

## Iteració d'un CU

Setembre de 2021

Un CU s'ha de poder executar tants cops com es vulgui



- Un **CU** expressa un procés que es pot iterar
- Una iteració en un **ES** expressa que dins del procés que correspon al CU es pot enviar de manera iterada una mateixa petició o conjunt de peticions



### 1. Iteració d'un CU

- Un CU és un **ús prototípic del sistema**, i per tant s'ha de poder executar tants cops com es vulgui, sempre que es compleixin els condicions exigides per a la seva execució
  - Un CU que només es pugui executar un cop no té cap mena de sentit com a "ús prototípic"
- A tots els efectes podem considerar que tot CU té una **iteració externa implícita**
  - L'anàlisi de la **robustesa** i **correctesa** de la seqüència ha de considerar com a vàlida la subsequència següent:

darrerESdelCU + primerESdelCU



Correctesa i  
robustesa en  
un DS  
(0030-0020-0020)

### Exemple 1. Iteració (implícita) d'un CU

- Sigui el CU *comprar*, amb el següent DS:
 

```
iniciCompra(client)
compra(producte, quant)*
fiCompra()
```
- Aquest CU es pot executar tants cops com es vulgui
  - Per iniciar el CU cal assegurar les PRE de *iniciCompra(client)*
  - durant l'execució, cal assegurar les PRE de l'ES enviat, i que és un dels permesos per la seqüència
- Això significa que l'ES *fiCompra()* ha de deixar el sistema en unes condicions que sigui possible complir amb les PRE de *iniciCompra(client)*

## 2. Iteració d'un CU versus iteració d'un ES

### Exemple 2. Iteració d'un ES

- Introduïm els nivells de  $CO_2$  detectats a la ciutat<sup>1</sup>
- Disposem de l'ES següent, que té la responsabilitat d'introduir en el sistema una nova lectura amb el seu valor i una marca de temes:

lectura(nivell, hora)

#### 1. Iteració d'un ES

- Les lectures arriben en bloc
- El DS proposat és:
 

```
lectura(nivell, hora)*
```

#### 2. Iteració d'un CU

- Les lectures arriben aleatòriament
- El DS proposat és:
 

```
lectura(nivell, hora)
```

- Si les lectures arriben en **bloc**, el procés d'introducció de les lectures exigeix introduir totes les lectures del bloc, una rera l'altra
  - Cal iterar una petició, la d'introduir al sistema una nova lectura amb la seva informació. És a dir, cal iterar l'ES *novaLectura(...)*
  - El CU, com a tal, també es pot iterar. En cada execució del

CU s'introdueix un nou bloc; i per introduir aquest cal una iteració sobre les lectures

- Si les lectures arriben aleatòriament, el CU només pot ser responsable d'introduir una lectura. D'aquí que no cal cap iteració
  - Si iterem l'ES, estem dient que hi ha un conjunt de lectures que s'introdueixen en un sol moment, en un sol lloc, i per una mateixa persona; però l'aleatorietat de l'arribada de les lectures diu, com a mínim, que la introducció no es fa en el mateix moment
  - Tot i que el CU només sigui responsable d'introduir una lectura, el podem iterar tants cops com calgui. D'aquesta manera es poden introduir múltiples lectures

Cal tenir present que un CU és un ús del sistema. Des d'un punt de vista de rendiment (ara ens allunyem de l'especificació) executar un CU pot significar, entre d'altres:

- Generar canals de comunicació de l'usuari amb el sistema, com sockets o accessos a una base de dades
- Fer comprovacions de seguretat (autenticitat i autorització de l'usuari)

L'inici d'execució d'un CU, per tant, pot ser car en recursos. Així sembla preferible iterar dins del CU que no pas iterar el propi CU. El problema és que un CU s'ha de realitzar en un temps curt (altrament estem bloquejant el sistema), per un mateix usuari i un mateix lloc. I com hem vist això no sempre és possible, i no hi ha més remei que iterar el CU.

- Si la informació arriba en **blocs**, cal iterar dins del CU
- Si la informació pot arribar en **qualsevol moment**, cada nova informació cal considerar-la com una nova execució del CU



<sup>1</sup>Els possibles DS d'aquest problema els hem analitzat a [Iteració sense factorització](#). Allà analitzàvem les diferents possibilitats de fer una iteració dins del CU. Ara comparem una d'aquestes possibilitats amb la iteració de tot el CU.

