

Empententa final

Activitat de disseny. INEP. Curs 20181

Josep M Merenciano
meren@cs.upc.edu
Departament de Ciències de la Computació
EPSEVG-UPC

15 de gener de 2019

Sumari

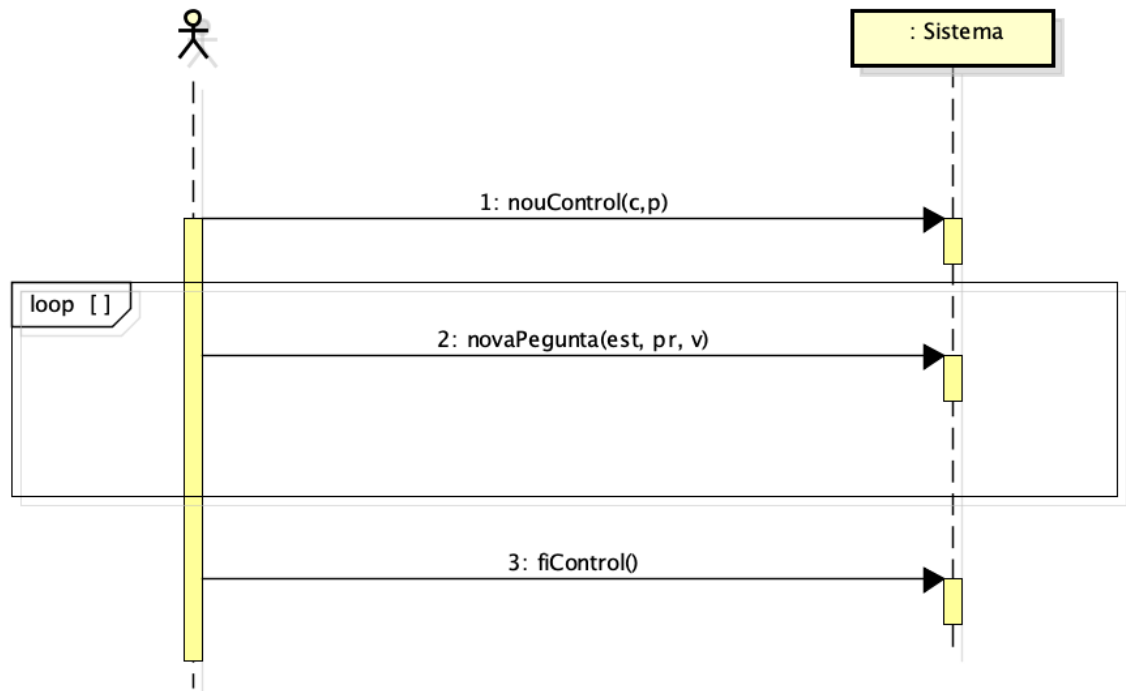
Enunciat	3
1 <i>TEORIA</i> Dels Diagrames de comunicació al Model de components	7
1.1 Una observació preliminar	7
1.2 Lectura dels diagrames de col·laboració (DC)	7
1.2.1 Qüestió de noms	7
1.2.2 Objectes vs identificadors	9
1.2.3 Components	9
1.2.4 Multiobjectes	9
1.2.5 Visibilitats	10
1.2.6 Multiobjectes i comunicació	11
1.3 Construcció del Model de components	12
1.3.1 Anàlisi dels DC	12
1.3.2 Dibuix de MComp	13
1.4 Contractes a partir del disseny	14
1.5 MC per enginyeria inversa	16
2 Solució	19
2.1 Guia de lectura	19
2.2 Model de components	19
2.2.1 Detall del procés: <code>nouControl()</code>	19
2.2.2 Detall del procés: <code>novaPregunta()</code>	23
2.2.3 Detall del procés: <code>fiControl()</code>	34
2.2.4 MComp	43
2.3 Contractes des del disseny	43
2.3.1 Contractes segons la semàntica dels DC	43
2.3.2 Correctesa i robustesa	49
2.4 MC per enginyeria inversa	54
2.4.1 Procés d'enginyeria inversa per a obtenir MC	54
2.4.2 El model conceptual (MC) resultant	64
2.5 Conclusions sobre l'enginyeria inversa	64

Activitat de disseny. INEP

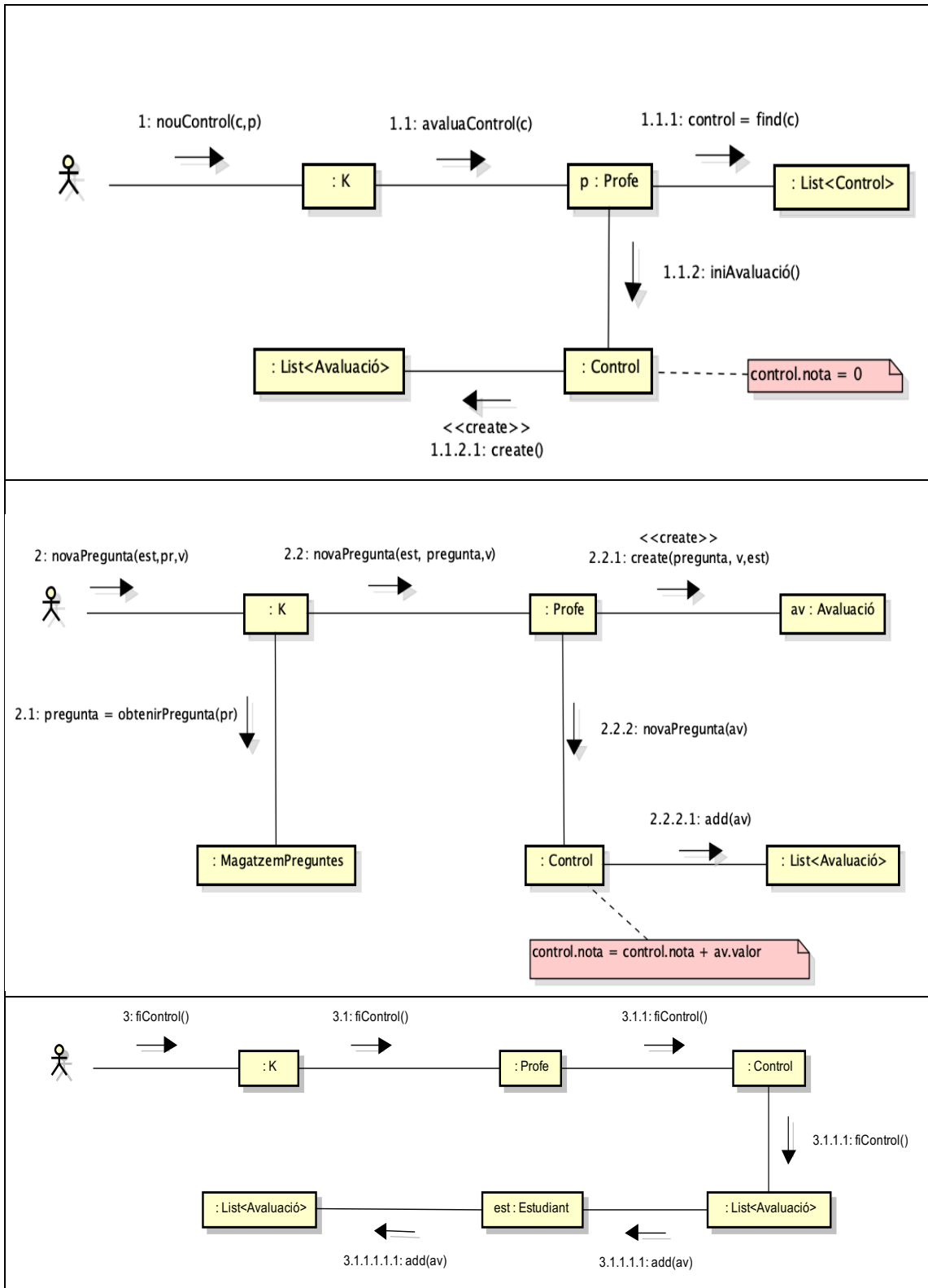
15 de novembre de 2018.

Empenteta Final va consolidant-se com a projecte. Per això, en *Mestre Tites*, que n'és el promotor, fa uns dies va contractar uns analistes per tal que fessin un disseny robust del sistema informàtic necessari per gestionar els exàmens i controls. Però per error va contractar uns impresentables que van desaparèixer d'un dia per l'altre. Revisant el que han deixat hem trobat documentació parcial i dispersa. En *Mestre Tites* ens demana que documentem el sistema que aquests analistes van deixar.

Revisant la documentació de l'especificació hem trobat aquest diagrama, corresponent al cas d'ús **introduirControl**.



La documentació del disseny conté els següents diagrames:



Es demana:

1. **MComp.** Construïu el model de components del sistema
 - Indiqueu totes les visibilitats: d'atribut, local i de paràmetre.
 - Indiqueu els mètodes i atributs de cada component
 - Per cada paràmetre dels ES digueu si és un literal, un identificador o una entitat
2. **Especificació.** Doneu l'especificació dels diferents esdeveniments de sistema
 - Tant pels arguments com pels diferents elements que apareguin en el contracte, indiqueu quins elements són realitzacions d'un component, i quins són simples identificadors.
 - Desconeixem el problema, i tenim part de la solució. El que es tracta és d'intentar reconstruir el problema (la seva especificació) a partir del fragment de solució donat
3. **Enginyeria inversa.** Tenint en compte el model de components obtingut, quin creieu que hauria de ser el model conceptual? Dibuixeu MC.

- *La separació en preguntes és per facilitar-vos la feina, i no correspon a cap criteri numèric. L'exercici es valora en tota la seva totalitat i no pas donant punts acumulatius als diferents apartats.*
- *Els punt interns de cada pregunta recorden coses que no us heu d'oblidar, però no els heu de llegir com la llista exhaustiva d'allò que es demana.*

Expliqueu i justifiqueu tot el que feu

Data màxima de lliurament: **20 de desembre de 2018**

Format de lliurament: **En paper**

Tipus de lliurament: **Individual**

Lloc de lliurament: **Bústia del despatx 120**

1 *TEORIA* Dels Diagrames de comunicació al Model de components

1.1. Una observació preliminar

En el que segueix es presenten de *forma esquemàtica* els aspectes teòrics necessaris per llegir el DC, i per construir a partir d'ells el MComp.

El redactat i l'estructuració permet usar-ho com *material de referència*. El preu a pagar és que el resultat pot resultar més feixuc de llegir del que potser caldria.

La lectura simultània, en paral·lel, d'aquests apunts teòrics i de la solució pas a pas de més avall ha donar llum al redactat.

1.2. Lectura dels diagrames de col·laboració (DC)

1.2.1 Qüestió de noms

Àmbit de visibilitat

- En principi cada DC és un àmbit de visibilitat independent
 - Res impedeix usar el mateix nom en DC diferents per a referenciar diferents elements
- Malgrat tot, sovint s'usa la coincidència de noms en DC diferents per expressar que hi ha una comunicació entre els ES
 - És a dir, que l'element és el mateix en ambdós DC
- *Davant d'un nom compartit entre dos DC cal:*
 - Observar si hi ha algun motiu clar per considerar que són elements diferents
 - * Per exemple, en un DC un nom s'usa per un identificadr, i en un altre DC s'usa per referenciar un objecte

- Determinar si per la semàntica del problema té sentit que siguin el mateix element
 - * Exemple. En un ES cerquem un objecte, i en el següent ES iterat, l'anem modificant

Anòmies

- Un objecte sense nom és una [anòmia](#)
- L'aparició d'una anòmia en un DC ot significar que:
 - Estem davant d'un singleton
 - En el DC hi ha prou context per saber de quin objecte estem parlant
 - * En aquest cas el nom és redundant, i per simplicitat el dissenyador ha preferit ometre'l
 - * Compte però: l'objecte té nom; simplement no l'explicitem
- Generalment el component que rep els ES és un singleton
- La resolució de les anòmies (això és, decidir quin és el nom no explicitat) es fa tenint en compte:
 - La semàntica del problema
 - * Hem creat un objecte i ara li volem delegar determinada tasca, significa que l'objecte creat i l'objecte que rep el missatge de delegació han de ser el mateix
 - Necessitats de visibilitat
 - * Suposem que per poder enviar un determinat missatge necessitem un enllaç dirigit amb destinació :B
 - * Suposem que en tot el nostre disseny del CU estem usant un únic objecte :B, que en un altre moment hem anomenat b
 - * Llavors l'objecte :B que és destinació de l'enllaç dirigit considerat només pot ser b

1.2.2 Objectes vs identificadors

Objectes

- Tot aquell qui en un **diagrama de col·laboració (DC)** rep un missatge és un **objecte**
- El `find()` retorna objectes
- L'argument d'un `add()` és un **objecte**

Identificadors

- Els arguments del `find()` són **identificadors**

1.2.3 Components

Detecció de components

- Tot objecte que apareix en algun **diagrama de col·laboració (DC)** és realització d'un **component**.
 - Els **multiobjectes** són una ficció emprada pels DC. No són la realització de cap component.
- Tot identificador ho és d'algun component
 - El **nom** del component és el que apareix en el mateix DC

1.2.4 Multiobjectes

Els multiobjectes en els models

- Els **multiobjectes** s'expressen en els **DC** en forma de pila d'objectes, o com un objecte de tipus `<List>B`
- Els **multiobjectes** no s'expressen en forma de component a **MComp**.
- La presència d'un **multiobjecte** en un **DC** indica una **visibilitat multiavaluada** a **MComp**

1.2.5 Visibilitats

Visibilitats de paràmetre

- Les **visibilitats de paràmetre** s'obtenen directament de les interfícies dels diferents components
 - D'aquí la importància d'explicitar quins són els missatges que pot rebre un component; i si els arguments del missatge són valors, identificadors o objectes

Visibilitats d'atribut

Definició visibilitat d'atribut

- Les visibilitats d'atribut tenen una vigència superior a la del context d'emissió

Casos particulars de visibilitat d'atribut

- **Visibilitats d'activació**
 - Per comunicar ES diferents
- **Visibilitat d'emmagatzematge**
 - Un ES emmagatzema un objecte que serà usat en futurs ES, del mateix CU o no
 - Els **magatzems**, **repositoris** o similars, són un cas especial d'emmagatzematge. Són objectes la finalitat dels quals és emmagatzemar-ne d'altres
 - * Una visibilitat local sobre el magatzem significa que tot el que aquest emmagatzema s'ha de crear i destruir en aquest àmbit local
 - * Per tant *cal tenir visibilitat d'atribut sobre els magatzems.*
 - * Tot magatzem té una **visibilitat multiavaluada d'atribut** sobre el component que emmagatzema
 - Tota dada usada en una creació ha de ser recuperable
 - **Visibilitat de recuperació**
 - * Tota dada usada per a l'emmagatzematge ha de ser recuperable

Visibilitats locals

- Tota visibilitat que no sigui de paràmetre i que no tinguem raons per considerar-la d'atribut és una visibilitat local

1.2.6 Multiobjectes i comunicació

Recepció de missatges

- Quan s'envia un missatge a un multiobjecte o és un missatge de grup o és un missatge d'enllaç
- **Missatge de grup**
 - Són una llista tancada de missatges, que afecten el multiobjecte
 - La semàntica de cadascun dels missatges està definida prèviament
 - La llista engloba:
 - * `find()`
 - * `add()`
 - * `create()`
- **Missatge d'enllaç**
 - Tot missatge que no sigui de grup és d'enllaç
 - Qui rep el missatge d'enllaç és cadascun dels objectes que conté el multiobjecte
 - El nom, els arguments, i la semàntica dels missatges d'enllaç les defineix el dissenyador

Emissió de missatges

- Si en un DC un multiobjecte emet un missatge, qui realment l'emeta és cadascun dels objectes del multiobjecte
- Els arguments explicitats en un missatge que emet un multiobjecte poden ser:
 - **Fix**
 - * Cada objecte del multiobjecte, en enviar el missatge usa el mateix valor com a argument que han usat els altres

objectes del multiobjecte

– Variable

- * Cada objecte del multiobjecte, en enviar el missatge pot usar un valor diferent com a argument del que han usat els altres objectes del multiobjecte
- * Com a argument variable es pot enviar el propi emissor. Per exemple, un multiobjecte `List<A>` pot enviar com a argument `a:A` (o `a` si no es pot produir confusió) per indicar que s'envia a si mateix
- * Similarment podem enviar un atribut de l'emissor. Per exemple, un multiobjecte `List<A>` pot enviar com a argument `a.attr` per indicar que cada objecte emissor envia el seu valor de l'atribut `attr`

Quants multiobjectes

- Un multiobjecte no és un component
- Donat un component `A` podem tenir tants multiobjectes `List<A>` com convingui
- Cal llegir atentament els DC per discernir si dos multiobjectes `List<A>` són dos multiobjectes diferents o no

1.3. Construcció del Model de components

1.3.1 Anàlisi dels DC

Procediment de construcció de MComp

- Per cada missatge de cada diagrama de col·laboració cal analitzar i estudiar, no necessàriament en aquest ordre:
 1. La comunicació és possible
 2. Arguments d'entrada
 3. Arguments de sortida

Comunicació possible

- Hi ha una visibilitat que permet l'enllaç per on es transmet el missatge
- Tot argument del missatge és accessible des de l'emissor
 - És un literal, un identificador, o bé l'emissor en té una visibilitat d'algun tipus

Arguments d'entrada

- Cal analitzar la tipologia de cada argument:
 - **Valor o literal**
No ens diu res
 - **Identificador**
 - * Ens indica l'existència d'un component, però no ens diu res de visibilitats
 - **Objecte**
 - * L'emissor del missatge n'ha de tenir algun tipus de visibilitat
 - * El receptor del missatge en té una àmetrevisibilitat de paràmetre

Arguments de sortida

- L'emissor del missatge té una visibilitat de l'objecte retornat
- El receptor del missatge té una visibilitat (d'atribut o de paràmetre) sobre l'objecte retornat
- Cal tenir en compte que tot **missatge de creació** (`create()`) té com a **argument de sortida** l'objecte creat)

1.3.2 Dibuix de MComp

Visibilitats vs dependències

- **Dependència:** Visibilitat local o de paràmetre
 - Es dibuixen en línia discontinua: `-- >`
- **Visibilitat d'atribut**

- Es dibuixen en línia contínua: →

1.4. Contractes a partir del disseny

Idea bàsica

- Cal assegurar que tot missatge emès té les condicions per ser emès
 - Això afecta a les PRE
- Cal explicar què fa cada missatge
 - Això afecta a les POST
- Cal propagar aquestes condicions als ES
 - De cara a l'especificació només ens interessen els contractes dels ES

Condicions que impedeixen l'emissió d'un missatge

- No existeix l'enllaç dirigit pel qual es transmet
 - Cal una PRE que assegurí l'existència de l'enllaç dirigit
- Per a algun dels seus arguments no existeix l'enllaç dirigit que permet accedir-hi
 - Cal una PRE que assegurí l'existència de l'enllaç dirigit

Utilitat dels arguments

- Tot argument (sigui d'entrada o de sortida) ha de tenir alguna utilitat. És a dir, s'ha d'usar, comunicar o emmagatzemar
 - **Ús.** Li enviem un missatge
 - **Comunicació.** Es passa com a argument d'un missatge, o es retorna com a argument de sortida del missatge
 - **Emmagatzematge.** En el cas dels multiobjectes hi ha un `add()`. En el cas dels emmagatzematges multiavaluats sovint no s'explicita l'emmagatzematge perquè es pot concloure per exclusió: si no l'usem ni el comuniquem, és que l'estem emmagatzemant

Un únic MComp

- Independentment dels DC que tinguem, **només hi ha un MComp**
- Això significa que les conclusions que extraiem d'un DC han de ser compatibles amb la dels altres DC

Casos típics de PRE

Visibilitat d'activació

- Es tracta per comunicar dos ES: un crea l'enllaç dirigit, i l'altre l'usa
- L'usuari de l'enllaç dirigit (qui l'usa per accedir a un argument, o per transmetre un missatge) necessita una PRE demanant l'existència de l'enllaç dirigit

`find()`

- Sovint es fa un `find()` i tot seguit s'usa el valor rebut
- Si el DC no comprova que aquest valor no sigui nul, llavors cal una PRE demanant l'èxit del `find(id)`

$$\exists a : A \text{ tal que } a.id = id$$
Contextualització del `find()`

- Si tots els objectes `:A` s'emmagatzemen en un mateix multiobjecte, n'hi h prou amb:

$$\exists a : A \text{ tal que } a.id = id$$

- Si un multiobjecte només emmagatzema alguns dels objectes `:A` llavors cal relativitzar la petició:

$$\exists a : A \in \text{multiobjecte tal que } a.id = id$$

Casos típics de POST

`add ()`

- L'`add ()` afegeix un objecte a un multiobjecte. Tot multiobjecte és un conjunt
- Cal assegurar, en forma de PRE si el DC no ho assegura prèviament, que l'objecte que es vol afegir no s'ha afegit anteriorment
- L'anàlisi es pot reduir a dos casos:
 - **Objecte nou**
 - * Cada nou objecte (aquell obtingut amb un `new`) té un identificador diferent
 - * Per tant no cal afegir cap PRE
 - **Objecte pre-existent**
 - * Hem obtingut, amb un `find ()` o similar, un objecte, que ara volem afegir en un multiobjecte
 - * Cal una PRE per assegurar que l'objecte a afegir no s'ha afegit prèviament

1.5. MC per enginyeria inversa

Procediment

1. Supressió dels emmagatzemadors

- Cal suprimir tot component tal ue la seva única responsabilitat sigui l'emmagatzematge
 - L'emmagatzematge és propi del disseny, però no té cap mena de sentit en l'especificació
 - Sovint els components dedicats a l'emmagatzematge s'anomenen *magatzem*, *repositoris*, *compendi*, etc
- Per suprimir-los cal curtcircuitar les visibilitats d'atribut incidents amb les sortints (les dependències no ens interessin)
 - Suposem que M és un magatzem, i que tenim les visibilitats $A \rightarrow M^*$ i $M \rightarrow B$

– Com a resultat de la suessió tindrem: $A \rightarrow B$

2. **Suprimim les dependències**

- Ens quedem només amb les visibilitats d'atribut

3. **Convertim les visibilitats en associacions**

- Això implica trobar-ne les multiplicitats i dotar-les de semàntica

4. **Suprimim les activacions locals al CU**

- Determinem quines són les associacions que provenen d'una visibilitat d'activació
- Analitzem si l'activació té sentit que es mantingui fora del CU o no. En cas negatiu, suprimim l'associació

5. **Analitzem les obligatorietats i optativitats**

Detecció de multiplicitats i optativitats

Fonts d'informació

- Cada problema és diferent. En general però cal cercar la informació en:
 1. DC
 2. Problema
 3. Preferències generals

Anàlisi dels DC

- Cal analitzar la semàntica dels DC
 - Què fan
 - Quin és el seu propòsit
 - Quines precondicions pressuposen
 - * Exemple. Si fem un `find()` i no comprovem si ens ha donat un resultat és perquè assumim que en tot moment hi ha un objecte amb el valor d'identificació donat

Anàlisi del problema

- Cal recórrer al nostre coneixent del problema i analitzar si hi ha alguna PRE que ens afecti
 - Exemple. Una pregunta pot estar en molts controls
- Cal tornar a analitzar els DC assumint les PRE trobades

Preferències generals

- Hi ha casos on ni la lectura dels DC ni el nostre coneixement del problema ens donen la informació necessària
- Dit amb d'altres paraules, hi ha casos on, per exemple, res ens permet decidir entre la multiavaluació o la monoavaluació d'una associació
- En aquests casos recorrem a preferències generals
 - Per exemple, quan dues opcions siguin possibles, ens decantem per la menys marcada
 - * Aquest criteri obliga a definir qui és més marcat que un altre

2 Solució

2.1. Guia de lectura

La construcció de MComp, l'obtenció dels contractes per enginyeria inversa o la construcció del MC també per enginyeria inversa, són de [lectura independent](#).

En el detall del procés de construcció del Model de components es fan explicacions que davant els aspectes teòrics expressats més amunt potser no caldrien. S'ha preferit però la [reiteració](#) per facilitar l'aprenentatge.

Es recomana que a mesura que es vagi llegint la solució, es vagin repassant els [aspectes teòrics](#) involucrats.

2.2. Model de components

2.2.1 Detall del procés: `nouControl()`

Components i primers objectes observant `nouControl(c,p)`

- Els objectes explícits del DC ens porten a considerar els següents [components](#):
 - K
 - Profe
 - Control
 - Avaluació
- El multobjecte `List<Control>` diu que hem de tenir una visibilitat multiavaluada sobre un component `Control`
- El multobjecte `List<Avaluació>` diu que hem de tenir una visibilitat multiavaluada sobre un component `Avaluació`. D'aquí que tenim `Avaluació`

- `c` és l'argument d'un `find()`. Per tant és un **identificador**. El multiobjecte on es fa el `find()` és de `Controls`; per tant l'identificador `c` ha de permetre referenciar un `:Control`
- `p` està etiquetat com un **objecte** realització del component `Profe`
- `control` és un **objecte**, ja que és el valor de retorn d'un `find()`. El component del qual n'és realització és `IControl`, ja que el `find()` el realitzem sobre un multiobjecte de `:Controls`

Primers resultats

Símbols del DC de `nouControl()`

<code>c</code>	identificador d'un objecte <code>:Control</code>
<code>p</code>	objecte , realització de <code>Profe</code>
<code>control</code>	objecte , realització de <code>Control</code>

Visibilitats

- $K \xrightarrow{par} Profe$
 - Pel missatge 1 i el fet que p és un objecte
- $Profe \xrightarrow{?} Control^*$
 - Pel multiobjecte
 - De moment no podem dir quin tipus de visibilitat és
- $Profe \xrightarrow{loc} Control$
 - Missatge 1.1.1. L'emissor del missatge `find()` té una visibilitat sobre l'objecte `:Control` retornat
 - L'objecte retornat s'usa per a enviar-li el missatge 1.1.2. Així, la visibilitat $Profe \rightarrow Control$ no és de paràmetre; no tenim de moment cap motiu per considerar que hi ha emmagatzematge; i per tant conclouem que és local
 - * El raonament parteix de la base que el `l:Control` a qui p envia un missatge és el mateix `control:Control` que ha obtingut amb el `find()`. Més endavant discutim aquesta afirmació
- $Control \xrightarrow{?} Avaluació^*$
 - Pel missatge de creació 1.1.2.1
 - De moment no podem dir quin tipus de visibilitat és

Comprovació

- Missatge 1
 - L'argument objecte indueix la visibilitat $K \xrightarrow{par} Profe$
 - L'argument identificador no indueix cap visibilitat
- Missatge 1.1
 - L'argument `c` el rep en el seu context d'emissió (el missatge 1)
 - L'enllaç dirigit $:K \rightarrow p$ es correspon a l'enllaç de paràmetre que és realització de la visibilitat $K \rightarrow Profe$
- Missatge 1.1.1
 - L'argument `c` el rep en el seu context d'emissió (el missatge 1.1)
 - Com a resultat del `find()` hi ha un enllaç dirigit $p \rightarrow control$
- Missatge 1.1.2
 - El canal de comunicació és un enllaç dirigit $p \rightarrow :Control$. Quin objecte, però és la destinació de l'enllaç dirigit?
 - L'únic $:Control$ sobre el que `p` en té visibilitat, és l'obtingut amb el `find()`. Per tant hem d'assumir que el canal de comunicació del missatge 1.1.2 és l'enllaç dirigit $p \rightarrow control$, realització de la visibilitat local $Profe \rightarrow Control$
 - * El DC l'hem de llegir com si enlloc de $:Control$ tinguéssim `control:Control`
- Missatge 1.2.1.1
 - Un missatge de creació sempre és possible
 - El missatge no té arguments, i per tant:
 - * No ens hem de preocupar d'obtenir-los
 - * No indueixen cap visibilitat
 - Després d'una creació el creador té una visibilitat sobre l'objecte creat
 - * Aquesta és la visibilitat $Control \xrightarrow{?} Avaluació^*$

Qüestions pendents

- Naturalesa de $Profe \xrightarrow{?} Control^*$
- Naturalesa de $Control \xrightarrow{?} Avaluació^*$
- La visibilitat $Profe \rightarrow Control$ hem dit que és local perquè no vèiem la necessitat de cap emmagatzematge. Aquesta suposició es referma a la vista dels altres DC?

2.2.2 Detall del procés: novaPregunta ()

Components i primers objectes observant novaPregunta ()

- Nous components explícits al DC:
 - MagatzemPreguntes
- $K, Profe, Control$ i $Avaluació$ són components que ja teníem
- El multjecte $List<Avaluació>$ diu que hem de tenir una visibilitat multiavaluada sobre un component $Avaluació$, que ja tenim
- av és l'argument d'un $add()$. Per tant és un **objecte**. El multjecte on es fa l' $add()$ és d' $Avaluacions$; per tant tenim l'objecte $av:Avaluació$
- De moment no podem dir res dels altres arguments

Primera anàlisi del DC novaPregunta ()

Símbols del DC de novaPregunta ()

est	???
pr	???
v	???
av	objecte , realització d' $Avaluació$

Visibilitats

- $K \xrightarrow{atr} \text{MagatzemPreguntes}$
 - Pel missatge 2.1. Hi ha una col·laboració amb un objecte que no rebem per paràmetre, i tampoc creem. Per tant ha de ser conegut prèviament
 - Una pista d'aquest coneixement previ és que en la col·laboració no es dona nom a l'objecte. No col·laborem amb *un* magatzem, ans amb *el* magatzem. Per tant, és conegut, i només n'hi ha un
- $K \xrightarrow{atr} \text{Profe}$
 - És el mateix argument que hem donat per $K \xrightarrow{atr} \text{MagatzemPreguntes}$ però aplicat al missatge 2.2.
- $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
 - El missatge 2.2.1 retorna un objecte $av:Avaluació$. Però aquest objecte es comunica a un objecte $c:Control$. De moment no veiem necessitats d'emmagatzematge, i per tant assumim que l'enllac dirigit $:\text{Profe} \rightarrow av$ és local
- $\text{Profe} \xrightarrow{atr} \text{Control}$
 - En el missatge 2.2.2 un objecte realització de Profe col·labora amb un objecte realització de Control . En el DC l'objecte $:\text{Profe}$ emissor del missatge 2.2.2 no té cap coneixement explícit de cap objecte realització de Control . Per tant hem d'assumir que en hi ha una visibilitat d'atribut $\text{Profe} \rightarrow \text{Control}$
- $\text{Control} \xrightarrow{par} \text{Avaluació}$
 - Pel missatge 1.1.2
- $\text{Control} \xrightarrow{?} \text{Avaluació}^*$
 - El multiobjecte indica una visibilitat multiavalada
 - L'add també indica una visibilitat multiavalada
 - De moment no tenim prou elements per determinar el tipus de la visibilitat

Qüestions pendents

- Visibilitats per analitzar
 - Control[?]→Avaluació*
- Visibilitats amb suposicions que cal refermar
 - Profe^{loc}→Avaluació
- Comprovació
 - Cal assegurar que tot missatge es pot enviar (té el canal i els arguments)
 - Que no ens haguem deixat cap visibilitat

Dos DC però un únic MComp

Semàntica del problema

- Quina és la responsabilitat del CU? Si ens fixem en els noms dels ES i en el DS, sembla que estem avaluant un control, i que per fer-ho cal indicar l'avaluació de cadascuna de les preguntes d'aquest control
- Si analitzem els DC veiem que el DC de `nouControl()` cerca un control i li envia un missatge de `iniAvaluació()`, que el que fa és esborrar la nota del control
- El DC de `novaPregunta()` en essència crea una avaluació i l'afegeix al control
- De tot plegat sembla clar que el control on en el segon ES s'afegeixen avaluacions, voldríem que fos el control que en el primer ES hem indicat que volem iniciar la seva avaluació
 - En termes de contractes, `nouContracte()` deixarà un `:Control` diferenciat (el control), i `novaPregunta()` exigirà l'existència d'un `:Control` diferenciat
- Però llavors el multiobjecte `List<Avaluació>` del DC de `nouControl()`, i el multiobjecte `List<Avaluació>` del DC de `novaPregunta()`, han de ser el mateix. I per tant les visibilitats `Control → Avalució*` obtingudes també han de ser la mateixa.
 - Hem resolt una anòmia

Comunicació entre ES

Control \rightarrow Avaluació*

- En el primer ES tenim Control \rightarrow Avaluació* com a resultat d'una creació
 - Hem creat un multiobjecte buit, on voldrem emmagatzemar les avaluacions que anem rebent
- En el segon ES tenim Control \rightarrow Avaluació* on hi fem add()
- Ambdues visibilitats per tant són la mateixa. I com que accedeix l'àmbit del context d'emissió (estan en ES diferents) s'ha de tractar d'una visibilitat d'**atribut**
 - Control \xrightarrow{atr} Avaluació*

K \rightarrow Profe

- En el primer DC tenim K \xrightarrow{par} Profe
- En el segon DC tenim K \xrightarrow{atr} Profe
- Ara cal resoldre dues preguntes:
 - Tenen alguna cosa a veure entre elles ambdues visibilitats?
 - Qui és el responsable de mantenir els enllaços dirigits d'atributs exigits al DC de novaPregunta() (això és, K \xrightarrow{atr} Profe) per tal que la comunicació sigui possible?
- El primer ES indica l'inici de l'avaluació d'un control per part d'un professor. El segon ES necessita col·laborar amb un professor. Amb quin? La semàntica del problema exigeix que ambdós professors, l'indicat al primer ES i el que necessitem en el segon ES, siguin el mateix. Per tant, l'ES nouControl() és responsable de construir K \xrightarrow{atr} Profe a partir de K \xrightarrow{par} Profe

Profe \rightarrow Control

- En el primer ES tenim Profe \xrightarrow{loc} Control. De fet hem dit que era local perquè no teníem cap necessitat d'emmagatzematge. Es tracta però d'una suposició, que tal i com ja hem indicat més amunt, caldria refermar
- En el segon ES per tal que la comunicació sigui possible cal Profe \xrightarrow{atr} Control
- Ara cal resoldre dues preguntes:
 - Tenen alguna cosa a veure entre elles ambdues visibilitats?
 - Qui és el responsable de mantenir els enllaços dirigits d'atributs exigits al DC de novaPregunta() (això és, Profe \xrightarrow{atr} Control) per tal que la comunicació sigui possible?
- Com hem vist en l'anàlisi de la semàntica dels DC, en el primer ES indiquem que volem avaluar un control, i en el segon ES hi introduïm les avaluacions de cada pregunta. Per tant ambdós controls han de ser el mateix. I això significa que en l'ES nouControl() sí que hi ha una necessitat que p:Profe emmagatzemi el :Control que ha recuperat amb el find()
- L'anàlisi conjunta d'ambdós DC ens porta a concloure que la visibilitat Profe \xrightarrow{loc} Control és de fet una visibilitat Profe \xrightarrow{atr} Control

Conclusions fins el moment

Símbols del DC de nouControl()

c	identificador d'un objecte :Control
p	objecte, realització de Profe
control	objecte, realització de Control

Símbols del DC de novaPregunta ()

est	???
pr	???
v	???
av	objecte , realització d'Avaluació

Visibilitats

- $K \xrightarrow{par} \text{Profe}$
- $K \xrightarrow{atr} \text{Profe}$
- $K \xrightarrow{atr} \text{MagatzemPreguntes}$
- $\text{Profe} \xrightarrow{?} \text{Control}^*$
- $\text{Profe} \xrightarrow{atr} \text{Control}$
- $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
- $\text{Control} \xrightarrow{par} \text{Avaluació}$
- $\text{Control} \xrightarrow{atr} \text{Avaluació}^*$

Qüestions pendents acumulades

- Tipus de visibilitat
 - $\text{Profe} \xrightarrow{?} \text{Control}^*$
- Suposicions per refermar
 - $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
- Comprovació
 - Cal assegurar que tot missatge es pot enviar (té el canal i els arguments)
 - Que no ens haguem deixat cap visibilitat

MOD

MOD

Anàlisi de l'emmagatzematge multiavaluat

- El missatge 1.1.1 és un `find()` sobre un multiobjecte. L'objectiu és obtenir un objecte a partir del seu identificador i d'un espai (el multiobjecte) on hi ha molts objectes que són realització del mateix component
- Si $\text{Profe} \rightarrow \text{Control}^*$ és **local** llavors l'enllaç dirigit $p:\text{Profe} \rightarrow \text{Control}^*$ és un de diferent cada cop que demanem analitzar un nou control. No té massa sentit. Tot fa pensar que els controls ja els tenim, i que el que fa `nouControl()` és cercar un control determinat dels preexistents. Per tant cal concloure que la visibilitat que tenim és $\text{Profe} \xrightarrow{\text{atr}} \text{Control}^*$
- En el missatge 2.2.2.1 passa una cosa similar. Estem afegint una nova avaluació; si $\text{Control} \rightarrow \text{Avaluació}^*$ és una visibilitat local llavors en cada petició de `novaPregunta()` l'enllaç dirigit és nou, i només hi fem un `add()`! Clarament el que es pretén és que en la iteració es vagi augmentat el contingut del multiobjecte. Per tant aquest ha de tenir un àmbit superior al del context d'emissió. Així el que tenim és $\text{Control} \xrightarrow{\text{atr}} \text{Avaluació}^*$
 - De fet ja havíem arribat a aquesta conclusió després de veure que aquesta visibilitat i la del primer DC eren la mateixa
- De manera més resumida podem argumentar que tant $\text{Profe} \rightarrow \text{Control}^*$ com $\text{Control} \rightarrow \text{Avaluació}^*$ són visibilitats d'emmagatzematge, i per tant han de ser d'atribut

Anàlisi dels magatzems

- Podem suposar que, pel seu nom i pel nom dels missatges que rep, `MagatzemPreguntes` té una funció d'emmagatzematge
- Sota aquest supòsit `obtenirPregunta()` (missatge 2.1) es comporta com un `find()`: li passem un identificador i retorna un objecte `:Pregunta`
- Llavors apareixen clarament noves visibilitats:
 - $\text{MagatzemPreguntes} \xrightarrow{\text{par}} \text{Pregunta}$
 - * El missatge 2.1 fa que el `MagatzemPreguntes` retorni una `Pregunta`
 - $K \xrightarrow{\text{loc}} \text{Pregunta}$

- * En enviar el missatge 2.1 l'objecte `:K` té una visibilitat sobre l'objecte `:Pregunta` retornat
 - * L'objecte retornat es comunica a través del missatge 2.2. Per tant, si assumim que no hi ha cap necessitat d'emmagatzemar la pregunta, el que tenim és una visibilitat local
 - Profe \xrightarrow{par} Pregunta
 - * Pel missatge 2.2
 - Avaluació \xrightarrow{par} Pregunta
 - * Pel missatge 2.2.1
- Internament podem suposar que, per gestionar l'emmagatzematge que li suposem, `MagatzemPreguntes` té una visibilitat d'emmagatzematge `MagatzemPreguntes \xrightarrow{atr} Pregunta*`

Noves conclusions fins el moment

Símbols del DC de `nouControl()`

c	identificador d'un objecte <code>:Control</code>
p	objecte, realització de <code>Profe</code>
control	objecte, realització de <code>Control</code>

Símbols del DC de `novaPregunta()`

est	???
pr	Identificador d'un objecte <code>Pregunta</code>
v	???
av	objecte, realització d' <code>Avaluació</code>

MOD

NOU

NOU

MOD

NOU

NOU

NOU

Visibilitats

- $K \xrightarrow{par} Profe$
- $K \xrightarrow{atr} Profe$
- $K \xrightarrow{atr} MagatzemPreguntes$
- $K \xrightarrow{loc} Pregunta$
- $Profe \xrightarrow{par} Pregunta$
- $Profe \xrightarrow{atr} Control^*$
- $Profe \xrightarrow{atr} Control$
- $Profe \xrightarrow{loc} Avaluació$
- $Control \xrightarrow{par} Avaluació$
- $Control \xrightarrow{atr} Avaluació^*$
- $Avaluació \xrightarrow{par} Pregunta$
- $MagatzemPreguntes \xrightarrow{par} Pregunta$
- $MagatzemPreguntes \xrightarrow{atr} pregunta^*$

Qüestions pendents acumulades

- Suposicions per refermar
 - $K \xrightarrow{loc} Pregunta$
 - $Profe \xrightarrow{loc} Avaluació$
 - $MagatzemPreguntes$ es comporta com un emmagatzematge multiavaluat de preguntes
- Comprovació
 - Cal assegurar que tot missatge es pot enviar (té el canal i els arguments)
 - Que no ens haguem deixat cap visibilitat

Comprovació de les conclusions actuals

- Missatges 1.x
 - Aquesta anàlisi ja ha estat feta durant l'anàlisi del DC de `nouControl()`
 - Llavors no hi havia cap llacuna que ara calgui revisar
- Missatge 2
 - L'únic argument del qual sabem la seva naturalesa (`pr`) és un identificador i per tant no indueix cap visibilitat
 - Més endavant caldrà revisar què passa amb els altres dos arguments
- Missatge 2.1
 - L'argument l'obtenim del context d'emissió (el missatge 2)
 - L'emissió del missatge és possible gràcies a $K \xrightarrow{atr} \text{MagatzemPreguntes}$
 - El valor de retorn del missatge indueix una visibilitat $K \xrightarrow{loc} \text{Pregunta}$
 - * La visibilitat no és de paràmetre, i no tenim motius per pensar en un emmagatzematge
- Missatge 2.2
 - Dos arguments els obtenim del context d'emissió (el missatge 2). L'altre argument, `pregunta`, l'obtenim com a resultat del missatge 2.1, i es correspon a una realització de la visibilitat $K \xrightarrow{loc} \text{Pregunta}$
 - L'emissió del missatge és possible gràcies a $K \xrightarrow{atr} \text{Profe}$
 - Queda pendent per analitzar la influència dels arguments `est i v`
- Missatge 2.2.1
 - Els arguments els obtenim del context d'emissió (el missatge 2.1)
 - Emetre un missatge de creació sempre és possible
 - La creació indueix un enllaç dirigit que és realització de la visibilitat $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
 - * Diem que és local perquè no tenim cap argument en contra

- Missatge 2.2.2
 - L'argument és l'objecte obtingut en la creació. En concret, a través de la visibilitat $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
 - L'emissió del missatge és possible gràcies a $\text{Profe} \xrightarrow{atr} \text{Control}$
- Missatge 2.2.2.1
 - L'argument l'obtenim del context d'emissió (el missatge 2.2.2)
 - L'emissió del missatge és possible gràcies a $\text{Control} \xrightarrow{atr} \text{Avaluació}^*$

2.2.3 Detall del procés: `fiControl()`

Components i primers objectes observant `fiControl()`

- Nous components explícits al DC:
 - Estudiant
- El multobjecte `List<Avaluació>` diu que hem de tenir una visibilitat multiavaluada sobre un component `Avaluació`
- `av` és l'argument d'un `add()`. Per tant és un **objecte**. Els multobjectes on es fa l'`add()` són d'Avaluacions; per tant tenim l'objecte `av:Avaluació`

Anàlisi del DC `fiControl()`

Símbols del DC de `fiControl()`

est	objecte, realització de Estudiant
av	???

Visibilitats

- $K \xrightarrow{atr} \text{Profe}$
 - Pel missatge 3.1
 - No és cap visibilitat nova
- $\text{Profe} \xrightarrow{atr} \text{Control}$
 - Pel missatge 3.1.1
 - No és cap visibilitat nova
- $\text{Control} \xrightarrow{atr} \text{Avaluació}^*$
 - Pel missatge 3.1.1.1
 - No és cap visibilitat nova
- $\text{Avaluació} \xrightarrow{?} \text{Estudiant}$
 - Pel missatge 3.1.1.1.1
 - Els missatges que emet un multiobjecte en un DC de fet són missatges que emet cada objecte emmagatzemat en el multiobjecte
 - Per tant és un objecte :Avaluació qui emet el missatge `add(av)`
 - * Cal analitzar com és que enviem un `add()` a un objecte que no és un multiobjecte
 - * Cal analitzar què es l'argument `av`
- $\text{Estudiant} \xrightarrow{atr} \text{Avaluació}^*$
 - Pel missatge 3.1.1.1.1.1
 - Aquest darrer missatge és un emmagatzematge sobre un multiobjecte. Per això demanem que la visibilitat sigui d'atribut

Missatges i multiobjectes

Què és *av*

- `List<Avaluació>` emet el missatge `add(av)`. Això vol dir que qui realment emet el missatge és cada objecte `:Avaluació` de dins el multiobjecte
- L'argument del missatge és *av*, que és un símbol que és el primer cop que apareix. Què pot representar?
 - El nom ens pot fer pensar que és un objecte `:Avaluació`. És a dir, interpretem que cada `:Avaluació` del multiobjecte s'envia a sí mateixa com a argument del missatge `add()`
 - Si mirem el missatge que emet `:Estudiant` veiem que és un `add(av)` sobre un multiobjecte d'Avaluacions. Per tant tenim que `av:Avaluació`
 - Això referma la idea que l'argument del missatge 3.1.1.1.1 és el propi emissor del missatge
- En conseqüència tenim les següents visibilitats:
 - $\text{Estudiant} \xrightarrow{\text{par}} \text{Avaluació}$
 - * L'estudiant rep una avaluació
 - $\text{Estudiant} \xrightarrow{\text{atr}} \text{Avaluació}^*$
 - * Hi ha emmagatzematge multiavalaut
 - * De fet això ja ho havíem vist

Què és el missatge add

- `add()` es un **missatge de grup**. Això és, està perfectament definit
- Això només és veritat si qui rep el missatge és un multiobjecte!! Altrament és un **missatge d'enllaç**, definit pel desenvolupador
- En aquest cas, qui ha dissenyat aquest DC ha posat el nom `add()` a un missatge d'`Estudiant` per remarcar que la seva semàntica és la de **delegar** l'`add()` autèntic sobre un multiobjecte
- Per tant, l'`add()` que cada `:Avaluació` envia a `Estudiant` és un **missatge d'enllaç**.
- Com hem vist, `av` és un objecte `:Avaluació`. I en conseqüència el missatge `add()` indueix la visibilitat $\text{Estudiant} \xrightarrow{\text{par}} \text{Avaluació}$

Avaluació $\xrightarrow{?}$ Estudiant

Necessitat d'una visibilitat d'atribut

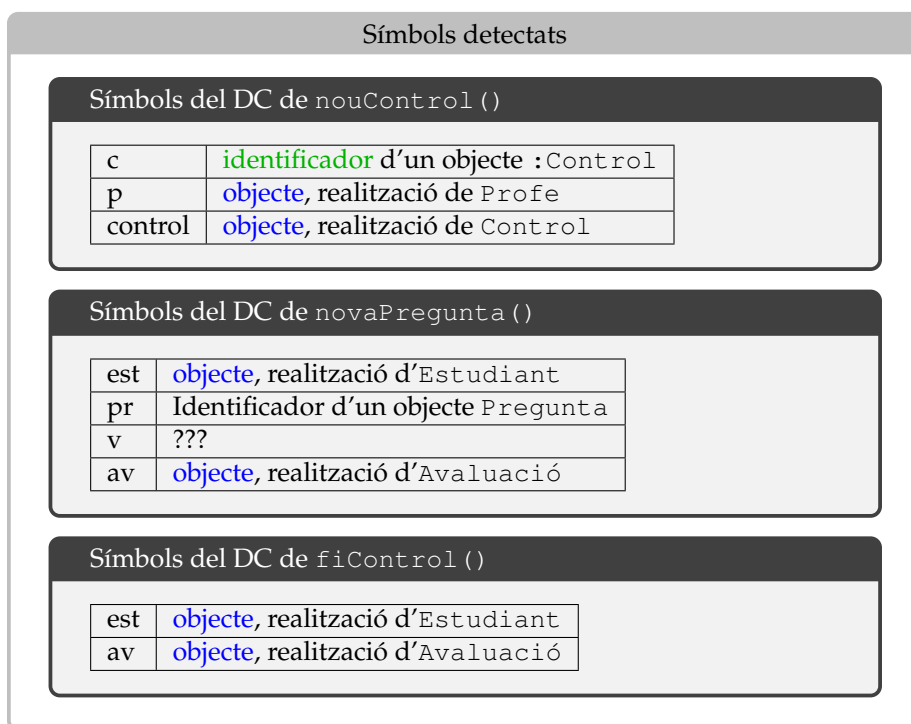
- El missatge `add()` és d'una `:Avaluació` a un `:Estudiant`
- L'`:Avaluació` emissora del missatge és alhora l'argument `av` que passem amb el missatge
- Però quin és l'`:Estudiant` receptor del missatge?
- Per tal de poder enviar el missatge cal una visibilitat $\text{Avaluació} \xrightarrow{?} \text{Estudiant}$. Com que l'estudiant no el rebem en el missatge `fiControl()` ni el creem, necessàriament la visibilitat ha de ser d'atribut: $\text{Avaluació} \xrightarrow{\text{atr}} \text{Estudiant}$

Possibilitat de la visibilitat d'atribut

- Necessitem $\text{Avaluació}^{\text{atr}} \rightarrow \text{Estudiant}$. La tenim?
- Tota :Avaluació del multiobjecte d'Avaluacions considerat, s'ha creat en el missatge 1.2.1
- La visibilitat que necessitem, per ser d'atribut, s'ha de crear en crear l' :Avaluació
- Però quin és l'estudiant que ha de mantenir, amb visibilitat d'atribut, cada :Avaluació?
- Un dels arguments del missatge de creació d'Avaluació és *est*, que de moment no sabem què és. Si admetem que aquest argument és un :Estudiant llavors el que té lògica és considerar que sigui aquest estudiant el que cal mantenir amb un visibilitat d'atribut
- En d'altres paraules, tenim un argument, *est*; i sabem que els arguments han de ser útils. No sabem si l'argument és un identificador, un valor o un objecte, però sabem que el mètode de creació ha de crear un enllaç dirigit d'atribut sobre un :Estudiant. Tot sembla indicar que aquest :Estudiant és l'*est* que ens passen!!
- Per tant:
 - *est*:Estudiant és un objecte
 - $\mathcal{K}^{\text{par}} \rightarrow \text{Estudiant}$
 - $\text{Profe}^{\text{par}} \rightarrow \text{Estudiant}$
 - $\text{Avaluació}^{\text{par}} \rightarrow \text{Estudiant}$
 - $\text{Avaluació}^{\text{atr}} \rightarrow \text{Estudiant}$

Una visibilitat oblidada

- La visibilitat $\text{Avaluació}^{\text{atr}} \rightarrow \text{Estudiant}$ és una **visibilitat de recuperació**: es tracta d'un argument rebut, que com que no s'usa ni es comunica, entenem que s'emmagatzema
- El mateix passa amb l'argument *pregunta*. Per tant tenim $\text{Avaluació}^{\text{atr}} \rightarrow \text{Pregunta}$



Conclusions quasi finals

Visibilitats

- $K \xrightarrow{par} Profe$
- $K \xrightarrow{par} Estudiant$
- $K \xrightarrow{atr} Profe$
- $K \xrightarrow{atr} MagatzemPreguntes$
- $K \xrightarrow{loc} Pregunta$
- $Profe \xrightarrow{par} Pregunta$
- $Profe \xrightarrow{par} Estudiant$
- $Profe \xrightarrow{atr} Control^*$
- $Profe \xrightarrow{atr} Control$
- $Profe \xrightarrow{loc} Avaluació$
- $Control \xrightarrow{par} Avaluació$
- $Control \xrightarrow{atr} Avaluació^*$
- $Avaluació \xrightarrow{par} Pregunta$
- $Avaluació \xrightarrow{atr} Pregunta$
- $Avaluació \xrightarrow{par} Estudiant$
- $Avaluació \xrightarrow{atr} Estudiant$
- $MagatzemPreguntes \xrightarrow{par} Pregunta$
- $MagatzemPreguntes \xrightarrow{atr} pregunta^*$
- $Estudiant \xrightarrow{par} Avaluació$
- $Estudiant \xrightarrow{atr} Avaluació^*$

Qüestions pendents

- Suposicions per refermar
 - $K \xrightarrow{loc} \text{Pregunta}$
 - $\text{Profe} \xrightarrow{loc} \text{Avaluació}$
 - `MagatzemPreguntes` es comporta com un emmagatzematge multiavaluat de preguntes
- Comprovació

Comprovació de les conclusions actuals

- Tots els missatges estaven comprovats. Calia només veure la influència de `est` i
 - Cal assegurar que tot missatge es pot enviar (té el canal i els arguments)
 - Que no ens haguem deixat cap visibilitat

`pr` i comprovar el missatge `add()` que rep `l' : Estudiant`
- Hem vist que `est` és un objecte, i que per tant indueix una visibilitat de paràmetre sobre `Estudiant` al llarg de tota la cadena on es passa l'argument
 - Les crides són correctes perquè els arguments es reben del context d'emissió; i els canals de comunicació ja els teníem comprovats
 - El missatge de creació 2.2.1 ha d'assegurar que els arguments són útils. Com que no els usem ni els comuniquem, és que els estem emmagatzemant
 - * D'aquí la visibilitat $\text{Avaluació} \xrightarrow{atr} \text{Estudiant}$
 - * Queda per veure com afecta l'argument `v`
 - Pel que fa a l'`add()` que rep `l' : Estudiant`
 - * La visibilitat $\text{Avaluació} \xrightarrow{atr} \text{Estudiant}$ assegura el canal de comunicació
 - * L'argument `av` és el propi emissor, i per tant no genera cap problema

Què és v

- Estem avaluant. En el segon DC creem una avaluació, per una pregunta i un estudiant, i li passem un v
- El més lògic és pensar que aquesta v és el **valor** de l'avaluació
- Així v no indueix cap visibilitat
 - Quan el passem com a argument no indueix una visibilitat de paràmetre
 - Quan l'enviem però no l'usem ni el comuniquem, cal emmagatzemar-lo. Però en tractar-se d'un valor no indueix cap visibilitat d'atribut

Conclusions finals: Símbols detectats

Símbols del DC de nouControl ()

c	identificador d'un objecte :Control
p	objecte, realització de Profe
control	objecte, realització de Control

Símbols del DC de novaPregunta ()

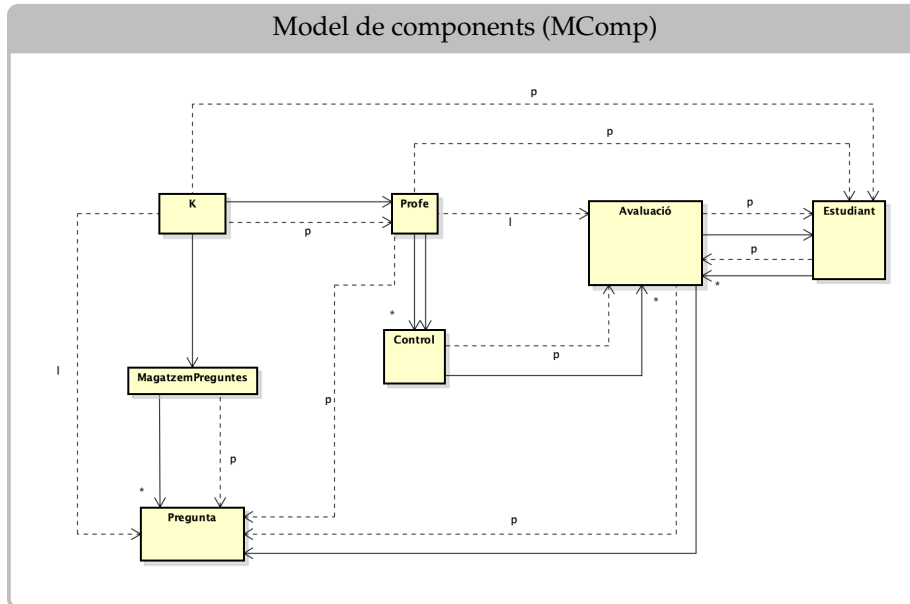
est	objecte, realització d'Estudiant
pr	Identificador d'un objecte Pregunta
v	valor
av	objecte, realització d'Avaluació

Símbols del DC de fiControl ()

est	objecte, realització d'Estudiant
av	objecte, realització d'Avaluació

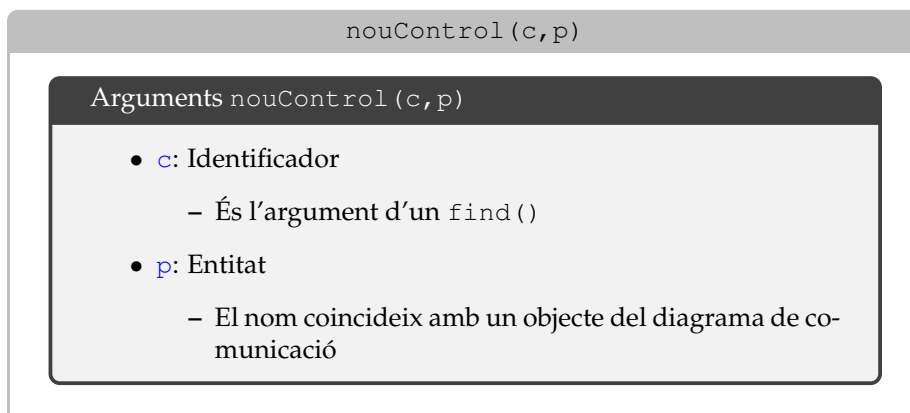
MOD

2.2.4 MComp



2.3. Contractes des del disseny

2.3.1 Contractes segons la semàntica dels DC



PRE nouControl(*c*, *p*)

1. El professor ha fet el control al que correspon l'identificador donat

- $\exists control : Control$ tal que:
 - 1.1. $realitza(p, control)$
 - 1.2. $control.id = c$

-
- 1.
- Cal assegurar que l'identificador permet recuperar una entitat (1.2)
 - Però aquesta recuperació només té sentit en el context d'un professor concret (1.1)
 - Interpretem que la semàntica de la visibilitat $Profe \rightarrow Control^*$ diu quins controls ha fet cada professor. (Fet tant pot fer referència a què el professor ha construït l'examen, o bé que l'ha proposat als seus estudiants. El glossari hauria de desfer l'equívoc, però des del disseny no tenim més informació.). Aquesta semàntica la modelitzem amb l'associació $realitza()$
2. Dels DC no en podem extreure res de si un control s'avalua més d'un cop, ni de si els controls avaluats ho han d'estar per a totes les seves preguntes
- El que sí veiem és que cada cop que s'avalua un control el multiobjecte $Control \rightarrow Avaluació^*$ es crea de nou. Per tant només mantenim la darrera avaluació del control

POST_{nouControl(c,p)}

1. c està diferenciat
 2. La nota del control és 0
 - $control.nota = 0$
 3. Dins del $c:Control$, $p:Profe$ està diferenciat
 4. El control c no té cap avaluació
 - $\exists av : Avaluació$ tal que $avalua(av, c)$
-
- Deixem com a diferenciables les entitats que necessitem en el proper ES
 - p el deixem diferenciat des de c , perquè és des d'on hi accedim
 - La visibilitat $Control \xrightarrow{atr} Avaluació^*$ considerem que és el model d'una associació $avalua()$

```
novaPregunta(est, pr, v)
```

Arguments novaPregunta(est, pr, v)

- **est**: Entitat
 - **pr**: Identificador
 - **v**: Valor literal
-
- *est* es delega fins al missatge de creació de *l':Avaluació*, que ni *l'*usa ni la comunica. Per tant hem de concloure que *l'*emmagatzema.
 - En el DC de `fiControl()` *l':Avaluació* ha d'enviar un missatge a un *:Estudiant*, la visibilitat del qual només pot ser conseqüència d *Avaluació*^{atr}*→Estudiant*
 - Aquesta visibilitat d'atribut només es pot haver inicialitzat en la creació de *l':Avaluació*. D'aquí concloem que *est* és l'objecte *:Estudiant* que és la destinació d'aquest enllaç dirigit d'atribut.
 - Pel que fa a *v*, no tenim cap argument per pensar que és una entitat o un identificador. Per tan considerem que és un valor literal.
 - Per una explicació més detallada de com obtenim la naturalesa d'*est* o de *v*, hem d'anar a [Model de components](#)^{→19}

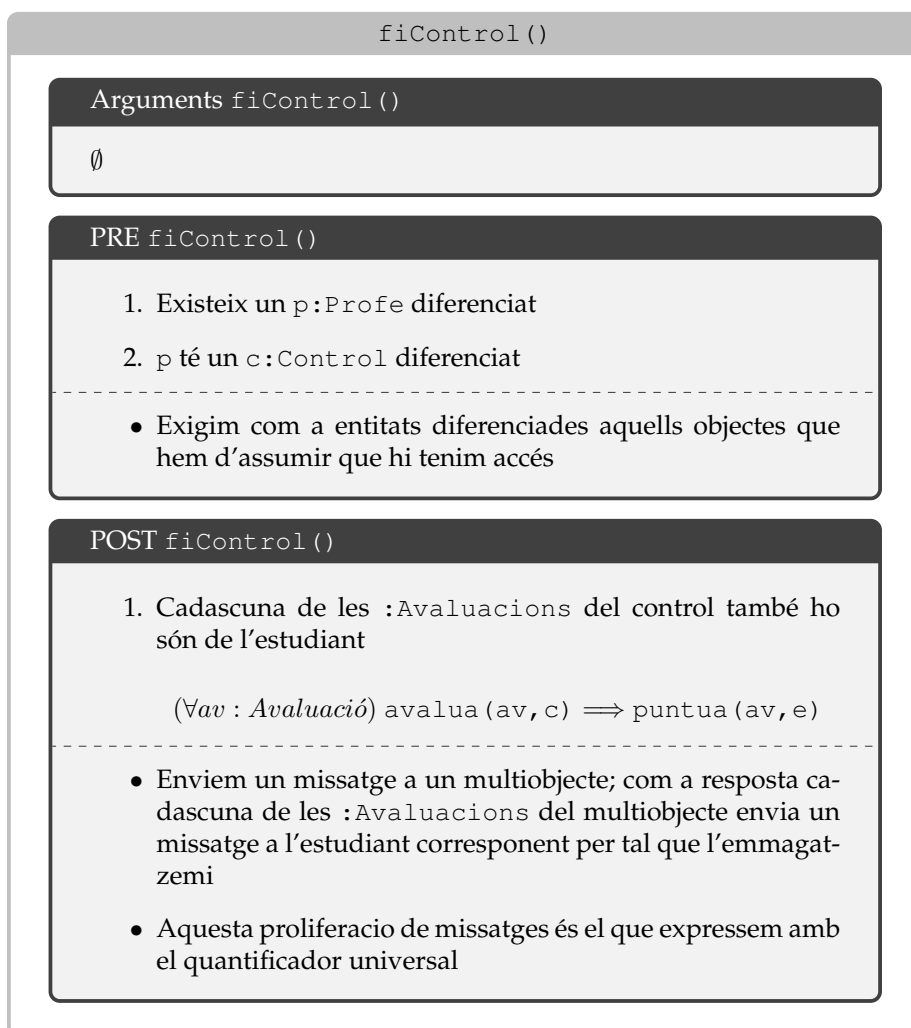
PRE. novaPregunta(est, pr, v)

1. Existeix un `p:Profe` diferenciat
 2. `p` té un `c:Control` diferenciat
 3. `pr` és un identificador correcte per a `Pregunta`
 - $\exists \text{pregunta: Pregunta tq } \text{pregunta.id} = \text{pr}$
-
1. Exigim com a entitats diferenciades aquells objectes que hem d'assumir que hi tenim accés
 - En concret, les dues primeres PRE asseguren que els missatges 2.2 i 2.2.2 tenen el canal per poder-se transmetre
 - Els altres canals estan assegurats per les visibilitats d'atribut
 2. L'exigència sobre l'identificador assegura que `obtenirPregunta()` sempre retorna una `:Pregunta` vàlida
 3. Desconeixem quin és l'identificador d'`Avaluació`. Però pel DC sabem que `l'add()` no dona problemes. Per tant `av:Avaluació` és nova
 - Si el mètode de creació genera un identificador nou cada cop, la correctesa de `l'add()` està assegurada i no cal afegir cap PRE
 - Si donat un `:Control` i una `:Pregunta` l'avaluació és única, llavors per assegurar que no tenim problemes amb `l'add()` cal una PRE que exigeixi que pel control donat no hi ha cap avaluació per a aquesta pregunta:
 - $\nexists \text{av: Avaluació tal que } \text{qui}(\text{av}, \text{p}) \text{ i } \text{realitza}(\text{p}, \text{c})$
 - Si bé aquesta condició es pot extreure del nostre coneixement del problema, no s'extreu del DC. Per tant no la considerem.

POST.novaPregunta(est, pr, v)

1. Hi ha una nova av:Avaluació per a la pregunta, l'estudiant i la nota indicada
 - 1.1. \exists nova av:Avaluació
 - 1.2. `puntua(av, est)`
 - 1.3. `què(av, pregunta)`
 - 1.4. `av.valor = v`
2. Aquesta nova av:Avaluació és pel :Control que estem avaluant
 - `puntua(av, c)`
3. El control que estem avaluant ha computat correctament el valor de la nova avaluació
 - `c.nota = c.nota + v`

-
1. Assumim una associació i què com a model de la visibilitat $Avaluació \xrightarrow{atr} Pregunta$, i una associació `puntua` com a model de la visibilitat de recuperació $Avaluació \xrightarrow{atr} estudiant$
 - Tot argument passat en una creació que no s'usi ni es comuniqui cal emmagatzemar-lo
 - `est` és un objecte i per tant l'emmagatzematge és una visibilitat d'atribut que volem expressar en l'especificació
 2. Per lligar l':Avaluació i el :Control usem l'associació `avalua()`, que ja teníem
 3. La nota del control s'incrementa, en el DC, amb el valor de l'avaluació. Però aquest valor és `v`



2.3.2 Correctesa i robustesa

- Tot seguit repetim els contractes, però sense els comentaris
- Afegim en color **blau** les assercions necessàries per a la **correctesa de la seqüenciació**
- Afegim en color **verd** les assercions necessàries per a la **robustesa de la seqüenciació**

Associacions que s'extreuen dels DC

- realitza: Profe--Control
- avalua: Avaluació--Control
- què: Avaluació--Pregunta
- puntua: Avaluació--Estudiant

nouControl(c,p)

Arguments nouControl(c,p)

- *c*: Identificador
 - És l'argument d'un find()
- *p*: Entitat
 - El nom coincideix amb un objecte del diagrama de comunicació

PRE nouControl(c,p)

1. El professor ha fet el control al que correspon l'identificador donat
 - $\exists control : Control$ tal que:
 - 1.1. realitza(p, control)
 - 1.2. control.id = c
2. $\nexists c : Control$ diferenciat

POST nouControl(*c*, *p*)

1. *c* està diferenciat
2. La nota del control és 0
 - *control.nota* = 0
3. Dins del *c*:Control, *p*:Profe està diferenciat
4. El control *c* no té cap avaluació
 - $\nexists av : \text{Avaluació tal que } \text{avalua}(av, c)$

novaPregunta(*est*, *pr*, *v*)Arguments novaPregunta(*est*, *pr*, *v*)

- **est**: Entitat
- **pr**: Identificador
- **v**: Valor literal

PRE. novaPregunta(*est*, *pr*, *v*)

1. Existeix un *p*:Profe diferenciat
2. *p* té un *c*:Control diferenciat
3. *pr* és un identificador correcte per a Pregunta
 - $\exists \text{pregunta} : \text{Pregunta tq } \text{pregunta.id} = \text{pr}$

POST.novaPregunta(est, pr, v)

1. Hi ha una nova av:Avaluació per a la pregunta, l'estudiant i la nota indicada
 - 1.1. \exists nova av:Avaluació
 - 1.2. `puntua(av, e)`
 - 1.3. `què(av, pregunta)`
 - 1.4. `av.valor = v`
2. Aquesta nova av:Avaluació és pel :Control que estem avaluant
 - `avalua(av, c)`
3. El control que estem avaluant ha computat correctament el valor de la nova avaluació
 - `c.nota = c.nota + v`
4. `p:Profe` diferenciat
5. Dins de `c:Control`, `p` està diferenciat

fiControl()

Arguments fiControl()

\emptyset

PRE fiControl()

1. Existeix un `p:Profe` diferenciat
2. `p` té un `c:Control` diferenciat
3. `c` té avaluacions
 - $\exists av : \text{Avaluació tal que } \text{avalua}(av, c)$

POST `fiControl()`

1. Cadascuna de les `:Avaluacions` del control també ho són de l'estudiant

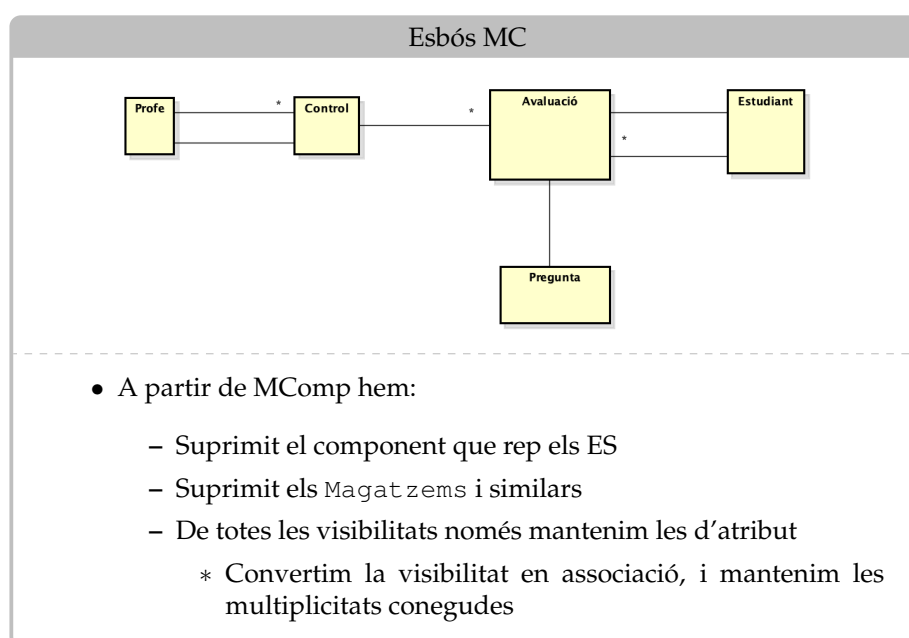
$(\forall av : Avaluació) \text{avalua}(av, c) \implies \text{puntua}(av, e)$

2. $\exists c : \text{Control diferenciat}$
3. $\exists c : \text{Control tal que } c \text{ té un } :Profe \text{ diferenciat}$

- La darrera POST no és estrictament necessària però evita efectes laterals indesitjats
 - L'activació del control es fa durant l'execució del CU; el que volem és que en acabar el sistema només s'hagi modificat en allò rellevant per a ell
 - L'activació o diferenciació del `:Profe` dins del `:Control` només té sentit durant l'execució del CU; després és irrellevant

2.4. MC per enginyeria inversa

2.4.1 Procés d'enginyeria inversa per a obtenir MC



Procediment d'anàlisi de les associacions

- Analitzem cada visibilitat d'atribut com apareix en els DC
 - Intentem extreure'n la multiplicitat en els dos extrems
 - Intentem extreure'n una semàntica, que ens permeti etiquetar l'associació de la qual la visibilitat n'és el model

Profe → Control*

Semàntica i primeres conclusions

- Cada professor *manté* un conjunt de controls; els *seus* (el glossari haurà de donar sentit a aquest possessiu)
- En l'enginyeria inversa dels contractes^{→49} hem anomenat *realitza* a l'associació corresponent
- Per tant:

realitza : Profe – Control, ? – N

Multiplicitats que manquen

- No tenim cap informació adicional que ens permeti extreure la multiplicitat a l'extrem de Profe
- Com que el `find()` el fem des del `:Profe` res impedeix que dos professors diferents tinguin controls amb el mateix identificador
- Per això considerarem que l'associació és multiavaluada a l'extrem de `:Profe`
- La suposició no s'extreu pròpiament dels diagramens, ni tampoc del coneixement del problema: cal algun supòsit adicional. Per tant la marcarem en **verd**
- Per tant:

realitza : Profe – Control, **M** – N

Obligatorietats i optativitats

- Per coneixement del problema, un professor pot haver fer controls, però no necessàriament
 - Com que el supòsit de l'optativitat parteix del nostre coneixement del problema, i no s'extreu per tant directament dels diagrames, ho indicarem amb **blau**
- Què significa *realitza*?
 - El professor construeix el control. Llavors no té massa sentit pensar en controls anònims, això és, que no sabem qui l'ha realitzat
 - El professor posa el control als seus alumnes. Llavors té tot el sentit pensar en controls construïts, però que encara cap professor ha realitzat
 - Prenem una de les possibilitats a l'atzar
 - Com que la conclusió depèn del glossari, o de supòsits addicionals, la marcarem amb **verd**
- Per tant:

realitza : *Profe* – *Control*, * – *N*, **optatiu**–**optatiu**

Control→Avaluació*

Semàntica i primeres conclusions

- Cada control manté un conjunt d'avaluacions
- A diferència de $\text{Profe} \rightarrow \text{Control}^*$, ara sabem qui crea el multiobjecte, i qui l'alimenta
 - L'ES `nouControl()` crea el multiobjecte
 - L'ES `novaPregunta()` crea l'avaluació i l'emmagatzema al multiobjecte
- En conseqüència de l'anàlisi dels DC tenim que cada `:Avaluació` la crea i emmagatzema un únic `:Control`
 - L'associació `Control – Avaluació` és **monoavaluada** a l'extrem de `Control`
 - L'associació `Control – Avaluació` és **obligada** a l'extrem de `Control`
- A l'associació de la qual enllacdirMControlAvaluació n'és el model prèviament l'hem anomenada `avalua`^{→49}
- Per tant:

avalua : Control – Avaluació, 1 – N

Obligatorietats i optativitats que manquen

- Primer tenim els controls i després els avaluem. Per tant l'extrem d'`Avaluació` és optatiu
- Per tant:

avalua : Control – Avaluació, 1 – N, obligat, optatiu

Estudiant→Avaluació*

Semàntica i primeres conclusions

- Cada cop que acabem l'avaluació d'un control (ES `fiControl()`) afegim les avaluacions del control a les avaluacions de l'estudiant
 - Cada estudiant manté el conjunt de les avaluacions dels controls que ha fet
- L'ES `novaPregunta()` és qui construeix l':Avaluació. I per fer-ho cal proporcionar, entre l'altres, l':Estudiant. Per tant:
 - L'associació *Estudiant – Avaluació* és **monoavaluada** a l'extrem de l'Estudiant
 - L'associació *Estudiant – Avaluació* és **obligada** a l'extrem de l'Estudiant
- A l'associació de la qual *Estudiant→Avaluació** n'és el model prèviament l'hem anomenada *puntua*.^{→49} Compte però perquè si bé les associacions són simètriques (*Estudiant – Avaluació* i *Avaluació – Estudiant* són la mateixa associació), les visibilitats no
 - La visibilitat que tenim és *Estudiant→Avaluació**
- Per tant:

$$puntua : Estudiant - Avaluació, 1 - N$$

Obligatorietats i optativitats que manquen

- Cada cop que s'avalua un control, es crea una avaluació per cada pregunta, que al final s'apunten a l'estudiant
- Així, $p_{i,j}$ ens indica que un estudiant ha estat avaluat per una pregunta, la que indica la interrelació que de l'avaluació de qui l'estudiant està lligat segons qui
- En paraules planeres, un estudiant té tantes avaluacions com preguntes se li han avaluat
- Si $p_{i,j}$ és optativa a l'extrem d'Avaluació significa hi ha estudiants que no han estat avaluats mai.
- Què és un i : Estudiant?
 - Algú a qui donem classe, i que en algun moment podem avaluar. Llavors pot ser que tinguem un estudiant nou a qui encara no hem passat cap control
 - Aquell qui en algun moment ha passat un control, i per tant hem avaluat. Llavors, per definició, tot estudiant té una avaluació com a mínima
- L'obligatorietat o no depèn del glossari. Prenem una decisió a l'atzar i la marquem en verd.
- Per tant:

$p_{i,j}$: Estudiant – Avaluació, 1 – N, obligat–optatiu

Avaluació→Pregunta

Semàntica i primeres conclusions

- Aquesta és una **visibilitat de recuperació**: l'ES `nouControl()` crea l':Avaluació, passant-li la :Pregunta, que l':Avaluació emmagatzema. Per tant:
 - L'associació *Avaluació – Pregunta* és **monoavaluada** a l'extrem de l'Pregunta
 - L'associació *Avaluació – Pregunta* és **obligada** a l'extrem de l'Pregunta
- A l'associació de la qual *Avaluació→Pregunta* n'és el model prèviament l'hem anomenada *què*.^{→49}
- Per tant:

què : *Avaluació – Pregunta*, ? – 1

Multiplicitats que manquen

- Si una pregunta pot estar en més d'un control, llavors quan avaluem ambdós controls cadascun d'ells generarà una avaluació amb la mateixa pregunta
- Com que la resposta depèn de la interpretació de l'enunciat, o de supòsits addicionals, el marquem en verd:
- Per tant:

què : *Avaluació – Pregunta*, *M* – 1

Obligatorietats i optativitats que manquen

- Primer tenim els controls, amb le seves preguntes; i després els avaluem. Per tant té tot el sentit considerar que hi ha controls, construïts, però que encara no s'han avaluat. Per tant les seves preguntes pot ser que no estiguin avaluades
- És més, dels DC no s'extreu que en valuar un control calgui avaluar tots els estudiants, i per cada estudiant totes les preguntes
- Així és perfectament factible que hi hagi preguntes no avaluades
 - Per exemple, en avaluar el control no hem introduït l'avaluació de la pregunta per cap estudiant
 - Un altre exemple: hi ha controls que no s'han avaluat
- Com que la conclusió s'extreu de la semàntica del problema, amb l'ajut dels DC, la marquem en blau
- Per tant:

*què : Avaluació – Pregunta, $N - 1$, **optativa** – obligada*

Profe→Control

Semàntica i primeres conclusions

- En l'ES `nouControl()` recuperem un `:Control` a partir del seu identificador. A partir d'aquí deixem aquest `control` diferenciat, per tal de tenir un camí de comunicació pels diferents missatges
- Aquesta visibilitat, per tant, només té sentit durant l'execució del cas d'ús: tot i ser d'atribut l'enllaç es genera en cada nova petició de `nouControl()`
 - En ser una visibilitat que només té sentit durant l'execució d'un procés, **no apareix** com a associació a MC
 - Tot i així n'analitzem les seves propietats
- Per tant

activació : Profe – Control, 1–?

Multiplicitats que manquen

- Durant l'execució del CU un `:Control` per quants `:Profe` pot estar activat?
- Clarament el CU només coneix un `:Profe`
- L'ES `fiControl()`, per assegurar la correctesa i la robustesa de la seqüenciació, destrueix l'activació
 - Els DC no ens en diuen res d'això
 - Cal anar al DS, que sí que ens donen, per veure que cal la desactivació del `Control`
- Així, en cada moment només hi ha un control que tingui un professor diferenciat. I aquest `:Control` és justament el que també tenim diferenciat.
- Per tant:

activació : Profe – Control, 1 – 1

Obligatorietats i optativitats

- El :Profe pot o no tenir un :Control diferenciat
 - Justament la responsabilitat bàsica de l'ES nouControl() és fer aquesta activació
- El control pot estar activat o no. És més, sabem que només hi ha un control activat
 - Una de les responsabilitats de l'ES nouControl() és decidir quin és el :Control que cal activar

- Per tant:

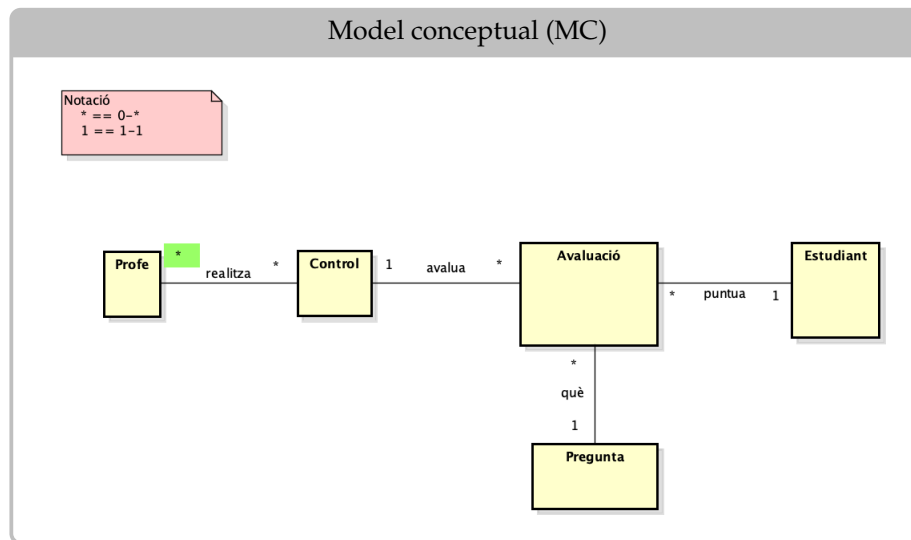
activació : Profe – Control, 1 – 1, optativa – optativa

Consideracions finals

- **Un detall innecessari**
 - Hem marcat en blau o verd algunes de les conclusions on hem arribat, on el color indica quin és l'origen d'aquesta conclusió
 - No cal mai arribar a aquest nivell de detall. Si aquí ho hem presentat així és per mostrar que els raonaments tenen diferents orígens:
 - * El que diuen i el que no diuen els DC
 - * La nostra interpretació de la semàntic dels DC
 - * Consideracions extres, definicions de glossari, etc, que no es poden extreure dels DC
- **Suposicions menys restrictives**
 - La notació adoptada per les multiplicitats prèn com a cas no marcat (aquell més curt d'escriure) el menys restrictiu
 - * 0-* és el cas no marcat. S'expressa com *, i indica moltes destinacions. Fixem-nos que el cas 1-* és més restrictiu: "Moltes destinacions, però com a mínim una"
 - * 1-1 és el cas no marcat, i s'expressa amb 1. Indica "Volem una destinació". El cas marcat 0-1 indica que voldríem una destinació, però no és necessari
 - Davant d'una elecció "a l'atzar" entre diferents supòsits, ens hem decantat sempre pel cas **no marcat**

* D'aquí que el model no té cap o gairebé cap marca explícita d'obligatorietat o optativitat

2.4.2 El model conceptual (MC) resultant



2.5. Conclusions sobre l'enginyeria inversa

Enginyeria inversa

- La lectura dels DC ens dona molta informació, però no la suficient
- Cal accedir a la semàntica del CU, a la semàntica del problema, o a criteris externs per a poder obtenir tota la informació necessària
- En conseqüència, l'enginyeria inversa és un procés complex, on es produeixen molts errors, i on sovint ens manca informació
- Això significa que fer primer el disseny, i a posteriori intentar extreure'n l'especificació no és, de cap de les maneres, una opció vàlida

MComp no és MC

- L'aspecte i els elements de **MComp** i de **MC** és força diferent
- L'enginyeria inversa no és una opció
 - Fer primer el disseny, i a posteriori intentar extreure'n l'especificació (enginyeria inversa), és un procés complex que cal alimentar amb molts supòsits addicionals
 - El resultat pot ser totalment arbitrari, complex o fosc. I en general estarà molt lluny d'una especificació mínimament decent