

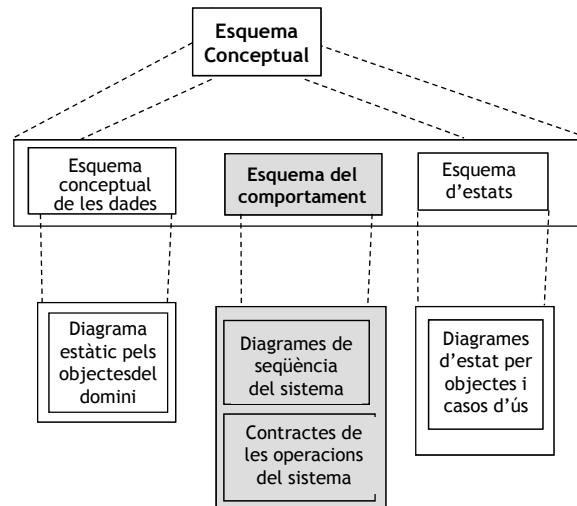
Especificació en UML: Esquema del comportament



Esquema del comportament en UML

- Especificació del comportament 'orientat a objectes'
- Diagrames de seqüència del sistema
- Contractes de les operacions del sistema
- Compartició d'informació entre operacions
- Redundància amb l'esquema conceptual de les dades
- Especificació en OCL de la sortida d'una operació
- Interacció amb actors software (serveis) externs
- Diagrama d'estats
- Bibliografia

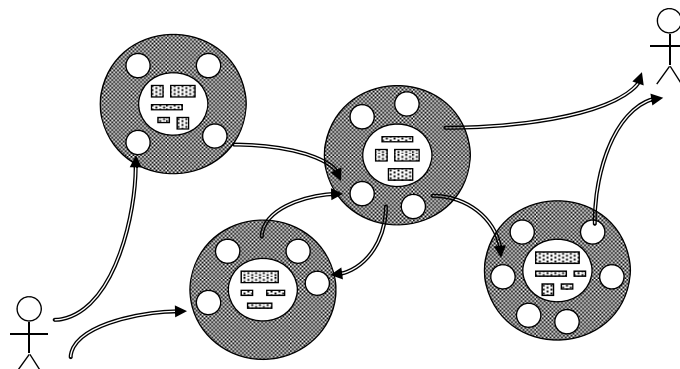
Esquema conceptual (especificació)



3

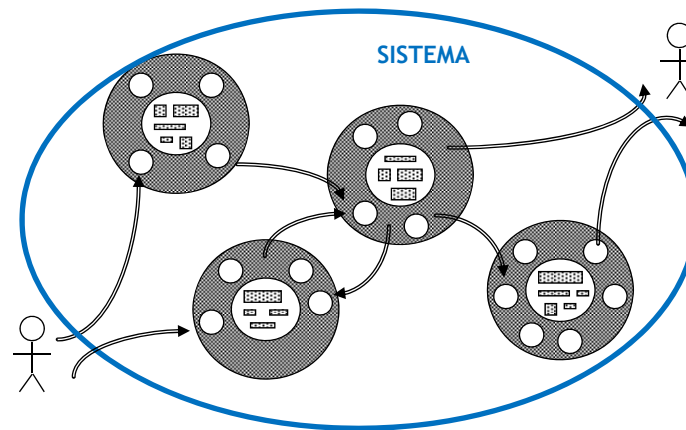
Descripció del comportament en OO

Els objectes es comuniquen mitjançant la invocació d'operacions d'altres objectes



4

“Especificació” del comportament en OO



Considerem un tipus especial “sistema” que engloba tots els objectes
L’especificació de la part dinàmica és l’esquema del comportament del “sistema”

5

Esquema del comportament del sistema

Diagrames de seqüència del sistema:

- Mostren la seqüència d’esdeveniments entre els actors i el sistema.
- Permeten identificar les operacions del sistema

Contractes de les operacions del sistema:

- Descriuen l’efecte de les operacions del sistema

6

Diagrames de seqüència del sistema

Objectius:

Construcció d'un diagrama que permeti identificar els esdeveniments externs i les operacions del sistema

Punt de partida:

Casos d'ús

Casos d'ús:

Descriuen la interacció entre els actors i el sistema software, com una seqüència d'esdeveniments externs rellevants per al sistema.

L'actor genera esdeveniments cap al sistema, que exigeixen l'execució d'alguna operació com a resposta.



Definirem un diagrama de seqüència del sistema
per cada curs rellevant d'esdeveniments d'un cas d'ús

7

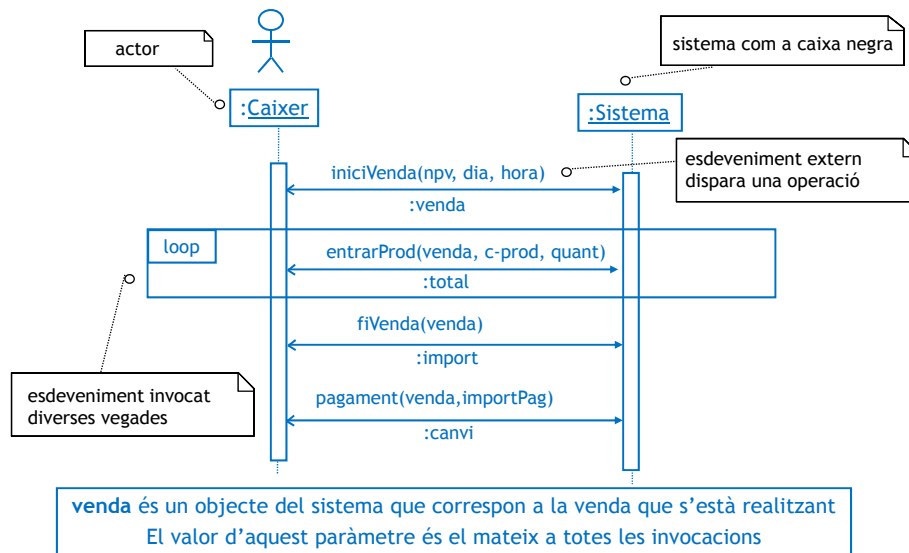
Exemple: cas d'ús Comprar Productes

escenari principal d'èxit

1. El Caixer indica al Sistema que comença una nova venda, amb el dia i l'hora actuals, en una caixa que té un número determinat
2. El Sistema enregistra el començament de la venda a la caixa
3. El Caixer entra al Sistema l'identificador de producte i la quantitat d'unitats
4. El Sistema enregistra la línia de venda corresponent al producte
5. El Sistema mostra el total acumulat
Es repeteixen els passos 3 a 6 fins que s'han processat tots els productes
6. El Caixer indica al Sistema l'acabament de la venda
7. El Sistema presenta el total de la venda
8. El Caixer comunica al Client el total de la venda
9. El Client entrega al Caixer una quantitat de diners, potser més gran que el total de la venda
10. El Caixer indica al Sistema els diners que ha rebut
11. El Sistema mostra el canvi i imprimeix el rebut
12. El Caixer diposita els diners rebuts a la caixa i n'extreu el canvi
13. El Caixer dóna el canvi i el rebut al Client
14. El Client se'n va amb els productes comprats i el rebut de la compra

8

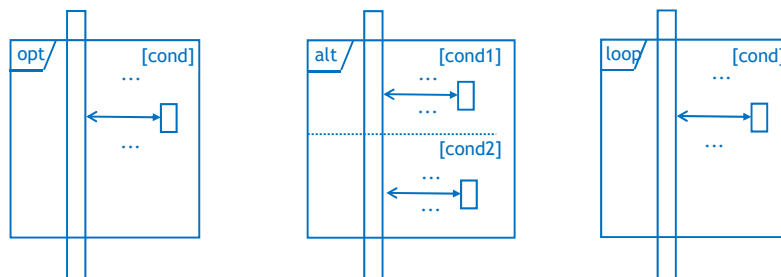
Exemple: diagrama de seqüència del sistema (DSS)



9

Diagrames de seqüència: ús de frames

- Els *frames* permeten estructurar informació en els diagrames UML aquí ens centrem en el seu ús en el context dels diagrames de seqüència
- Tres tipus principals de *frames*:
 - execució opcional (*opt*)
 - execució alternativa (*alt*)
 - execució repetida (*loop*)
- Els *frames* es poden aniar lliurement



10

Esdeveniments i operacions

- **Esdeveniment del sistema:** Esdeveniment extern generat per un actor
- **Operació del sistema:** Operació interna que s'executa com a resposta a la comunicació de l'esdeveniment



La comunicació d'un esdeveniment del sistema provoca l'execució d'una operació del sistema amb el **mateix nom** i els **mateixos paràmetres**

11

Operacions del sistema

Les operacions del sistema **s'agrupen** com operacions de la classe "sistema", la qual usem per a representar el sistema software.

Per tant, **les operacions no s'assignen a objectes concrets durant l'etapa d'especificació**

Exemple:

Sistema
iniciVenda(npv: Integer): venda
entrarProd(venda: Venda, c-prod: Integer, quant: Integer)
fiVenda(venda: Venda): Import
pagament(venda: Venda, importPag: Import): Import

12

Contractes de les operacions

Operació: nom i paràmetres de l'operació (*signatura de l'operació*)

Precondicions:

Condicions que cal satisfer per poder executar l'operació.

Postcondicions:

Canvis d'estat que es produeixen com a conseqüència de l'execució:

- altes/baixes d'instàncies de classes d'objectes i d'associacions
- modificació d'atributs
- generalització o especialització d'un objecte
- canvi de subclasse d'un objecte

Sortida:

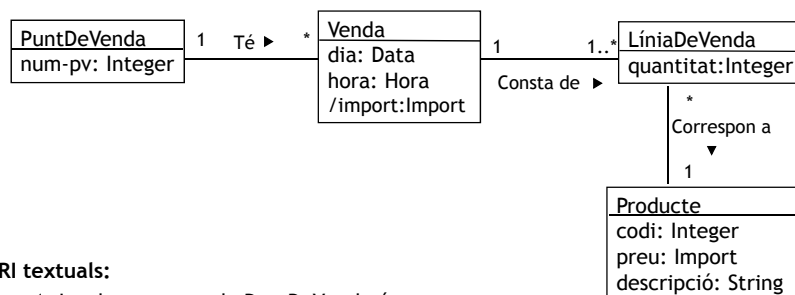
Descripció de la sortida que proporciona l'operació



Observeu el lligam que existeix entre els contractes de les operacions i l'esquema conceptual de les dades

13

Exemple: esquema conceptual de partida



RI textuais:

- 1- La clau externa de PuntDeVenda és num-pv
- 2- La clau externa de Producte és codi
- 3- Un punt de venda no pot tenir més d'una venda amb el mateix dia i hora

Informació derivada:

- 1- L'import d'una venda és igual al cost total dels productes de que consta

14

Exemple: operació *iniciVenda*

Operació: `iniciVenda(npv: Integer, dia: Data, hora: Hora): Venda`

Precondicions:

Existeix un `PuntDeVenda` amb `num-pv=npv`

Postcondicions:

S'ha creat una instància `V` de `Venda` amb `dia` i `hora`

`V` s'ha associat amb la instància de `PuntDeVenda` amb `num-pv=npv`

Sortida: `V`

En OCL:

context: `Sistema::iniciVenda(npv:integer, dia:integer, hora:Integer): Venda`

pre: `PuntDeVenda.allInstances()->exists(p|p.num-pv = npv)`

post:

`Venda.allInstances()->exists(v|v.oclIsNew() and v.dia=dia
and v.hora=hora and v.puntDeVenda.num-pv=npv and result =v)`

15

Exemple: operació *entrarProd*

Operació: `entrarProd(venda: Venda, c-prod: Integer, quant: Integer)`

Precondicions:

Existeix un `Producte` amb `codi = c-prod`

Postcondicions:

S'ha creat una instància de *LíniaDeVenda* `L` amb `quantitat=quant`

`venda` s'ha associat amb `L`

`L` s'ha associat amb el *Producte* amb `codi = c-prod`

En OCL:

context: `Sistema::entrarProd(venda:Venda, codi:Integer, quant:Integer)`

pre: `Producte.allInstances()->exists(p|p.codi = codi)`

post:

`LíniaDeVenda.allinstances()->exists(l|l.oclIsNew() and
l.quantitat = quant and l.venda = venda
and l.producte.codi = codi)`

16

Exemple: operació *fiVenda*

Operació: fiVenda(venda: Venda) :Import

Precondicions:

Postcondicions:

Sortida: es mostra l'import de la venda

En OCL

context: Sistema::fiVenda(venda:Venda) :Import

body: venda.import

17

Exemple: operació *pagament*

Name: pagament(venda: Venda, importPag: Import): Import

Precondicions:

importPag \geq venda.import

Postcondicions:

Sortida: importPag - venda.import

En OCL

context: Sistema::pagament(venda:Venda, importPag:Import):Import

pre: importPag \geq venda.import

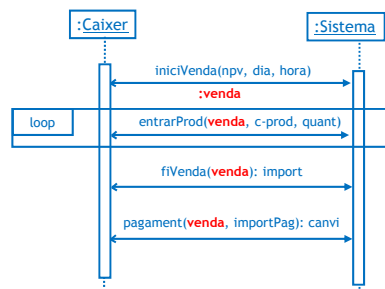
body: importPag-venda.import

18

Compartició d'informació entre operacions d'un DSS

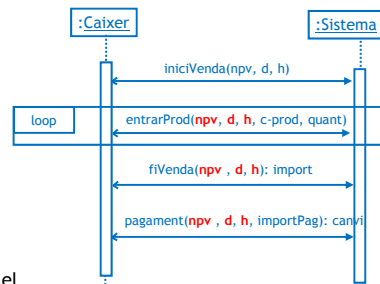
UML no precisa de quina manera les operacions d'un diagrama de seqüència del sistema poden compartir informació

Nosaltres ho especificarem mitjançant arguments addicionals de les operacions



Restricció implícita: el valor de l'argument 'venda' és el mateix a tots els esdeveniments del diagrama

Alternativament, ús de la clau externa:



Restricció implícita: el valor dels argument npv, d i h és el mateix a tots els esdeveniments del diagrama

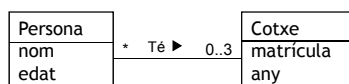
19

Redundància amb l'Esquema Conceptual de les Dades

L'esquema conceptual de les dades conté restriccions d'integritat: *estructurals*, *gràfiques* i *textuals*

L'esquema conceptual ja garanteix que les condicions imposades per aquestes restriccions mai es violaran (independentment de com es modifiqui el seu contingut)

La precondition d'una operació no ha d'incloure les comprovacions necessàries per garantir que l'execució del cas d'ús no viola les restriccions d'integritat definides a l'esquema conceptual de les dades



Restriccions textuals:

1. Claus externes: (Persona,nom); (Cotxe,matricula)
2. Les persones menors de 18 anys no poden tenir cotxe

context: Sistema::comprarCotxe (nom, matr)

post:

Persona.allInstances()->select(plp.nom=nom).
cotxe.matricula->includes(matr)

20

Redundància - Definició

Una especificació és redundant si un **mateix aspecte** del sistema està **definit diverses vegades**.

La redundància **dificulta la modificabilitat** de l'especificació perquè si varia aquell aspecte cal modificar tots els models que hi fan referència.

L'especificació no ha de ser redundant !!!

Redundàncies possibles:

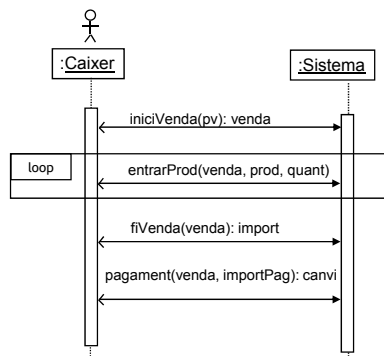
- Entre una precondició i l'esquema conceptual de les dades
- Entre una precondició i els diagrames de seqüència del sistema
- ...

21

Redundància - Diagrames de Seqüència i Contractes

Els diagrames de seqüència defineixen un **ordre d'invocació** de les operacions

Les operacions no han d'incorporar informació per garantir que aquest ordre es compleixi



Operació: Pagament (venda, importPag): canvi

Precondicions:

- existeix venda "venda"
- la venda "venda" ha finalitzat

Postcondicions:

...

Sortida:

són redundants

22

Redundància - Interpretació dels contractes

Com s'han d'interpretar els contractes en relació a l'esquema conceptual?

Precondicions:

què ha de contenir l'Esquema Conceptual per intentar executar una operació

Restriccions d'integritat:

s'han de satisfer després de l'execució de totes les operacions d'un cas d'ús



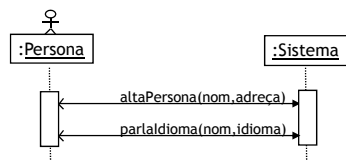
Suposem que es rebutgen totes les operacions d'un cas d'ús si la seva execució viola (globalment) alguna restricció d'integritat de l'esquema conceptual de les dades

23

Redundància - Exemple de diagrama de seqüència incorrecte



Cas d'ús: Afegir-Persona-1



Operació: altaPersona(nom,adreça)

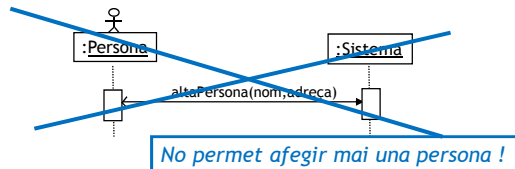
Precondicions:

Postcondicions:

- creació d'un objecte Persona amb el nom i l'adreça especificats

Sortida:

Cas d'ús: Afegir-Persona-2



Operació: parlaldioma(nom,idioma)

Precondicions:

Postcondicions:

- creació d'una associació Parla entre la persona amb nom=nom i l'idioma

Sortida:

24

Especificació en OCL de la sortida d'una operació

La sortida d'una operació acostuma a ser informació composta. O sigui, no elemental.

resumVendes(num-pv: Integer) retorna un llistat de vendes amb la informació:
Per cada Producte venut al PuntDeVenda num-pv mostrar el seu codi i el seu preu

```
resumVendes (num-pv:Integer): ???
```

Definició d'informació composta en OCL:

Tuples: composició de diversos elements, possiblement de tipus diferents

Definició del tipus: `TupleType(camp1:Tipus1,...,campN:TipusN)`

Assignació de valors: `Tuple{camp1=valor1,...,campN=valorN}`

Col·leccions: Set, Sequence i Bag.

```
resumVendes (num-pv:Integer) :  
    Set(TupleType(codiprod:Integer, preu:Import))
```

25

Exemple d'operació de consulta en OCL

context: `Sistema::resumVendes(npv:Integer) :`
`Set(TupleType(cprod:Integer, preu: Import))`

pre: `PuntDeVenda.allInstances()->exists(p|p.num-pv=num-pv)`

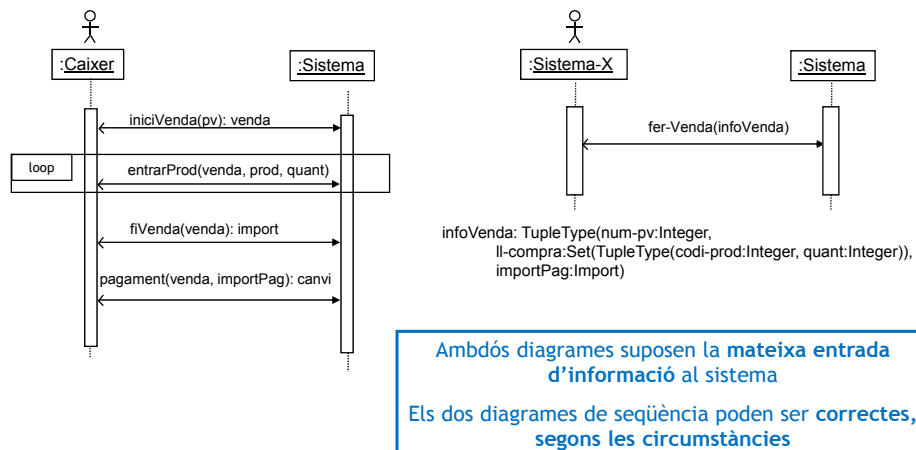
body:

```
let prods:Set(Producte) =  
    Producte.allInstances()->  
    select(p|p.liniaDeVenda.venda.puntDeVenda.num-pv->includes(num-pv))  
in  
prods->collect(p | Tuple{cprod = p.codi, preu = p.preu})
```

26

Diagrama de seqüència: quants esdeveniments?

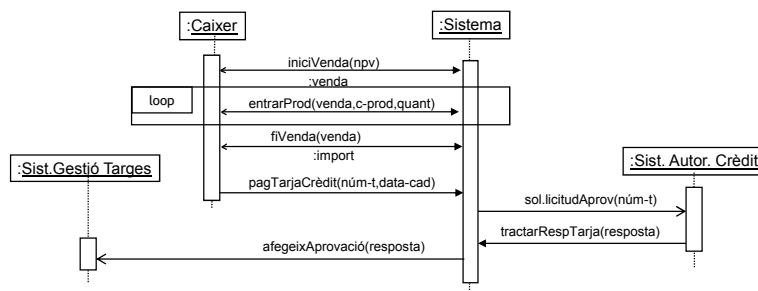
El nombre d'esdeveniments d'un diagrama de seqüència depèn de com es produeix la interacció entre els actors i el sistema software.



27

Interacció amb actors software (serveis) externs

- De vegades un actor d'un cas d'ús pot ser un sistema software (servei) extern al sistema que estem desenvolupant.
- Exemple: cas d'ús comprar productes amb tarja de crèdit*

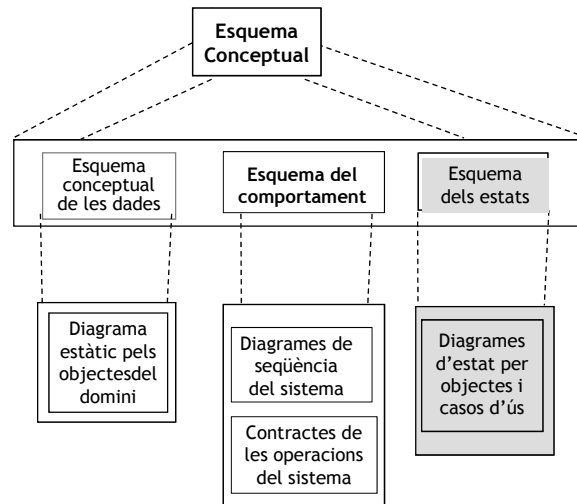


La postcondició de l'operació *pagTarjaCrèdit* inclou la invocació del servei demanat al *Sistema d'Autorització de Crèdit*, mitjançant la frase:

‘El sistema invoca l'execució de l'operació *sol·licitudAprov* al sistema d'autorització de crèdit i li passa *núm-t* com a paràmetre’

28

Esquema conceptual (especificació)



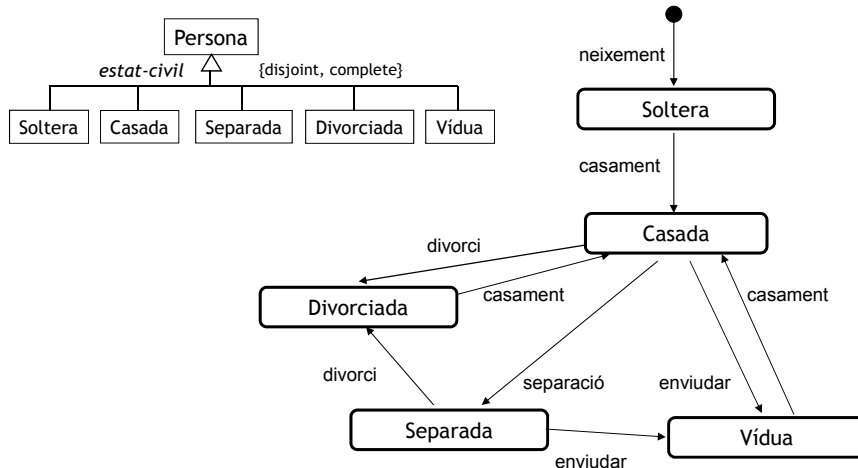
29

Esquema dels Estats

- **Objectius:**
 - crear diagrames d'estat per objectes i casos d'ús
- **Estats, transicions i esdeveniments:**
 - Estat:
 - condició d'un objecte o d'un cas d'ús en un moment del temps
 - Transició:
 - canvi d'estat com a conseqüència d'un esdeveniment
 - Esdeveniment:
 - tot allò que requereix una resposta del sistema software

30

Canvi d'estat civil d'una persona



31

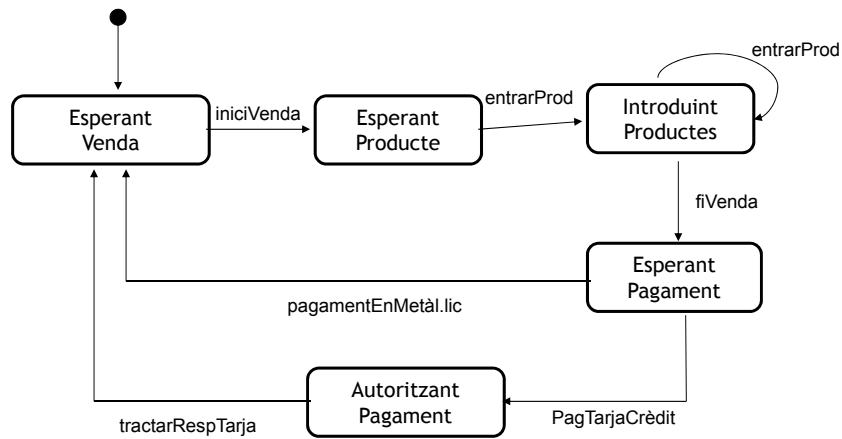
Ús dels diagrames d'estat

- Un diagrama d'estats es pot especificar per a una:
 - **Classe d'objectes:**
 - per descriure perquè els objectes canvien de subclasse
 - les subclasses d'un diagrama d'estats no tenen perquè aparèixer explícitament a l'esquema conceptual
 - per descriure classes d'objectes amb important "comportament dinàmic"
 - **Cas d'ús:**
 - per descriure la seqüència legal en la que els esdeveniments es poden produir al món real

p.ex. en una compra de producte no es pot fer el pagament fins que no s'hagi tancat la venda.

32

Diagrama d'estats del cas d'us “comprar productes”



33

Bibliografia

- Larman, C. “*Applying UML and Patterns. An Introduction to Object-oriented Analysis and Design*”, Prentice Hall, 2005, (3^a edició)
- Rumbaugh, J.; Jacobson, I.; Booch, G. “*The Unified Modeling Language Reference Manual*”, 2^a edició, Addison-Wesley, 2004
- UML 2.3 Superstructure Specification
<http://www.omg.org/spec/UML/2.3/Superstructure/PDF>
OMG, 2010

34