SOFTWARE DESIGN ESSENTIALS

HOOFDSTUK 4: SYSTEM SEQUENCE DIAGRAMS EN INTERACTION DIAGRAMS

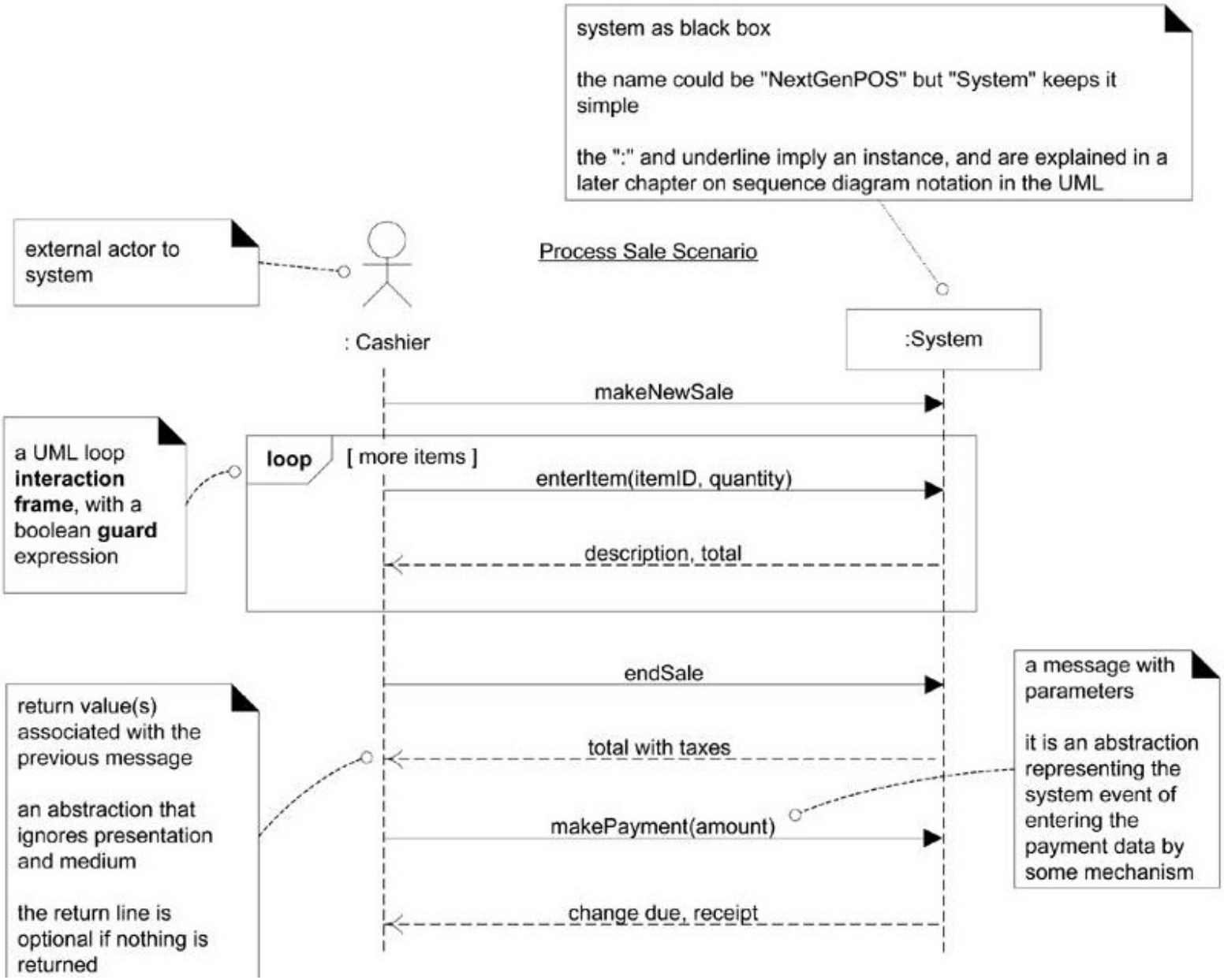
OVERZICHT

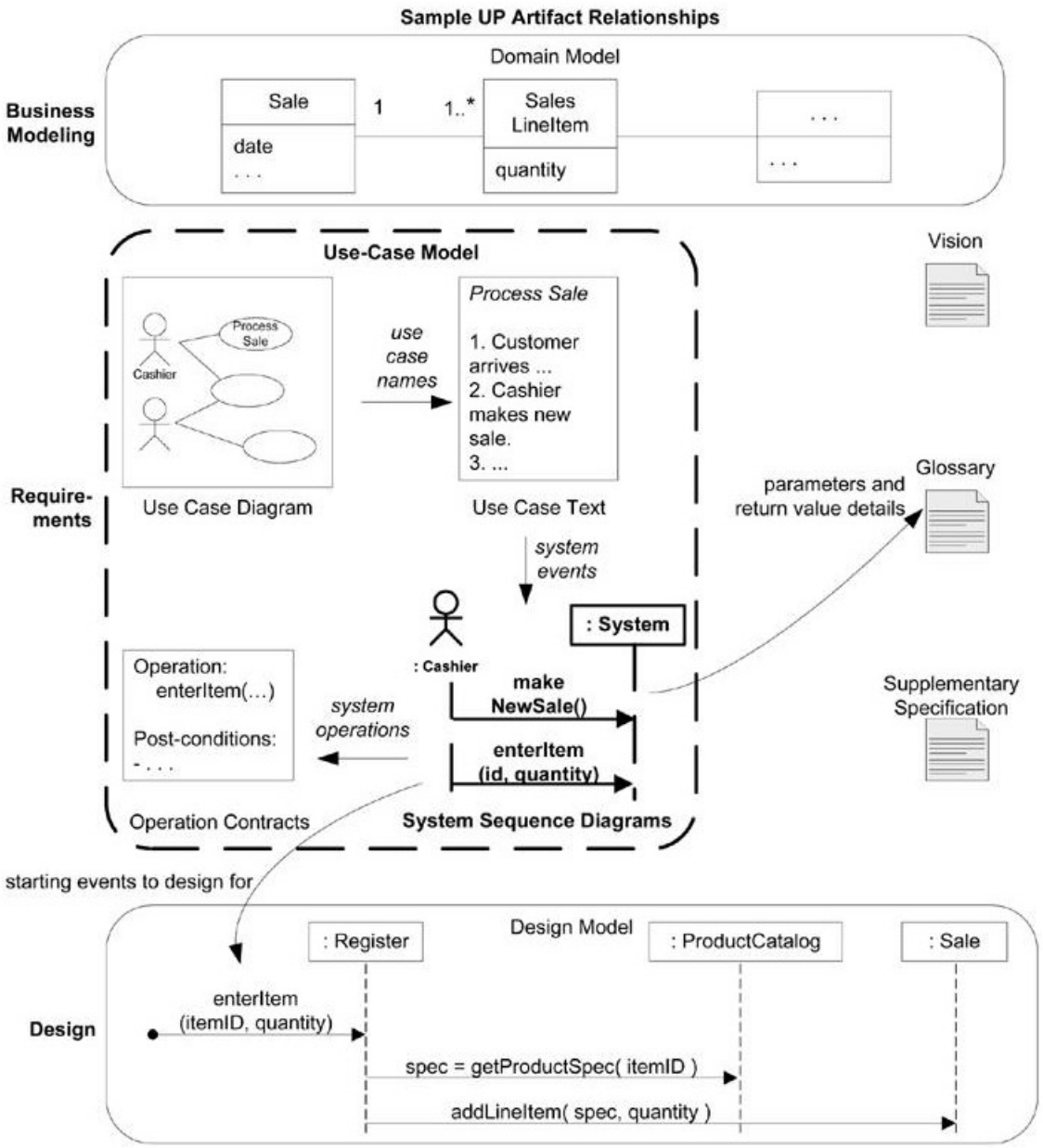
- System Sequence Diagrams (SSDs)
- Interaction Diagrams
  - Sequence Diagram
  - Collaboration Diagram

SYSTEM SEQUENCE DIAGRAMS (SSDS)

SYSTEM SEQUENCE DIAGRAMS

- Wat is een system sequence diagram (SSD)?
  - SSD geeft interactie tussen de gebruiker en het systeem weer op basis van system operations
  - SSD geeft voor een specifieke uitvoering van een use case (dus een use case scenario) de actor, het systeem en de system events die de actor op het systeem genereert


system as black box
the name could be "NextGenPOS" but "System" keeps it simple
the ":" and underline imply an instance, and are explained in a later chapter on sequence diagram notation in the UML
external actor to system
Process Sale Scenario
: Cashier
:System
makeNewSale
a UML loop interaction frame, with a boolean guard expression
loop
[more items]
enterItem(itemID, quantity)
description, total
endSale
a message with parameters
it is an abstraction representing the system event of entering the payment data by some mechanism
return value(s) associated with the previous message
an abstraction that ignores presentation and medium
the return line is optional if nothing is returned
total with taxes
makePayment(amount)
change due, receipt


Sample UP Artifact Relationships
Business Modeling
Domain Model
Sale
date
. . .
1
1..*
Sales LineItem
quantity
. . .
. . .
Use-Case Model
Vision
Process Sale
Cashier
use case names
Process Sale
1. Customer arrives ...
2. Cashier makes new sale.
3. ...
Require-ments
Use Case Diagram
Use Case Text
parameters and return value details
Glossary
system events
: System
: Cashier
Operation: enterItem(...)
Post-conditions: - . . .
system operations
make NewSale()
enterItem (id, quantity)
Supplementary Specification
Operation Contracts
System Sequence Diagrams
starting events to design for
Design
Design Model
: Register
: ProductCatalog
: Sale
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
addLineItem(spec, quantity)

SYSTEM SEQUENCE DIAGRAMS

Wanneer een SSD tekenen?

- Enkel een SSD voor de main success scenario's van elke use case, en voor frequente of complexe alternatieve scenario's
- Systeem blijft nog steeds **black box!**

Waarom een SSD tekenen?

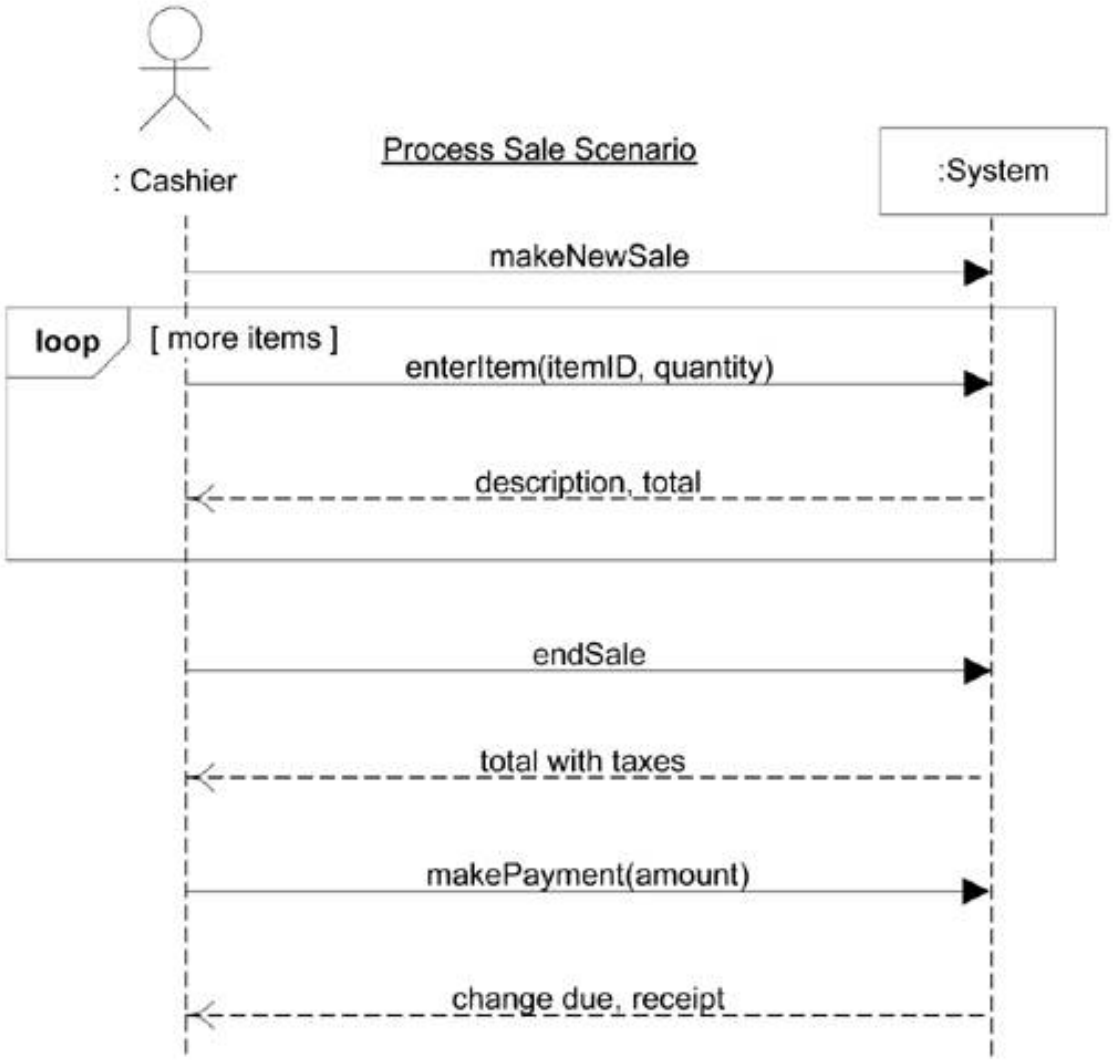
- "Welke events komen binnen in het systeem?"
- Typisch moet de software keyboard, muis, ... input afhandelen
- Een systeem reageert eigenlijk op 3 dingen
  - Externe events van de actoren (muis, keyboard, ...)
  - Timer events
  - Falingen of excepties

SSD tekenen aan de hand van UML

- UML voorziet een "sequence diagram" notatie

SYSTEM SEQUENCE DIAGRAMS

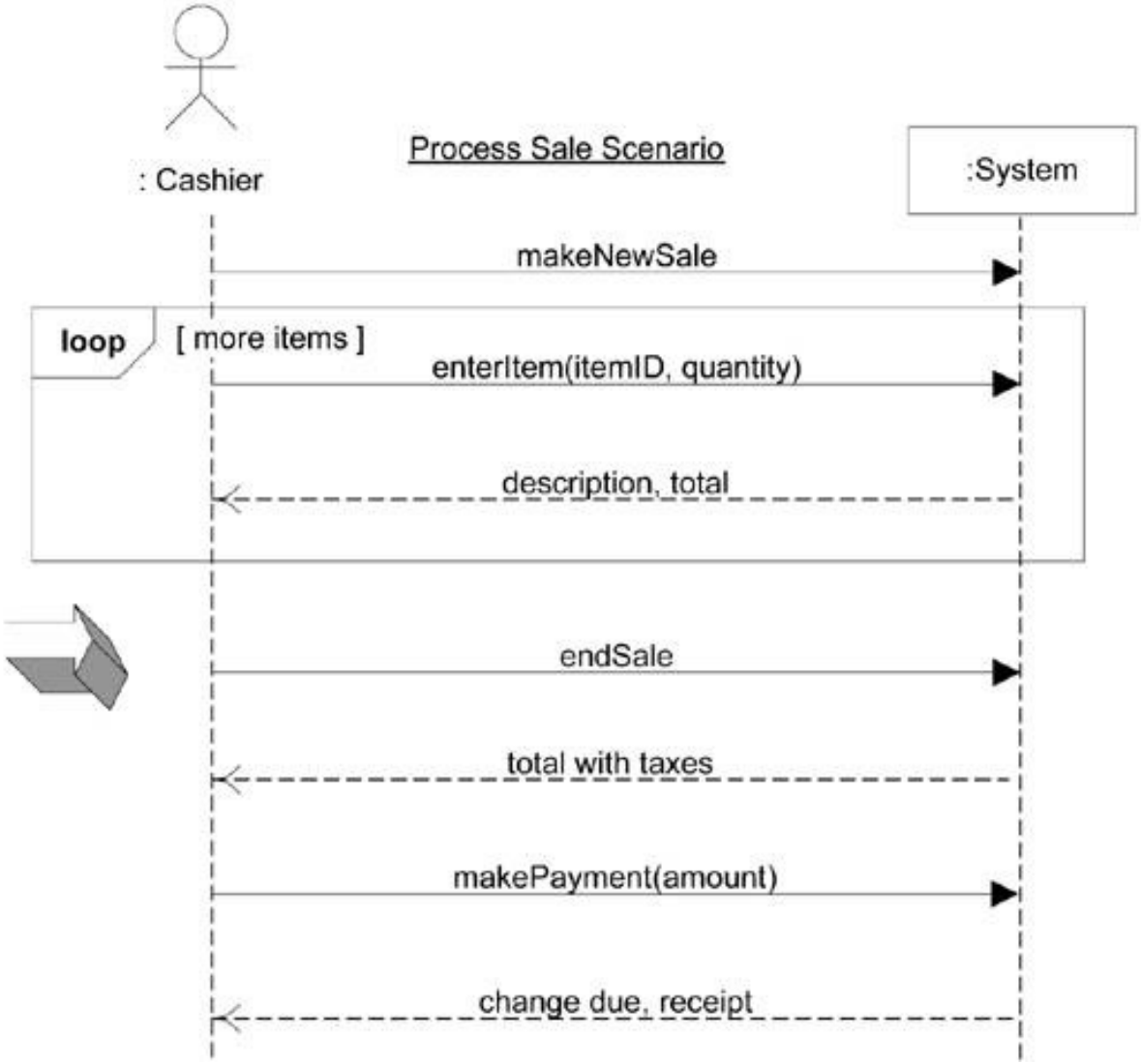
- UML sequence diagram notatie

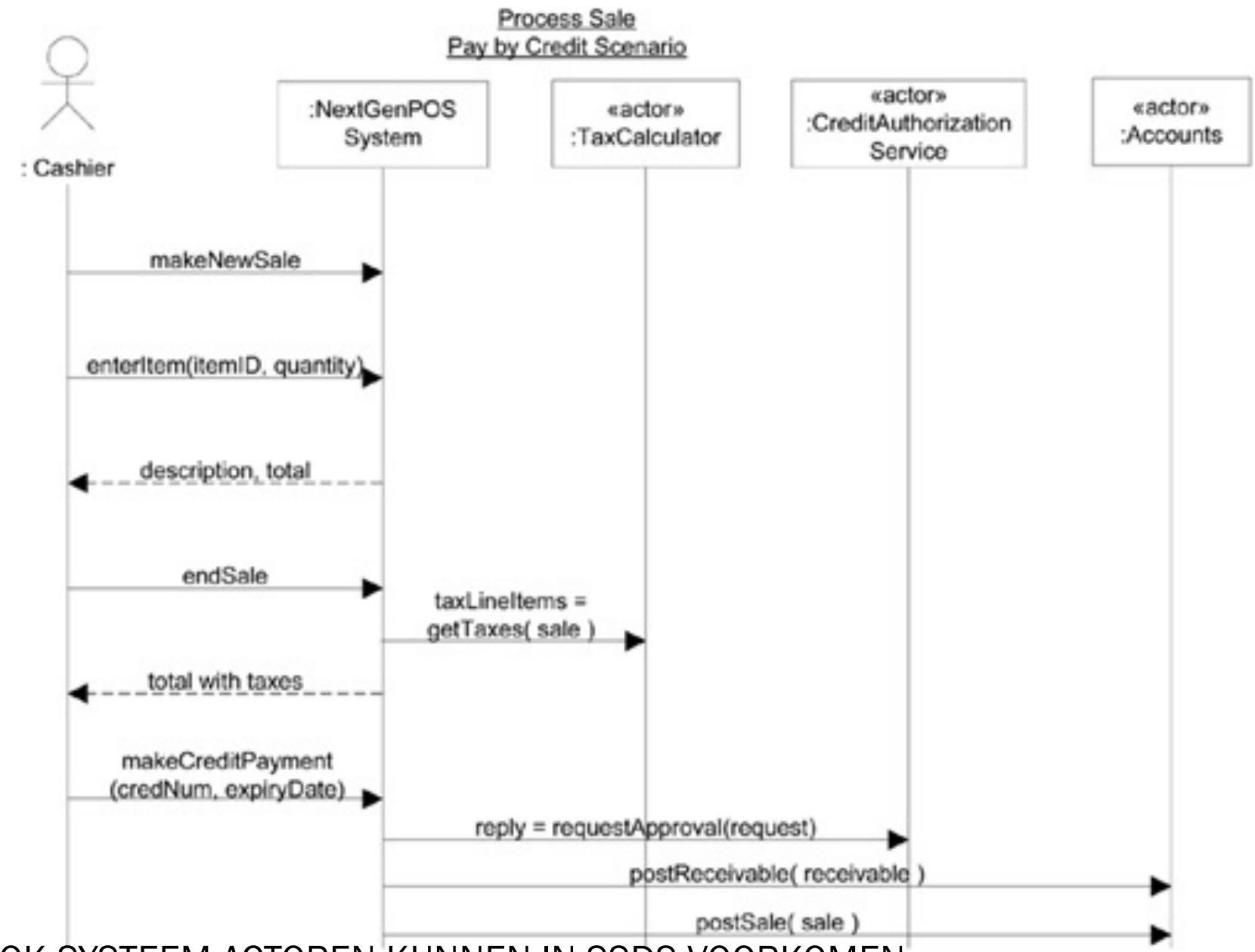


De SSDs zijn afgeleid van de Use Cases

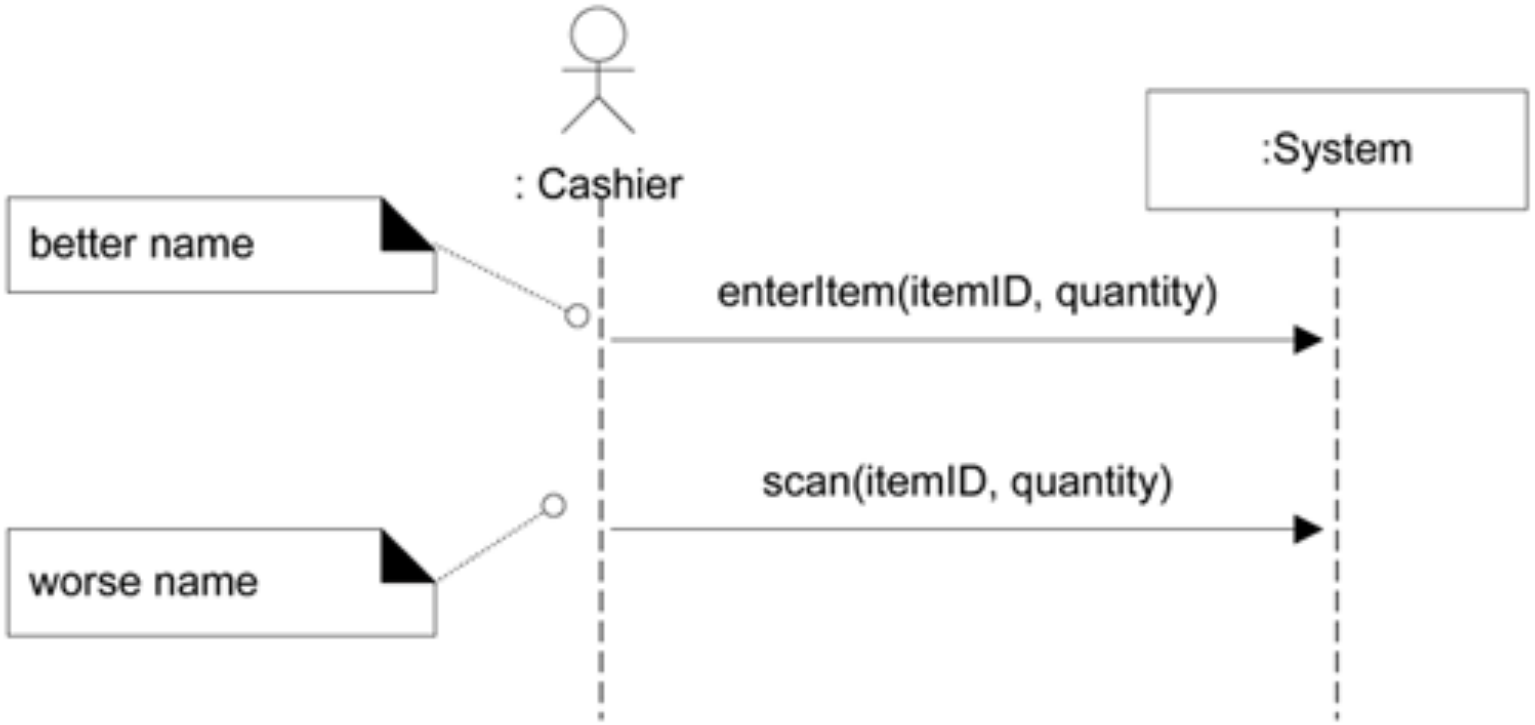
Simple cash-only Process Sale scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
...





OOK SYSTEEM ACTOREN KUNNEN IN SSDS VOORKOMEN



SYSTEM SEQUENCE DIAGRAMS

- UML sequence diagram notatie

INTERACTION DIAGRAMS

- **INTERACTION DIAGRAMS**

SEQUENCE DIAGRAM

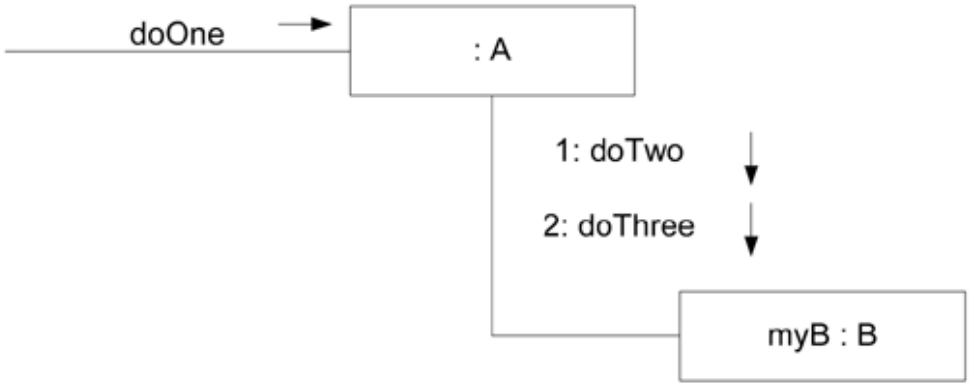
COLLABORATION DIAGRAM

INTERACTION DIAGRAM

Wat is een interaction diagram?

- Doel: Object interacties weergeven
- Algemene term voor 2 soorten diagramma's
  - Sequence diagram
  - Communication diagram (=collaboration diagram)

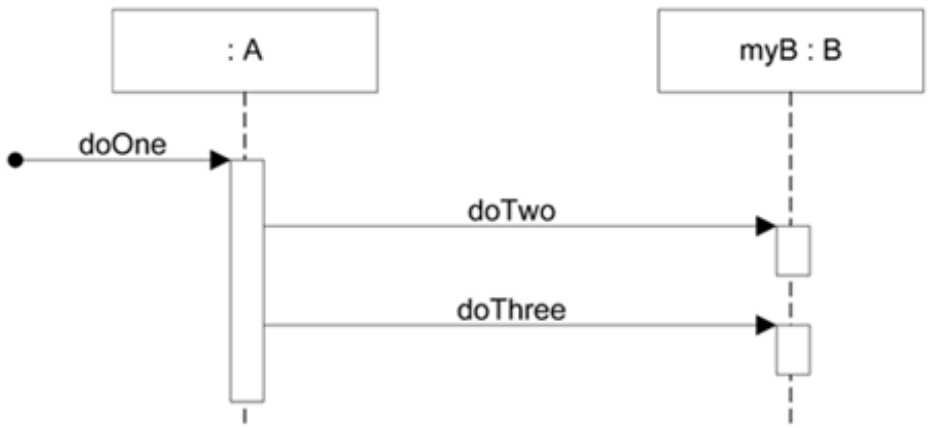
INTERACTION DIAGRAMS




```
public class A
{
private B myB = new B();

public void doOne()
{
   myB.doTwo();
   myB.doThree();
}
// …
}
```

Communication/collaboration diagram

INTERACTION DIAGRAMS




```
public class A
{
private B myB = new B();

public void doOne()
{
   myB.doTwo();
   myB.doThree();
}
// …
}
```

Sequence diagram

INTERACTION DIAGRAMS

Wat is een interaction diagram?

- **Doel:** Object interacties weergeven
- Beide soorten hebben voor- en nadelen
  - Sequence diagram
    - Eenvoudig te lezen flow
    - Betere UML specificatie (dus betere tool support)
  - Communication/collaboration diagram
    - Gemakkelijker om te schetsen
    - Mogelijkheid tot “verticale expansie” bij tekenen

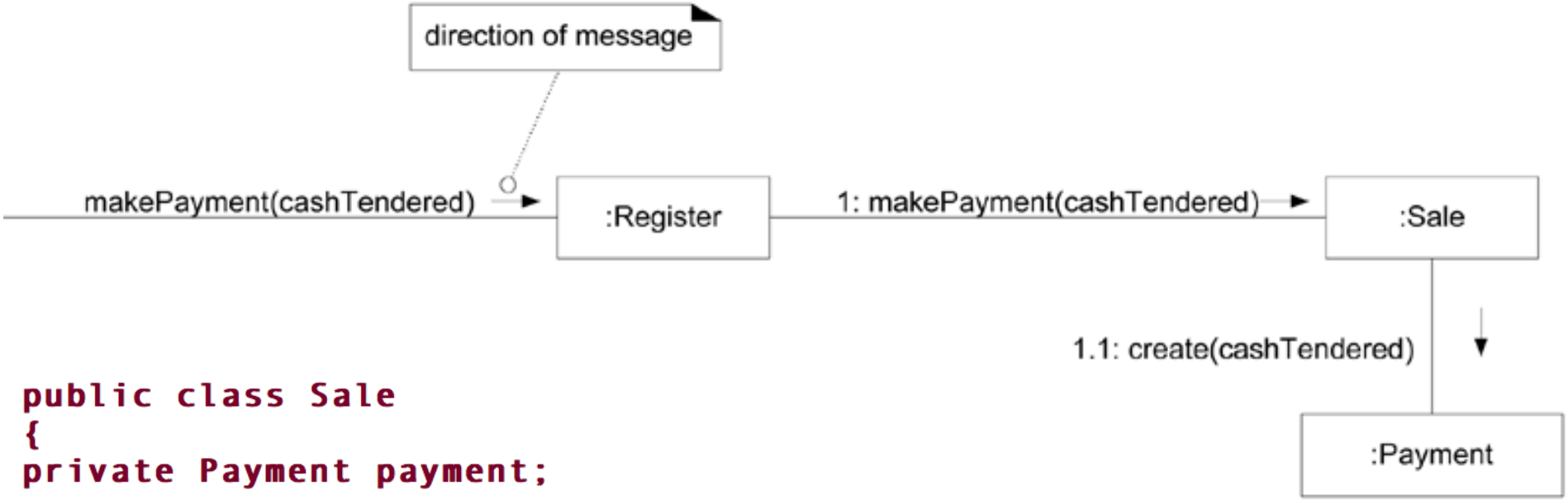
INTERACTION DIAGRAMS

- Tip: spendeer voldoende tijd aan het opstellen van interaction diagrams!
  - Beginnende programmeurs spenderen meestal te veel tijd aan class diagram

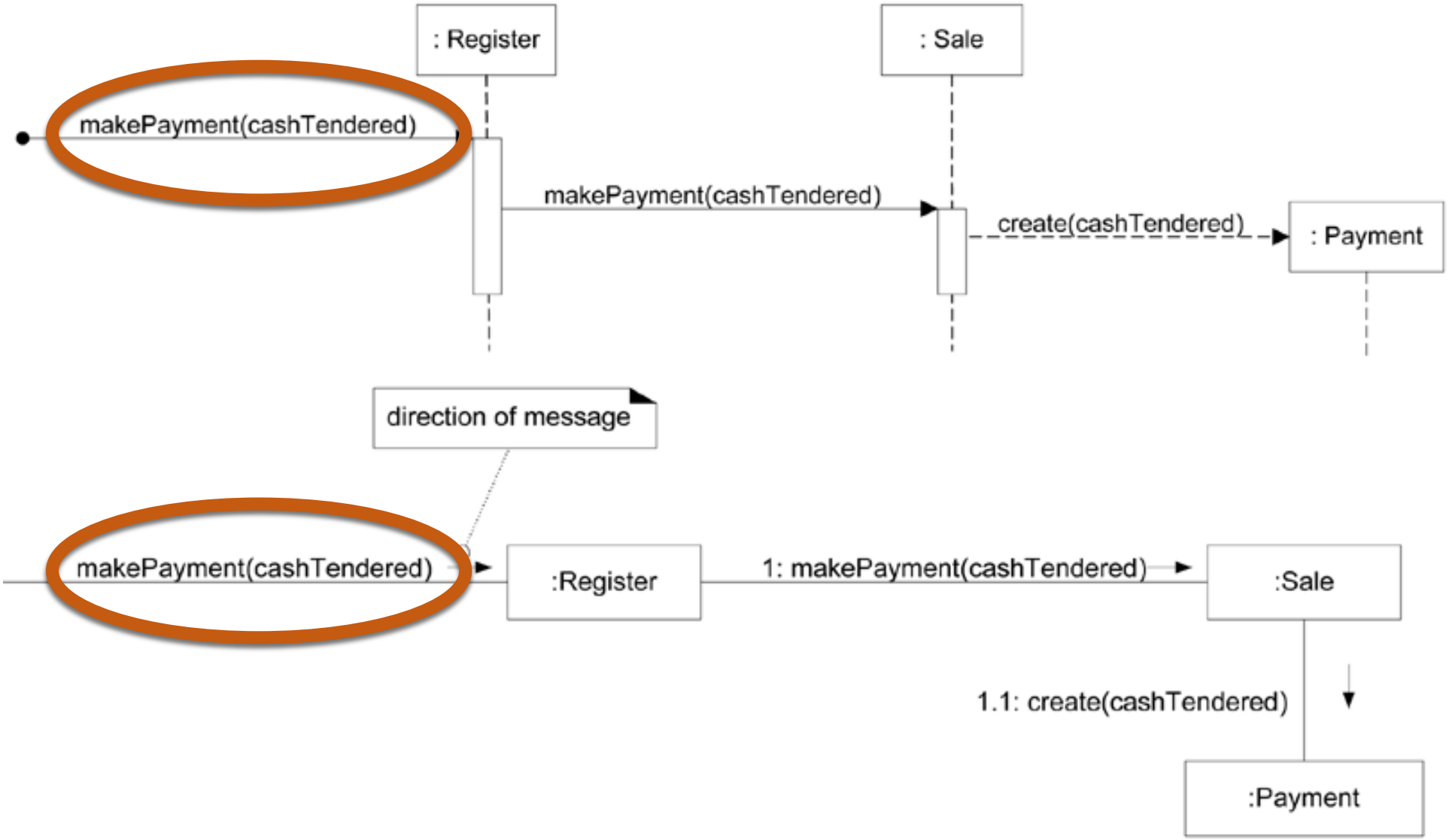
INTERACTION DIAGRAMS


```
public class Sale
{
private Payment payment;

public void makePayment( Money cashTendered )
{
   payment = new Payment( cashTendered );
   //…
}
// …
}
```

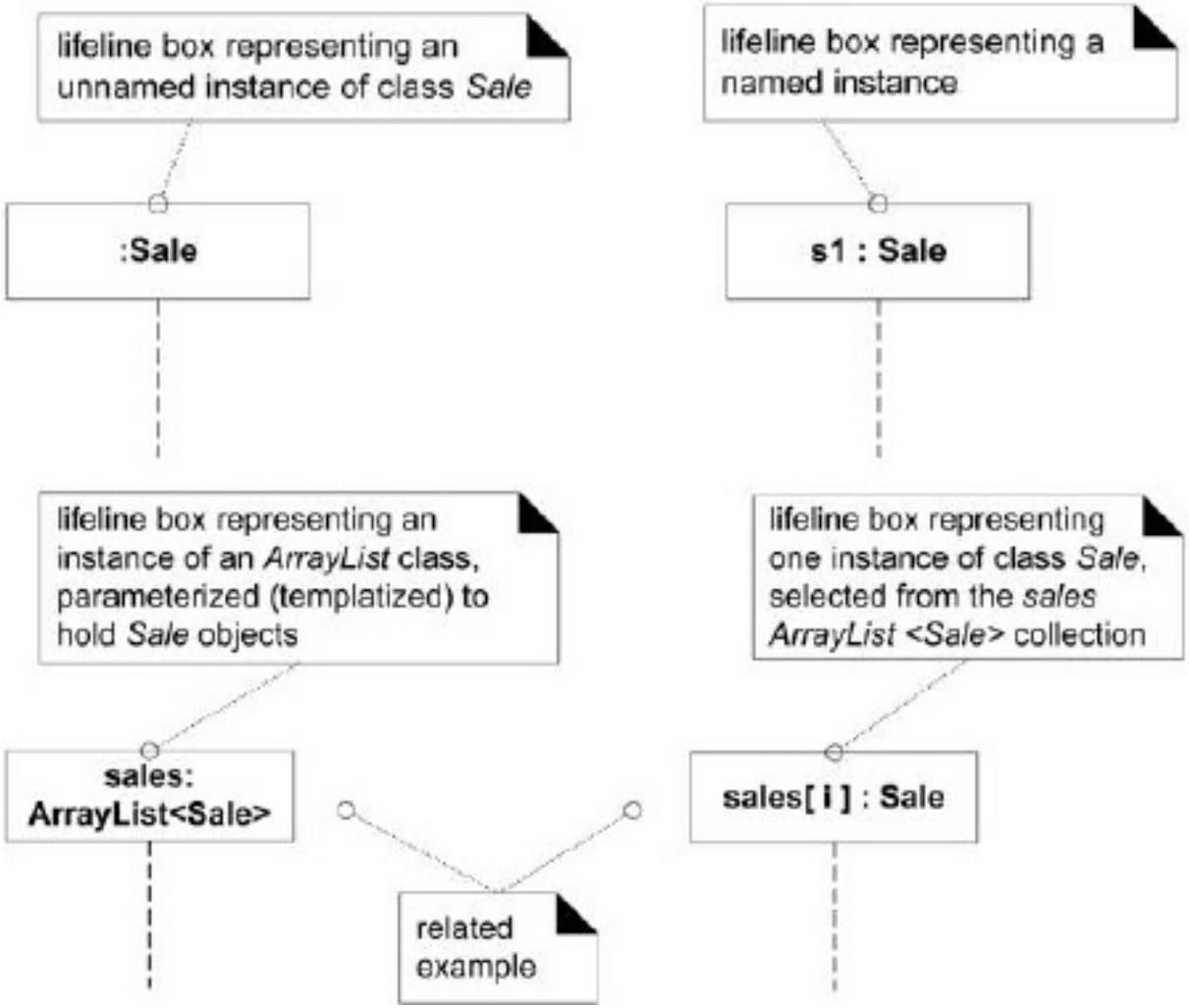
INTERACTION DIAGRAMS




```
public class Sale
{
private Payment payment;

public void makePayment( Money cashTendered )
{
   payment = new Payment( cashTendered );
   //…
}
// …
}
```


: Register
: Sale
makePayment(cashTendered)
makePayment(cashTendered)
create(cashTendered)
: Payment
direction of message
makePayment(cashTendered)
:Register
1: makePayment(cashTendered)
:Sale
1.1: create(cashTendered)
:Payment

INTERACTION DIAGRAMS



INTERACTION DIAGRAMS

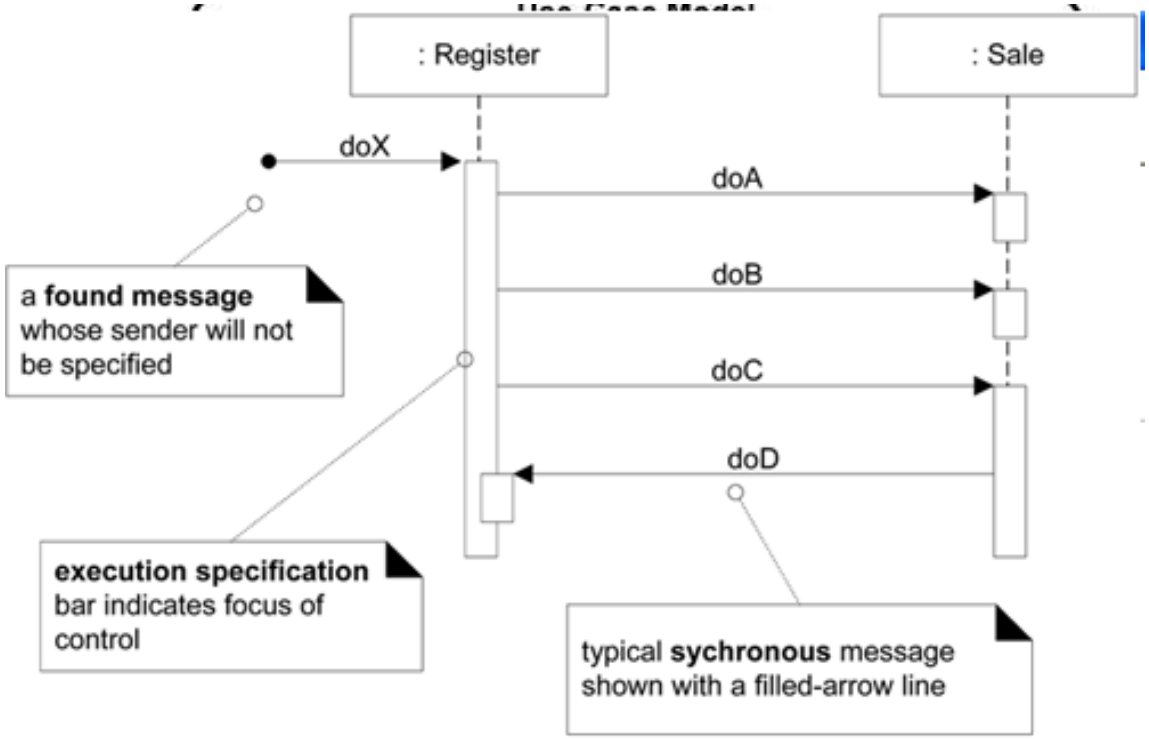
Algemene message syntax

- UML standaard:
  - Return = message (parameter:parameterType):returnType
- Voorbeelden
  - initialize(code)
  - Initialize (haakjes mogen weg indien geen param)
  - d = getProductBeschrijving(id)
  - d = getProductBeschrijving(id:ItemID)
  - d = getProductBeschrijving(id:ItemID):ProductDescription

INTERACTION DIAGRAMS

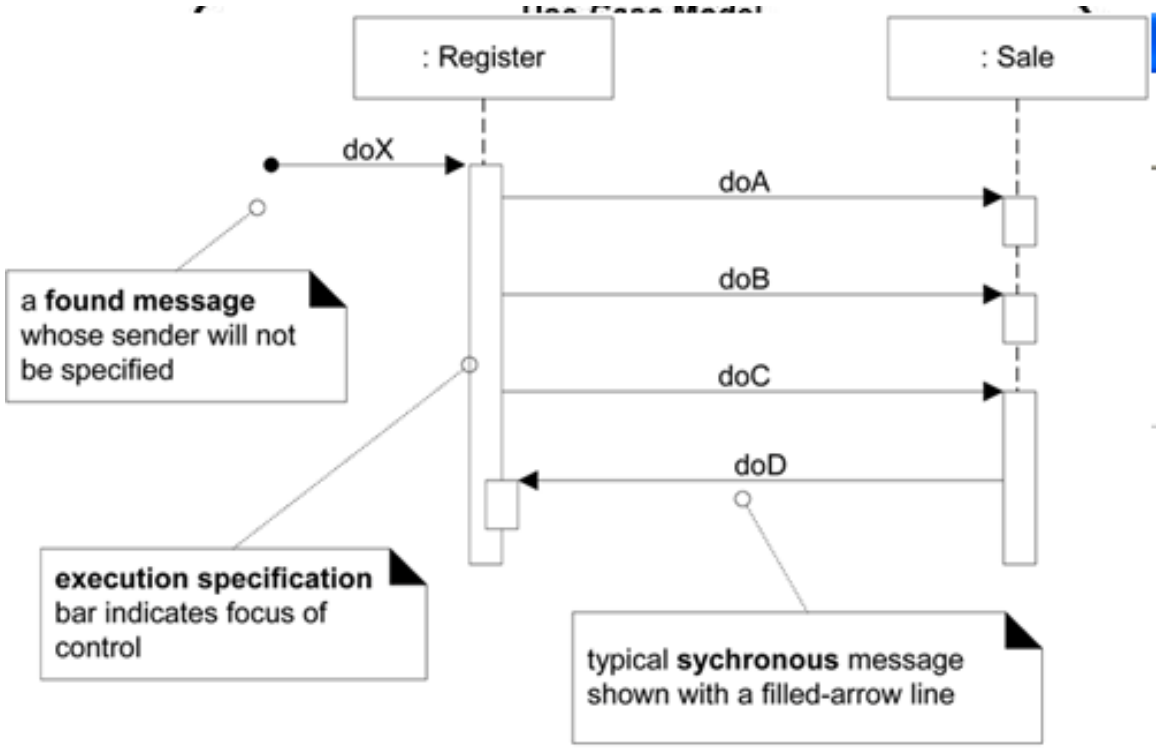
Sequence Diagram

INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

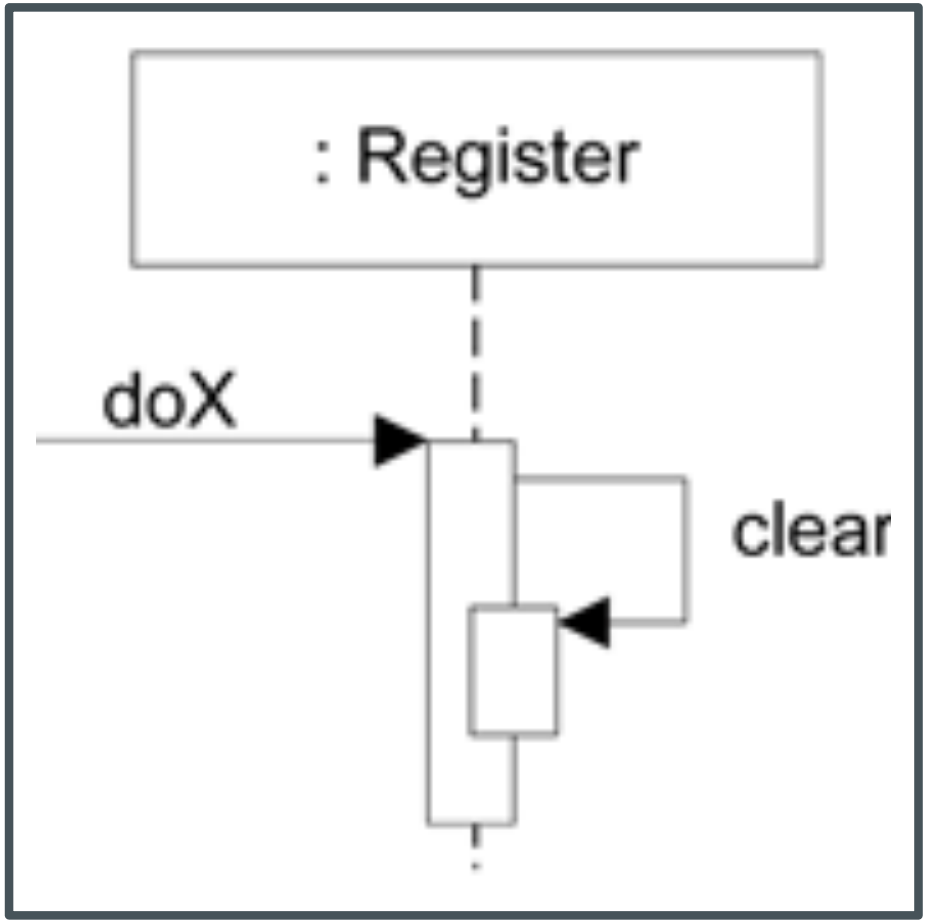


INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

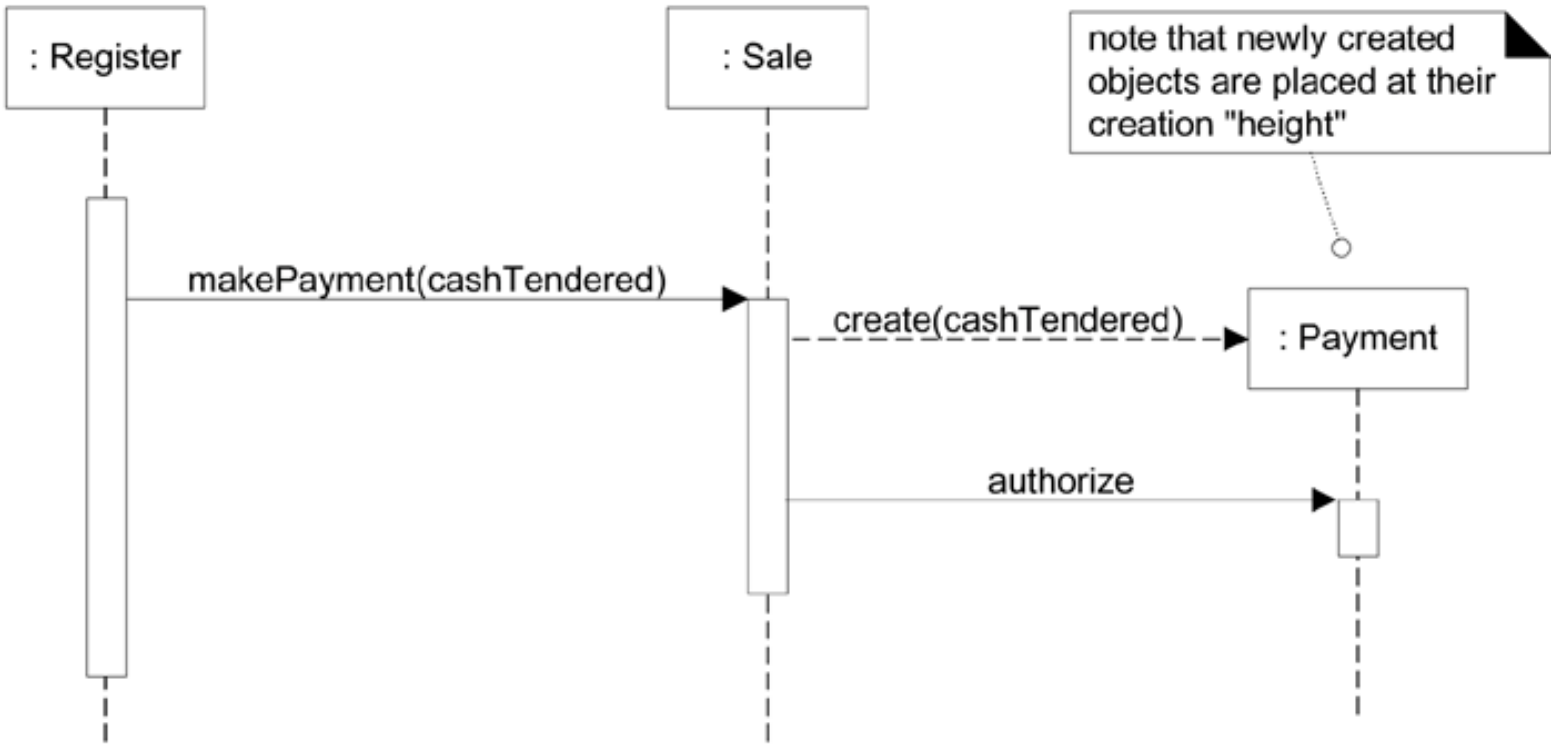
- “Executation specification bar”
  - In UML Case tools meestal getekend
  - In snelle schetsen meestal niet getekend



INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

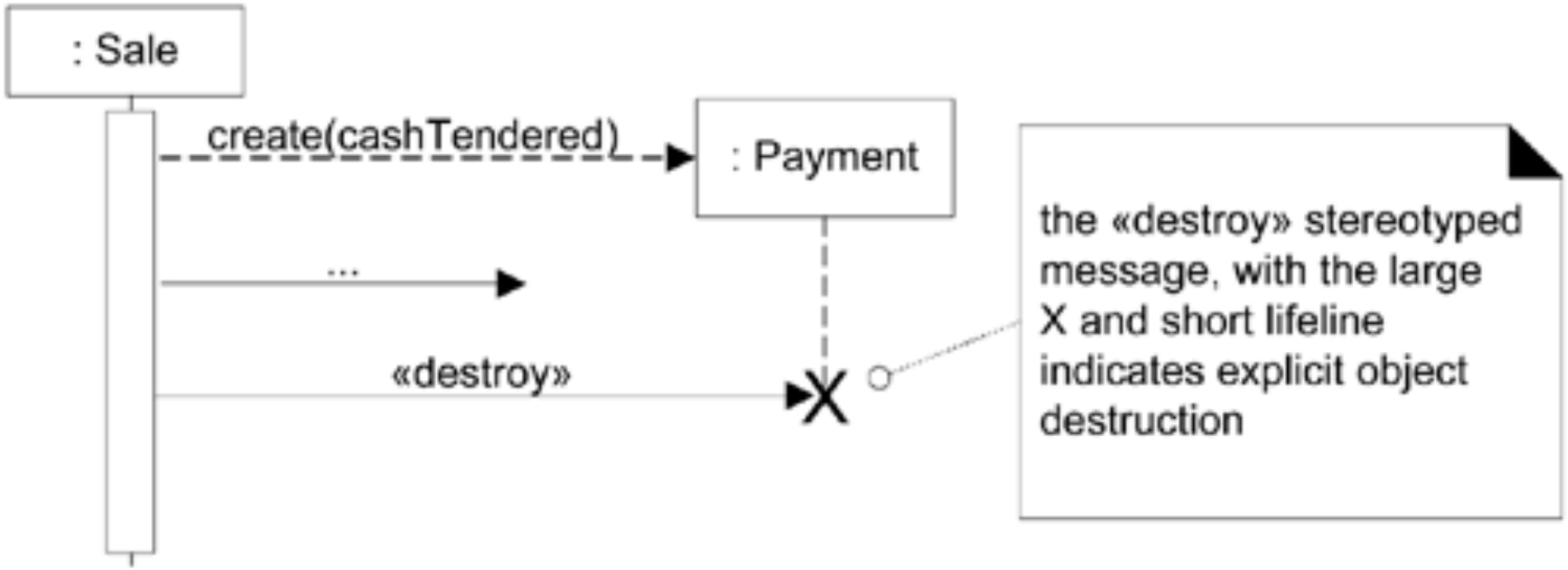


Bericht naar eigen object



INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Object creatie (“new”)

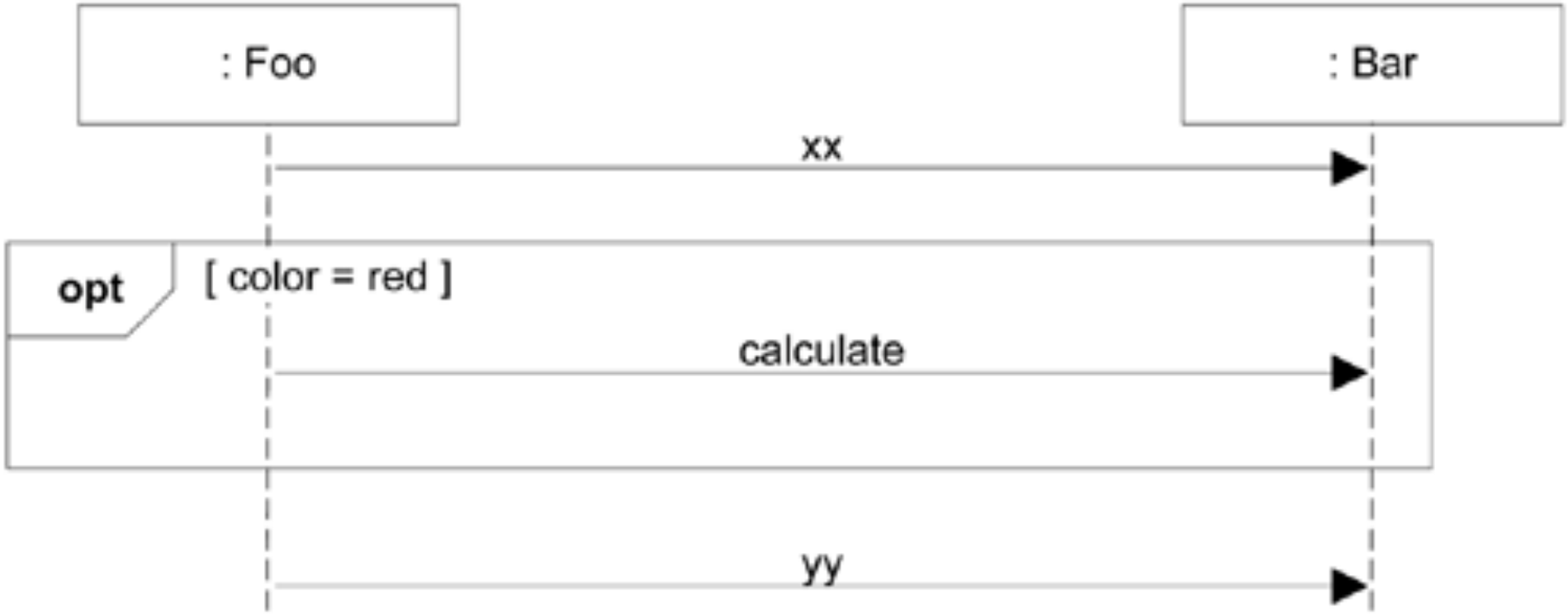


INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Object destructie
  - Enkel in talen zonder garbage collector

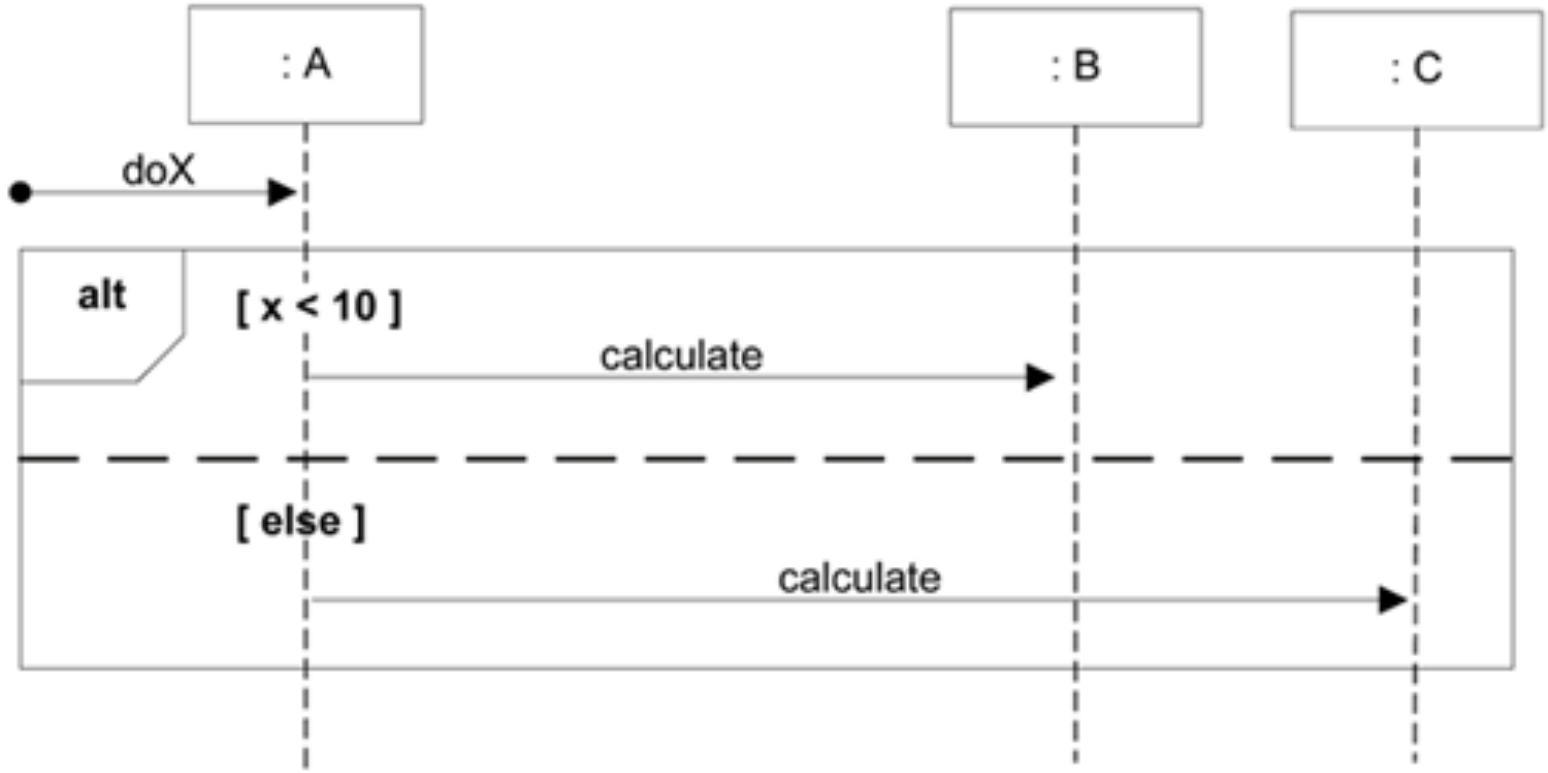
INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Loops



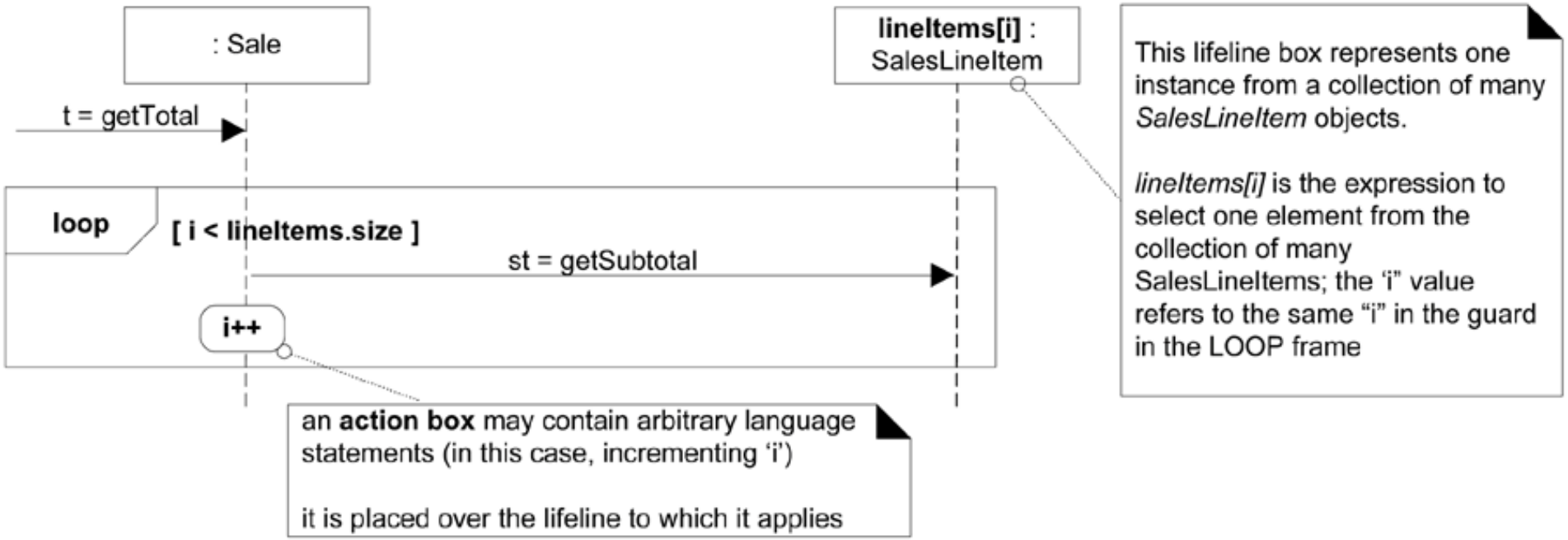
INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Condities



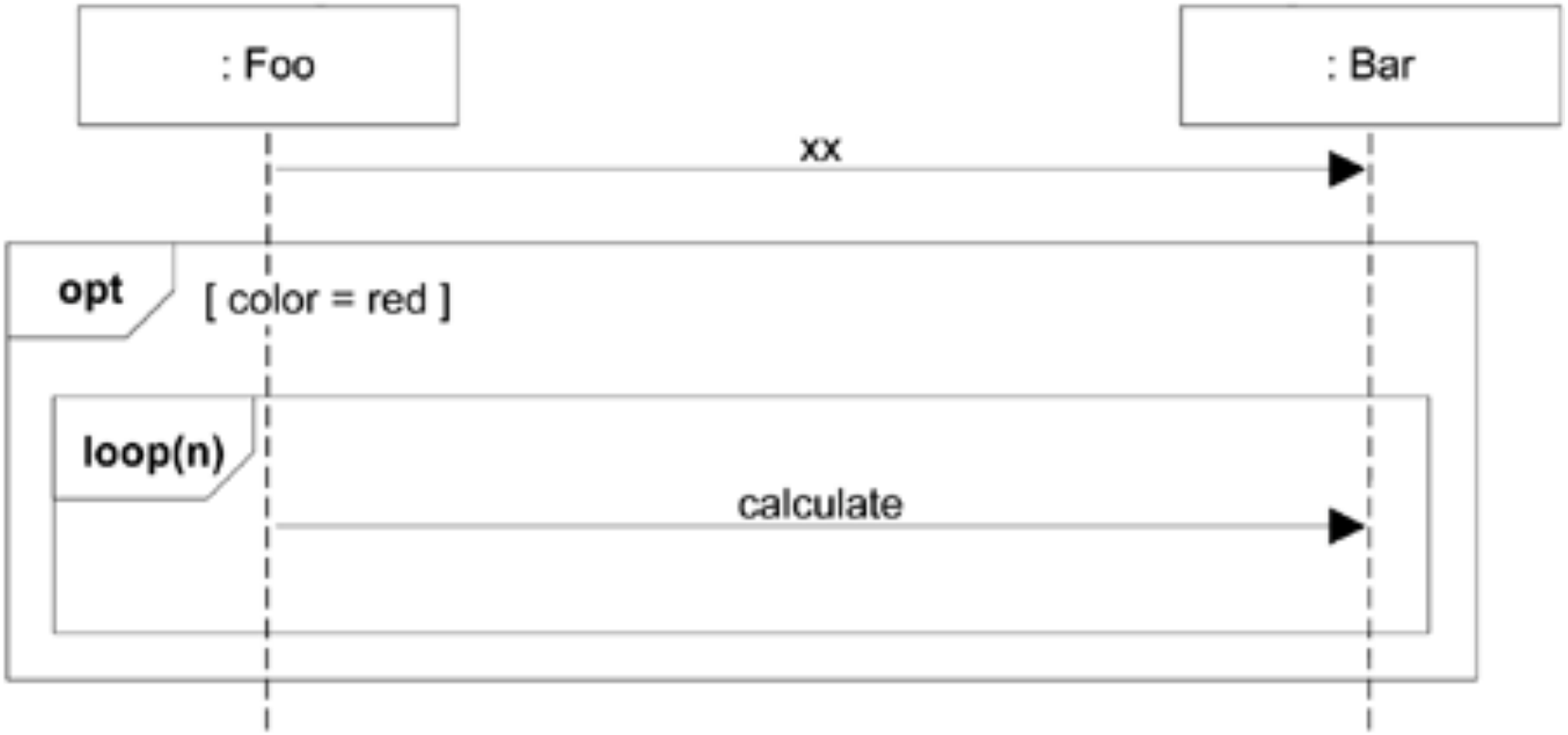
INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Condities



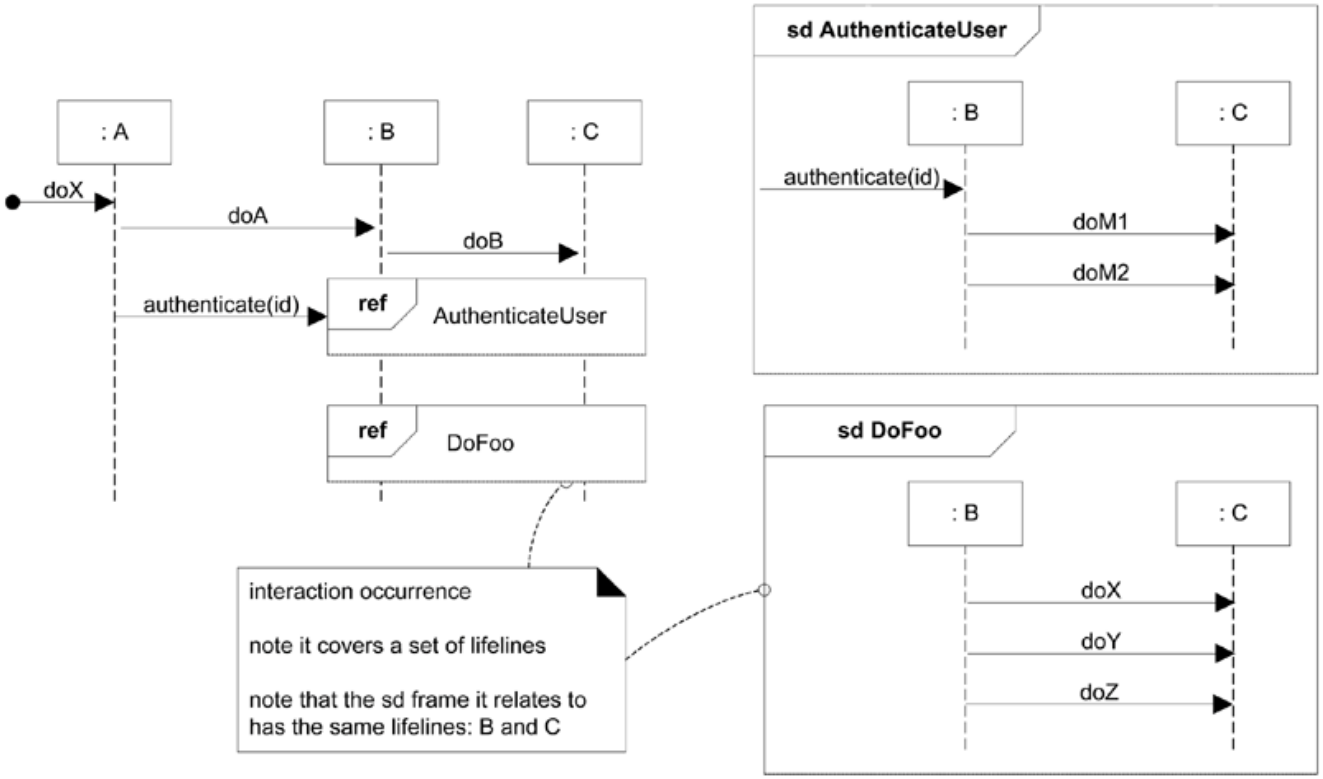
INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Iteratie over collecties



INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Nesten van frames

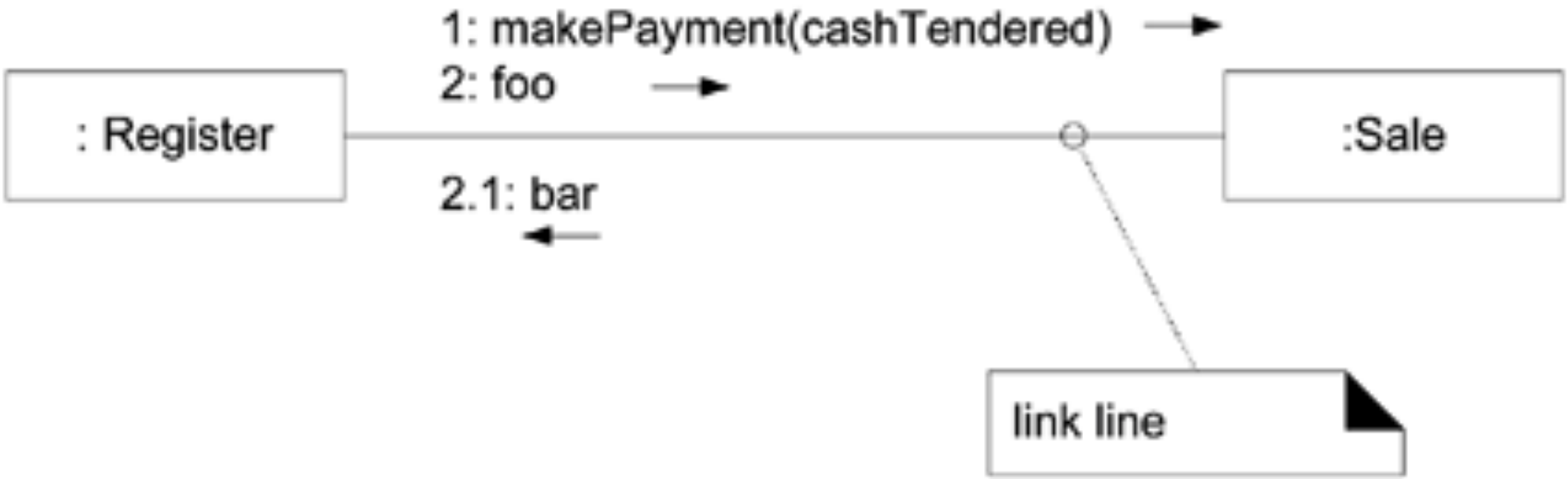


INTERACTION DIAGRAMS - SEQUENCE DIAGRAM NOTATIE

- Referenties

INTERACTION DIAGRAMS

Collaboration Diagram

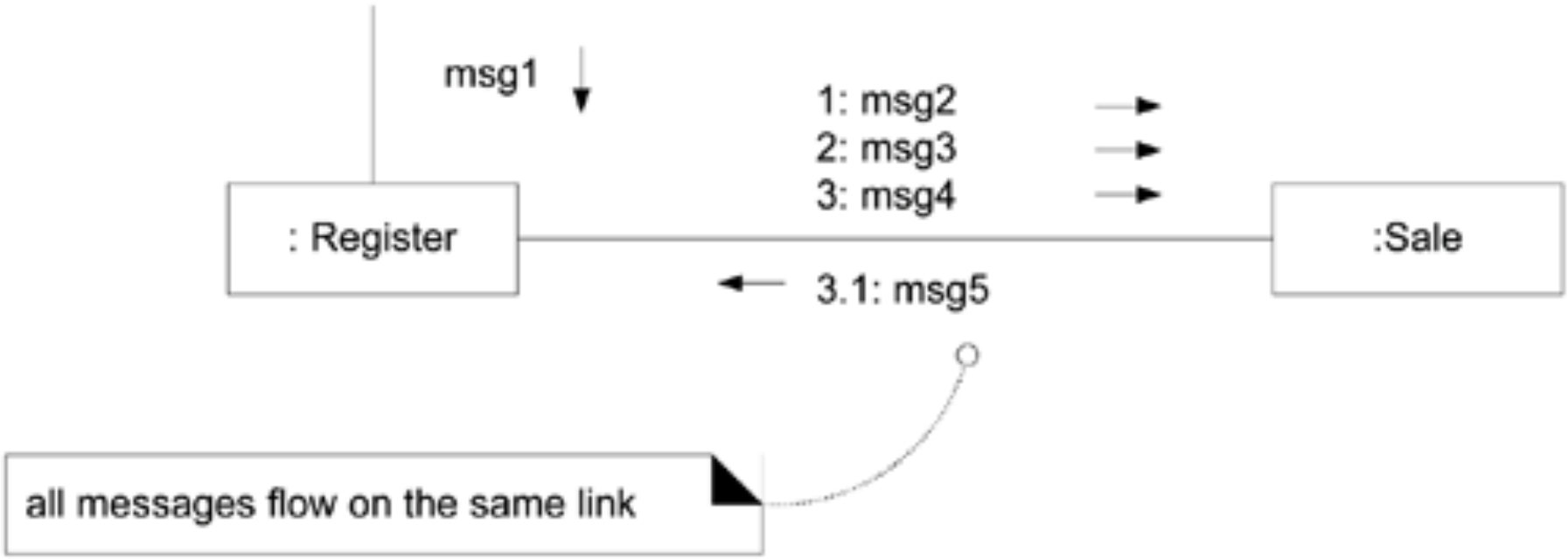


INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

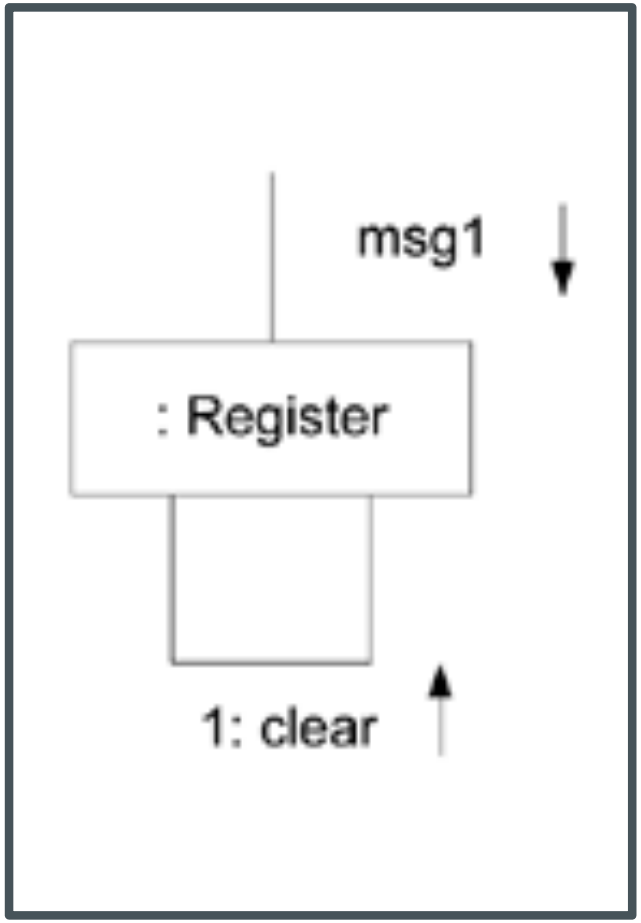
- Links tussen klassen

INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

Eerste bericht krijgt geen nummer

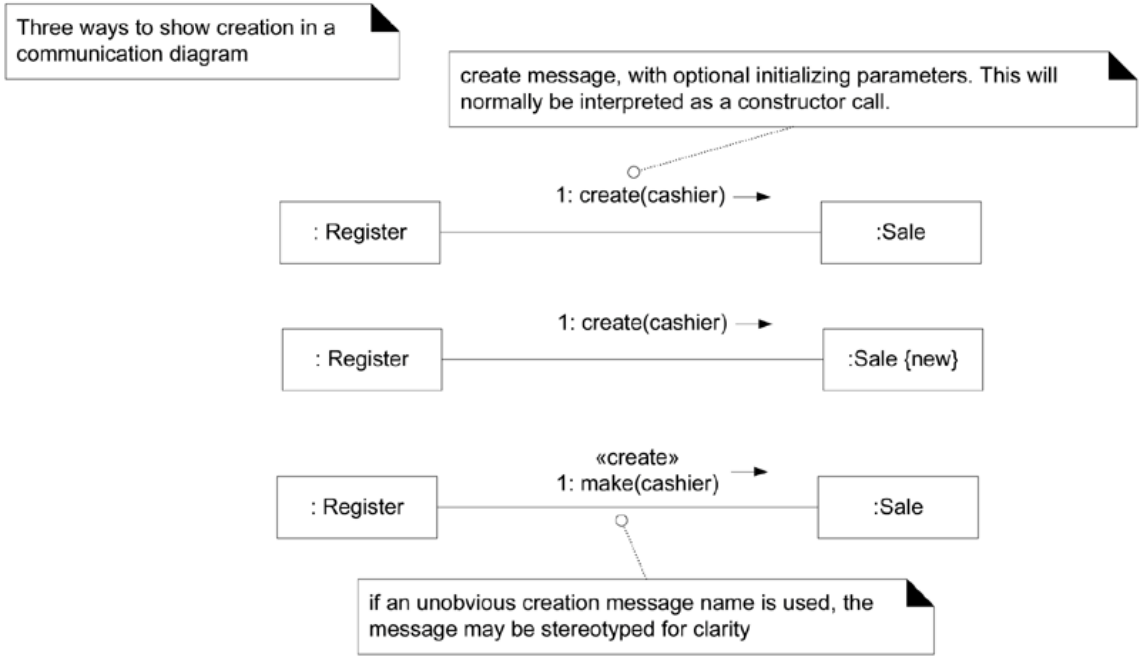


INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

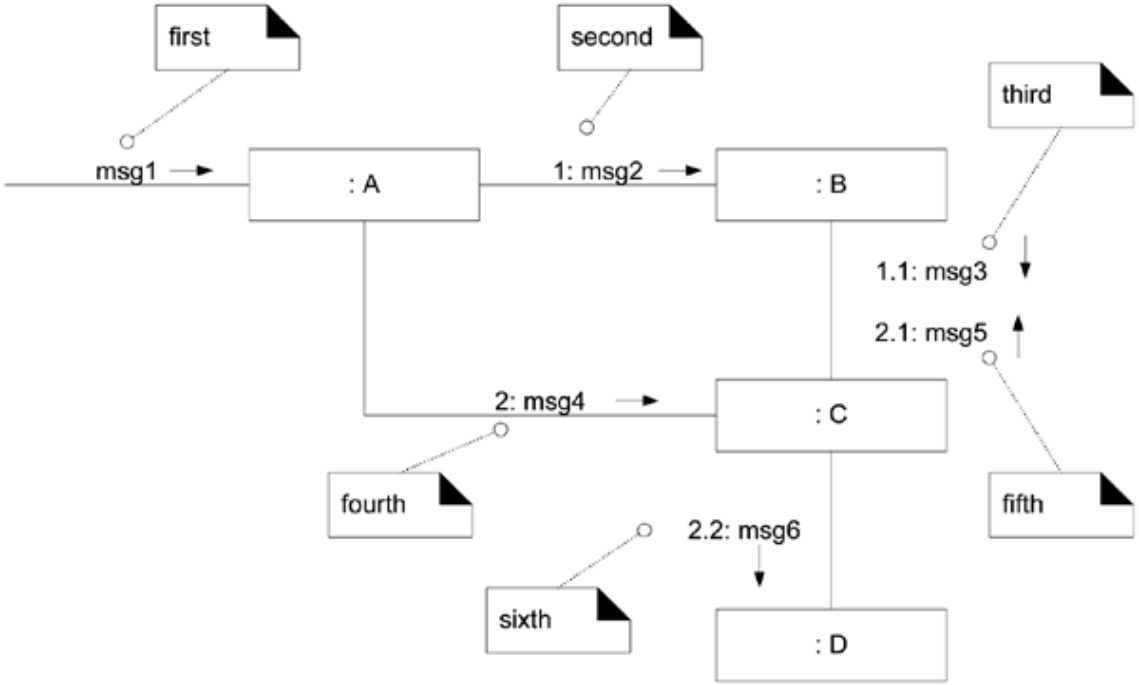


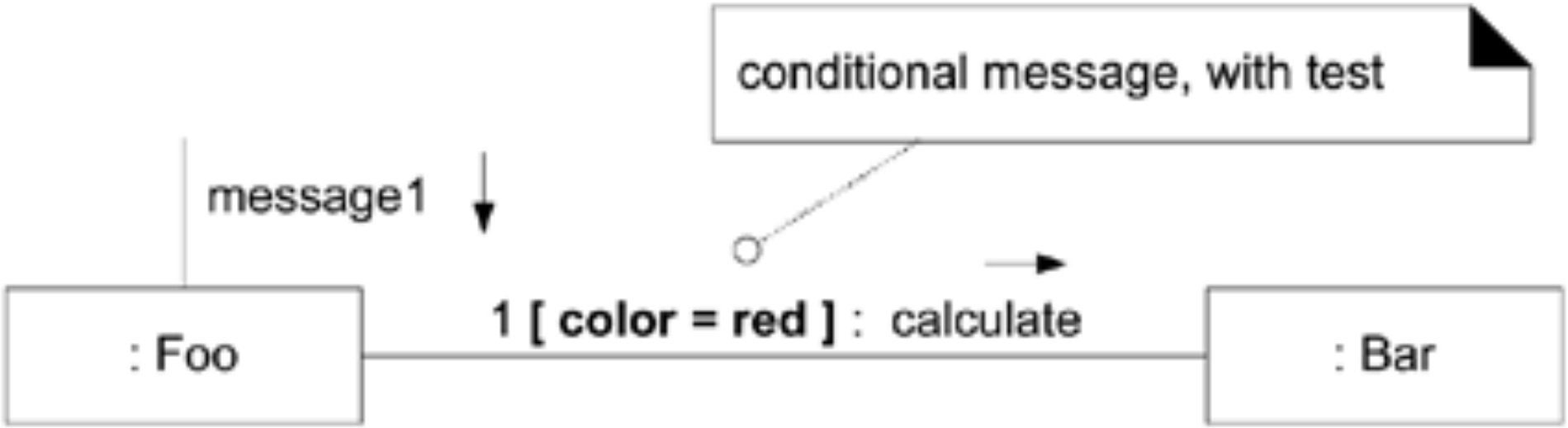
- Verwijzing naar zichzelf

INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE



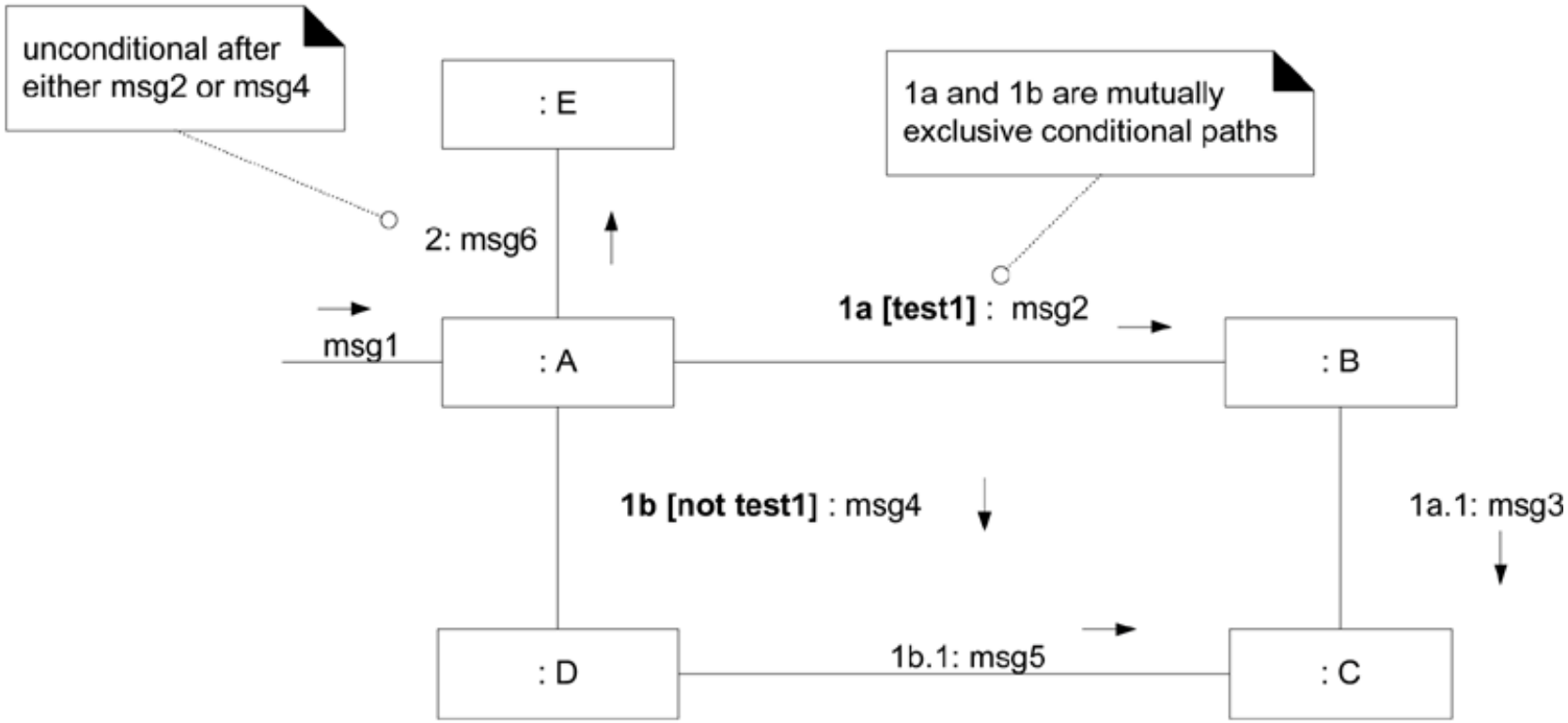
INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE





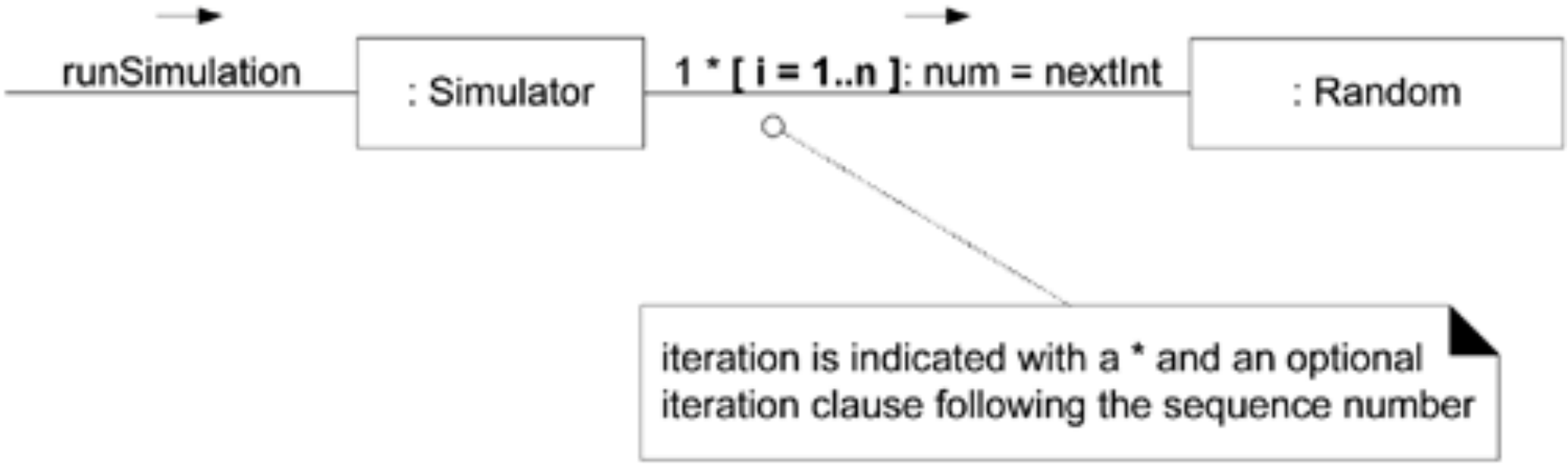
INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

- Condities



INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

- Condities



INTERACTION DIAGRAMS - COMMUNICATION DIAGRAM NOTATIE

- Iteraties