



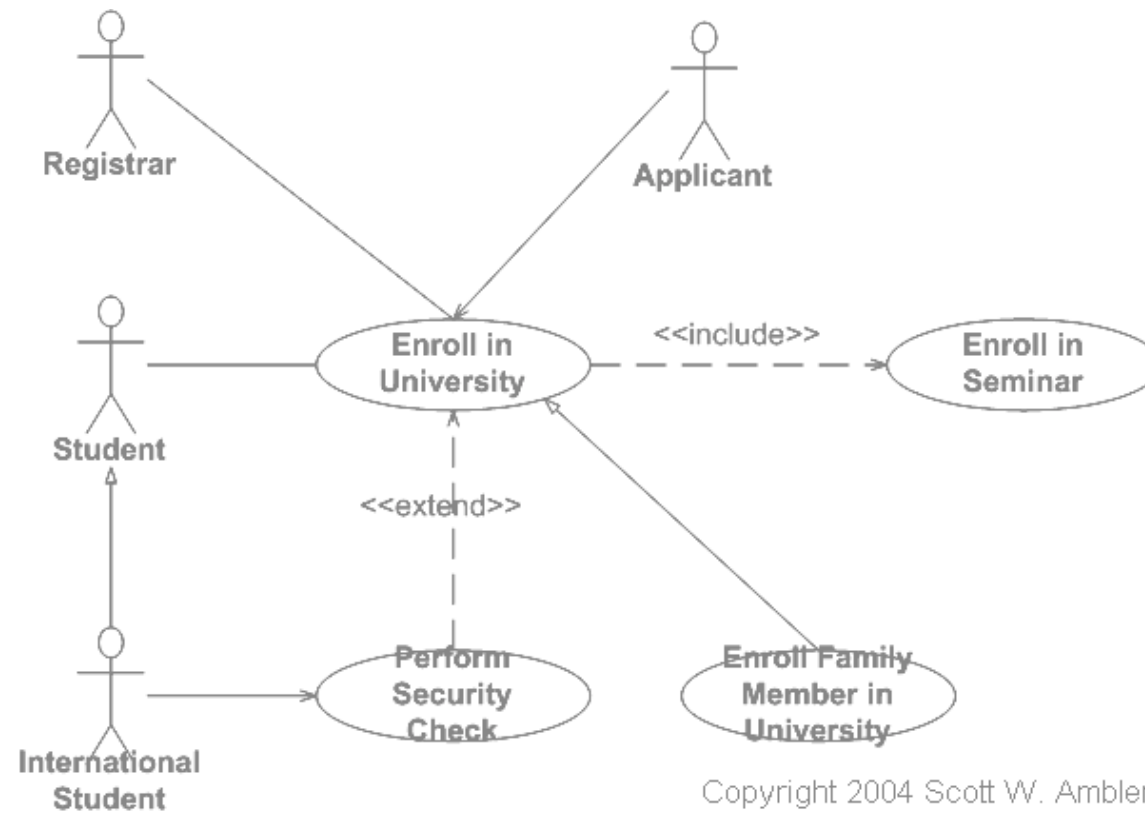
UNIVERSITATEA
DIN BUCUREȘTI

Metode de dezvoltare software

Diagrame UML de cazuri de utilizare

21.03.2023

Alin Ștefănescu



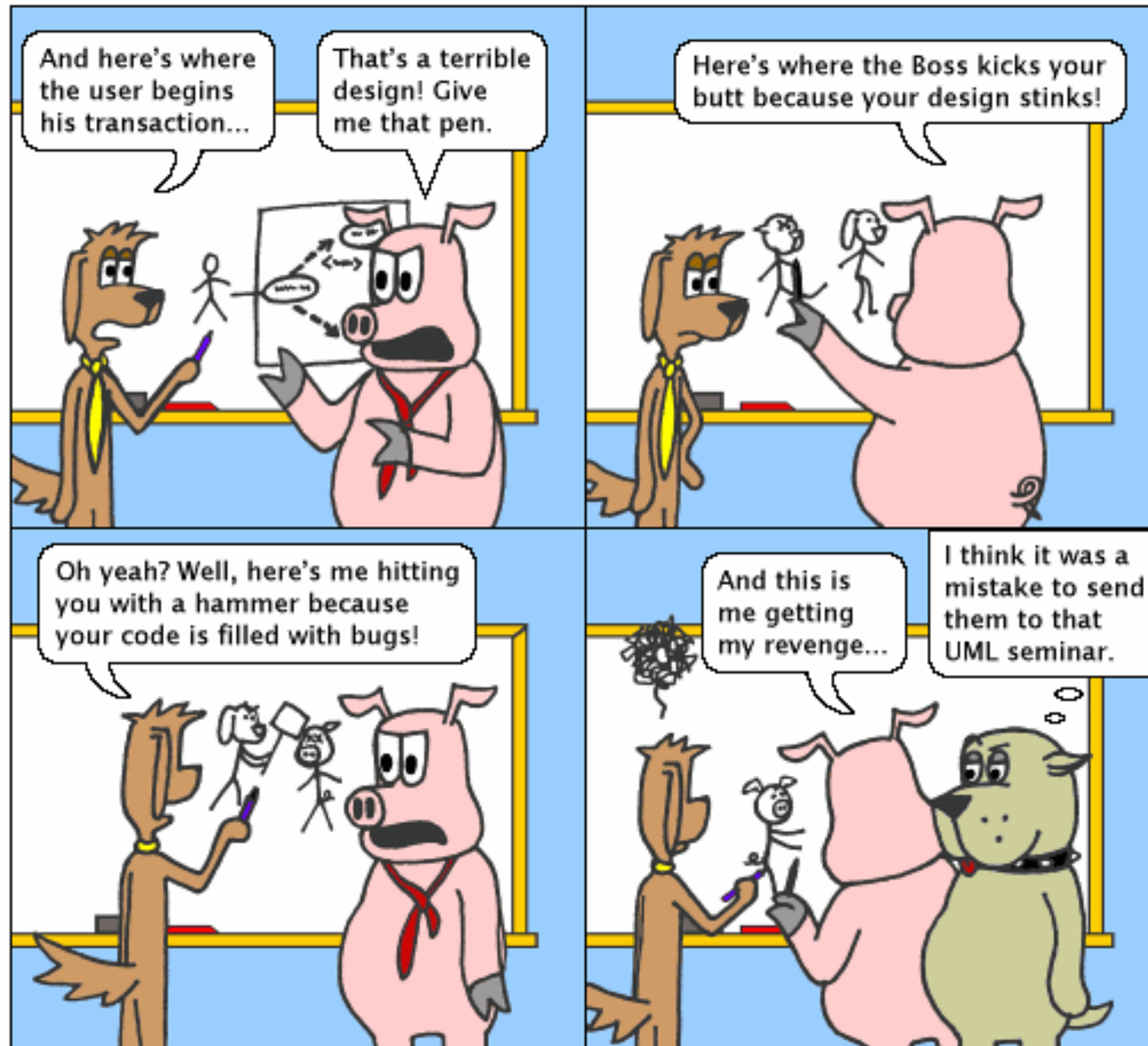
Copyright 2004 Scott W. Ambler

Cazuri de utilizare

Cazuri de utilizare... în practică

Hackles

By Drake Emko & Jen Brodzik



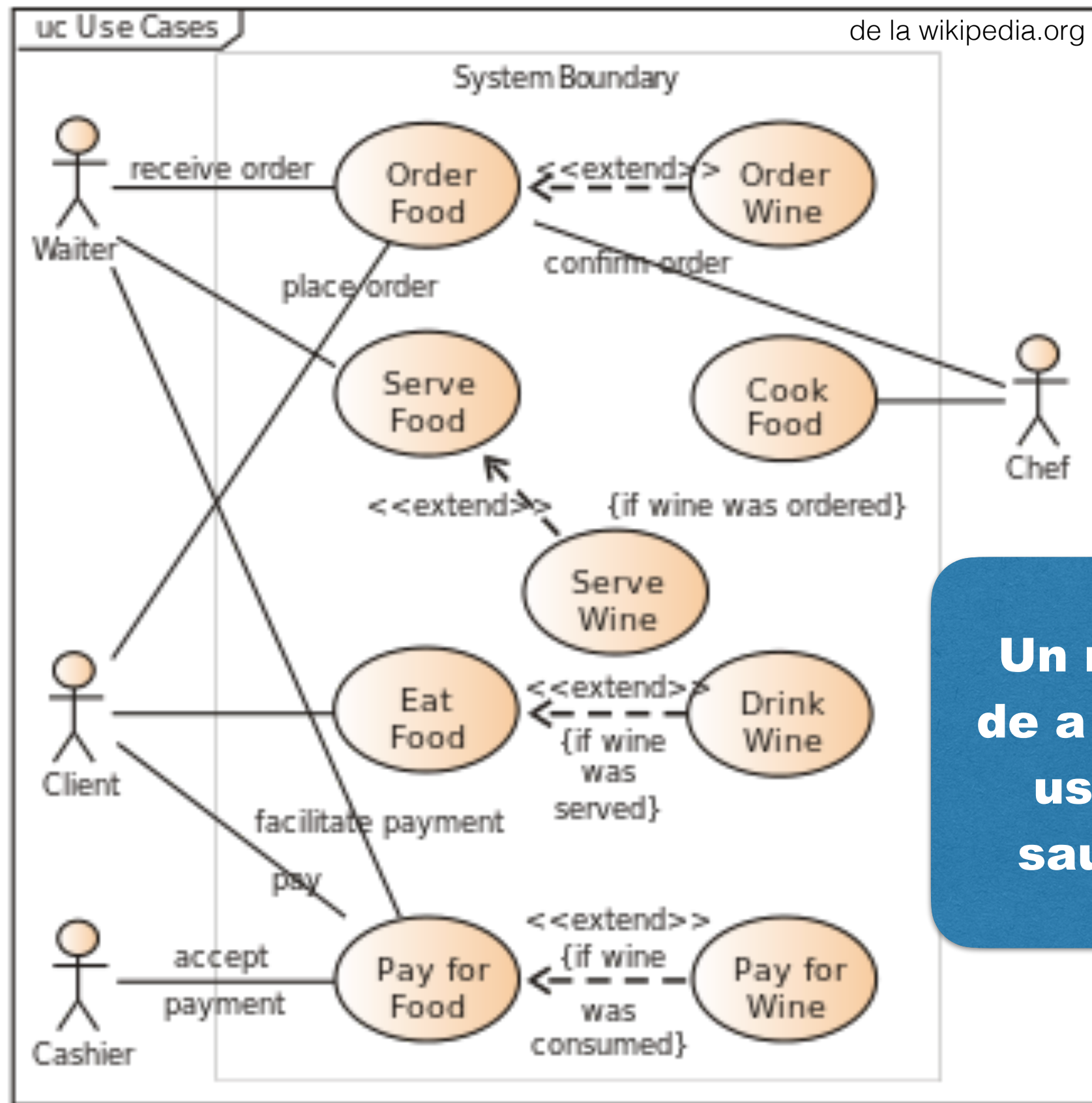
Cazuri de utilizare (Use Cases)

Introduse de Ivar Jacobson la începutul anilor '90.
Adoptate în standardul UML.

- Descriu comportamentul sistemului din punct de vedere al utilizatorului
- Două părți principale:
 - *sistem* (componente și descrierea acestora)
 - *utilizatori* (elemente externe)
- Cuprinde:
 - *diagrama* cazurilor de utilizare
 - *descrierea* cazurilor de utilizare

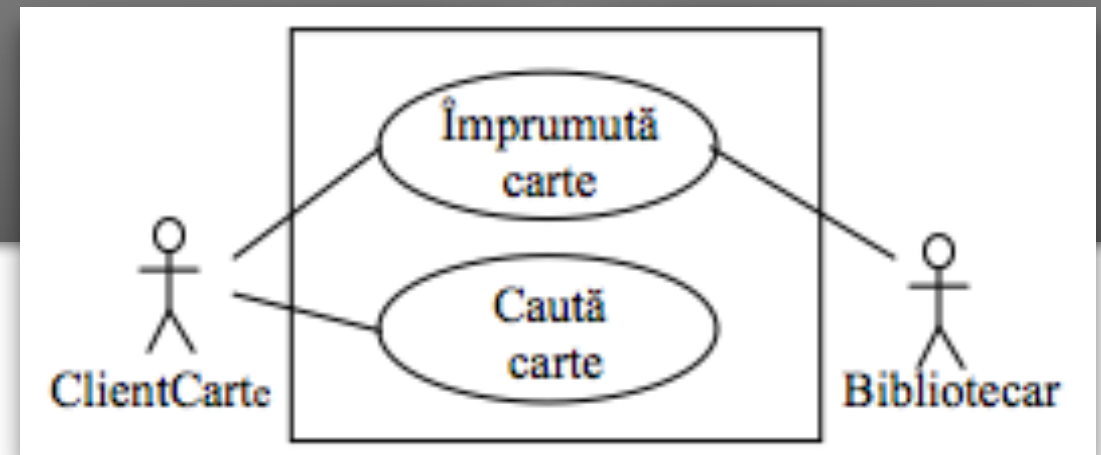


Diagrama cazurilor de utilizare - un exemplu



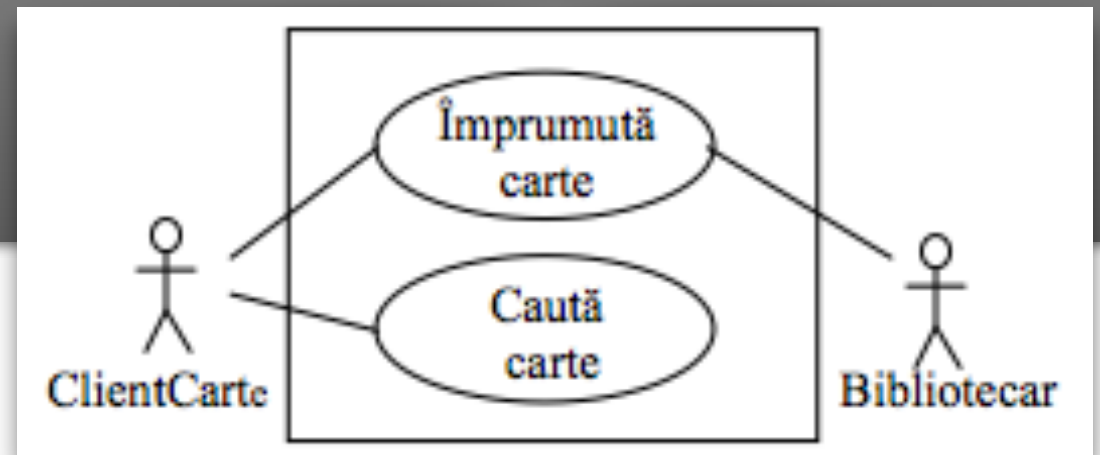
Un mod **vizual**
de a reprezenta
user stories
sau cerințele

Elementele principale



- **Caz de utilizare** (componentă a sistemului): unitate coerentă de funcționalitate sau task; reprezentată printr-un oval.
- **Actor** (utilizator al sistemului): element extern care interacționează cu sistemul; reprezentat printr-o figurină
- **Asociații de comunicare**: legături între actori și cazuri de utilizare; reprezentate prin linii solide
- **Descrierea cazurilor de utilizare**: un document (nativ) care descrie secvența evenimentelor pe care le execută un actor pentru a efectua un caz de utilizare

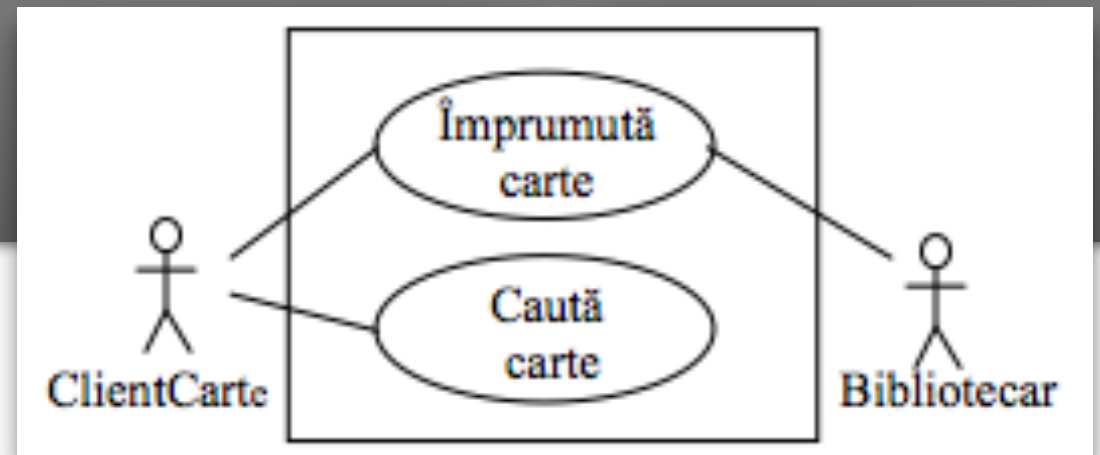
Actori



- Un actor reprezintă un **rol** jucat de un utilizator.
- Nu reprezintă un singur utilizator, ci o clasă de utilizatori.
- Același utilizator poate avea diferite roluri (d.ex. *Personal* sau *ClientCarte*), după cum un rol poate fi avut de mai mulți utilizatori.

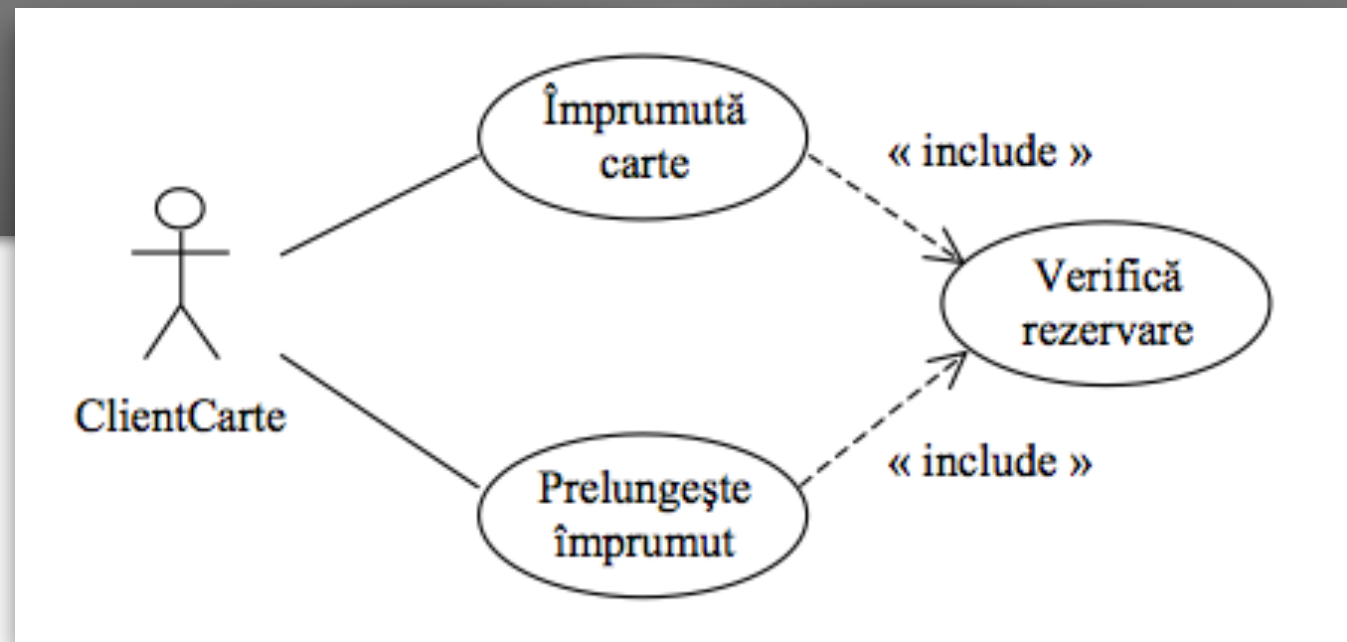
Identificarea actorilor = identificarea rolurilor avute de utilizatori în cadrul sistemului

Cazuri de utilizare



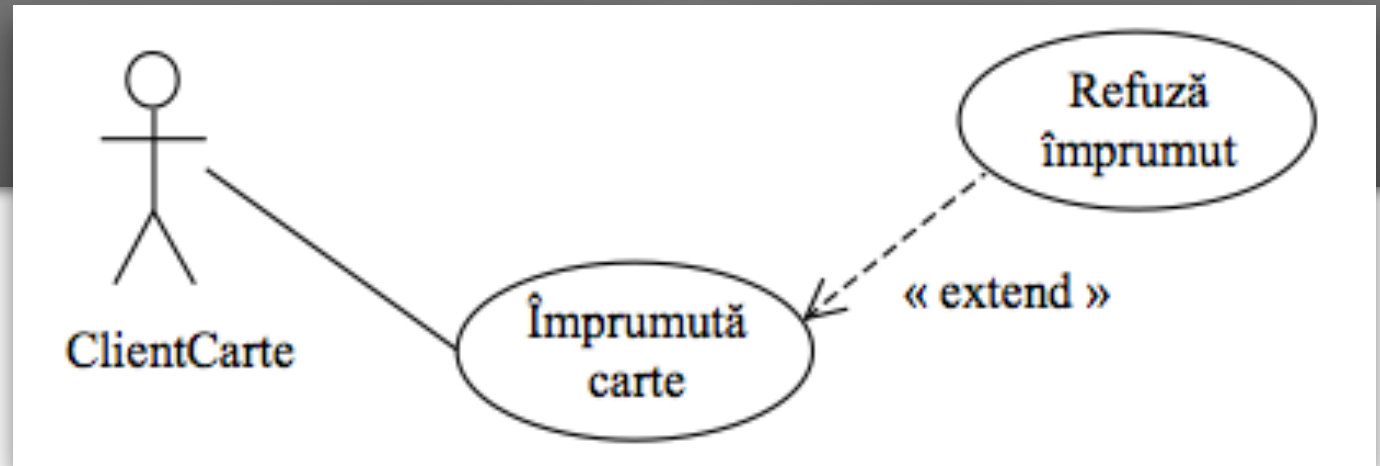
- Un caz de utilizare este o unitate coerentă de funcționalitate.
- Un caz de utilizare înglobează un set de cerințe ale sistemului care reies din specificațiile inițiale și sunt rafinate pe parcurs.
- Cazurile de utilizare pot avea complexități diferite; de exemplu “Împrumută carte” și “Caută carte”.

Relația « include »



- Dacă două sau mai multe cazuri de utilizare au o componentă comună, aceasta poate fi reutilizată la definirea fiecăruia dintre ele.
- În acest caz, componenta refolosită este reprezentată tot printr-un caz de utilizare legat prin relația « **include** » de fiecare dintre cazurile de utilizare de bază.
- Practic, relația « include » arată că secvența de evenimente descrisă în cazul de utilizare inclus se găsește și în secvența de evenimente a cazului de utilizare de bază.

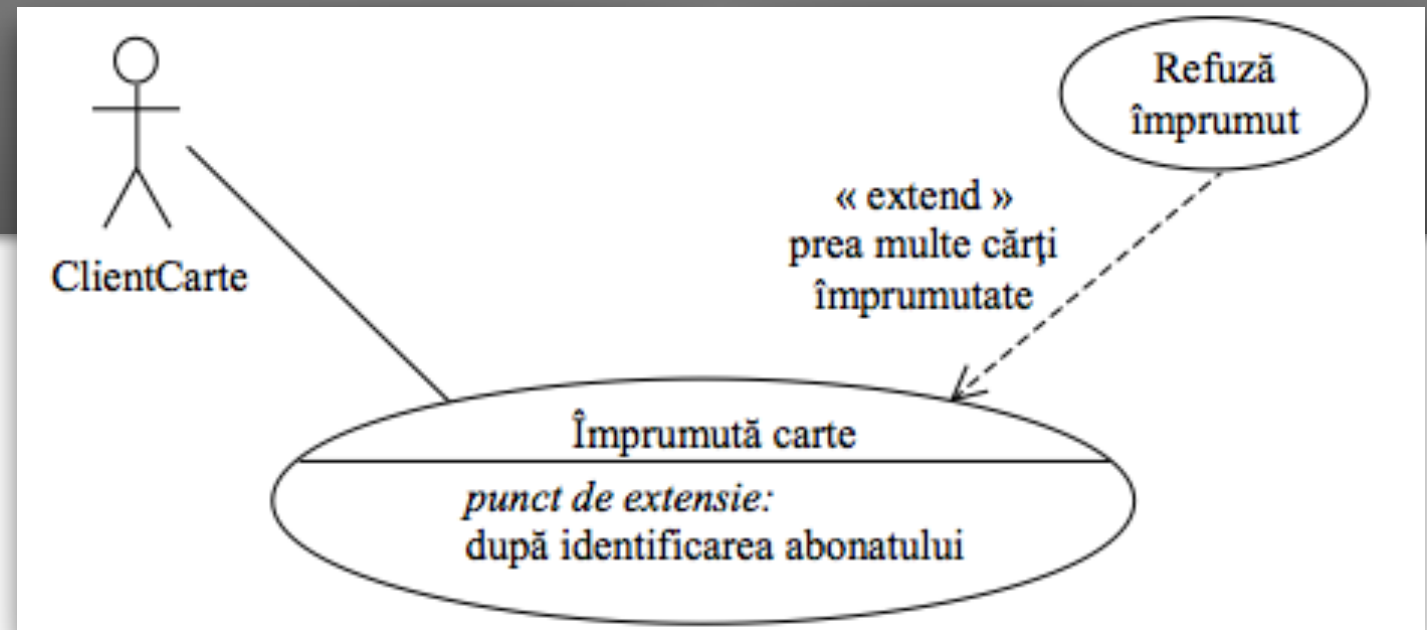
Relația « extend »



Relația « **extend** » se folosește pentru separarea diferitelor comportamente ale cazurilor de utilizare.

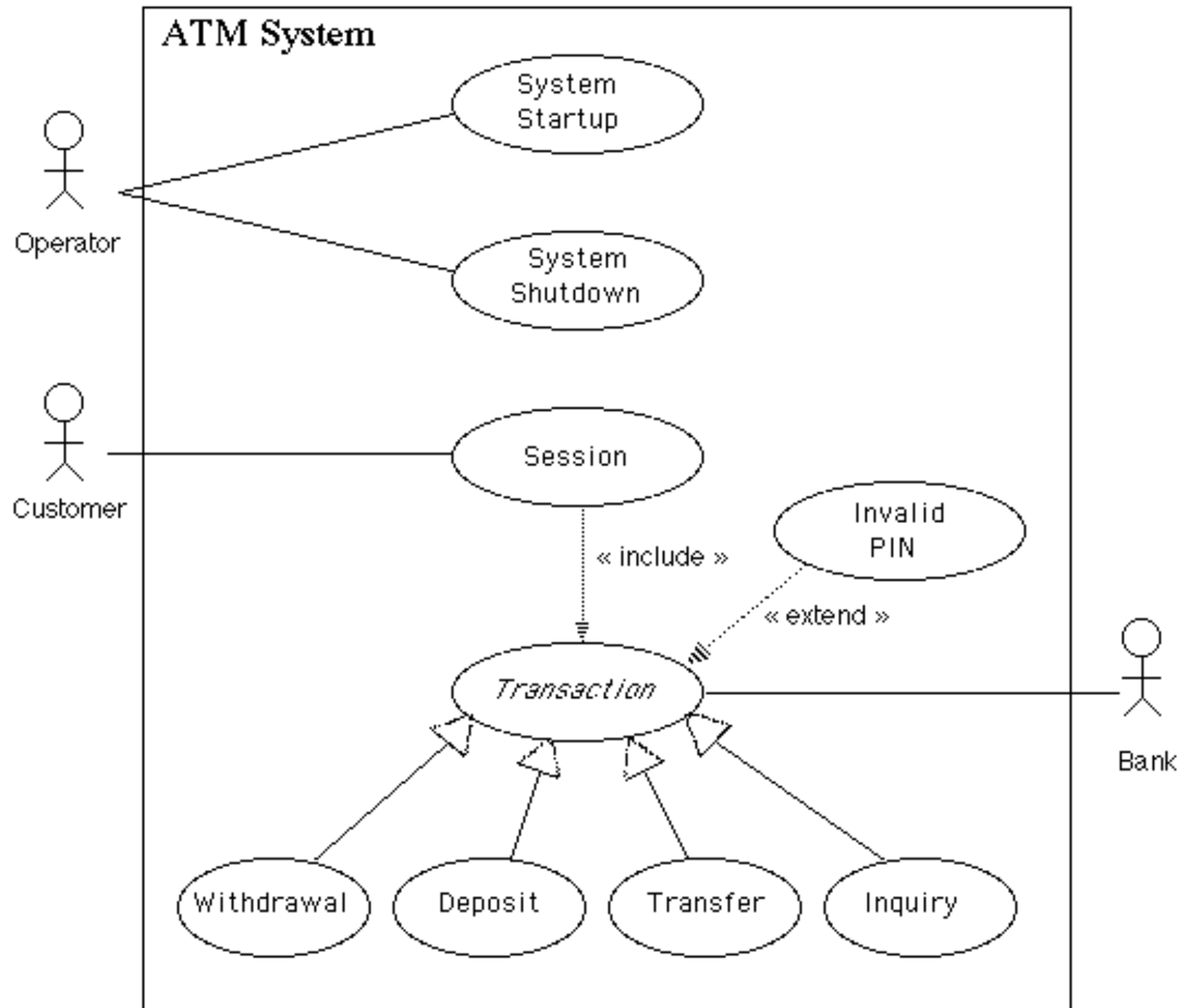
- Dacă un caz de utilizare conține două sau mai multe scenarii semnificativ diferite (în sensul că se pot întâmpla diferite lucruri în funcție de anumite circumstanțe), acestea se pot reprezenta ca un caz de utilizare principal și unul sau mai multe cazuri de utilizare excepționale.

Relația « extend »



- De obicei, se folosește pentru a pune în evidență **excepțiile**. D.ex. putem separa cazul de utilizare "Împrumută carte" într-un caz de utilizare frecvent întâlnit (principal), în care utilizatorului îi este permis să împrumute o carte și un caz de utilizare mai rar întâlnit (excepțional), în care utilizatorului îi este refuzat împrumutul deoarece depășește numărul maxim de cărți permise.
- Se poate specifica și "punctul de extensie"
- Atenție și la direcția săgeții (care arată spre cazul principal)

Încă un exemplu



Descriere la:

<http://www.math-cs.gordon.edu/courses/cs211/ATMExample/UseCases.html>

Atenție!

- **scopul principal** al acestui tip de diagrame este de a arăta **relația dintre actori și cazuri de utilizare**, nu relația dintre cazuri de utilizare (deși și aceasta este posibil prin relațiile *include* sau *extend* sau *generalizare*)
- nu se arată ordinea diferitelor cazuri de utilizare (aceasta se poate exprima prin alte diagrame - de stări sau de secvență).
- deci **NU există linii simple între cazuri de utilizare!**
(am observat această greșeală făcută de studenți destul de frecvent în decursul timpului)

Importanța cazurilor de utilizare

- Cazurile de utilizare descriu funcționalitatea sistemului; adică modul lui de folosire perceput de utilizatori (actorii externi).
- Scopul final al sistemului este de a realiza funcționalitatea descrisă în modelul cazurilor de utilizare (alături de cerințele nefuncționale).
- Cazurile de utilizare sunt folosite pentru:
 - analiză: identifică funcționalitatea cerută și o validează împreună cu clienții
 - design și implementare: trebuie realizate
 - testare: verifică sistemul; ele devin baza pentru generare de cazuri de testare.