

4. Extensii ale modelului

Modelul entitate-asociere clasic are unele lipsuri in ceea ce priveste posibilitatea modelarii caracteristicilor asociate unor subclase de obiecte modelate prin simple entitati. Pentru aceasta, la modelul original au fost adaugate doua noi concepte: **ierarhia de generalizare** și **ierarhia de incluziune**. Prima defineste partitionarea instantelor unei entitati in **n** subclase diferite iar a doua permite clasarea unora dintre instantele unei entitati in **m** subclase care nu reprezinta o partitie in sens matematic. Din punct de vedere formal, cele doua concepte se pot defini astfel:

Definitie (ierarhia de incluziune): O entitate E_1 este o submultime a entitatii E (sau este inclusa in entitatea E) daca fiecare instanta a lui E_1 este de asemenea o instanta a lui E .

Un exemplu de incluziune este definirea in cadrul entitatii ANGAJATI a unor subclase modelate prin entitatile INGINERI, ECONOMISTI si COLABORATORI.

In cazul ierarhiei de incluziune entitatile fiu pot sa nu fie disjuncte doua cite doua: pentru exemplul dat, exista angajati ingineri si care sunt incadrati cu contract de colaborare. De asemenea reuniunea lor poate sa nu acopera in intregime entitatea tata: exista angajati care nu sunt nici ingineri, nici economisti si nici colaboratori.

Definitie (ierarhia de generalizare): O entitate E este generalizarea entitatilor E_1, E_2, \dots, E_n daca orice instanta a lui E este de asemenea instanta in una și numai una din entitățile E_1, E_2, \dots, E_n .

Un exemplu de generalizare este clasarea instantelor entității ANGAJATI in subclasele BARBATI și FEMEI.

Caracteristica ierarhiei de generalizare este ca din punct de vedere matematic entitățile fii reprezintă o partiție a entității tata:

- $E_1 \cup E_2 \cup \dots \cup E_n = E$ și
- $E_i \cap E_j = \emptyset$ pentru orice $i \neq j$ din intervalul $1..n$

Ierarhiile de incluziune și generalizare se folosesc doar in cazul in care pentru subclasele unor clase modelate prin entitati este nevoie de stocarea unor informatii suplimentare specifice.

In cazul unei baze de date de personal este nevoie de exemplu sa fie memorat numarul de copii ai fiecarui angajat. Acest fapt se poate modela in doua feluri: fie prin adaugarea la entitatea ANGAJATI a unui atribut suplimentar *NumarCopii* (care va avea valoarea 0 pentru angajatii fara copii) fie prin aparitia unei entitati suplimentare CU_COPII aflata intr-o relatie de incluziune cu entitatea ANGAJATI și care va avea ca atribute de identificare pe cele ale tatalui iar ca atribut descriptiv numarul de copii, acesta fiind atributul specific subclasei.

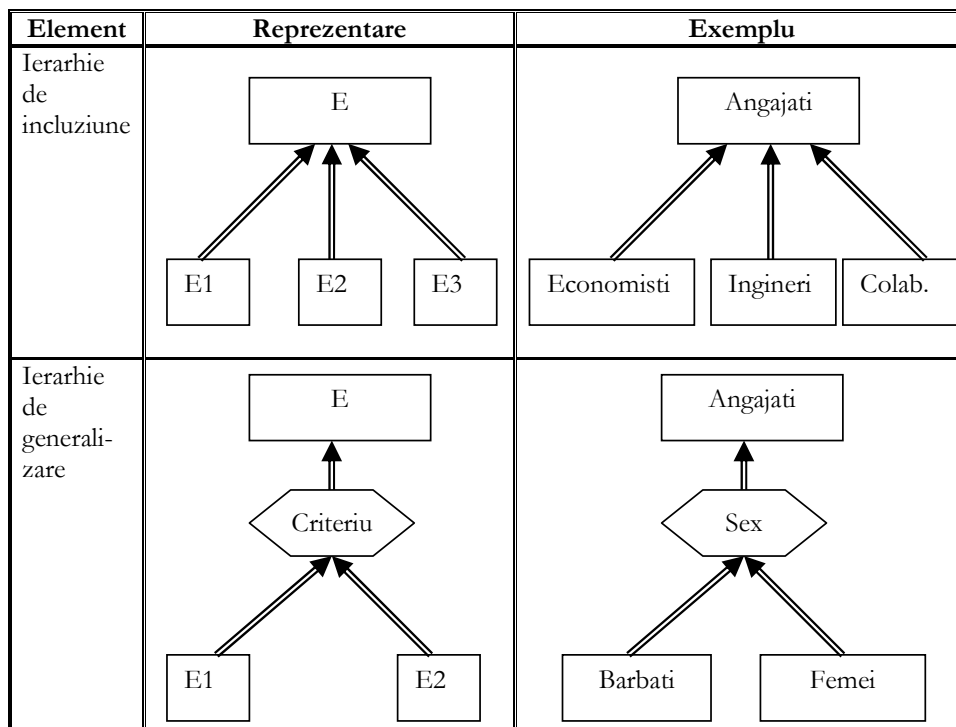


Fig. 2.3. Conventia grafica de reprezentare grafica a ierarhiilor

Vom alege a doua varianta in cazul in care numarul angajatilor cu copii este mult mai mic decat numarul total al angajatilor, fapt care va duce la economie de spatiu pe disc: in acest

caz o noua entitate - din care va rezulta o tabela a bazei de date – va ocupa mai puțin spațiu decât un atribut suplimentar - din care rezulta o coloana suplimentară a tablei SALARIATI.

Convenția grafică de reprezentare a celor două tipuri de ierarhii se găsește în fig. 2.3.

2.2.3. Caracteristici ale elementelor modelului

Așa cum entitățile au atribute care specifică diversele proprietăți ale clasei de obiecte modelate, și asocierile au caracteristici care aduc informații suplimentare. Acestea sunt următoarele:

Gradul asocierii

Este o valoare numerică întreaga și este dată de numărul de entități care participă la acea asocieră. Asocierile de grad 1, 2 și 3 se mai numesc și asocieri *unare*, *binare* și respectiv *ternare*.

Pentru exemplificare vom considera o bază de date conținând informații despre studenți, proiectele realizate de aceștia, calculatoarele pe care au alocate ore de lucru și facultățile la care sunt înscriși. De asemenea vom considera că unii dintre studenți au un *tutor* care îi îndrumă, acesta fiind un student dintr-un an mai mare.

Diagrama EA a bazei de date este prezentată în figura 2.4. Pentru simplificarea figurii, nu s-au reprezentat decât entitățile și asocierile dintre ele nu și atributele fiecărei entități în parte.

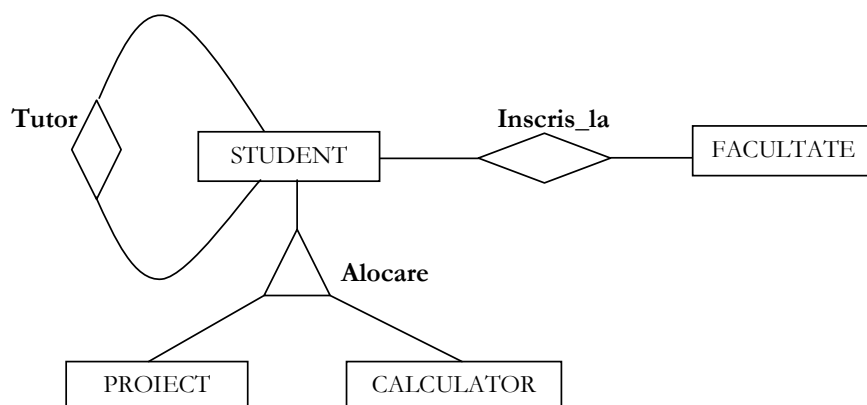


Fig. 2.4. Exemple de asocieri de grad 1, 2 și 3

Asocieră **TUTOR** este o asocieră unară deoarece la ea participă doar entitatea **STUDENT**. Aceasta asocieră arată cine este tutorul fiecărui student (dacă există).

Asocieră **INSCRIS_LA** este o asocieră binară între entitățile **STUDENT** și **FACULTATE** și arată la ce facultate/facultăți este înscris fiecare student.

Asocieră **ALOCARE** este o asocieră ternară între entitățile **STUDENT**, **PROIECT** și **CALCULATOR**. Ea modelează pe ce calculatoare are alocate ore de lucru fiecare student pentru fiecare proiect.

Un exemplu de asocieră de grad mai mare ca trei este orarul unui an de studiu al unei facultăți. Acesta este o asocieră între următoarele entități:

- GRUPE. Fiecare grupa are un cod unic.
- SALI. Salile sunt etichetate printr-un indicativ alfanumeric.
- INTERVALE ORARE. Un interval orar este un triplet (Zi, De la ora, La ora)
- ACTIVITATE. Este o activitate prezenta in orar (curs, laborator, seminar, proiect).
- PROFESOR. Este cadrul didactic titular pentru o activitate

Modelarea acestor clase de obiecte ca entitati presupune faptul ca in baza de date sunt stocate si alte informatii despre ele. Diagrama EA este prezentata in figura 2.5. Ea modeleaza programarea activitatilor didactice efectuate de profesori pe intervale orare, sali si grupe.

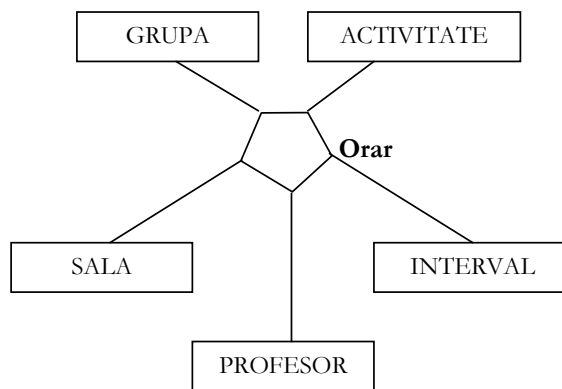


Fig. 2.5. Asociere de grad 5

Conectivitatea asocierii

Este specifica fiecărei ramuri a unei asocieri și poate avea una din următoarele două valori: **unu** sau **multi**. Determinarea ei pentru ramura spre o entitate E se face astfel: fixand arbitrar cite o instanta pentru celelalte entitati care participa la asociere se pune intrebarea: cite instante ale lui E pot fi conectate cu acestea? Daca poate fi cel mult una, conectivitatea ramurii este **unu**, altfel conectivitatea este **multi**.

Pentru exemplul din figura 2.4.:

- Asocierea TUTOR este unu-unu sau multi-uni dupa cum un student poate fi tutor pentru un singur alt student sau pentru mai multi studenti de an inferior.
- Asocierea INSCRIS_LA este multi-unu (multi spre STUDENT) sau multi-multi dupa cum un student poate fi inscris la una sau mai multe facultati.
- Asocierea ternara ALOCARE (aplicam definitia):
 - ramura spre STUDENT: fiind dat un proiect si un calculator, citi studenti au ore alocate pe acel calculator pentru respectivul proiect? Presupunand ca mai multi studenti lucreaza pentru acelasi proiect pe acelasi calculator ramura va fi **multi**.
 - ramura spre PROIECT: fiind dat un student și un calculator, la cite proiecte are acesta alocate ore pe acel calculator? Presupunand ca pentru fiecare proiect exista un calculator dedicat, ramura va fi **unu**.
 - ramura spre CALCULATOR: fiind dat un student și un proiect, pe cate calculatoare are alocate acesta ore pentru realizarea proiectului? Presupunand ca la un proiect se lucreaza pe un singur calculator, ramura va fi **unu**.

Deci asocierea ALOCARE este multi-unu-unu.

Observam ca raspunsul la fiecare din cele trei intrebari se da in functie de realitatea modelata. Aceeasi asociere poate avea conectivitati diferite in cazuri diferite: daca exista chiar si un singur proiect la care un student are ore alocate pe mai mult de un calculator, ramura spre CALCULATOR va fi **multi** iar asocierea va fi multi-unu-multi.

Conventia de reprezentare grafica a conectivitatii folosita in aceasta prezentare este urmatoarea: ramurile 'unu' vor fi reprezentate sub forma de sageata. In figura 2.6. sunt prezentate cele trei asocieri avand figurata si conectivitatea. S-a presupus ca asocierile TUTOR si INSCRIS_LA sunt unu-unu respectiv multi-unu.

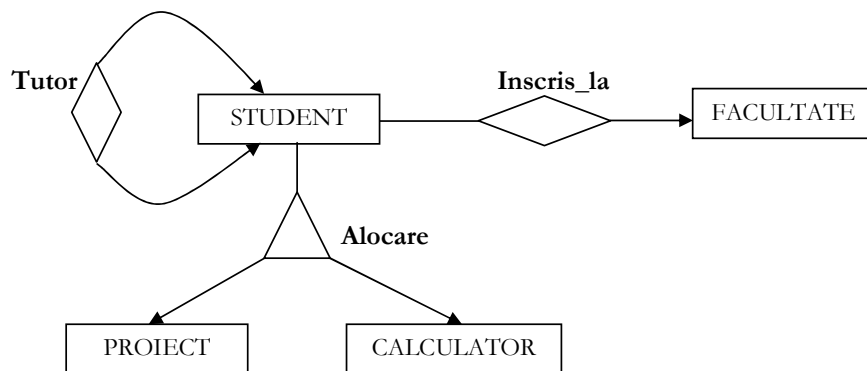


Fig. 2.6. Reprezentarea conectivitatii

Obligativitatea asocierii

Ca si conectivitatea, aceasta se determina pentru fiecare ramura si poate avea doar una din urmatoarele valori: **obligatorie** sau **optionala**. Determinarea ei pentru ramura spre o entitate E se face astfel: fixand arbitrar cite o instanta pentru celelalte entitati care participa la asociere se pune intrebarea: este obligatoriu sa existe o instanta a lui E asociata cu acestea? Daca raspunsul este 'Da' ramura este **obligatorie** altfel este **optionala**.

In exemplul anterior ramurile asocierilor TUTOR si ALOCARE sunt optionale iar cele ale asocierii INSCRIS_LA sunt obligatorii deoarece:

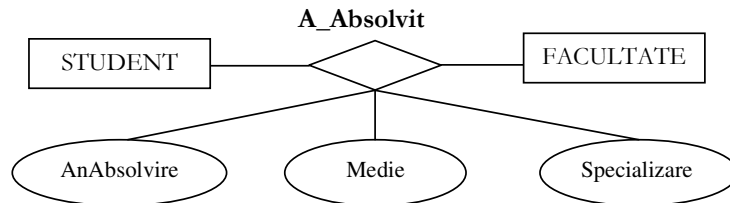
- Pentru asocierea TUTOR: exista studenti care nici nu au un tutor nici nu sunt tutori pentru alti studenti
- Pentru asocierea ALOCARE:
 - Un student la un proiect poate sa nu aiba alocate ore pe nici calculator (de exemplu in cazul unui proiect la o materie umanista)
 - Un student si un calculator respectiv un calculator si un proiect pot sa nu fie asociati prin alocare de ore de lucru (de exemplu pentru calculatoarele din birourile cadrelor didactice)
- Pentru asocierea INSCRIS_LA: nu exista studenti care nu sunt inscristi la nici o facultate si nici facultati fara studenti inscristi.

Conventia de reprezentare grafica a clasei de apartenenta folosita in continuare este urmatoarea: ramurile obligatorii vor fi reprezentate prin linie continua iar cele optionale prin linie intrerupta. In figura 2.7 este prezentata diagrama anterioara avand figurata si obligativitatea.

Daca gradul și conectivitatea unei asocieri sunt folosite in proiectarea conceptuala a schemei bazei de date, obligativitatea se modeleaza pentru definirea unui criteriu de integritate specificand posibilitatea de aparitie a valorilor nule: la transformarea diagramei EA in model relational attributele tabelor care modeleaza informatia reprezentata de asocieri pot avea sau nu valori nule dupa cum ramurile acestora sunt optionale sau obligatorii.

Atributele asocierilor

In unele cazuri o anumita informatie descriptiva nu este asociata cu o clasa de obiecte ci cu un ansamblu de clase diferite modelate fiecare prin entitati. In acest caz aceasta va fi modelata ca un atribut al asocierii dintre entitatile respective. Sa luam de exemplu cazul unei asocieri multi-multi A_ABSOLVIT intre entitatile STUDENT și FACULTATE care contine informatii privind facultatile absolvite anterior de unii studenti.



In acest caz informatii ca anul absolvirii, media, specializarea nu pot fi conectate nici la STUDENT (pentru ca un student poate fi absolventul mai multor facultati in ani diferiti, cu medii diferite, etc.) si din motive similare nici la FACULTATE. Ele descriu asocierea unui student cu o facultate si de aceea vor fi atasate asocierii A_ABSOLVIT. Toate attributele unei asocieri sunt attribute descriptive, neexistand in acest caz un identificator al asocierii.

Rolul

In cazul in care de la o asociere pornesc mai multe ramuri catre aceeasi entitate, fiecareia dintre acestea i se poate asocia un rol. Acesta arata semnificatiile diferite pe care le are aceeasi entitate in cadrul asocierii respective.

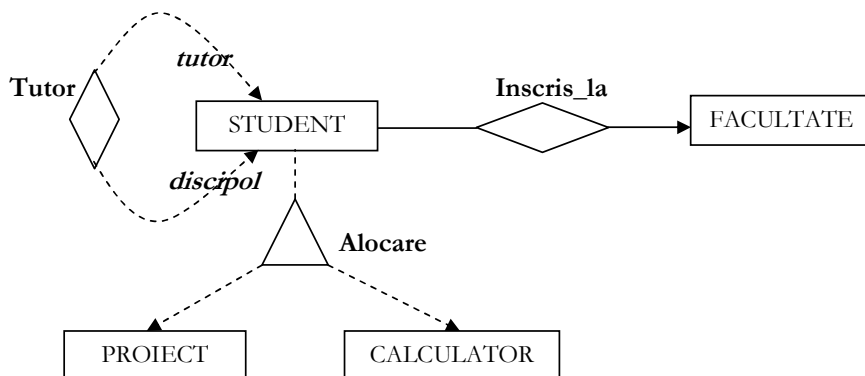


Fig. 2.7. Reprezentarea obligativitatii. Roluri

In cazul asocierii TUTOR cele doua ramuri pot fi etichetate de exemplu cu **tutor** și **discipol** aratand ca instante diferite ale aceleiasi entitati au rolurile respective (fig. 2.7.).

2.2.4. Criterii de modelare

a. Clasificarea in entitati și atribute

Deși definiția notiunilor de entitate, atribut, asociere este destul de simplă, în practica modelării apar dificultăți în clasificarea diverselor informații într-una din aceste categorii. De exemplu în cazul sediilor unei banci localizate în diverse orașe: obiectul ORAS este entitate distinctă sau atribut descriptiv al entității SEDIU?

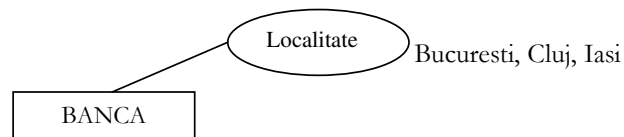
Pentru a putea clasifica corect informațiile, există câteva reguli care trebuie respectate și pe care le prezentăm în continuare. Prima regulă da un criteriu general de împărțire în entități și atribute, următoarele două semnalează excepții iar ultimele două reguli au un caracter mai puțin normativ ci mai degrabă orientativ.

Regula 1. Entitățile au informații descriptive, pe când atributele nu posedă astfel de informații.

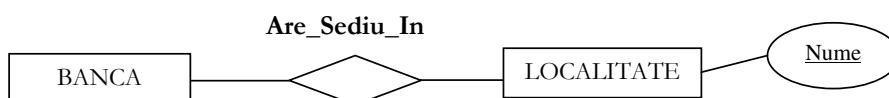
Dacă există informații descriptive despre o anumită clasă de obiecte, aceasta va fi modelată ca o entitate. În cazul în care pentru acea clasă de obiecte nu este nevoie decât de un identificator (codul, denumirea, etc), ea va fi modelată ca un atribut. De exemplu dacă despre un ORAS este necesară stocarea în baza de date unor informații ca JUDEȚ, POPULAȚIE, etc. atunci ORAS va fi o entitate. Dacă singura informație necesară este numele sau atunci NUME_ORAS va fi un atribut al altei entități.

Regula 2. Atributele multivaloarea vor fi reclassificate ca entități.

Dacă la o valoare a unui identificator corespund mai multe valori ale unui descriptor, acesta va fi clasat ca entitate. De exemplu, în cazul unei baze de date privind localizarea în teritoriu a unor banci, dacă se memorează informații doar despre banci care au un singur sediu, LOCALITATE este atribut al entității BANCA.

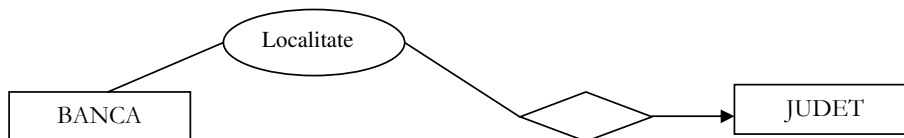


Dacă însă se memorează informații despre banci care au sucursale și filiale în diverse localități, deci pentru o singură bancă (o valoare a identificatorului entității BANCA) avem mai multe localități în care aceasta are sedii (mai multe valori ale descriptorului LOCALITATE), atunci LOCALITATE va fi entitate distinctă deși nu are decât un singur atribut. Pentru a modela localizarea sediilor în diverse localități între cele două entități va exista o asociere binară unu-multi (unu spre BANCI) numită de exemplu ARE_SEDIU_IN.

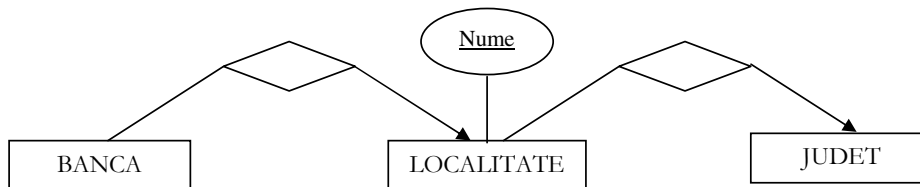


Regula 3. Atributele unei entitati care au o asociere multi-unu cu o alta entitate vor fi reclassificate ca entitati.

Asa cum am vazut asocierile pot lega doar entitati. Daca un descriptor al unei entitati este intr-o relatie multi-unu cu o alta entitate acel descriptor va fi trecut in categoria entitatilor. De exemplu, daca avem entitatile BANCA avand ca atribut descriptiv monovaloric LOCALITATE și JUDET, daca se doreste modelarea apartenentei la judete a localitatilor va exista o asociere multi-unu între atributul LOCALITATE și entitatea JUDET.



In acest caz LOCALITATE va fi reclassificata ca entitate desi nu sunt necesare alte informatii in afara numelui localitatii.



Regula 4. Atributele vor fi atasate la entitatile pe care le descriu in mod nemijlocit. De exemplu, UNIVERSITATE va fi atasat ca atribut al entitatii FACULTATE și nu al entitatilor STUDENT sau PROFESOR.

Regula 5. Folosirea identificatorilor compusi va fi evitata. Identificatorul unei entitati este acea submultime de atribute ale acesteia care identifica in mod unic fiecare instanta a sa. In model relational pentru atributele de acest fel se construiesc de regula structuri de cautare rapida (indecsi) care functioneaza cu atat mai lent cu cat complexitatea indecsului creste. Aplicarea acestei reguli se poate face in diverse moduri:

1. Daca identificatorul unei entitati este compus din mai multe atribute care sunt toate identificatori in alte entitati, acea entitate se elimina. Informatia continuta de aceasta va fi modelata sub forma unei asocieri între acele entitati.
2. Daca identificatorul unei entitati este compus din mai multe atribute care nu sunt toate identificatori in alte entitati, exista doua solutii:
 - Entitatea respectiva se elimina și este inlocuita prin alte entitati si asocieri astfel incit pe ansamblu informatia modelata in varianta originara sa fie pastrata.
 - Entitatea respectiva ramine in forma originara, cu dezavantaje insa in privinta vitezei operatiilor.

Se vede ca procedura clasificarii obiectelor in entitati și atribute este iterativa:

- se face o prima impartire conform primei reguli
- parte din atributele astfel obtinute se reclassifica in entitati conform regulilor 2 si 3
- se face o rafinare finala conform regulilor 4 si 5.

b. Identificarea ierarhiilor de generalizare și incluziune.

În cazul în care despre anumite subclase ale unei clase de obiecte există informații specifice, clasa și subclasele (care la pasul anterior au fost catalogate ca entități) sunt interconectate într-o ierarhie de incluziune sau generalizare, după cum este cazul. La acest pas se face și o reatașare a atributelor pentru evitarea redundanței, astfel:

- La entitatea tată vor fi atașate attributele care formează identificatorul și descriptorii care modelează informații specifice întregii clase.
- La entitățile fiu vor fi atașate attributele de identificare (aceleși ca ale tatălui) și attributele care modelează informații specifice doar acelei subclase de obiecte.

Să considerăm o ierarhie care împarte studenții după în căministi și necăministi (fig. 2.7). În acest caz atributul NUME trebuie eliminat de la entitatea NECAMINIST deoarece el este prezent deja la tata.

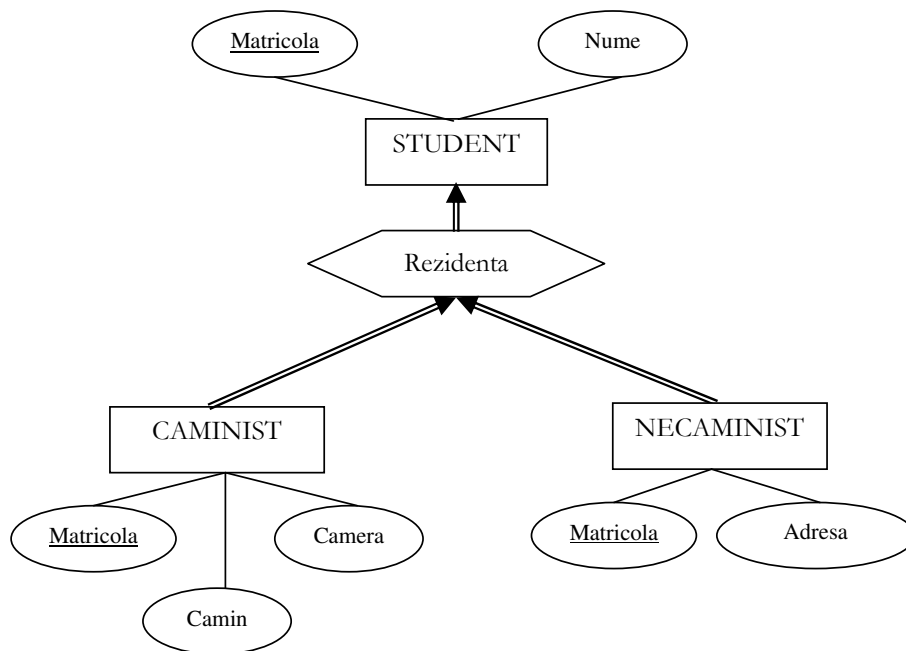


Fig. 2.8. Atributele entităților unei ierarhii

Rezultă următoarele reguli:

1. Tatăl și fiii unei ierarhii au același identificator.
2. Descriptorii care apar și la tată și la fii se elimină de la fii.
3. Descriptorii care apar la toți fiii unei ierarhii de generalizare și nu apar la tată se mută la tată.

c. Identificarea asocierilor

În această etapă se tratează informațiile care nu au fost clasificate ca entități sau attribute și reprezintă interdependențe între clase de obiecte. Ele sunt modelate ca asocieri între entități. Pentru fiecare asociere se specifică gradul, conectivitatea, obligativitatea și dacă este cazul și attributele asocierii precum și rolurile ramurilor sale.

Ca și în cazul clasificării în entități și atribute, există câteva reguli de urmat în operația de definire a asocierilor:

Eliminarea asocierilor redundante. În cazul în care o asociere poate fi dedusă din alte asocieri deja catalogate, aceasta se elimină. De reținut că între două entități pot să existe oricâte asocieri și ele nu sunt considerate redundante atâ timp cât au semnificație diferită. Un caz des întâlnit de redundanță este cel al compunerii (tranzitivității) asocierilor. Prezentăm un exemplu:

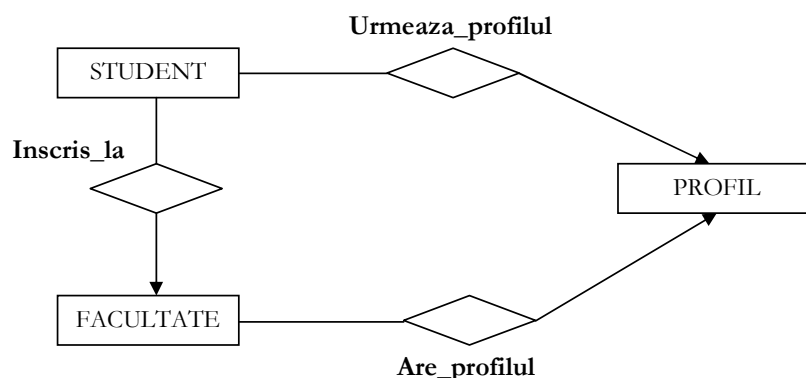


Fig. 2.9. Asocieri redundante

În acest exemplu, asocierea INSCRIS_LA modelează apartenența fiecărui student la o facultate a unui institut de învățământ superior. Fiecare facultate are un profil unic descris de asocierea ARE_PROFILUL. Ambele asocieri sunt multi-unu în sensul $STUDENT \rightarrow FACULTATE \rightarrow PROFIL$. Deoarece asocierile multi-unu (ca și cele unu-unu) sunt din punct de vedere matematic funcții, din compunerea lor putem afla profilul la care este înscris fiecare student. Rezultă că asocierea URMEAZA_PROFILUL care are chiar această semnificație este redundanță și trebuie eliminată.

Asocieri de grad mai mare ca 2. Asocierile ternare (sau de grad mai mare ca trei) se folosesc doar atunci când sunt strict necesare. Este de multe ori posibil ca o aceeași informație să fie modelată ca o asociere ternară sau ca un ansamblu de asocieri binare și unare. În cazul acesta, este de preferat ca să se opteze pentru a doua variantă. Doar când asocierile binare nu pot modela întreaga semnificație dorită se va opta pentru asocieri de grad mai mare ca doi. Această cerință derivă din faptul că la trecerea în modelul relațional asocierile de grad superior devin scheme de relații de sine statatoare, mărinț numărul de tabele din baza de date pe când cele de grad unu și doi (cu excepția celor multi-multi) nu au acest efect.

d. Integrarea vederilor.

În cazul proiectării bazelor de date complexe, activitatea se desfășoară uneori de către mai multe colective simultan, fiecare modelând o porțiune distinctă a bazei de date. Deoarece în final trebuie să se obțină o singură diagramă a bazei de date, după terminarea modelării pe porțiuni diagramele rezultate sunt integrate eliminându-se redundanțele și inconsistențele.