

Ampliació a l'enginyeria del programari

Arguments implícits

Què hi ha en aquest material

2

- Activació de realitzacions
- Eficiència en segon pla
- Desactivació de realitzacions

Activació

3

EFICIÈNCIA EN SEGON PLA

Repàs: Comunicació entre ES

4

- 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 interactuin els propers ES
- Els ES són externs, **asíncrons** i no els tenim sota control
 - Com ho podem fer per comunicar-los?

- Els ES són capturats per un controlador
- Un controlador comú pot mantenir la informació a comunicar com a valors dels seus atributs

Controlador de CU

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

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

- Podem parlar de “Controlador del CU” enlloc del “Controlador de l'ES”

Visibilitat de les realitzacions actives

6

- En una **PRE** l'exigència d'una **realització activa** es correspon a l'existència d'un argument per omissió o **implícit**
 - L'operació que respon el missatge actua sobre la realització activa que exigeix la PRE
- Per tant, l'operació que respon a l'ES ha de tenir **visibilitat** sobre la realització activa exigida.
 - I aquesta visibilitat només pot ser d'**atribut**

Alguns sinònims:

- Actiu
- Diferenciat
- Apartat
- Pintat
- Separat

Perquè?

Quina visibilitat necessitem?

7

- **N'hi ha prou amb la visibilitat d'atribut?**
 - La visibilitat multiavaluada no permet saber quin dels objectes que són destinació de l'enllaç dirigit és el que cal emprar com a argument implícit
 - ✦ Cal un mecanisme de desambiguació!
- **Cal la visibilitat monoavaluada d'atribut?**
 - És imprescindible o es pot relaxar la condició?

Assumim un sol argument implícit. Altrament la multiavaluació pot tenir sentit

Les realitzacions actives en el disseny

8

- Per tal de poder usar un argument implícit en els ES cal que el controlador en tingui **visibilitat d'atribut indirecta monoavaluada**
 - No cal restringir-nos a la visibilitat directa d'atribut
 - ✦ N'hi ha prou amb l'existència d'una cadena de visibilitats des del controlador

Cadena de visibilitat = Seqüència enllaçable amb enllaços dirigits que s'encadenen. Totes les visibilitats (amb la possible excepció de la darrera) han de ser monoavaluades

Repàs: Visibilitat indirecta d'atribut

9

- Donada una **visibilitat indirecta** direm que és **d'atribut** si totes les visibilitats que formen la cadena pertinent de visibilitats directes són visibilitats d'atribut
 - **Condició suficient:** En el MComp la visibilitat indirecta d'atribut es correspon a la composició d'una seqüència de visibilitats d'atribut obligades, tal que l'extrem de sortida de cada associació de la seqüència té multiplicitat 1.



M1: i es manté activa

10

- POST: cal donar al controlador **visibilitat d'atribut monoavaluada** de la `i:Inscripció` acabada de crear:

`gls → i:Inscripció`

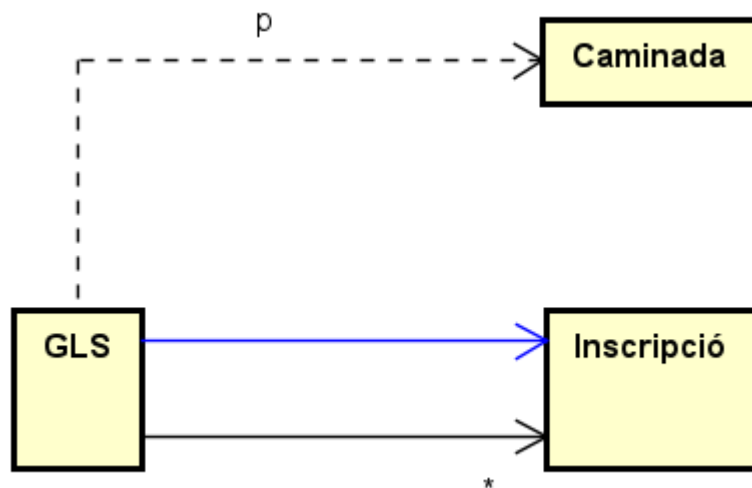
- Però en aquest cas l'enllaç dirigit necessari per l'emmagatzematge és **multiavaluat**:

`gls → Inscripció*`

- Cal desambiguar
 - El més simple és dotar al controlador de **visibilitat directa monoavaluada** de la inscripció activa

M1. DMC amb activació

11



M1: Inscripció activa i *Espill*

12

- La visibilitat d'atribut per mantenir la inscripció activa no sembla que es correspongui a cap associació de MC (violem *Espill*)
- En l'especificació no havíem indicat cap associació “activa” entre *GLS* i *Inscripció*
 - Malgrat tot la petició en els contractes de realitzacions actives imposa unes condicions no reflectides en el MC
 - ✦ La **visibilitat d'atribut** modelitza els enllaços explicitats per les associacions, però també altres **lligams implícits** com el manteniment d'una realització activa

No violem
Espill

M1: Inscripció activa i *Creador efectiu*

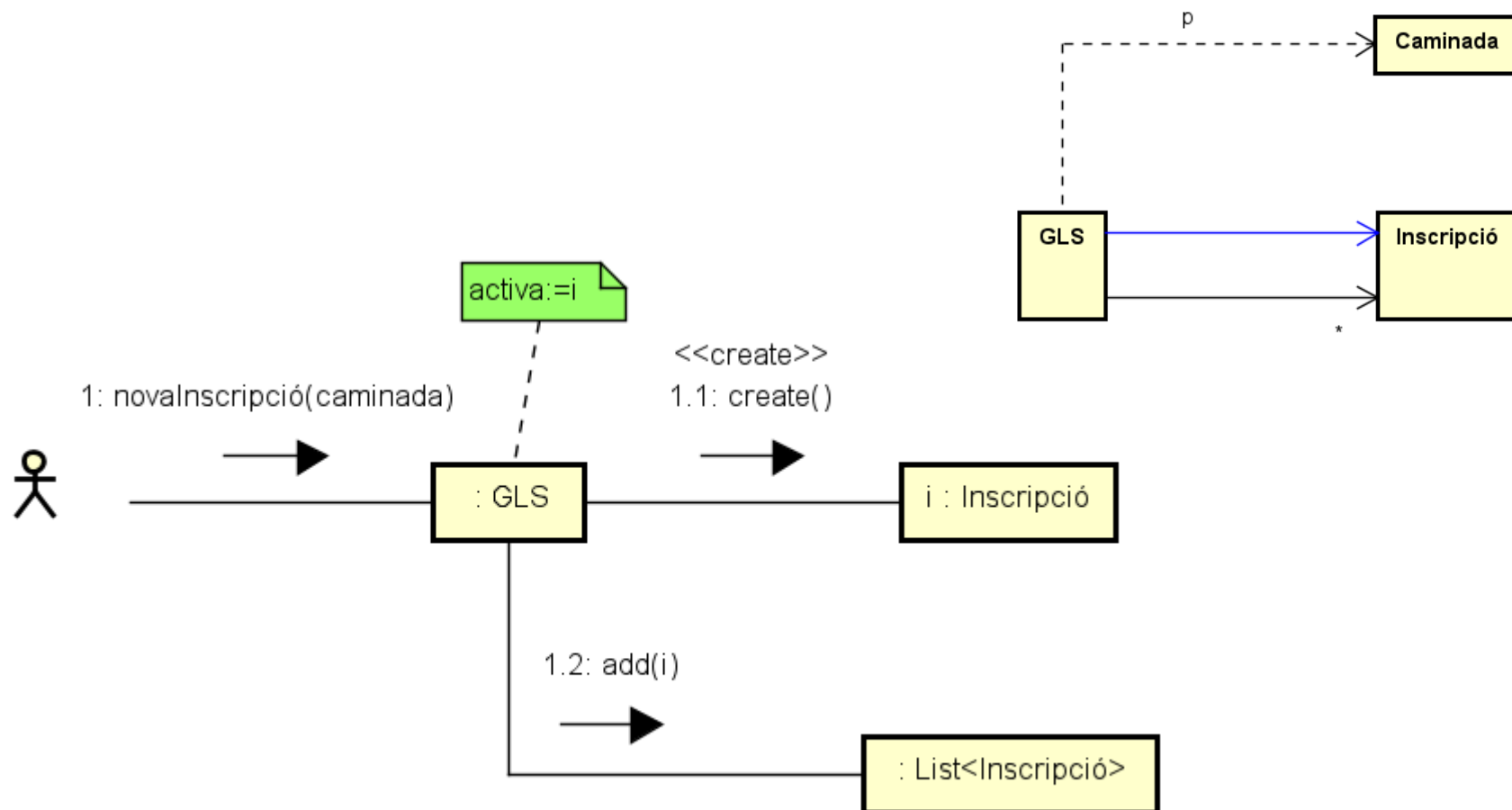
13

- En mantenir la inscripció activa amb visibilitat d'atribut monoavaluada, tenim que el controlador està acoblat a la inscripció
- Per tant el **creador efectiu** d'inscripció no genera cap nou acoblament
 - La inscripció activa significa acoblament a **una** inscripció
 - L'emmagatzematge significa acoblament al **conjunt** d'inscripcions

Han canviat les condicions d'anàlisi que ens portaren a preferir M2 per sobre de M1

M1: Disseny

14



M2: i es manté activa

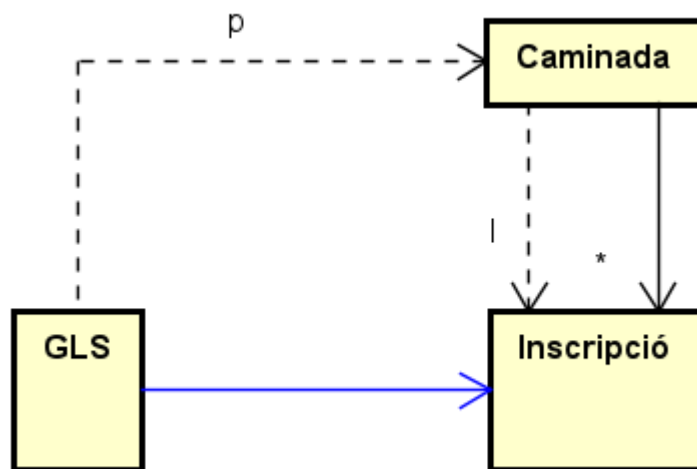
15

- Usem les mateixes observacions que hem fet per **M1**:
 - POST: cal donar al controlador **visibilitat d'atribut monoavaluada** de la `i:Inscripció` acabada de crear:
`gls → i:Inscripció`
 - En aquest cas el controlador no té visibilitat d'emmagatzematge sobre les inscripcions:
`Caminada → Inscripció*`
 - Per tant el més simple és dotar al controlador de **visibilitat directa monoavaluada** de la `inscripció` activa.

La **visibilitat d'activació** és una visibilitat totalment nova

M2. Activació directa

16



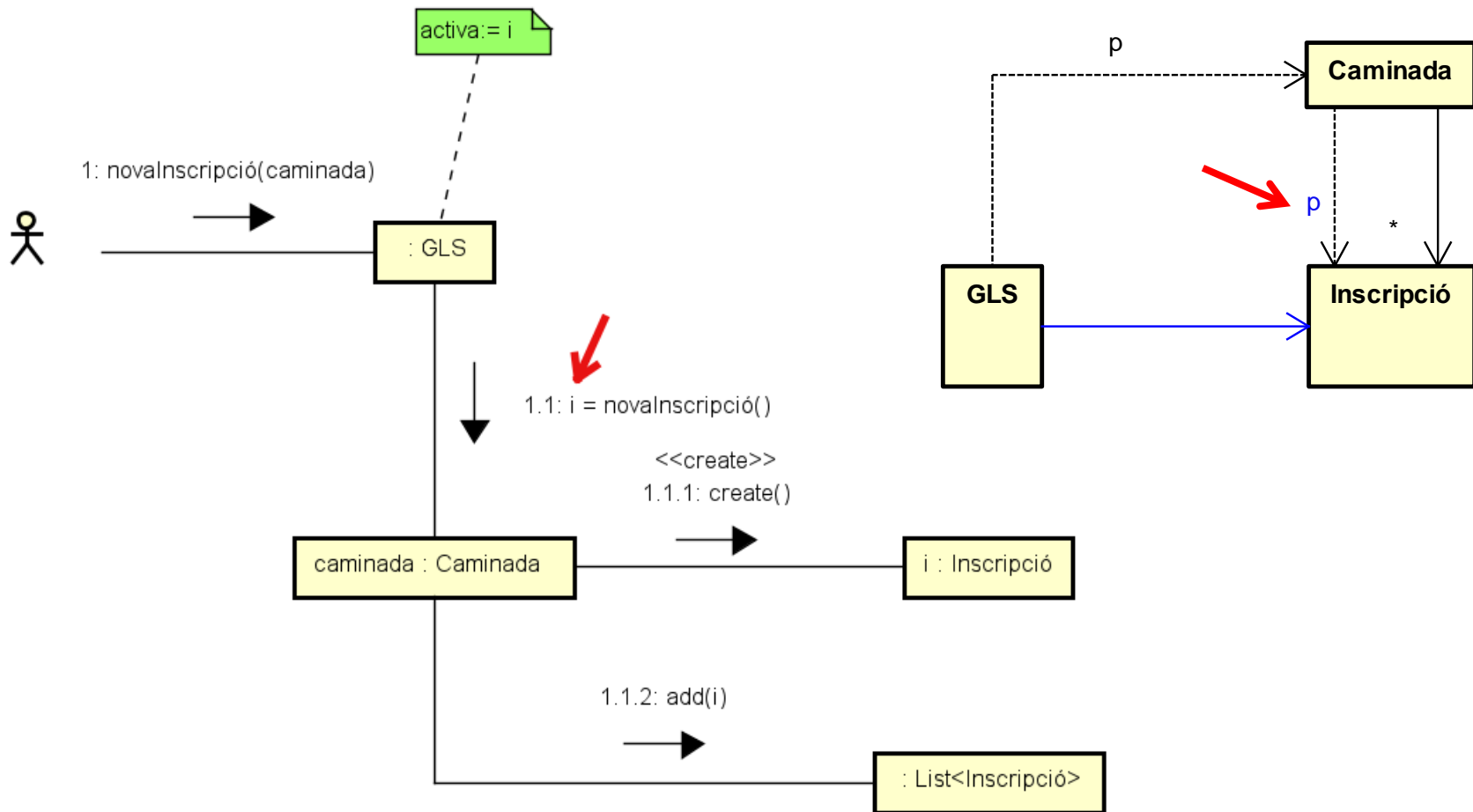
M2: Comunicació necessària per a l'activació

17

- De moment l'únic mecanisme d'activació que hem vist és l'ES *novaInscripció (camínada)*
 - La POST demana haver creat una nova *inscripció* i haver-la deixada activa
 - Per tant és el creador d'*inscripció* qui ha de comunicar aquesta al controlador, per tal que el controlador mantingui **visibilitat directa d'activació** de la *inscripció* acabada de crear

M2. Disseny amb activació directa

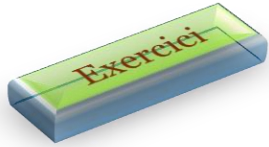
18



M2: Creador que comunica

19

- Responsabilitats de Caminada respecte la Inscripció
 - Crear i emmagatzemar (és un creador canònic)
 - Comunicar enrere
- Conseqüències
 - El controlador queda acoblat a la Inscripció
 - ✦ La decisió de mantenir l'activació amb visibilitat directa d'atribut ja ho implicava
 - ✦ Caminada té visibilitat de paràmetre d'inscripció (a part de la d'atribut per l'emmagatzematge)



Exercici

20

- Decisió de disseny que volem mantenir:
 - El controlador té visibilitat d'atribut directa monoavaluada de la Inscripció activa
- Davant d'aquesta decisió de disseny analitzeu en termes d'acoblament i cohesió la bondat de M1 i M2



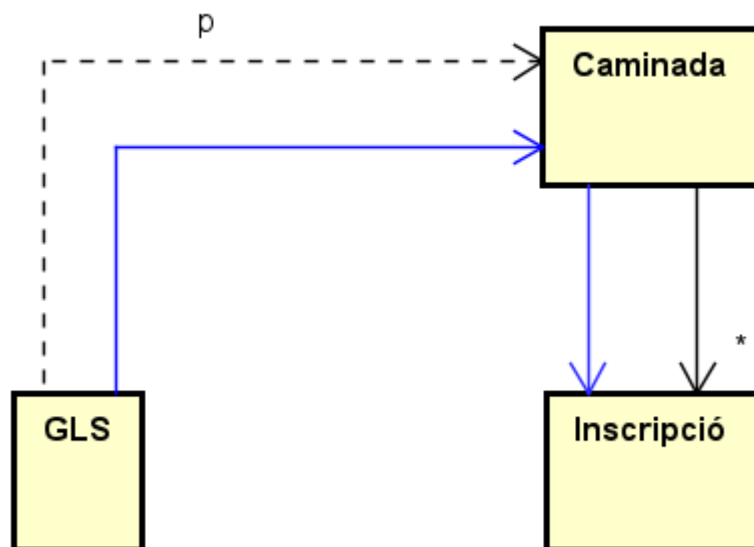
M2: Activació per visibilitat indirecta

21

- Alternativa a la visibilitat directa de la inscripció activa des del controlador
 - El controlador manté **visibilitat indirecta** de la inscripció activa
 - ✦ La caminada és l'intermediari desitjat
- Avantatges de l'alternativa
 - No s'introdueixen nous acoblaments
 - No hi ha necessitat de comunicar l'activació

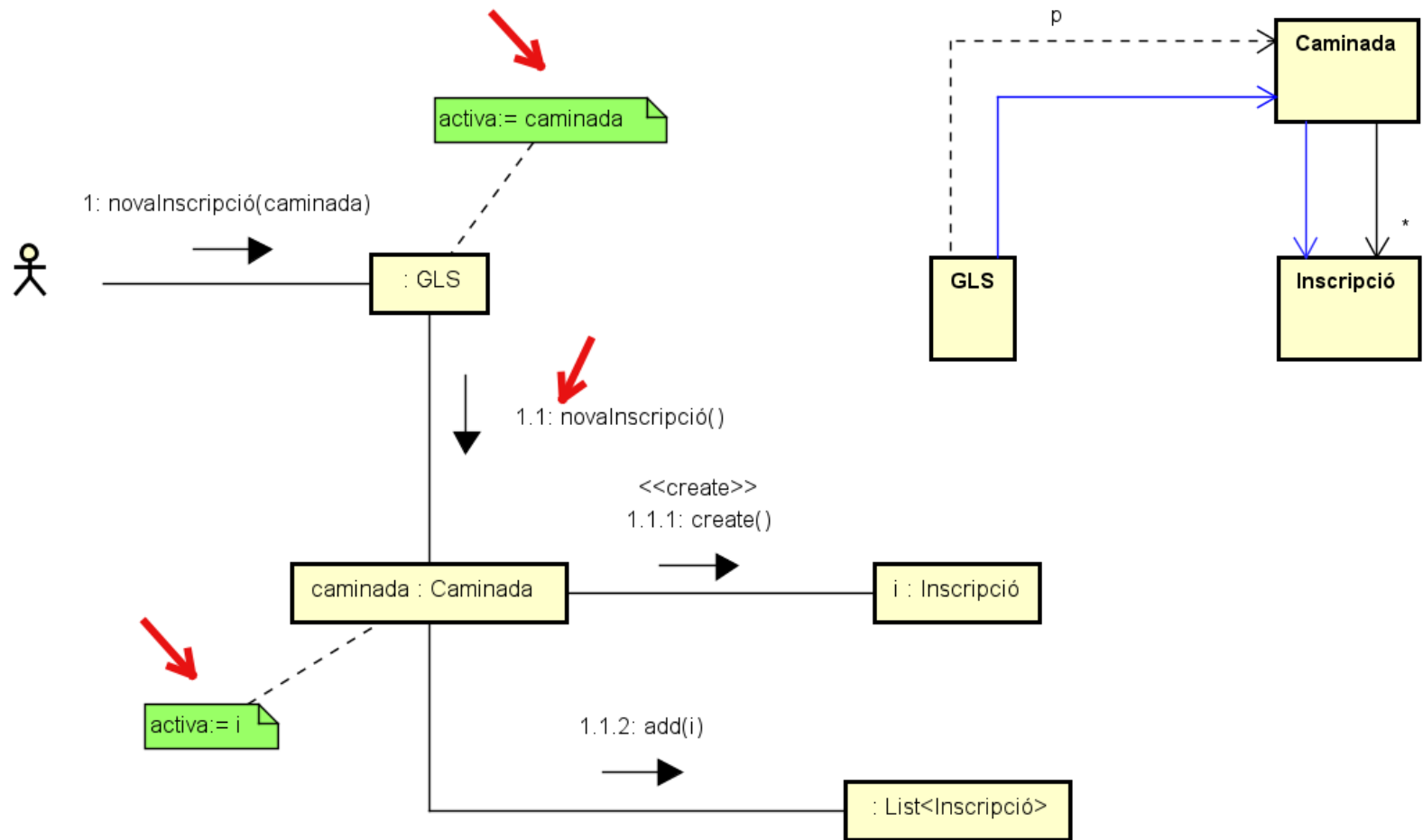
M2. Visibilitat indirecta

22



M2. disseny amb visibilitat indirecta

23



M2: Inconvenients de la visibilitat indirecta

24

- Ineficiència en temps
 - Tota petició sobre la inscripció activa haurà de seguir la cadena
controlador → caminada → inscripció
- Múltiple activació
 - La visibilitat directa des de caminada de la inscripció activa exigeix una visibilitat d'atribut caminada → inscripció
 - Tota caminada manté una inscripció activa
 - ✦ Ineficiència en espai

- En el moment de dissenyar, les qüestions d'eficiència s'han de mantenir en un segon pla
 - Cal fugir de l'eficiència prematura
 - El disseny no pot estar condicionat, inicialment, per qüestions d'eficiència
 - ✦ L'eficiència local no ens assegura l'eficiència global
- **Primer el disseny; després l'eficiència**

Desactivació

26

Activació i desactivació

27

- **Activació d'una realització**
 - Defineix un argument implícit
 - Exigeix una visibilitat monoavaluada d'atribut visible des del controlador
- **Desactivació d'una realització**
 - Esborra qualsevol traça d'un argument implícit
 - Exigeix trencar l'enllaç dirigit d'activació pertinent

Motius per a la desactivació

28

- Evitar operacions errònies
 - Acabada la tasca amb un argument implícit, esborrem la traça d'aquest per evitar que es pugi realitzar de nou la tasca sense haver canviat el valor de l'argument implícit
- Evitar seqüències errònies d'ES
 - L'activació serveix tant per a la comunicació entre ES com pel control de la seqüència
 - ✦ Ex. `fiInscripció()`

Com desactivar

29

Cal trencar **totes les baules** de la cadena d'enllaços dirigits que mantenen la realització activa

Com trencar una baula

30

- Una baula és un enllaç dirigit: $a \rightarrow$
 - Per trencar un enllaç dirigit el que fem és modificar la destinació de l'enllaç
 - ✦ Una nova destinació: $a \rightarrow x$
 - Hem canviat l'enllaç, però no l'hem trencat
 - Es poden produir efectes laterals indesitjables
 - Exemple: Noves cadenes
 - ✦ Una destinació fictícia: $a \rightarrow \text{null}$
 - No es produeixen cadenes indesitjades

Semàntica del null

31

- `null` és un valor present en la majoria de llenguatges de programació
 - Tot i que el tracten de diferents maneres
- És un objecte
 - Es pot usar com a destinació d'un enllaç dirigit
 - ✦ En termes de llenguatge de programació, es pot assignar sobre una variable
- No és un objecte
 - No pot enviar ni rebre missatges; no és emmagatzemable
 - ✦ En termes de llenguatge de programació, només es pot assignar sobre una variable

Hi ha tants
`null`s
diferents com
components

El perquè del mecanisme

32

- **Hi ha algun altre mecanisme per “trençar” un enllaç?**
 - De fet no hem trencat l'enllaç, l'hem modificat
 - El trencament de l'enllaç no pot afectar les seves potencialitats
 - Crear un enllaç dirigit en els llenguatges de programació ho fem via assignació
 - `origen:=destinació`
 - Trençar l'enllaç significa des-assignar
 - ✦ La majoria de llenguatges de programació no permeten la desassignació

Desactivar la inscripció

33

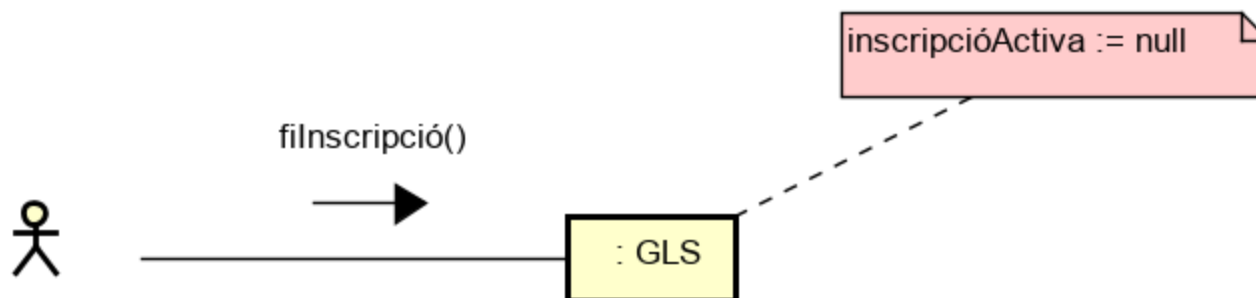
- **ES** `fiInscripció()`
 - POST: \nexists cap inscripció activa
- **M1**
 - Activació per visibilitat directa
- **M2**
 - Activació per visibilitat directa, o
 - Activació per visibilitat indirecta a través de `Caminada`

Hem bandejat aquesta opció per *Baix Acoblament*

Visibilitat directa

34

- Responsabilitat
 - Assignar a null la destinació de l'enllaç d'activació
 - ✦ `gls.InscripcióActiva:=null`
- Expert
 - El controlador, que és qui manté la visibilitat d'activació
- Disseny



Visibilitat indirecta

35

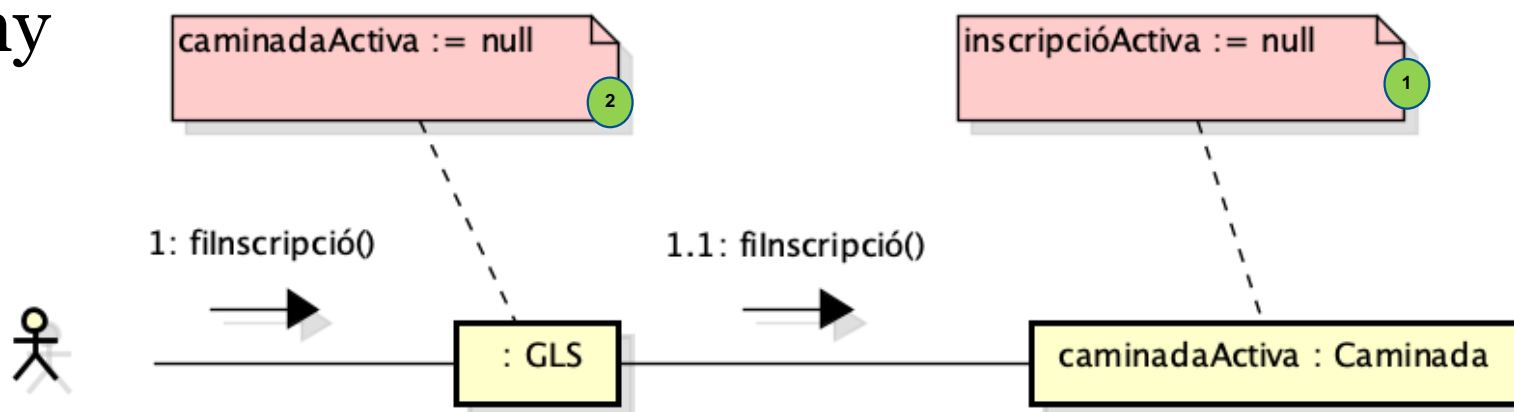
- Responsabilitat

- Trencar la cadena $\rightarrow \text{caminada} \rightarrow \text{inscripció}$
- Assignar a `null` la destinació de l'enllaç d'activació
 - ✦ `gls.InscripcióActiva:=null`

- Expert

- El controlador, que és qui manté la visibilitat d'activació

- Disseny



L'activació en els contractes

36

Contracte de l'especificació

37

- Les activacions expressen
 - Arguments implícits
 - Comunicació entre els ES
- Sempre però en termes del problema i de les necessitats d'aquest

Activació en l'especificació

38

- Tenim un ES d'introducció d'una participació en una inscripció
- Aquest ES l'iterem, i volem aconseguir que dins d'una mateixa iteració sempre usem la mateixa inscripció
- Per fer-ho, factoritzem la inscripció, i la donem en un ES independent
- D'aquesta manera en la introducció d'una participació la inscripció és un **argument implícit**
- La nostra manera de modelitzar el problema exigeix com a PRE de novaParticipació l'exigència d'una **inscripció activa**

Contracte del disseny

39

- Cada **activació** de l'especificació en el disseny la podem materialitzar amb una **visibilitat monoavaluada d'atribut** directa o indirecta
- En el cas d'usar una activació indirecta en el disseny apareixen activacions intermèdies
 - Per mantenir actiu un objecte hem de mantenir actius tots els objectes que formen part de l'enllaç dirigit indirecte d'activació



Activació indirecta en el disseny

40

- Decidim mantenir la inscripció activa a través de la **visibilitat indirecta**

GLS→Caminada→Inscripció

- GLS manté una :Caminada activa

- GLS→Caminada

- ✦ Cada :Caminada manté una :Inscripció activa

- ✦ Caminada→Inscripció

Activacions intermèdies

41

- Les activacions intermèdies són conseqüència de la visibilitat indirecta
 - Són una necessitat del disseny, i no pas de l'especificació
- Els contractes es construeixen durant l'especificació
 - Una altra cosa és que des del disseny els llegim en termes de la solució
- **Conclusió.** Les activacions intermèdies no han de figurar en els contractes

Contracte de l'especificació

42

- **novaInscripció (c:Caminada)**
 - PRE
 1. No existeix cap :Inscripció activa
 - POST
 1. Existeix una nova i:Inscripció
 2. i:Inscripció es manté activa
- **novaParticipació (nom)**
 - PRE
 1. Existeix una i:Inscripció activa
 - POST
 1. i:Inscripció es manté activa

Contracte del disseny

43

- **novaInscripció (c:Caminada)**

- PRE

1. No existeix cap :Caminada activa
2. No existeix cap :Inscripció activa

- ✦ POST

1. Existeix una nova i:Inscripció
2. c:Caminada es manté activa
3. i:Inscripció es manté activa **per a c**

- **novaParticipació(nom)**

- PRE

1. Existeix una c:Caminada activa
2. Existeix una i:Inscripció activa **per a c**

- POST

1. c:Caminada es manté activa
2. Inscripció es manté activa **per a c**

El fragment en **verd** emfatitza la idea d'activació indirecta o activació relativa la baula anterior

La problemàtica dels dos contractes

44

- Inconsistència
 - Cal tenir present que sempre han de ser consistents
 - ✦ Si canviem el contracte de l'especificació també cal canviar el del disseny
- Exigència de context
 - Sempre que ens donin un contracte ens han de dir si és el del disseny o el de l'especificació
- Complexitat
 - Cal tenir present de què va a cada contracte, tant per la seva lectura com per a la seva escriptura

Solució amb un sol contracte

45

- Mantenim un sol contracte
 - El de l'especificació
- En el disseny expressem de manera explícita com mantenim cadascuna de les activacions de l'especificació
 - En una eina CASE podem definir una capa de disseny sobre el contracte
 - ✦ En aquesta capa expressem les necessitats del disseny

Contracte de l'especificació

46

- **novaInscripció (c:Caminada)**
 - PRE
 1. No existeix cap `:Inscripció` activa
 - POST
 1. Existeix una nova `i:Inscripció`
 2. `i:Inscripció` es manté activa
- **novaParticipació (nom)**
 - PRE
 1. Existeix una `i:Inscripció` activa
 - POST
 1. `i:Inscripció` es manté activa



L'activació de la `:Inscripció` es fa per visibilitat indirecta a través de `:Caminada`

La nostra proposta

47

- Només tenim **un contracte**, el de l'especificació
- Les necessitats del disseny les expressem explícitament d'alguna manera