2009 - **2010**

Inginerie Software pentru Comunicatii (ISC / RST)

Titular curs: **Eduard-Cristian Popovici**

Suport curs: *http://discipline.elcom.pub.ro/isc/*

Moodle: *http://electronica07.curs.ncit.pub.ro/course/category.php?id=4*

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

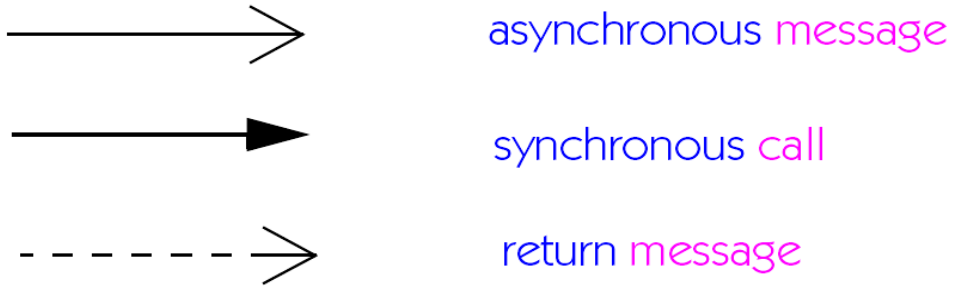
4. Diagrame UML dinamice

4.2. Diagrame UML de comunicatie si de robustete

Diagrame UML 2 de comunicatie
(de colaborare, in UML 1)

Diagramele de comunicatie

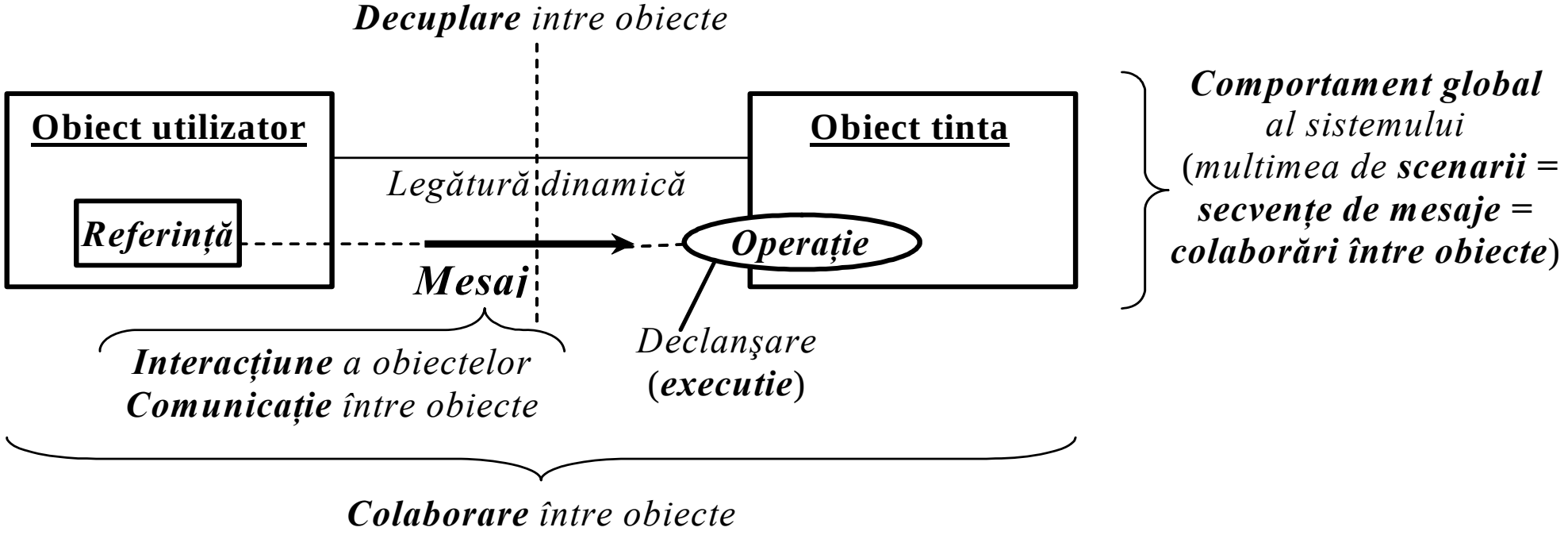
- au fost numite diagrame **“de colaborare” inainte de UML 2.0**
- sunt o **extensie a diagramelor de obiecte**
- prezinta **interactiunile intre obiecte (comunicatiile)**
  - prin reprezentarea **trimiterilor de mesaje** (semnaturi operatii)
    - in plus fata de diagramele de obiecte



  - motiv pentru care se numesc acum **de comunicatie**
  - si sunt incadrate in categoria diagramelor **de interactiune**

Diagramele de comunicatie

- insista pe **structura statica** si **spatiala** (obiectele si legaturile lor)
  - astfel pot exprima **contextul grupului de obiecte**
  - si pot evidentia **colaborarea** intr-un **grup de obiecte**
    - motiv pentru care au fost denumite de **colaborare**



Diagramele de comunicatie

Obiectele se reprezinta ca si in diagramele de obiecte

- **complet** - prin nume si numele clasei — **numeObiect : NumeClasa**
- **doar prin nume** - fara a specifica tipul — **numeObiect**
  - pentru o modelare incompleta in care clasa nu a fost inca decisa
- "**anonime**" - doar prin numele clasei — **:NumeClasa**
  - pentru a evita introducerea de nume inutile in diagrame
  - permitand exprimarea general valabila pentru mai multe obiecte

Numele clasei poate contine calea completa (numele **calificat**)

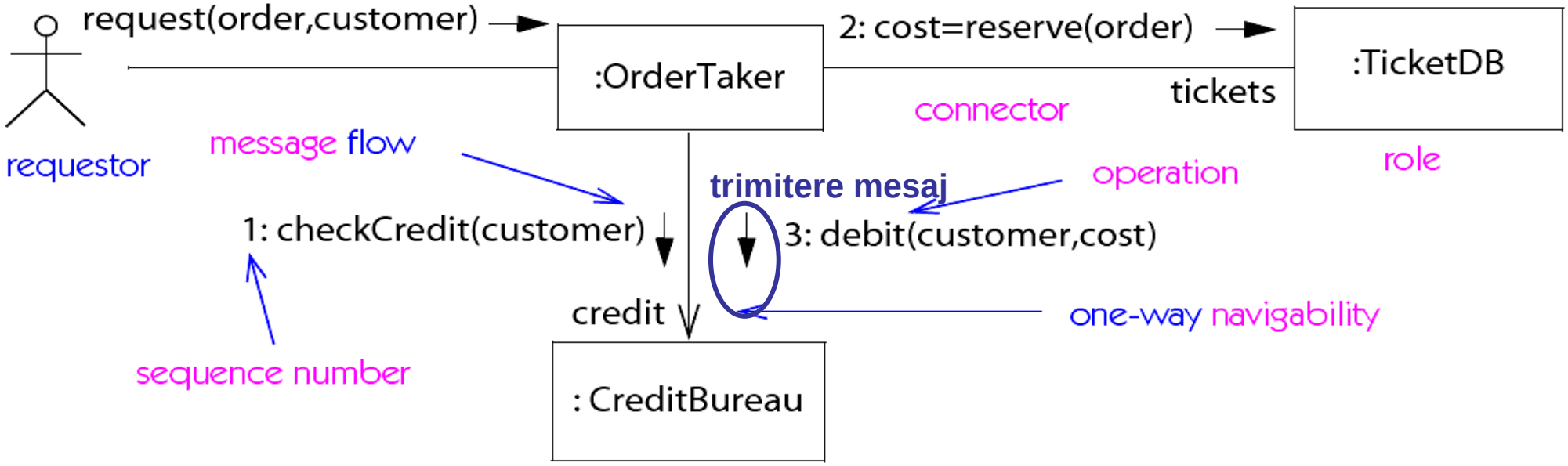
- compusa din **numele pachetelor care o inglobeaza**, separate prin " **::** "

ButonOK : InterfataUtilizator :: Controale :: ButonSonerie

Diagramele de comunicatie

Trimiterea mesajelor

- se reprezinta cu **sageti indicand apelul unei operatii**
  - **de-a lungul legaturilor** care leaga obiecte
  - indreptate **spre destinatarul mesajului**

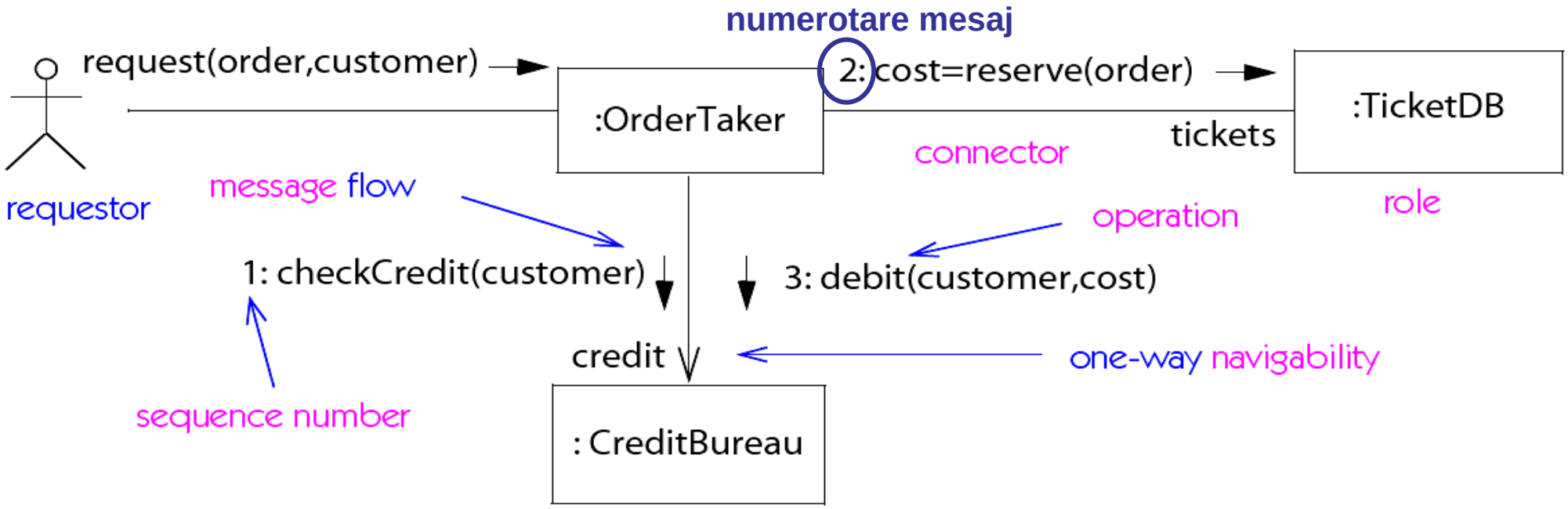


Diagramele de comunicatie

Intr-o diagrama de comunicatie timpul nu este reprezentat **implicit** (ca in diagrama de secventa**)**

- **ordinea trimiterii** mesajelor se specifica prin **numerotare** explicita

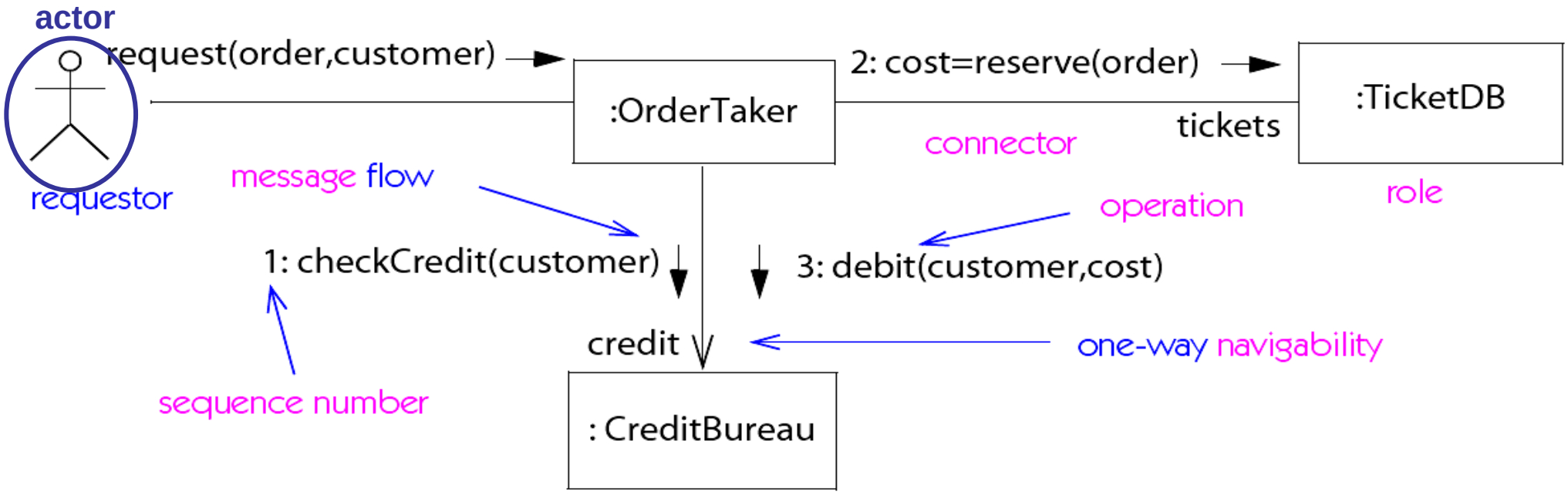
numarSecventa : valoareReturnata := mesaj (argumente)



Diagramele de comunicatie

Notatia permite

- **figurarea unui actor** in diagrama de comunicatie pentru a figura **declansarea interactiunilor de catre un element extern**
- astfel incat **interactiunea poate fi descrisa in mod mai abstract**



Diagramele de comunicatie

Trimiterile <u>conditionate</u> de mesaje

- modeleaza **structurile de control alternative** (if..else, switch..case)

[condition-clause]

- textul conditiilor putand fi in **pseud-cod**, **OCL** (*Object Constraint Language*), etc.

[x > y]

Structurile de control <u>iterative</u> (for, while, do…while) pot fi modelate utilizand simbolul " * "

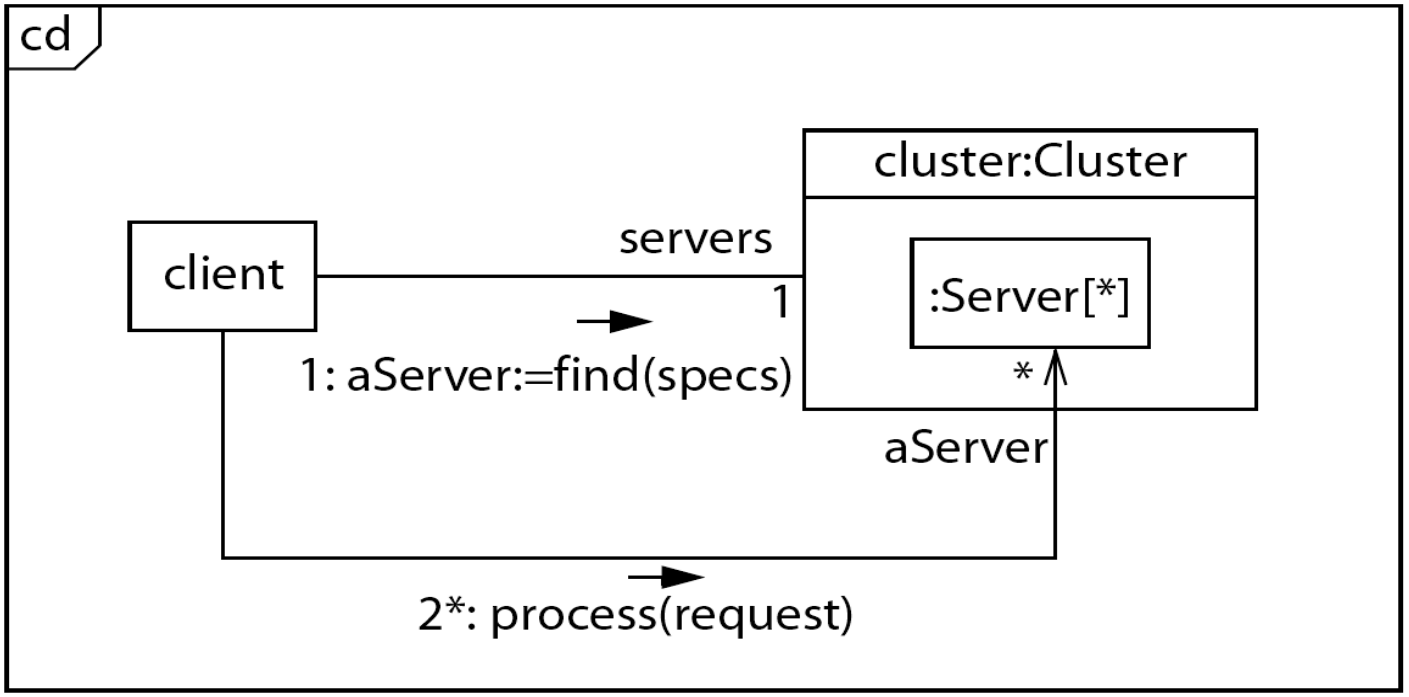
* [iteration-clause]

2: display (x, y)	Simple message
1.3.1 : p= find(specs):status	Nested call with return value
1b.4 [x < 0] : invert (x, color)	Conditional within second concurrent thread
3.1*[i:=1..n]: update ( )	Iteration

Diagramele de comunicatie – operatii pe Colectii

Operatiile pe Colectii (structuri de **obiecte multiple**: liste, arbori, etc.)

- pot fi reprezentate **in UML 2** folosind **structuri composite**
- si **mesaje catre obiectele claselor interne**



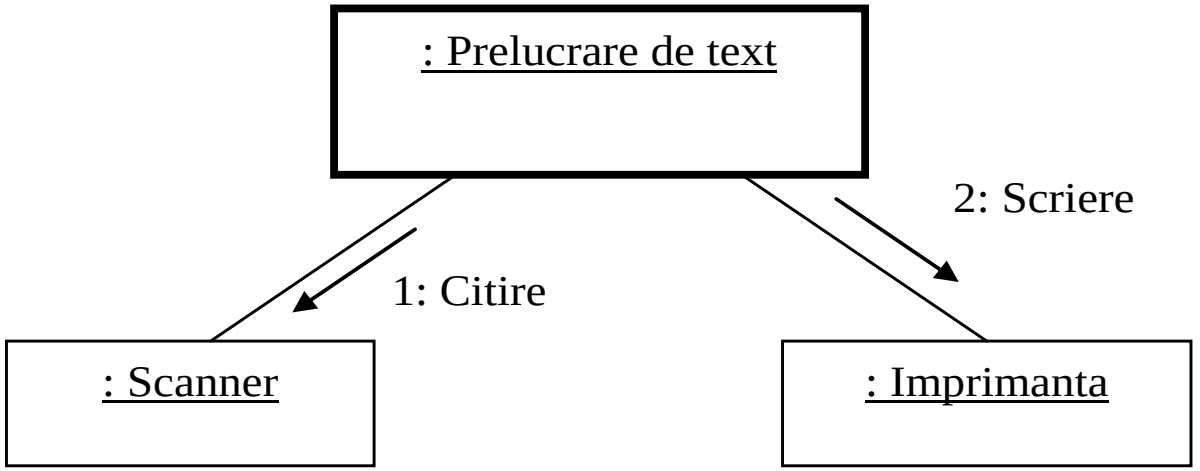
Diagramele de comunicatie – cu obiecte active si pasive

Obiectele care **poseda flux de control** sunt denumite **active**

Un obiect activ

- **poate activa** un **obiect pasiv** pe durata unei operatii, **trimitandu-i** acestuia **un mesaj**

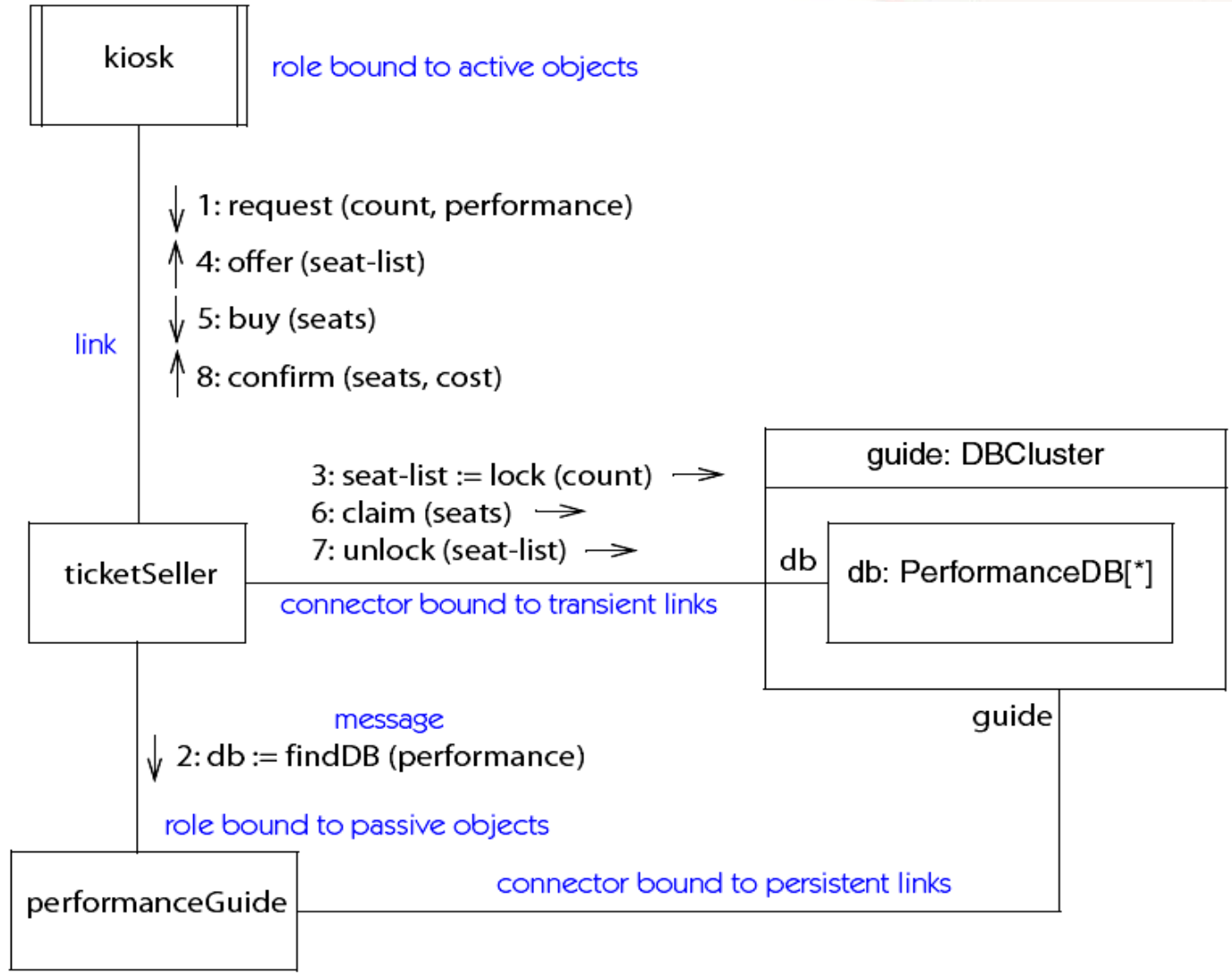
-se reprezinta printr-un **dreptunghi al carui chenar este mai ingrosat** decat cele ale obiectelor pasive



Odata mesajul tratat

- fluxul de control este **restituit obiectului activ**

Diagramele de comunicatie – cu obiecte active si pasive

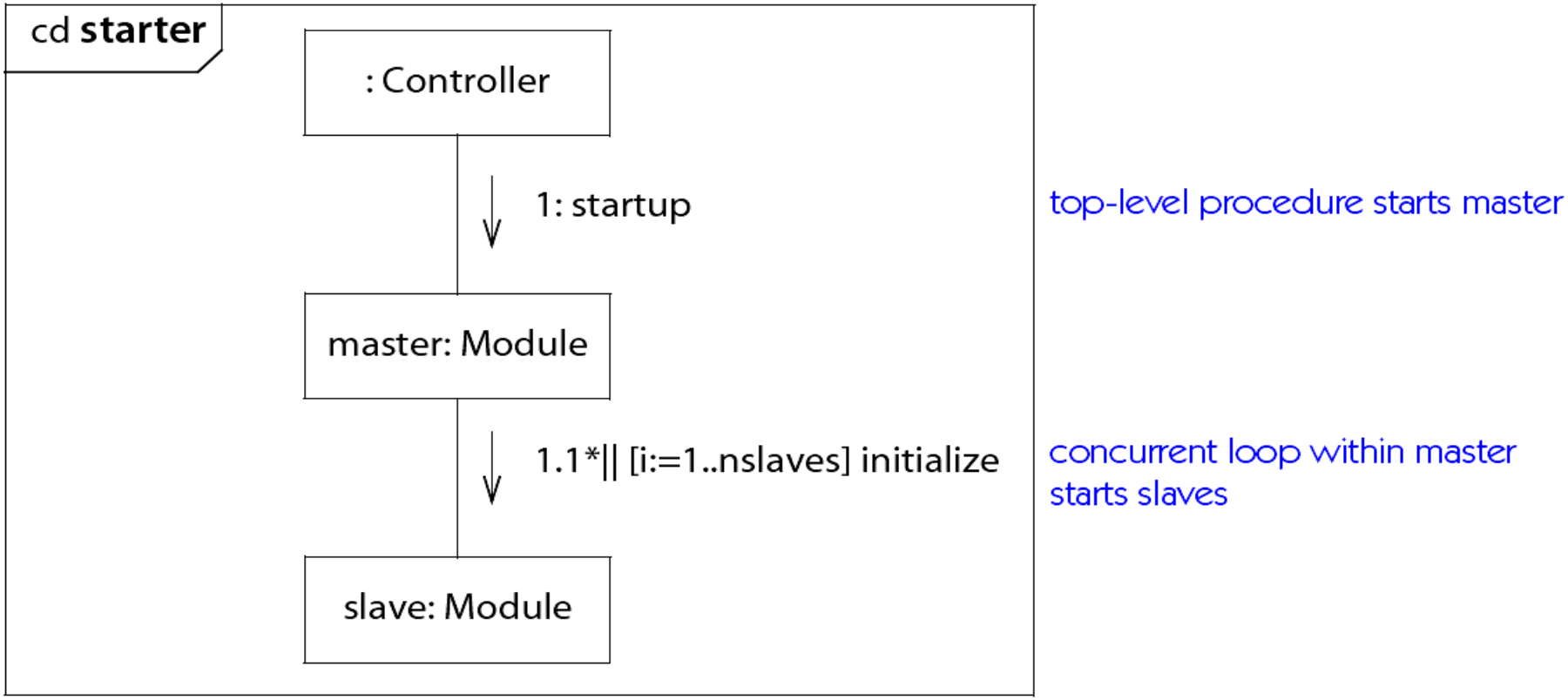


Diagramele de comunicatie

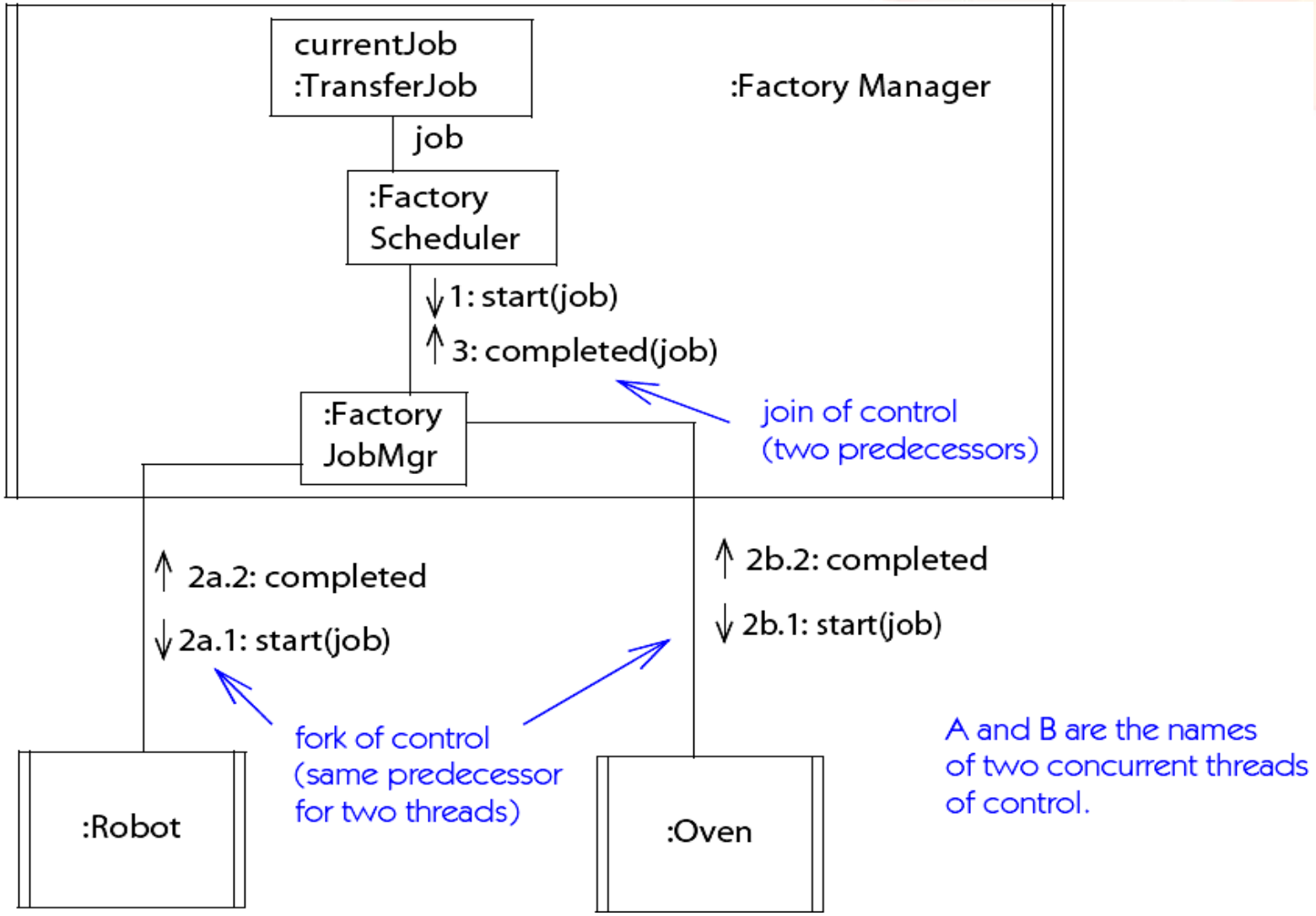
Intr-un mediu multiproces, **mai multe obiecte** pot fi **active simultan**

Paralelismul trimiterii mesajelor

- poate fi **figurat cu** simbolul " **||** ": *[i:=1..n]|| q[i].calculateScore ()

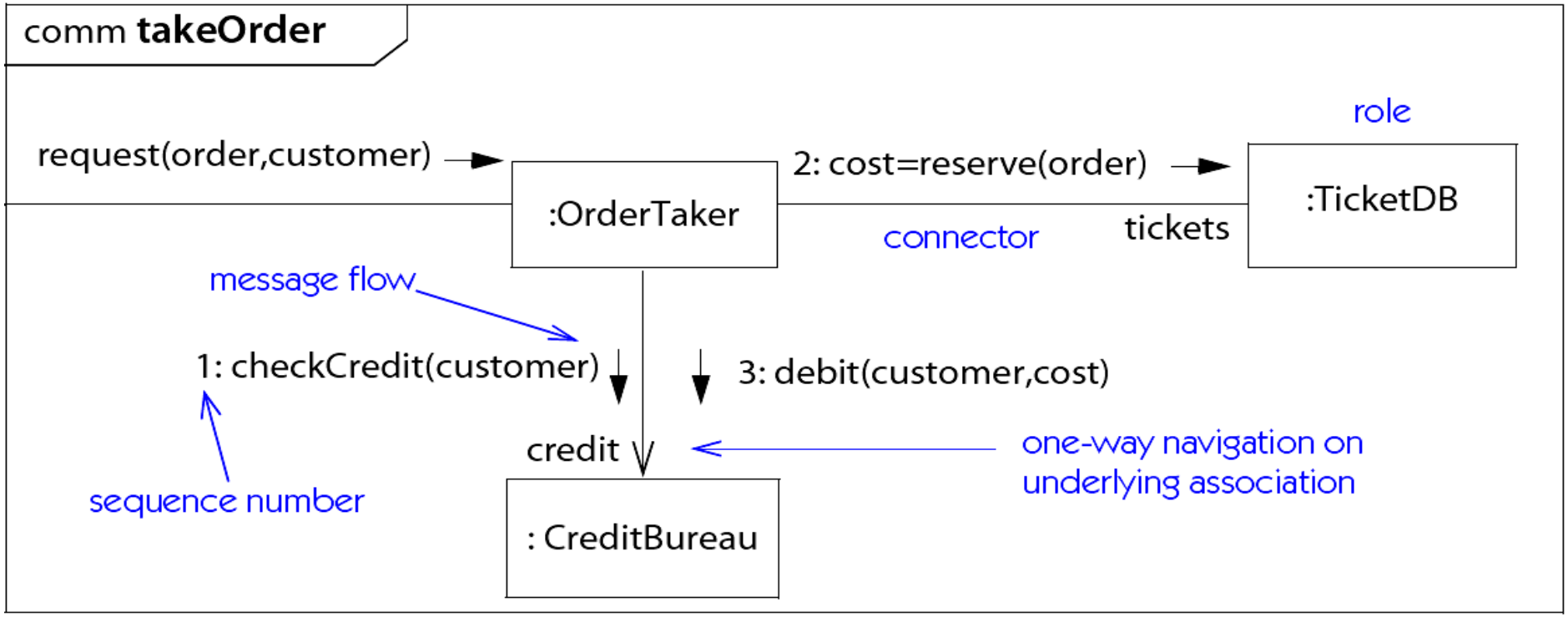


Diagramele de comunicatie - cu fire de control concurente



Diagramele de comunicatie

Diagramele de comunicatie sunt marcate cu tag-ul comm sau cu tag-ul cd



4.2. Diagrame UML de comunicatie si de robustete

Tag-urile folosite pentru marcarea diferitelor diagrame UML

Tag	*Name*
activity	activity diagram
class	class diagram
comm	communication diagram
component	component diagram
class	composite structure diagram
deployment	deployment diagram
intover	interaction overview diagram
object	object diagram
package	package diagram
state machine	state machine diagram
sd	sequence diagram
timing	timing diagram
use case	use case diagram

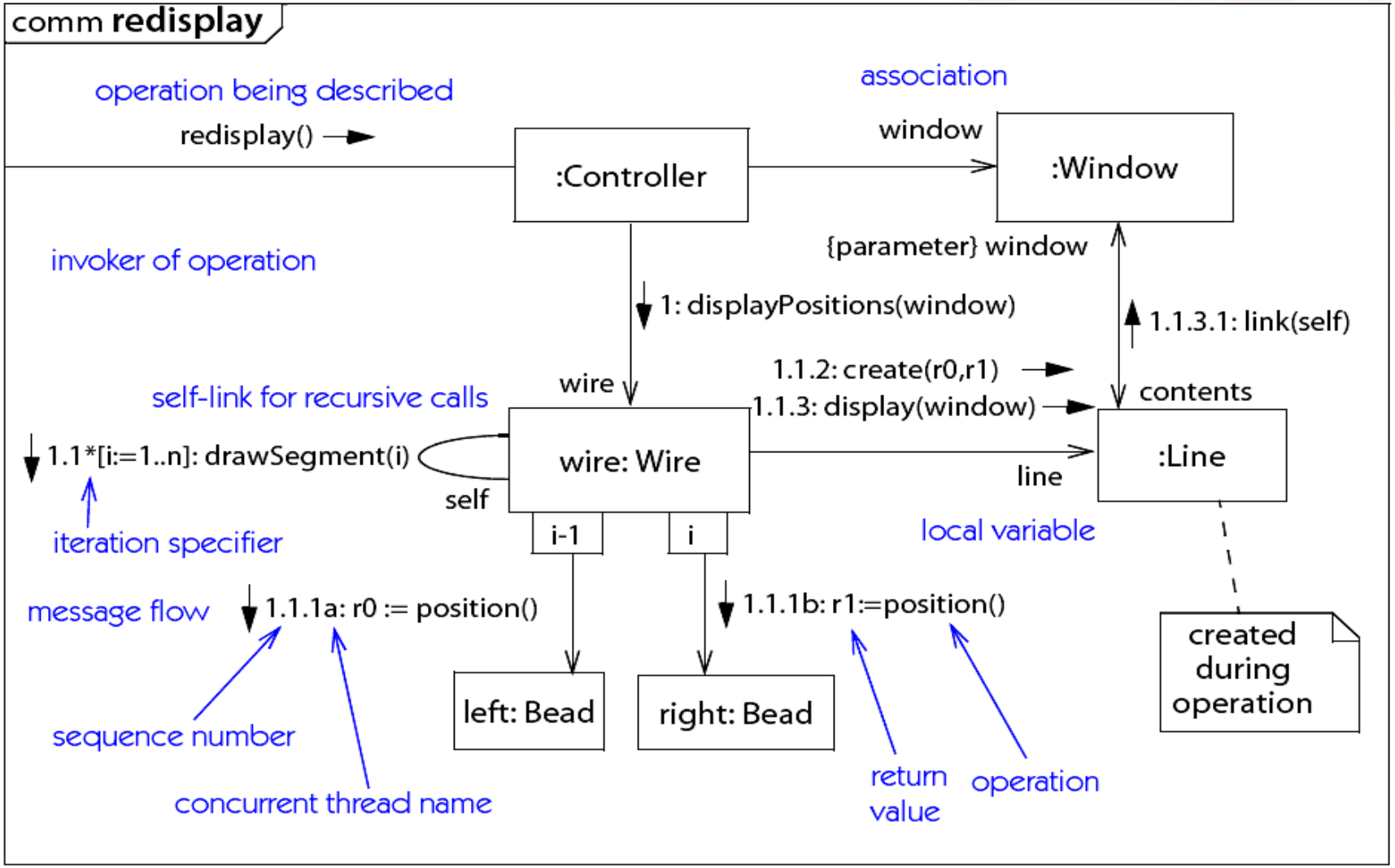
Pe langa acestea, exista si tag-ul **ref**

folosit **pentru a referi o diagrama de interactiune**

(comunicatie sau secventa)

definita separat

Diagramele de comunicatie – descrierea unei operatii complexe



Diagramele de robustete

Analiza robustetii (*robustness analysis*)

Analiza are ca tinta -> constructia **sistemului potrivit**

Proiectarea (design) are ca tinta -> **constructia potrivita** a sistemului

Proiectarea preliminara

- numita si **analiza a robustetii** (*robustness analysis*)
- este un **pas intermediar** intre analiza si proiectare
- realizeaza o **proiectare exploratorie** necesara pentru
  - a intelege complet cerintele,
  - rafinandu-le si eliminand ambiguitatile din ele, si
  - **legand mai bine cerintele comportamentale** (*Use Cases*) **de obiecte** (*Entities / Objects Domain Model*)

Analiza robustetii (*robustness analysis*)

Analiza robustetii inseamna

- crearea unei **diagrame de robustete** care este
  - o **reprezentare vizuala a pasilor dintr-un UC**
- pe parcurs **rescriind textul UC**

Analiza robustetii ajuta la **umplerea golului** dintre

- analiza (UC) si

- proiectarea initiala

prin **legarea UC de obiecte**

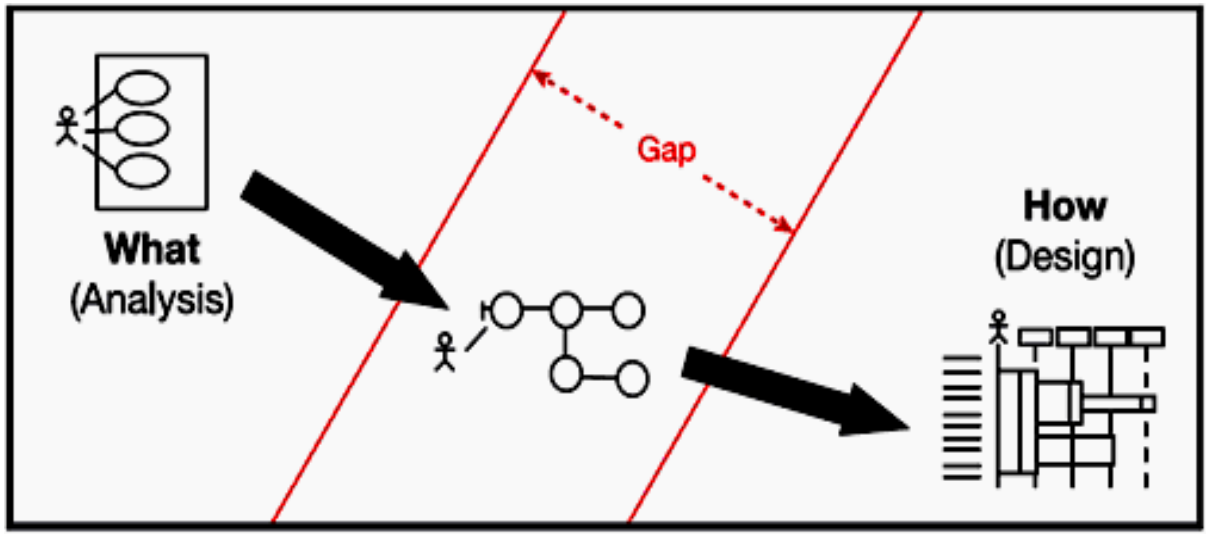


Diagrama de robustete (*robustness diagram*)

- este un fel de **hibrid** intre
  - **diagramele de clase** (schemele bloc) si
  - **diagramele de activitati** (organigramele)

  (sau mai degraba o **versiune a diagramelor de comunicatie**)

- care **reprezinta vizual comportamentul descris de UC**
  - aratand atat **clasele participante**
  - cat si **activitatile desfasurate**

- desi in mod intentionat **evita sa arate pentru fiecare** element de **comportament**
  - care sunt **clasele responsabile**

Diagrama de robustete (*robustness diagram*)

Stereotipurile de clase folosite **in diagramele de robustete** sunt

- clasele "**entitate**" (*entity object* / ***M**odel*)
  - formeaza **modelul domeniului**

- clasele "**interfata**" (*boundary object* / ***V**iew*)
  - **intre lumea exterioara** si **interiorul sistemului**

    (ecrane, pagini, senzori, etc)

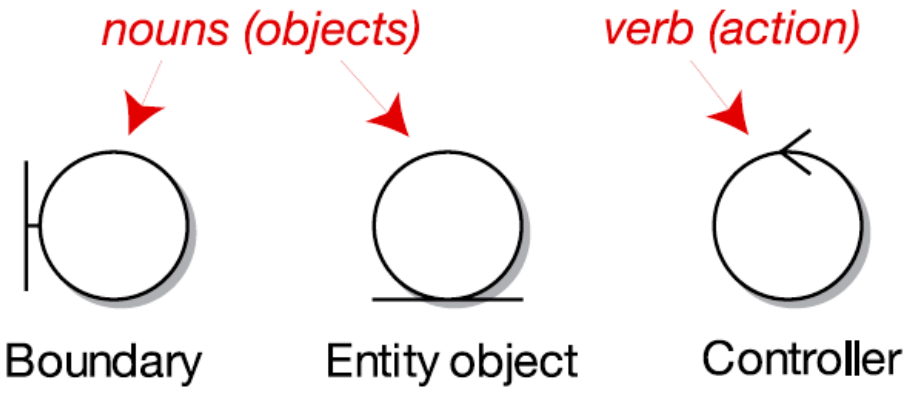
- clasele "**controler**" (*controller* / ***C**ontroller*)
  - **imbina** (*glue*) clasele ***boundary*** si clasele ***entity***

Diagrama de robustete (*robustness diagram*)

In analiza robustetii

- obiectele **interfata** (*boundary*) si **entitate** sunt considerate ca fiind **substantive** (*nouns*) iar

- **controlerele** sunt considerate ca fiind **verbe**



Exista urmatoarele **reguli pentru crearea diagramelor de robustete**

- **substantivele pot fi legate de verbe** (si **reciproc**)
- **substantivele NU pot fi legate de** alte **substantive**
- **verbele pot fi legate de** alte **verbe**

Diagrama de robustete (*robustness diagram*)

Aceasta **regula** va fi folosita pentru a **impune in textele UC** o schema

substantiv – verb – substantiv

Astfel, **analiza robustetii** permite **imbunatatirea textelor UC**

Exercitiu

Doar o parte dintre urmatoarele **constructii** sunt **permise** de regula anterioara.

a. Boundary ➢ Controller ➢ Entity

b. Entity ➢ Entity

c. Controller ➢ Controller

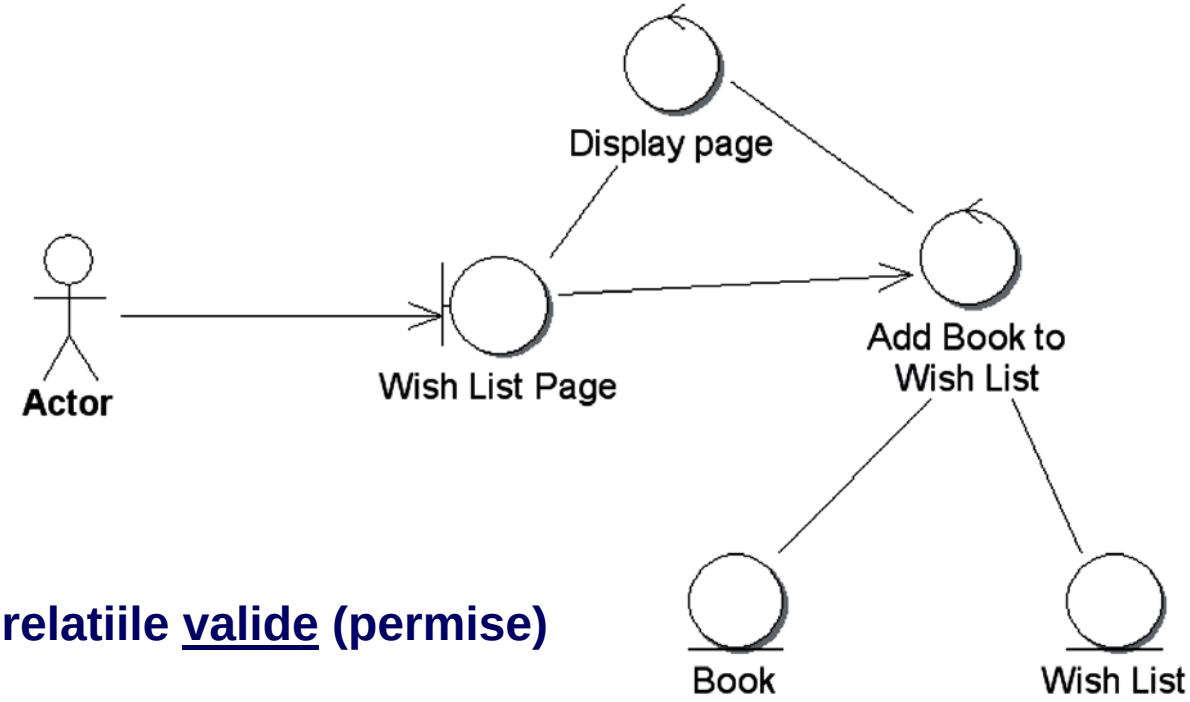
d. Boundary ➢ Boundary ➢ Controller

Care dintre ele sunt permise?

Diagrama de robustete (*robustness diagram*)

Toate relatiile valide posibile (incluzandu-I pe actori) sunt urmatoarele

- un **actor** poate fi **legat** de o **clasa interfata** (*boundary object*)
- o **clasa interfata** poate fi **legata** de o **clasa controler**
- o **clasa controler** poate fi **legata** de o **clasa controler**
- o **clasa controler** poate fi **legata** de o **clasa entitate** si reciproc

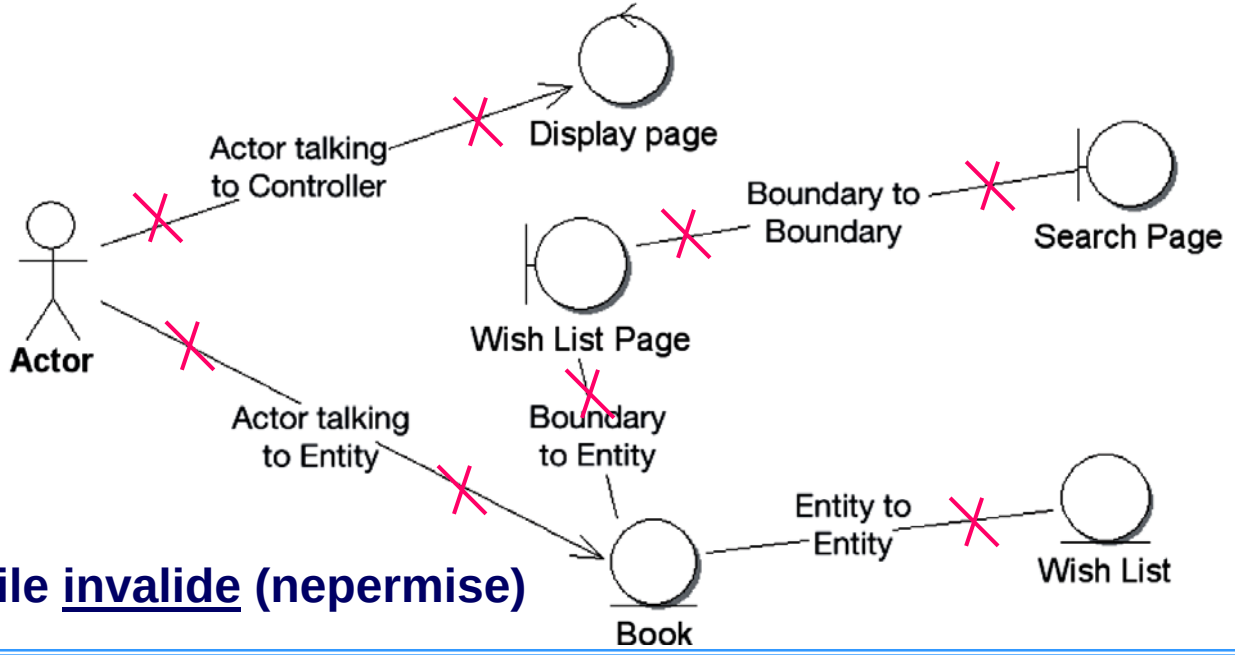


relatiile valide (permise)

Diagrama de robustete (*robustness diagram*)

Toate relatiile invalide posibile sunt urmatoarele

- un **actor NU** poate fi legat de o **clasa controler** sau **entitate** (trebuie **legat** de o **clasa interfata**)
- o **clasa interfata NU** poate fi legata de o **clasa entitate**
- o **clasa interfata NU** poate fi legata de o alta **clasa interfata**
- o **clasa entitate NU** poate fi legata de o **clasa entitate**



relatiile invalide (nepermise)

Analiza robustetii (*robustness analysis*)

Analiza robustetii - **activitati**

Pasul 1

- copierea (*copy-paste*) **textului UC** direct **in diagrama robustetii**

Pasul 2

- copierea claselor entitate din **modelul domeniului** si

- adaugarea elementelor lipsa

- clase entitate, clase interfata si **controlere**

Acolo unde este cazul se recurge la

- rescrierea textelor UC pe masura crearii diagramei robustetii (dezambiguarea textelor)

- crearea unei clase interfata pentru fiecare ecran / pagina si

- denumirea lor **fara ambiguitati**

Analiza robustetii (*robustness analysis*)

De la UC la clasele din diagramele de robustete

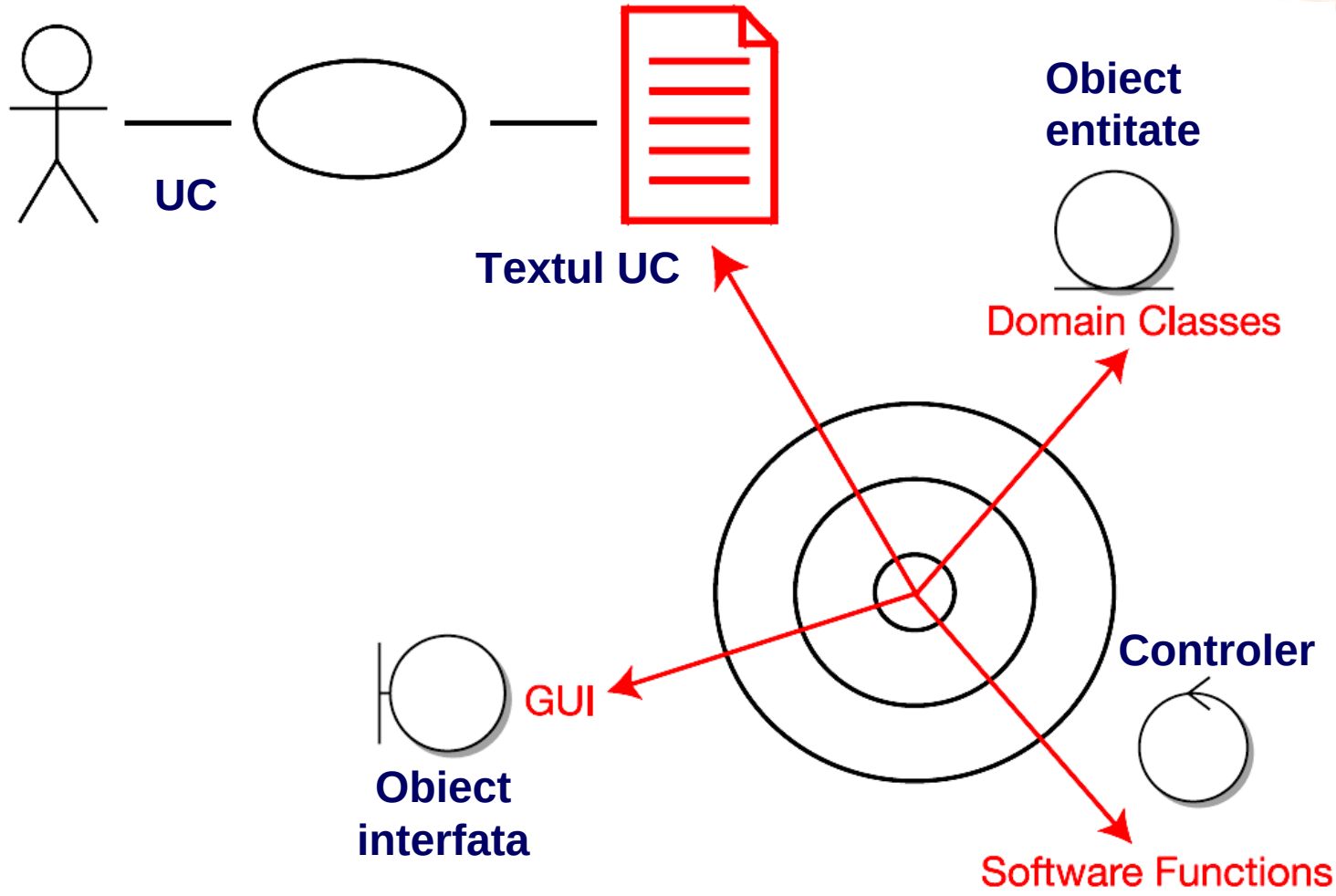


Diagrama de robustete (*robustness diagram*)

Analiza robustetii - exemplu

***Show Book Details* UC**

BASIC COURSE:
The Customer types in the URL for the bookstore's home page, which the system displays. Then the Customer clicks a link to view a Book. The system retrieves the Book details and displays the Book Details screen.

ALTERNATE COURSES:
Book not found: The system displays a Book Details Not Found screen.