

# Diagrammi di Sequenza

## Sequence Diagram

a cura di **Angelo Furfaro**  
da “UML Distilled”  
Martin Fowler

Dipartimento di Ingegneria Informatica Elettronica Modellistica e Sistemistica  
Università della Calabria, 87036 Rende(CS) - Italy  
Email: [a.furfaro@dimes.unical.it](mailto:a.furfaro@dimes.unical.it)  
Web: <http://angelo.furfaro.dimes.unical.it>

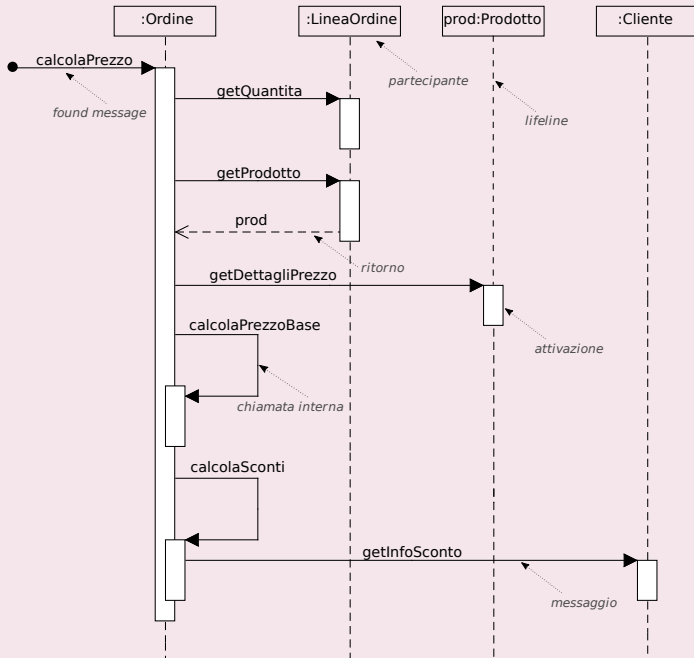
# Diagrammi di Sequenza

- I diagrammi di **sequenza** appartengono alla categoria dei diagrammi di **interazione** i quali descrivono la collaborazione di un gruppo di oggetti che devono implementare un certo comportamento.
- Un diagramma di sequenza documenta tipicamente il comportamento di un singolo scenario.
- Il diagramma include un certo numero di oggetti e i messaggi scambiati tra di essi durante l'esecuzione del caso d'uso.

## Si consideri il seguente scenario: calcolo del prezzo di un ordine

- L'ordine deve calcolare il prezzo di ciascuna linea.
- Per farlo occorre ottenere la quantità relativa al prodotto ed il suo prezzo unitario.
- Una volta calcolato il prezzo totale, si calcola lo sconto che dipende dallo specifico cliente.
- Queste interazioni sono illustrate nel diagramma di sequenza riportato nella slide seguente.

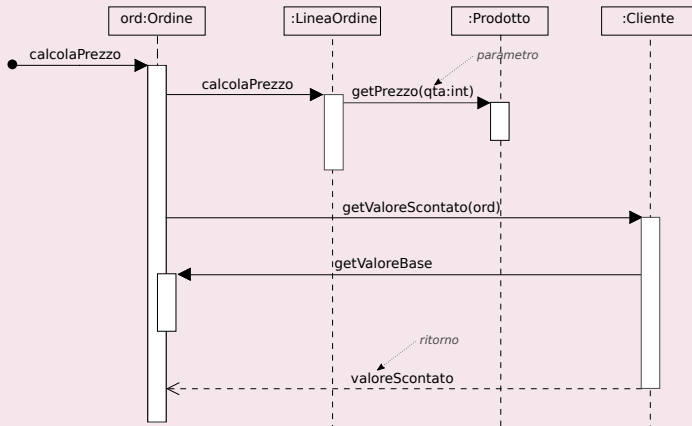
# Diagramma di sequenza a controllo centralizzato



# Elementi di un diagramma di sequenza

- Ciascun partecipante è rappresentato da un box che contiene il nome (nella notazione degli Object Diagram ma non sottolineato) e da una linea tratteggiata verticale detta “linea di vita” (*lifeline*).
- L'ordinamento dei messaggi è determinato scorrendo il diagramma dall'alto verso il basso.
- UML 1 prescriveva che i partecipanti fossero oggetti.
- Ogni linea di vita ha una *barra di attivazione* che indica quando il partecipante è attivo nell'interazione.
- Le frecce indicano i messaggi (invocazioni) e sono etichettate con il nome del messaggio (nome del metodo).
- Le frecce di ritorno sono opzionali. Quando presenti, possono riportare il valore risultante dall'invocazione.
- Nell'esempio, il primo messaggio non scaturisce da un partecipante ed è detto *messaggio trovato* (found message).
- Quando un partecipante invia un messaggio a se stesso, di solito si sovrappone una barra di attivazione a quella già presente.

## Diagramma di sequenza a controllo distribuito



- Il seguente diagramma illustra una differente modalità di interazione tra i partecipanti.
- Nel caso precedente, un partecipante svolge tutta l'interazione, gli altri forniscono solo i dati (controllo centralizzato).
- In questo caso, i compiti sono ripartiti tra i partecipanti (controllo distribuito).

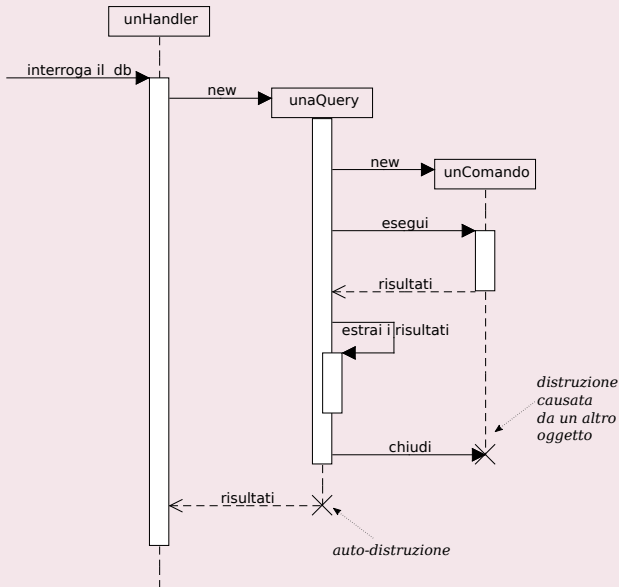
# Controllo centralizzato e controllo distribuito

- Ognuno dei due stili ha i suoi punti di forza e di debolezza.
- I progettisti che non hanno ben assimilato il paradigma Object-Oriented spesso adottano quello centralizzato.
- I puristi degli oggetti preferiscono quello distribuito.
- Per limitare gli effetti dei cambiamenti è opportuno che dati e logica che li gestisce siano concentrati nello stesso posto.
- Distribuendo il controllo si può utilmente sfruttare il polimorfismo: se gli algoritmi che determinano il prezzo sono diversi in funzione della tipologia del prodotto si possono utilizzare le sottoclassi per gestire le variazioni.
- Lo stile Object-Oriented prevede l'uso di molti piccoli oggetti, ognuno dotato di diversi metodi che possono essere ridefiniti e consentono di inserire cambiamenti.

# Creazione e distruzione dei partecipanti

- I diagrammi di sequenza prevedono una notazione particolare per indicare la creazione e distruzione dei partecipanti.
- Per la creazione si disegna la freccia del messaggio che causa la creazione in modo che punti al box del partecipante creato.
- Il nome del messaggio è opzionale se si utilizza il costruttore.
- Per indicare che il partecipante creato fa subito qualcosa, si disegna la barra di attivazione attaccata al box.
- La distruzione di un partecipante è indicata con una X posizionata sulla lifeline.
- Se la freccia di un messaggio termina sulla X, si intende che un partecipante (il mittente del messaggio) ne sta cancellando un altro.
- Se la X non è raggiunta da nessun messaggio, si intende che il partecipante si autodistrugge ( o che la cancellazione è gestita da un garbage collector).

# Creazione e distruzione dei partecipanti





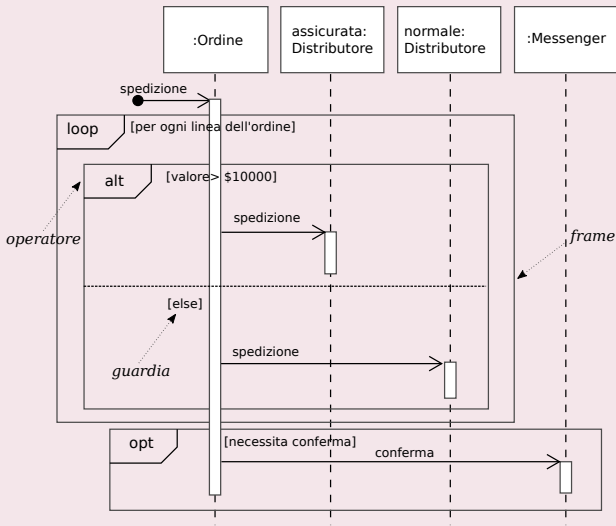
# Frame di interazione

- Uno dei problemi dei diagrammi di sequenza è la loro inadeguatezza a rappresentare comportamenti ciclici e/o condizionali.
- Per fare ciò ricorre all'uso di **frame di interazione**, cornici che includono uno o più frammenti di diagramma.
- Ogni frame ha un operatore e ogni frammento può avere una guardia.

Operatore	Significato
alt	Frammenti multipli in alternativa; si esegue solo quello per cui la condizione è soddisfatta.
opt	Opzionale; il frammento è eseguito solo se la condizione è verificata.
par	Parallelo; ogni frammento è eseguito in parallelo.
loop	Ciclo; il frammento può essere eseguito più volte, la guardia indica la condizione di iterazione.
region	Regione critica; il frammento può essere eseguito da un solo thread alla volta.
neg	Negativo; il frammento mostra un'interazione non valida.
ref	Riferimento; si riferisce ad un'interazione mostrata in un altro diagramma.
sd	Sequence Diagram; usato per racchiudere un intero diagramma di sequenza.

# Cicli e condizioni

## Frame di interazione



## Pseudocodice

```
procedure spedizione
  foreach (lineaordine)
    if (prodotto.valore > $10K)
      assicurata.spedizione
    else
      normale.spedizione
    end if
  end for
  if (necessita conferma)
    messenger.conferma
  end if
end procedure
```

# Chiamate sincrone ed asincrone

- UML distingue i **messaggi sincroni** dai **messaggi asincroni** dal tipo di freccia utilizzata.
- Una freccia con punta a forma di triangolo pieno indica un messaggio sincrono.
- Una freccia la cui punta è costituita da due linee sottili denota un messaggio asincrono.
- Un oggetto che invia un messaggio sincrono deve attendere la risposta prima di poter proseguire.
- Le chiamate asincrone sono comuni nelle applicazioni multi-thread o nelle applicazioni distribuite.
- Versioni precedenti ad UML 1.4 utilizzavano una notazione differente. In particolare i messaggi asincroni erano indicati con “mezze frecce”.
- Anche i frame di interazione sono una novità di UML 2.0