

UML Dinàmic

Diagrames d'Estat

UF3

INTRODUCCIÓ AL DISSENY
ORIENTAT A OBJECTES

M05. Entorns de desenvolupament

Hèctor López
hector.lopez@fje.edu



JESUÏTES El Clot
Escola del Clot



Introducció

Anàlisi i disseny OO

Per fer l'anàlisi i el disseny orientat a objectes fem servir conceptes i notacions UML.

Els diagrames d'UML es poden classificar en tres grups:

- El **model estàtic**, que descriu l'estructura de classes i objectes.
- El **model dinàmic** o de **comportament**, que defineix les interrelacions entre objectes.
- El model d'**implementació**, que descriu el programari en termes de components i la seva ubicació.



UNIFIED MODELING LANGUAGE™



Diagrames d'estat

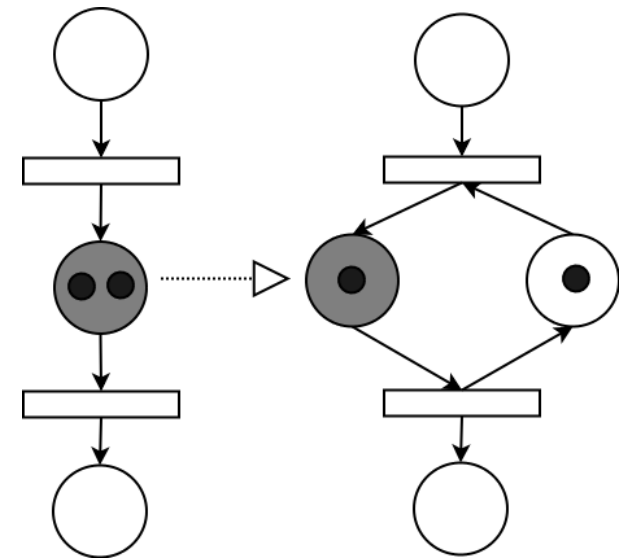
Introducció

Tot sovint hi ha **objectes** que **varien** el seu **comportament** al llarg del temps. Quan això succeeix, es diu que aquest objecte té **estats**.

En algun tipus d'aplicacions, el diagrama d'estats és molt important, com per exemple en les aplicacions en **temps real**.



Altres llenguatges de modelat molt utilitzats en aplicacions en temps real: **PetriNets** & **SDL** (*Specification and Description Language*).



Exemple de diagrama de xarxes de petri

Diagrames d'estat

Conceptes

- Els objectes i els sistemes a cada instant estan en un **estat** concret (v.inicial d'UML).
- L'estat actual d'un objecte o sistema en determina el seu **comportament futur**.
- Els **canvis d'estats** són possibles, i en UML s'anomenen **transicions**.
- La transició entre estats és deguda a un **esdeveniment**.
- Farem servir els diagrames per a **modelar el comportament** (complex) d'objectes i sistemes.
- El diagrama d'estat es documenta en la **fase d'anàlisi**.
- Darrerament s'han incorporat els **diagrames de Harel** a UML, mitjançant els quals, es permet que un objecte o un sistema pugui estar en dos estats de manera simultània (veure secció de transicions complexes).

De la mateixa manera que vam fer al primer bloc d'aquesta unitat formativa, anem a veure poc a poc com es representa cadascun dels conceptes que anem introduint...

Diagrames d'estat

Notació

En un diagrama d'estat, els estats es representen per **rectangles arrodonits**:



El nom d'aquest estat **NO es pot repetir en tot el diagrama**. Ha de ser únic, tot i que un objecte pot arribar diverses vegades al llarg de la seva vida a un determinat estat.

Aquests rectangles arrodonits poden tenir tres compartiments:

- El primer és pel **nom** de l'**estat**.
- El segon per a **valors característics** del sistema o l'objecte quan es troba en aquest **estat**.
- El tercer per a les **accions** que s'executen **a l'entrar, mentre s'està, o al sortir** de l'**estat**.

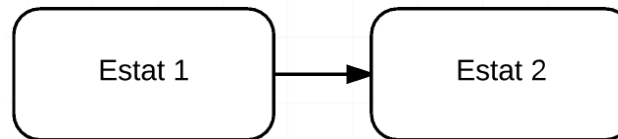
Diagrames d'estat

Notació

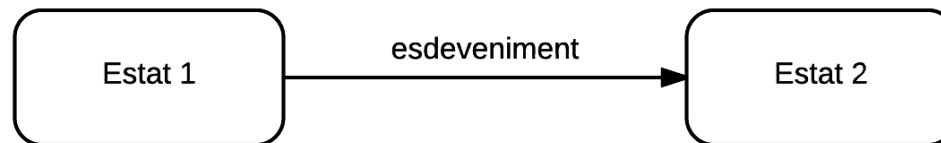
L'estat inicial es simbolitza amb un cercle negre i l'estat final amb un cercle negre/blanc.



Les **transicions** es representen per fletxes, que van d'un estat a l'altre:



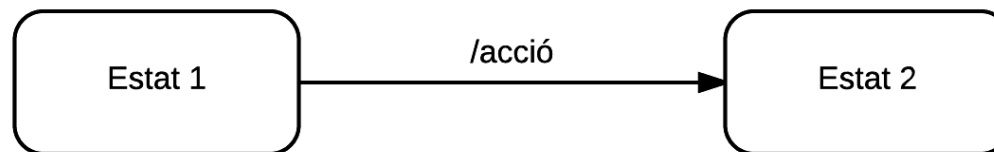
El nom dels **esdeveniments** que disparen les transicions s'escriuen al costat de les fletxes:



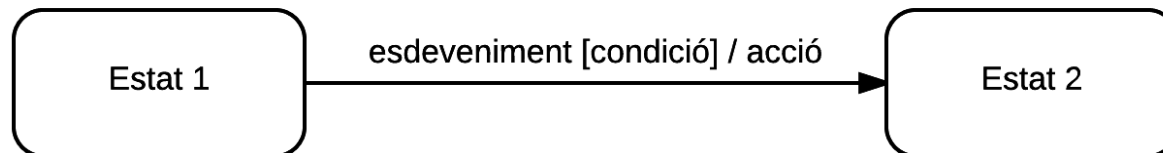
Diagrames d'estat

Notació

El nom dels esdeveniments que disparen les transicions s'escriuen al costat de les fletxes. Pot ser que una transició comporti una acció, que s'indica per **/acció**.



Una acció pot anar precedida d'una **condició de guarda** que es representa per **[condició]**. El text que acompanya a una transició té la forma: **esdeveniment [condició] / acció**.



D'**accions** en podem tenir una, diverses, o cap, i l'ordre d'aquestes és important perquè especifica l'ordre d'execució.

Diagrames d'estat

Diagrames d'estat i modelat d'objectes

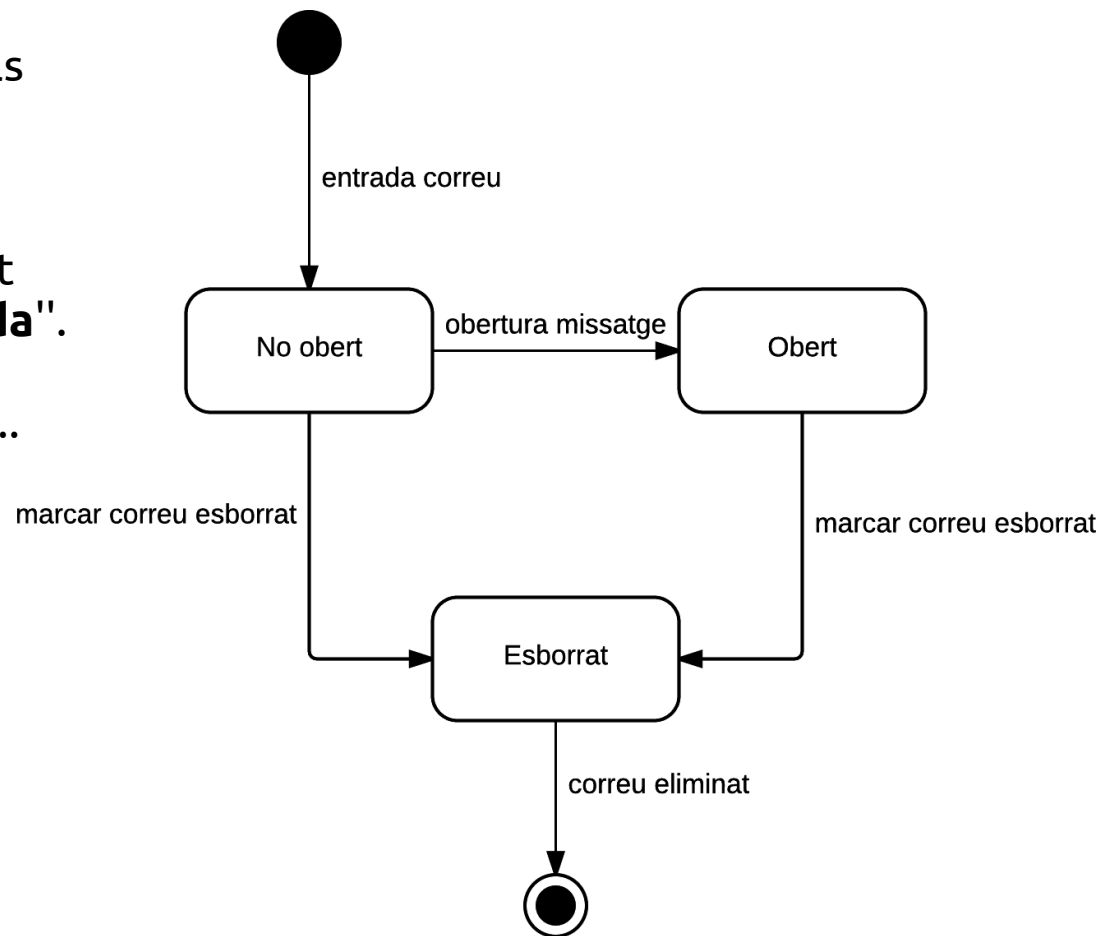
L'**estat** d'un objecte està determinat pels valors dels seus **atributs**.

Els diagrames d'estat ens serveixen per especificar **tots** els **estats** pels quals pot passar un **objecte** al llarg de la seva "**vida**".

Continuem estudiant-ho amb exemples...

Exemple 1 – Correu electrònic

El següent exemple mostra un diagrama d'estat amb els possibles estats d'un (objecte) correu electrònic:



Diagrames d'estat

Diagrames d'estat i modelat d'objectes

La **dinàmica** d'un sistema està determinada per:

- tots els possibles estats que poden tenir els seus objectes.
- totes les seqüències d'esdeveniments possibles
- totes les transaccions possibles d'un estat a un altre, conseqüència dels esdeveniments que afecten els objectes.

Podem fer servir un diagrama d'estat per a **modelar** la **dinàmica** d'un sistema.

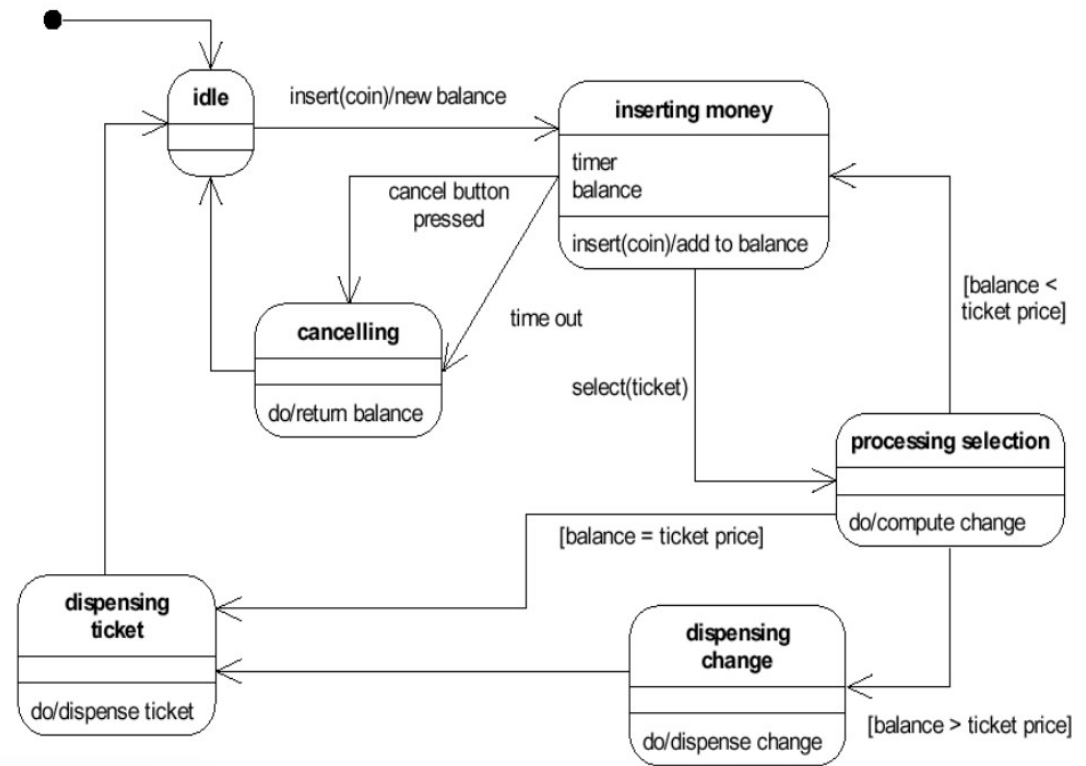
A continuació presentarem dos exemples més:

- **Exemple 2.** Un primer on mostrarem un diagrama d'estat que modela el funcionament d'una màquina expenedora de bitllets.
- **Exemple 3.** Un segon en el qual especificarem el *login* d'un sistema de banca *online*.

Diagrames d'estat

Diagrames d'estat i modelat d'objectes

Exemple 2 – Màquina expenedora



Diagrames d'estat

Diagrames d'estat i modelat d'objectes

Exemple 3 – Login online banking

L'exemple que veurem a continuació mostra un diagrama d'estat que modela la part del "login" d'un sistema de banca on-line.

Per fer el "logging" al sistema, cal introduir un número de seguretat (**SSN**) i un número d'identificació personal (**PIN**), que després **s'han de validar**.

El procés de "logging" es pot factoritzar en quatre estats del sistema disjunts:
→ **GettingSSN, GettingPIN, Validating, Rejecting**.

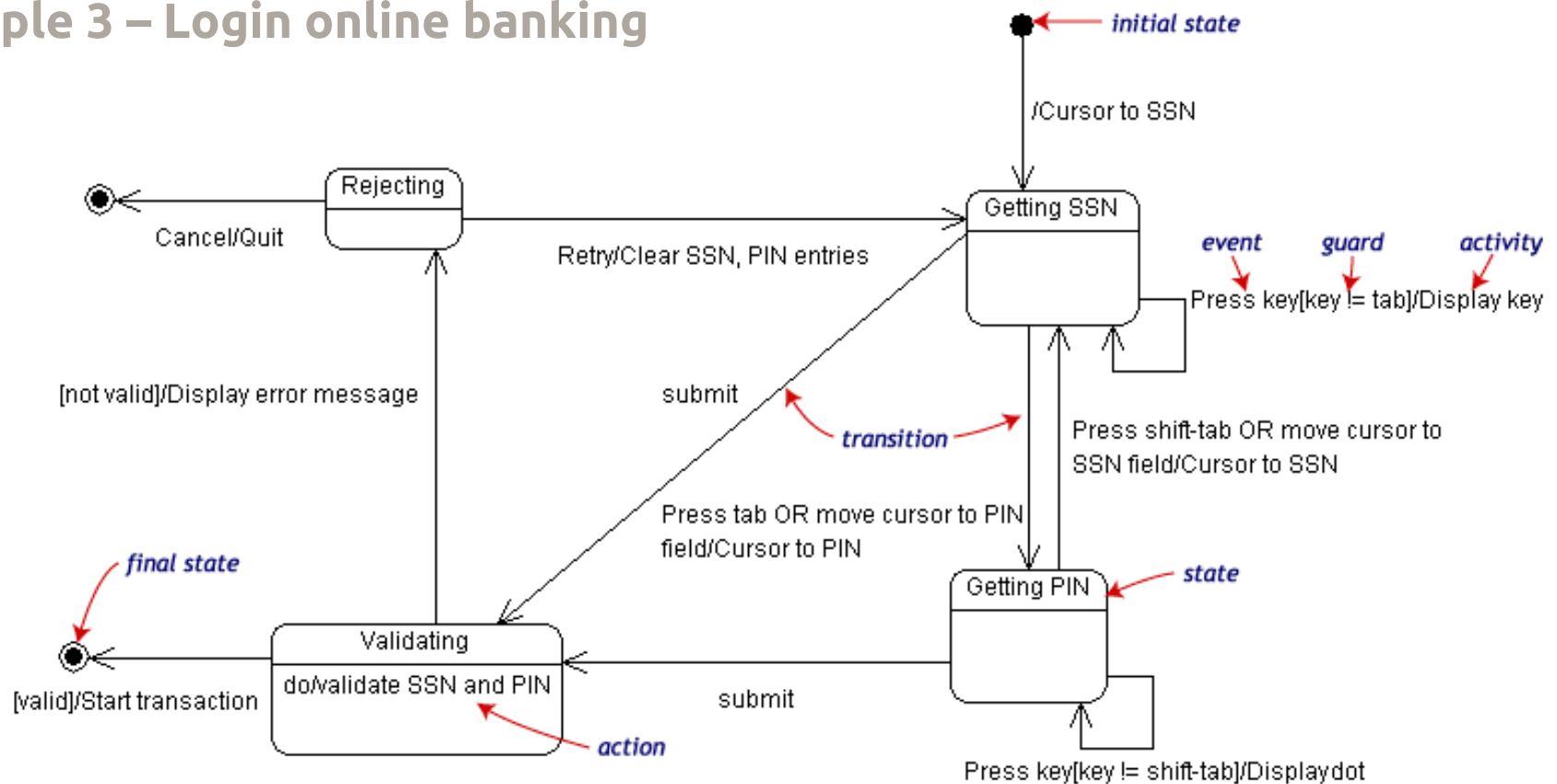
Nota: De cada estat se n'obté un conjunt complet de transicions que determina l'estat subsegüent.

Hi ha dues auto-transicions: **GettingSSN** i **GettingPIN**. A l'estat **Validating** l'objecte no espera que un esdeveniment dispari una transició, però executa una activitat. El resultat d'aquesta activitat en determina el seu proper estat.

Diagrames d'estat

Diagrames d'estat i modelat d'objectes

Exemple 3 – Login online banking



Diagrames d'estat

Glossari (I)

- **Estat:** és una situació determinada dins de la vida d'un objecte. També pot ser la durada d'una interacció durant la qual es compleix alguna condició, es duu a terme alguna acció, o s'espera algun esdeveniment. No necessàriament es correspon a un instant en el temps (pot tenir una durada).
- **Transició Simple:** una transició simple consisteix en què un objecte o interacció passa d'un estat a un altre (**estat origen** → **estat destinació**), podent ser fins i tot el mateix. La transició es dóna quan hi ha un esdeveniment més una condició de "guarda". Tot estat pot tenir transicions d'arribada i transicions de sortida.
- **Transició Interna:** són pseudotransicions en les quals no hi ha canvi d'estat. Serveixen per especificar accions que s'han d'executar en resposta a esdeveniments que no provoquen cap canvi d'estat en l'objecte.

Diagrames d'estat

Glossari (II)

- **Acció:** una acció és l'especificació d'un procés atòmic (**atòmic**: o s'executa totalment, o no s'executa). D'una transició es pot executar una acció o més. Les accions es poden descriure mitjançant procediments.
- **Senyal:** els objectes poden rebre a través de missatges peticions d'operacions o senyals. Les senyals es diferencien de les operacions en què no fan cap procés i només poden produir esdeveniments. A la seva vegada, els esdeveniments provoquen transicions que sí que són capaces de cridar accions.

senyal → esdeveniment → transicions → accions

- **Esdeveniments:** fets que quan es produeixen provoquen transicions d'un estat a un altre; i també es poden donar determinades accions. Els esdeveniments no van lligats a cap objecte en concret, sinó que formen part del paquet. Cada esdeveniment té nom i paràmetres. Quan es produeix un esdeveniment ha de ser tractat, si no, es perd, a no ser que es declari *diferit*. Els esdeveniments poden ser del tipus *síncron*, o *asíncron*.

Diagrames d'estat

Glossari (III)

- **Tipus d'esdeveniment:**

- **De crida:** es produeixen quan es crida una operació d'un objecte.
 - **De senyal:** es produeix quan es rep una senyal.
 - **De canvi:** és una notificació que una determinada condició ha esdevingut certa.
 - **De temps:** representa que o bé ha passat un determinat temps, o bé és una determinada hora.
-
- Hi ha uns esdeveniments que s'anomenen **interns** i són pseudoesdeveniments perquè estan lligats a un estat en comptes d'una transició. Serveixen per engegar accions que no estan vinculades a cap canvi d'estat. Poden ser d'entrada, de sortida, o d'acció.
 - **El.Entrada:** es produeixen quan l'objecte entra en l'estat corresponent. No té paràmetres ni guarda, i s'identifica amb la paraula "**entry**".
 - **El.Sortida:** es produeixen a la sortida de l'estat i s'identifiquen amb la paraula "**exit**".
 - **El.Acció:** especifiquen accions que es generen quan s'arriba a un determinat estat, i desapareixen en el moment que se'n surt. Es declaren amb la paraula "**do**".

Diagrames d'estat

Sintaxi

La **signatura** de l'esdeveniment doncs, dependrà del seu tipus:

- De crida / senyal:

nom_esdeveniment (nom_parametre : exp_tipus, ...)

- De temps (1):

after (expressio_de_temps)

on *after* és la durada, per exemple, *when (hora/data)*

- De temps (2):

when (expressio_booleana)

Diagrames d'estat

Sintaxi

- La **condició de guarda** és una expressió que pot prendre el valor de *cert* o *fals*; s'escriu en pseudocodi, i determina quan es compleix l'esdeveniment.
- L'**acció** és una especificació en pseudocodi que indica quins procediments s'hauran d'executar quan es compleixi l'esdeveniment.
- Una **tramesa** pren la forma *destinació* *'.' missatge('argument','...')*, on destinació pot ser un o més objectes i missatge és una operació de l'objecte de destinació o una senyal.

Amb UML, si hem de definir una senyal, ho podem fer com una classe, que no té operacions ni relacions (només herència), i que només té com atribut els paràmetres de senyal (amb l'estereotip «signal»). Al símbol de classe se li pot afegir un compartiment especial per indicar quines són les senyals a les quals són sensibles.

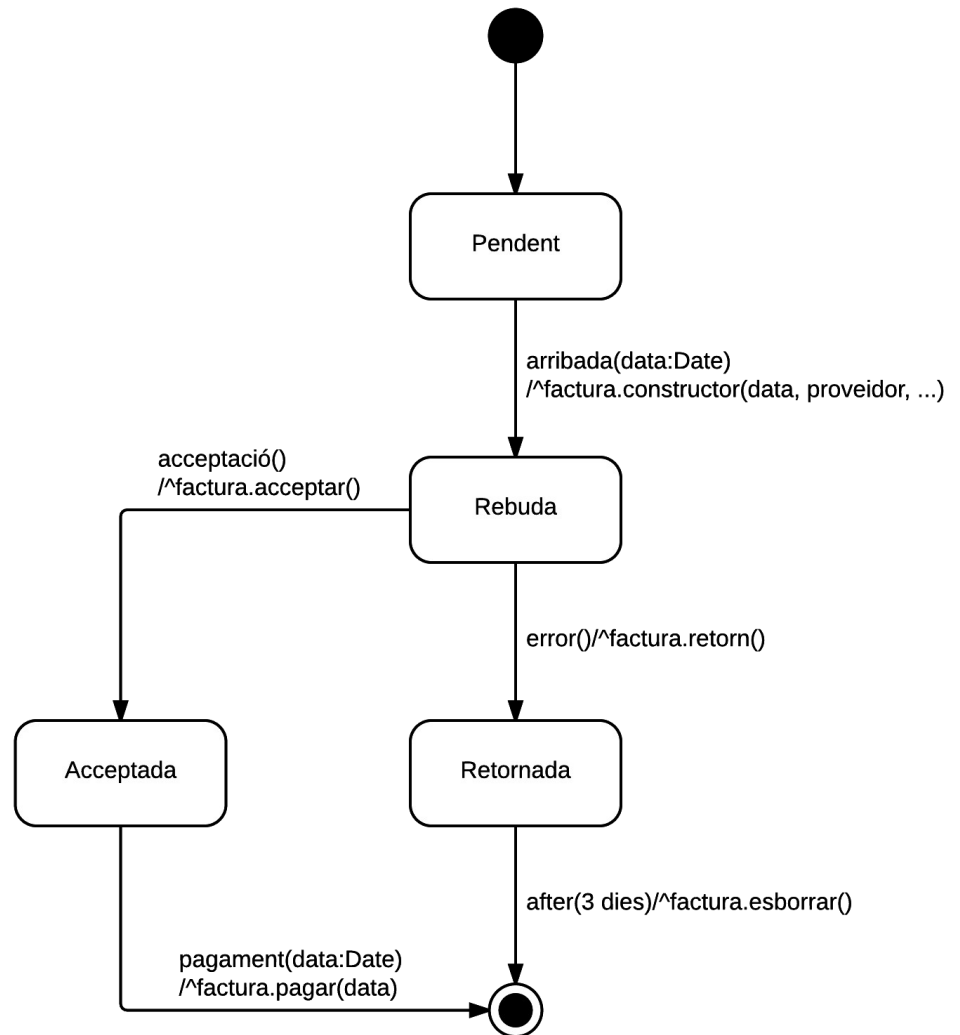
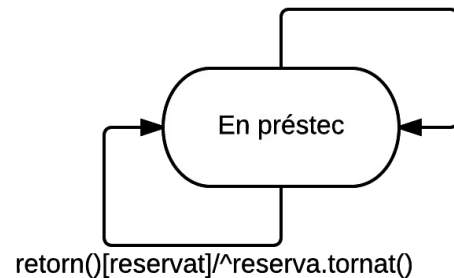
Exemple d'esdeveniment: `introduir_targeta [targeta=valida] / obrir_porta ^ registrar_entrada`

O bé: `introduir_targeta [targeta=valida] / defer obrir_porta ^ registrar_entrada`
(*aquest ho deixa per més endavant: diferit*)

Diagrames d'estat

Més exemples...

peticio_renovacio(data:Date, codiLlibre: String) [no_reservat]
/ ^prestec_renovacio(data)



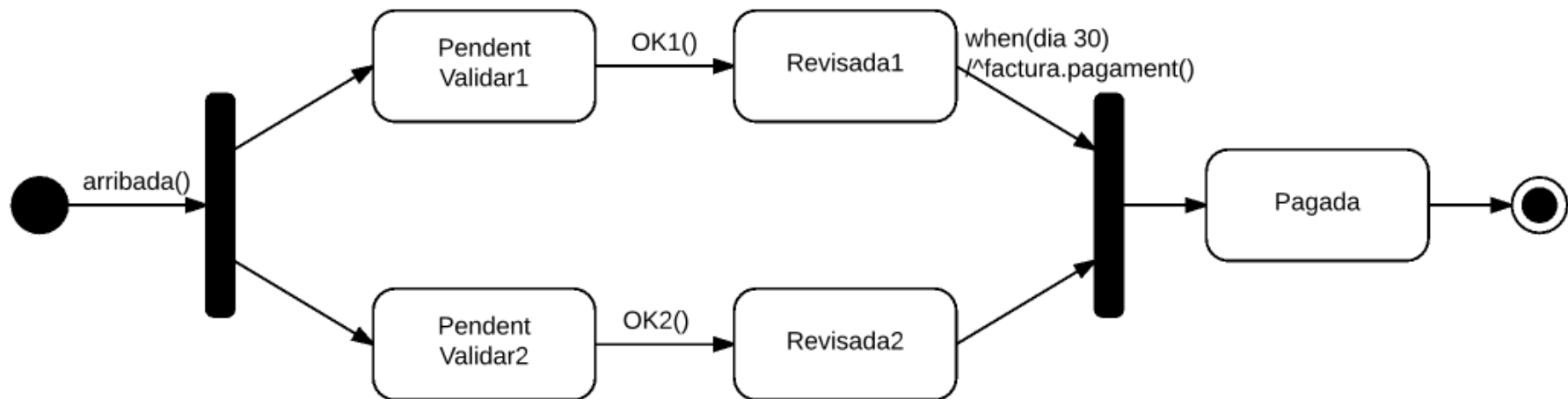
Diagrames d'estat

Transicions complexes

Un **objecte** o **interacció** pot estar en **més d'un estat** al mateix temps, i hi pot haver **més d'una transició** sortint d'un estat i **més d'una transició** arribant a un estat.

Aquestes s'anomenen **transicions complexes**. S'utilitza un pseudoestat intermedi que s'anomena de **sincronització** o de **bifurcació**.

Un altre exemple de factures:



Diagrames d'estat

Estats compostos

Un **estat compost** és aquell en el qual hi ha **diversos subestats possibles**, cadascun dels quals també pot ser a la vegada compost o simple. Tot estat compost té un **diagrama de subestats**, que poden ser **concurrents** (a la vegada) o **seqüencials** (un darrere de l'altre).

Es representa com un estat simple però amb 2 compartiments. Dins d'un estat compost poden haver transicions que van de subestat a subestat; transicions que entren o surten de l'estat compost, o bé transicions que tenen com a destinació un indicador d'història d'estats, que és un pseudoestat que recorda a l'estat del qual va sortir.

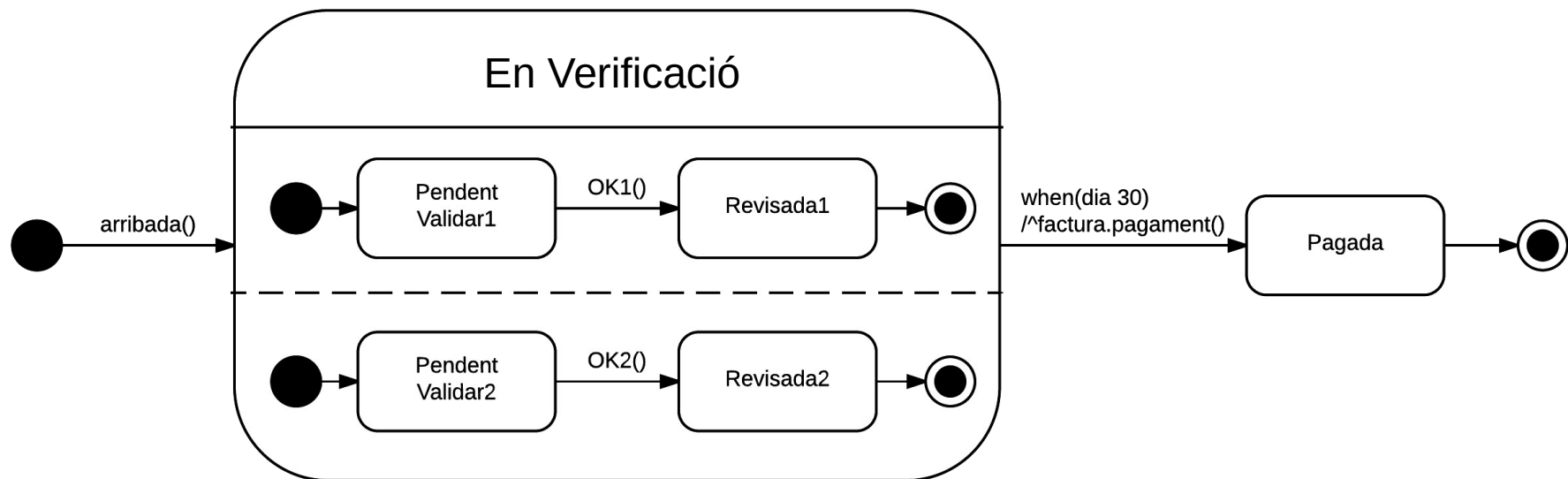
De l'**indicador d'història** pot sortir una transició cap a un subestat i serà la destinació en el cas que la transició tingui lloc sense que s'hagi estat cap vegada a l'estat compost.

Finalment hi ha unes transicions que s'anomenen **stubbed**, que tenen com a destinació o origen algun subestat d'un estat compost que no s'identifica en el diagrama.

Diagrames d'estat

Estats compostos

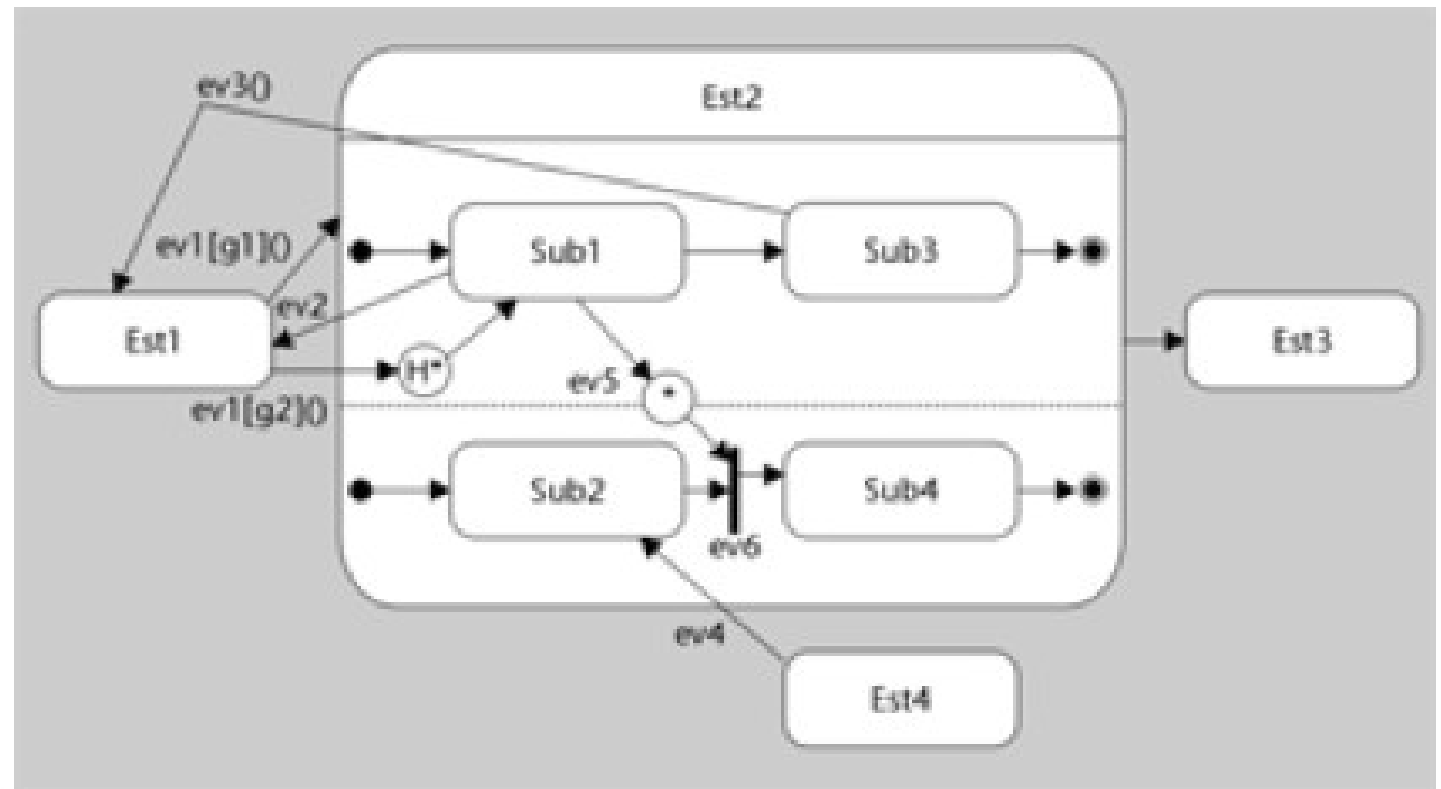
Exemples



Diagrames d'estat

Estats compostos

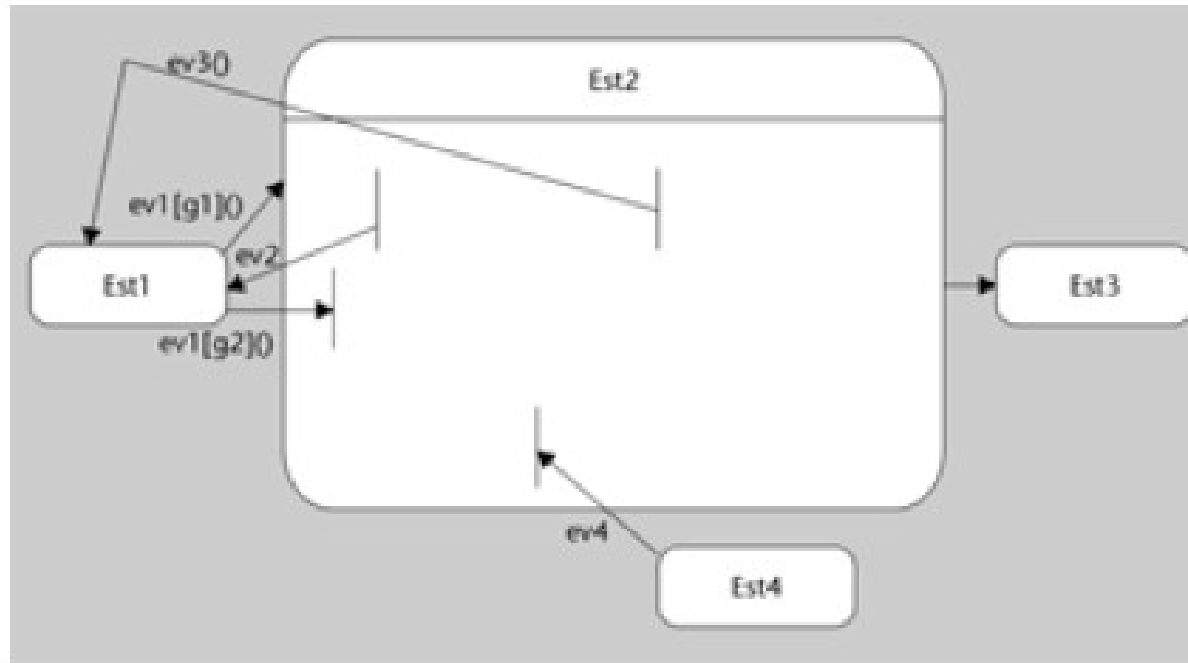
Exemples



Diagrames d'estat

Estats compostos

Exemples



Bibliografia

UF3

M05 Entorns de Desenvolupament

**INTRODUCCIÓ AL DISSENY
ORIENTAT A OBJECTES**

- UML. OMG. <http://www.uml.org>
- OMG Unified Modeling Language™. OMG Document Number: formal/2015-03-01.
- Enginyeria del Software. Universitat Politècnica de Catalunya. FIB. Xavier Franch i Dolors Costal.
- UML & Patterns. Ed. Prentice Hall. 2nd Edition. Craig Larman.
- El Lenguaje Unificado de Modelado. Manual de Referencia. Ed. Addison Wesley. VVAA
- Enginyeria del Software I. Apunts de teoria. Univesitat Pompeu Fabra. VVAA.
- Enginyeria del Software I. Problemes. Universitat Pompeu Fabra. VVAA.
- Enginyeria del Programari Orientada a Objectes. Universitat de Girona. Toni Sellarès.
- Enginyeria del Programari Benet Campderrich Falgueras. Universitat Oberta de Catalunya.
- Diagrama d'activitat i d'estats. Universitat de Castelló. Departament LSI. Reyes Grangel Seguer.
- Desarrollo Orientado a Objetos con UML. Programación C.E.C.yT. Juan de Dios Bátiz Paredes – IPN.
- Modelos y Herramientas UML: Modelo Estructurado. Máster en Computación. Elena Mediavilla.
- Ingeniería del Software. Facultad de Informática de San Sebastián UPV/EHU. Germán Rigau.
- Ingeniería del Software. Universidad de Sevilla. Departamento de Informática y Automatas. VVAA.
- Diagramas UML de casos de uso y requisitos. Universidad de Sevilla. Dr. Javier Jesús Gutiérrez.
- Ingeniería del Software. UNED. VVAA.
- Introduction to OCL. Universität Koblenz-Landau. Bernhard Beckert.

