



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

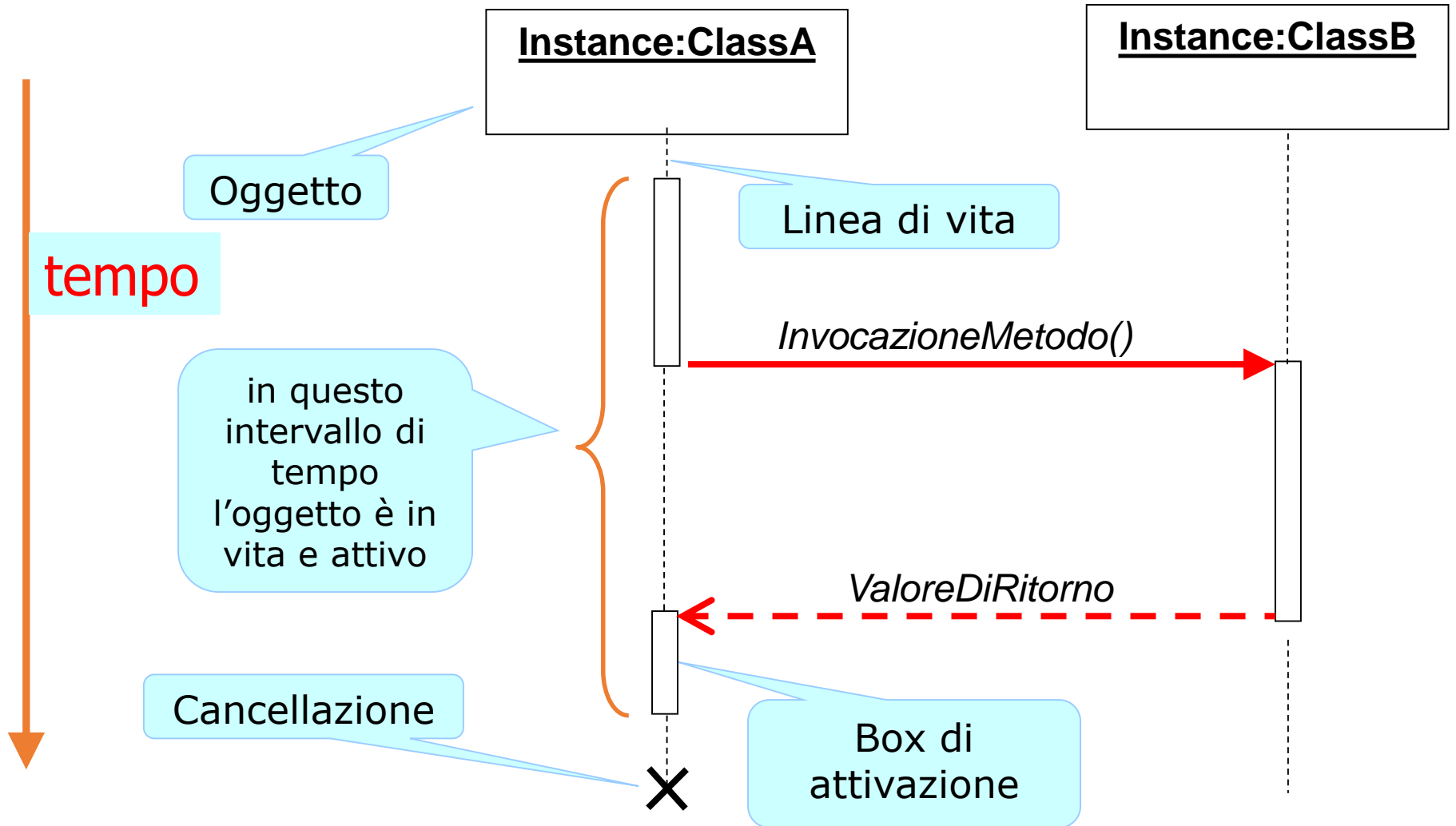
# Ingegneria del Software – Sequence Diagrams

Prof. Sergio Di Martino

# Sequence Diagram

- Mostra la sequenza temporale dei messaggi che gli oggetti si scambiano per portare a termine una funzionalità.
- E' un diagramma di interazione: evidenzia come una funzionalità è realizzata tramite la collaborazione di un insieme di oggetti
- E' uno dei principali input per l'implementazione dello scenario
  - E' utilizzato in analisi e poi ad un maggior livello di dettaglio in design

# Sequence Diagram



# Sequence Diagram

- La ricezione di un messaggio determina l'attivazione di un metodo
  - L'attivazione è rappresentata da un rettangolo sulla linea della vita, da cui altri messaggi possono prendere origine
  - La lunghezza del rettangolo rappresenta il tempo durante il quale l'operazione è attiva
- La vita degli oggetti
  - Il tempo procede verticalmente dal top al bottom
  - Al top del diagramma si trovano gli oggetti che esistono prima del 1° messaggio inviato
  - Oggetti creati durante l'interazione sono illustrati con il messaggio <<create>>
  - Oggetti distrutti durante l'interazione sono evidenziati con una croce
  - La linea tratteggiata indica il tempo in cui l'oggetto può ricevere messaggi

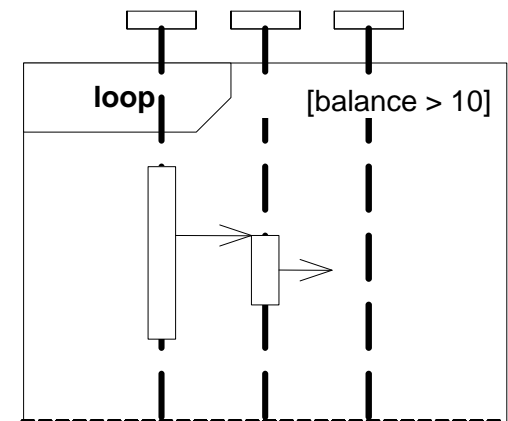
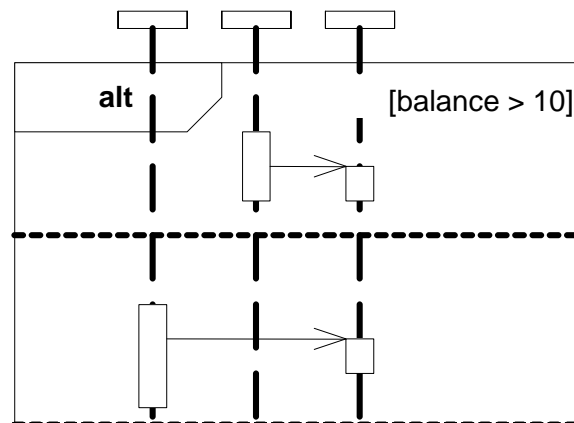
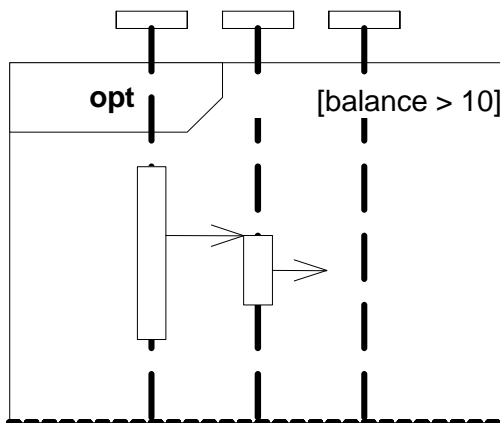
# Messaggi nei Sequence

- In generale un messaggio rappresenta il trasferimento del controllo da un oggetto ad un altro
- Se l'oggetto che invia il messaggio rimane in attesa che l'oggetto ricevente ritorni, si ha un messaggio **sincrono**
- Se l'oggetto che invia il messaggio prosegue la propria elaborazione in parallelo all'oggetto chiamato, siamo in presenza di un messaggio **asincrono**.
- Il valore restituito all'oggetto chiamante si indica con un **messaggio di ritorno**

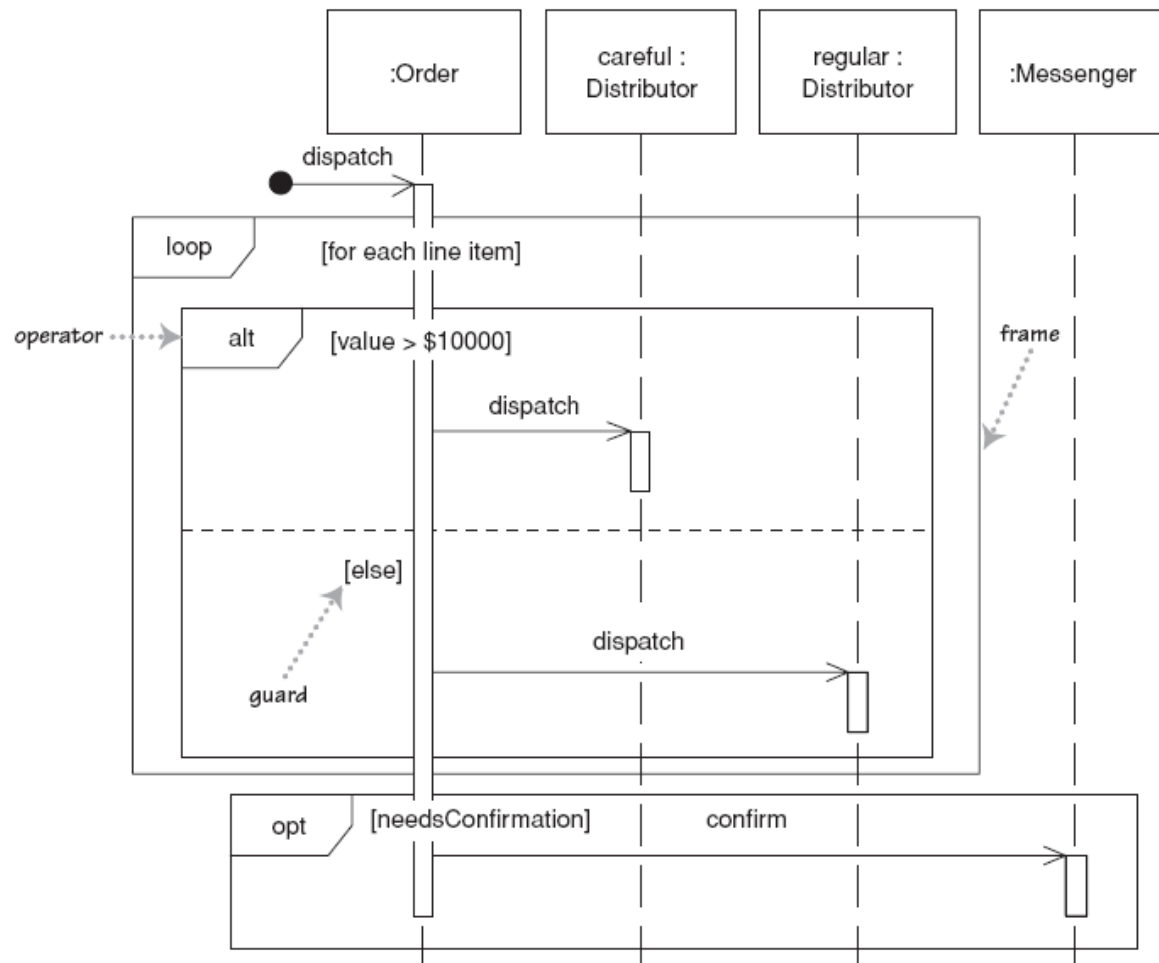


# Opzioni e Cicli

- Si indicano con una cornice intorno ad una parte del Sequence, per indicare che quella sezione è opzionale o viene ripetuta.
- Rappresentazioni:
  - if -> OPT [condition]
  - if/else -> ALT [condition], separati da linea orizzontale tratteggiata
  - cicli -> LOOP [condition o items su cui ciclare]

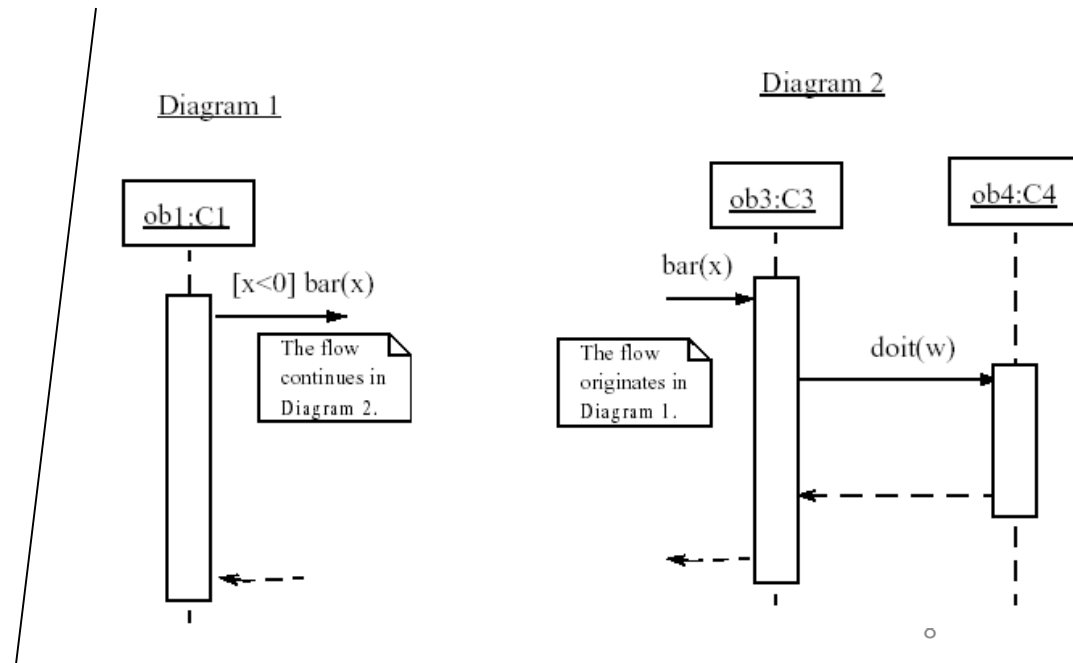
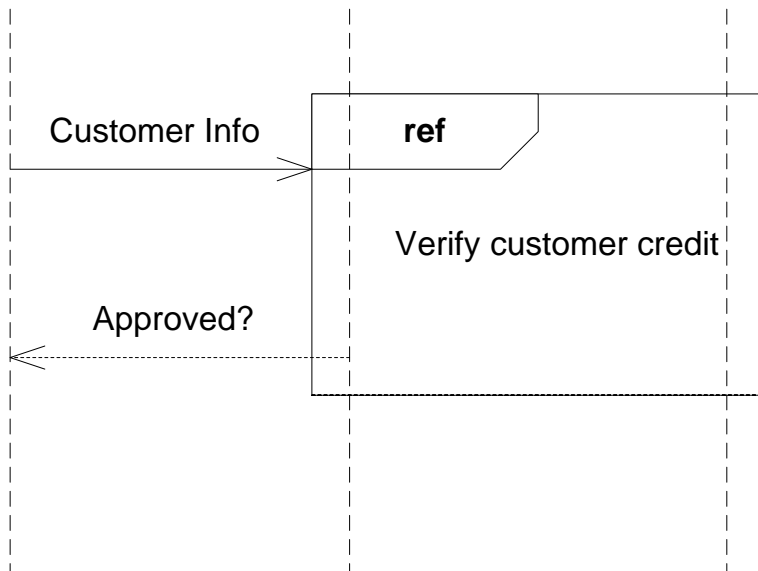


# Esempio



# Linking sequence diagrams

- Se un Sequence è troppo complesso o fa riferimento ad un altro Sequence, si può utilizzare il REF, con:
  - Un rettangolo con label REF, col nome dell'altro diagramma
  - Una freccia che punta a tale rettangolo
  - Una eventuale condizione per specificare quando si fa il riferimento



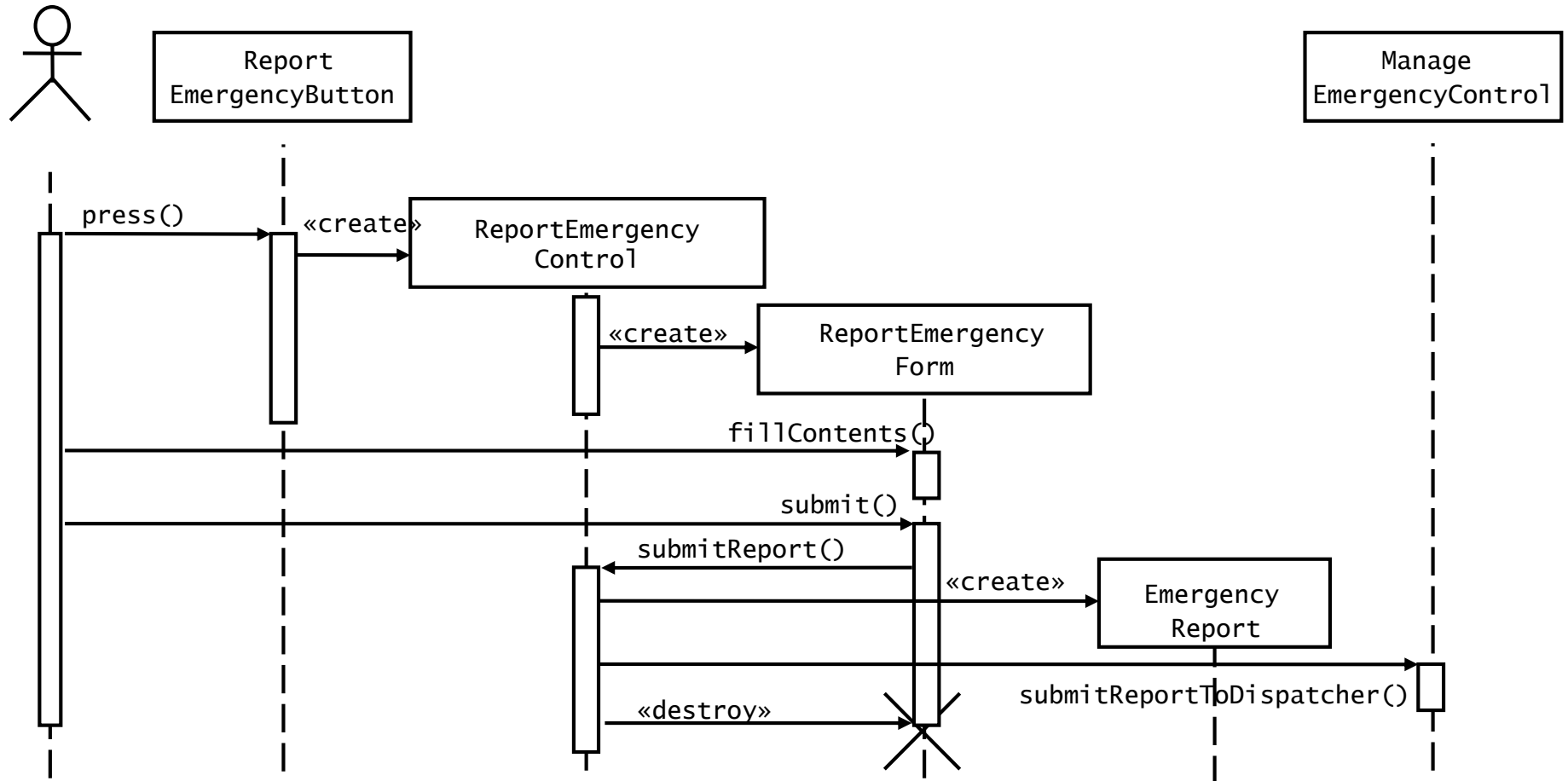


# Sequence Diagram in fase di analisi

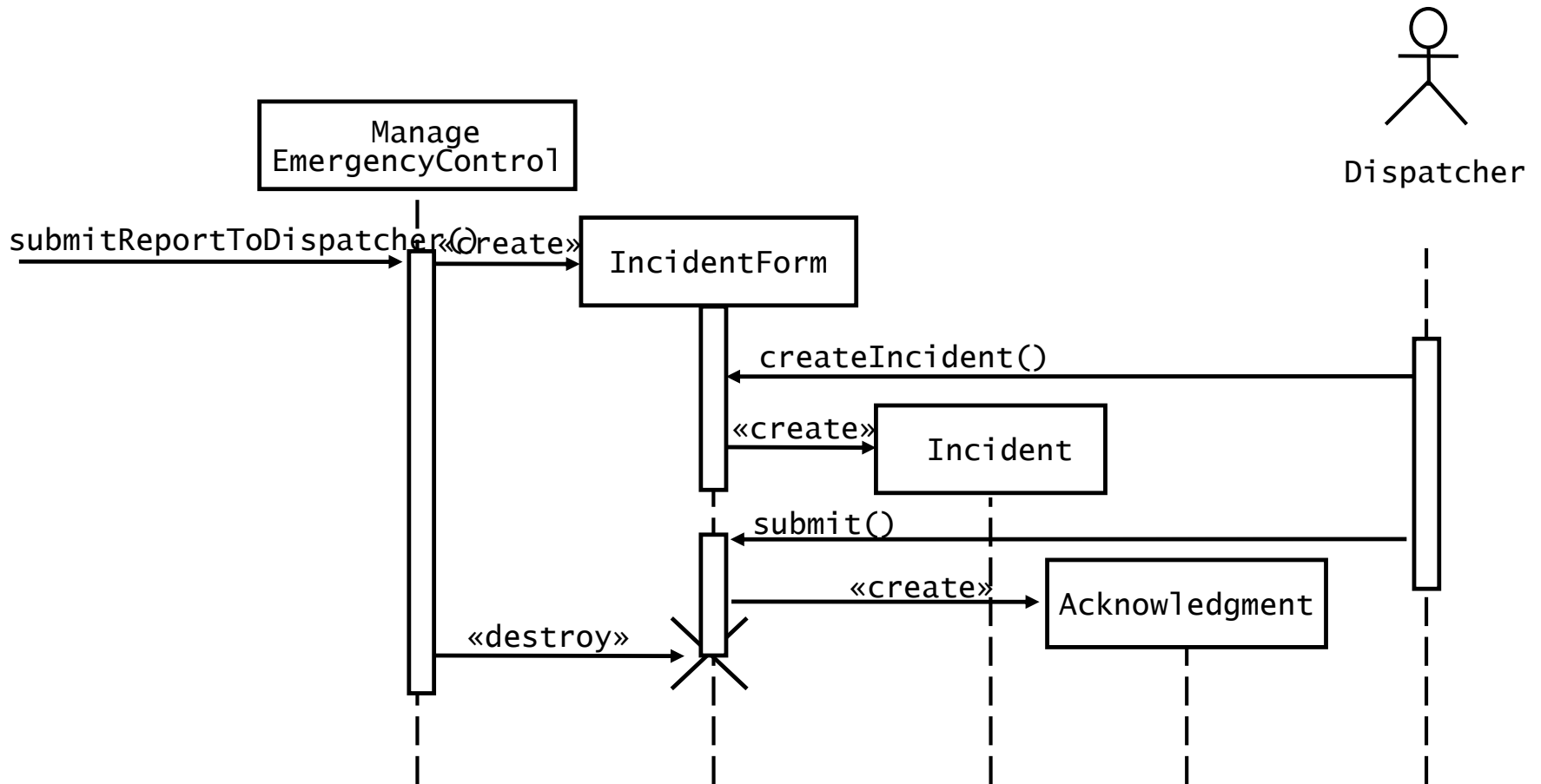
- Convenzioni per utilizzare i Sequence Diagrams con l'euristica Three-Object-Type in fase di analisi:
  - La colonna più a sinistra rappresenta l'attore che inizia lo use case
  - La seconda colonna -> oggetto Boundary con cui l'attore interagisce per iniziare lo use case
  - La terza colonna -> oggetto Control che gestisce il resto dello use case
  - Le altre colonne possono rappresentare qualunque oggetto che interviene nel caso d'uso.
    - Gli oggetti Control creano altri oggetti Boundary/Entity e possono interagire con altri oggetti
  - Oggetti Control accedono ad altri oggetti Entity e Boundary
  - Gli oggetti Entity non accedono mai agli oggetti Control e Boundary: ciò rende più facile condividere oggetti Entity tra più use case

# Sequence diagram for the ReportEmergency use case.

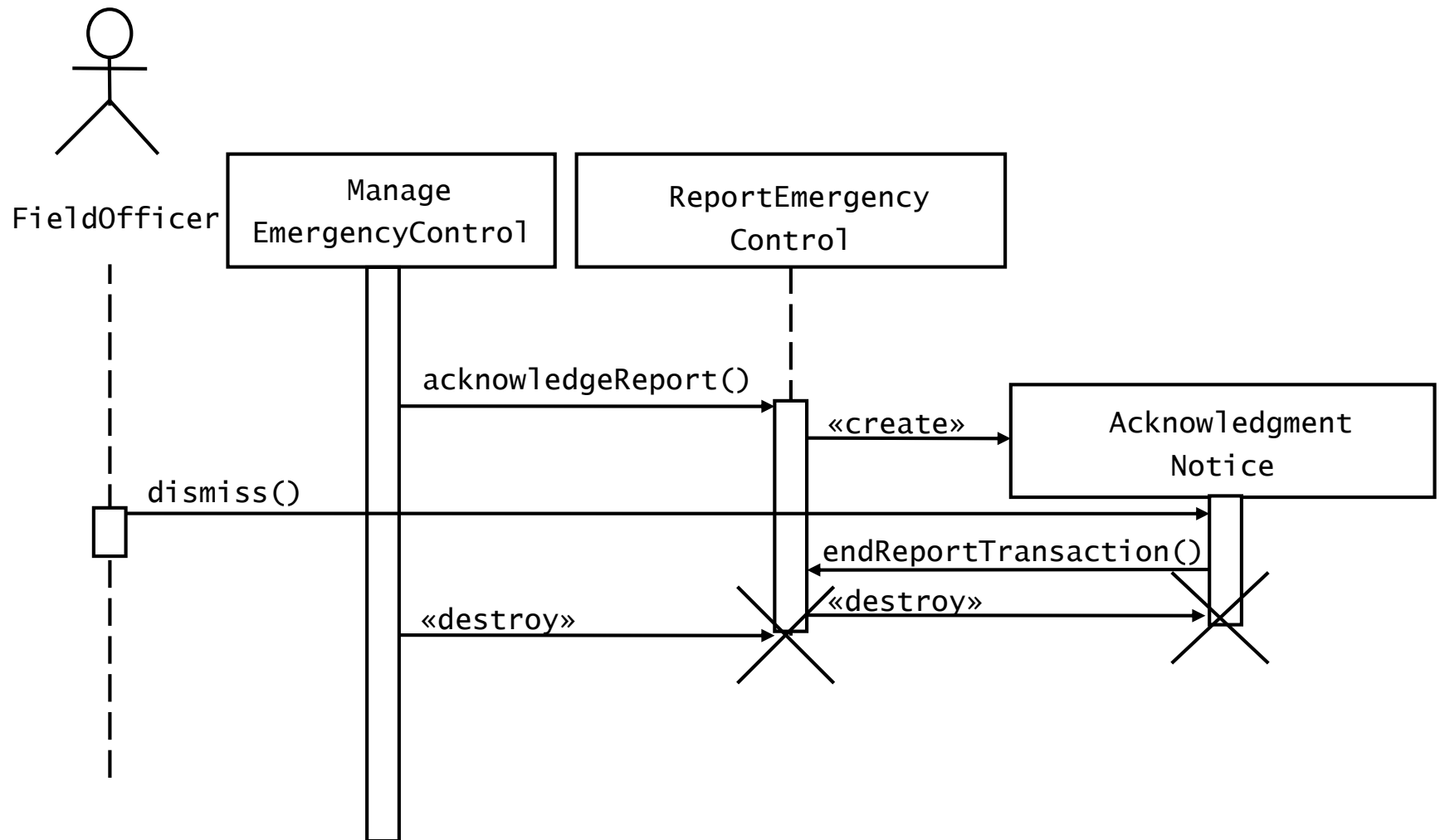
FieldOfficer



# Sequence diagram for the ReportEmergency use case (continued).



# Sequence diagram for the ReportEmergency use case (continued).



# Analisi e Sequence Diagram

- Durante l'analisi i Sequence Diagram sono usati per individuare
  - nuovi oggetti
  - comportamenti mancanti
- Disegnare Sequence Diagram è un'attività laboriosa, quindi:
  - Occorre dare priorità a quelle funzionalità problematiche o non ben specificate
  - Per le parti ben definite può essere utile solo per evitare di posticipare alcune decisioni chiave