(19) OFICIUL DE STAT
PENTRU INVENŢII ŞI MĂRCI
Bucureşti

ROMÂNIA

(11) RO 128784 A2
(51) Int.Cl.
G06Q 99/00 (2006.01)

(12) # CERERE DE BREVET DE INVENŢIE

(21) Nr. cerere: **a 2012 00074**

(22) Data de depozit: **31.01.2012**

(41) Data publicării cererii:
30.08.2013 BOPI nr. **8/2013**

(71) Solicitant:
• **ALEXANDRU BOGDAN ŞERBAN**,
STR. ANTON BACALBAŞA NR. 17, BL. 143, AP. 31, SECTOR 4, BUCUREŞTI, B, RO;
• **FIERBINŢEANU ANDREI**,
STR. DRUMUL TIMONIERULUI NR. 6, BL. 111B, SC. A, ET. 6, AP. 27, SECTOR 6, BUCUREŞTI, B, RO

(72) Inventatori:
• **ALEXANDRU BOGDAN ŞERBAN**,
STR. ANTON BACALBAŞA NR. 17, BL. 143, AP. 31, SECTOR 4, BUCUREŞTI, B, RO;
• **FIERBINŢEANU ANDREI**,
STR. DRUMUL TIMONIERULUI NR. 6, BL. 111B, SC. A, ET. 6, AP. 27, SECTOR 6, BUCUREŞTI, B, RO

(74) Mandatar:
CABINET ENPORA S.R.L.,
STR. GEORGE CĂLINESCU NR. 52A, AP. 1, SECTOR 1, BUCUREŞTI

(54) # TRANSFORMAREA APELURILOR DE FUNCŢIE, PENTRU INTERACŢIUNEA CU STRUCTURILE DE DATE IERARHICE

(57) **Rezumat:**

Invenţia se referă la o metodă şi la un sistem, utilizate la transformarea apelurilor de funcţie, pentru interacţiunea cu structurile de date ierarhice. Metoda conform invenţiei constă în primirea, de către procesor, a unui apel de funcţie, format pentru a opera pe o bază de date relaţională, în transformarea de către procesor a apelului de funcţie într-un apel de funcţie transformat, format, pentru interacţiunea cu o structură de date ierarhică, care structură de date are în componenţă mai multe noduri, fiecare nod având un singur nod părinte, şi în aplicarea, de către procesor, a apelului de funcţie transformat, la unul sau mai multe noduri ale structurii de date ierarhice. Sistemul conform invenţiei are în componenţă un procesor configurat pentru a executa instrucţiuni stocate într-un mediu netranzitoriu, citibil pe calculator, care asigură o aplicaţie de mapare, ce are în componenţă unul sau mai multe module configurate, pentru a performa operaţiile de primire, transformare şi aplicare a unui apel de funcţie.

Revendicări: 20
Figuri: 5

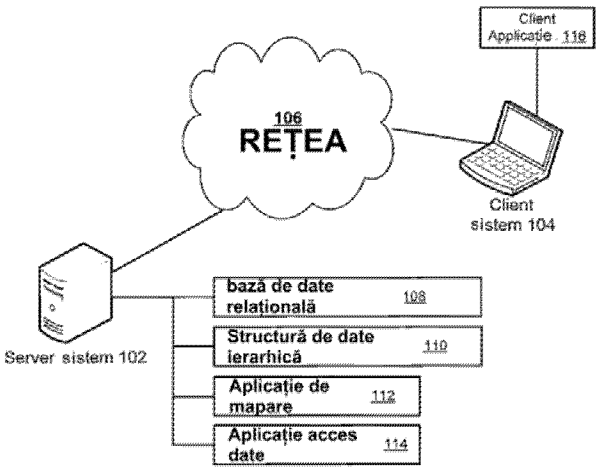

Fig. 1

RO 128784 A2

Domeniul tehnic

[0001] Această prezentare se referă în general la un software de calculator şi mai precis se referă la transformarea apelurilor de funcţie pentru interacţiunea cu structuri de date ierarhice.

Stadiul tenhicii mondiale din domeniu

[0002] Limbajele de programare operează în mod frecvent pe modele de software care au unul sau mai multe atribute cunoscute ca obiecte. Un limbaj de programare poate să interacţioneze cu un obiect folosind diferite persistence framework-uri. Un persistence framework poate să includă structuri de date întrebuinţate pentru a stoca valori pentru diferitele atribute ale unui obiect şi unul sau mai multe procese pentru interacţionarea cu obiectul stocat. Persistence framework-urile pot să includă, de exemplu, baze de date relaţionale şi sisteme de stocare ierarhice. O bază de date relaţională poate să stocheze un obiect cum ar fi un rând al unui tabel care are un număr predefinit de coloane, cu fiecare atribut al obiectului mapat la o coloană a tabelului. Un sistem de stocare ierarhic poate să stocheze un obiect ca un nod într-o structură de date arborescentă asemănătoare unui sistem fişier, fiecare atribut al obiectului fiind mapat la o proprietate a nodului.

[0003] Un sistem de stocare ierarhic poate să asigure avantaje în comparaţie cu o bază de date relaţională, cum ar fi o flexibilitatea crescută şi claritate pentru manevrarea datelor în aplicaţiile de administrare de conţinut. De exemplu, un sistem de stocare ierarhic poate să fie întrebuinţat pentru a stoca conţinut nestructurat care nu se potriveşte neapărat într-o schemă predefinită, contrar unei baze de date relaţionale care are cerinţa ca un obiect să se conformeze unei scheme de date (adică, coloanele unui tabel). În felul acesta, un sistem de stocare ierarhic poate fi mai bine adaptat decât o bază de date relaţională pentru stocarea datelor ne structurate sau semi-structurate cum ar fi un conţinut de web.

[0004] Cu toate acestea, dezvoltatorii de aplicaţii pot să nu fie familiarizaţi cu standardele pentru sistemul de stocare ierarhic şi din această cauză să dezvolte mai curând aplicaţii pentru sisteme de baze de date relaţionale decât pentru sistemul de stocare ierarhic. De exemplu, sistemele de baze de date relaţionale

pot să pună la dispoziție metode simple pentru manevrarea logicului de business al unei aplicații. În consecință, simplitatea dezvoltării de aplicații configurate pentru a interacționa cu sistemele de baze de date relaționale pot să-i convingă pe dezvoltatori să renunțe la testarea avantajelor sistemelor de stocare ierarhice.

Descrierea pe scurt

[0005] Sunt prezentate sisteme şi metode pentru transformarea apelurilor de funcție formate pentru interacțiunea cu o bază de date relațională într-un apel de funcție pentru o structură de date ierarhică. O modalitate de realizare cu caracter de exemplu implică un procesor care primeşte un apel de funcție format pentru a opera pe o bază de date relațională. Apelul de funcție se referă la un obiect din baza de date relațională prin referirea cel puțin la un tabel din baza de date relațională şi la cel puțin o coloană a acelui cel puțin un tabel. Procesorul transformă apelul de funcție într-un apel de funcție transformat format pentru interacțiunea cu o structură de date ierarhică. Structura de date ierarhică are în componență mai multe noduri, fiecare nod având un singur nod părinte. Procesorul aplică apelul de funcție transformat la unul sau la mai multe noduri ale structurii de date ierarhice.

[0006] Aceste trăsături ilustrative sunt menționate nu pentru a limita sau defini prezentarea, ci pentru a pune la dispoziție exemple care să ajute la înțelegerea acesteia. Modalități de realizare suplimentare sunt discutate în cadrul Descrierii detaliate, unde este dată o descriere suplimentară. Avantajele oferite de una sau de alta din diversele modalități de realizare pot să fie înțelese de asemenea prin examinarea acestei specificații sau prin punerea în practică a unuia sau a mai multora dintre modalitățile de realizare prezentate.

Scurtă descriere a figurilor

[0007] Aceste şi alte caracteristici, aspecte şi avantaje ale prezentei prezentări sunt mai bine înțelese atunci când este citită Descrierea detaliată cu referire la desenele însoțitoare, în care:

[0008] Figura 1 este o diagramă de rețea care ilustrează un sistem de client care accesează date stocate la un server sistem într-o structură de date ierarhică şi o bază de date relațională;

[0009] Figura 2 este o diagramă de modelare care ilustrează o bază de date relațională cu caracter de exemplificare şi o structură de date ierarhică cu caracter de exemplificare;

[0010] Figura 3 este o diagramă de modelare care ilustrează un flux al comunicațiilor dintre o aplicație de client şi o structură de date ierarhică, cu caracter de exemplificare;

[0011] Figura 4 este o schemă bloc care ilustrează dispozitive de calcul cu caracter de exemple într-un mediul de calcul cu caracter de exemplu pentru implementarea anumitor modalități de realizare; şi

[0012] Figura 5 este o schemă logică care ilustrează o metodă cu caracter de exemplu pentru transformarea apelurilor de funcție formate pentru interacțiunea cu o bază de date relațională.

Descrierea detaliată

[0013] Sunt puse la dispoziție sisteme şi metode pentru transformarea apelurilor de funcție pentru interacțiunea cu structuri de date ierarhice, sistemele şi metodele pot să includă o aplicație de mapare care să asigure o punte între aplicațiile desemnate pentru interacțiunea cu baze de date relaționale şi sisteme de stocare care întrebuințează structuri de date ierarhice. Aplicația de mapare poate să transforme un apel de funcție format pentru a interacționa cu o bază de date relațională într-un apel de funcție format pentru a interacționa cu o structură de date ierarhică. Aplicația de mapare poate astfel să permită unui dezvoltator care nu este familiarizat cu sistemele de stocare de date ierarhice să folosească aceeaşi interfață de programare a aplicației („API") pentru accesarea datelor stocate utilizând fie o bază de date relațională fie o structură de date ierarhică.

[0014] Exemplul care urmează ilustrează modul în care o aplicație de mapare poate să transforme un apel de funcție format pentru interacțiunea cu baze de date relaționale într-un apel de funcție format pentru interacțiunea cu structuri de date ierarhice. O bază de date relațională poate să includă un tabel „maşini"

pentru stocarea obiectelor maşină, un tabel „bărci" pentru stocarea obiectelor barcă şi un tabel „vehicule" pentru stocarea obiectelor vehicul. În mod alternativ, obiecte care corespund cu maşini, bărci sau alte vehicule pot să fie stocare întrebuinţând o structură de date ierarhică. Structura de date ierarhică poate să includă o ierarhie a nodurilor, fiecare nod fiind corespondent unui obiect. De exemplu, nodurile care corespund cu obiecte maşină pot să fie stocate ca noduri fiu ale unui nod părinte „Maşini" şi nodurile care corespund obiectelor barcă pot să fie stocate ca noduri fiu ale unui nod părinte „Barcă". Atât „Maşini" cât şi „Bărci" pot la rândul lor să fie stocate ca noduri fiu ale unui nod părinte „Vehicule".

[0015] O interogare pentru obiecte maşină poate să fie formată în mod diferit pentru fiecare bază de date relaţională în comparaţie cu o structură de date ierarhică. De exemplu, deşi tabelul „Vehicule" poate fi pus în relaţie cu fiecare dintre tabelele „Maşini" şi „Bărci", o interogare către o bază de date relaţională poate să facă trimitere la tabelul „Maşini" fără a face trimitere la tabelul „Vehicule".

Spre deosebire de aceasta, o interogare către o structură de date ierarhică poate să facă trimitere la tabelul „Maşini" prin identificarea căii de la nodul părinte „Vehicule" (adică, „/Vehicule"Maşini". În consecinţă, o interogare formată pentru interacţiunea cu baza de date relaţională prin referenţiere la „Maşini" fără identificarea căii „/Vehicule/Maşini" nu va reuşi să se întoarcă cu niciunul dintre obiectele maşină.

[0016] Pentru a împiedica un astfel de rezultat eronat, o aplicaţie de mapare poate, de exemplu, să transforme interogarea astfel încât orice referire la „Maşini" este transformată într-o referire la „/Vehicule/Maşini". În consecinţă, o aplicaţie dezvoltată pentru interacţiunea cu o bază de date relaţională poate să utilizeze aplicaţia de mapare ca o punte către o structură de date ierarhică.

[0017] Într-o modalitate de realizare cu caracter de exemplu, o aplicaţie de mapare executată de către un procesor al unui server sistem poate să primească un apel de funcţie format pentru a opera pe o bază de date relaţională. Apelul funcţie poate să facă trimitere la un obiect din baza de date relaţională prin referenţierea cel puţin a unui tabel din baza de date relaţională şi a cel puţin unei

coloane a tabelului. Aplicația de mapare poate să transforme apelul de funcție într-un apel de funcție transformat format pentru interacționarea cu o structură de date ierarhică. Structura de date ierarhică poate să includă mai multe noduri care au relații părinte-fiu, în care fiecare nod fiu are un singur nod părinte. Apelul de funcție transformat poate să referențieze obiectul din structura de date ierarhică prin referențierea unei relații între un nod care stochează obiectul şi cel puțin un nod adițional care este un părinte al nodului care stochează obiectul. Aplicația de mapare poate aplica apelul de funcție transformat unuia sau mai multor noduri ale structurii de date ierarhice.

[0018] După cum este utilizat aici, termenul „obiect" este utilizat pentru referirea la orice entitate logică care poate să fie manevrată prin comenzi ale unui limbaj de programare. Exemple ale unui obiect pot să includă (dar fără a se limita la acestea) o variabilă, o funcție, o structură de date, o combinație de variabile, funcții şi structuri de date etc. Obiectul poate să includă un identificator care să distingă obiectul de alte obiecte. Obiectul poate să includă date stocate în obiect. Obiectul poate să includă metode prin care obiectul poate să fie folosit de către o aplicație.

[0019] După cum este utilizat aici, termenul „apel de funcție" este întrebuințat pentru a se referi la o referință de către o aplicație a unui sistem de calcul către o subrutină, procedură sau altă funcție care specifică o operație care trebuie îndeplinită pe un obiect al unui model stocat. Exemple de apel de funcție pot să includă (fără a se limita la acestea) o comandă pentru recuperarea unuia sau mai multor obiecte dintr-un model de stocare, o comandă pentru adăugarea unuia sau mai multor obiecte la un model de stocare, sau o comandă pentru a şterge unul sau mai multe obiecte dintr-un model de stocare. Un apel de funcție poate să execute o subrutină, o procedură, sau o altă funcție ca răspuns la intrarea primită prin intermediul unui dispozitiv de intrare aflat în comunicație cu un sistem de calcul sau ca răspuns la o comandă generată de la operarea unei aplicații executate la un sistem de calcul. De exemplu, un prim modul de software poate să configureze un procesor al unui sistem de calcul pentru a executa o funcție de la un al doilea modul de software prin asigurarea unui apel de funcție pentru acest al doilea modul software.

[0020] După cum este utilizat aici, termenul „model de stocare" este întrebuințat pentru a face referire la un model pentru organizarea datelor stocate ca obiecte. Exemple de model de stocare pot să includă (dar nu sunt limitate la acestea) o bază de date relațională sau o structură de date ierarhică. Un model de stocare poate să includă schema de date pentru organizarea obiectelor şi a atributelor obiectelor.

[0021] După cum este utilizat aici, termenul „bază de date relațională" este întrebuințat pentru a face referire la un model de stocare în care schema de date include stocarea de obiecte în unul sau mai multe tabele asociate. Fiecare tabel poate include unul sau mai multe rânduri. Un rând poate să fie o înregistrare care stochează datele unui obiect. Fiecare rând dintr-o bază de date relațională poate să corespundă unui obiect. Fiecare rând poate să aibă una sau mai multe coloane care corespund unui atribut al înregistrării de date. Un obiect poate să fie stocat într-o bază de date relațională prin creerea unui rând care corespunde cu obiectul şi prin stocarea fiecărui atribut al unui obiect într-o coloană a rândului. Dat fiind faptul că fiecare rând dintr-un tabel are aceleaşi coloane ca oricare alt rând din tabel, un tabel poate să stocheze mai multe obiecte care au atribute identice. Fiecare obiect dintr-o bază de date relațională poate să fie accesat prin identificarea unui tabel în care obiectul este stocat şi prin identificarea a cel puțin unei coloane care are o valoare de identificare a obiectului. Două sau mai multe tabele dintr-o bază de date relațională pot să fie legate printr-o cheie. O cheie poate să fie o coloană dintr-un prim tabel care include valori care corespund cu valorile dintr-o coloană din al doilea tabel.

[0022] După cum este folosit aici, termenul „structură de date ierarhică" este întrebuințat pentru a face referire la un model de stocare în care schema de date include stocarea obiectelor ca noduri organizate în conformitate cu o ierarhie. Un nod poate să fie un container pentru datele unui obiect dintr-o structură de date ierarhică. Fiecare nod poate stoca atribute ale unui obiect ca proprietăți ale nodului. Proprietățile au valori, care reprezintă datele actuale stocate în structura de date ierarhică. Nodurile pot să fie organizate în conformitate cu o ierarhie care se bazează pe una sau mai multe relații de tipul părinte/fiu între noduri. Fiecare nod părinte poate să aibă unul sau mai multe noduri fiu. Fiecare nod fiu poate să

aibă un singur nod părinte. Fiecare nod dintr-o structură de date ierarhică poate să aibă atribute care sunt diferite de cele ale altor noduri din structura arborescentă. Fiecare nod poate să includă o relație cu un alt nod şi proprietățile acelui nod. Nodurile unei structuri de date ierarhică pot să fie accesate prin intermediul căilor. O cale poate să fie o descriere a relației dintre noduri, cum ar fi „Rădăcină/Părinte/Fiu". Astfel de noduri includ căi absolute care încep de la un nod rădăcină sau căi relative către alte noduri.

[0023] În cadrul unei modalități de realizare cu caracter de exemplu, transformarea apelului de funcție poate să includă maparea unei referințe incluse în apelul de funcție de la un tabel la o cale. Calea poate să identifice o relație între un nod care corespunde cu tabelul şi un nod părinte al unui nod care corespunde cu tabelul. De exemplu, o interogare care face referire la tabelul „Maşini" poate să fie transformată pentru a referenția calea de la un nod părinte „Vehicule" la un nod fiu „Maşini" care corespunde cu tabelul „Maşini" (adică, „/Vehicule/Maşini").

[0024] În cadrul unor modalități de realizare suplimentare sau alternative, transformarea funcției poate să includă maparea unei referințe incluse în apelul de funcție dintr-un rând al tabelului bazei de date relaționale într-un nod fiu al nodului care corespunde cu tabelul. De exemplu, maparea aplicației poate să transforme un apel de funcție adăugând un rând „Maşină 1" la tabelul „Maşini" dintr-o bază de date relațională într-un apel de funcție prin adăugarea unui nod „Maşină1" ca un fiu al nodului „Maşini" dintr-o structură de date ierarhică. Transformarea funcției poate să includă de asemenea maparea unei referințe dintr-o coloană a tabelului bazei de date relaționale într-o proprietate a unui nod fiu. De exemplu, maparea aplicației poate să transforme un apel de funcție modificând valoarea unei coloane etichetate „Fabricant" de la un rând "Maşină1" dintr-un tabel „Maşini" într-un apel de funcție prin modificarea unei proprietăți etichetate „Fabricant" a nodului „Maşini1".

[0025] În cadrul unor modalități de realizare suplimentare sau alternative, transformarea funcției poate să includă maparea unei referințe dintr-un apel de funcție într-o relație dintre un prim tabel al bazei de date relaționale şi un al doilea tabel al bazei de date relaționale într-o relație între un prim nod care corespunde

unui prim tabel şi un al doilea nod care corespunde unui al doilea tabel. De exemplu, un tabel „Vehicule" poate să fie relaționat cu fiecare dintre tabelele „Maşini" şi „Bărci". Un apel de funcție poate să includă o referință către relația dintre „Vehicule" şi „Maşini" şi/sau relația dintre „Vehicule" şi „Maşini". Maparea aplicației poate să determine că tabelul „Vehicule" are o relație de tipul de la unul la multe cu tabelele „Maşini" şi „Bărci". Maparea aplicației poate astfel să determine că un nod care corespunde cu tabelul „Vehicule" este un nod părinte al nodurilor care corespund cu tabelele „Maşini" şi „Bărci".

[0026] În cadrul unor modalități de realizare suplimentare sau alternative, maparea aplicației poate să genereze un rezultat bazat pe aplicarea apelului de funcție transformat la acel unul sau la acele mai multe noduri dintr-o structură de date ierarhică. Maparea aplicației poate să transforme rezultatul astfel încât rezultatul este format pentru baza de date relațională. De exemplu,o interogare către structura de date ierarhică poate să se întoarcă cu unul sau cu mai multe obiecte care corespund cu interogarea. Rezultatul poate să referențieze obiectele dintr-o structură de date ierarhică întrebuințând calea de la un nod părinte la noduri care corespund cu obiectele. Aplicația de mapare poate să transforme referințele pentru cale în referințe pentru tabelul în care va fi stocat obiectul într-o bază de date relațională corespondentă. Rezultatul transformat poate să fie pus la dispoziția aplicației în care îşi are originea apelul funcție.

[0027] Aceste exemple ilustrative sunt date pentru a introduce cititorul în subiectul general discutat aici şi ele nu au intenția de a limita domeniul conceptelor prezentate. Secțiunile care urmează descriu diverse modalități de realizare suplimentare şi exemple cu referire la desene în care referințele numerice asemănătoare indică elemente asemănătoare.

[0028] Caracteristicile discutate aici nu sunt limitate la nicio arhitectură de hardware sau configurație specială. Un dispozitiv de calcul poate să includă orice aranjament potrivit al componentelor care asigură un rezultat condiționat de unul sau de mai multe apeluri funcție. Dispozitivele de calcul corespunzătoare includ sisteme de calcul pe bază de microporcesoare multidestinație care accesează software stocat care programează sau configurează sistemul de calcul de la un aparat de calcul multi-destinație într-un aparat de calcul specializat prin

implementarea uneia sau a mai multora dintre modalitățile de realizare ale prezentului subiect. Oricare dintre limbajele de programare, de scripting sau oricare alt tip de limbaj sau combinații de limbaje poate să fie folosite pentru a implementa cunoştințele conținute aici în software-ul de utilizat în programarea sau configurarea dispozitivului de calcul.

[0029] Dacă ne referim acum la desene, Figura 1 este o diagramă de rețea care ilustrează un sistem client 104 care accesează date stocate într-un server sistem 102 într-o bază de date relațională 108 şi o structură de date ierarhică 110.

[0030] Serverul sistem 102 poate să includă, dar nu este limitat la acesta, un singur server sistem, un sistem de calcul de tip nor, un sistem de calcul de tip grid etc. Sistemul de client 104 poate să fie un sistem de calcul cum ar fi, fără a se limita la acestea, un calculator tip laptop, un calculator de birou, un calculator tabletă sau un alt sistem de calcul. Sistemul de client 104 poate să comunice cu serverul sistem 102 prin intermediul rețelei 106. Exemple de rețea 106 pot să includă, fără a se limita la acestea, o rețea locală, o rețea de arie largă, o rețea fără fir etc.

[0031] Sistemul de client 104 poate să execute o aplicație de client 116. Aplicația de client 116 poate să includă orice aplicație care are un modul, cum ar fi (fără a se limita la acesta) un API, pentru accesarea sau utilizarea în alt fel a datelor stocate la distanță în serverul sistem 102. Datele pot să fie stocate în serverul sistem 102 într-o bază de date relațională 108, cum ar fi (fără a se lirnita la aceasta) una sau mai multe baze de date relaționale, sau într-o structură de date ierarhică 110.

[0032] Aplicația de client 116 poate să acceseze sau să folosească în alt fel datele stocate pe serverul sistem 102 prin intermediul sistemului de client 104 furnizând apeluri de funcție de la aplicație de client 116 la serverul sistem 102 prin intermediul rețelei 106. Apelul de funcție primit de către serverul sistem 102 poate să producă executarea, de către o aplicație de acces la date 114, a unei operații specificate de apelul de funcție. Operația specificată de către apelul de funcție poate să includă aceesarea sau folosirea în alt mod a datelor stocate fie în baza de date relațională 108 fie în structura de date ierarhică 110.

[0033] Un apel de funcție de la o aplicație de client 116 poate să fie format pentru interacțiunea cu baza de date relațională 108 dar să fie adresat către structura de date ierarhică 110. Serverul sistem 102 poate să execute o aplicație de mapare 112 pentru a transforma apelul de funcție format pentru interacțiunea cu baza de date relațională 108 în apelul de funcție format pentru interacțiunea cu structura de date ierarhică 110.

[0034] Deşi aplicația de mapare 112 este descrisă în Figura 1 ca o aplicație separată executată la serverul sistem 1023, sunt posibile şi alte configurații. De exemplu, în modalitățile de realizare, aplicația de mapare 112 poate să fie executată la sistemul de client 104 ca o aplicație cuplată la client 116 sau ca o aplicație de sine stătătoare care operează pe sistemul de client 104.

[0035] Figura 2 este o diagramă de modelare care ilustrează o bază de date relațională 108 şi o structură de date ierarhică 110.

[0036] Baza de date relațională 108 poate să includă, de exemplu, o bază de date relațională care are unul sau mai multe tabeluri asociate, cum sunt tabelele 202a-b. În cadrul exemplului descris în Figura 2, tabelul 202a poate să fie etichetat „Maşini" şi poate să stocheze înregistrări de date care corespund cu obiecte maşini. Atributele asociate cu un obiect maşină pot să includă un identificator pentru maşină, un fabricant pentru maşină şi un model al maşinii. Atributele unui obiect pot să fie mapate la coloanele bazei de date a tabelului 202a întrebuințând metadate care identifică relația dintre un atribut al unui obiect şi baza de date în care este stocat obiectul. De exemplu, un obiect poate să includă metadate care specifică că obiectul trebuie să fie stocat în tabelul numit „Maşini", că un unic identificator al obiectului trebuie să fie stocat în „Maşină_Id" şi alte atribute ale obiectului trebuie să fie stocate respectiv în „Fabricant_Id" şi „Model_Id".

[0037] Fiecare din rândurile 204a-c ale tabelului poate să corerspundă unui obiect maşină. Pentru fiecare obiect maşină care coresapunde cu un rând al tabelului, poate să fie asociat un atribut al obiectului maşină cu o coloană a tabelului 202a. Coloana 206a poate să fie etichetată „Maşină_Id" şi poate să includă date care identifică o anumită maşină. „Maşină_Id" poate să fie un identificator unic pentru fiecare rând al tabelului. Coloana 206b poate să fie

etichetată „Fabricant_Id" şi poate să includă date care identifică fabricantul maşinii. Coloana 206c poate să fie etichetată „Model_Id" şi poate să includă date care identifică modelul maşinii. Rândul 204a poate să corespundă unui obiect maşină care are o valoare „Maşină1" pentru Maşinşă_Id, o valoare „FabricantA" pentru Fabricant_Id şi o valoare „Model1" pentru Model_Id. Rândul 204b poate să corespundă unui obiect maşină care are o valoare „Maşină2" pentru Maşină_Id, o valoare „Fabricant B" pentru Fabricant_Id şi o valoare „Model2" pentru Model_id. Rândul 204 poate să corespundă unui obiect maşină care are o valoare „Maşină3" pentru Maşină_Id, o valoare „Fabricant B" pentru Fabricant_Id şi ovaloare „Model3" pentru Model_Id.

[0038] Tabelul 202b poate să stocheze înregistrări de date care corespund cu obiecte vehicule. Coloana 207a a tabelului 202b poate să includă date care identifică fiecare rând din tabelul 202b. Coloana 207b a tabelului 202b poate să includă date care identifică un tip de vehicul într-un rând al tabelului 202b, cum ar fi „Maşini", „Avioane" şi „Bărci". Tabelul 202a poate să fie asociat cu tabelul 202b pe baza unei valori a coloanei 207b, după cum este descris cu linia întreruptă care conectează înregistrarea pentru „Maşini" din tabelul 202b cu tabelul 202a.

[0039] Accesarea unui obiect stocat ca unul dintre rândurile 204a-c din baza de date relațională 108 poate să includă identificarea tabelului 202a în care este stocat obiectul şi identificarea că una sau mai multe dintre coloanele 206a-c ale tabelului 202a au o valoare care corespunde obiectului. De exemplu, pentru a accesa obiecte care au un Frabricant_Id FabricantB, o aplicație poate să pună la dispoziție o interogare către baza de date relațională 108 care identifică tabelul 202a şi selectează rândurile din tabelul 202a în care Fabricat_Id = FabricantB.

[0040] O structură de date ierarhică 110 poate să stocheze obiecte maşină folosind relațiile de tip ierarhic. De exemplu, un nod 208 poate să fie etichetat „Maşini". Nodul 208 poate să fie asociat cu nodurile fiu 210a-c. fiecare dintre nodurile fiu 210a-c poate să fie asociat cu o maşină care are proprietățile Maşină_Id, Fabricant_Id şi Model_Id. Nodul fiu 210a poate să corespundă cu o maşină care are o valoare „Maşină1" pentr Maşină_Id, o valoare „FabricantA" pentru Fabricant_Id şi o valoarea „Model1" pentru Model_Id. Nodul fiu 210 poate să corespundă cu o maşină care are o valoare „Maşină2" pentr Maşină_Id, o

valoare „FabricantB" pentru Fabricant_Id şi o valoarea „Model2" pentru Model_Id. Nodul fiu 210c poate să corespundă cu o maşină care are o valoare „Maşină3" pentr Maşină_Id, o valoare „FabricantB" pentru Fabricant_Id şi o valoarea „Model3" pentru Model_Id.

[0041] Nodul 208 poate să fie un fiu al nodului 212, etichetat „Vehicule". Accesarea unui obiect dintr-o structură de date ierarhică 110 poate să includă identificarea unei căi către nodul care corespunde cu obiectul. De exemplu, pentru a accesa obiecte care au un Fabricant_Id FabricantB, o aplicație va supune o interogare către baza de date relațională 108 pentru identificarea căii de la nodul rădăcină 212 la nodul 208 şi selectarea nodurilor fiu ale nodului 208 în care Fabricant _Id = Fabricant B.

[0042] Diferite noduri dintr-o structură de date ierarhică 110 pot avea proprietăți diferite. Unele noduri pot să includă numai un identificator pentru nod şi o relație către alte noduri. De exemplu, fiecare dintre nodurile 208, 212 poate să includă numai identificatorii lor („Maşini" şi „Vehicule") şi relațiile dintre ei. Alte noduri pot să includă identificatori, relații şi una sau mai multe proprietăți suplimentare. De exemplu, fiecare din nodurile 210a-c include identificatorul lor respectiv („Maşină1", „Maşină2" şi „Maşină3"), relația lor cu nodul 208 şi valorile lor respective pentru Fabricant_Id şi Model_Id. Ca un alt exemplu, deşi nodurile 210a-c sunt noduri fiu ale nodului 208, proprietățile fiecărui nod nu sunt identice. Nodul 210c are o proprietată suplimenată de „An" cu o valoare „1984".

[0043] După cum este prezentat în Figura 2, înregistrările de date în baza de date relațională 108 pot să includă proprietăți care corespund cu înregsitrările de date din structura de date ierarhică 110. Tabelul 202, care are mai multe rânduri care descriu obiecte maşină, corespund cu nodul 208, având mai multe noduri fiu care descriu obiecte maşină. Fiecare dintre rândurile 204a-c corespunde respectiv cu unul dintre nodurile fiu 210a-c. Fiecare din coloanele 206a-c corespunde respectiv cu o proprietate a nodurilor fiu 210a-c.

[0044] O aplicație de client 116 poate să fie configurată pentru a accesa sau a întrebuința în alt mod obiecte prin intermediul unui apel de funcție format pentru o bază de date relațională 108. O aplicație de mapare 112 executată la serverul sistem 102 poate să transforme apelul de funcție astfel încât apelul de funcție

este format pentru o bază de date relațională 108. Figura 3 este o diagramă de modelare care ilustrează un flux de comunicații cu caracter de exemplu dintre o aplicație de client 116 şi o structură de date ierarhică 110.

[0045] Aplicație de client 112 a unui sistem de client 104 poate să asigure un apel de funcție 302a pentru o aplicație de client 116 către serverul sistem 102. Apelul de funcție 302a poate să fie format pentru o bază de date relațională 108 şi adresat structurii de date ierarjhice 110.

[0046] Un apel de funcție 302a poate să includă o interogare pentru recuperarea de date de la o sursă de date care stochează date folosind o bază de date relațională 108. Interogarea poate să identifice o structură de date ierarhică 110 ca sursă a datelor şi să referințieze una sau să înregistreze sintaxa pentru o bază de date relațională 108. De exemplu, apelul de funcție 302a poate să fie o comandă de la o aplicație de client 112 care încorporează un pachet Java Persistence API („JPA") pentru administrarea bazelor de date relaționale şi direcționată către un sistem de stocare folosind Content Repository API pentru Java („JCR"). În felul acesta, o aplicare de acces de date 114 poate să nu fie în stare să regăsească corespunzător datele folosind comanda inclusă în apelul de funcție 302a şi astfel să nu reuşească să execute corespunzător comanda.

[0047] De exemplu, o interogare pentru regăsirea de obiecte maşină din structura de date ierarhică 110 poate să fie formată pentru interacțiunea cu o bază de date relațională 108. Interogarea poate astfel să includă o referință care identifică tabelul „Maşini". Interogarea nu ar identifica calea către nodul „Maşini" (adică, „Vehicule/Teren/Maşini") întrebuințată de către structura de date ierarhică 110 pentru a stoca obiecte maşină. În felul acesta, interogarea către structura de date ierarhică 110 formată pentru interacțiunea cu baza de date relațională nu ar reuşi să aducă obiectele maşină tocate în structura de date ierarhică 110.

[0048] O aplicație de mapare 112 poate să împiedice o astfel de nereuşită. Aplicația de mapare 112 poate să primească apelul de funcție 302a de la aplicația de client 116. aplicația de mapare 112 poate să transforme apelul de funcție 302a într-un apel de funcție transformat 302b întrebuințând relația dintre înregistrări ale bazei de date relaționale 108 şi structura de date ierarhică 110.

[0049] De exemplu, aplicație de mapare 112 poate să transforme o referință către un tabel în apelul de funcție 302a într-o referință care identifică o cale către un nod care corespunde cu tabelul din apelul de funcție transformat 302b. De exemplu, un apel de funcție care identifică tabelul „Maşini" poate să fie transformat într-un apel de funcție care identifică calea „/Vehicule/Maşini". Aplicația de mapare 112 poate să transforme o funcție referențiază o coloană din apelul de funcție 302a într-o funcție corespondentă care referențiază la o proprietate a unui nod din apelul de funcție transformat 302b. Aplicația de mapare 112 poate să transforme o funcție care referențiază un rând din apelul de funcție 302a într-o funcție corespondentă care referențiază un nod fiu dintr-un apel de funcție transformat 302b.

[0050] Aplicația de mapare 112 poate să transforme relația dintre tabelele 112 identificate într-un apel de funcție în relații între noduri. De exemplu, aplicația de mapare 112 poate să determine că un apel de funcție 302a include o referință către relația dintre tabelul „Maşini" şi tabelul „Vehicule". Aplicația de mapare 112 poate să determine că relația de la „Vehicule" la „Maşini" este o relație de tipul „de la mulți la unul". Aplicația de mapare 112 poate astfel să determine că relația ierarhică dintre „Vehicule" şi „Maşini" este una părinte-fiu, după cum este ilustrat de către nodurile 212 în Figura 2.

[0051] Aplicația de mapare 112 poate să asigure apelul de funcție transformat 302b pentru aplicația de acces la date 114. Aplicația de acces la date 114 poate să acceseze sau să folosească în alt fel structura de date ierarhice 110 într-o manieră specificată de apelul de funcție transformat 302b.

[0052] Aplicația de mapare 112 poate să transforme o ieşire de la aplicația 114 pentru accesul datelor astfel încât ieşirea poate să fie folosită de către o funcție pentru baza de date relațională. Aplicația pentru acccesul datelor 114 poate să obțină un rezultat 304a prin accesarea sau utilizarea în alt mod a structurii de date ierarhice 110 într-o manieră specificată de apelul de funcție transformat 112. Aplicația de mapare 112 poate să furnizeze rezultatul 304a către aplicația de mapare 112. Aplicația de mapare 112 poate să transforme rezultatul 304a în rezultatul transformat 304b pentru utilizarea de către aplicația de client 116.

[0053] Aplicația de client 112 poate să transforme o referință către un nod din rezultatul 304a într-o referință corespondentă către un tabel din rezultatul transformat 304b. Aplicația de mapare 112 poate să transforme o referință către un nod fiu 304a într-o referință corespondentă către un rând din rezultatul transformat 304b. Aplicația de mapare 112 poate să transforme o referință către o proprietate a unui nod din rezultatul 304a într-o referință corespondentă către o coloană în rezultatul transformat 304b.

[0054] De exemplu, o interogare de la o aplicație de client 116 către structura de date ierarhice 110 pentru obiecte mașină care au o valoare „FabricantA" pentru atributul „Fabricant_Id" poate avea ca rezultat „Vehicule/Mașini/Mașina1". Aplicația de client 116 poate face referire la obiecte mașină prin referirea la tabelul „Mașini". Aplicația de mapare 112 poate să transforme rezultatul „Vehicule/Mașini/Mașina1" în rezultatul „Mașini.Mașina1" pentru utilizarea de către aplicația de client 116.

[0055] Figura 4 este o diagramă bloc care prezintă dispozitive de calcul cu caracter de exemplu într-un mediu de calcul cu caracter de exemplu pentru implementarea anumitor modalități de realizare. Maparea aplicației 112 execută sau este întrebuințată în alt fel pe serverul sistem cu caracter de exemplu 102 şi este arătată utilizând componente funcționale sau module. După cum este cunoscut pentru un specialist din domeniu, un astfel de conținut electronic poate să existe în orice mediu potrivit netranzitoriu citibil pe calculator şi poate să fie executat pe orice procesor corespunzător.

[0056] De exemplu, după cum este arătat, un server sistem cu caracter de exemplu 102 poate să includă un mediu netranzitoriu citibil pe cxalculator, cum ar fi o memorie cu acces aleatoriu (RAM) 402, cuplată la un procesor 404 care execută instrucțiuni de program citibile pe calculator şi/sau informație de acces stocată într-o memorie 402. Un astfel de procesor 404 poate să includă un microprocesor, un circuit integrat specific pentru aplicație (ASIC), o mașină de stare, sau alt procesor şi poate să fie oricare dintr-un număr de procesoare de calculator. Un astfel de procesor poate să includă, sau poate să fie în comunicare cu un mediu netranzitoriu citibil pe calculator care stochează instrucțiuni care,

atunci când sunt executate de către procesorul 404, face ca procesorul 404 să realizeze etapele descrise aici.

[0057] Un mediu ne-tranzitoriu citibil pe calculator poate să includă, dar nu este limitat la, un dispozitiv electronic, optic, magnetic sau alt dispozitiv de stocare care poate să asigure un procesor cu instrucțiuni citibile pe calculator. Alte exemple includ, dar nu sunt limitate la, un fisc floppy, CD-ROM, DVD, disc magnetic, chip de memorie, ROM, RAM, un ASIC, un procesor configurat, stocare optică, bandă magnetică sau altă stocare magnetică, sau oricare alt mediu de la care un procesor de calculator poate să citească instrucțiunile. Instrucțiunile pot să aibă în componență instrucțiuni specifice pentru procesor generate de un compilator şi/sau un interpretor de la codul scris în oricare limbaj de programare de calculator corespunzător, inclusiv, de exemplu, C, C++, C#, Visual Basic, Java, Python, Perl, JavaScript, şi ActionScript.

[0058] Sistemul server 102 poate să primească intrare şi să asigure ieşire prin intermediul interfeței de intrare/ieşire (I/O) 408. Interfața I/O 408 poate să includă, de exemplu, o interfață de rețea pentru comunicarea prin intermediul rețelei 106. Este inclusă o magistrală, cum ar fi magistrala 406, în serverul sistem 102. Sistemul de server 102 poate să fie oricare tip de sistem de calcul inclus într-o rețea la un domeniu corespunzător pentru asigurarea uneia sau mai multora dintre caracteristicile descrise aici.

[0059] Figura 4 ilustrează un server sistem cu caracter de exemplu 102 care include, într-o memorie 402, o aplicație de mapare 112 şi o aplicație pentru accesul datelor 114. Aplicația de mapare 112 poate să configureze procesorul 404 pentru a primi apel de funcție prin intermediul interfeței I/O 408. Aplicația de mapare 112 poate să configureze procesorul 404 pentru a transforma apelul de funcție primit prin intermediul interfeței I/O 408 şi să asigure apelul de funcție transformat pentru aplicație pentru accesul datelor 114. Aplicația pentru accesul datelor 114 poate să configureze procesorul 404 să recupereze date de la o sursă de date 410 care are o structură de date ierarhică 110. O sursă de date 410 poate să fie oricare sursă de date care asigură date după solicitare, date împinse, sau asigură în alt mod articole de date pentru utilizarea de alte aplicații. În modalități de realizare alternative, sursa de date 410 poate să fie dispusă în

serverul sistem 102 sau poate să fie pusă la dispoziție de la un server sistem sau de la o altă locație exterioară. Aplicația pentru accesul datelor 114 poate să configureze procesorul 404 pentru a încărca date de la sursa de date 410 în memoria 402 şi să aplice apelul de funcție transformat la date pentru a obține un rezultat. Aplicația de mapare 112 poate să configureze procesorul 404 pentru a transforma rezultatul şi poate să scoată rezultatul transformat prin interfața I/O 408.

[0060] Figura 5 este o schemă logică care ilustrează o metodă cu caracter de exemplu 500 pentru transformarea apelurilor de funcție formate pentru interacțiunea cu o bază de date relațională. Pentru scopuri de ilustrare, metoda 500 este descrisă cu referire la fluxul de comunicații ilustrate în Figura 3 iar implementarea sistemului este ilustrată în Figura 4. Sunt totuşi posibile şi alte implementări.

[0061] Metoda 500 cu caracter de exemplu implică primirea unui apel de funcție format pentru a opera pe o bază de date relațională, după cum este arătat în blocul 510. Aplicația de mapare 112 poate să primească apelul de funcție. Apelul de funcție 302a poate referenția un obiect în baza de date relațională 108 prin referențierea a cel puțin unui tabel 202a al bazei de date relaționale 108 şi a cel puțin unei coloane a tabelului.

[0062] Metoda 500 cu caracter de exemplu implică de asemenea transformarea apelului de funcție 302a într-un apel de funcție format pentru interacționarea cu o structură de date ierarhică 110, după cum este arătat în blocul 520. Aplicația de mapare 112 poate să transforme apelul de funcție 302a în apelul de funcție transformat 302b.

[0063] În cadrul unei modalități de realizare cu caracter de exemplu, aplicația de mapare 112 poate transforma apelul de funcție 302a prin maparea unei referințe incluse în apelul de funcție dintr-un tabel într-o cale. De exemplu, calea poate să identifice o relație între un nod 208 care corespunde cu tabelul 202 şi un nod părinte 212 al nodului 208 care corespunde cu tabelul, după cum a fost descris mai sus cu referire la Figura 3.

[0064] În cadrul unor modalități de realizare suplimentare sau alternative, aplicația de mapare 112 poate mapa o referință inclusă în apelul de funcție 302a

dintr-un rând al tabelului 202a al bazei de date relaționale 108 într-un nod fiu al nodului care corespunde cu tabelul. Aplicația de mapare 112 poate de asemenea să mapeze o referință inclusă în apelul de funcție 302a dintr-o coloană a tabelului 202a al bazei de date relaționale 108 într-o proprietate a nodului fiu.

[0065] În modalități de realizare suplimentare sau alternative, aplicația de mapare 112 poate transforma apelul de funcție 302a prin maparea unei referințe în apelul de funcție 302a într-o relație între tabelele 202a-b ale bazei de date relaționale 108 într-o relație între un nod părinte şi un nod fiu, cum este relația părinte-fiu a nodului 208 cu nodurile 210a-c.

[0066] Metoda 500 cu caracter de exemplu implică de asemenea aplicarea apelului de funcție transformat la unul sau la mai multe noduri ale structurii de date ierarhice, după cum este arătat în blocul 530. De exemplu, un apel de funcție transformat care este o interogare pentru recuperarea unuia sau a mai multor obiecte poate să fie aplicată la unul sau mai multe noduri ale structurii de date ierarhice prin recuperarea unuia sau a mai multor noduri care corespund obiectelor. Un apel de funcție transformat care este o comandă pentru a modifica unul sau mai multe obiecte poate să fie aplicat la unul sau la mai multe noduri ale structurii de date ierarhică prin modificarea unuia sau a mai multor noduri care corespund cu obiectele. Un apel de funcție transformat care este o comandă pentru a şterge unul sau mai multe obiecte poate să fie aplicat la unul sau la mai multe noduri ale structurii de date ierarhice prin ştergerea a unuia sau a mai multor noduri care corespund cu obiectele.

Implementare cu caracter de exemplu folosind Java

[0067] În cadrul unei modalități de realizare cu caracter de exemplu, aplicația de mapare 112 poate să transforme un apel de funcție de la o aplicație de client 116 întrebuințând un pachet Java Persistence API („JPA") pentru administrarea bazelor de date relaționale într-un apel de funcție transformat folosind Content Repository API pentru Java („JCR").

[0068] JCR poate să includă sau să utilizeze mai multe limbaje de interogare, cum ar fi (fără a se limita la acestea) JCR-SQL şi XPATH. JCR-SQL poate partaja caracteristici altele decât limbajele de interogare bazate pe SQL la

folosirea cuvintelor cheie cum ar fi SELECTEAZĂ, DIN, UNDE etc. JCR-SQL poate să fie utilizat cu un model bazat pe nod sau cu alt model de stocare ierarhică. De exemplu, JCR-SQL asigură funcții precum CALE şi NUME pentru recuperarea informației specifice nodului care se poate să nu fie folosită într-un model de stocare de bază de date relațională. O astfel de funcție poate să permită la interogare structuri de date ierarhice sau alte sisteme de stocare bazate pe nod. XPATH poate să fie utilizată cu o structură de date ierarhică pentru a, de exemplu, extrage noduri Extensible Markup Language („XML") dintr-un document.

[0069] În cadrul unui exemplu, un apel de funcție JPA poate fi o interogare pentru a recupera exemple ale unui obiect, cum ar fi "SELECTEAZĂ e DE LA ANGAJAT e UNDE e.salariu > :salariu de bază." Pentru a transforma apelul de funcție JPA într-un apel de funcție JCR-SQL sau XPATH, aplicația de mapare 112 poate transforma referințele într-un tabel „Angajat". Exemplele unui apel de funcție transformat pot să includă un apel de funcție JCR-SQL "SELECTEAZĂ * DIN [nt:nestructurat] ca nod UNDE nod.salariu > :Salariu bază ŞI CALE(nod) CA '/tabele/angajați/%'" sau un apel de funcție XPATH "/jcr:rădăcină/tabele/angajați//element(*, nt:nestructurat)[@salariu > :Salariu bază]."

[0070] În cadrul altui exemplu, un apel de funcție JPA poate să fie o comandă pentru a actualiza exemplele unui obiect, cum ar fi "ACTUALIZEAZĂ Angajat e SETEAZĂ e.salariu = e.salariu * 1.2 UNDE e.angajatId = 1234." Aplicația de mapare 112 poate să transforme apelul de funcție JPA astfel încât să se acomodeze la absența unui apel de funcție ACTUALIZEAZĂ în JCR. Aplicația de mapare 112 poate să transforme formularea ACTUALIZEAZĂ pentru a selecta un obiect nod folosind un apel de funcție SELECTEAZĂ care se potriveşte cu criteriile formulării SQL ACTUALIZEAZĂ furnizate. Un astfel de apel de funcție SELECTEAZĂ poate să fie un apel de funcție JCR-SQL"SELECTEAZĂ * DIN [nt:nestructurat] ca nod UNDE nod.id = 1234 CALE(nod) CA '/tabele/angajați/%'" sau un apel de funcție XTATH "/rădăcină/angajați//element(*, nt:nestructurat)[@id = 1234]." Aplicația de mapare 112 poate să utilizeze un apel de funcție pentru modificarea unuia sau mai multor atribute ale nodului selectat, cum ar fi apelul de

funcție "nod.setează Proprietate("salariu", nod.obține Proprietate("salariu").obține Dublu() * 1.2)."

[0071] În cadrul altui exemplu, un apel de funcție JPA poate să fie o comandă pentru a şterge exemplele unui obiect, cum ar fi "ŞTERGE DIN Manager m UNDE e.angajați ESTE GOL." Aplicația de mapare 112 poate să transforme apelul de funcție JPA astfel încât să se acomodeze pentru absența unui apel de funcție ŞTERGE în JCR. Aplicația de mapare 112 poate să transforme formularea ŞTERGE pentru a selecta un obiect nod folosind un apel de funcție SELECTEAZĂ care se potriveşte cu criteriile formulării furnizate SQL ŞTERGE. Apelul de funcție SELECTEAZĂ poate recupera unul sau mai multe obiecte nod care corespund cu criteriile specificate. Un astfel de apel de funcție SELECT EAZĂ poate să fie un apel de funcție JCR-SQL "SELECTEAZĂ * DIN [nt:nestructurat] ca nod UNDEnod.angajați ESTE ZERO şi CALE(nod) CA '/tabele/manageri/%'" sau un apel de funcție XPATH "/rădăcină/manageri//element(*, nt:nestructurat)[nu(@angajați)]." Aplicația de mapare 112 poate să utilizeze un apel de funcție pentru a şterge un nod, cum ar fi "nod.îndepărtează()".

Generalități

[0072] Numeroase detalii specifice au fost expuse aici pentru a asigura o înțelegere de profunzime a subiectului revendicat. Cu toate acestea, specialiştii din domeniu vor înțelege că subiectul revendicat poate să fie pus în practică fără aceste detalii specifice. Alte exemple, metode, echipamente sau sisteme care pot să fie cunoscute de specialist nu au fost descrise în detaliu astfel încât să nu ascundă subiectul revendicat.

[0073] Unele porțiuni sunt prezentate în termeni de algoritmi sau reprezentări simbolice ale operațiilor pe biți de date sau semnale digitale binare stocate în memoria unui sistem de calcul, cum este memoria unui calculator. Aceste descrieri algoritmice sau reprezentări sunt exemple ale tehnicilor utilizate de către specialişti în domeniul procesării datelor pentru a transmite rodul muncii lor altor specialişti din domeniu. Un algoritm este o secvență consecventă cu ea însăşi de operații sau procesare similară care conduce la un rezultat dorit. În acest context,

operațiile sau prelucrarea implică manipularea fizică a cantităților fizice. În mod obişnuit, deşi nu necesar, astfel de cantități pot să ia forma unor semnale electrice sau magnetice care sunt capabile să fie stocate, transferate, combinate, comparate sau manevrate în alt fel. S-a dovedit convenabil în timp, în principal din motive de utilizare obişnuită, de se face referire la astfel de semnale ca la biți, date, valori, elemente, simboluri, semne, termeni, numere, numerale sau alte asemenea. Trebuie să fie înțeles, totuşi, că toate acestea şi alți termeni similari trebuie să fie asociați cu cantități fizice potrivite şi sunt în special nişte denumiri. În afara cazului că a fost în mod precis altfel formulat, se va înțelege că în întreaga această specificație discuțiile care utilizează termeni cum ar fi „procesare", „calculare" „socotire", „determinare" şi „identificare" sau alții asemănători se referă la acțiuni şi procese ale unui dispozitiv de calcul, cum ar fi unul sau mai multe calculatoare sau un dispozitiv sau dispozitive de calcul elctronic similare, care manevrează sau transformă date reprezentate ca cantități fizice electronice sau magnetice din memorii, regiştrii sau alte dispozitive de stocare, dispozitive de transmisie sau dispozitive de afişare ale platformei de calcul.

[0074] Sistemul sau sistemele discutate aici nu sunt limitate la nicio anumită arhitectură sau configurație de hardware. Un dispozitiv de calcul poate să includă oricare aranjament corespunzător de componente care asigură un rezultat condiționat sau unul sau mai multe apeluri de funcție. Dispozitive de calcul corespunzătoare includ sistemele de calcul bazate pe microprocesoare multi-destinație care accesează software care programează sau configurează sistemul de calcul al unui echipament de calcul de destinație generală într-un echipament de calcul care implementează una sau mai multe modalități de realizare a prezentului subiect. Pentru implementarea cunoştințelor conținute aici în software-ul care urmează să fie folosit la programarea sau configurarea unui dispozitiv de calcul, poate să fie folosit orice tip de limbaj de programare, de scripting sau alt tip de limbaj sau combinație de limbaje.

[0075] Modalitățile de realizare a metodelor prezentate aici pot să fie realizate în funcționarea unor astfel de dispozitive de calcul. Ordinea blocurilor prezentate în exemplele de mai sus poate să fie diferită – de exemplu, blocurile pot să fie re-

ordonate, combinate şi/sau sparte în sub-blocuri. Anumite blocuri sau procese pot să fie realizate în paralel.

[0076] Utilizarea aici a lui "adaptat pentru" sau „configurat pentru" este înțeleasă ca un limbaj deschis şi incluziv care nu exclude dispozitive adaptate pentru sau configurate pentru a realiza sarcini sau paşi suplimentari. În plus, utilizarea lui „bazat pe" este înțeleasă a fi deschisă şi incluzivă, prin aceea că un proces, pas, calculare sau altă acțiune „bazată pe" una sau mai multe condiții sau valori expuse poate, în parctică, să se bazeze pe condiții sau valori suplimentare în afara celor menționate. Titluri, liste şi numerotatea incluse aici sunt numai pentru uşurarea explicației şi nu sunt destinate a avea un scop limitativ.

[0077] Chiar dacă prezentul subiect a fost descris în detaliu cu referire la modalități de realizare specifice ale acestuia, este de la sine înțeles că specialiştii din domeniu, după ce vor ajunge la înțelegerea celor de mai sus, vor putea realiza cu uşurință modificări la, variații ale şi echivalențe cu astfel de modalități de realizare. În consecință, este de la sine înțeles că prezenta prezentare a fost prezentată numai pentru scopuri de exemplu şi nu de limitare, şi nu împiedică includerea unor astfel de modificări, variații şi/sau completări ale prezentului subiect după cum va fi cu uşurință evident pentru specialistul din domeniu.

REVENDICĂRI

1. Metodă care are în componență:

primirea, de către un procesor, a unui apel de funcție format pentru a opera pe o bază de date relațională;

transformarea, de către procesor, a apelului de funcție într-un apel de funcție transformat format pentru interacțiunea cu o structură de date ierarhică, în care structura de date ierarhică are în componență mai multe noduri, în care fiecare nod are un singur nod părinte; şi

aplicarea, de către procesor, a apelului de funcție transformat la unul sau la mai multe noduri ale structurii de date ierarhică.

2. Metodă în conformitate cu revendicarea 1, în care apelul de funcție referențiază un obiect din baza de date relațională prin referențierea a cel puțin unui tabel al bazei de date relaționale şi a cel puțin unei coloane a acelui cel puțin un tabel.

3. Metodă în conformitate cu revendicarea 2, în care apelul de funcție are în componență maparea unei prime referințe incluse în apelul de funcție dintr-un tabel către o cale, în care tabelul include un rând care corespunde cu obiectul, în care calea identifică o relație între un nod care corespunde cu obiectul şi cel puțin un nod adițional care este părinte al nodului care corespunde cu obiectul.

4. Metodă în conformitate cu revendicarea 3, în care apelul de funcție are de asemenea în componență:

maparea unei a doua referințe incluse în apelul de funcție dintr-un rând al tabelului bazei de date relaționale într-un nod fiu al nodului care corespunde cu tabelul; şi

maparea unei a treia referințe incluse în apelul de funcție dintr-o coloană a tabelului bazei de date relaționale într-o proprietate a nodului fiu.

5. Metodă în conformitate cu revendicarea 4, în care transformarea apelului de funcție are de asemenea în componență maparea unei referințe din apelul de funcție într-o relație dintre un prim tabel al bazei de date relaționale şi un al doilea tabel al bazei de date relaționale într-o relație dintre un prim nod care corespunde unui prim tabel şi un al doilea nod care corespunde unui al doilea tabel.

6. Metodă în conformitate cu revendicarea 1, în care apelul de funcție are în componență o comandă pentru a realiza cel puțin una dintre stocarea obiectului, modificarea obiectului sau ştergerea obiectului.

7. Metodă în conformitate cu revendicarea 1, în care apelul de funcție are de asemenea în componență:

generarea, de către procesor, a rezultatului pe baza aplicării apelului de funcție transformat la unu sau la mai multe noduri ale structurii de date ierarhice; şi

transformarea, de către procesor, a rezultatului astfel încât rezultatul este format pentru a interacționa cu baza de date relațională.

8. Metodă în conformitate cu revendicarea 1, în care apelul de funcție are în componență o comandă Java Persistance Application Programing Interface şi în care apelul de funcție transformat are în componență o comandă Java Content Repository Application Programing Interface.

9. Sistem care are în componență:

un procesor configurat pentru a executa instrucțiuni stocate într-un mediu netranzitoriu citibil pe calculator care asigură o aplicație de mapare;

în care aplicația de mapare are în componență unul sau mai multe module configurate pentru a performa operații care au în componență:

primirea unui apel de funcție format pentru a opera pe o bază de date relațională;

transformarea apelului de funcție într-un apel de funcție transformat format pentru a interacționa cu o structură de date ierarhică, în care structura de date ierarhică are în componență mai multe noduri, în care fiecare nod are un singur nod părinte; şi

aplicarea apelului de funcție transformat la unul sau la mai multe noduri ale structurii de date ierarhice.

10. Sistem în conformitate cu revendicarea 9, în care apelul de funcție referențiază un obiect în baza de date relațională prin referențierea a cel puțin unui tabel din baza de date relațională şi cel puțin unei coloane a acelui cel puțin un tabel.

11. Sistem în conformitate cu revendicarea 10, în care sunt configurate unul sau mai multe module pentru a transforma apelul funcție prin maparea a cel puțin unei referințe incluse în apelul de funcție dintr-un tabel într-o cale, în care tabelul include un rând care corespunde cu obiectul, în care calea identifică o relație între un nod care corespunde cu obiectul şi cel puțin un nod adițional care este părinte al unui nod care corespunde cu obiectul.

12. Sistem în conformitate cu revendicarea 11, în care sunt configurate unul sau mai multe module pentru a transforma apelul de funcție prin îndeplinirea operațiunilor suplimentare care au în componență:

maparea unei a doua referințe incluse în apelul de funcție dintr-un un rând al unui tabel al bazei de date relaționale într-un nod fiu al nodului care corespunde cu tabelul; şi

maparea unei a treia referințe incluse în apelul de funcție de la o coloană a tabelului bazei de date relaționale la o proprietate a nodului fiu.

13. Sistem în conformitate cu revendicarea 11, în care sunt configurate modulul sau mai multe module pentru a transforma apelul de funcție prin îndeplinirea unor operații suplimentare care au în componență maparea unei referințe din apelul de funcție într-o relație între un prim tabel al bazei de date relaționale şi un al doilea tabel al bazei de date relaționale într-o relație între un prim nod care corespunde cu primul tabel şi un al doilea nod care corespunde cu al doilea tabel.

14. Sistem în conformitate cu revendicarea 9, în care sunt configurate modulul sau mai multe module pentru a îndeplini operații suplimentare care includ:

generarea unui rezultat pe baza aplicării apelului de funcție transformat acelui nod sau mai multor noduri ale structurii de date ierarhice; şi

transformarea rezultatului astfel încât rezultatul este format pentru a interacționa cu baza de date relațională.

15. Sistem în conformitate cu revendicarea 9, în care apelul de funcție are în componență o comandă Java Persistence Application Programming Interface şi în care apelul de funcție transformat are în componență o comandă Java Content Repository Application Programming Interface.

16. Mediu netranzitoriu citibil de calculator care încorporează un cod de program executabil de către un sistem de calcul, mediul netranzitoriu citibil pe calculator:

cod de program pentru primirea unui apel de funcție format pentru a opera pe o bază de date relațională;

cod de program pentru transformarea apelului de funcție într-un apel de funcție transformat pentru a interacționa cu o structură de date ierarhică, în care structura de date ierarhică are în componență mai multe noduri, în care fiecare nod are un singur nod părinte; şi

cod de program pentru aplicarea apelului de funcție transformat la unul sau la mai multe noduri ale structurii de date ierarhice.

17. Mediu netranzitoriu citibil pe calculator în conformitate cu revendicarea 16, în care apelul de funcție referențiază un obiect din baza de date relațională prin referențierea a cel puțin unui tabel din baza de date relațională şi cel puțin unei coloane a acelui cel puțin un tabel.

18. Mediu netranzitoriu citibil pe calculator în conformitate cu revendicarea 17, în care codul de proigram pentru transformarea apelului de funcție are în componență codul de program pentru maparea unei prime referințe incluse în apelul de funcție dintr-un tabel într-o cale, în care tabelul include un rând care corespunde cu obiectul, în care calea identifică o relație între un nod care corespunde cu obiectul şi cel puțin un nod suplimentar care este părinte al nodului care corespunde cu obiectul.

19. Mediu netranzitoriu citibil pe calculator în conformitate cu revendicarea 18, în care codul de program pentru transformarea apelului de funcție are de asemenea în componență:

cod de program pentru maparea unei a doua referințe incluse în apelul de funcție dintr-un rând al tabelului al bazei de date relaționale într-un nod fiu al nodului care corespunde cu tabelul; şi

cod de program pentru maparea unei a treia referințe incluse în apelul de funcție dintr-o coloană al tabelului bazei de date relaționale într-o proprietate a nodului fiu.

20. Mediu netranzitoriu citibil pe calculator în conformitate cu revendicarea 16, care are de asemenea în compoziție:

cod de program pentru generarea unui rezultat pe baza aplicării apelului de funcție transformat unuia sau mai multor noduri ale structurii de date ierarhice: şi

cod de program pentru transformarea rezultatului astfel ca rezultatul este format pentru a interacționa cu baza de date relațională.

a-2012-00074--
31 -01- 2012

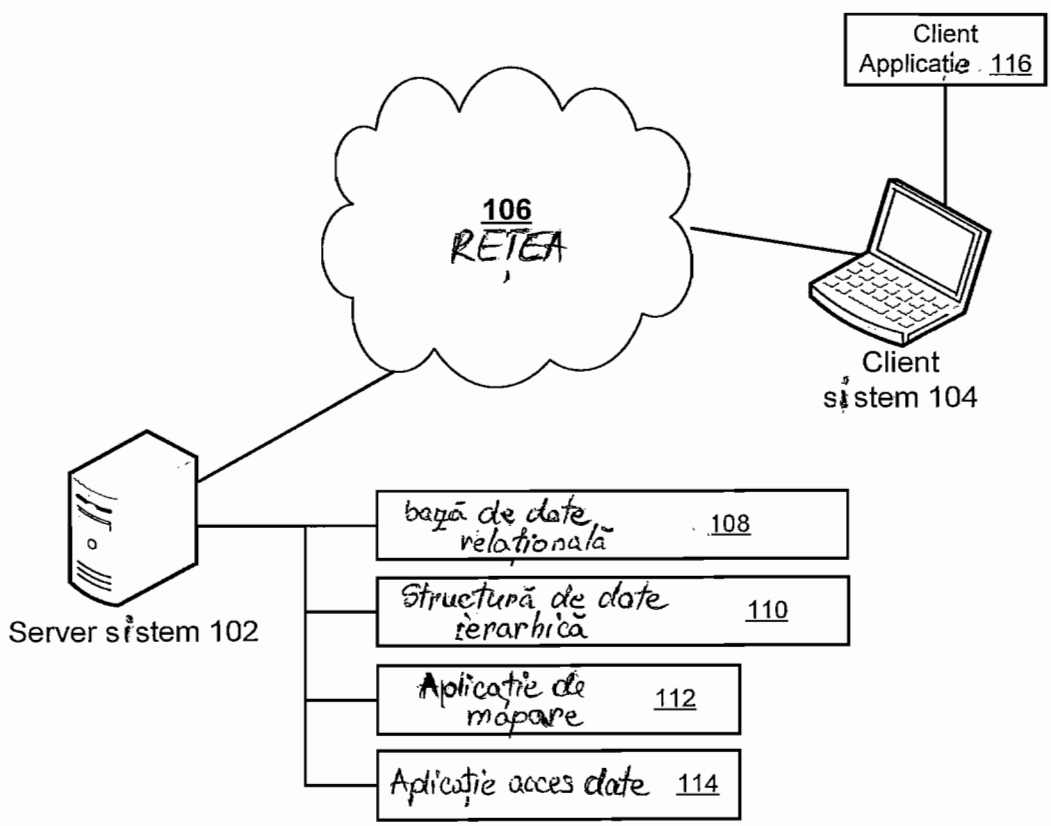

FIG. 1

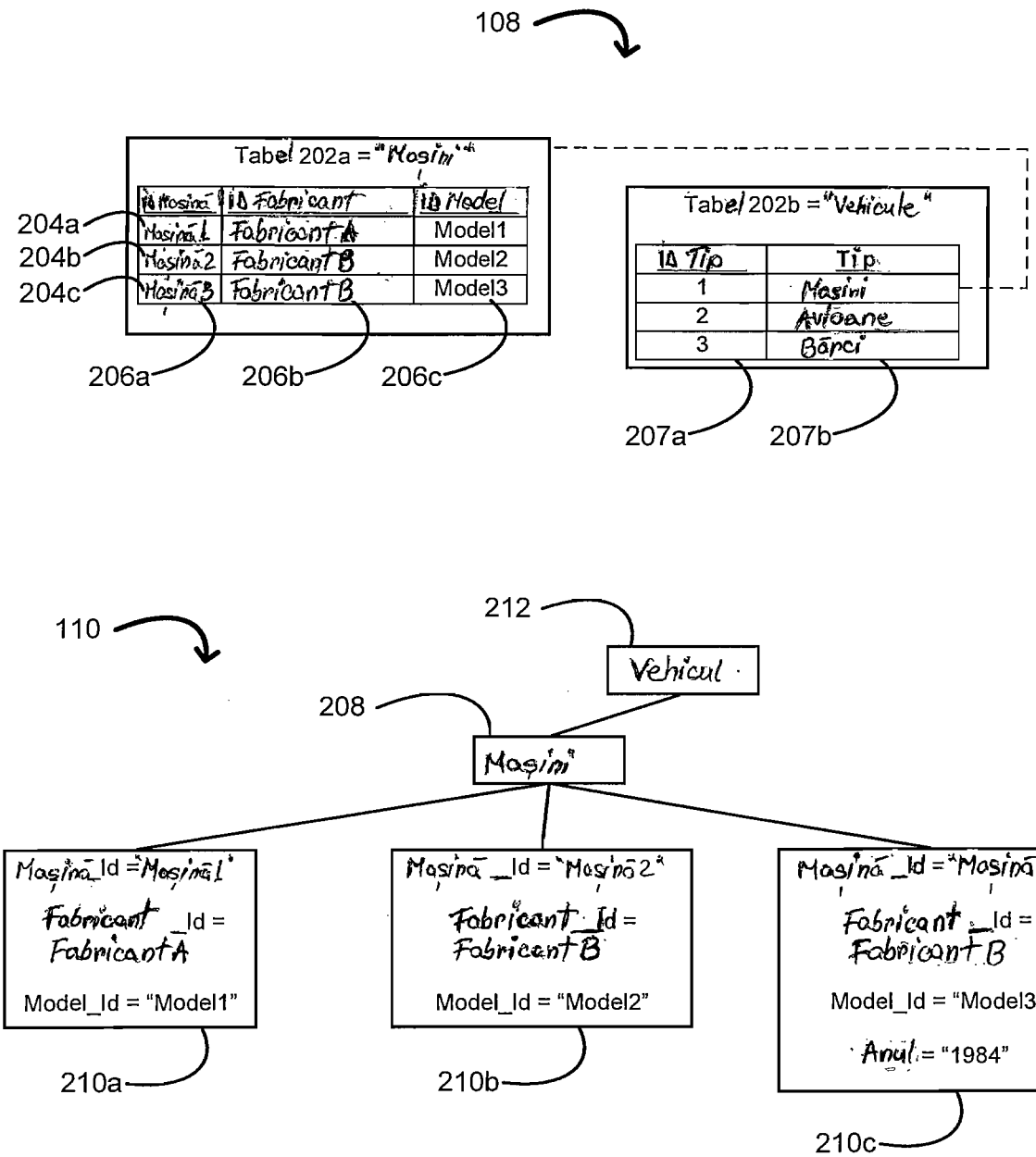
108
Tabel 202a = "Masini"
ID Masina
ID Fabricant
ID Model
Masina1
Fabricant A
Model1
Masina2
Fabricant B
Model2
Masina3
Fabricant B
Model3
204a
204b
204c
206a
206b
206c
Tabel 202b = "Vehicule"
ID Tip
Tip
1
Masini
2
Avioane
3
Barci
207a
207b
110
212
Vehicul
208
Masini
Masina_Id = "Masina1"
Fabricant_Id = Fabricant A
Model_Id = "Model1"
Masina_Id = "Masina2"
Fabricant_Id = Fabricant B
Model_Id = "Model2"
Masina_Id = "Masina3"
Fabricant_Id = Fabricant B
Model_Id = "Model3"
Anul = "1984"
210a
210b
210c

FIG. 2

a-2012-00074--
31-01-2012

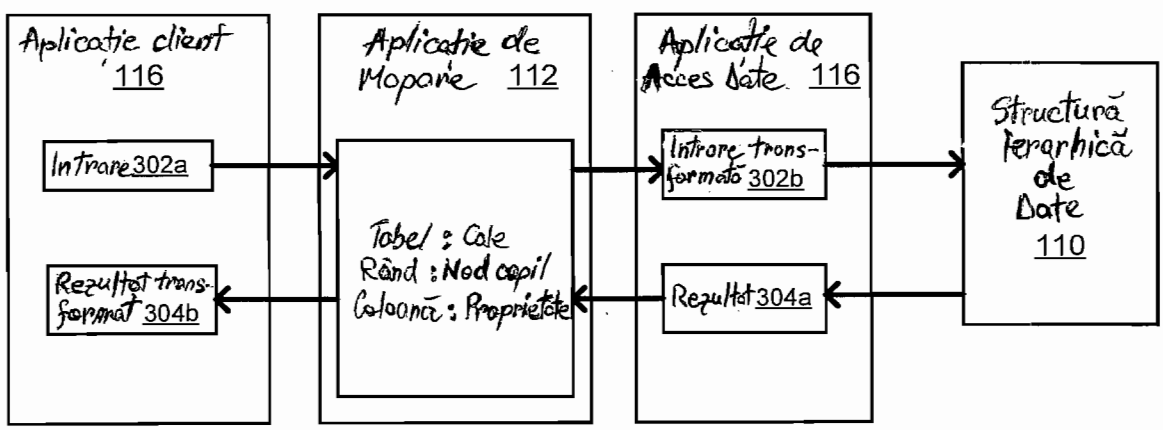

FIG. 3

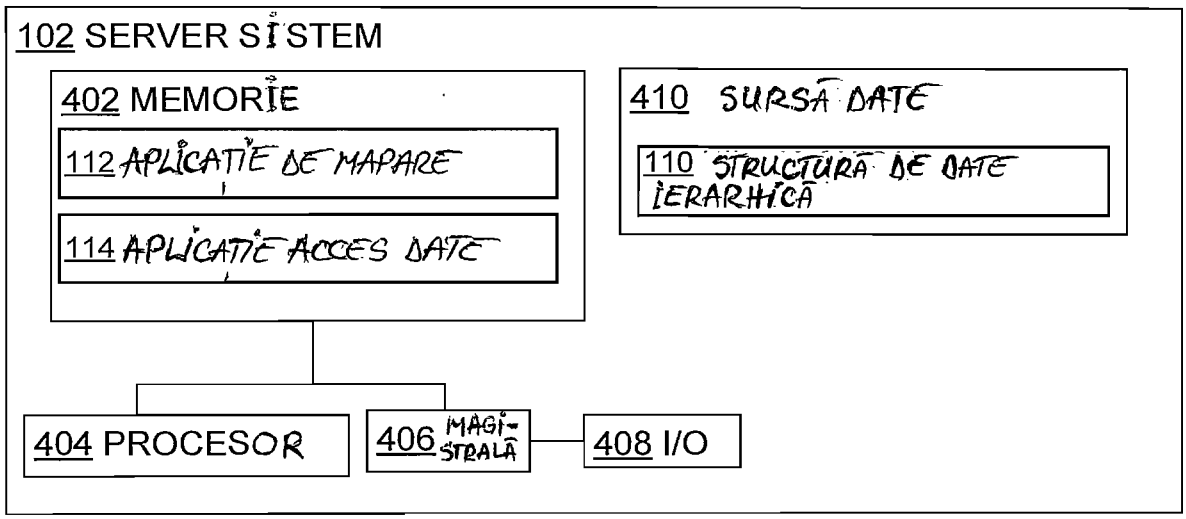
102 SERVER SISTEM
402 MEMORIE
112 APLICATIE DE MAPARE
114 APLICATIE ACCES DATE
410 SURSĂ DATE
110 STRUCTURĂ DE DATE IERARHICĂ
404 PROCESOR
406 MAGISTRALĂ
408 I/O

FIG. 4

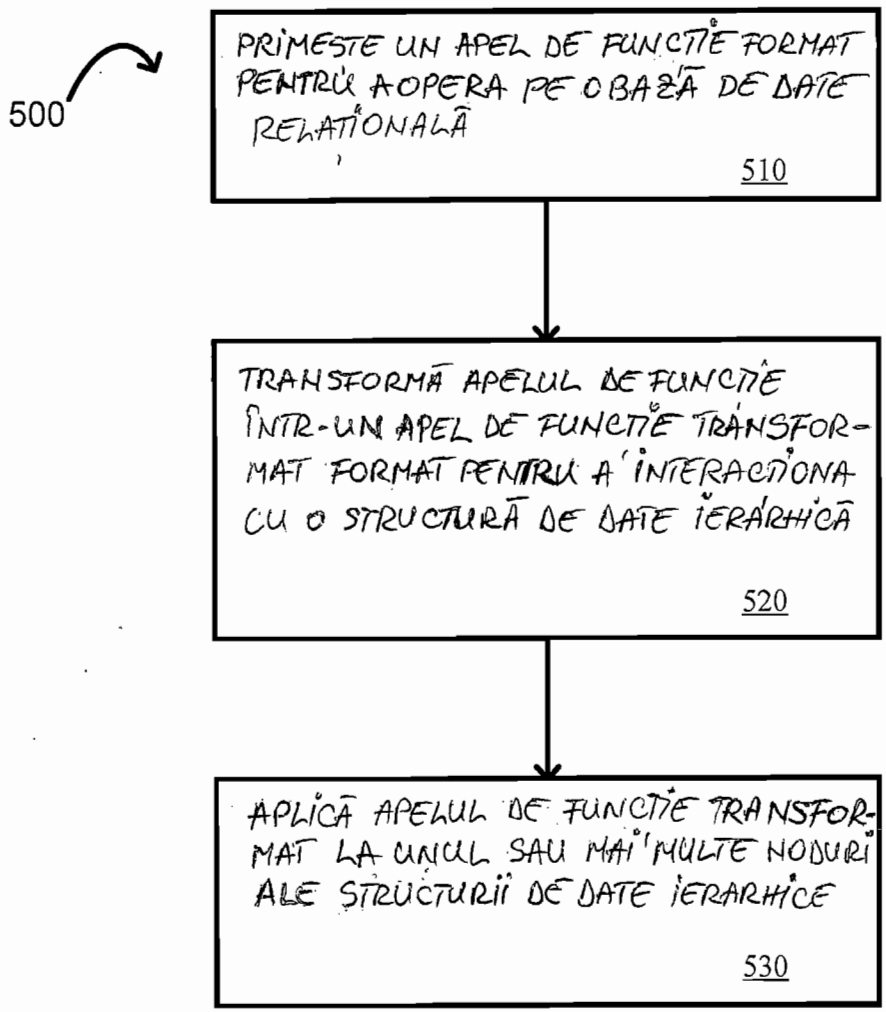

FIG. 5