



Sviluppo delle applicazioni software

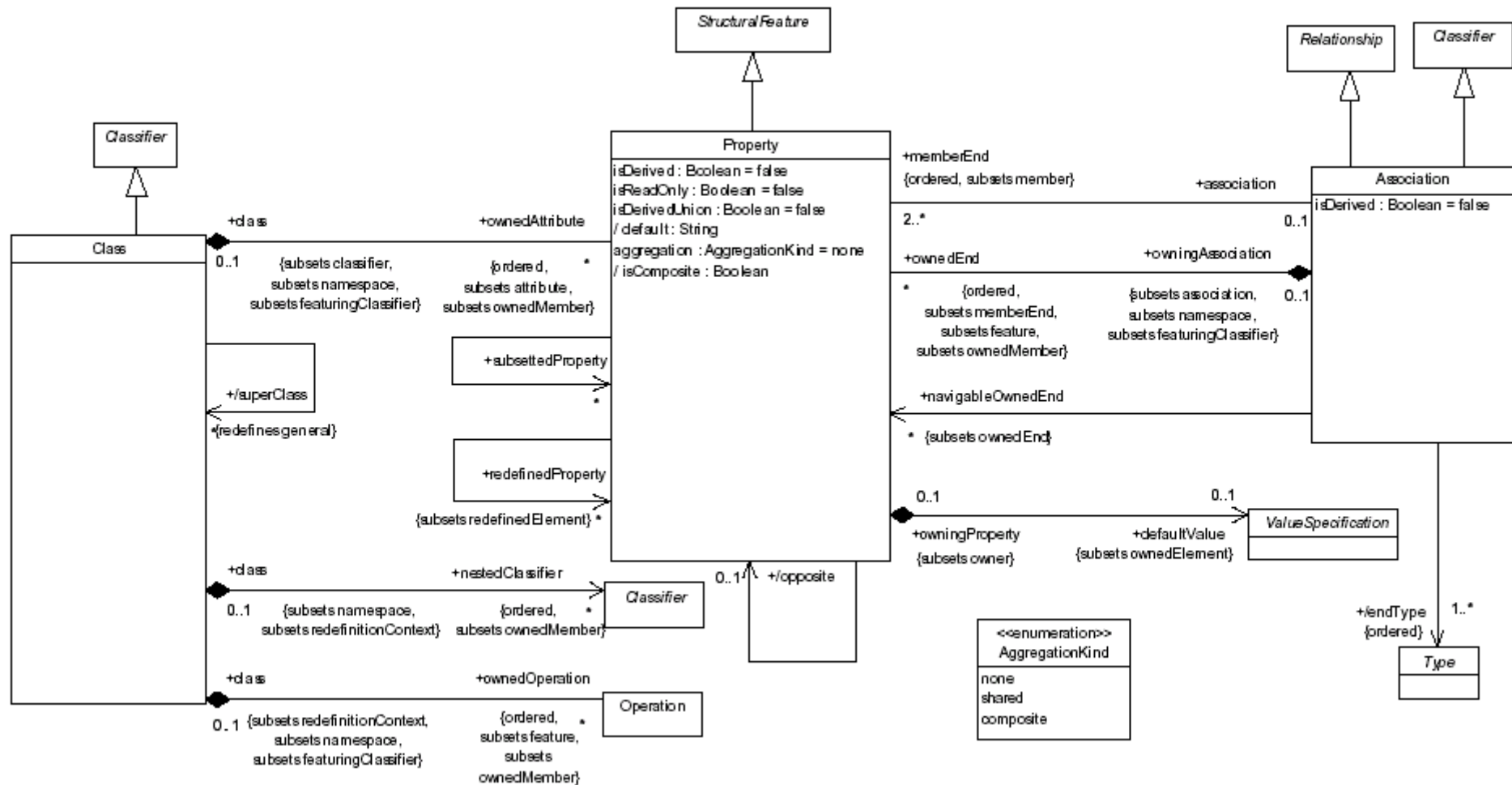
- Panoramica su Unified Modeling Language

Grazie a Simona Bernardi (Universita` di Saragoza, Spagna), che ha messo a disposizione il suo materiale

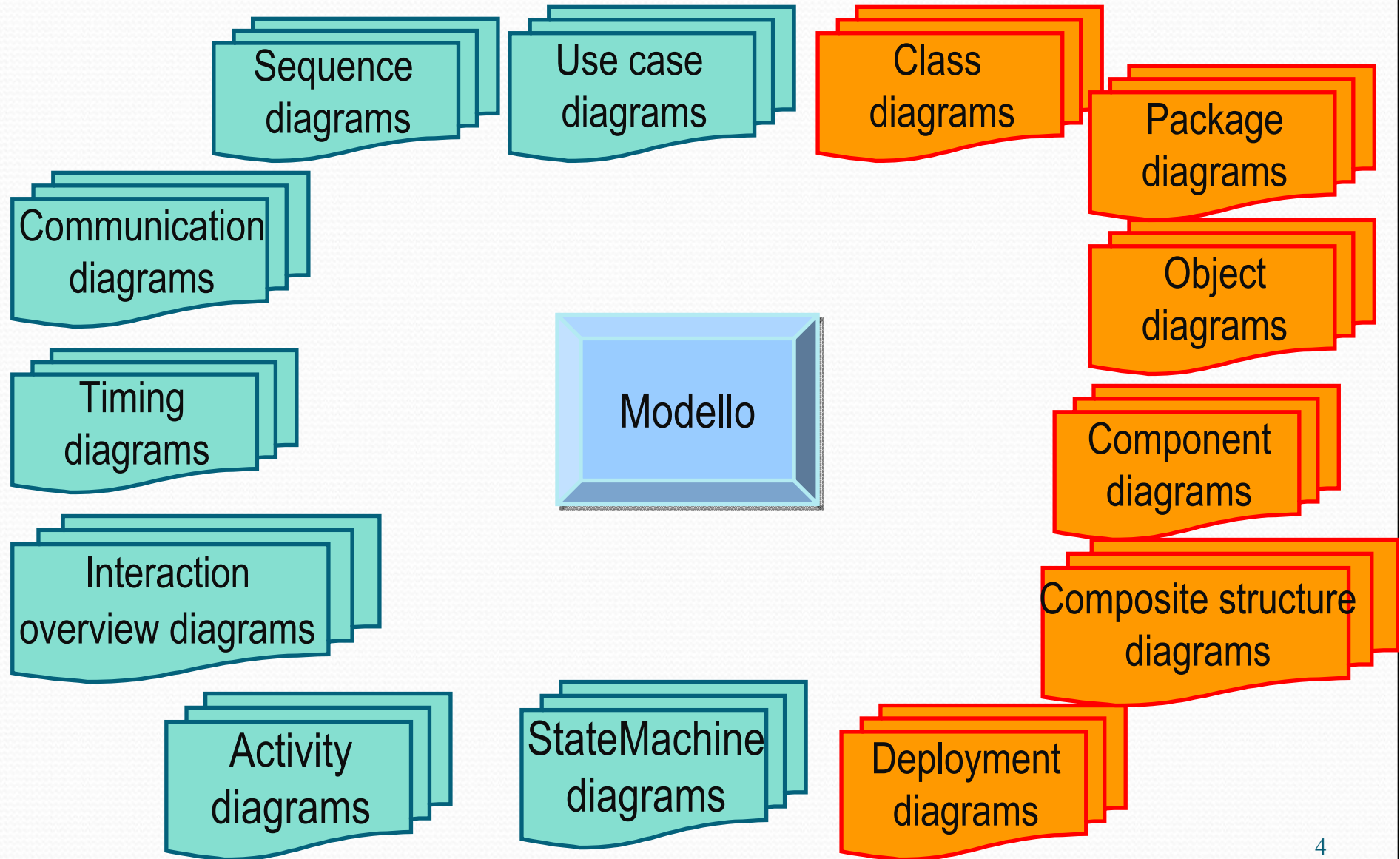
Introduzione

- E' un linguaggio di modellazione semi-formale
 - Notazione grafica
 - Regole di costruzione modelli
 - Semantica non definita completamente
 - Meta-modello
 - Ogni meta-classe è descritta con linguaggio naturale e con Object Constraint Language (OCL)

Meta-modello: Kernel package



Modellazione con UML



Diagrammi di interazione

- Diagrammi di sequenza
 - Ordinamento (temporale) dei messaggi scambiati tra gli oggetti sw
- Diagrammi di comunicazione
 - Collaborazione tra gli oggetti sw
- Interaction overview diagrams
 - Relazioni ad alto livello tra i diversi diagrammi di sequenza/comunicazione
- Timing diagram
 - Rilevanza sul tempo di occorrenza degli eventi che causano cambiamenti di stato tra gli oggetti sw che comunicano

Altri diagrammi dinamici...

- Diagrammi dei casi d'uso
 - Funzionalità/modi di utilizzo del sistema, sugli utilizzatori ed altri sistemi che interagiscono con il sistema
- Macchine a stati
 - Modellazione del comportamento (reattivo) di classi/oggetti/componenti
- Diagrammi di attività
 - Specifica dei flussi delle informazioni, attività interne, etc..

Diagrammi statici

- Diagrammi delle classi
- Diagrammi degli oggetti
- Diagrammi dei package
- Diagrammi delle componenti
 - Rappresentazione di componenti, interfacce richieste/offerte
- Diagrammi delle strutture composte
 - Rappresentazione interna di classi/componenti complesse, collaborazioni tra oggetti
- Diagrammi di sviluppo
 - Specifica della architettura sw/hw, allocazione run-time componenti sw sui nodi (hardware)

Estensioni UML - I

- UML contiene meccanismi che permettono di estendere le meta-classi (classi che definiscono gli elementi di modellazione UML) per diversi scopi
 - Differenti piattaforme (es., EJB, CORBA)
 - Differenti domini di applicazione (es., real-time, business process modeling)
- Profiling e' una tecnica di meta-modellazione
 - Stereotipi sono meta-classi specifiche
 - Tags sono meta-attributi (degli stereotipi)
 - Profili sono dei tipi di package UML

Estensioni UML - II

- Uno stereotipo definisce come una meta-classe debba essere estesa
- Quando uno stereotipo viene applicato ad un elemento del modello, i valori assegnati ai tag sono detti tagged-values

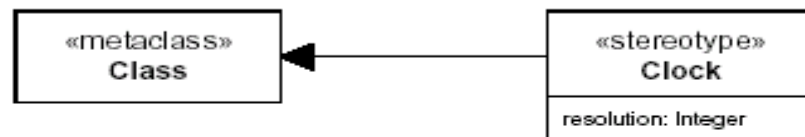


Figure 18.12 - Defining a stereotype

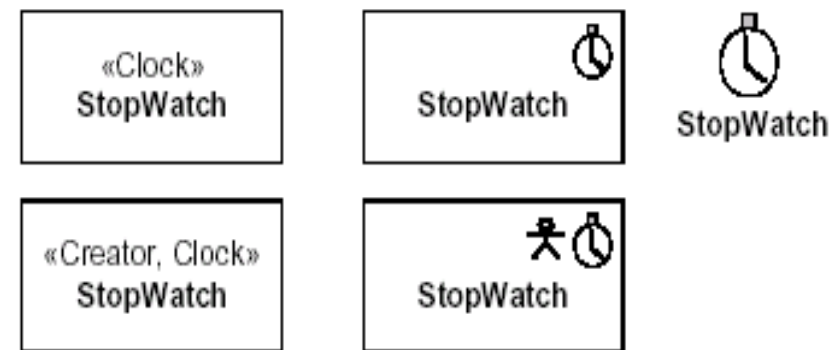


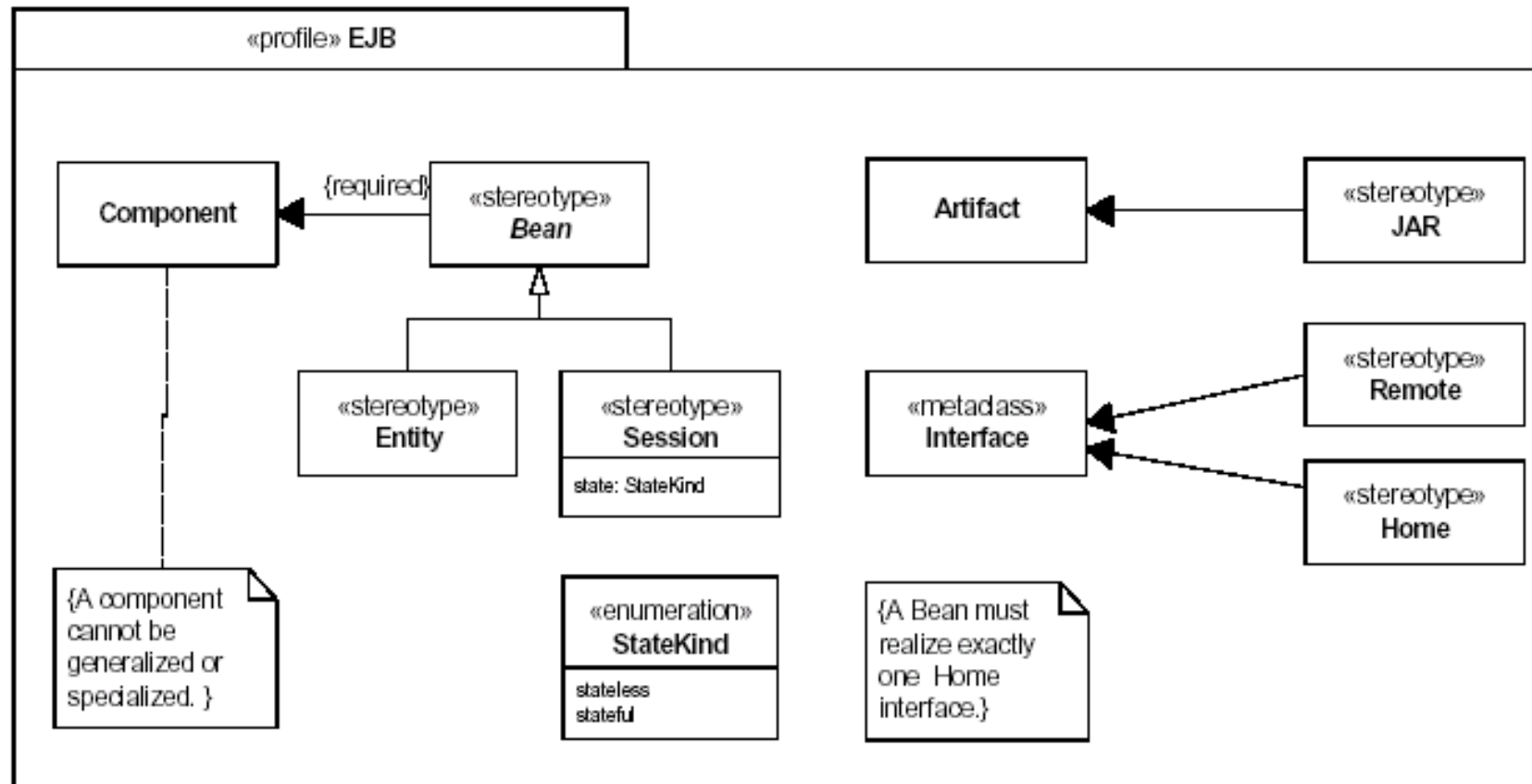
Figure 18.13 - Presentation options for an extended class

***Livello meta-modellazione
(definizione stereotipi)***

***Livello modellazione
(applicazione stereotipi)***

Estensioni UML - III

- Un profilo e' un tipo di package UML che contiene un insieme di stereotipi (tra loro eventualmente relazionati), tag e vincoli (espressi con OCL) da utilizzare in un determinato dominio di applicazione/piattaforma.



Modellazione con UML

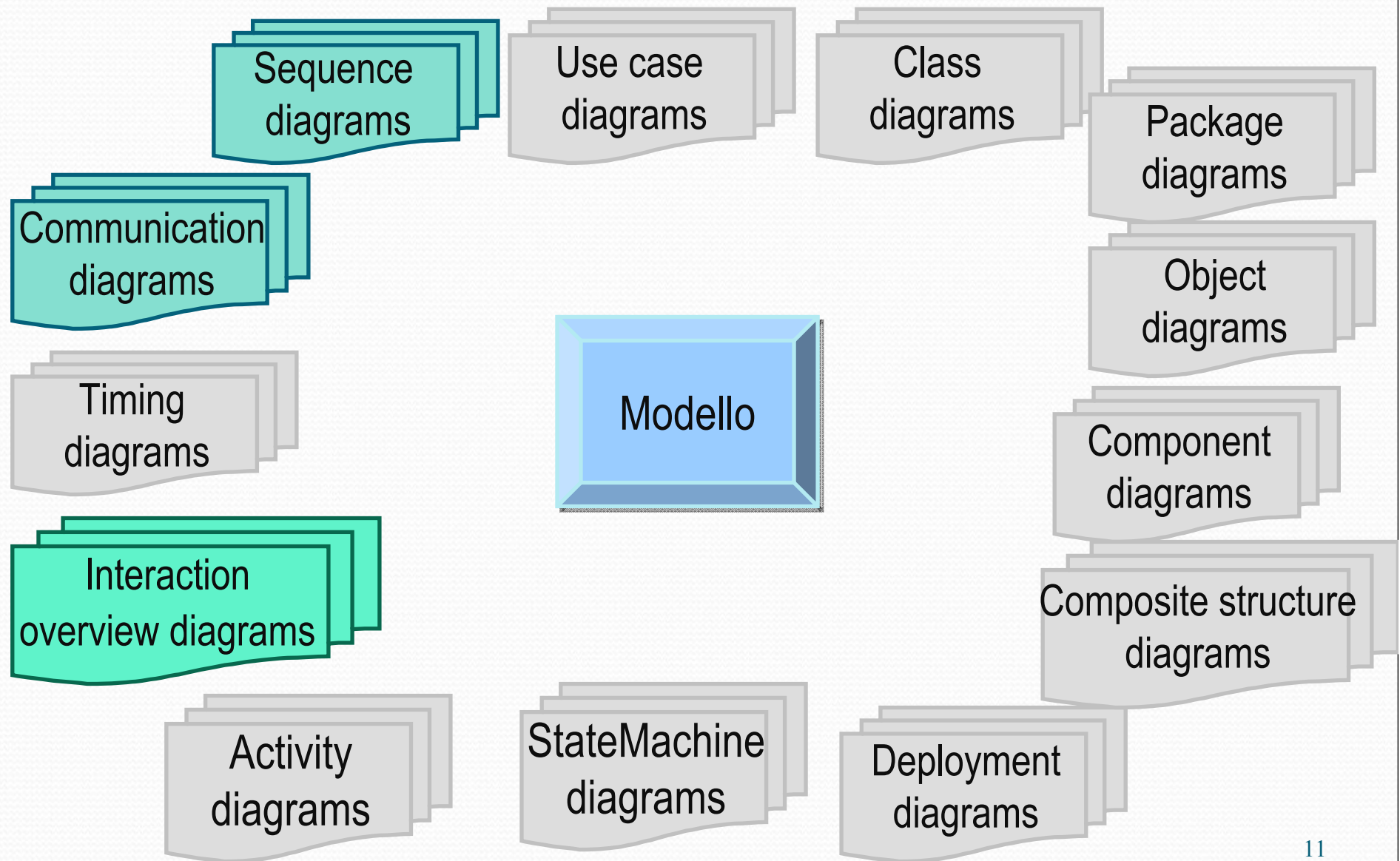


Diagramma di sequenza

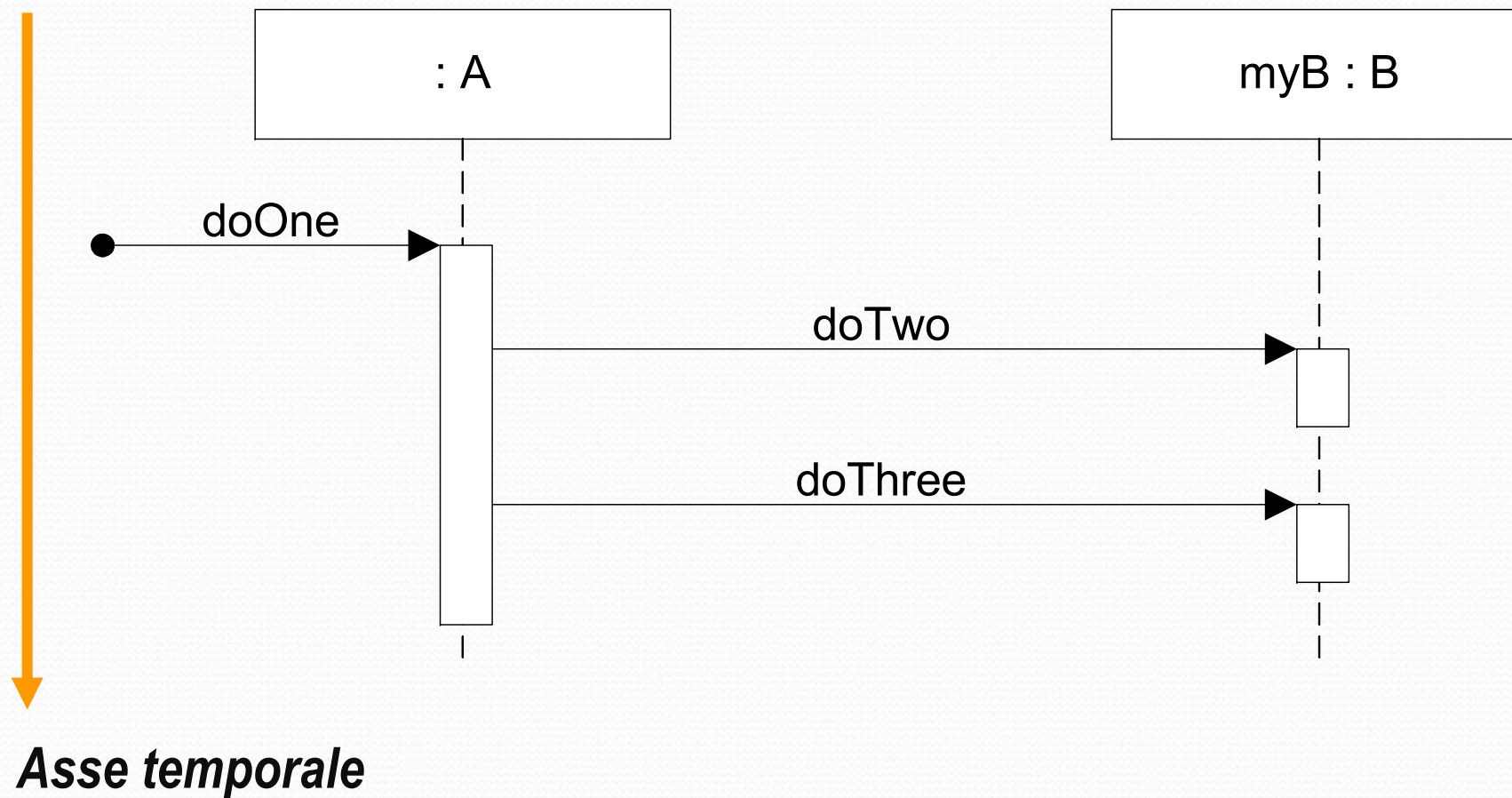
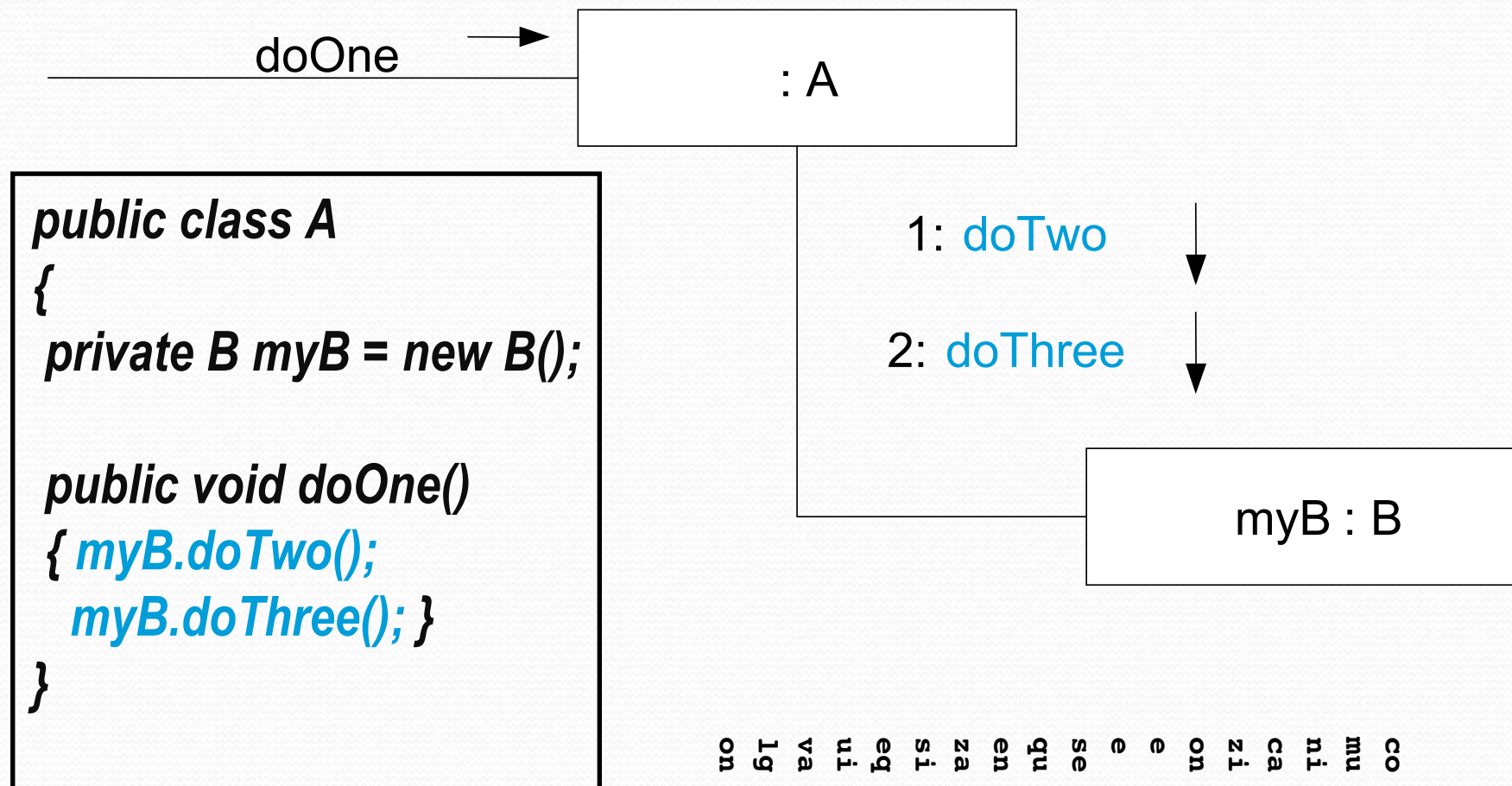
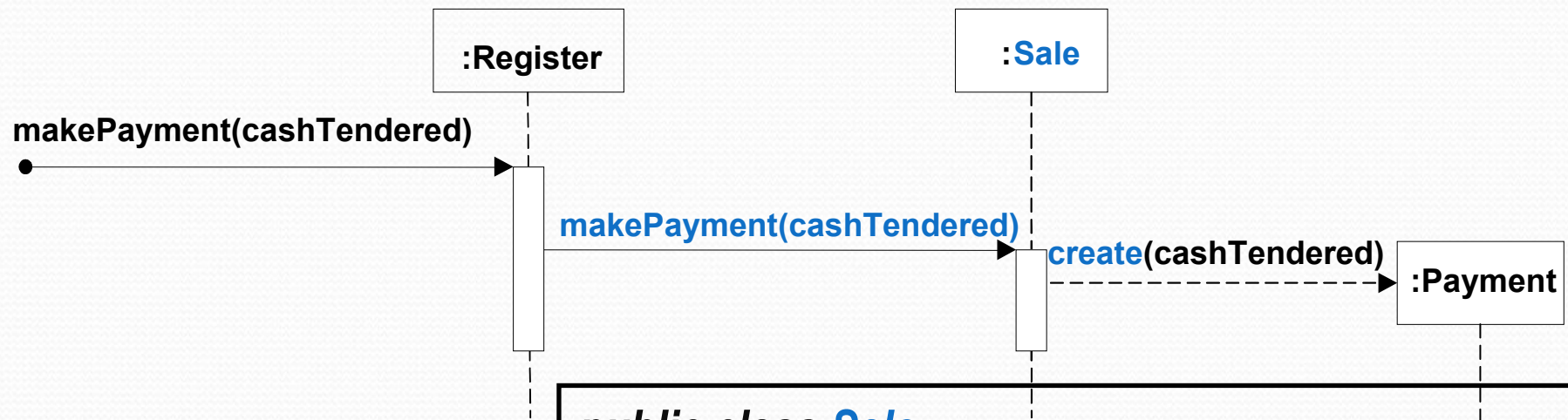


Diagramma di comunicazione



on
lg
va
ui
eq
si
za
en
qu
se
e
e
on
zi
ca
ni
mu
co

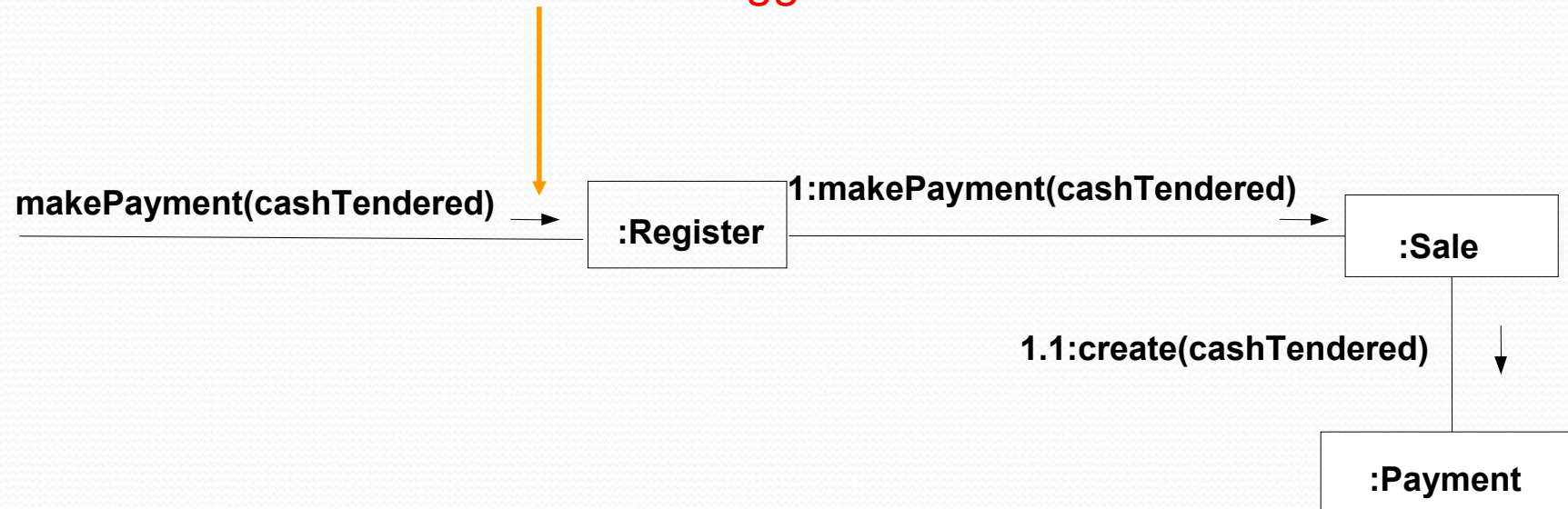
Esempio SD: POS



```
public class Sale
{private Payment payment;
  public void makePayment(Money cashTendered)
  {
    payment = new Payment(cashTendered);
    .....
  }
  .....
}
```


Esempio Comm.D: POS

Direzione del messaggio



Lifelines: partecipanti

*Istanza generica
della classe Sale*

:Sale

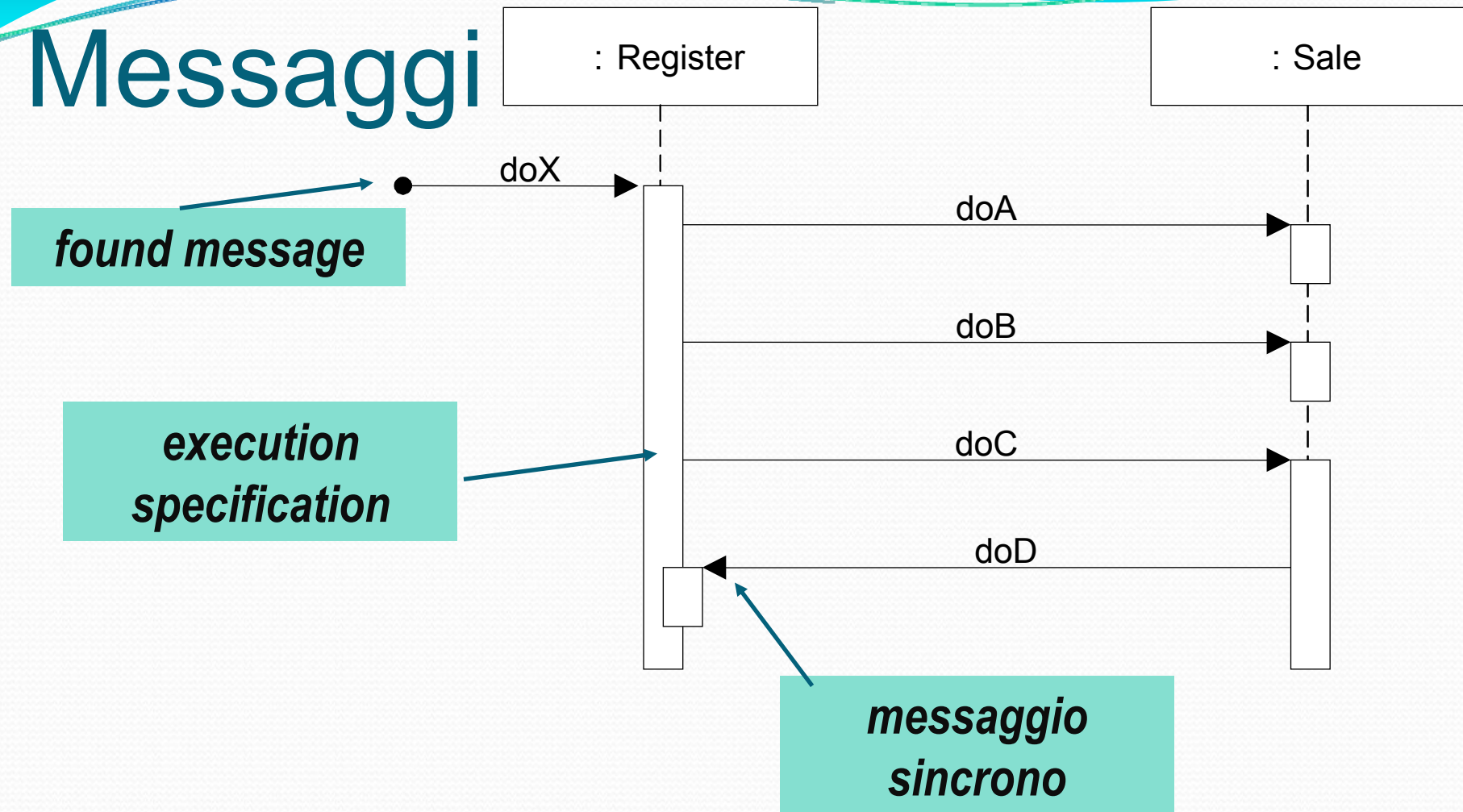
*Istanza (nominata)
s1 di Sale*

s1:Sale

Istanza i-esima di classe Sale

sales[i]:Sale

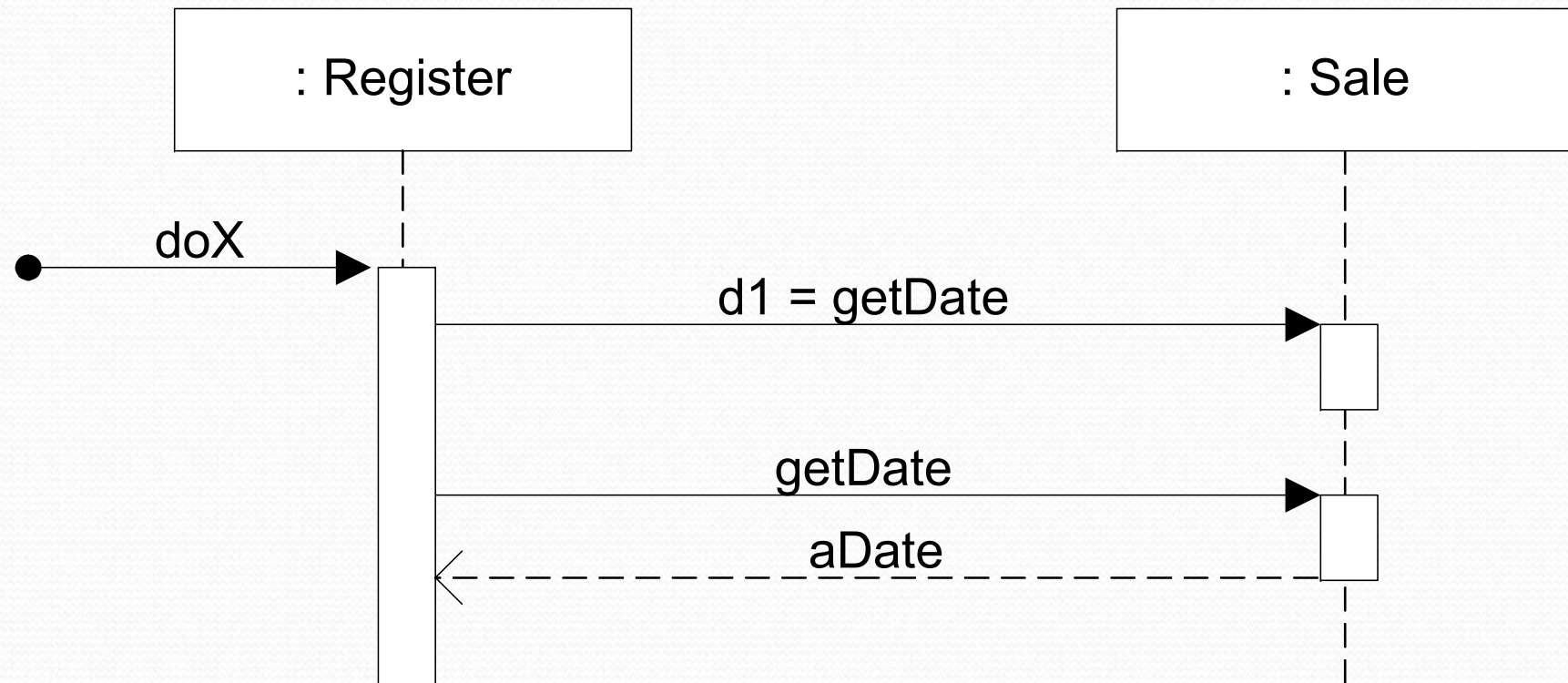
Messaggi



Sintassi:

`return = message(parameter: ParameterType):returnType`

Return



ione

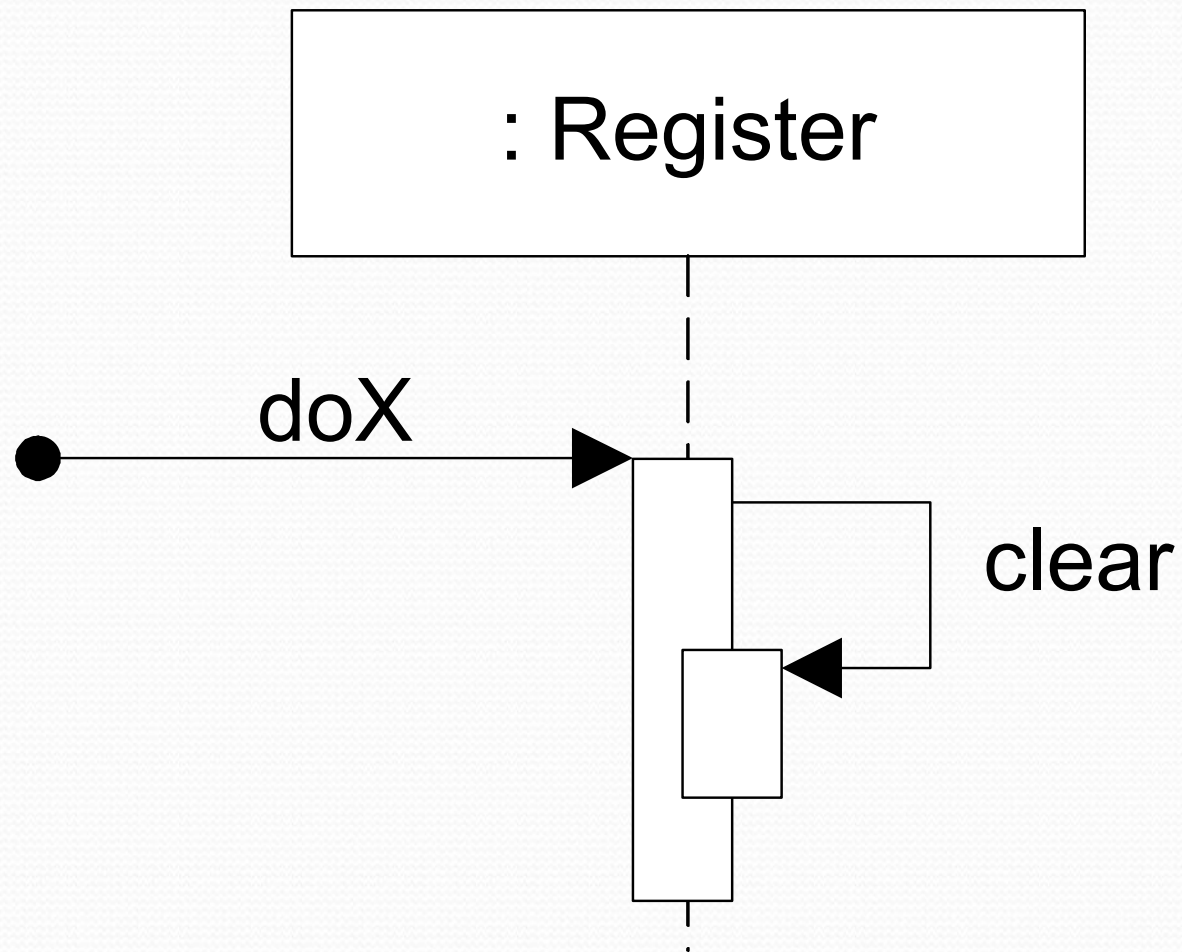
caso

cioè

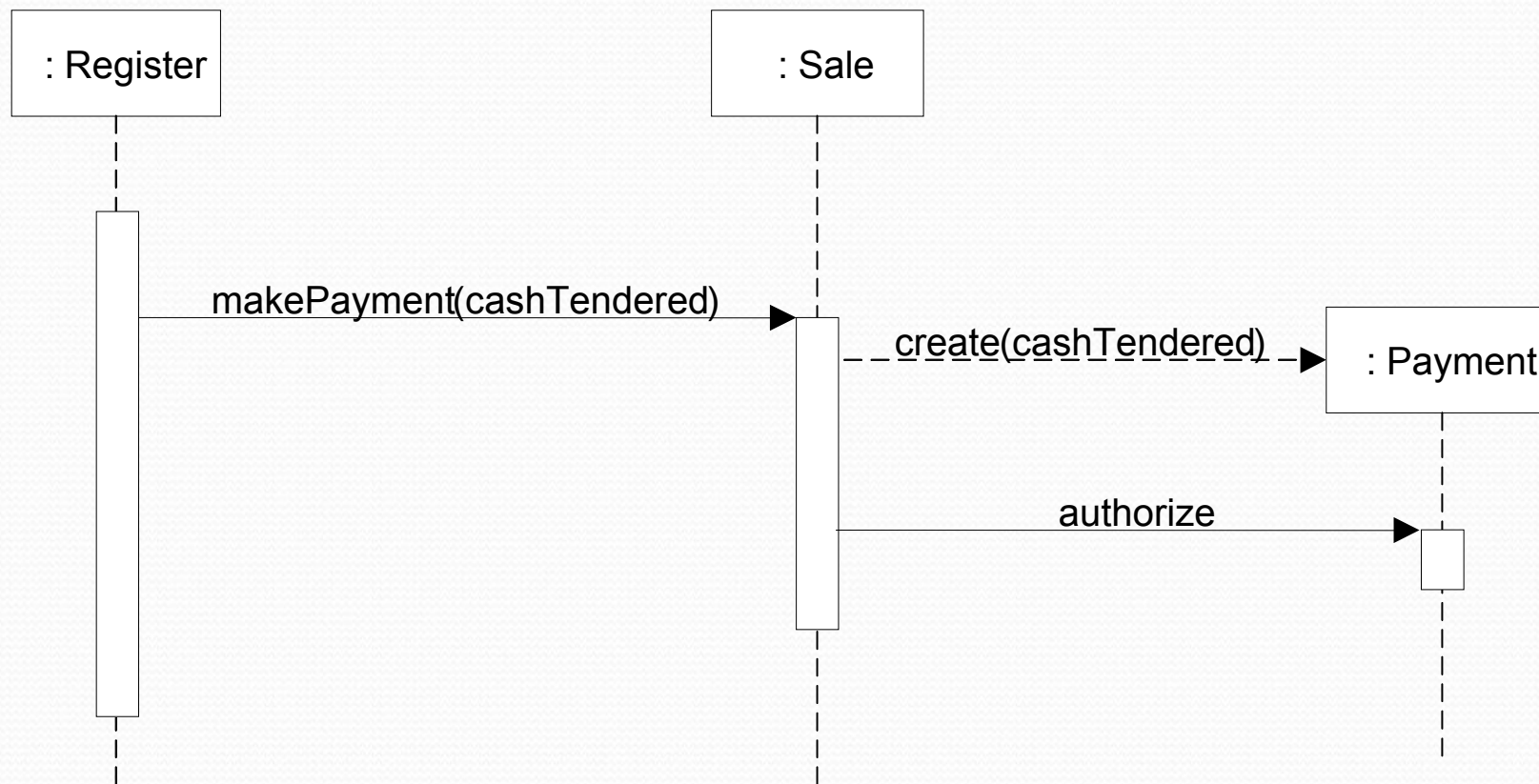
orno

iata

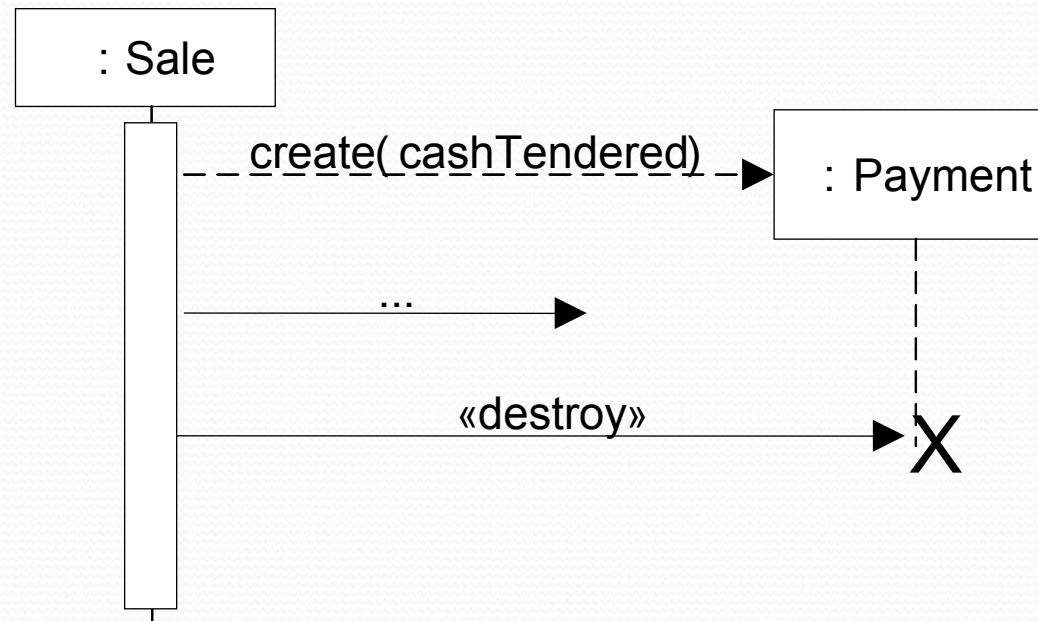
Messaggi self



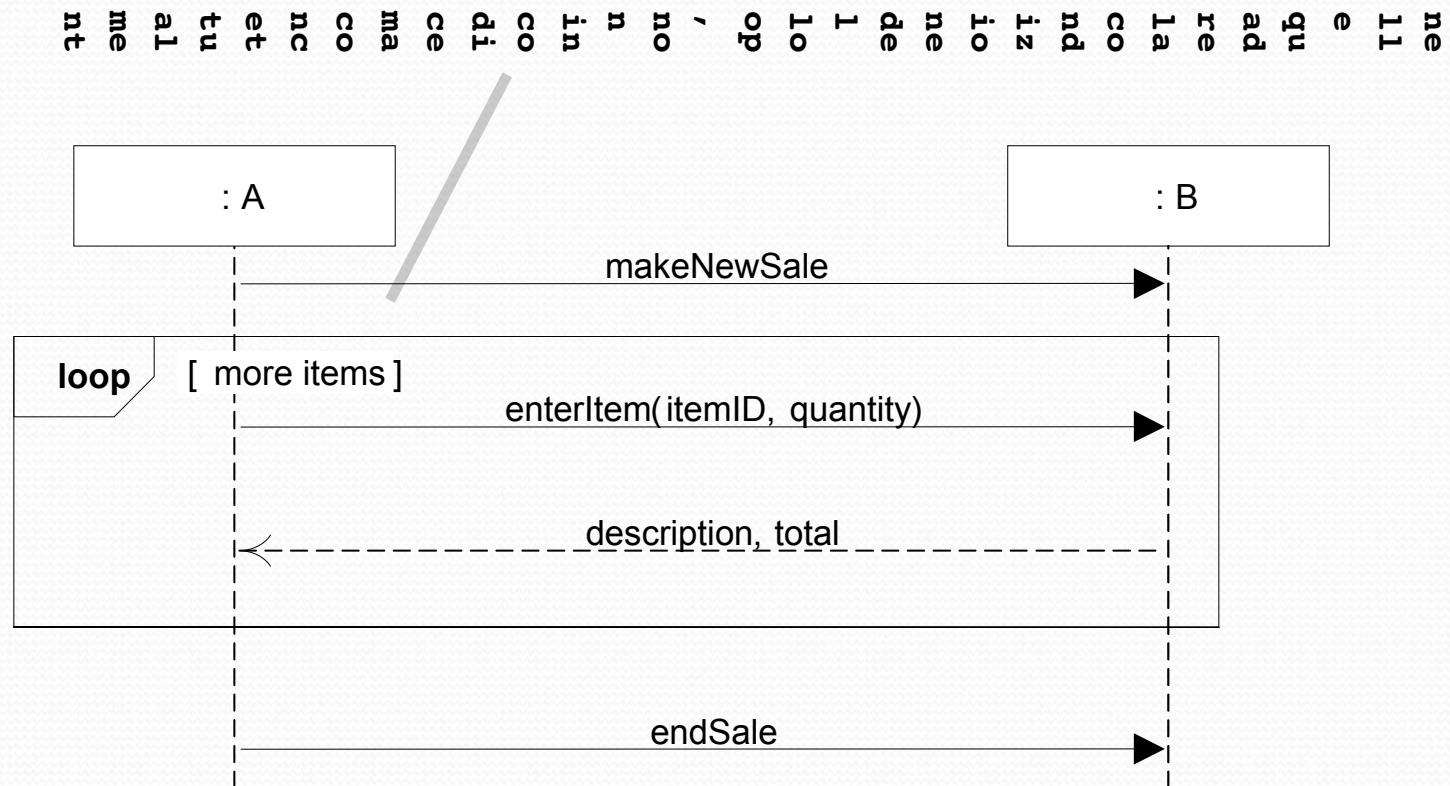
Creazione di istanze



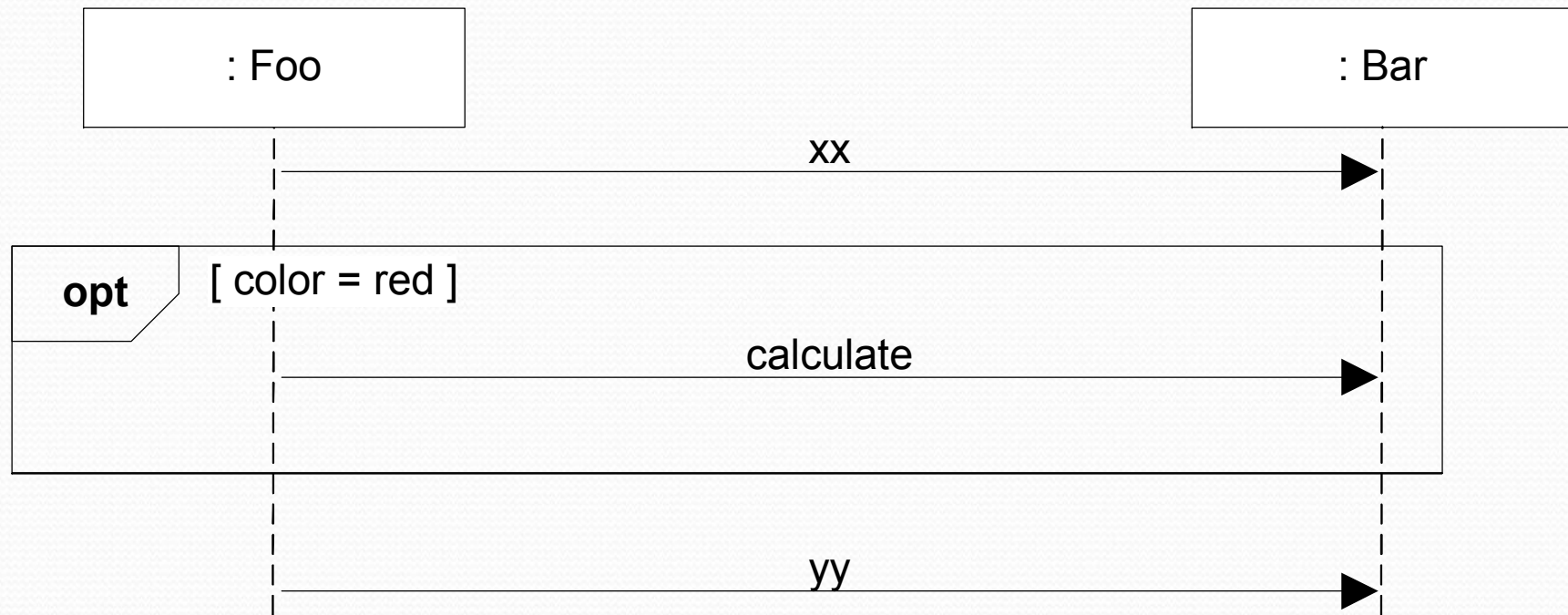
Distruzione oggetti



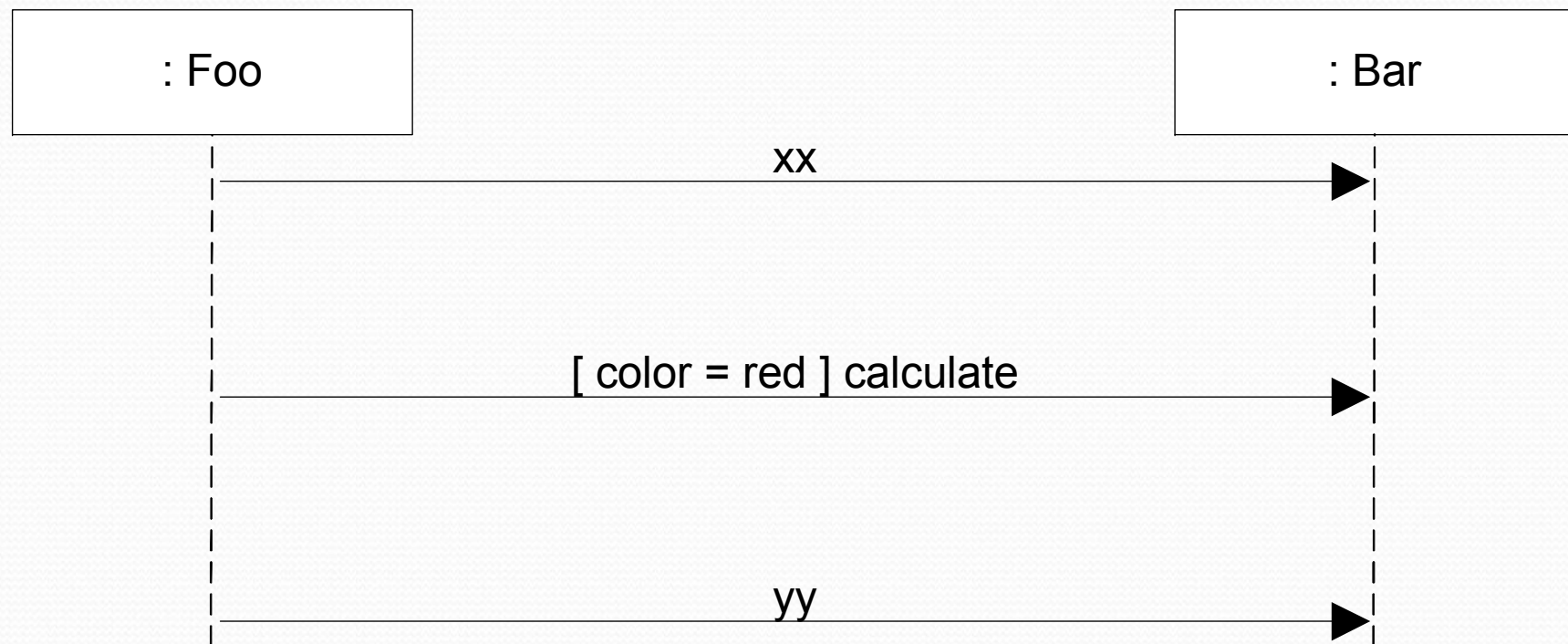
Combined fragment (loop)



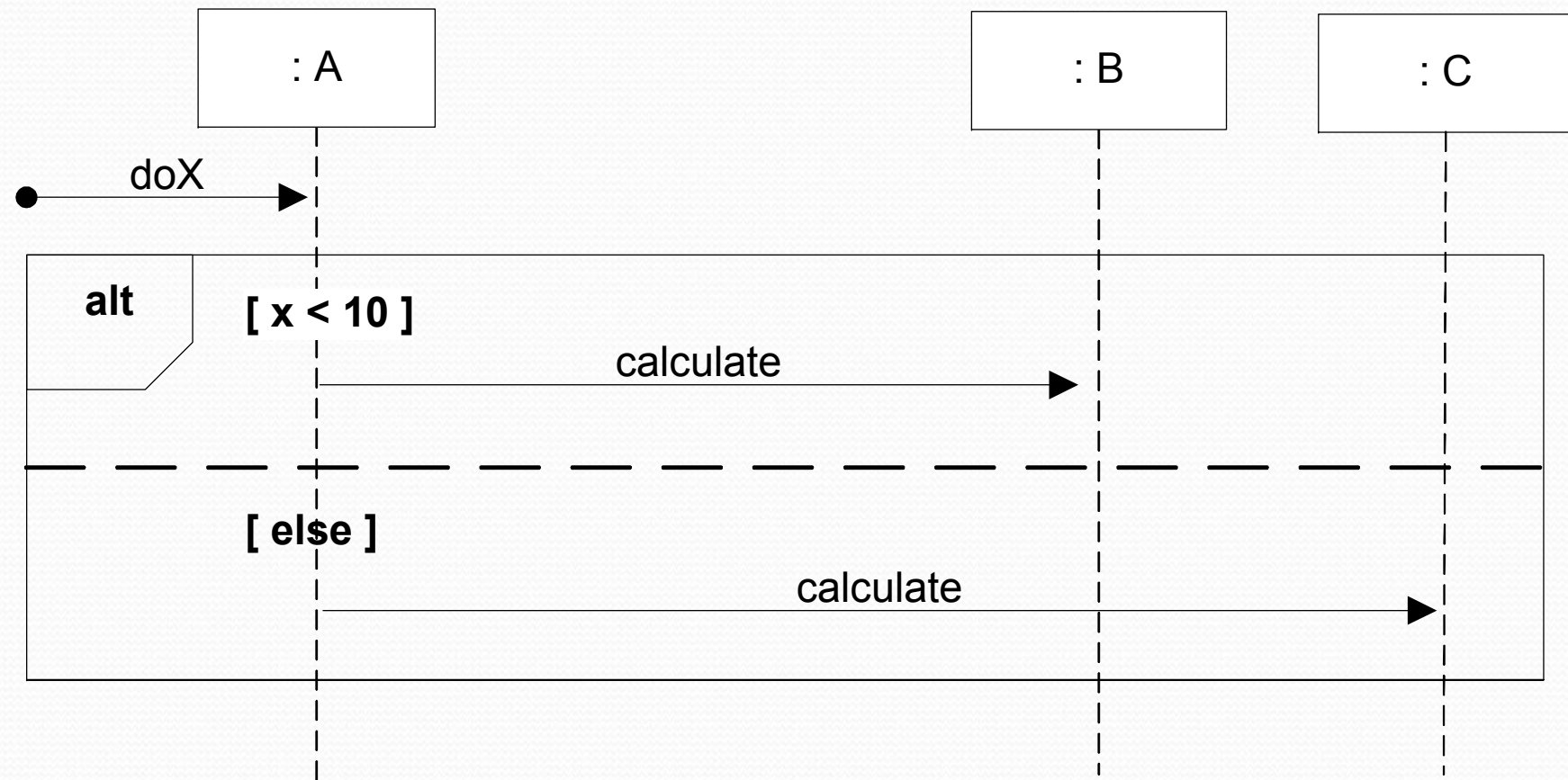
Combined fragment (opt)



Alternativa a “opt” di un messaggio

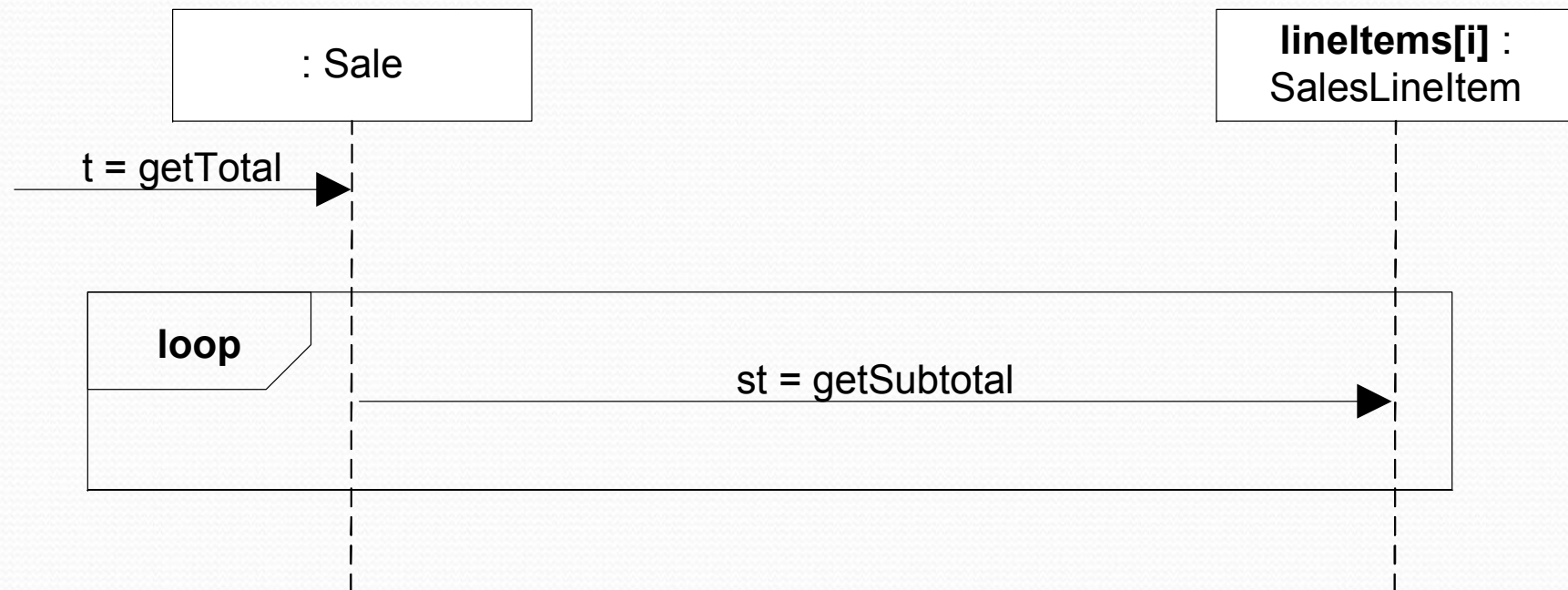


Combined fragment (alt)

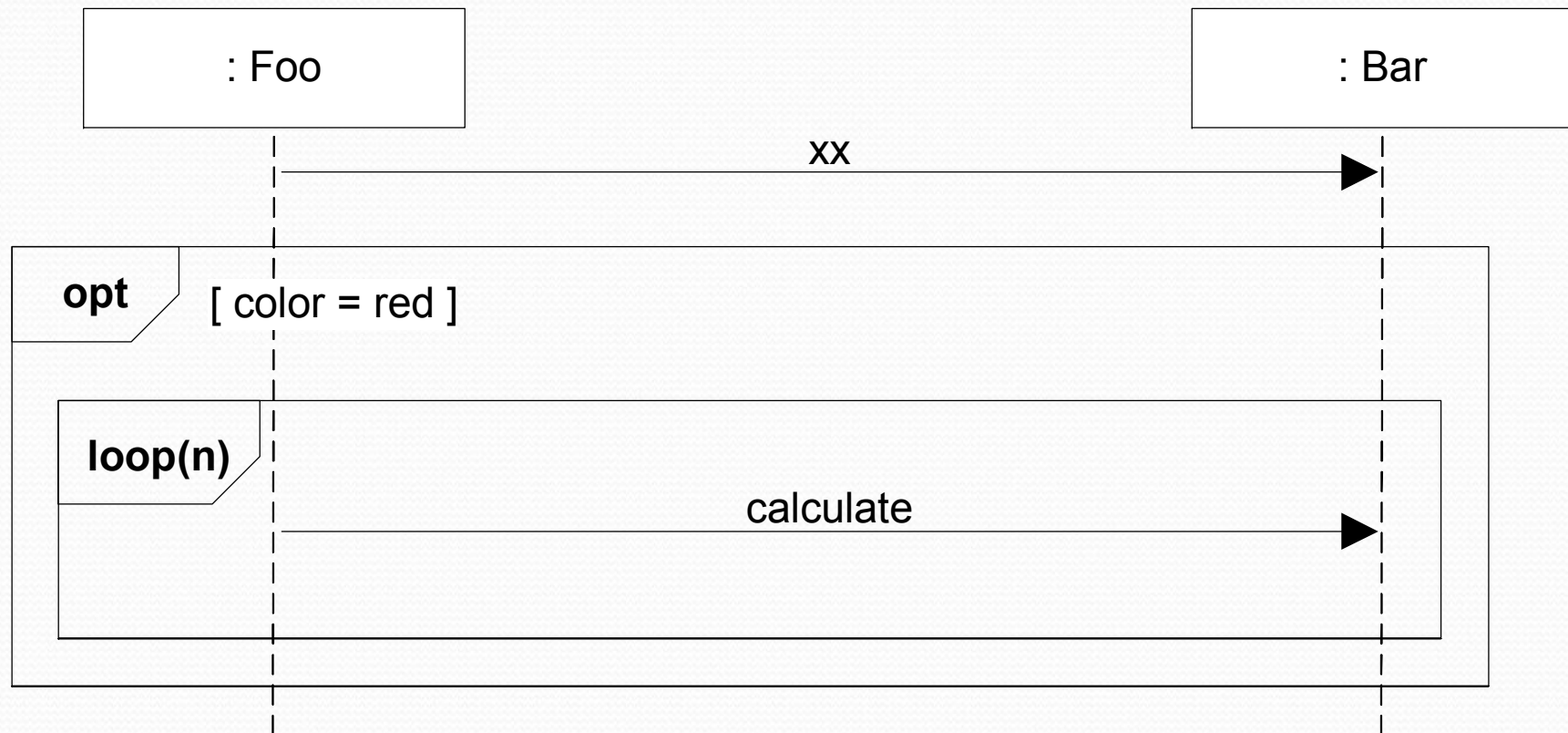


Iterazione su una collezione di oggetti – I

for
each
item
in
collection

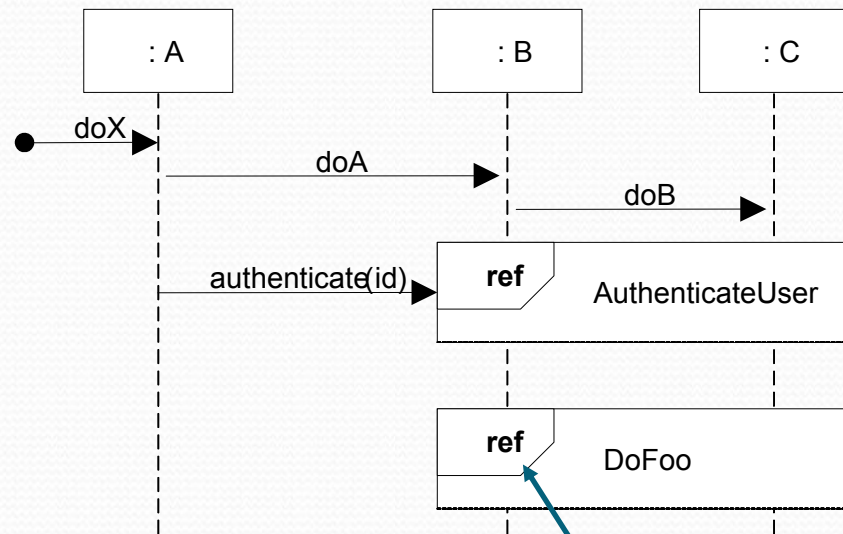


Nesting di combined fragment

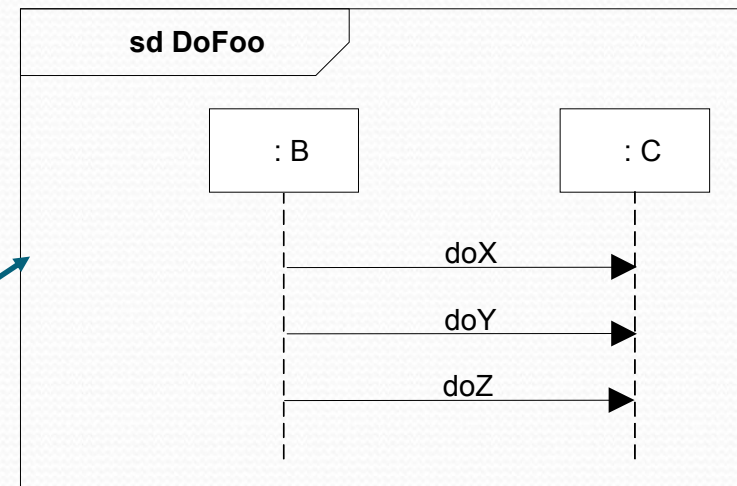
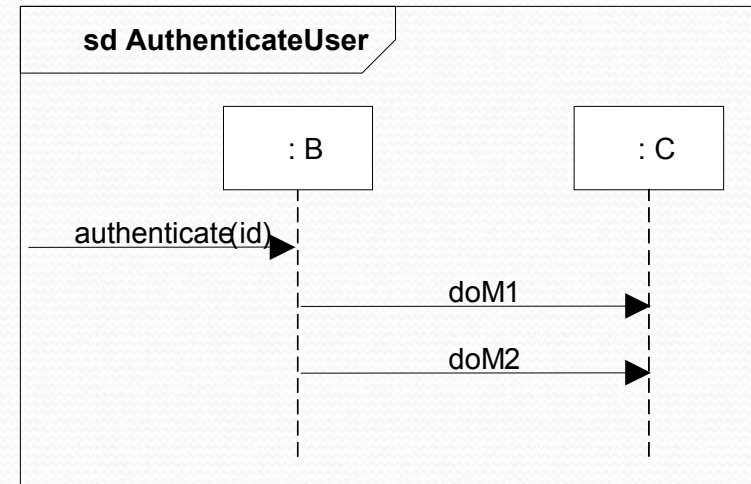


Relazionare SD

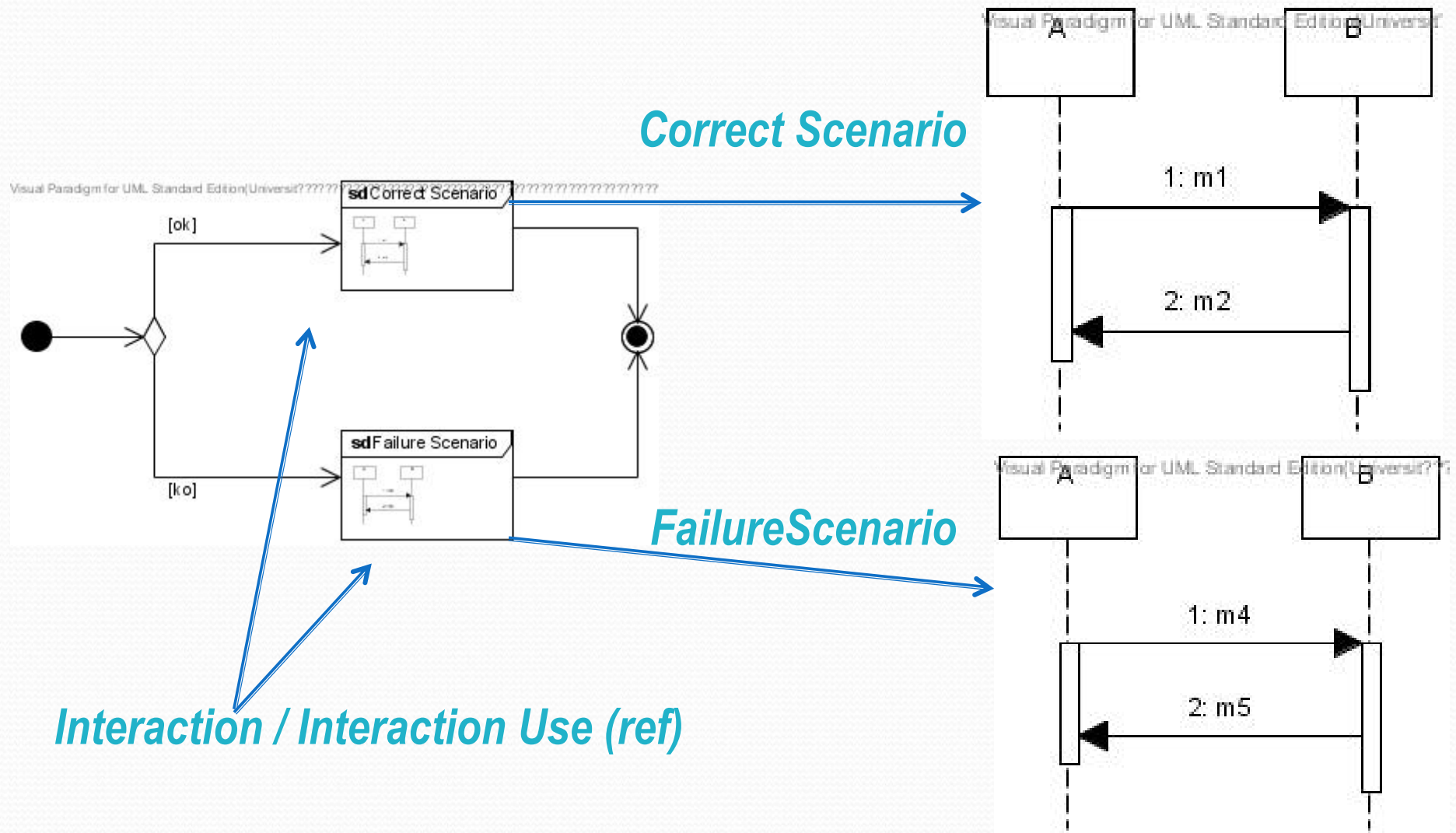
meta
o di
a di



***interaction occurrence:
riferimento ad una interazione
all'interno di un'altra***

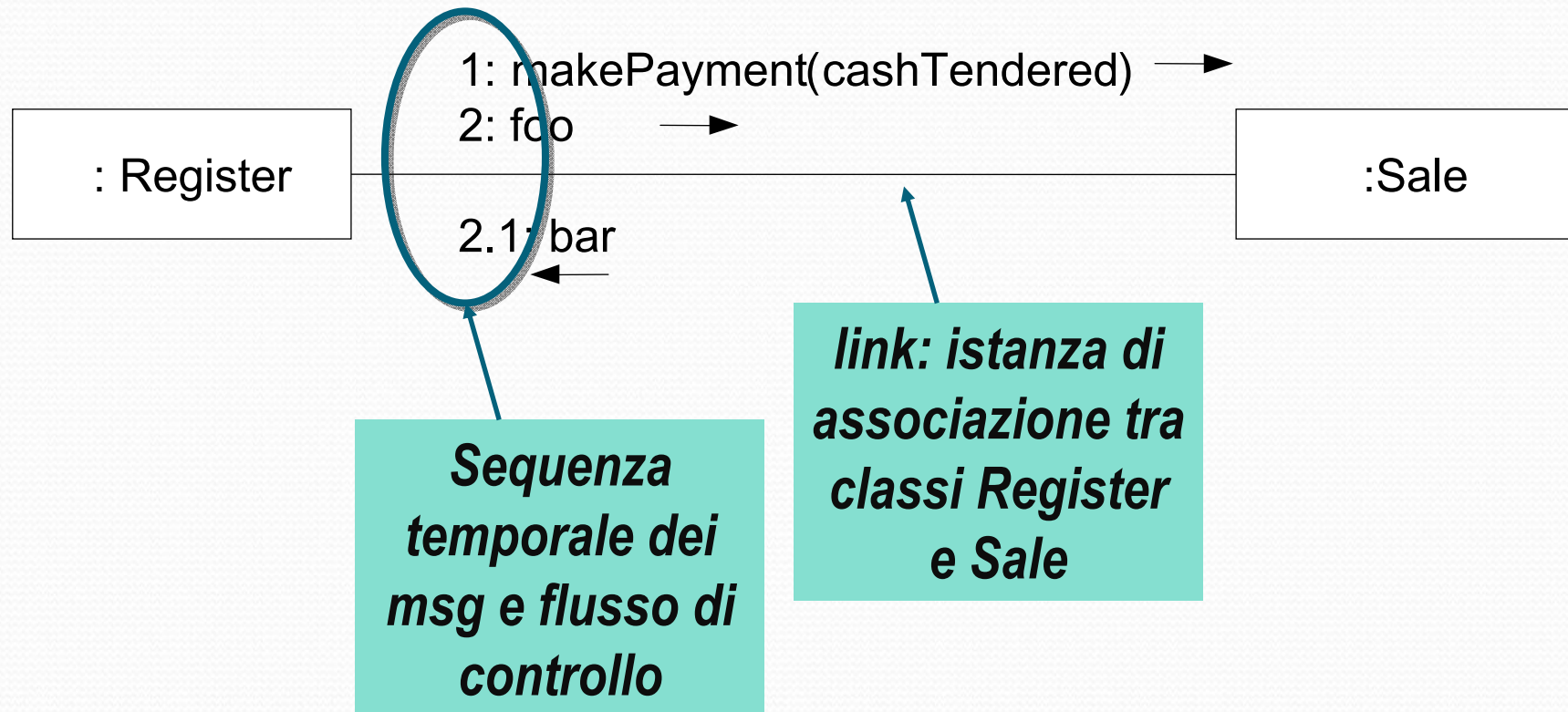


Uso dell'Interaction Overview Diagram

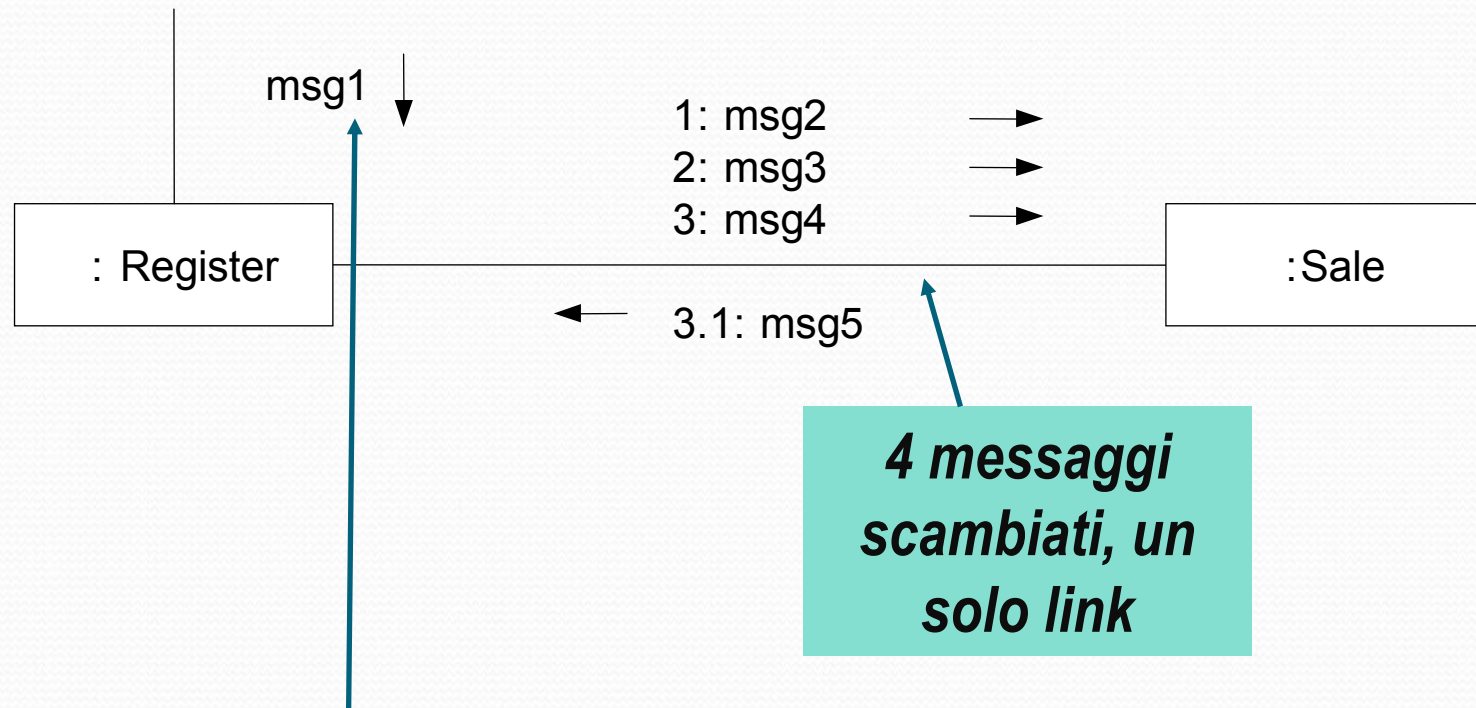


Diagrammi di comunicazione

La nuova maniera di comunicare è impostata per consentire di realizzare applicazioni di computer in modo più semplice e veloce. In alcuni casi, invece, si può avere bisogno di eseguire operazioni più complesse.

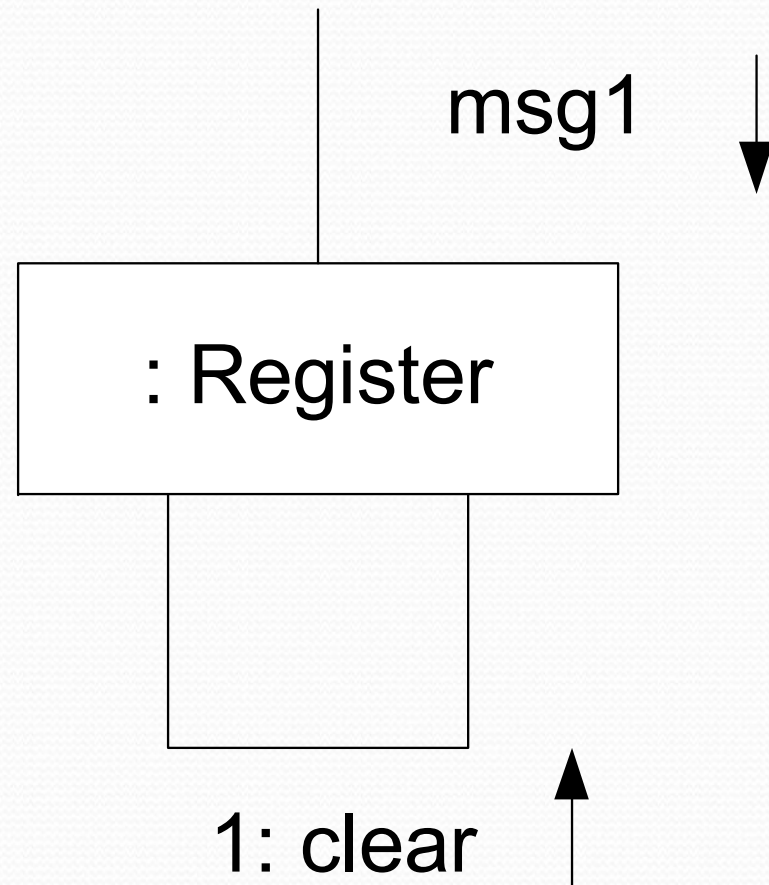


Messaggi



Non numerare il messaggio iniziale (msg1): semplifica la numerazione dei messaggi successivi – si evita di mettere un ulteriore livello di innestamento

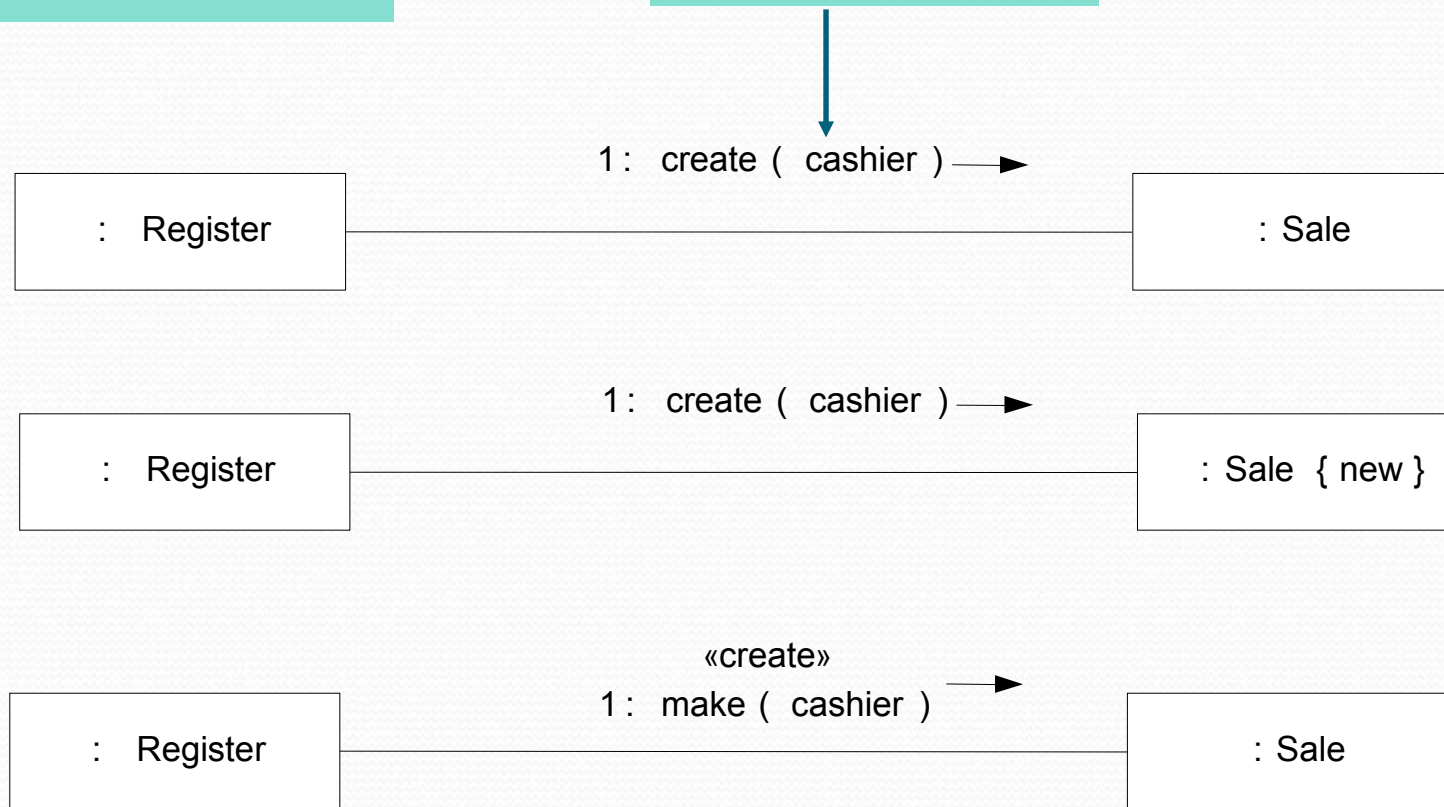
Messaggio self



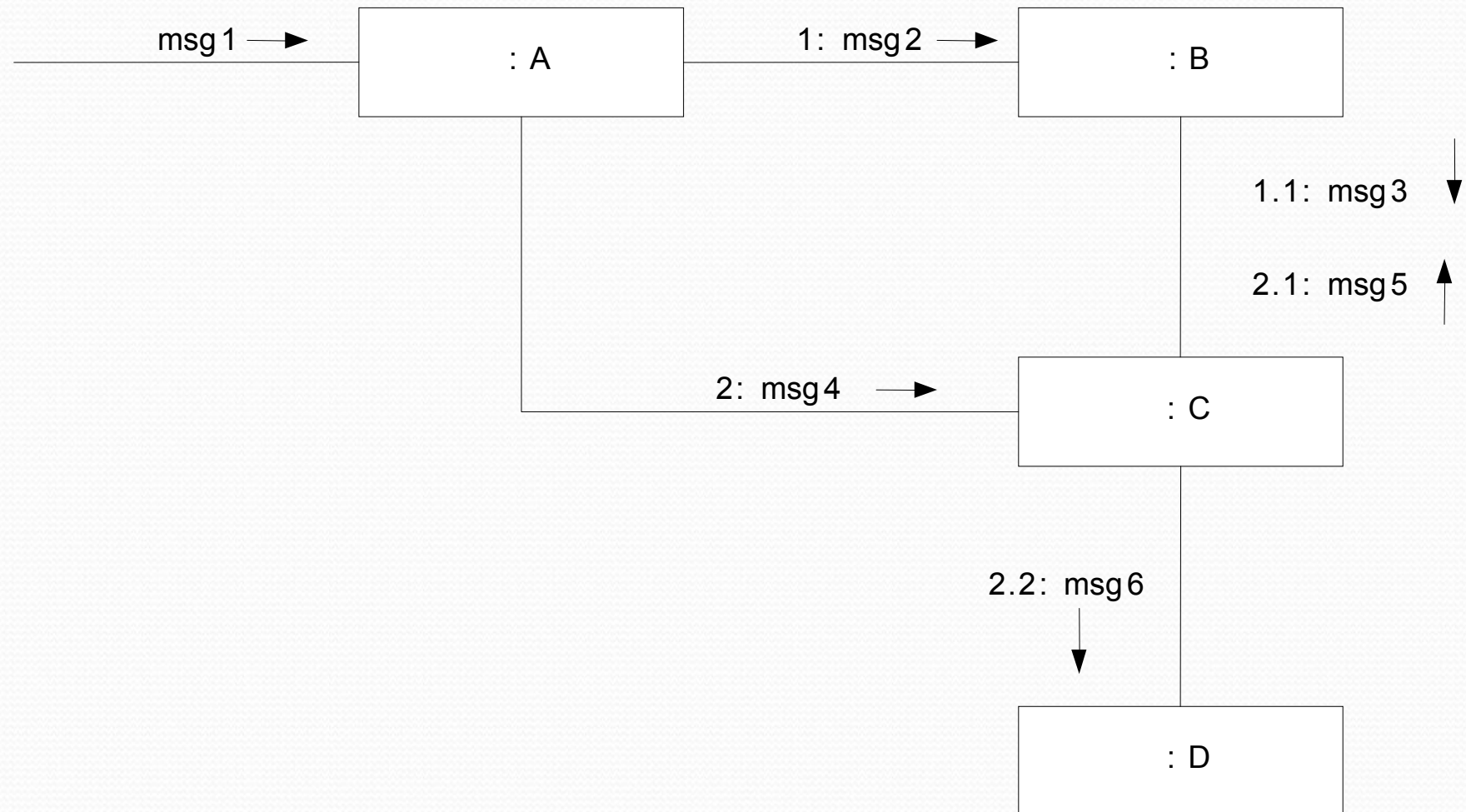
Creazione istanze

**3 modi
alternativi**

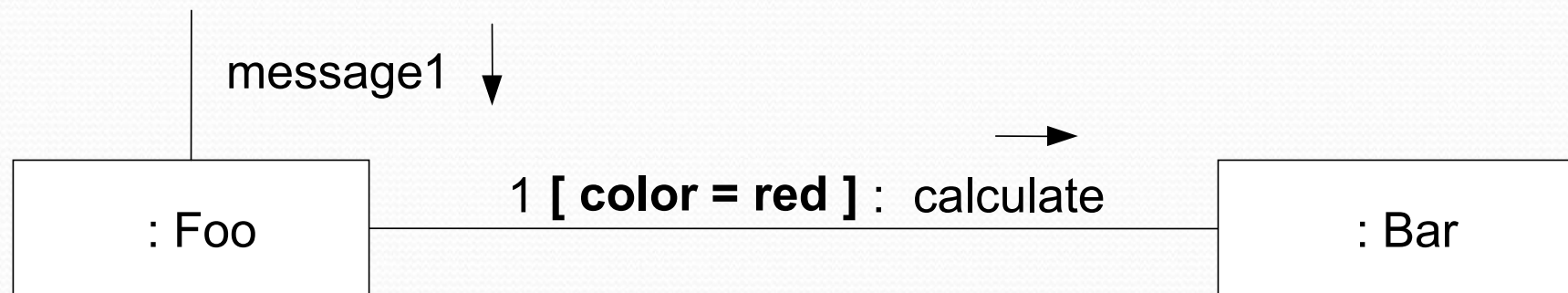
**Chiamata a
costruttore**



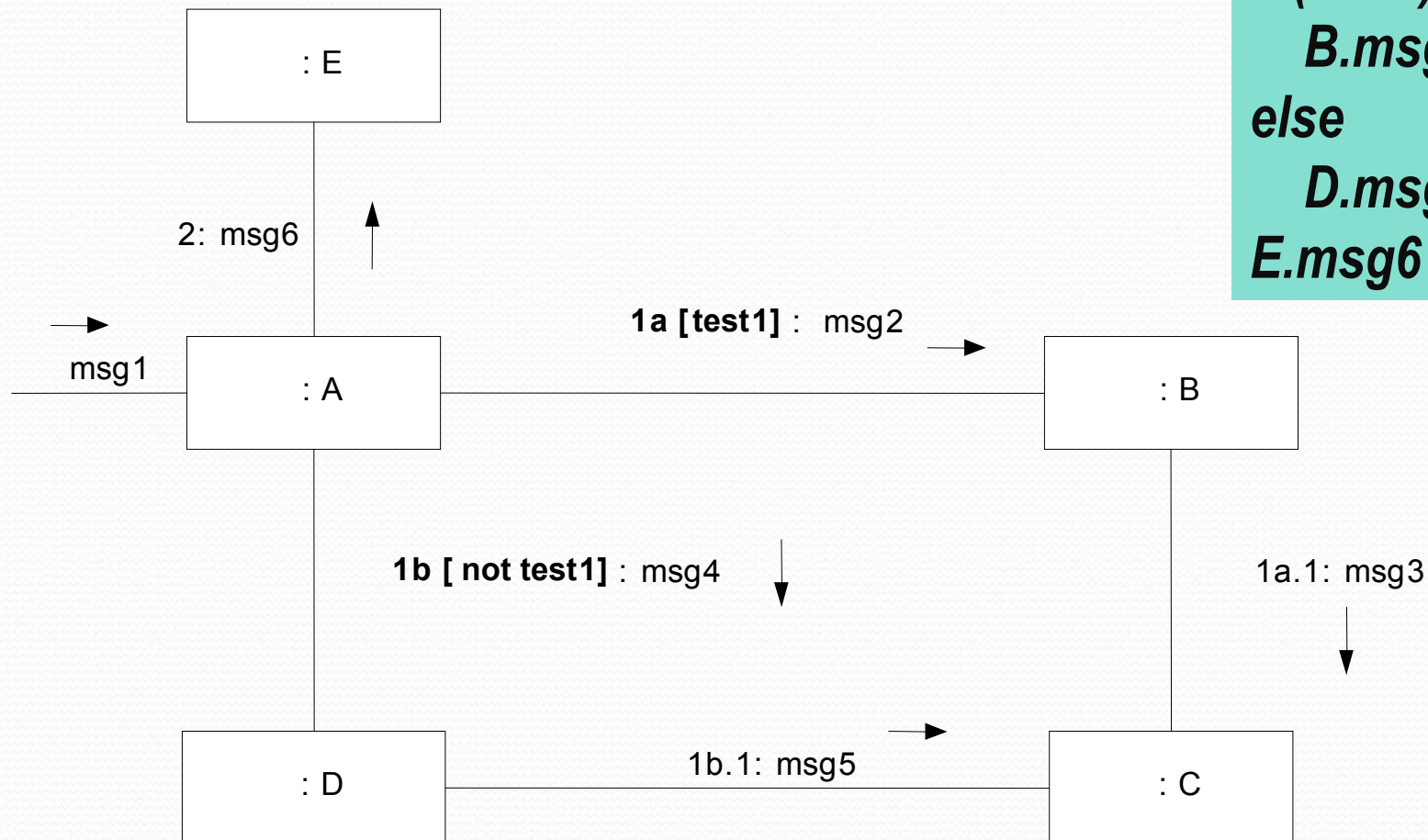
Numerazione messaggi



Messaggio condizionale

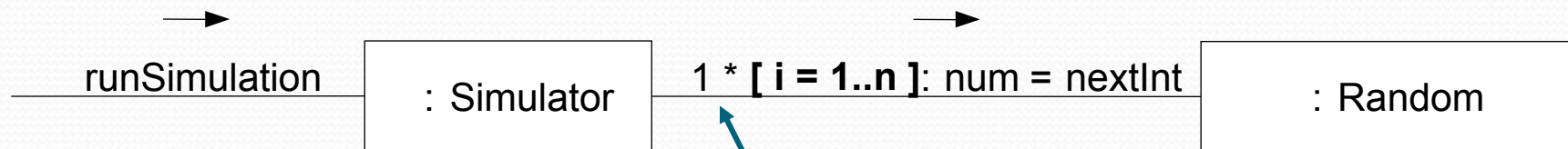


Sequenze alternative



```
// Class A
if (test1)
    B.msg2
else
    D.msg4
E.msg6
```

Iterazioni o loop



L'iterazione si indica con un “*” ed una clausola di iterazione dopo il numero di sequenza del messaggio

Iterazione su una collezione di oggetti

