

Ampliació a l'enginyeria del programari



Primers principis de disseny

Què hi ha en aquest material

2

- Què són els principis **GRASP**
- Principi general d'assignació: **Expert**
- Captura d'ES: **Controlador**
- Principis avaluatius: **Alta Cohesio;**
Baix Acoblament



- **GRASP** (General Responsibility Assignment Software Principles)
 - Principis bàsics de disseny per a l'assignació de responsabilitats
 - Introduïts per en **Larman**
 - Constitueixen un vocabulari universalment conegut
 - ✦ Facilitat de comunicació

Un principi general



4

- **Problema:** Quin principi general podem aplicar per assignar alguna responsabilitat a un objecte?
- Solució: **Principi Expert**



La responsabilitat R l'assignem a un component C que tingui (o sàpiga com obtenir) tota la informació necessària per poder realitzar la responsabilitat demanada

- Diem que C és **expert** en R

Dos significats d'expert



6

- **Principi Expert**
 - Principi general d'assignació de responsabilitats
- **Component (o objecte) expert en R**
 - Component o objecte que té el potencial o coneixement suficient per realitzar la responsabilitat R
- **Dos o un**
 - *El principi Expert assigna la responsabilitat R a algun dels experts en R.*

Experts parcials i per delegació



7

- **Expert parcial** (en R)
 - Coneix part de la informació necessària per exercir la responsabilitat R, però no la coneix pas tota
- **Expert per delegació** (en R)
 - No és un expert però se'n pot convertir si li enviem la informació pertinent

Diferents formes d'usar Expert



8

- **Expert**: La responsabilitat R l'assignem a algun dels seus experts
 - L'expert pot necessitar **col·laborar** amb algú altre per obtenir part de la informació d'expertesa
 - L'existència d'experts **parcials** pot exigir la necessitat d'una col·laboració gestionada
 - La **delegació** de l'expertesa augmenta els candidats a exercir la responsabilitat

Primera responsabilitat

9

- Quina és la primera responsabilitat que cal resoldre?
 - Quina és la primera responsabilitat que cal assignar?

Captura d'un ES



10

- És la primera responsabilitat que cal exercir
 - Una cosa és capturar la petició externa
 - Una altra cosa exercir la responsabilitat demanada amb la petició
- Recordatori:
 - ES = Petició al sistema per sota de la capa d'interacció amb l'usuari

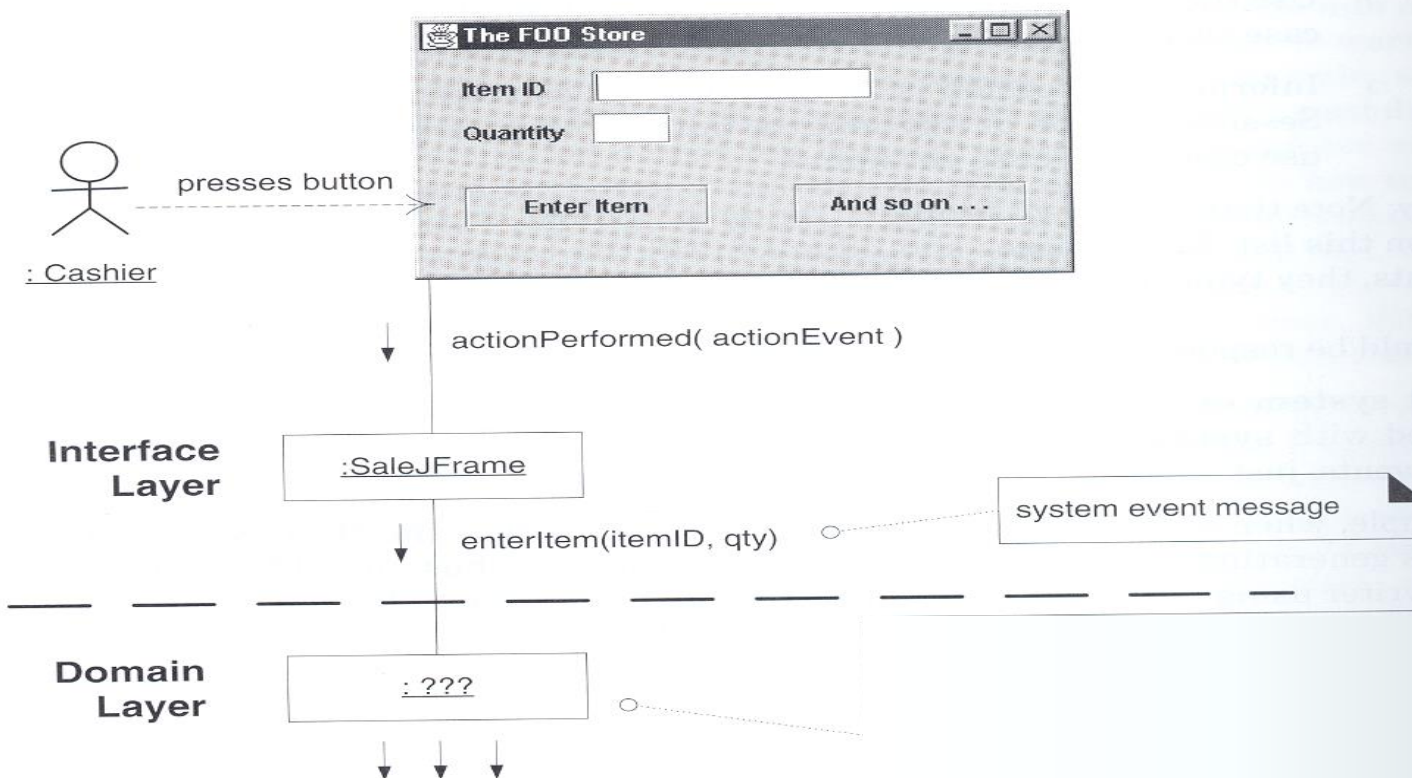


Figure 16.14 Controller for enterItem?

Captura d'un ES segons Expert

12

- Què diu **Expert**

- Un ES pot fer qualsevol tipus de petició. Per tant les necessitats d'informació poden ser qualssevol
 - ✦ Potencialment necessitem conèixer tot el sistema
- L'expert ha de conèixer tot el **Domini** del sistema i la capa d'interacció d'aquest amb l'exterior (**Presentació**)



La responsabilitat de **capturar** un ES l'assignem a un component C que representi o modelitzi la **totalitat del sistema** o subsistema considerat

- Diem que C és el **controlador** d'aquest ES
- *Des de la Presentació, el controlador és l'expert en el Domini*

Controlador:
principi i component
(o objecte) que rep
la responsabilitat de
captura

Controlador

14

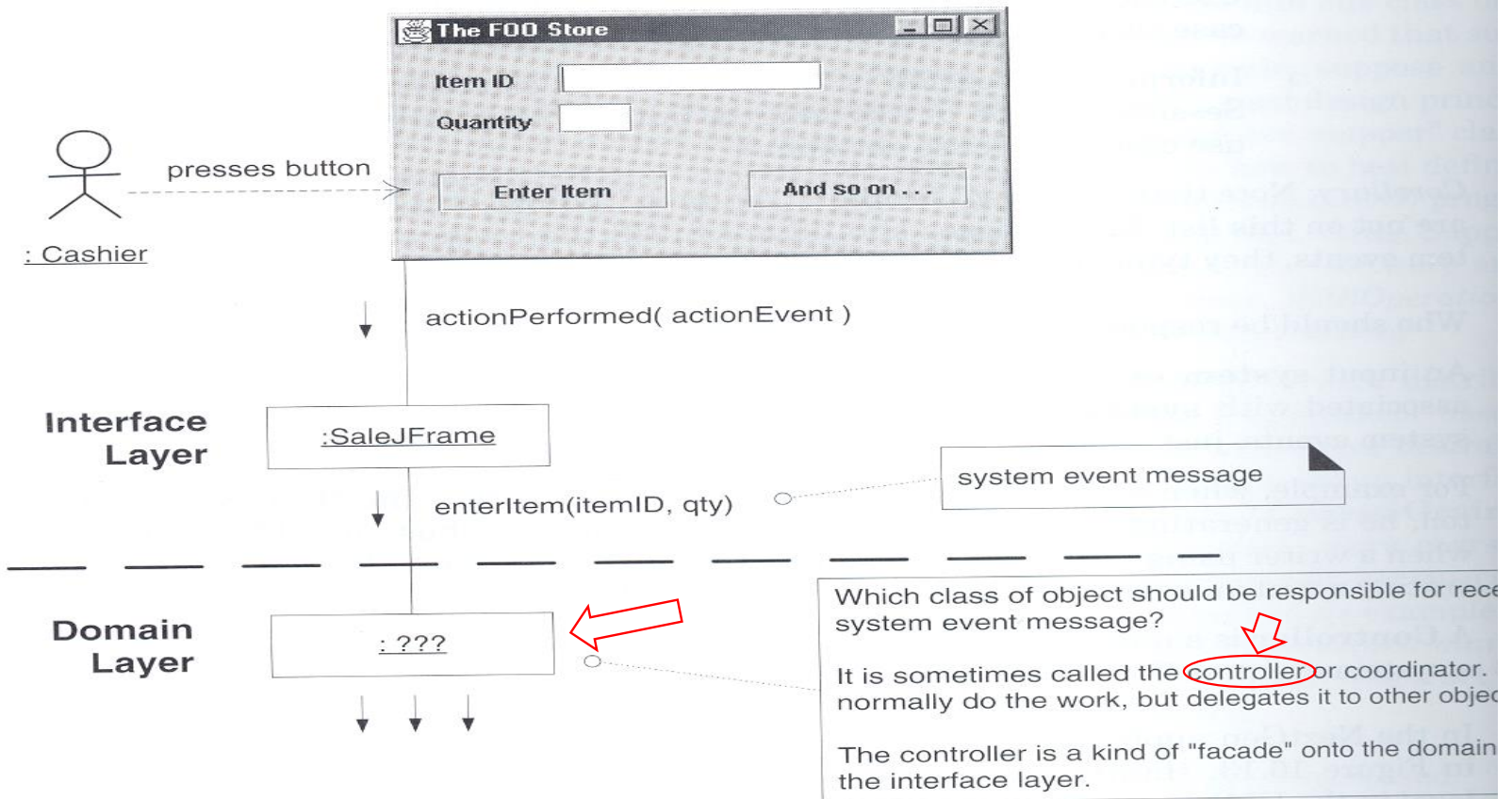


Figure 16.14 Controller for enterItem?

Comunicació entre ES

15

- Els diferents ES que participen en un mateix CU poden tenir necessitat de comunicar-se entre ells
 - Exemple: Un ES prepara un objecte per tal que hi interactuïn els propers ES
- Els ES són externs, asíncrons i no els tenim sota control
 - Com ho podem fer per comunicar-los?



Tots els ES d'un mateix CU són capturats pel mateix objecte controlador

Mateix = mateix **objecte**,
no pas component

- Els ES són capturats per un controlador
 - Un controlador comú pot mantenir la informació a comunicar com a valors dels seus atributs
 - Podem parlar de “Controlador del CU” enlloc del “Controlador de l'ES”

Controlador artificial



17

- Situació:
 - Hem d'usar un controlador
 - No podem assignar aquesta responsabilitat a cap component
 - ✦ No tenim cap concepte que modelitzi tot el sistema o subsistema considerat
 - ✦ Els components que poden rebre la responsabilitat no són pas adients
 - En assignar-los la responsabilitat, violem d'altres principis de disseny
- Solució
 - Introduïm un component “artificial”
 - ✦ **Violem Espill**

El principi de la **Fabricació Pura** ens diu sota quines condicions es pot violar Espill. El cas del controlador artificial compleix aquestes condicions.



La tasca bàsica d'un controlador és la **delegació** de les responsabilitats que els ES li demanen

- El controlador no fa res; és el pont entre la *Presentació* i el *Domini*

CU ferInscripció



CU ferInscripció

20

Contracte *novaInscripcio(c:Caminada)*

Paràmetres

- *c*: Caminada a la que es vol fer la inscripció

PRE

1. No hi ha cap *Inscripció* activa

POST

1. S'ha creat una nova realització *i:Inscripció*
2. S'ha creat un enllaç entre *i* i *c* corresponent a l'associació relativa a
3. La inscripció *i* passa a estar activa

POSTs mal
expressades.
Perquè??

- Partim del MComp buit
- Accedim a MC a veure si hi ha algun **concepte** que representi tot el sistema
 - No n'hi ha cap
 - Introduïm el **component** `GastemLaSola`

D'on pot sortir GLS

22

- Ja el teníem a MC

No és el nostre cas

- Afegim un component homòleg a MComp (Espill)

- No el teníem a MC

Ara sí que és el nostre cas

- Podem modificar MC per tal que el tingui

- ✦ Desenvolupament àgil

- ✦ **Cal analitzar les seves interrelacions**

- Podem introduir un controlador de CU artificial

- ✦ **Viollem Espill**

- Hem vist que és un cas on es pot violar

El principi de la **Fabricació pura** és qui permet aquesta violació

Amb l'objectiu d'evitar al màxim les reconsideracions, sempre que sigui possible intentarem mantenir el MC

Singleton

23

- Concepte *GastemLaSola* (en cas d'haver-lo introduït)
 - En volem una sola realització (Singleton)
- Component *GastemLaSola*
 - En volem una sola realització (Singleton)

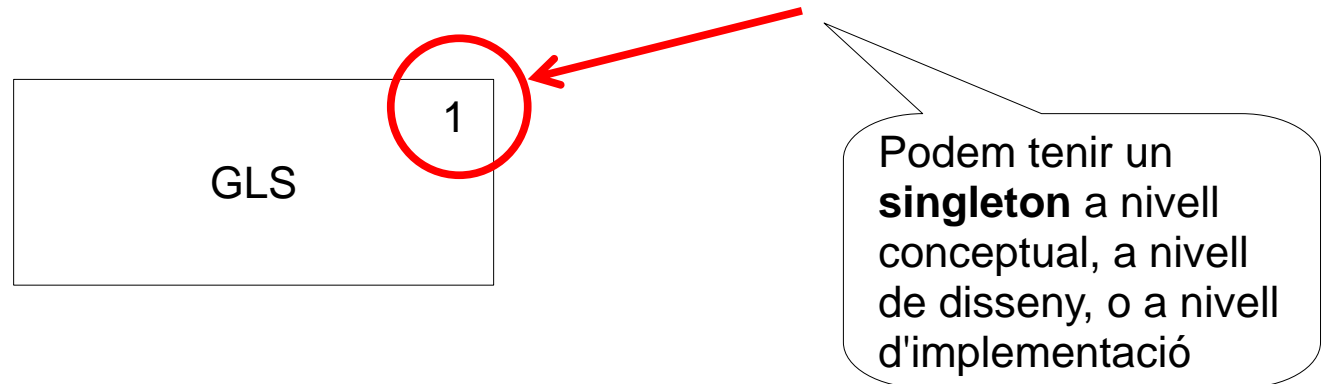
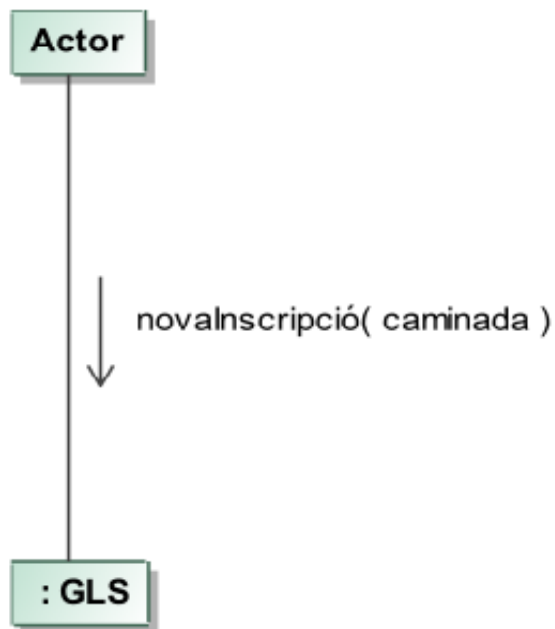
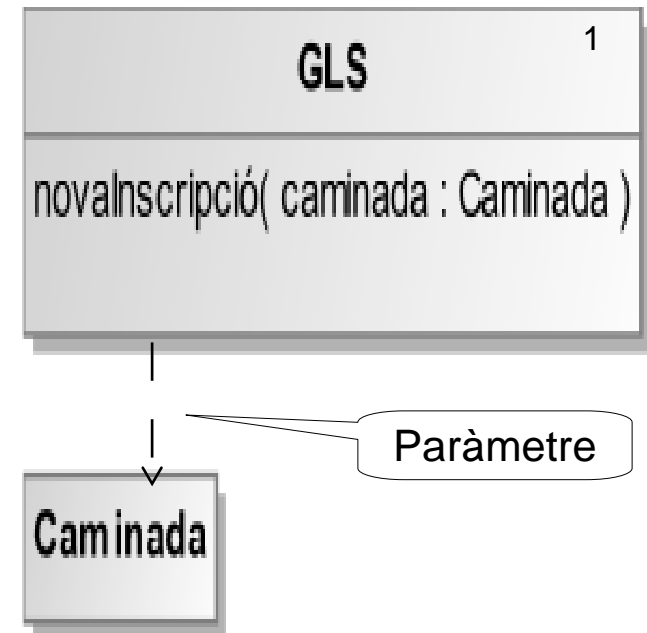




Diagrama de comunicació



MComp



Creació de la `i:Inscripció`

25

- Ja hem resolt la primera responsabilitat: la captura de l'ES `novaInscripció (c:Caminada)`
- Ara toca assignar les responsabilitats que pertoqueuen a aquest ES
 - Les responsabilitats d'un ES estan expressades en les POST del seu contracte
 - ✦ Les PRE indiquen quines condicions podem assumir per tal d'arribar a les POST
- A `novaInscripció (c:Caminada)` la primera responsabilitat és assegurar l'existència d'una nova `i:Inscripció`
 - Cal, per tant, crear una `i:Inscripció`



- Informació d'expertesa

- Caminada +
- Arguments de l'ES
 - ✦ Data, ... (Caminada)

Les necessitats d'informació dels ES poden ser *explícites* o *implícites* (data del sistema, objecte actiu, ...)

- Experts

- Controlador
 - ✦ Com a capturador de l'ES, en coneix els seus arguments
 - ✦ També coneix la Caminada perquè és un argument de l'ES
- Caminada
 - ✦ Expert per delegació des del controlador
 - La delegació és possible perquè l'ES aporta la Caminada



Necessitat dels principis avaluatius

27

- Tenim dos experts en la creació
- **Amb quin dels dos ens quedem?**
 - Exercici
 - ✦ Assignar la responsabilitat a algun dels dos experts viola algun principi?
 - **Necesitem principis avaluatius**

Principis avaluatius

28



- **Mesura qualitativa** sobre el grau de relació i focalització de les tasques o responsabilitats assignades a un mateix component
- **Aproximacions quantitatives:**
 - Nombre de responsabilitats assignades
 - Nombre de missatges que sap respondre

Llegible tant a MComp com des dels diagrames d'interacció

Alta Cohesió (GRASP)

30

- Davant de dues alternatives d'assignació de responsabilitats, ens decidim per aquella que en el disseny resultant els components tinguin una **cohesió més elevada**



- **Mesura qualitativa** de la força amb què un component està connectat o té coneixement d'altres components
- **Aproximació quantitativa:**
 - Nombre de dependències
 - ✦ Visibilitats d'atribut i dependències

Llegible sobretot a MComp

Baix Acoblament (GRASP)

32

- Davant de dues alternatives d'assignació de responsabilitats, ens decidim per aquella que en el disseny resultant els components tinguin un **acoblament més baix**

Dos models per comparar



33

- Model 1

- GLS

- ✦ Responsabilitats

- Controlador

- Creador d'Inscripció

- ✦ Coneixement

- Caminada

- Inscripció

- Caminada

Sense responsabilitats ni necessitats de coneixement conegudes

Expert
en la
creació

- Model 2

- GLS

- ✦ Responsabilitats

- Controlador

- ✦ Coneixement

- Caminada

- Caminada

- ✦ Responsabilitats

- Creador d'Inscripció

- ✦ Coneixement

- Inscripció

- Caminada

Expert
en la
creació



- **Model 1**

- GLS

- ✦ més acoblat
 - ✦ menys cohesionat

- Caminada

- ✦ menys acoblada
 - ✦ més cohesionada

- **Model 2**

- GLS

- ✦ menys acoblat
 - ✦ més cohesionat

- Caminada

- ✦ més acoblada
 - ✦ menys cohesionada

- L'anàlisi ha de ser global, no pas individual per cada component



- **Model 1**

- Tres* components,
- Dues responsabilitats
 - ✦ Assignades a un sol component
- Un component ha de conèixer els altres dos

* Estem creant una Inscripció. Per tant aquest és un component que segur que hem de tenir

- **Model 2**

- Tres* components,
- Dues responsabilitats
 - ✦ Assignades a DOS components
 - ✦ Cal afegir una delegació de responsabilitat
- Cada component n'ha de conèixer com a molt un altre

Màxima distribució de coneixement
(**Baix Acoblament**)

Màxima distribució de responsabilitats
(**Alta Cohesió**)

Conclusions actuals

36

- M1

- Viola *Alta Cohesio*
- Viola *Baix Acoblament*
- Viola *Encarrilament*

- ✦ El podem veure com un cas particular d'Alta Cohesió

- M2

- Aparentment no viola cap dels principis

- Model considerat

- Controlador: GLS
- Creador d'Inscripció: Caminada