Ingineria Programelor

Unified Modeling Language – UML

Mihai TOGAN
mihai.togan@mta.ro

Agenda

- *Introduction to UML*
- *UML Diagrams*
  - *Use-case diagrams*
  - *Sequence diagrams*
  - *Activity diagrams*
  - …

Unified Modeling Language – UML

for object-oriented development

- **Limbaj de modelare obiect (NU este un proces/metoda de dezvoltare)**
- **Independent de procesul de dezvoltare folosit**
- **Necesitatea unei standardizari a elementelor de modelare folosite in metodele de dezvoltare orientata obiect**
  - **intre 1989 si 1994 erau folosite mai mult de 50 de limbaje de modelare software, fiecare cu propriile notatii**
  - **utilizatorii doreau un limbaj standardizat, o lingua franca a modelării**

Unified Modeling Language – UML

for object-oriented development

- Limbaj pentru specificarea, vizualizarea, construirea si documentarea elementelor sistemelor software:
  - un limbaj grafic care ne permite să reproducem „pe hârtie” ceea ce este produs în procesul de dezvoltare a unui sistem software
  - poate fi folosit si pentru alte sisteme, cum ar fi procesele de afaceri (business processes)
- UML este un instrument de comunicare
- Ca orice limbaj, UML are:
  - Notatii (simboluri, alfabetul)
  - Sintaxă si gramatică (reguli pentru combinarea simbolurilor)

UML Specifications

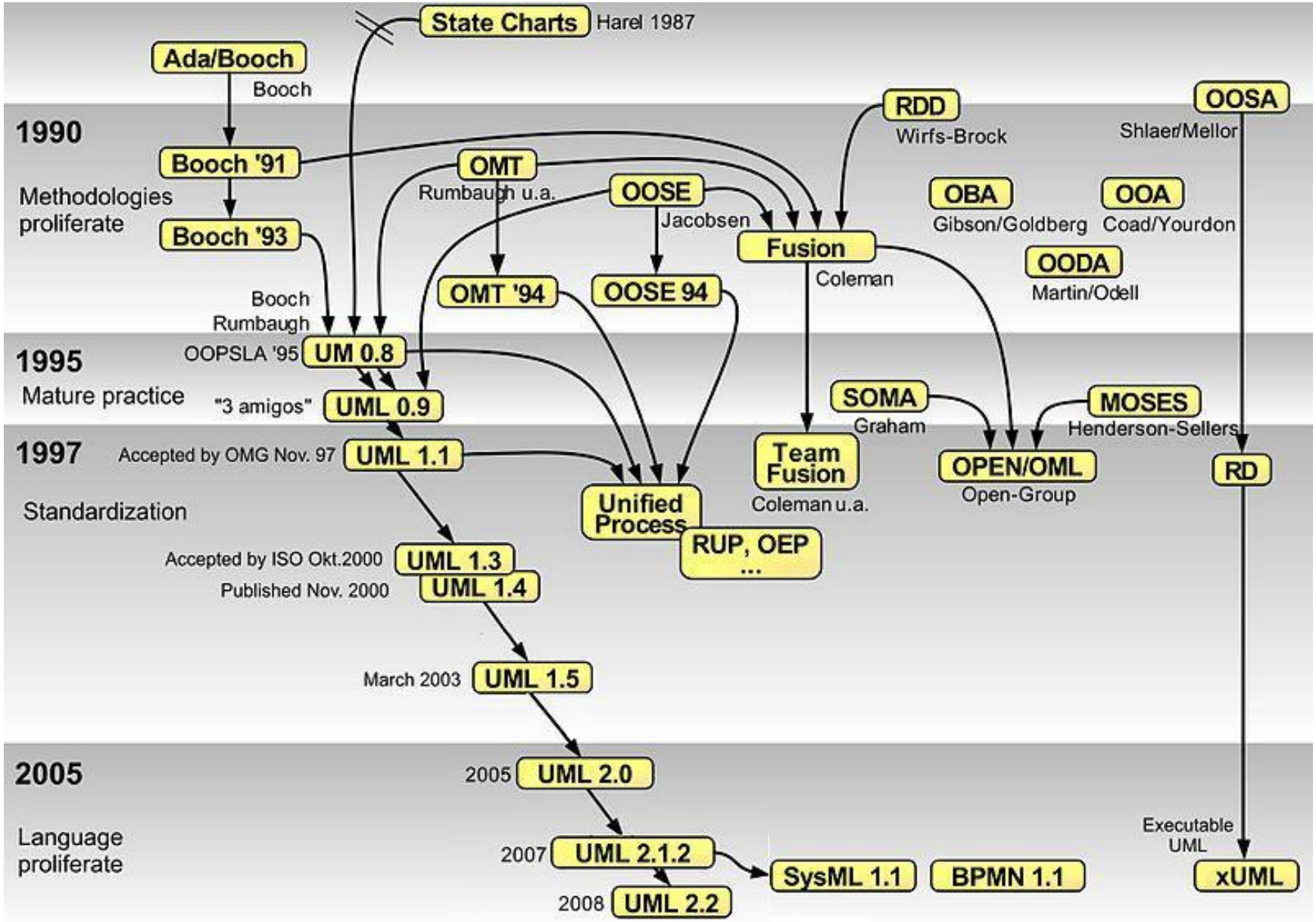
- Prima versiune UML a fost publicata in 1996
  - rezultatul unificarii a trei limbaje de modelare software object-oriented: Booch, OMT (Rumbaugh) si OOSE (Jacobson)
  - UML se constituie din unirea acestor limbaje şi în plus are o expresivitate mai mare

- Ianuarie 1997: UML 1.0
  - a fost propus spre standardizare în cadrul Object Management Group (OMG)
- Noiembrie 1997: UML 1.1

- 1998: UML 1.2
- 1999: UML 1.3
- 2002: UML 1.4
- 2004: UML 2.0
- 2010: UML 2.3
- 2011: UML 2.4
- 2015: UML 2.5 (versiune curenta)

http://http://wwhttp://wwwhttp://www.uml.org

History


State Charts
Harel 1987
Ada/Booch
Booch
1990
Booch '91
Booch '93
Methodologies
proliferate
OMT
Rumbaugh u.a.
OOSE
Jacobsen
RDD
Wirfs-Brock
OOSA
Shlaer/Mellor
OBA
Gibson/Goldberg
OOA
Coad/Yourdon
Fusion
Coleman
OODA
Martin/Odell
Booch
Rumbaugh
OOPSLA '95
OMT '94
OOSE 94
1995
Mature practice
UM 0.8
"3 amigos"
UML 0.9
SOMA
Graham
MOSES
Henderson-Sellers
1997
Accepted by OMG Nov. 97
UML 1.1
Team
Fusion
Coleman u.a.
OPEN/OML
Open-Group
RD
Standardization
Unified
Process
RUP, OEP
...
Accepted by ISO Okt.2000
UML 1.3
Published Nov. 2000
UML 1.4
March 2003
UML 1.5
2005
2005
UML 2.0
Language
proliferate
2007
UML 2.1.2
SysML 1.1
BPMN 1.1
Executable
UML
xUML
2008
UML 2.2

UML – Diagrams

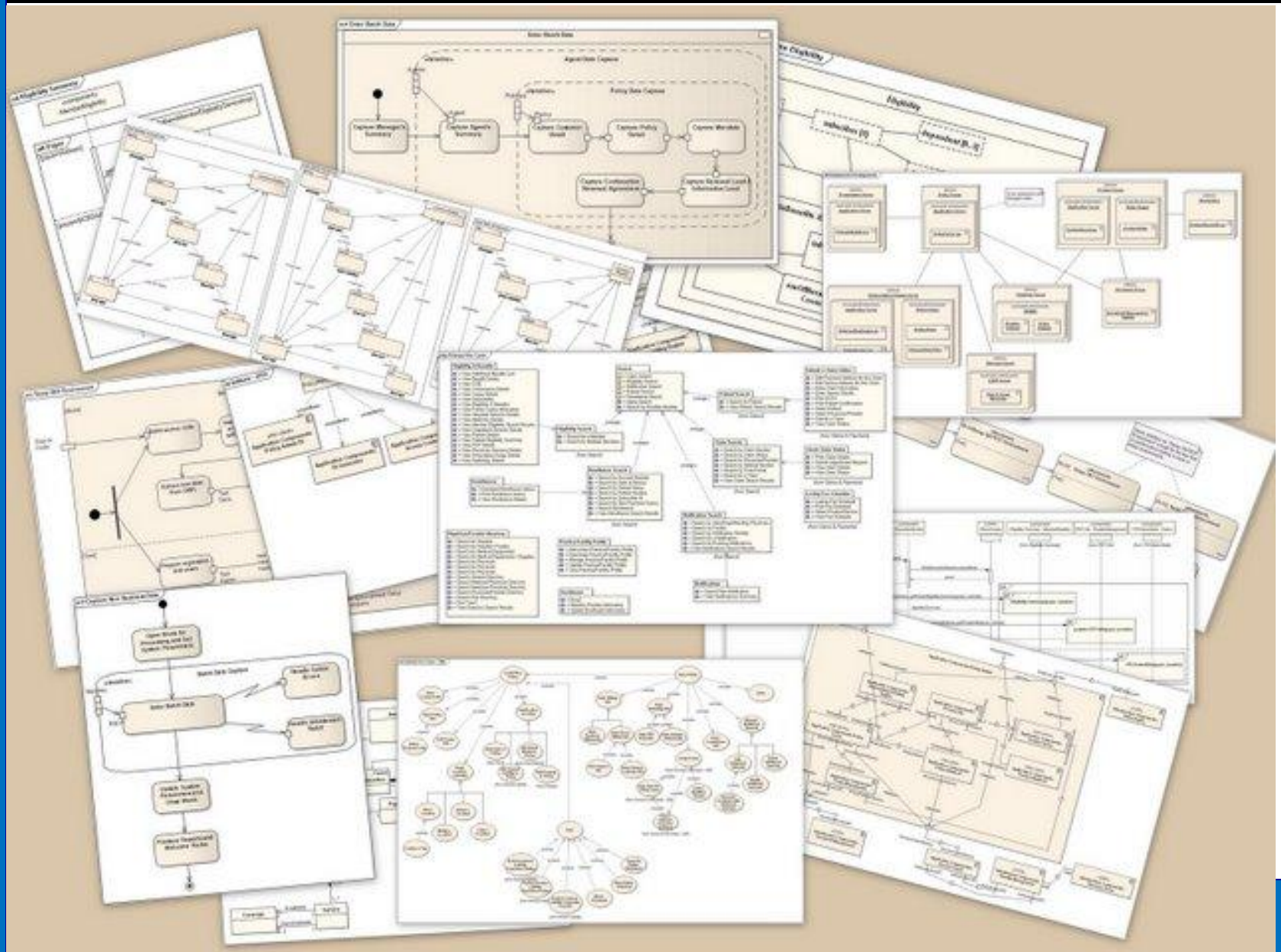
- **Elementul principal de modelare UML DIAGRAMA**

- Diagrame de structură:
  ce conține sistemul
  - Clase
  - Componente
  - *Structuri compuse*
  - Desfăşurare
  - Obiecte
  - *Pachete*

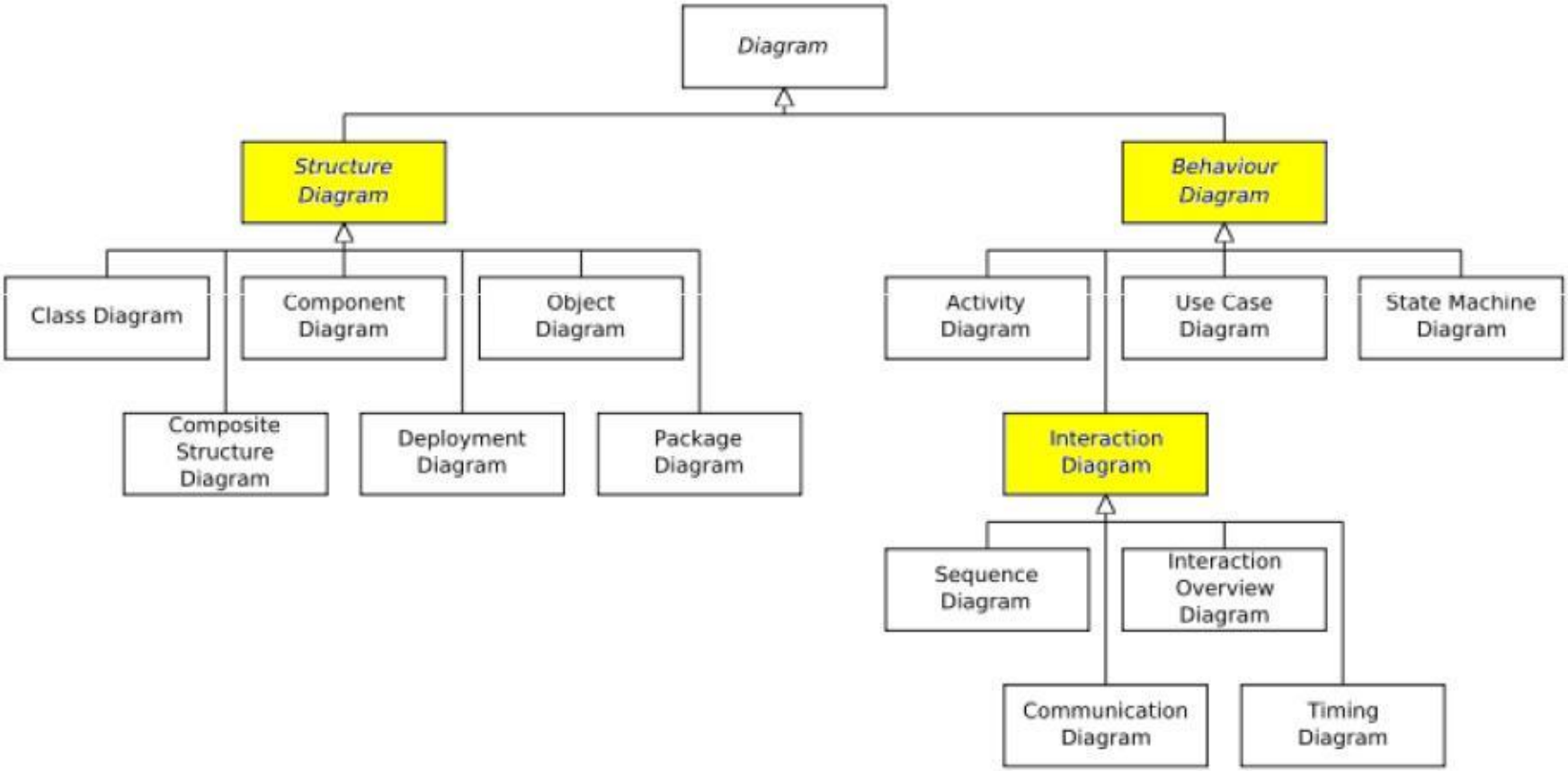
Diagrame introduse în UML 2.0

- Diagrame de comportament:
  ce se întâmplă în sistem
  - Activități
  - Maşini de stare
  - Cazuri de utilizare
- Diagrame de interacțiune:
  fluxurile de control şi date dintre componentele sistemului
  - Comunicare
  - *Interacțiuni generale*
  - Secvențe
  - *Cronometrare*

UML – Diagrams



UML – Diagrams



UML Diagrams

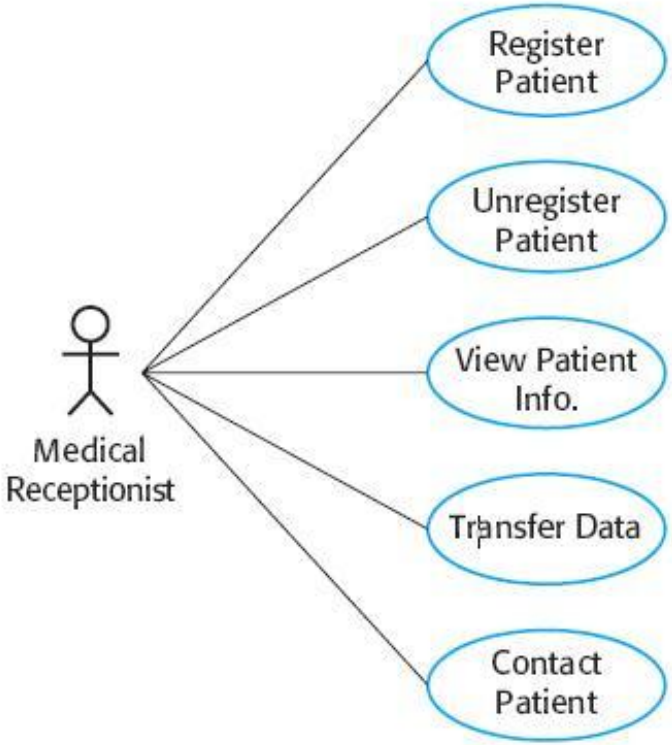
- Activity Diagram - Activities
- Class Diagram - Structured Classifiers
- Communication Diagram - Interactions
- Component Diagram - Structured Classifiers
- Composite Structure Diagram - Structured Classifiers
- Deployment diagram - Deployments
- Interaction Overview Diagram - Interactions
- Object Diagram - Classification
- Package Diagram - Packages
- Profile Diagram - Packages
- State Machine Diagram - State Machines
- Sequence Diagram - Interactions
- Timing Diagram - Interactions
- Use Case Diagram - Use Cases

Most common UML-diagrams

- Diagrama cazurilor de utilizare (*use cases diagram*)
- Diagrama de secventa (*sequence diagram*)
- Diagrama de activitati (*activity diagram*)
- Diagrama de clase (*class diagram*)
- Diagrama de stari (*state diagram*)
- Diagrama de componente (*components diagram*)
- Diagrama de distributie (*deployment diagram*)

Use-case diagram

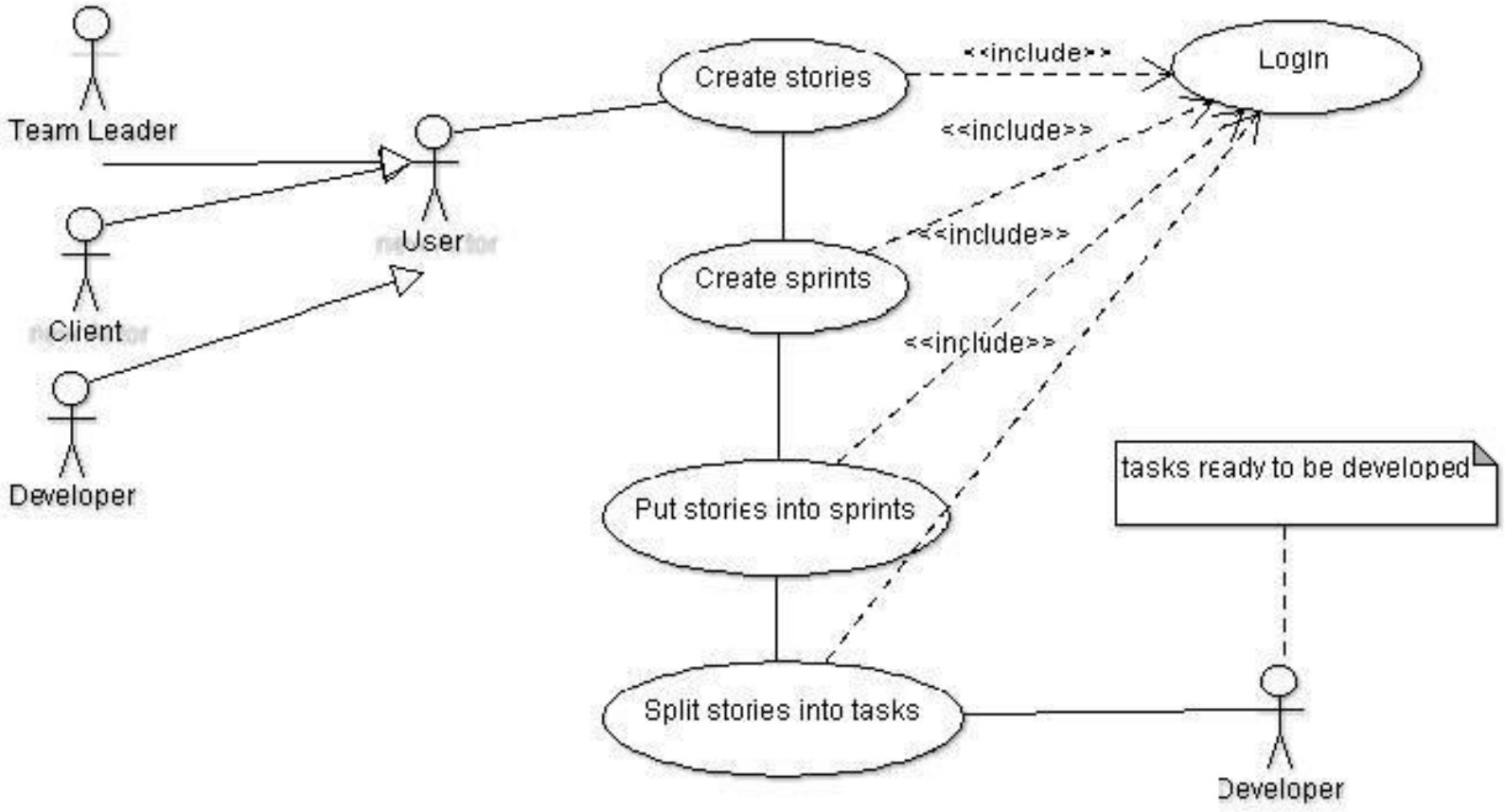
- **Prezintă funţionalităţile sistemului folosind *actori*, *use case*-uri şi *dependenţe* între ele (comportamental)**



Use cases involving the role 'medical receptionist'

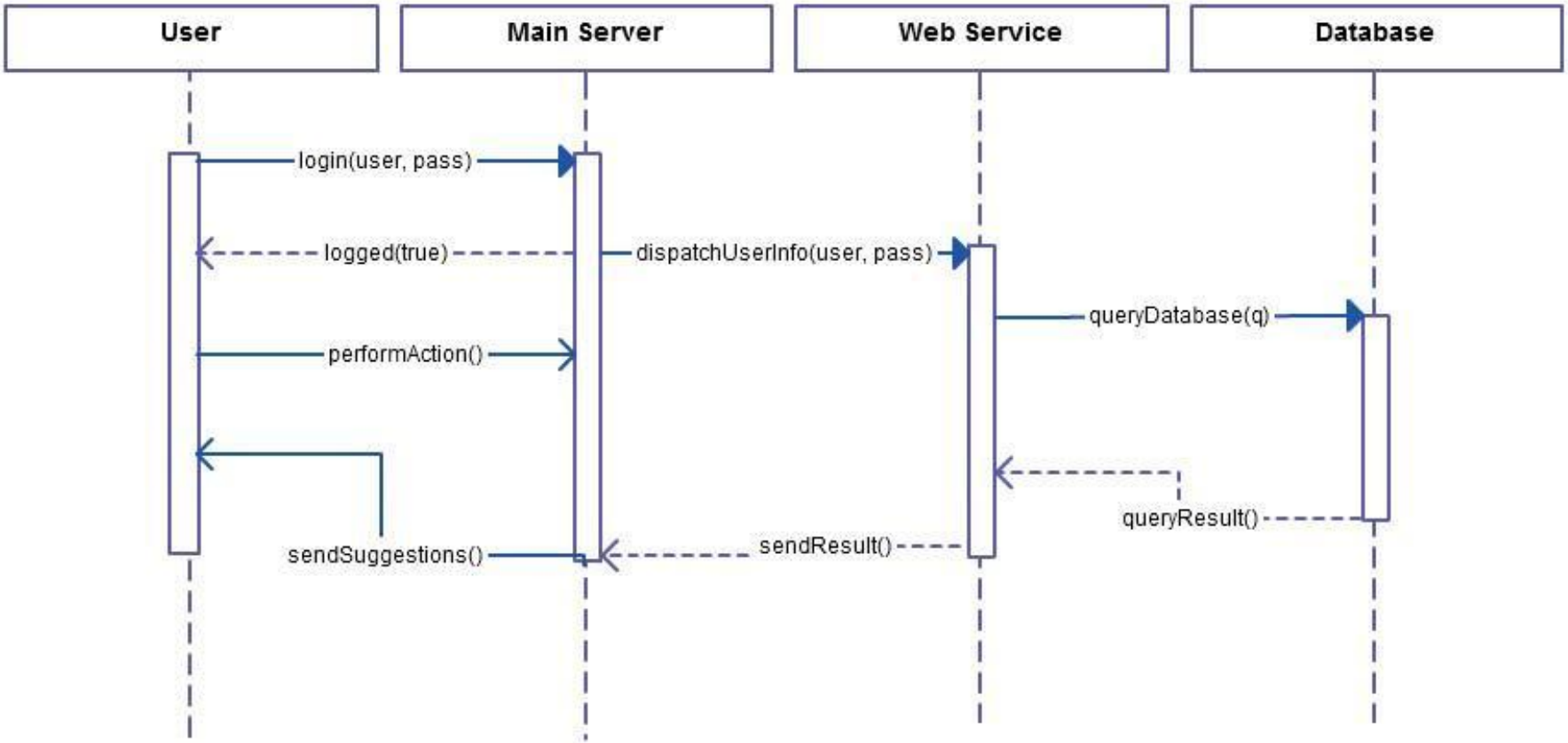
Use-case diagram – Example

- **Prezintă funţionalităţile sistemului folosind *actori*, *use case*-uri şi *dependenţe* între ele (comportamental)**



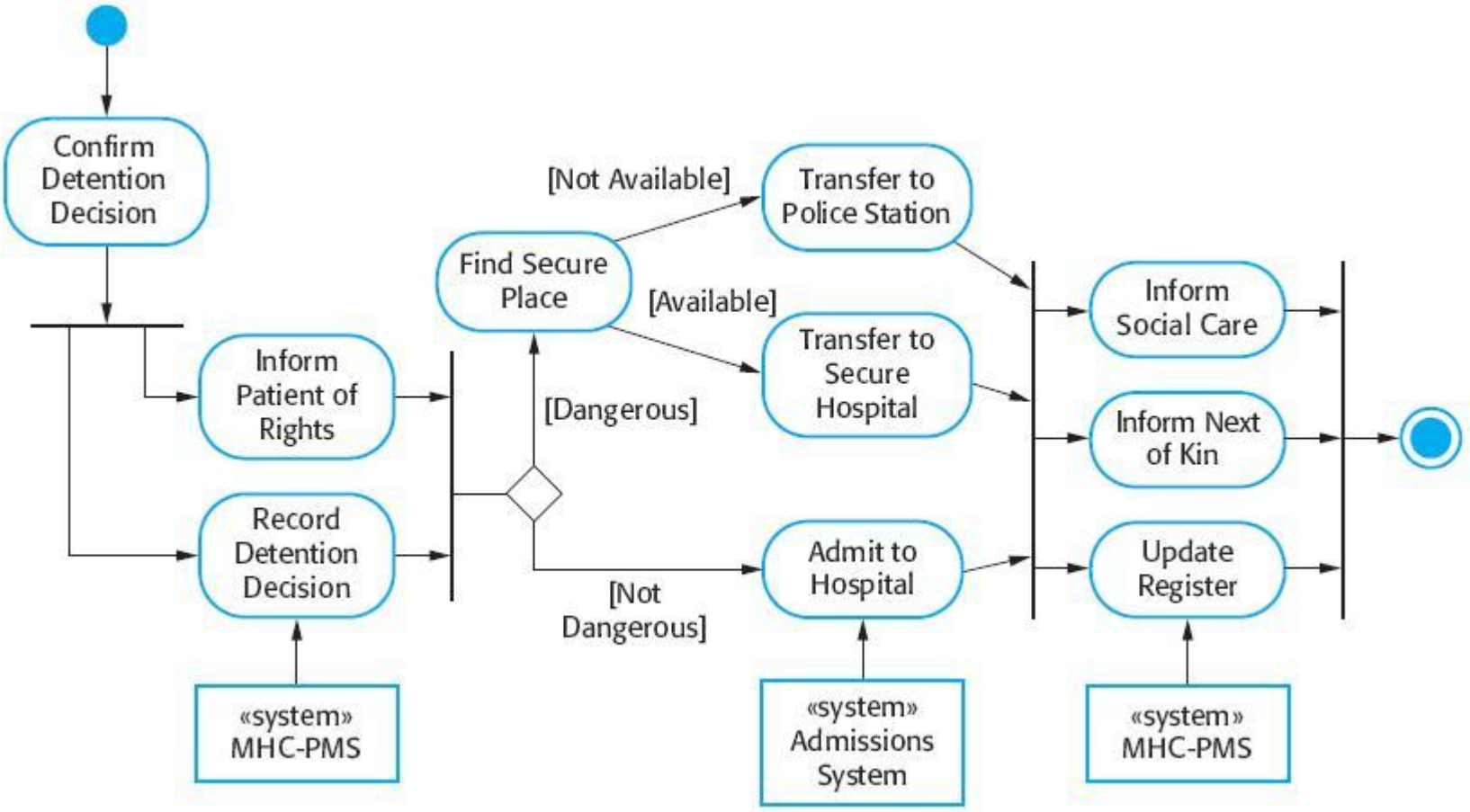
Sequence diagram

- **Prezinta interactiunea componentelor sistemului şi legăturile între ele (comportamental)**



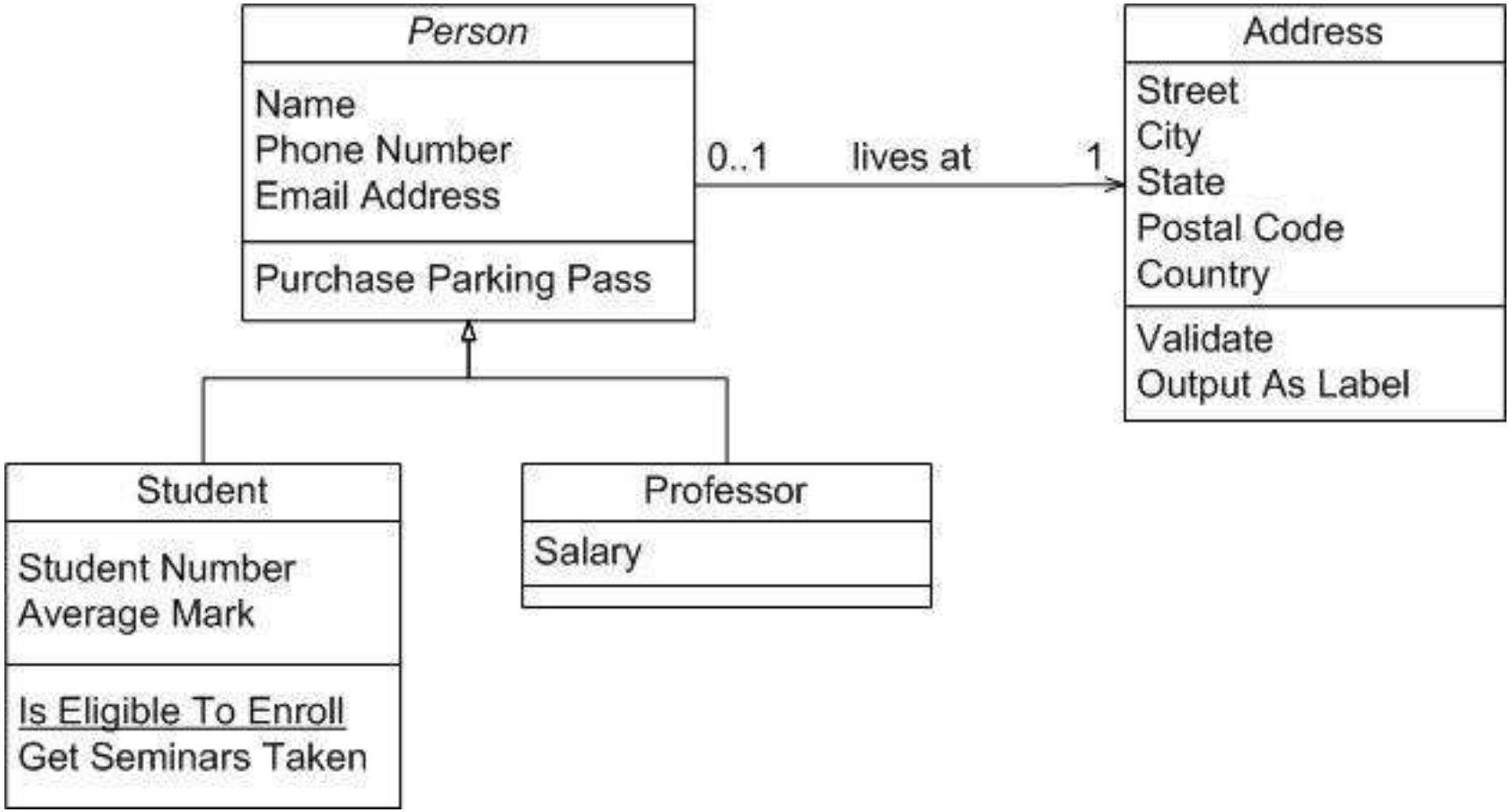
Activities diagram

- **Prezinta componentele sistemului şi legăturile între ele (comportamental)**



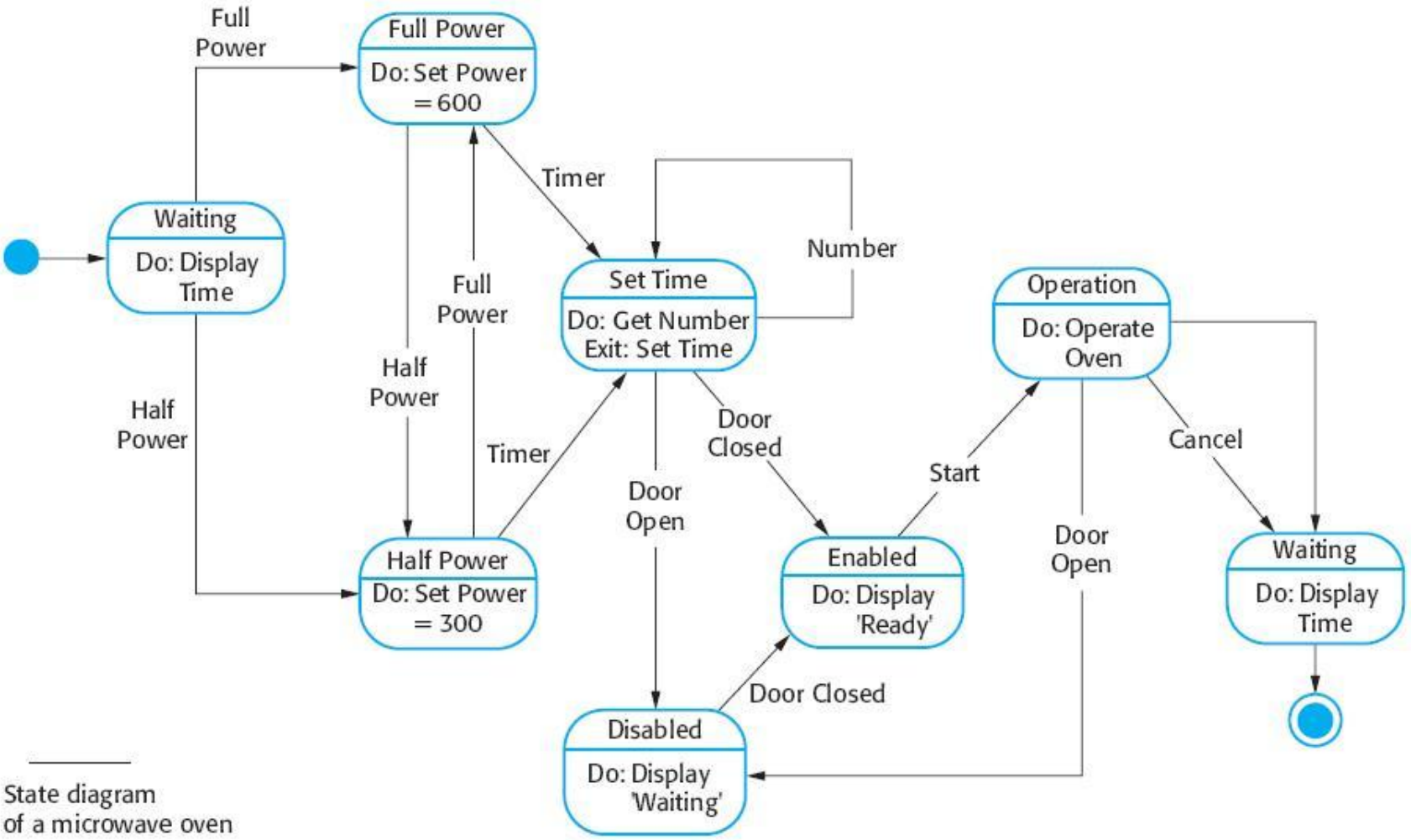
Class diagram

- **Prezinta clasele (atribute, metode) sistemului şi relatiile între ele (structural)**



States diagram

- **Prezinta starile sistemului şi tranzitiile intre aceste stari (comportamental)**



State diagram
of a microwave oven

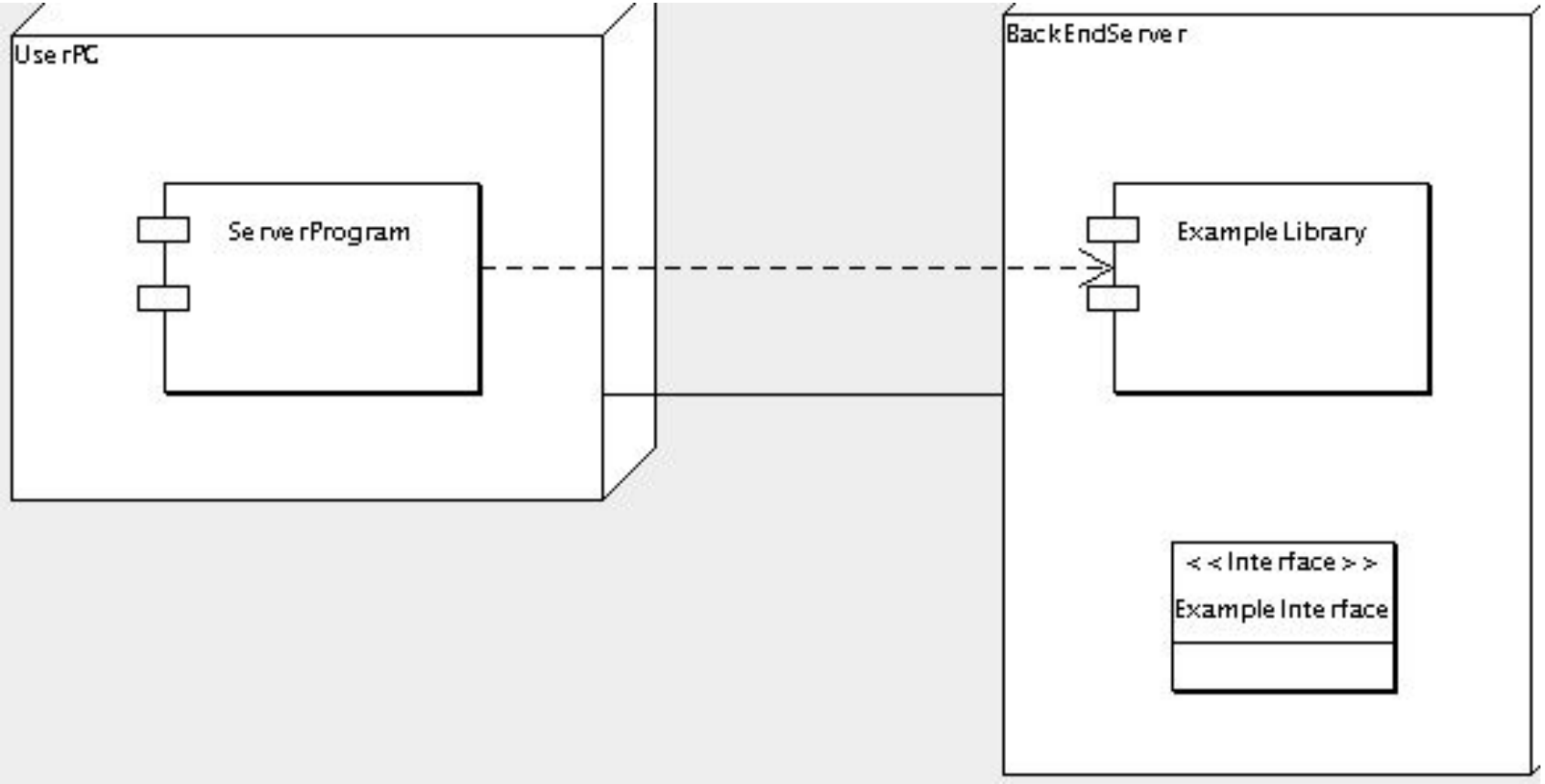
States diagram

State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for 5 seconds. Oven light is on. Display shows 'Cooking complete' while buzzer is sounding.

Stimulus	Description
Half power	The user has pressed the half power button
Full power	The user has pressed the full power button
Timer	The user has pressed one of the timer buttons
Number	The user has pressed a numeric key
Door open	The oven door switch is not closed
Door closed	The oven door switch is closed
Start	The user has pressed the start button
Cancel	The user has pressed the cancel button

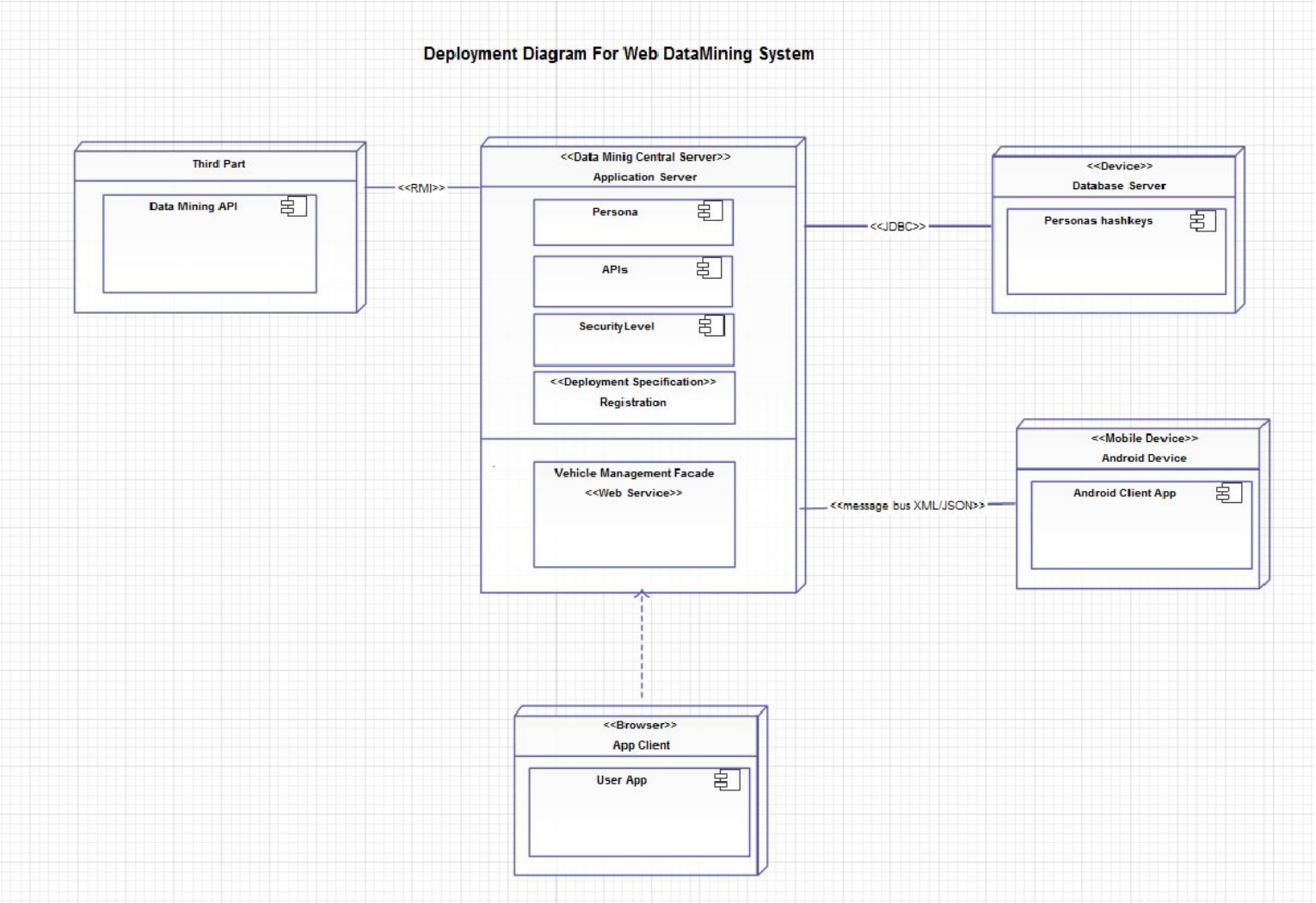
Components diagram

- **Prezinta componentele sistemului şi legăturile între ele (structural)**



Deployment diagram

- **Prezinta modelarea structurii hardware si software a sistemului şi legăturile între ele (structural)**



Use-cases diagram (UCD)

- Prezinta scenariile de utilizare pentru sistemul software

- Se elaboreaza in fazele initiale ale procesului de definire a cerintelor software
  - specificarea cerintele dpdv al utilizatorului
  - specificarea cerintelor de functionalitate dpdv al sistemului

- Suma cazurilor de utilizare este întregul sistem
- Delimiteaza granitele sistemului
- Permit comunicarea cu persoane fără cunostinte tehnice IT
- Partitionează functionalitatea
- Ajută la: planificare, testare, crearea manualurilor de utilizare

Use-cases diagram (UCD)

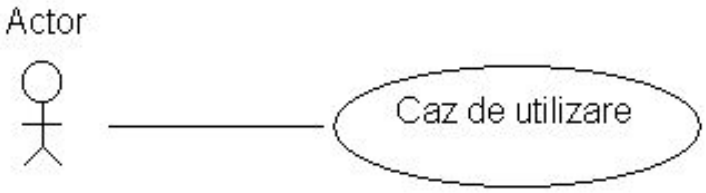
- **Exprima interactiunea ENTITATE (externa) ⇔ SISTEM**
  - UTILIZATOR ⇔ **SISTEM**
  - Componenta EXTERNA ⇔ **SISTEM**

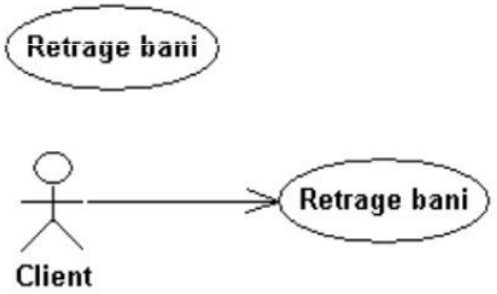
- **Modelul include:**

  - **Actori (roluri) - entitati externe cu care sistemul interactioneaza**

  - **Use-cases - cazuri de utilizare (scenariile) ⇒ functionalitatile sistemului**

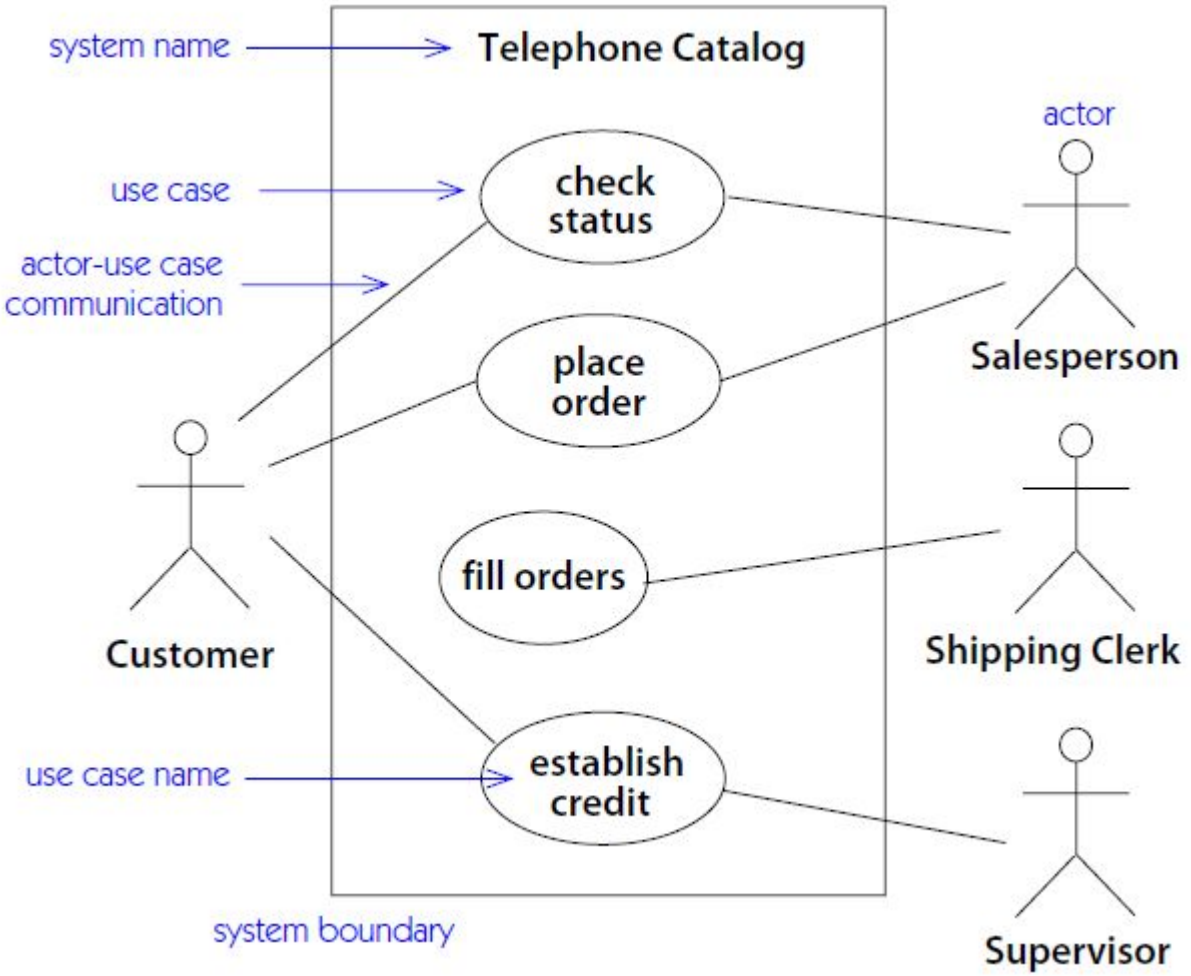
  - **Relatiile dintre ele ⇒ diagrama cazurilor de utilizare (use-cases diagram)**





Use-cases diagram (UCD)

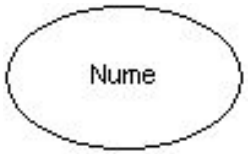
- **Exprima interactiunea ENTITATE (externa) ⇔ SISTEM**



Use-case

- **Use-Case**
  - Reprezinta o functionalitate a programului
  - Precizeaza ce face un program sau subprogram
  - Nu precizeaza cum se implementeaza o functionalitate

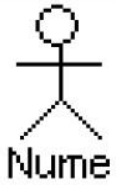
- Identicarea use case-urilor se face pornind de la cerintele clientului si analizand descrierea problemei

- Notatie: 

- Atribute:
  - Nume fraza verbala ce denumeste o operatie sau comportament din domeniul problemei
- Restrictie: Numele este unic

Actor

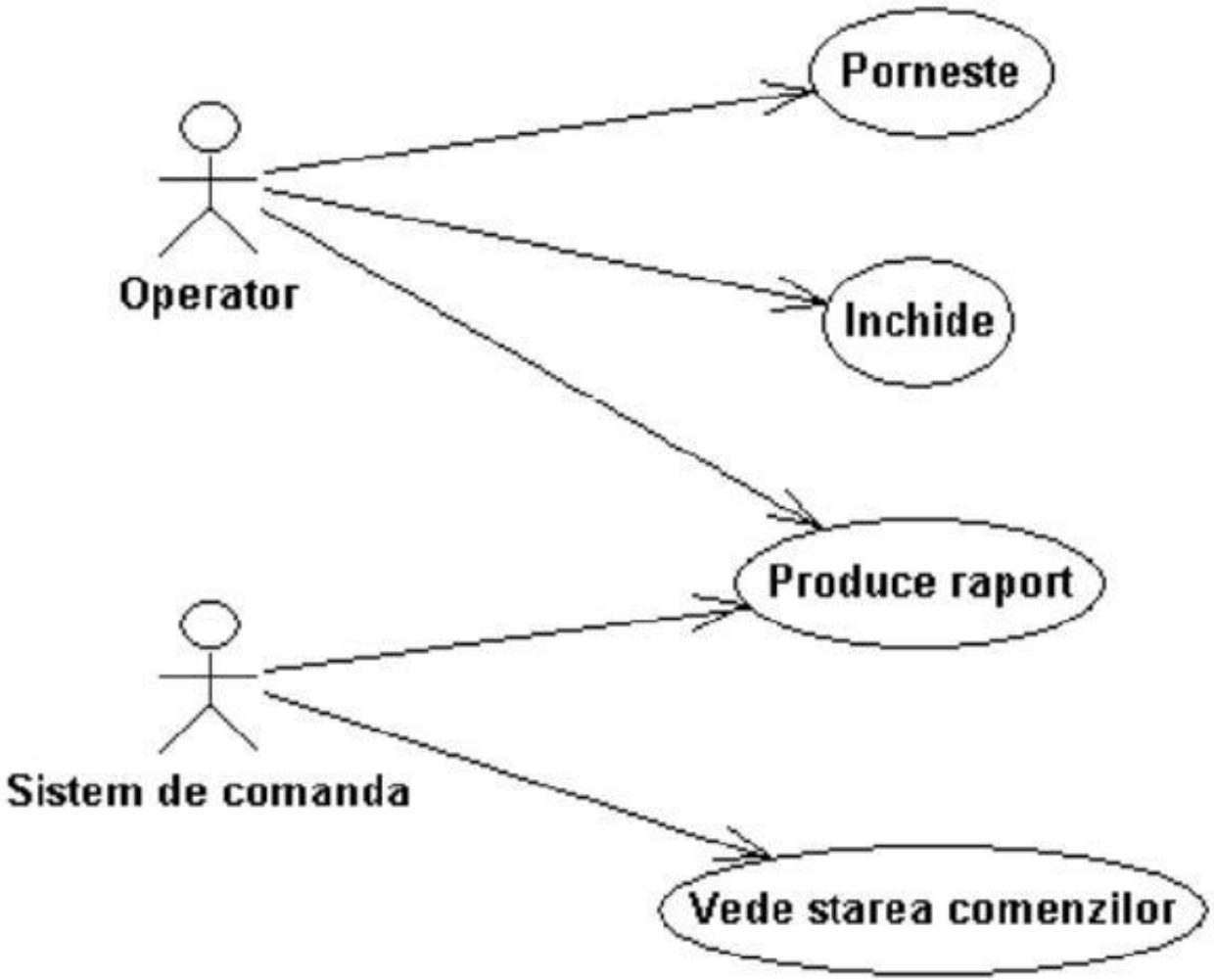
- **Notatie:**



- **Atribute**
  - **Numele actorului indica rolul pe care actorul il joaca in interactiunea cu un use-case (restrictie numele trebuie sa fie unic)**
- **Reprezinta un rol pe care utilizatorul unui use-case il joaca atunci cand interactioneaza cu sistemul**
  - **Initiaza executia unor cazuri de utilizare**
  - **Ofera functionalitate pentru realizarea unor cazuri de utilizare**
- **Poate fi:**
  - **utilizator (uman)**
  - **sistem software**
  - **sistem hardware**
  - **…**

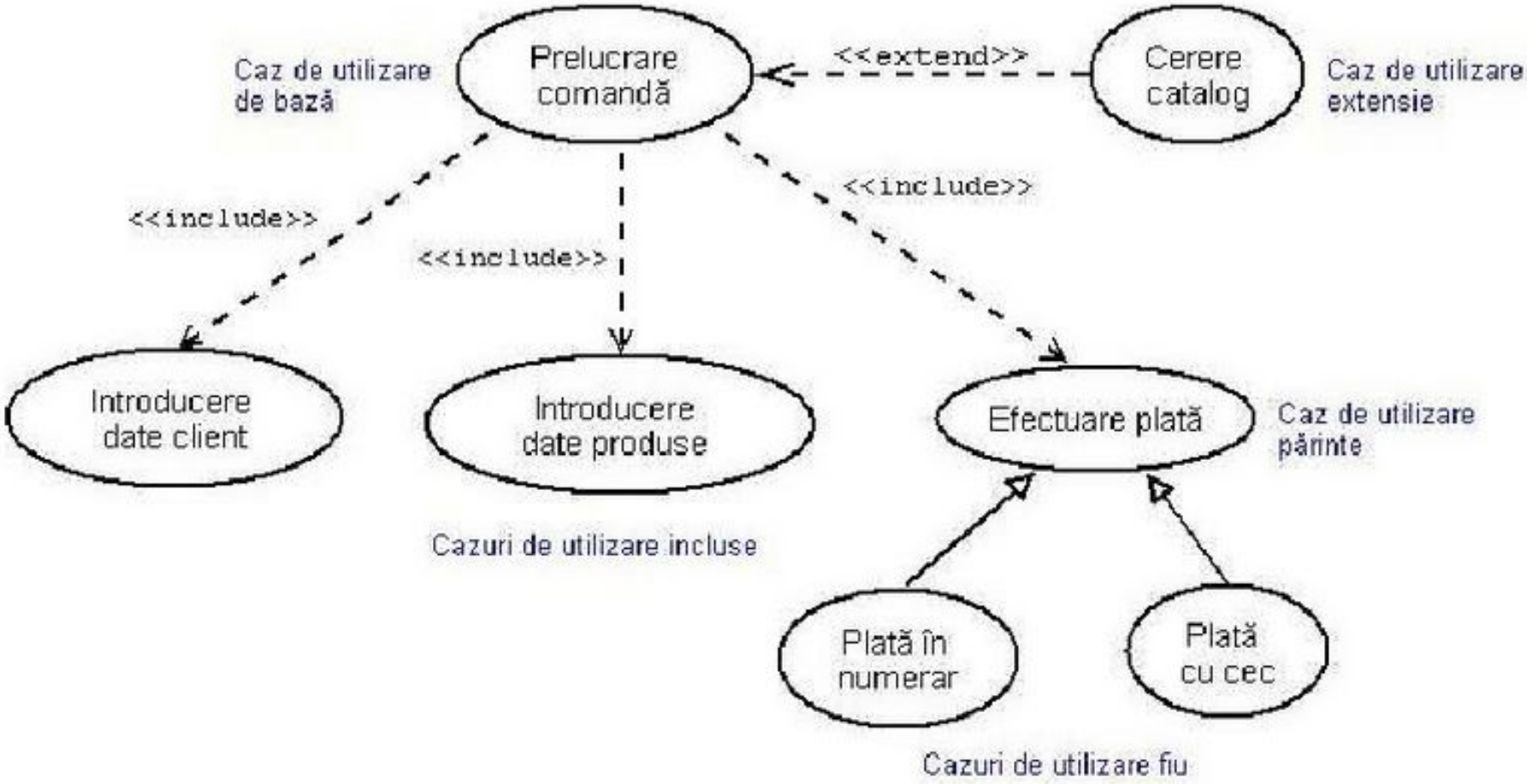
UCD – Example

- Cazuri de utilizare multiple



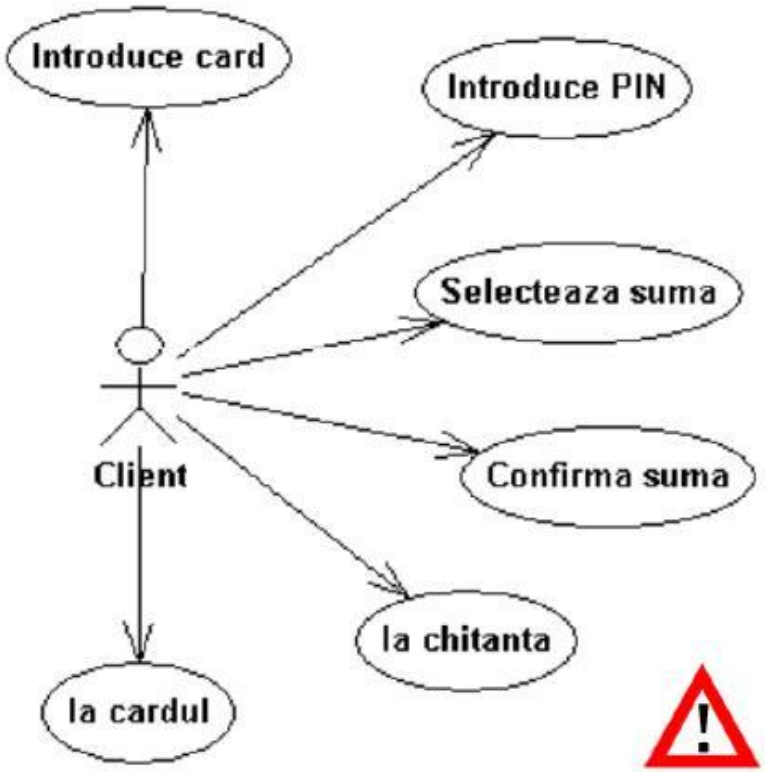
UCD – Example

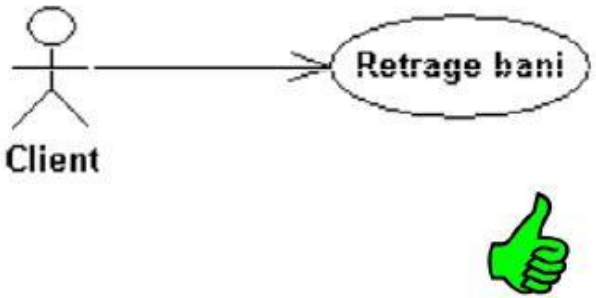
- **Exemplu: achizitionare de produse**



UCD – Example

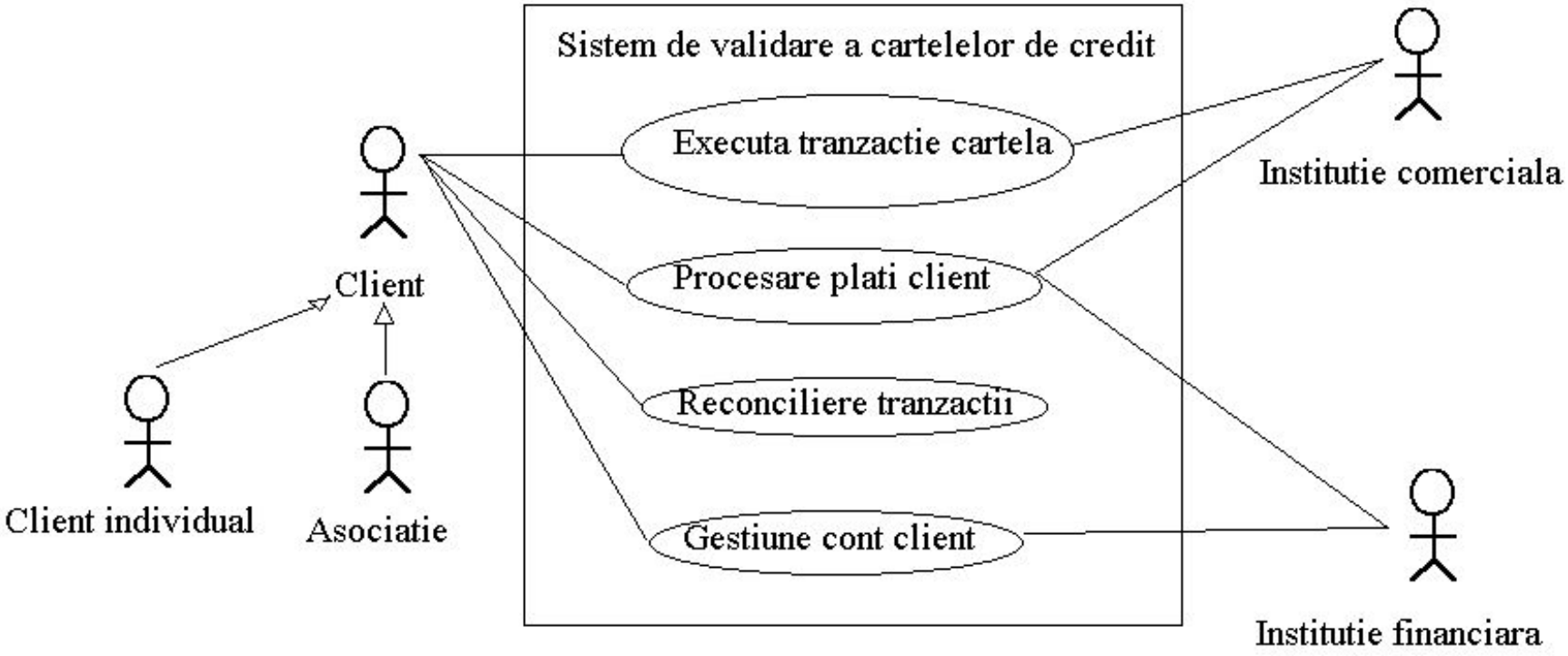
- **Atentie la granularitate !!**



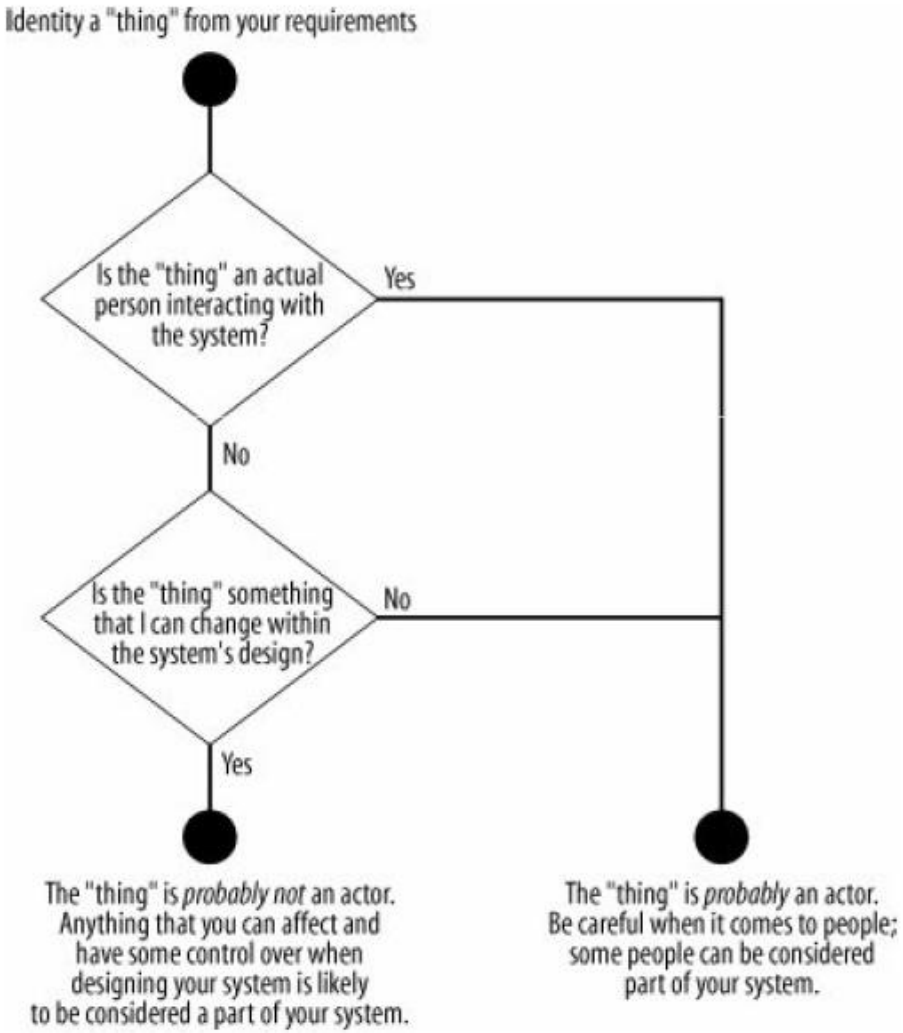


Un caz de utilizare trebuie să satisfacă un scop pentru actor

UCD – Example


Sistem de validare a cartelelor de credit
Executa tranzactie cartela
Procesare plati client
Reconciliere tranzactii
Gestiune cont client
Client
Client individual
Asociatie
Institutie comerciala
Institutie financiara

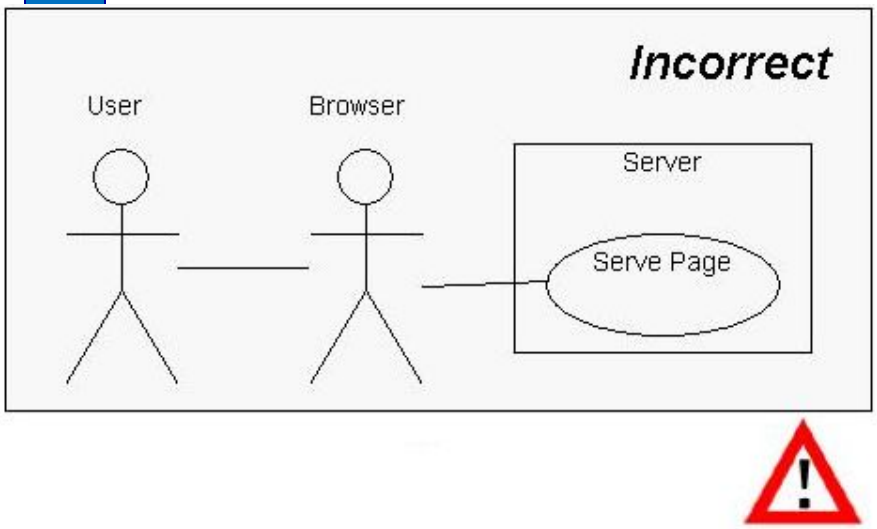
UCD – Identificarea actorilor

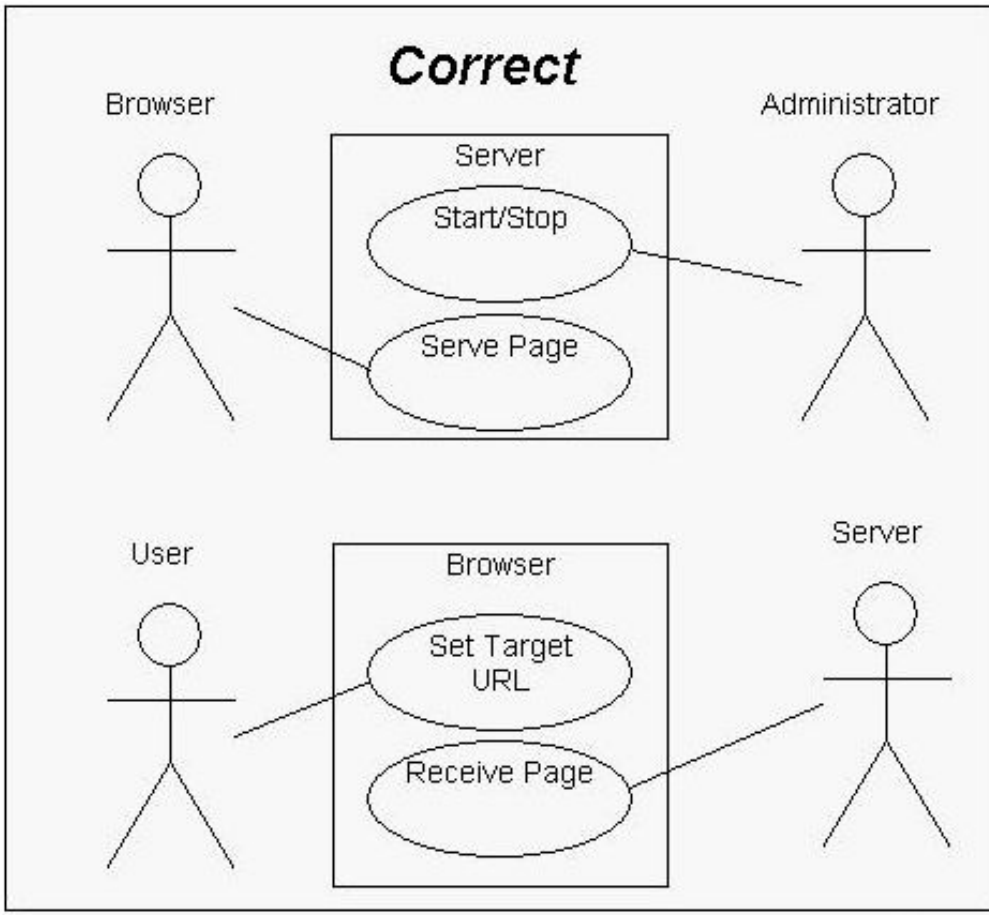


Ex.: timpul poate fi un actor

UCD – Example

- **Fara interactiuni directe intre actori !!**





UCD – Relatii

- @todo

Bibliography:

James Rumbaugh, et al.
The Unified Modeling Language Reference Manual