

2010 - 2011



Inginerie Software pentru Comunicatii (ISC / RST)

Titular curs: **Eduard-Cristian Popovici**

Suport curs: <http://discipline.elcom.pub.ro/isc/>

Moodle: <http://electronica.curs.ncit.pub.ro/>



Continut curs

1. Introducere in ingineria software

- 1.1. Necesitatea unei abordari sistematice a dezvoltarii software
- 1.2. Abordari si metodologii larg utilizate in ingineria software

2. Introducere in limbajul UML

- 2.1. Definirea, rolul si istoricul limbajului de modelare unificat (UML)
- 2.2. Tipuri de diagrame UML. Organizarea ierarhica a diagramelor

3. Diagrame UML statice

- 3.1. Diagrame UML de clase
- 3.2. Diagrame UML de obiecte
- 3.3. Diagrame UML de pachete
- 3.4. Diagrame UML de componente
- 3.5. Diagrame UML de structuri compozite
- 3.6. Diagrame UML de *deployment* (amplasare)

Continut curs

4. Diagrame UML dinamice

- 4.1. Diagramele UML de caz de utilizare
- 4.2. Diagrame UML de comunicatie si de robustete
- 4.3. Diagrame UML de secventa si de sumar al interactiunilor
- 4.4. Diagrame UML de masini de stari
- 4.5. Diagrame UML de activitati
- 4.6. Diagrame UML de timp

5. Introducere in procesul de dezvoltare Rational unificat (RUP)

- 5.1. Organizarea iterativa a proiectelor
- 5.2. Fazele si activitatile procesului RUP

6. Introducere in managementul si organizarea proceselor de dezvoltare

7. Elemente de reutilizabilitate a software-ului. Pattern-uri de proiectare

A picture is worth more than 1024 lines of code

4. Diagrame UML dinamice

4.1. Diagramele UML de caz de utilizare

4.1. Diagrame UML de caz de utilizare

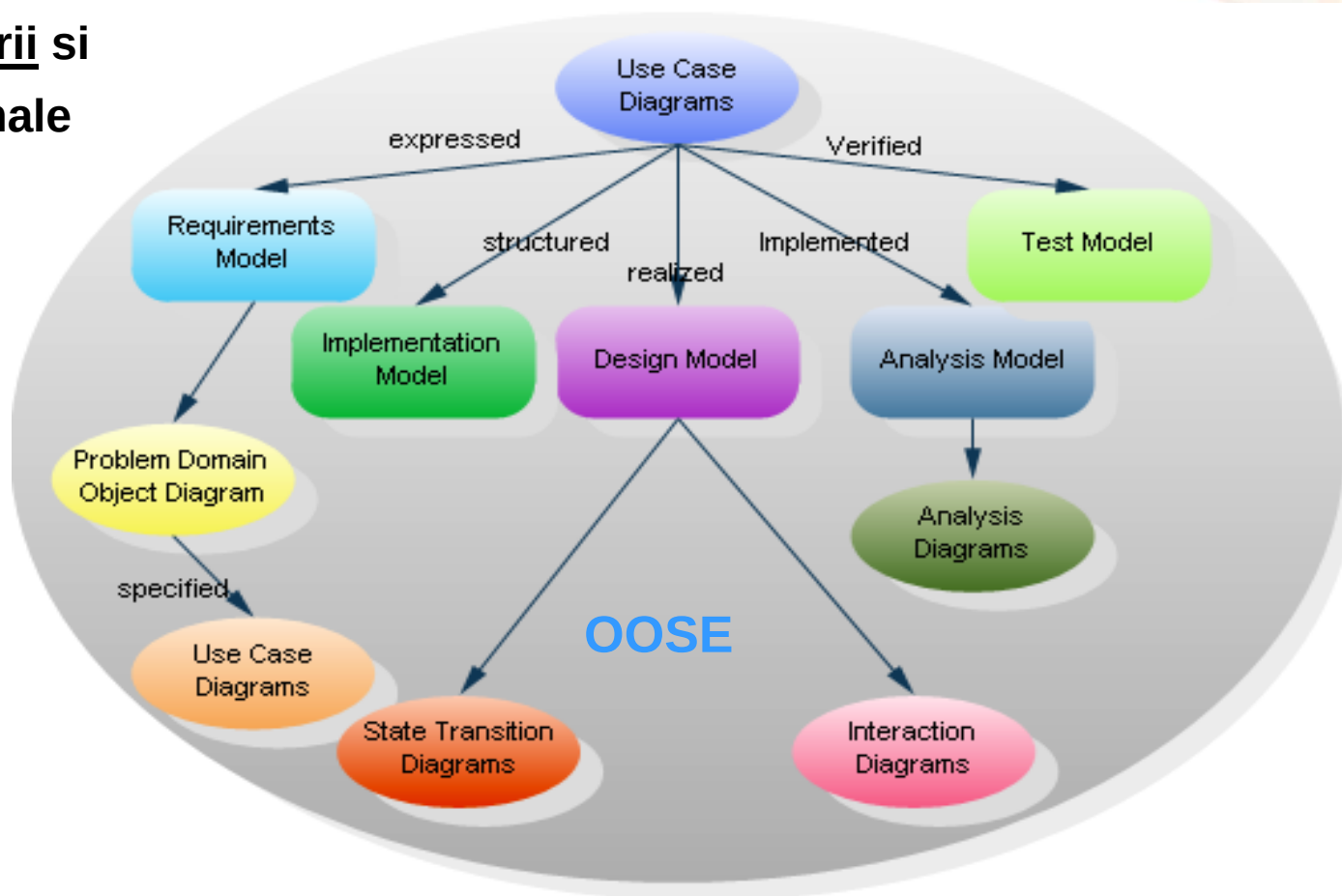
Diagramele UC: motivatie

4.1. Diagrame UML de caz de utilizare

Diagramele cazurilor de utilizare (UC – *use cases*)

➤ au fost **formalizate de Ivar Jacobson** in **1992** in metodologia lui numita **OOSE** (*Object-Oriented Software Engineering*)

➤ cu scopul capturarii si reprezentarii formale a cerintelor utilizatorului



4.1. Diagrame UML de caz de utilizare

Diagramele UC - motivatie

Caiele de sarcini sunt adesea “**lungi litanii de paragrafe**” care contin expresii de genul:

- “sistemul ***va face eventual...***”
- “***este absolut necesar ca...***”
- “***ar fi interesant ca...***”

Frecvent:

- caietul de sarcini initial e **vag** si in continua **evolutie**
- cerintele se **contrazic**, sunt **uite** sau **imprecise**
- cerintele sunt **exprimate** in mod **nestructurat**, **incoerent**
- analiza sistemului porneste de la **premize gresite**

Atunci **cand cerintele evolueaza**

- devine **dificil** de **apreciat impactul** si **costul unei modificari**

4.1. Diagrame UML de caz de utilizare

Diagramele UC – definitii si roluri

- descriu **comportamentul** extern al unui sistem **din punct de vedere al utilizatorului**:
 - sub forma **actiunilor** utilizatorului asupra sistemului
 - si a **reactiilor** sistemului la aceste actiuni
- permit definirea
 - **limitelor** sistemului (interfata grafica, etc.) si a
 - **relatiilor** intre **sistem** si **mediu** (lumea exterioara)
- sunt **modalitati specifice de utilizare** a sistemului
- sunt imagini ale **functionalitatilor** sistemului

4.1. Diagrame UML de caz de utilizare

Diagramele UC - motivatie

Ansamblul cerintelor – problema de tip “*cum sa mananci un elefant?*”



4.1. Diagrame UML de caz de utilizare

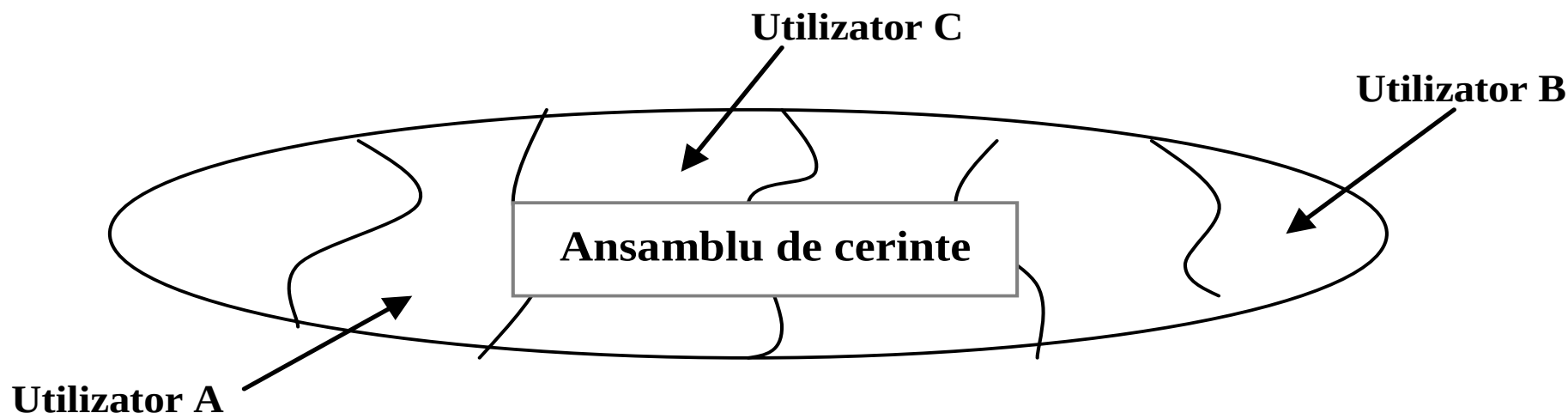
Diagramele UC: concepte

4.1. Diagrame UML de caz de utilizare

Diagramele UC – separarea ansamblului de cerinte

Cazurile de utilizare (UC)

- pleaca de la ideea ca **un sistem** e in primul rind **construit pentru utilizator**
- **structurarea UC** se efectueaza in functie de interactiunile unei **categorii de utilizatori** (denumita **actor**)
- **partitionarea ansamblului** de cerinte **pe categorii de utilizatori** reduce **complexitatea** determinarii cerintelor



4.1. Diagrame UML de caz de utilizare

Diagramele UC – rolul descrierilor textuale

Descrierile textuale ale UC sunt **bazate pe limbajul natural**

- **terminologia** fiind cea folosita de utilizatori in viata lor de toate zilele
- UC fiind astfel **accesibile** fara o **formatie particulara** a utilizatorilor

Utilizatorii

- isi pot astfel **exprima** usor **asteptarile si cerintele** (care sunt **capturate astfel structurat**)
- **comunicand** usor **cu expertii domeniului si informaticienii**

Informaticienii la randul lor

- sunt **readusi** in mod constant **la cerintele reale ale utilizatorilor**
- **evitand devierea** catre constructii tehnice seducatoare dar inutile
- astfel incat UC **focalizeaza efortul dezvoltarii pe adevaratele cerinte**

4.1. Diagrame UML de caz de utilizare

Diagramele UC – elementele componente

Modelul cazurilor de utilizare cuprinde

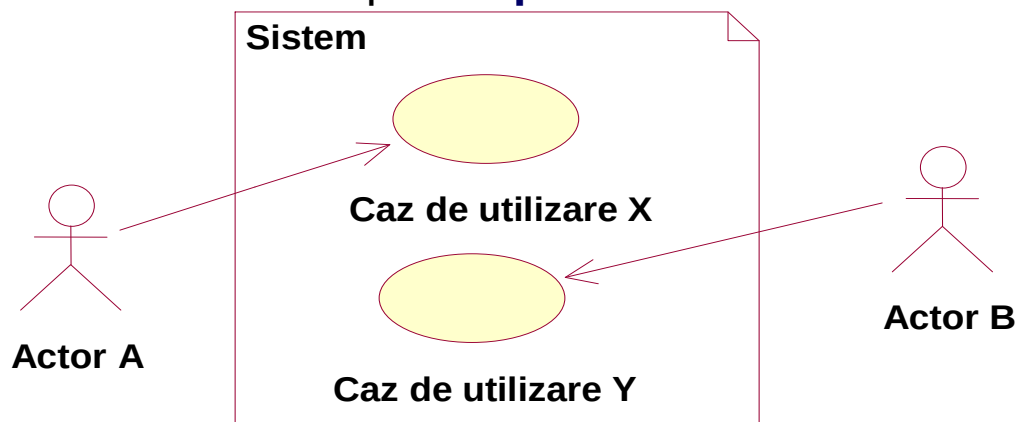
- **actorii** externi sistemului (noduri ale “grafului”)
- sistemul si **cazurile de utilizare** ale sistemului (noduri ale “grafului”)
- si **relatiile** dintre cazuri si actori (arce ale “grafului”)

Actorii

- se reprezinta in UML sub forma unor **mici personaje** care **declanseaza** cazuri de utilizare

Cazurile de utilizare

- sunt reprezentate in UML prin **elipse continute de sistem**



4.1. Diagrame UML de caz de utilizare

**Diagramele UC: identificarea si
descrierea actorilor**

4.1. Diagrame UML de caz de utilizare

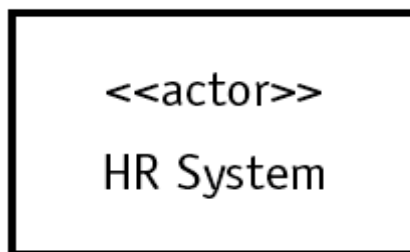
Diagramele UC - actorii

Actorii se determina observand

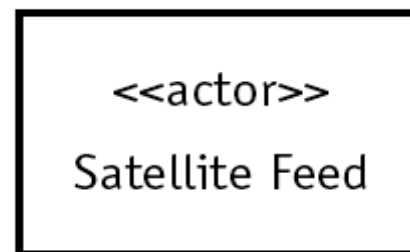
- **utilizatorii directi** ai sistemului
- **responsabilii** de exploatarea sau intretinerea lui
- **celelalte sisteme** care interactioneaza cu sistemul in chestiune



person example



system example



device example

Aceeasi persoana fizica poate juca **rolul mai multor actori**

Mai multe persoane pot juca **acelasi rol** fiind reprezentate de **acelasi actor**

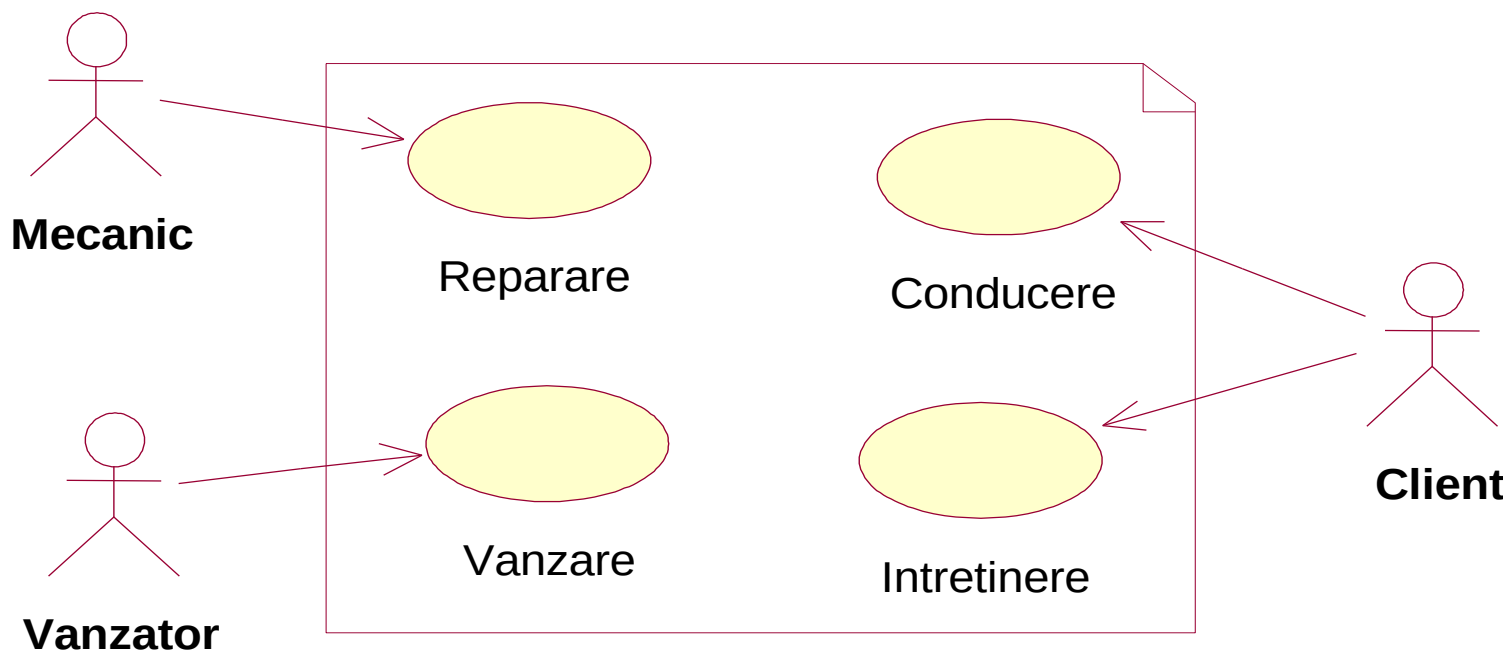
Numele actorului descrie **rolul** jucat de catre actor

4.1. Diagrame UML de caz de utilizare

Diagramele UC - actorii

In exemplul urmator, **aceeasi persoana**, angajata la un magazin auto cu service:

- joaca in majoritatea timpului sau de lucru rolul **mecanicului**
- poate ocazional sa joace si rolul **vanzatorului**
- in afara programului de lucru rolul **clientului**, conducand si intretinand masina personala



4.1. Diagrame UML de caz de utilizare

Diagramele UC – actorii si limitele sistemului

Determinarea actorilor permite **precizarea limitelor** sistemului in mod **progresiv**:

- vagi la inceput
- **se precizeaza pe masura ce se elaboreaza** cazurile de utilizare

Activitatea de delimitare a sistemului este extrem de importanta deoarece

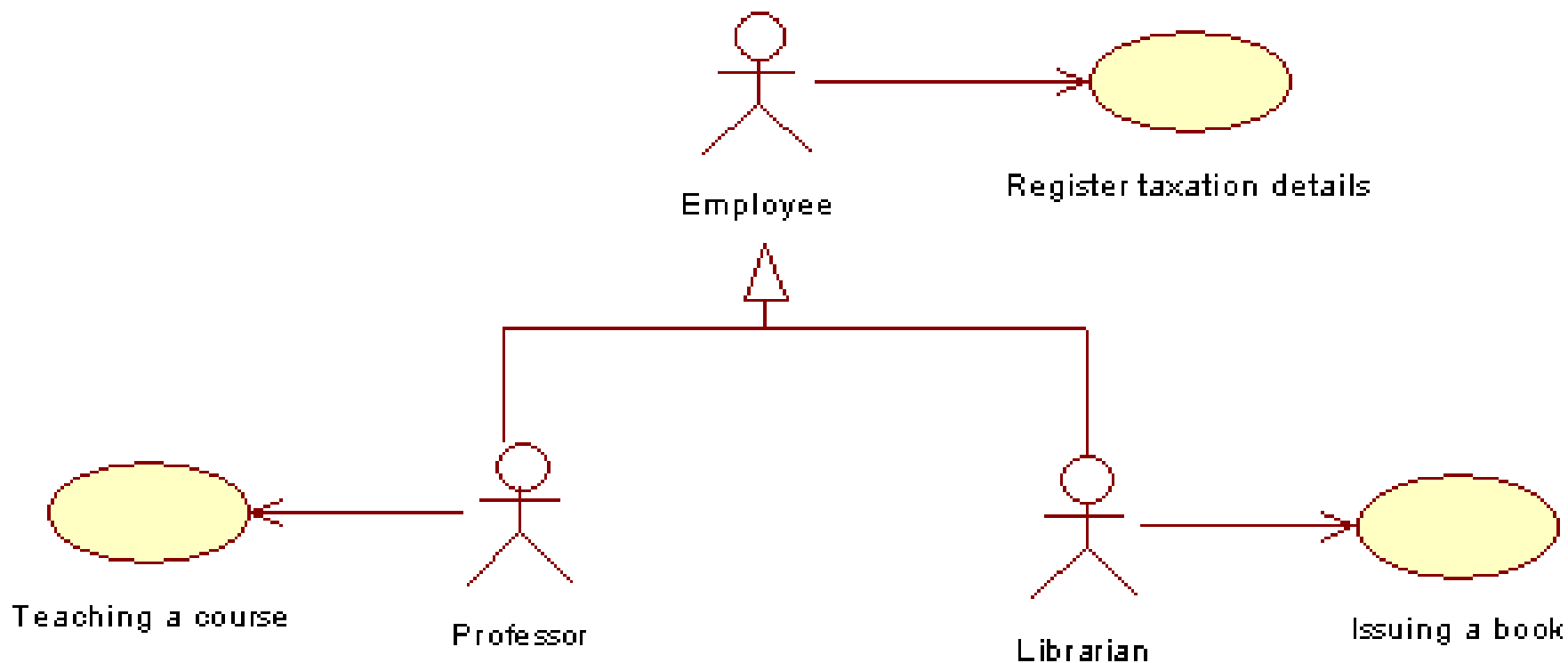
- serveste ca o **baza contractuala**, stabilind
 - ceea ce trebuie sa fie **facut**
 - ceea ce **face parte** din sistemul de dezvoltat si
 - ceea ce **nu** face parte din el

4.1. Diagrame UML de caz de utilizare

Diagramele UC – actorii si limitele sistemului

Cand **exista multi actori** intr-un sistem

- este **recomandat sa fie grupati pe categorii**, pentru a usura navigarea in modelul cazurilor de utilizare
- urmand ca in diagrama sa poata fi reprezentati prin **generalizarea actorilor**



4.1. Diagrame UML de caz de utilizare

Diagramele UC - actorii

Odata identificati, **actorii** trebuie sa fie

- **descriși** intr-un mod **clar si concis**, in **3-4 linii** de text **cel mult**

Exemplu de lista a actorilor

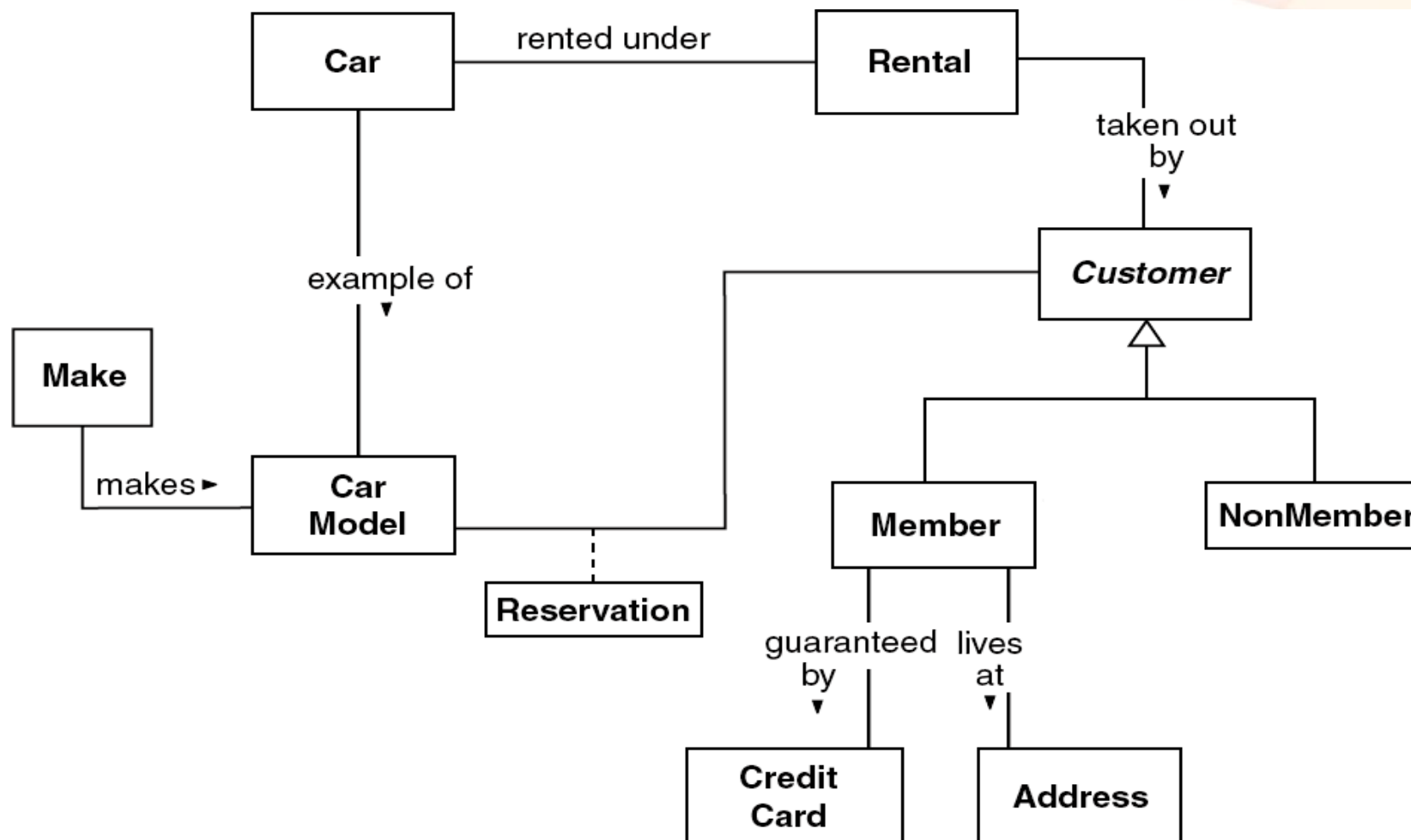
Actor List

- **Customer:** *A person using a Web browser to access iCoot*
- **Member:** *A Customer who has presented their name, address and CreditCard details at one of our Stores; each Member is given an Internet password to accompany their membership number (Specializes Customer)*
- **NonMember:** *A Customer who is not a Member (Specializes Customer)*
- **Assistant:** *An employee at a Store who contacts Members to tell them about the progress of their Reservations*

4.1. Diagrame UML de caz de utilizare

Diagramele UC – glosarul de termeni

Modelul obiectual al domeniului folosit in descrierile anterioare



4.1. Diagrame UML de caz de utilizare

**Diagramele UC: identificarea si
reprezentarea UC**

4.1. Diagrame UML de caz de utilizare

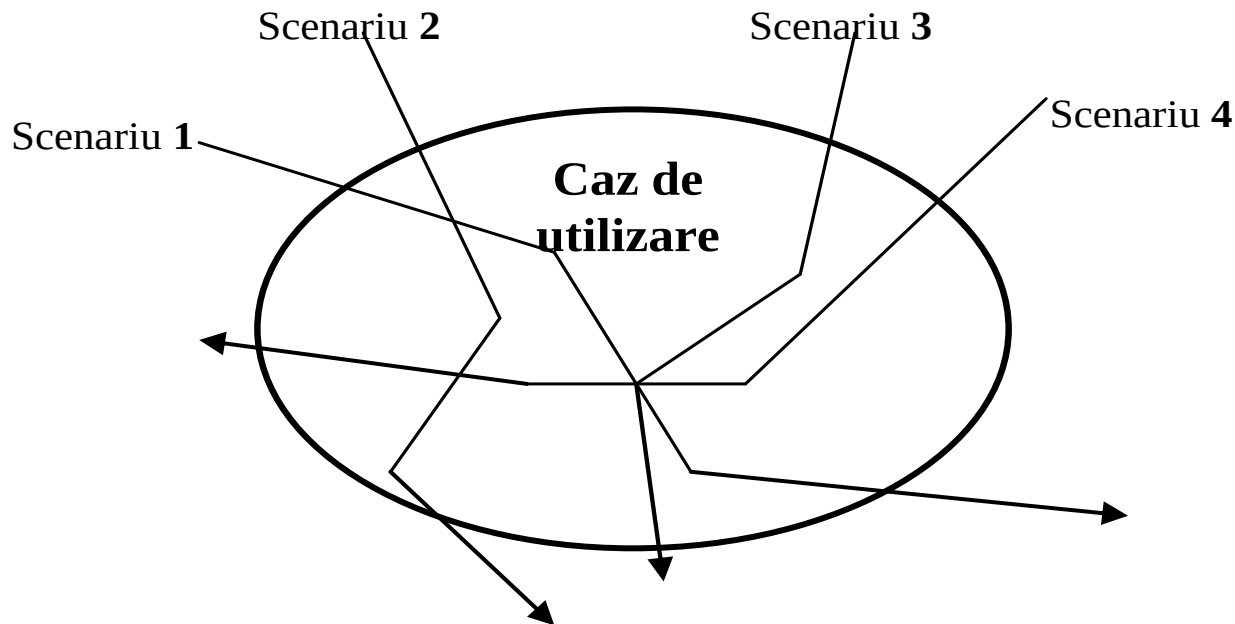
Diagramele UC – cazurile de utilizare

Cazurile de utilizare

- se determina identificand **pentru fiecare actor**
- **scenariile** (secventele de interactiuni) din perspectiva utilizatorului

Un caz de utilizare

- **regrupeaza** dupa criteriul functional o **familie de scenarii** de utilizare



4.1. Diagrame UML de caz de utilizare

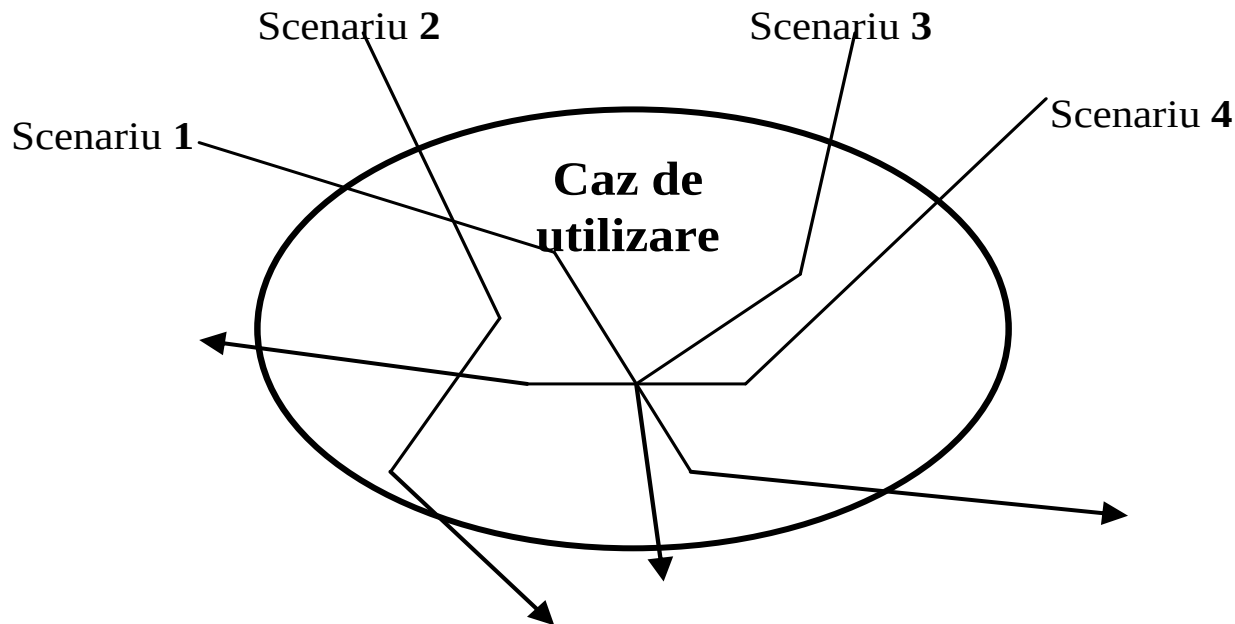
Diagramele UC – cazurile de utilizare

Un scenariu este

- o **cale particulara** prin descrierea abstracta si generala oferita de UC

Scenariile traverseaza cazul de utilizare

- urmarind **calea principala** precum si **alternativele** sau **situatiile exceptionale**

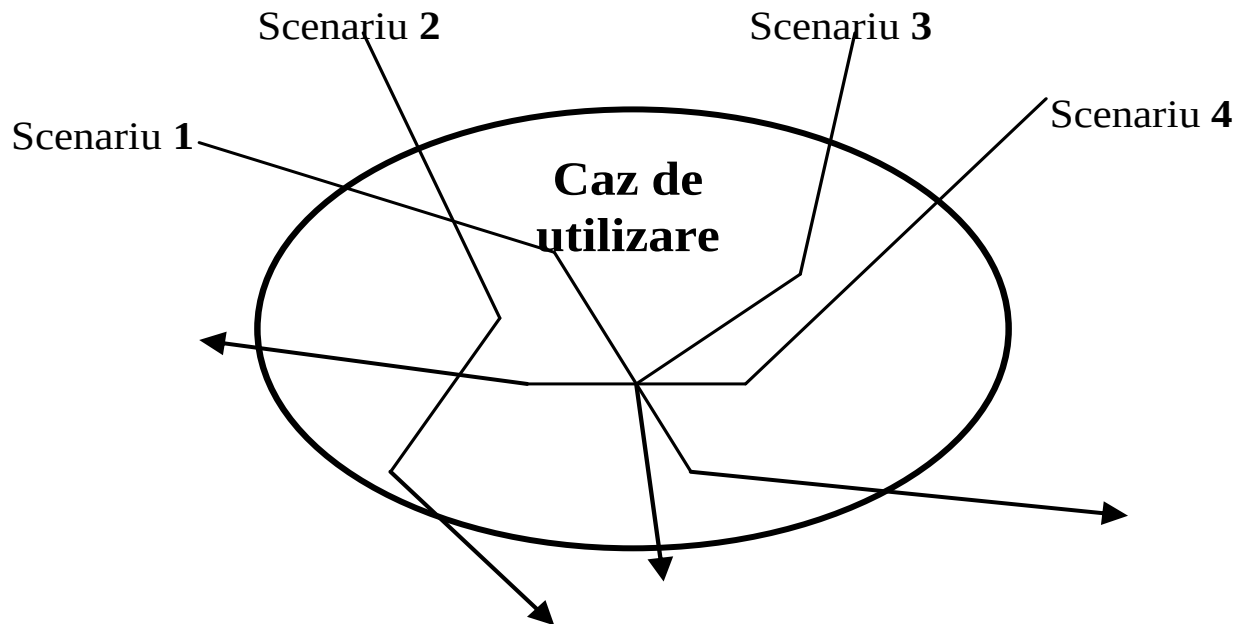


4.1. Diagrame UML de caz de utilizare

Diagramele UC – cazurile de utilizare

Cazurile de utilizare

- sunt **abstractii ale dialogului** între actori și sistem, care
- descriu interacțiuni potențiale **fără a intra în detalii privind scenariul**
- trebuie să fie văzute **ca niste clase** ale caror **instante sunt scenariile**

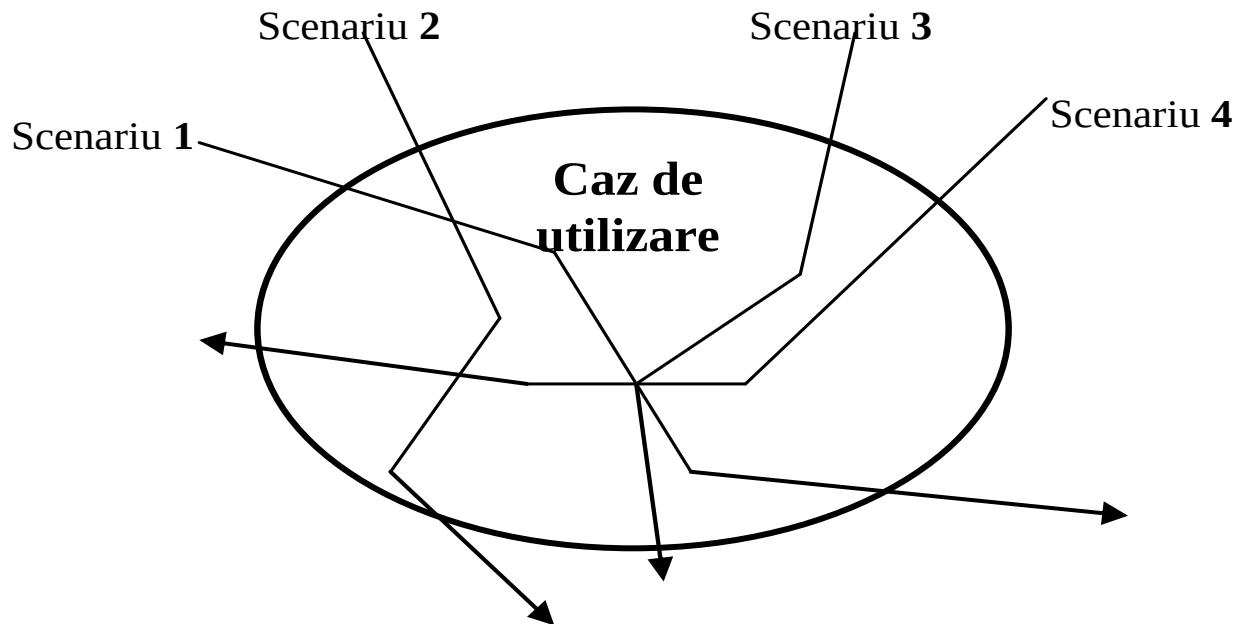


4.1. Diagrame UML de caz de utilizare

Diagramele UC – cazurile de utilizare

Fiecare interactiune a unui actor cu sistemul

- conduce la **instantierea unui scenariu** de catre cazul de utilizare
- **sub forma fluxului de mesaje schimbate** de catre obiecte (reprezentat in faza concretizarii diagramei UC sub forma unei diagrame de interactiune - MSC sau de comunicatie)



4.1. Diagrame UML de caz de utilizare

Diagramele UC

Exemplu de
diagrama UC
simpla



4.1. Diagrame UML de caz de utilizare

**Diagramele UC: reprezentarea relatiilor
intre UC**

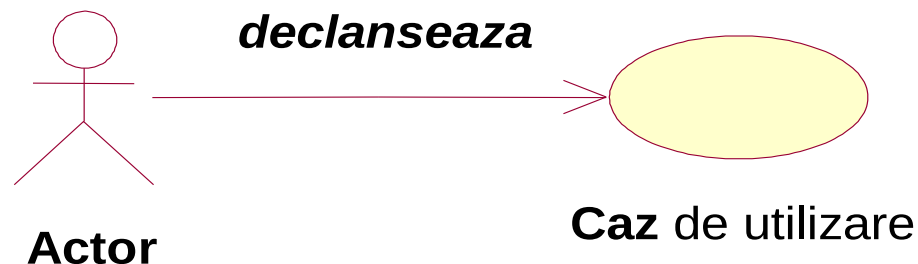
4.1. Diagrame UML de caz de utilizare

Relatiile intre cazurile de utilizare (si actori)

UML defineste 4 tipuri de relatii intre actori si cazurile de utilizare

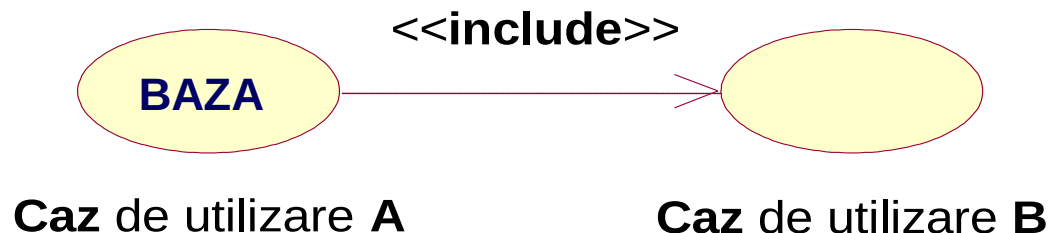
(1) comunicatie (asociere) intre un actor si un caz de utilizare:

- participarea actorului e figurata printr-o sageata de la actor la cazul de utilizare (sensul indicind initierea interactiunii)



(2) includere intre 2 cazuri de utilizare:

- semnifica faptul ca o instanta a cazului de utilizare sursa (A) cuprinde **intotdeauna** comportamentul descris de cazul de utilizare destinatie (B)

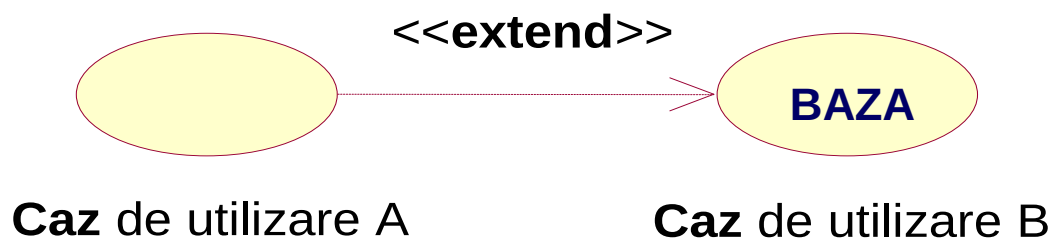


4.1. Diagrame UML de caz de utilizare

Relatiile intre cazurile de utilizare (si actori)

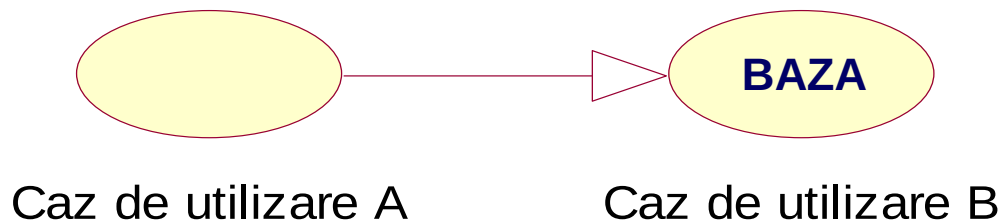
(3) extindere intre 2 cazuri de utilizare:

- semnifica faptul ca o instanta a cazului de utilizare sursa (A) **poate** extinde (conditionat) in mod explicit comportamentul cazului de utilizare destinatie (B) :



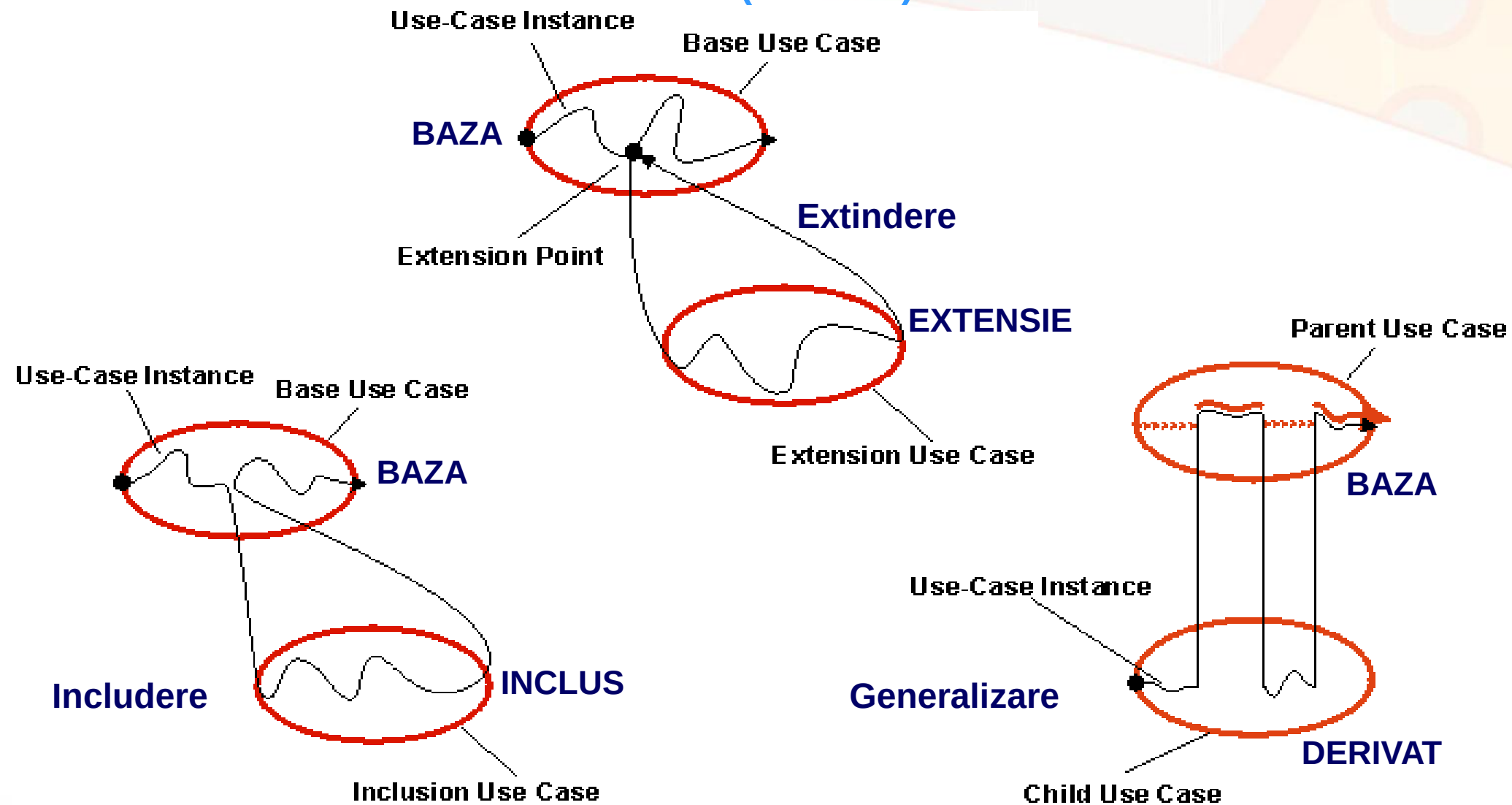
(4) generalizare intre 2 cazuri de utilizare

- semnifica faptul ca o instanta a cazului de utilizare sursa (A) **extinde in mod neprecizat** (rescrie, completeaza) comportamentul cazului de utilizare destinatie (B)



4.1. Diagrame UML de caz de utilizare

Relatiile intre cazurile de utilizare (si actori)

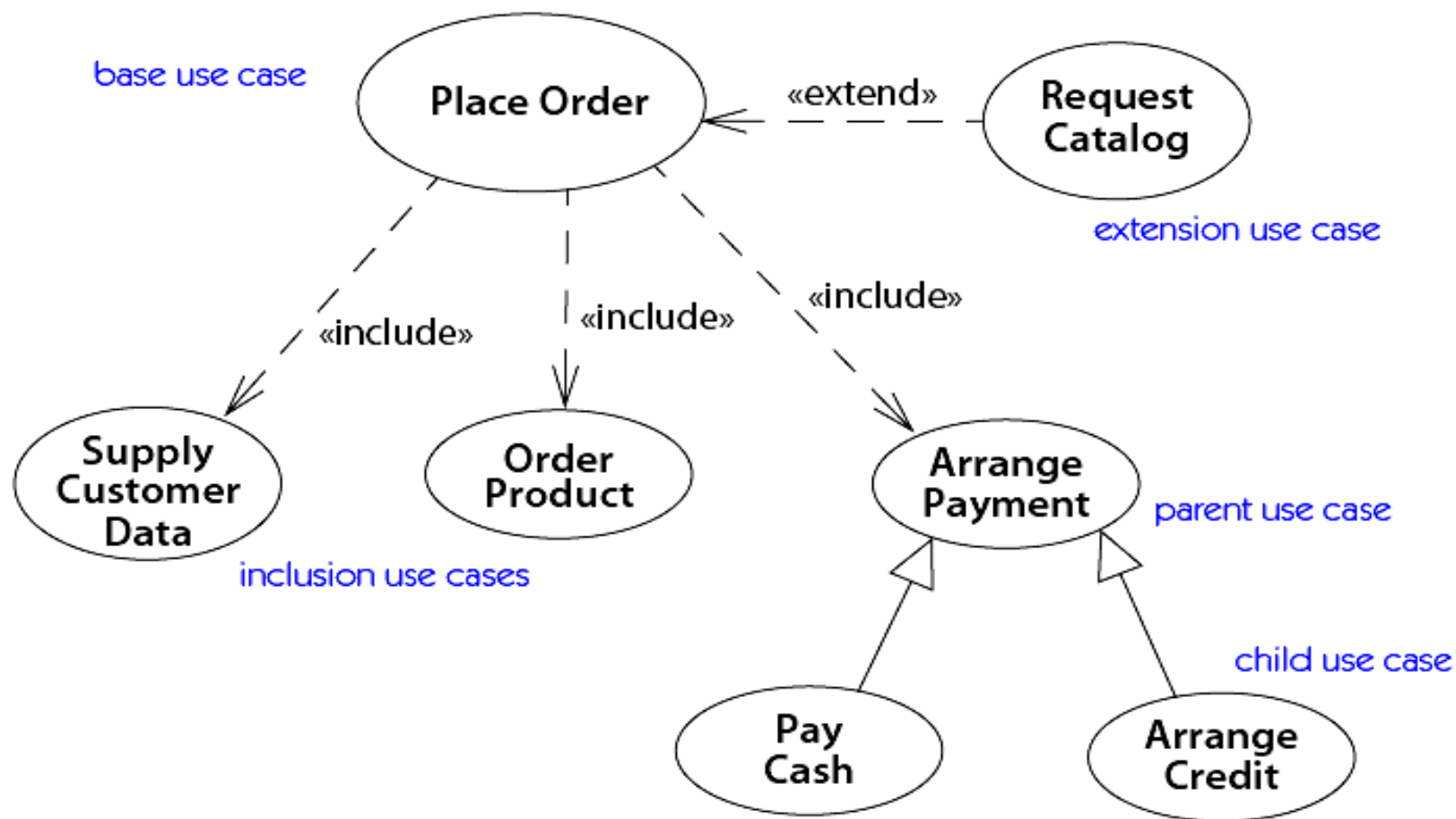


4.1. Diagrame UML de caz de utilizare

Relatiile intre cazurile de utilizare (si actori)

Exemplu de relatii intre UC

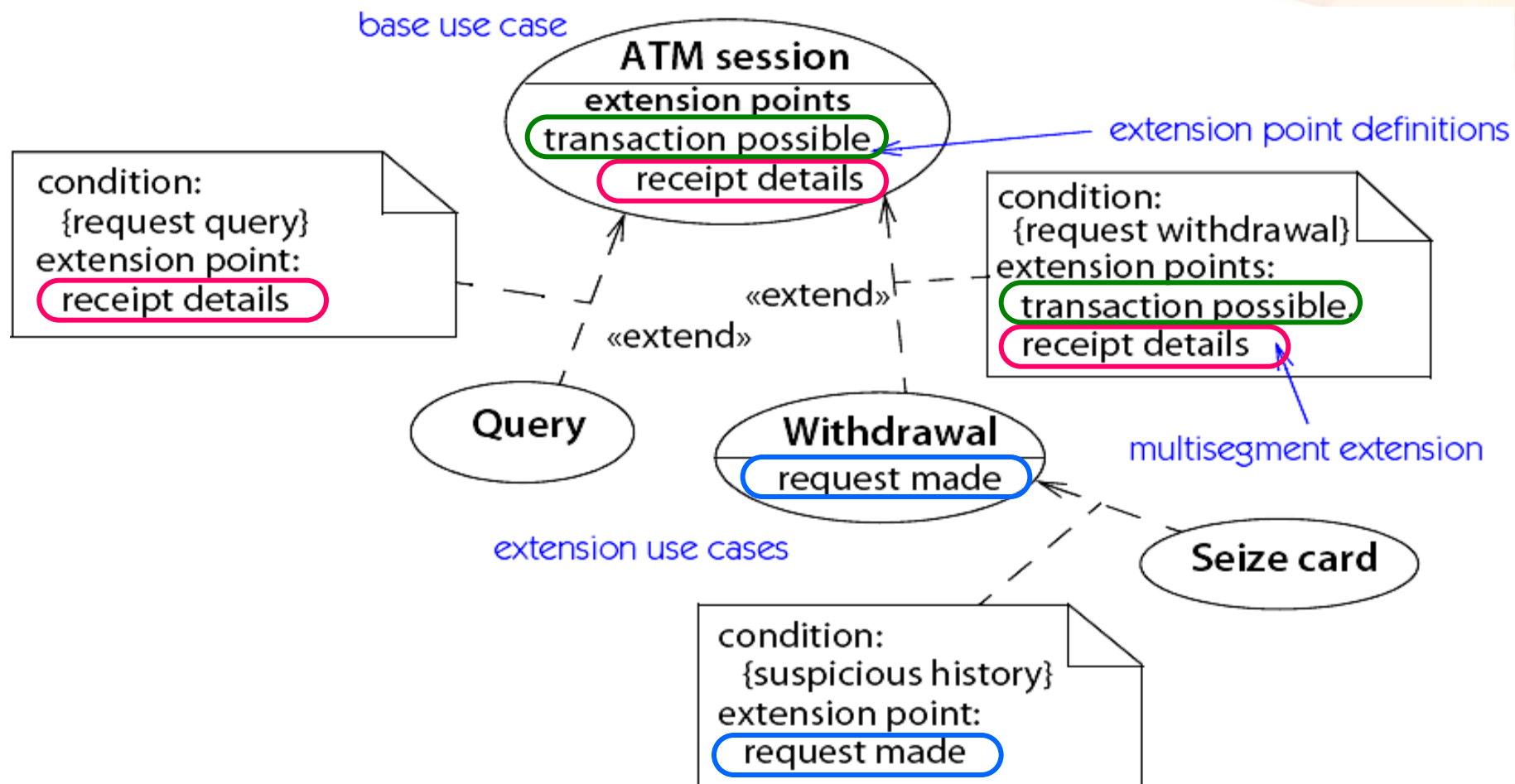
(2 - *include*, 3 - *extend*, 4 - generalizare)



4.1. Diagrame UML de caz de utilizare

Relatiile intre cazurile de utilizare (si actori)

Exemplu de reprezentare a punctelor de extensie



4.1. Diagrame UML de caz de utilizare

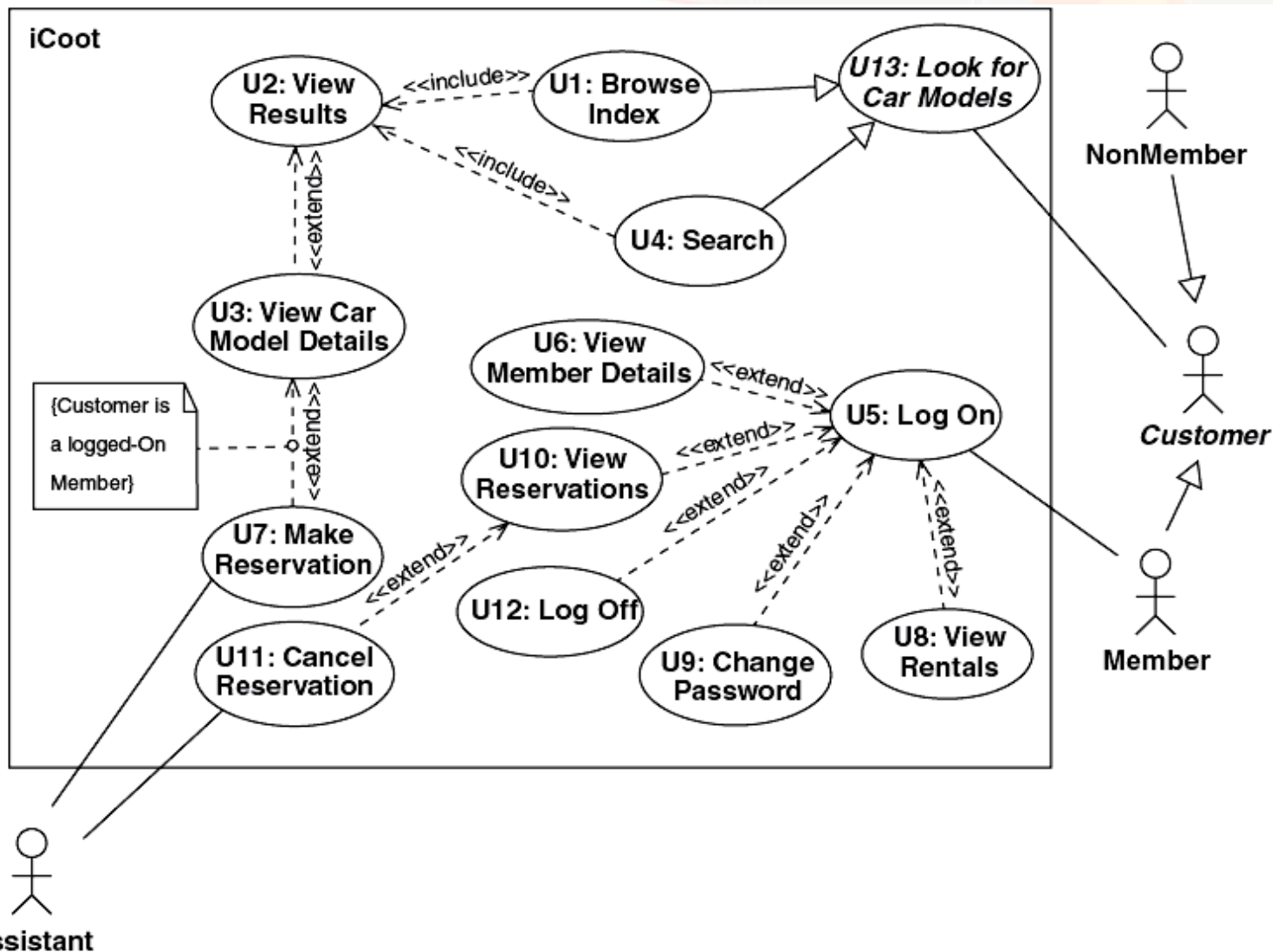
Diagramele UC

**Exemplu de
diagrama UC**

**in care sunt
precizate**

**relatiile intre
UC**

**si relatiile
intre actori**



4.1. Diagrame UML de caz de utilizare

Diagramele UC: gruparea si prioritizarea UC

4.1. Diagrame UML de caz de utilizare

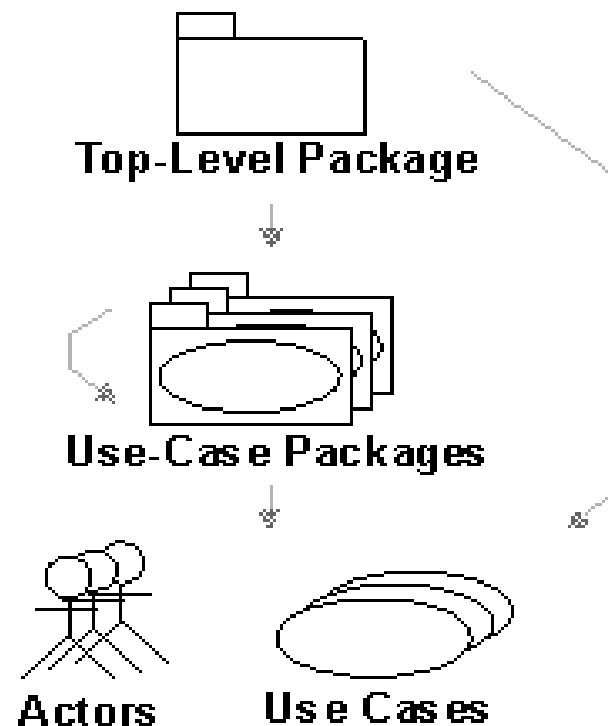
Constructia cazurilor de utilizare

Dupa

- **identificarea** actorilor si a cazurilor de utilizare se trece la
- **constructia** cazurilor de utilizare

Cazurile de utilizare (UC) pot fi

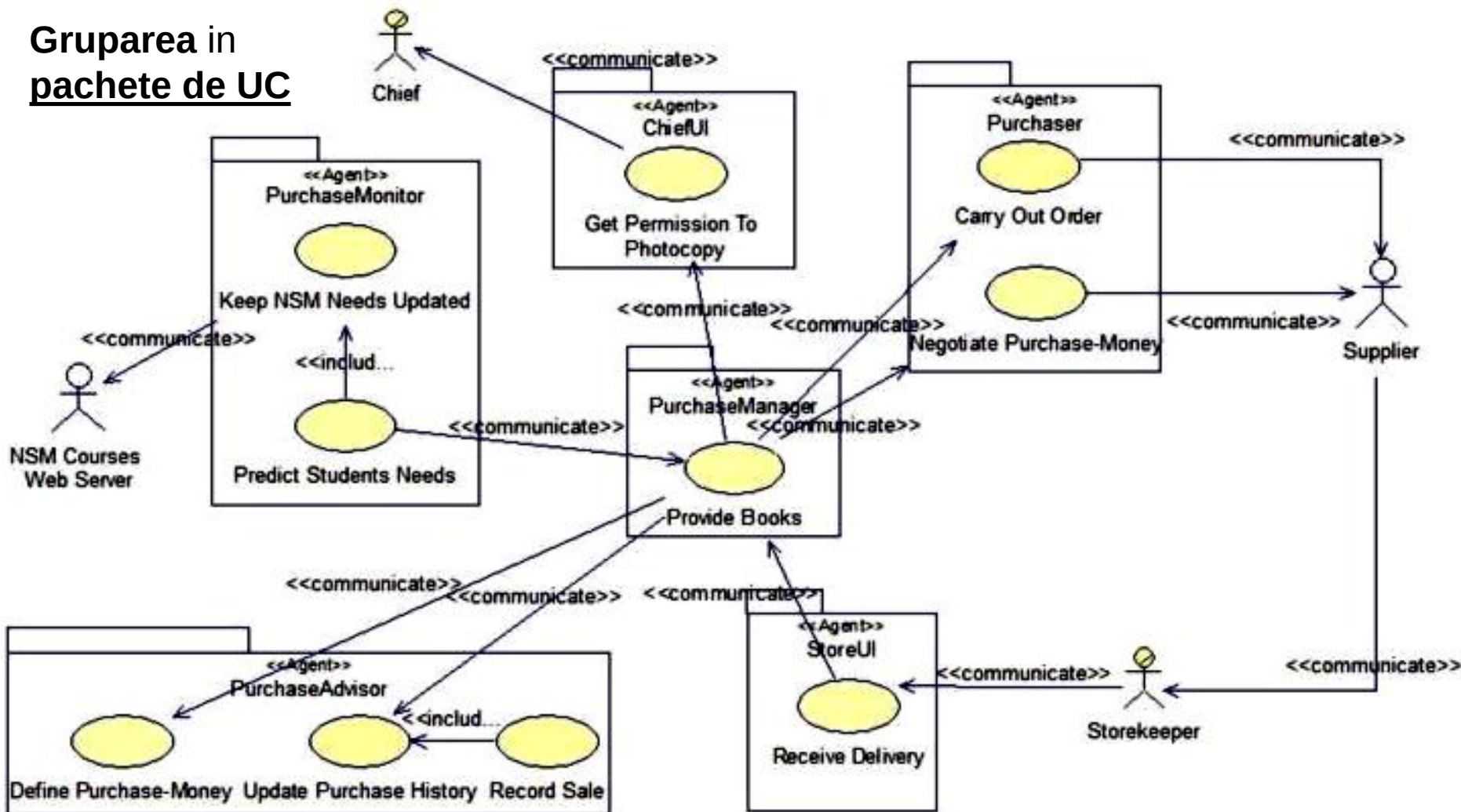
- **grupate** in **pachete de UC**
 - în **functie de înlantuirea lor**
 - **corelate cu marile categorii de actori**
 - client,
 - furnizor,
 - administrator,
 - etc.



4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Gruparea in pachete de UC



4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Dupa gruparea UC este recomandata prioritizarea UC, adica

- clasificarea UC în functie de importanta / prioritatea lor în raport cu:
 - activitatea **organizatiei**
 - **interesul clientilor** pentru ele
 - natura **riscurilor** asociate
 - **simplitatea** realizarii lor sau
 - **impactul** lor asupra proceselor existente
- urmand ca ordinea constructiei UC sa fie
 - dictata de prioritatea UC

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Exemplu de prioritizare a UC

Prioritate maxima (sigur in prima iteratie):

- U1: Browse Index
- U4: Search
- U2: View Results
- U3: View CarModel Details
- U5: Log On

Prioritate medie (eventual in prima iteratie):

- U12: Logoff
- U6: View Member Details
- U7: Make a Reservation
- U10: View Reservations

Prioritate minima (sigur NU in prima iteratie):

- U11: Cancel Reservation
- U8: View Rentals
- U9: Change Password

4.1. Diagrame UML de caz de utilizare

Diagramele UC: descrierile textuale ale UC

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Descrierea textuala a interactiunii trebuie sa se concentreze pe

- ceea **ce** trebuie sa fie facut în UC dat
- **NU** pe **cum** se face
- si pe **simplitate** (principiul KISS)

Pentru asta nu trebuie ezitat în a **fragmenta** un UC

- daca **interactiunile** devin prea **complexe**, prea **confuze**, sau
- daca se gasesc **parti independente** ale UC

Descrierea UC trebuie

- sa **evite** expresii vagi si imprecise
 - de genul: **multe**, **majoritatea**, **destule**, **suficiente**, etc.

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Exemplu de descriere textuala (naratiune) a unui UC

U1: Browse Index. (Specializes U13, includes U2)

Preconditions: None.

Basic course:

1. Customer selects an index heading.
2. Customer elects to view CarModels for the selected index heading.
3. Include U2.

Postconditions: None.

U2: View Results. (Included by U1 and U4, extended by U3)

Preconditions: None.

Basic course:

1. iCoot presents Customer with a summary of each retrieved CarModel, including model number and price.
2. Extend with U3.

Postconditions: None.

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

U7: Make Reservation. (Extends U3.)

Preconditions: *Customer is a Member who has logged on.*

Basic course:

1. *Member elects to reserve CarModel for the details on display.*
2. *iCoot asks Member for confirmation, issuing a warning that failure to collect a reserved CarModel will result in a fine.*
3. *Member confirms Reservation.*
4. *iCoot shows Member the Reservation number and indicates that Assistant will be in touch when a Car is available.*
5. *When an Assistant logs on to Coot, Assistant is given a list of Reservations that require action.*
6. *Assistant takes necessary action to progress Reservations (e.g. promoting to Collectable if a Car is available and moving the Car to the reserved area).*

Postconditions: *Any requested Reservations have been made.*

Abnormal Paths: **a1.** *If Member declines Reservation conditions, no Reservation is made.*

4.1. Diagrame UML de caz de utilizare

Diagramele UC

Exemplu de lista a UC (I)

Use Case List

- **U1: Browse Index:** *A Customer browses the index of CarModels.*
(Specializes U13, includes U2)
- **U2: View Results:** *A Customer is shown the subset of CarModels that were retrieved.* (Included by U1 and U4, extended by U3)
- **U3: View CarModel Details:** *A Customer is shown the details of a retrieved CarModel, such as description and advert.* (Extends U2, extended by U7)
- **U4: Search:** *A Customer searches for CarModels by specifying Categories, Makes and engine sizes.* (Specializes U13, includes U2)
- **U5: Log On:** *A Member logs on to iCoot using their membership number and current password.* (Extended by U6, U8, U9, U10 and U12)
- **U6: View Member Details:** *A Member views some of the details stored by iCoot, such as name, address and CreditCard details.* (Extends U5)

4.1. Diagrame UML de caz de utilizare

Diagramele UC

Exemplu de lista a UC (II)

- **U7: Make Reservation:** *A Member reserves a CarModel when viewing its details.* (Extends U3)
- **U8: View Rentals:** *A Member views a summary of the Cars they're currently renting.* (Extends U5)
- **U9: Change Password:** *A Member changes the password that they use to log on.* (Extends U5)
- **U10: View Reservations:** *A Member views summaries of their unconcluded Reservations, such as date, time and CarModel.* (Extends U5, extended by U11)
- **U11: Cancel Reservation:** *A Member cancels an unconcluded Reservation.* (Extends U10)
- **U12: Log Off:** *A Member logs off from iCoot.* (Extends U5.)
- **U13: Look for CarModels:** *A Customer retrieves a subset of CarModels from the catalog.* (Abstract, generalized by U1 and U4)

4.1. Diagrame UML de caz de utilizare

Diagramele UC – caietul de sarcini rescris ca sinteza a UC

Exemplu de sinteza a UC (I) - descrie modul in care UC se leaga intre ele

*Any Customer can look for CarModels in the catalog, by browsing the CarModel index (**U1**) or by searching (**U4**). In the latter case, the Customer specifies the Categories, Makes and engine sizes that they're interested in. Either way, after each retrieval, the Customer is shown the results as a collection of matching CarModels (**U2**), along with basic information such as CarModel name. The Customer can then choose to view extra information about particular CarModel objects such as a description and an advert (**U3**).*

*A Customer who has become a Member can log on (**U5**) and gain access to extra services. The extra services are: making a Reservation (**U7**), canceling a Reservation (**U11**), checking membership details (**U6**), viewing outstanding Reservations (**U10**), changing their log-on password (**U9**), viewing their outstanding Rentals (**U8**) and logging off (**U12**).*

4.1. Diagrame UML de caz de utilizare

Diagramele UC – caietul de sarcini rescris ca sinteza a UC

Exemplu de sinteza a UC (II) - descrie modul in care UC se leaga intre ele

Customers come in two varieties, Members and NonMembers. In order to reserve a CarModel, a Member must be viewing its details (NonMembers can't make reservations, even when they're viewing details). In order to cancel a Reservation, a Member must be viewing their outstanding Reservations.

*Browsing the index and searching for CarModels are two different ways of looking for CarModels (**U13**). In order to view CarModel details, a Customer must be viewing the results of looking for models (via the browsing or searching route).*

4.1. Diagrame UML de caz de utilizare

**Diagramele UC: constructia sistematica si
colaborativa a UC**

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Este foarte important

- sa se gaseasca **nivelul de abstractie** potrivit (volumul de detalii care trebuie precizate)
- sa se faca o **distinctie** clara între un **scenariu** si un **UC** (familie de scenarii)

Din pacate însa

- nu exista solutii generale, astfel ca
- aprecierea nivelului de abstractie se bazeaza mult pe **experienta**

UC usureaza lucrul în echipa

- **cu conditia** definirii unui **ghid** privind **stilul de redactare**, care
 - contine o **descriere a formatului UC** si a nivelului de detalii
 - astfel încât un model al UC este **urmat de toti membrii echipei**

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Exemplu de **template** al **descrierii textuale** (naratiunii) **cazurilor de utilizare**

1. **Numele** cazului de utilizare (*cat mai sugestiv, pentru a sintetiza UC*)
2. **Scurta descriere** a cazului de utilizare (*un paragraf*)
3. **Actori** (*entitati exterioare sistemului modelat, implicate in UC*)
4. **Preconditii** (*conditiile necesare pentru declansarea UC*)
5. **Evenimentul care declanseaza UC** (*modul in care incepe UC*)
6. **Descriere a interactiunii** dintre actori si fiecare UC (**scenariul principal**)
7. **Alternative** la UC principal (**scenariile alternative si situatiile exceptionale**)
8. **Evenimentul care produce oprirea UC** (*modul in care se incheie UC*)
9. **Postconditii** (*efectele incheierii UC*)

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

Echipa implicata in constructia UC începe cu descrierea lor în linii mari

- sunt descrise **principalele etape** ale fiecarui UC
- facând distinctia între
 - cazul **nominal** (de baza, principal) si
 - situatiile **alternative** si **exceptionale**

Odata terminat acest efort

- **grupuri / membri** ai echipei **aprofundeaza** descrierea unui UC
- fiecare grup / membru **identifica scenariile** si cum se diferentiaza ele
- pornind de la
 - **cunostintele domeniului** si
 - **specificatiile cerintelor** utilizatorilor

4.1. Diagrame UML de caz de utilizare

Constructia cazurilor de utilizare

In cazul lucrului in echipa este foarte indicata

- sincronizarea **evolutiei** cazurilor de utilizare
- prin **reuniuni regulate** între grupurile si membrii individuali ai echipei

Aceste reuniuni permit

- **armonizarea** nivelului de detalii ale diferitelor UC
- **posibilitatea** realocarii **scenariilor** ramase neacoperite
- validarea **UC** creat de un grup sau un membru de catre alte grupuri sau alti membri

4.1. Diagrame UML de caz de utilizare

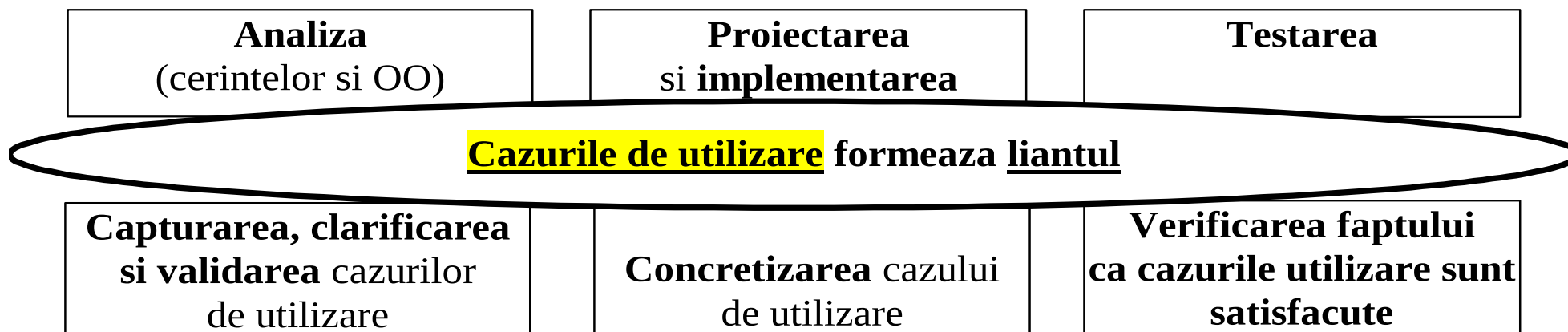
**Diagramele UC: procesele de dezvoltare
ghidate prin UC**

4.1. Diagrame UML de caz de utilizare

Diagramele UC

Cazurile de utilizare intervin de-a lungul intregului proces de dezvoltare

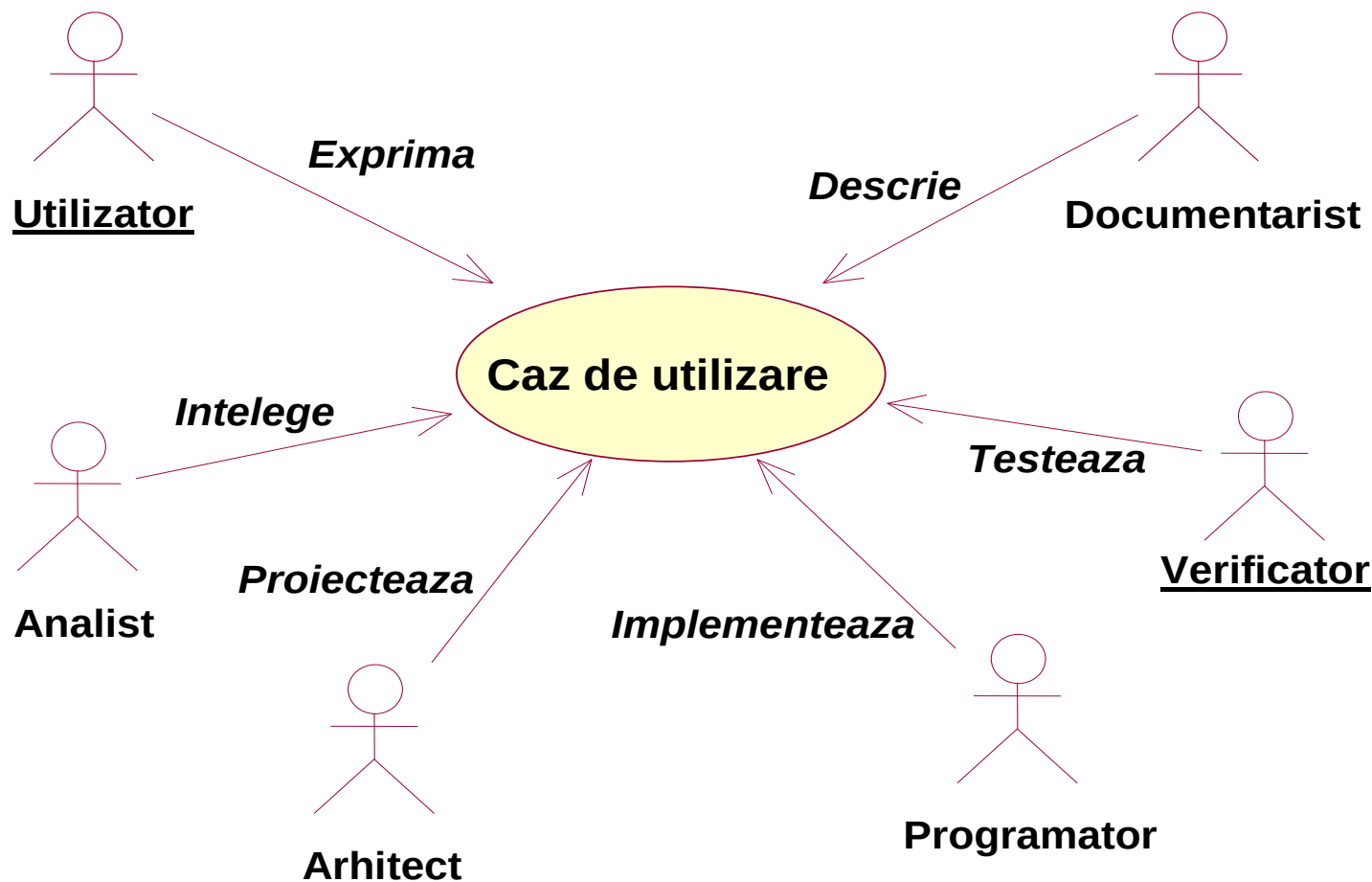
- de la **caietul de sarcini** si pina la **teste** trecand prin
- **analiza, proiectare, codare si redactarea documentatiei** pt. utilizator



4.1. Diagrame UML de caz de utilizare

Diagramele UC

Rolurile dezvoltatorilor in crearea si utilizarea UC intr-un proces de dezvoltare ghidat de UC

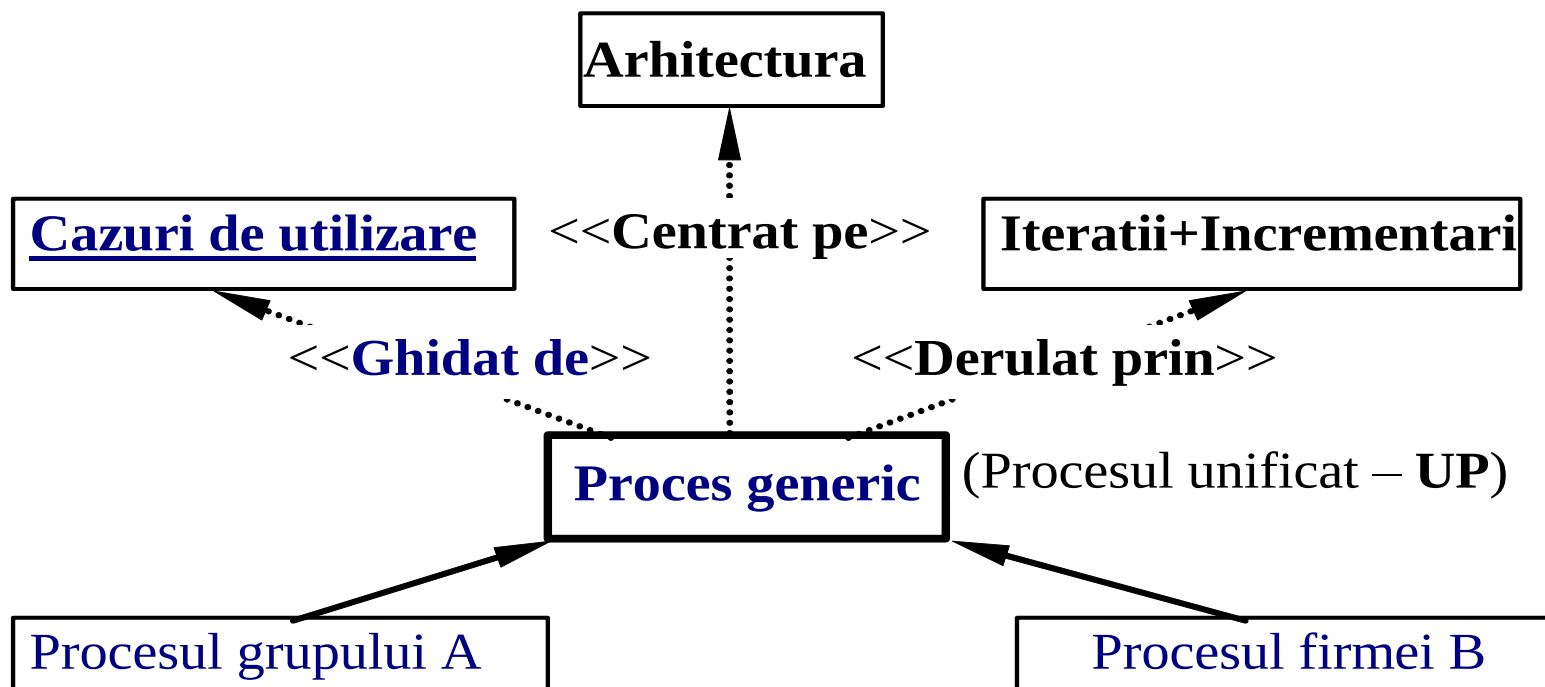


4.1. Diagrame UML de caz de utilizare

Diagramele UC

Analiza cerintelor prin UC se potrivește foarte bine unei abordări a dezvoltării sistemelor software

- centrat pe **arhitectura** și
- derulat **iterativ** și prin **incrementări**



4.1. Diagrame UML de caz de utilizare

Diagramele UC

