

# 07 . Diagrammi di Sequenza di Sistema (SSD)

Sviluppo di Applicazioni Software

---

Matteo Baldoni

a.a. 2023/24

Università degli Studi di Torino - Dipartimento di Informatica

## Attenzione!



©2024 Copyright for this slides by Matteo Baldoni. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

<https://creativecommons.org/licenses/by/4.0/>

## Si noti che

questi lucidi sono basati sul libro di testo del corso “C. Larman, *Applicare UML e i Pattern*, Pearson, 2016” e parzialmente sul materiale fornito da Viviana Bono, Claudia Picardi e Gianluca Torta dell’Università degli Studi di Torino.

# Table of contents

1. Disciplina dei requisiti: Diagrammi di Sequenza di Sistema
2. Un po' di notazione

**Disciplina dei requisiti:  
Diagrammi di Sequenza di  
Sistema**

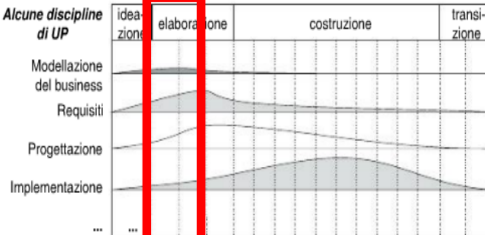
---

# UP maps

**Tabella 2.1** Scenario di Sviluppo di esempio (i – inizio; r – raffinamento).

Disciplina	Pratica	Elaborato Iterazione →	Ideazione I1	Elaboraz. Et..En	Costr. C1..Cn	Transiz. T1..T2
Modellazione	modellazione agile	Modello		i		
Requisiti	workshop requisiti esercizio sulla visione	Modello dei Casi d'Uso	i	r		
		Specifica Supplementare	i	r		
		Glossario	i	r		
Progettazione	modellazione agile sviluppo guidato dai test	Modello di Progetto Documento dell'Architettura Software		i	r	
		Modello dei Dati		i	r	
Implementazione	sviluppo guidato dai test programmazione a coppie integrazione continua standard di codifica	...				
Gestione del progetto	gestione del progetto agile riunioni Scrum giornaliere	...				
...						

*Alcune discipline  
di UP*



L'impegno relativo  
nelle discipline cambia  
a seconda delle fasi.

Questo esempio è solo  
un suggerimento, non è  
da prendere alla lettera.

# Diagramma di sequenza di sistema

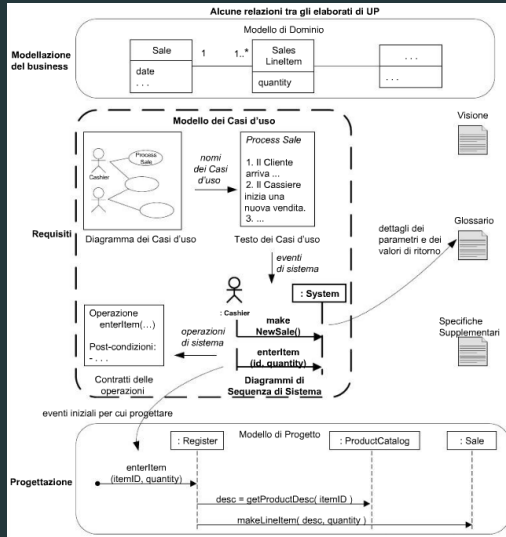
## Il Diagramma di sequenza di sistema (SSD)

è un elaborato della disciplina dei requisiti che illustra **eventi** di input e di output relativi ai sistemi in discussione.

Nota: non è menzionato esplicitamente in UP.

- I diagrammi di sequenza di sistema sono espressi attraverso i **diagrammi di sequenza di UML**
- Il sistema è modellato come una *“scatola nera”*
- Usualmente si modella un SSD per ogni caso d'uso per lo scenario principale e per ogni scenario alternativo
- Lo SSD costituisce un input per i **contratti** delle operazioni e, soprattutto, per la progettazione degli oggetti

# Relazioni tra elaborati di UP



I casi d'uso descrivono il modo in cui gli attori esterni interagiscono con il sistema software che interessa creare.

## Eventi

Durante un'interazione con il sistema software, un attore genera degli **eventi di sistema**, che costituiscono un input per il sistema, di solito per richiedere l'esecuzione di alcune **operazioni di sistema**.

- Le operazioni di sistema sono operazioni che il sistema deve definire proprio per gestire tali eventi
- Un evento è qualcosa di importante o degno di nota che avviene durante l'esecuzione di un sistema
- Un evento di sistema è un evento esterno al sistema, di input, di solito generato da un attore per interagire con il sistema



I diagrammi di sequenza sono utili per illustrare interazioni tra attori e le operazioni iniziate da essi.

## Diagrammi di sequenza di sistema

È una figura che mostra, per un particolare scenario di un caso d'uso, gli **eventi** generati dagli attori esterni al sistema, il loro **ordine** e gli eventi inter-sistema.

Nota: la qualifica “di sistema” enfatizza l'applicazione dei diagrammi di sequenza UML ai sistemi, considerati a *scatola nera*.

Un sistema reagisce a tre cose:

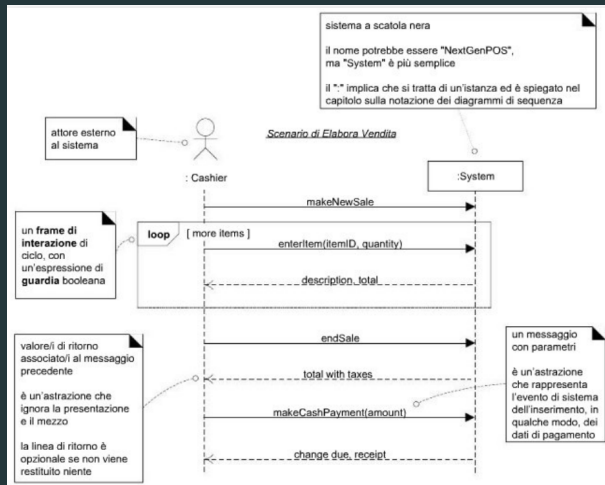
- **eventi esterni** da parte di attori umani o sistemi informatici
- **eventi temporali**
- guasti o **eccezioni**

Il software deve essere progettato proprio per gestire questi eventi e generare delle risposte.

# Esempio di SSD

Eventi di sistema:

- *makeNewSale*: il cassiere inizia una nuova vendita
- *enterItem*: il cassiere inserisce il codice identificativo di un articolo
- *endSale*: il cassiere indica di aver terminato l'inserimento degli articoli acquistati
- *makeCashPayment*: il cassiere indica che il cliente sta pagando in contanti e inserisce l'importo offerto dal cliente



Usualmente un SSD mostra gli eventi di sistema per un solo scenario di un caso d'uso, e può essere generato per ispezione da tale scenario.

Un SSD mostra:

- l'attore primario del caso d'uso
- il sistema in discussione
- i passi che rappresentano le interazioni tra il sistema e l'attore

Le interazioni iniziate dall'attore primario nei confronti del sistema sono mostrate come messaggi con parametri.

# Gli SSD derivano dai casi d'uso

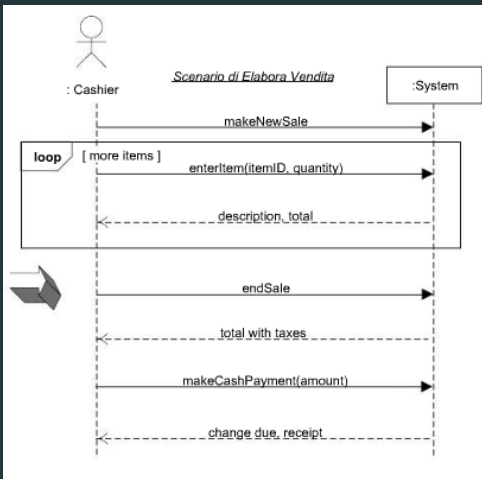
Scenario di base di *Elabora Vendita*,  
con pagamento in contanti:

1. Il Cliente arriva alla cassa POS con gli articoli e/o i servizi da acquistare.
2. Il Cassiere inizia una nuova vendita.
3. Il Cassiere inserisce il codice identificativo di un articolo.
4. Il Sistema registra la riga di vendita per l'articolo e mostra una descrizione dell'articolo, il suo prezzo e il totale parziale.  
*Il Cassiere ripete i passi 3-4 fino a che non indica che ha terminato.*
5. Il Sistema mostra il totale.
6. Il Cassiere riferisce il totale al Cliente, e richiede il pagamento.
7. Il Cliente paga (in contanti) e il sistema gestisce il pagamento.
8. Il Sistema registra la vendita completata.
9. Il Sistema genera la ricevuta.
10. Il Cliente va via con la ricevuta e gli articoli acquistati.

# Gli SSD derivano dai casi d'uso

## Scenario di base di *Elabora Vendita*, con pagamento in contanti:

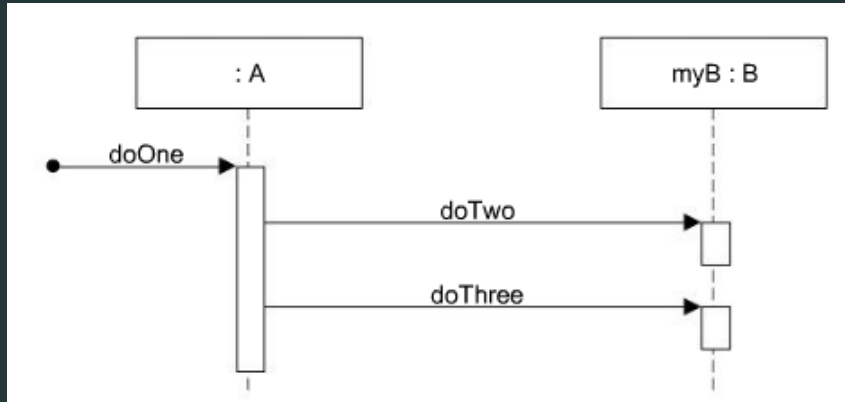
1. Il Cliente arriva alla cassa POS con gli articoli e/o i servizi da acquistare.
2. Il Cassiere inizia una nuova vendita.
3. Il Cassiere inserisce il codice identificativo di un articolo.
4. Il Sistema registra la riga di vendita per l'articolo e mostra una descrizione dell'articolo, il suo prezzo e il totale parziale. *Il Cassiere ripete i passi 3-4 fino a che non indica che ha terminato.*
5. Il Sistema mostra il totale.
6. Il Cassiere riferisce il totale al Cliente, e richiede il pagamento.
7. Il Cliente paga (in contanti) e il sistema gestisce il pagamento.
8. Il Sistema registra la vendita completata.
9. Il Sistema genera la ricevuta.
10. Il Cliente va via con la ricevuta e gli articoli acquistati.



## Un po' di notazione

---

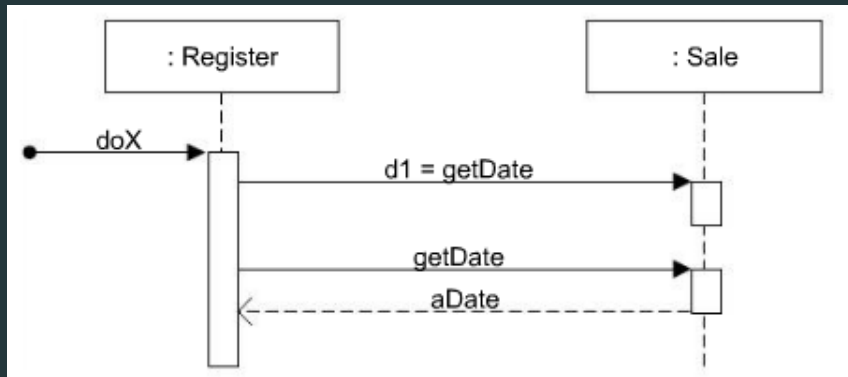
## Semplice diagramma di sequenza



©C. Larman. Applicare UML e i Pattern. Pearson, 2016.

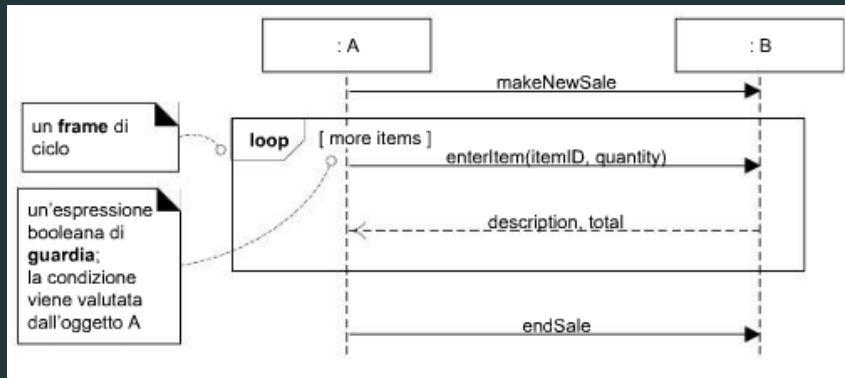


## Due modi per mostrare un risultato di ritorno da un messaggio



©C. Larman. Applicare UML e i Pattern. Pearson, 2016.

# Esempio di frame di UML

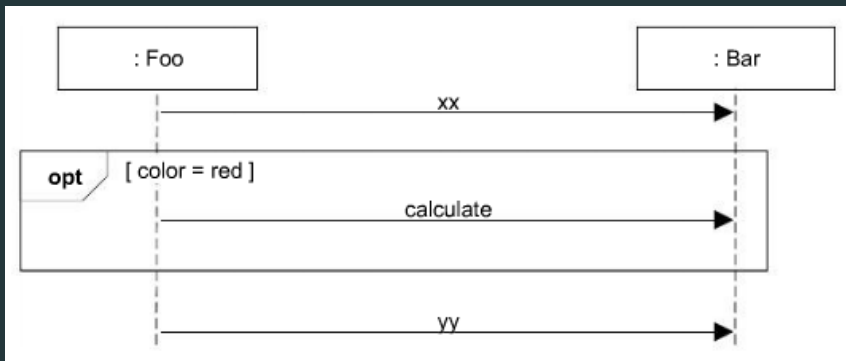


©C. Larman. Applicare UML e i Pattern. Pearson, 2016.

# Operatori comuni per i frame di UML

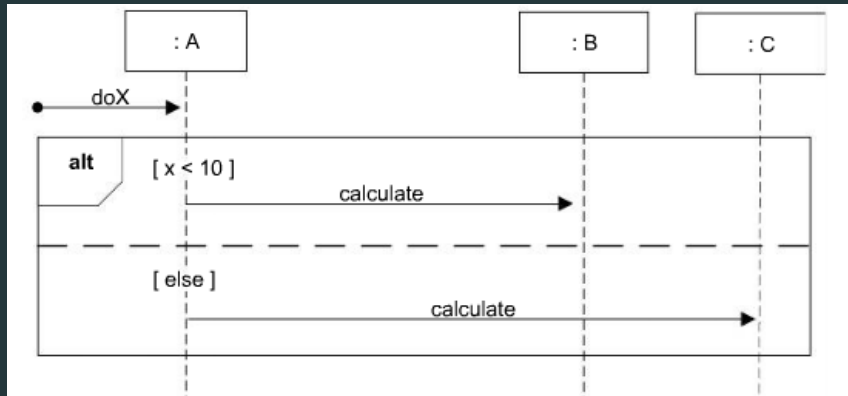
Operatore frame	Significato
alt	Frammento alternativo per logica mutuamente espressa nella guardia (un'istruzione <i>if-else</i> di Java o del C).
opt	Frammento opzionale che viene eseguito se la guardia è vera (un'istruzione <i>if</i> ).
loop	Frammento da eseguire ripetutamente finché la guardia è vera (un'istruzione <i>while</i> o <i>for</i> ). Si può anche scrivere <i>loop(n)</i> per indicare un ciclo da ripetere n volte. Può rappresentare anche l'istruzione <i>foreach</i> del C# o l'istruzione <i>for</i> "avanzata" di Java.
par	Frammenti che vengono eseguiti in parallelo.
region	Regione critica all'interno della quale può essere in esecuzione un solo thread.

# Un messaggio condizionale



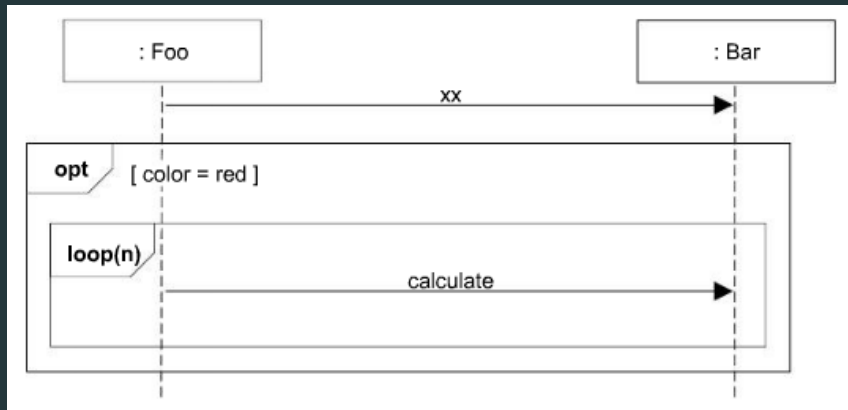
©C. Larman. Applicare UML e i Pattern. Pearson, 2016.

## Messaggi condizionali mutuamente esclusivi



©C. Larman. Applicare UML e i Pattern. Pearson, 2016.

## Annidamento di frame



©C. Larman. Applicare UML e i Pattern. Pearson, 2016.