

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and circles of varying sizes, resembling a circuit board or a neural network diagram.

MODELARE SISTEMELOR INFORMATICE

Efectuat de:
Alina Maftor,
Elena Vasiliev

NOTIUNI GENERALE

- ❖ Modelarea este reprezentarea într-un mediu controlat, a proprietăților sau a fenomenelor și proceselor care caracterizează un obiect sau un sistem real. Modelarea presupune abstracție și aducerea în atenție numai a unor aspecte ale realității studiate și anume acele aspecte care prezintă interes pentru modelator.
- ❖ Modelul informațional este o abstracție a unei entități și această abstracție poate fi făcută fie pentru a crea un model general (de referință) care să fie apoi folosit pentru a crea exemple concrete de sisteme informatice (cazul arhitecturilor de referință), fie pentru a crea modelul informatic al unei entități anume, deci un model de transpunere.
- ❖ Metodologia elaborării sistemelor informatice a fost concepută inițial ca un ansamblu de principii și indicații, tehnici și metode grupate și ordonate ca să ducă la realizarea sistemului informatic.

Metodologia de realizare a sistemelor informatice
trebuie să cuprindă:

Etapele/procese de realizare a unui sistem informatic structurate în subetape, activități sarcini precum și conținutul lor

Fluxul realizării acestor etape sau procese, subetape și activități

Modalitatea de derulare a ciclului de viață a sistemului informatic

Tehnicile, procedurile, instrumentele, normele și standardele utilizate

Regulile de formalizare a componentelor sistemului informatic

Strategiile de lucru/metodele de realizare

Modul de abordare a sistemului

Modalitățile de conducere a proiectului (planificare, programare, urmărire)

Modul de utilizare a resurselor financiare, umane și materiale

PRINCIPALELE SCOPURI ALE MODELARII SISTEMELOR INFORMATICE

- vizualizarea, ca mijloc de usurare a comunicarii si intelegerii;
- specificarea, prin construirea de modele precise si complete;
- documentarea cerintelor, a solutiilor de proiectare si a modului de realizare.
- fiecare tip de model informtional permite evidentierea unei anumite laturi, dimensiuni a sistemului la un anumit nivel de abstractizare.
- numai prin utilizarea combinata a mai multor tipuri de modele este posibila realizarea unei analize complexe (aceasta permite obtinerea unor informatii generale, globale despre sistemul analizat)

METODE DE ANALIZA SI DE PROIECTARE

Proiectarea unui sistem are loc pe baza unei specificatii a cerintelor, deci este o continuare a procesului de analiza. Metodele de proiectare sunt strans legate de cele folosite in analiza, modelele de proiectare fiind adesea construite plecand de modelele de analiza.

Exista doua strategii de structurare a unui sistem informatic, pe baza carora metodele de analiza si proiectare sunt clasificate in metode functionale si metode orientate obiect.

METODE FUNCTIONALE

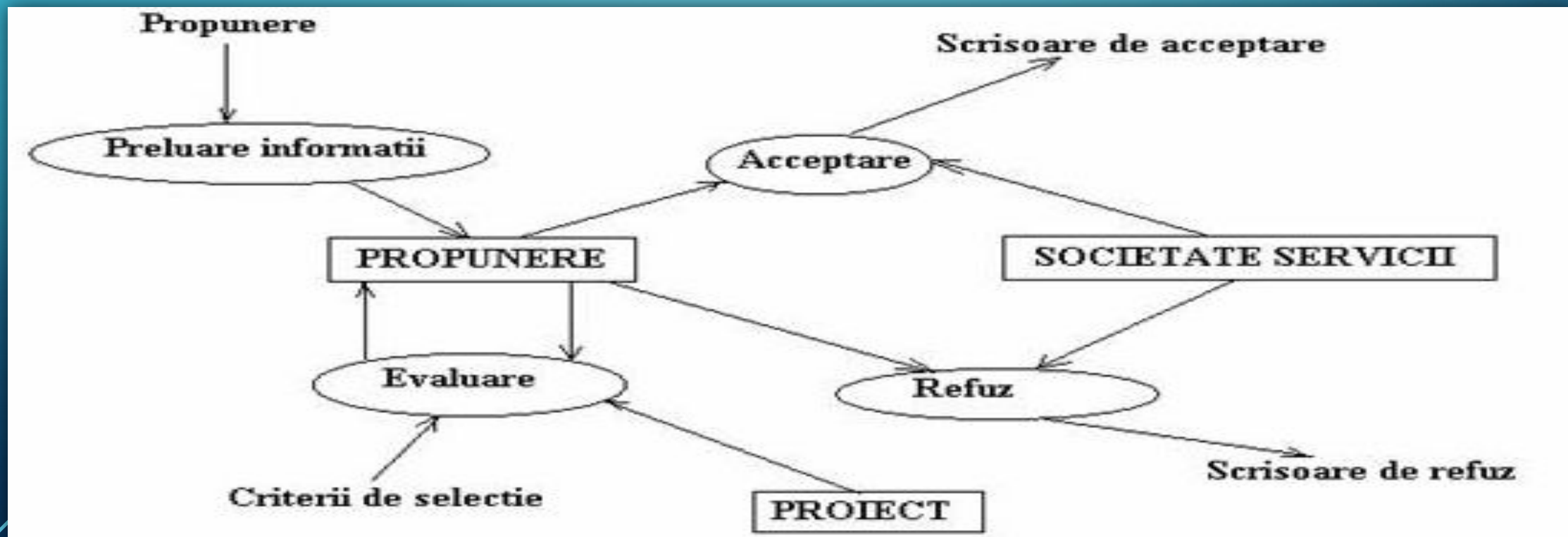
Aceste metode isi au originile in dezvoltarea limbajelor procedurale. Mai orientate catre prelucrari decat spre date, ele propun o abordare ierarhica descendenta, bazata pe descompunerea prelucrarilor care trebuie sa fie efectuate de un sistem.

Metodele functionale cuprind:

1. Diagramele de flux de date
2. Diagramele de stari-tranzitii
3. Diagramele de enittate/legatura
4. Diagramele de structura
5. Dictionarul de date

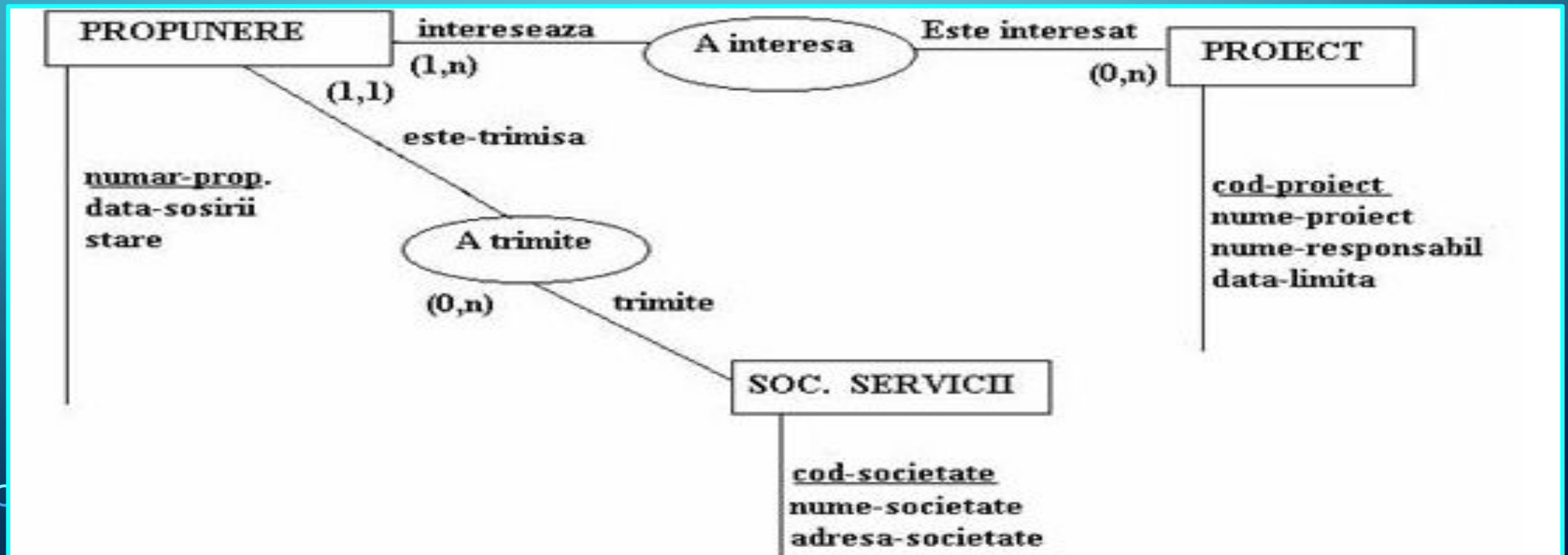
DIAGramele de flux de date

Se folosesc pentru a modela transformările datelor pe măsura ce acestea tranzitează sistemul. O diagramă de flux de date este alcătuită din blocuri de prelucrare și blocuri 'rezervoare de date'. Fluxul datelor este reprezentat prin săgeți. Figura următoare ilustrează tratarea propunerilor făcute unei întreprinderi de către societăți de servicii. Prelucrările sunt reprezentate prin elipse iar rezervoarele prin dreptunghiuri.



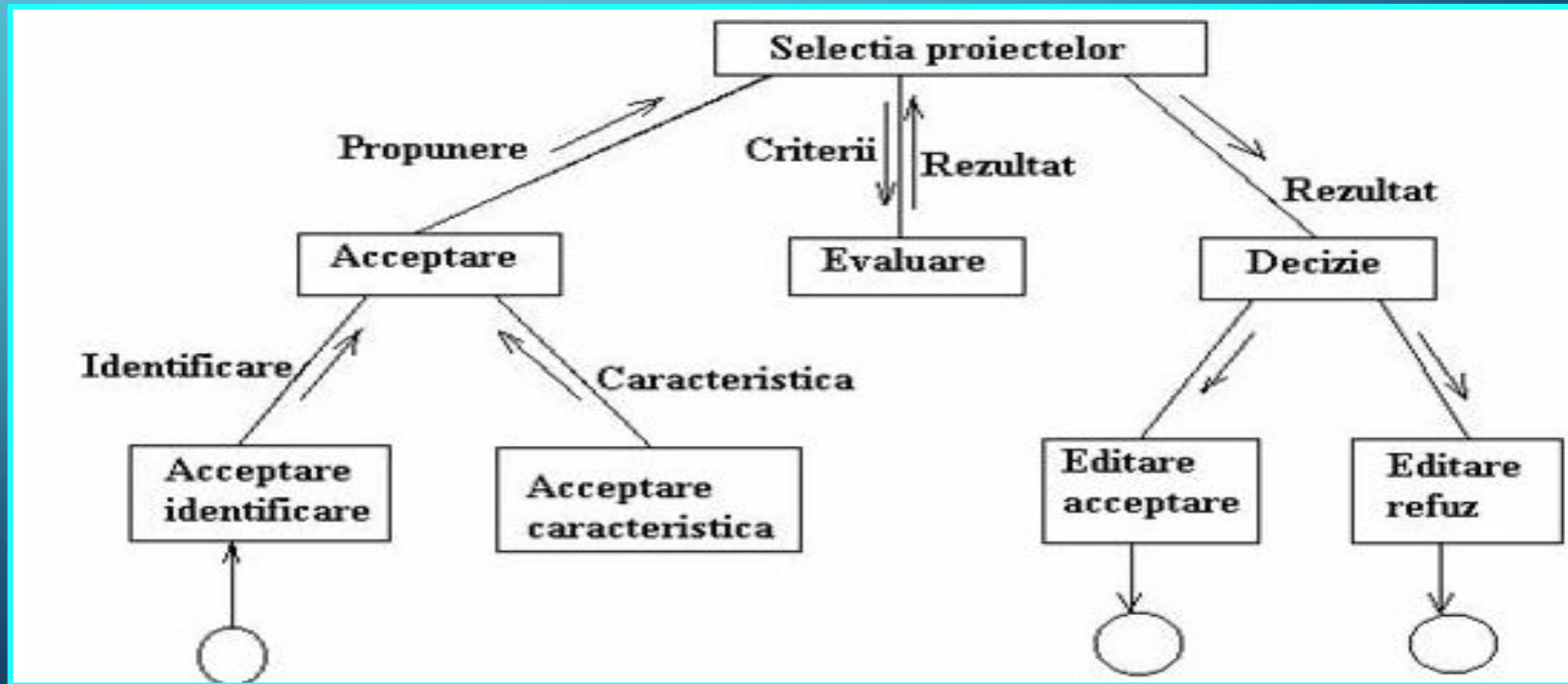
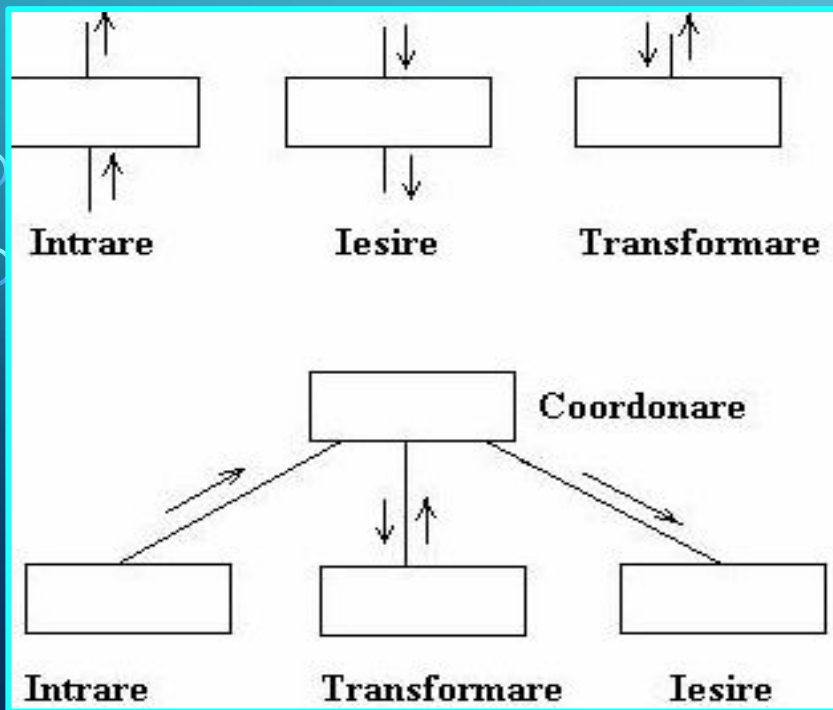
DIAGramele ENTITATE/LEGATURA

Reflecta relatiile dintre rezervoarele de date. Fiecare 'entitate' corespunde unui rezervor de date dintr-o diagrama de flux de date. Relatiile dintre entitati sunt numite 'asocieri'. Entitatile si asocierile pot fi caracterizate prin attribute. Figura urmatoare pune in evidenta trei entitati: proiect, propunere si societate servicii, reprezentate prin dreptunghiuri, fiecareia fiindu-i asociate attribute.



DIAGramele de Structura

Modeleaza arhitectura unui sistem ca o ierarhie de module (functii) si o prezinta sub forma unei structuri arborescente. Modulele sunt reprezentate prin noduri iar conexiunile intre module prin arce. Un arc conecteaza un modul, situat pe nivelul n , de modulul care-l apeleaza, situat pe nivelul $(n-1)$. Parametrii de intrare si de iesire sunt indicati de-a lungul conexiunilor, prin texte si sageti.



Diagramele de stari - tranzactii

Se folosesc pentru a modela comportamentul dependent de timp al sistemului. Ele sunt similare celor din notatia UML.

Dictionarul de date

Contine detalii care nu sunt cuprinse in diagramele prin care se modeleaza sistemul. El descrie fluxuri de date, rezervoare de date, entitati, module si semnificatia numelor atribuite.

Dictionarul de date este un mijloc de management al numelor. Dictionarul de date permite verificarea unicitatii numelor. Crearea, actualizarea si interogarea dictionarului de date sunt necesare pe intreaga durata de viata a unui sistem.

METODE ORIENTATE OBIECT

Aceste metode se bazeaza pe conceptele de clasa, obiect, abstractie, specializare si comunicare prin mesaje.

In majoritatea metodelor orientate obiect, studiul unei probleme este realizat urmarind trei aspecte:

- aspectul *static* sau descriptiv, care reda obiectele si legaturile dintre ele;
- aspectul *dinamic*, care precizeaza comportamentul obiectelor, diferitele stari prin care ele trec si evenimentele care declanseaza trecerea dintr-o stare in alta.
- aspectul *functional*, care precizeaza functiile realizate de obiecte prin intermediul metodelor.

Metodele orientate obiect :

Metoda Grady Booch,

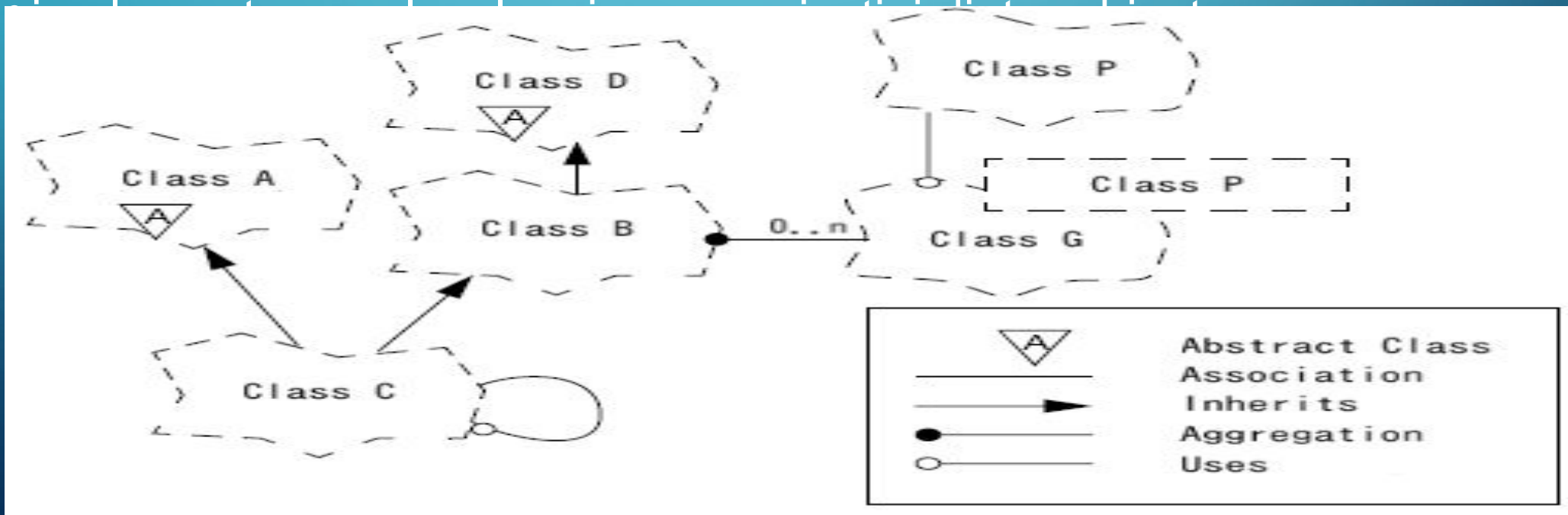
Metoda Jackson,

OMT - Object Modeling Technique

Metoda Grady Booch

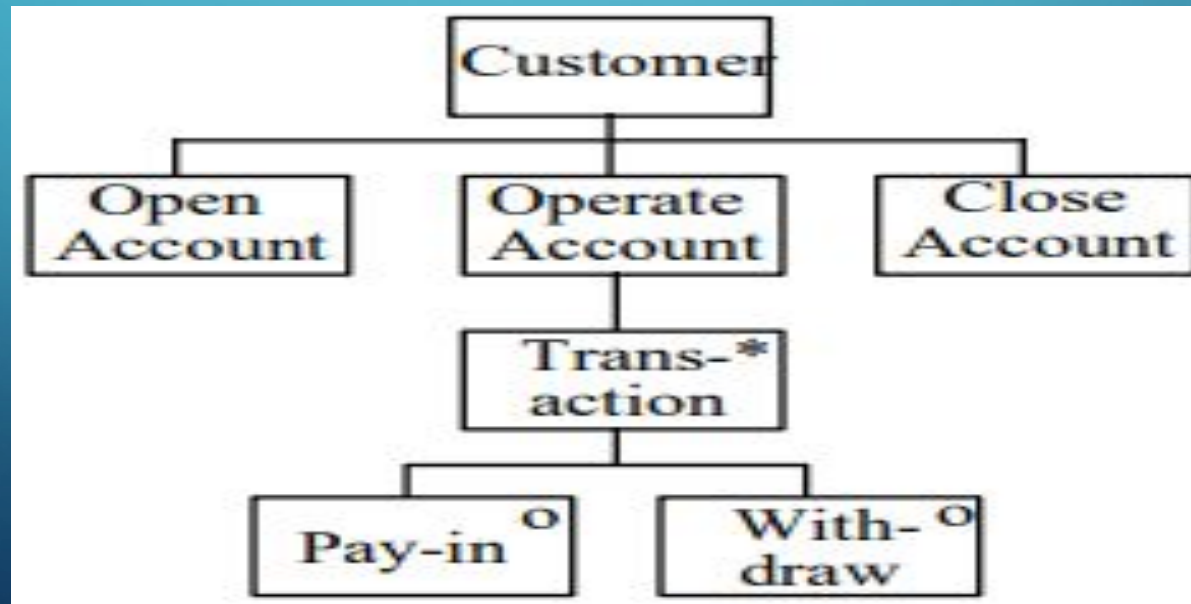
Metoda Booch propune patru etape:

- identificarea obiectelor si a claselor la un nivel de abstractie dat;
- precizarea semanticii claselor precum si a interfetei fiecarei clase;
- identificarea relatiilor dintre clase, distingand pe de o parte aspectele statice iar pe de alta parte aspectele dinamice;



METODA JACKSON (JACKSON STRUCTURED DEVELOPMENT)

Metoda JSD este conceputa in special pentru aplicatii in care este important elementul timp. Un model JSD descrie lumea reala in termeni de entitati, de actiuni si de ordonare a actiunilor. Dezvoltarea unui program consta din sase etape secventiale: etapa actiune a entitatilor, etapa de structurare a entitatilor, etapa de modelare initiala, etapa functie, etapa de analiza a aspectelor temporale ale sistemului si etapa de implementare.



OMT (OBJECT MODELING TECHNIQUE)

OMT propune modelarea unui sistem pe baza a trei puncte de vedere corelate dar distincte, fiecare evidențiind aspecte importante ale sistemului:

- aspectele statice, care sunt reprezentate în modelul obiect;
- aspectele temporale, comportamentale și de 'control' ale sistemului, redate în modelul dinamic;
- aspectele functionale și de transformare de date, reprezentate în modelul functional.

Cele trei modele decupează sistemul în vederi ortogonale care pot fi reprezentate cu o notatie uniforma. Interconexiunile între modele sunt limitate și explicite.

METODELE FUNCTIONALE vs METODELE ORIENTATE OBIECT

COMUN: utilizeaza constructii de modelare similare si suporta cele trei vederi ortogonale ale unui sistem.

DIFERENTE:

In abordarea functionala, modelul functional domina, urmeaza apoi ca importanta modelul dinamic, iar modelul obiect este cel mai putin important. Metodele obiect considera modelul obiect ca cel mai important, apoi modelul dinamic si la sfarsit modelul functional.

Metodele functionale organizeaza un sistem in jurul procedurilor. Invers, o tehnica de modelare obiect (cum ar fi OMT) organizeaza un sistem in jurul obiectelor lumii reale sau al obiectelor conceptuale care exista in viziunea utilizatorului din lumea reala.

UML - UNIFIED MODELLING LANGUAGE

UML (The Unified Modeling Language for Object-Oriented Development) este un limbaj de modelare obiect. UML este independent de procesul de dezvoltare folosit.

UML este un limbaj pentru:

- Vizualizare si comunicare (usureaza comunicarea intre diversele categorii de persoane implicate in procesul de dezvoltare a unui sistem informatic)
- Specificare si construire (permite specificarea sistemelor prin modele precise, ne-ambigue si complete la toate nivelele de detaliu: analiza, proiectare si implementare)
- Documentare (specificarea cerintelor utilizatorilor si a cerintelor software, proiectarea arhitecturala si de detaliu)

ELEMENTELE DE MODELARE DEFINITE IN UML POT FI IMPARTITE IN 3 CATEGORII:

Modelare

comportamentala:

- Cazuri de utilizare
- Diagrame de cazuri de utilizare
- Diagrame de interactiune
- Diagrame de stari
- Diagrame de activitati

Modelare structurala

- Clase
- Diagrame de clase
- Diagrame de obiecte
- Interfete
- Pachete

Modelare arhitecturala

- Componente
- Diagrame de componente
- Diagrame de distributie

CONCLUZII

Unul dintre aspectele importante in intelegerea si definirea cerintelor unui sistem este acela al interactiunii dintre sistem si utilizatori sau alte componente externe.

Modelele construite pe parcursul dezvoltarii unui sistem reflecta o anumita vedere asupra sistemului si corespunde unui nivel de detaliu.

In etapa de analiza se construiesc modele care exprima cerintele impuse sistemului.

In etapa de proiectare se construiesc modele care redau arhitectura sistemului, alocarea cerintelor pe subsisteme, distributia proceselor in sistem, sincronizarea lor, starile si tranzitiile intre stari.

Alte modele descriu realizarea fizica a sistemului, echipamentele din componenta sa si repartitia componentelor program.