Gral un sort este folosit pentru modelarea relațiilor, celelalte corespund tipurilor de date atomice. Operațiile caracteristice pot fi relaționale (de exemplu selecția sau filtrarea) sau operații asociate tipurilor de date atomice (de exemplu operații aritmetice sau operații geometrice de felul intersecției de poligoane). Sorturile și operatorii independenți de aplicație sunt grupate în așa numitul "cadru de modelare". Operatorii suportați sunt relaționali binari (reuniune, produs cartezian, filtrare, concatenare), relaționali unari (selecție, proiecție, ordonare, extend - care realizează extinderea unei relații cu un atribut) operatorul extract (REL → ATOM) și operatori booleani (and, or, not). În porțiunea bazei de date care este specifică unei aplicații se pot descrie atât operații referitoare la tipuri de date atomice cât și la relații.

Modelul de date de aplicație prezentat de Güting se numește algebra geo-relațională și este adecvat tratării aplicațiilor geometrice în plan fiind destinat unor aplicații GIS. Sunt definite tipuri pentru puncte, linii poligonale și regiuni plane (fără goluri). Operatorii asociați tipurilor spațiale (POINT, LINE, PGON, AREA, REG={PGON,AREA}, EXT={LINE}∪REG, GEO={POINT}∪EXT) sunt:  
inside (GEO x REG → BOOL),  
dist (POINT x POINT → REAL),  
length (LINE → REAL),  
perimeter și area (REG → REAL).

Dacă considerăm o bază de date referitoare la orașe, state, râuri și autostrăzi, ale cărei relații sunt:  
oras (onume:STR, centru:POINT, pop:INT)  
judet(jnume:STR, regiune:AREA, pop:INT)  
rau (rnume:str, curs:LINE)  
auto (anume:STR, drum:LINE)

Câteva exemple de cereri în acest limbaj algebric sunt:

```
oras, judet join [centru inside regiune]  

project [onume, jnume]
```

Asociază la fiecare oraș numele județului în care este situat respectivul oraș.  
judet select [jnume = "Prahova"] rau  
join [curs intersect regiune]  
project [rnume]  
Determină numele râurilor care trec prin județul Prahova.

Extinderea limbajului de interogare prin specificarea de noi tipuri de date sau operatori necesită descrierea unor aspecte sintactice cum ar fi poziția operatorilor față de operanzi sau structura listei de parametri a unui operator. În general, dacă cel puțin unul din operanzi este o relație se utilizează o notație postfixată.

Intr-un SGBD extensibil este necesar ca proiectantul de aplicații să înțeleagă

limbajul de specificare a planului de execuție al cererilor; acesta poate fi bazat tot pe modelul algebrei multisort. În sistemul Gral, algebra asociată limbajului de cereri se zice descriptivă iar cea în care se specifică modul de execuție al cererilor este algebra executabilă. Pentru a specifica prelucrări eficiente la nivelul algebrei executabile se pot descrie structurile de indexare, ordinea în care se fac operațiile asupra tuplelor relațiilor, astfel încât la fiecare acces la suportul de informație externă să fie prelucrată cât mai multă informație.

Dintre sorturile specifice algebrei executabile pot fi menționate:

REL o mulțime de sorturi în care fiecare din elemente corespunde unei reprezentări fizice (index primar) a unei relații. Valori posibile sunt SREL (reprezentare secvențială) STIDREL (o relație specificată în acest fel poate fi accesată secvențial sau printr-un index secvențial) BTREE (reprezentare sub forma unui arbore B\* ordonat după atributele relației).

INDEX o mulțime de sorturi ale cărei elemente sunt structuri de index secundar. Valorile posibile sunt SBTREE (arbore B\* utilizat ca index secundar) SGRID (fișier grilă) SMLGRID (fișier multigrilă folosit pentru indexarea într-o mulțime de dreptunghiuri care pot delimita obiecte de forma unor linii poligonale).

TUPLE este un sort ce descrie structura unui tuplu

ATOM sort pentru descrierea de obiecte atomice ele sunt caracteristice și algebrei descriptive

TIDSEQ e un sort ce descrie o mulțime de identificatori de tupluri la care se poate avea acces printr-una din reprezentările din REL.

Exemple de operatori ai algebrei executabile sunt:

scan:REL → SREL (parcurge toate tuplele unei relații indiferent de modul de reprezentare și verifică satisfacerea unei

condiții)

sort:REL → SREL (implementează operatorul ord din algebra descriptivă și permite filtrarea tuplelor obținute prin sortare în conformitate cu valoarea unei expresii booleene de filtrare)

De exemplu cererea "determină tuplul din relația orașe care corespunde orașului Ploiești":

```
oras select [onume = "Ploiești"]
```

are corespondentul în algebra executabilă:

```
oras "Ploiești" exactmatch [true; tuple]
```

Extensibilitatea SGBD-ului Gral poate fi deci realizată la trei nivele: extensibilitate prin adăugare de noi tipuri de date, extensibilitate prin adăugare de noi operatori executabili și extensibilitate prin adăugarea unor operatori descriptivi și a regulilor de optimizare asociate.

### Bibliografie:

[Catt 94] R. G. Cattell, Object Data Management, Addison Wesley Publishing Company, 1994.

[Egen 94] M. J. Egenhoffer, Spatial SQL: Q Query and Presentation Language, IEEE Transactions on Knowledge and Data Engineering, February 1994.

[Gupta 91] A. Gupta, T. Weymouth, R. Jain, Semantic Queries with Pictures: The VIMSYS Model, Proceedings of the 17-th International Conference on VLDB, 1991.

[Gütt 89] R. H. Güting, Gral: An Extensible Relational Database System for Geometric Applications, Proceedings of the Fifteenth International Conference on Very Large Databases, 1989.

[Khos 96] J. Khoshafian, A. B. Baker, Multimedia and Imaging Databases, Morgan Kauffman Publishers Inc., 1996.

[OMG 92] OMG ORBTF, Common Object Request Broker Architecture, Object Management Group, Framingham

Massachusetts, 1992.

[Samet 94] H. Samet, Spatial Data Models and Query Processing, apărut în Modern Database Systems, W. Kim (ed.), Addison Wesley, Reading, MA, 1994.