I modelli dinamici descrivono il comportamento del sistema in funzione del tempo. Quindi per poter descrivere il sistema per mezzo dei modelli dinamici, UML fornisce Interaction diagram e StateChart Diagram.

I diagrammi di interazioni descrive il comportamento dinamico e come avviene la comunicazione tra gli oggetti. Sono utili per identificare oggetti aggiuntivi che partecipano ai casi d'uso. Una interazione è un comportamento che comprende un insieme di messaggi scambiati tra un insieme di oggetti nell'ambito di un contesto per raggiungere uno scopo.

Una interazione avviene tra oggetti tra cui esiste un link. Un oggetto manda un messaggio ad un'altro oggetto per mezzo di questo link; nel messaggio sono contenuti delle informazioni con l'aspettativa che quando arriverà all'oggetto ricevente, ne scaturirà l'inizio di una attività. Conoscendo il meccanismo, è necessario quindi che durante l'invio di un messaggio bisogna specificare il

ricevente, il messaggio ed eventuali informazioni aggiuntive. UML propone due diversi tipi di Interaction Diagramm:

- 1. Sequence Diagram
- 2. Collaboration Diagram

Entrambi esprimo informazioni simili ma con una logica diversa.

The Sequence Diagram si concentrano sull'ordine dei messaggi che scorrono tra gli oggetti. Tuttavia, è conveniente usarli entrambi per descrivere il comportamento di un sistema.

The Collaboration Diagram vengono utilizzati per visualizzare l'organizzazione strutturale degli oggetti e le loro interazioni. Sequence Diagram

funzione di sistema. Come funziona? C' è un "attore" che rappresenta un tipo di ruolo in cui interagisce con il sistema e i suoi oggetti. E' possibile avere più attori in un diagramma di sequenza. Il sequence diagram inizia con l'avvio del

I diagrammi di sequenza descrive l'interazione tra oggetti in un ordine sequenziale, cioè l'ordine in cui avvengono queste interazioni. Descrivono quindi come e in quale ordine gli oggetti in una

caso d'uso da parte dell'attore partecipante.

Una "linea vita" è un elemento che raffigura un singolo partecipante in un diagramma di sequenze. Quindi ogni istanza in un diagramma di sequenza è rappresenta da un'ancora di salvezza e si trovano nella parte superiore del diagramma di sequenza.

Qual è la differenza tra un ancora di salvezza e un attore?

Un'ancora di salvezza ritrae sempre un oggetto interno al sistema mentre gli attori oggetti esterni al sistema.

La comunicazione tra oggetti è rappresentata mediante "messaggi". I messaggi vengono visualizzati in ordine sequenziale sulla linea di vita. Rappresentiamo i messaggi utilizzando le frecce. I messaggi possono essere classificati nelle sequenze categorie:

- Messaggi sincroni: un messaggio sincrono attende una risposta prima che l'interazione possa andare avanti. Il mittente attende che il destinatario abbia completato l'elaborazione del messaggio. Usiamo una solida fresca per rappresentare tale messaggio; • Messaggi asincroni: un messaggio asincrono non attende una risposta dal destinatario. L'interazione procede indipendentemente dal fatto che il destinatario elabori o meno il messaggio
- precedente. • Crea messaggio: utilizziamo un messaggio <<create>> per creare un'istanza di un nuovo oggetto nel diagramma di sequenza.
- Elimina messaggio: viene utilizzato un messaggio <<destroy>> per eliminare un oggetto. Quando un oggetto viene deallocato nella memoria o viene distrutto all'interno del sistema,
- utilizziamo tale messaggio. E' rappresentato da una freccia che termina con una X
- Quando sono usati:
- Durante l'analisi dei requisiti:

- Aiuta a perfezionare le descrizioni dei casi d'uso;
- Aiuta a trovare gli oggetti partecipanti; Aiuta a identificare le responsabilità delle classi;
- Aiuta a identificare la comunicazione necessaria tra gli oggetti;
- Durante la progettazione del sistema:
  - Può aiutare a perfezionare le interfacce del sottosistema

  - Può aiutare a identificare il flusso di controllo e i messaggi tra gli oggetti, una volta progettati le classi concrete
- Linee guida:
- I diagrammi di sequenza possono essere basati su un caso d'uso o uno scenario.

  - Disposizione: • Rendi la prima colonna corrispondente all'attore che avvia il caso d'uso( o scenario);
  - Rendi la seconda colonna un oggetto Boundary( l'attore lo utilizza per avviare il caso d'uso) • Rendi la terza colonna un oggetto Control, che gestisce il resto del caso d'uso

  - Creazione di oggetti:

Gli oggetti Control devono essere creati da oggetti Boundary che avviano casi d'uso

 Gli oggetti Control possono creare altri oggetti Boundary Accesso:

• Gli oggetti Entity sono accessibili da oggetti Control e Boundary.

- Gli oggetti Entity non devono chiamare il confine o controllare gli oggetti. Ciò semplice la condivisione degli oggetti entity tra i casi d'uso.
- Collaboration Diagram Specifica gli oggetti che collaborano tra loro in un dato scenario, ed i messaggi che si indirizzano. La sequenza dei messaggi è meno evidente che nel diagramma di sequenza mentre sono più evidenti i legami tra gli oggetti; per meglio visualizzare l'ordine sequenziale dello scambio dei messaggi è possibile numerare i message anteponendo al loro nome un numero che indica l'ordine

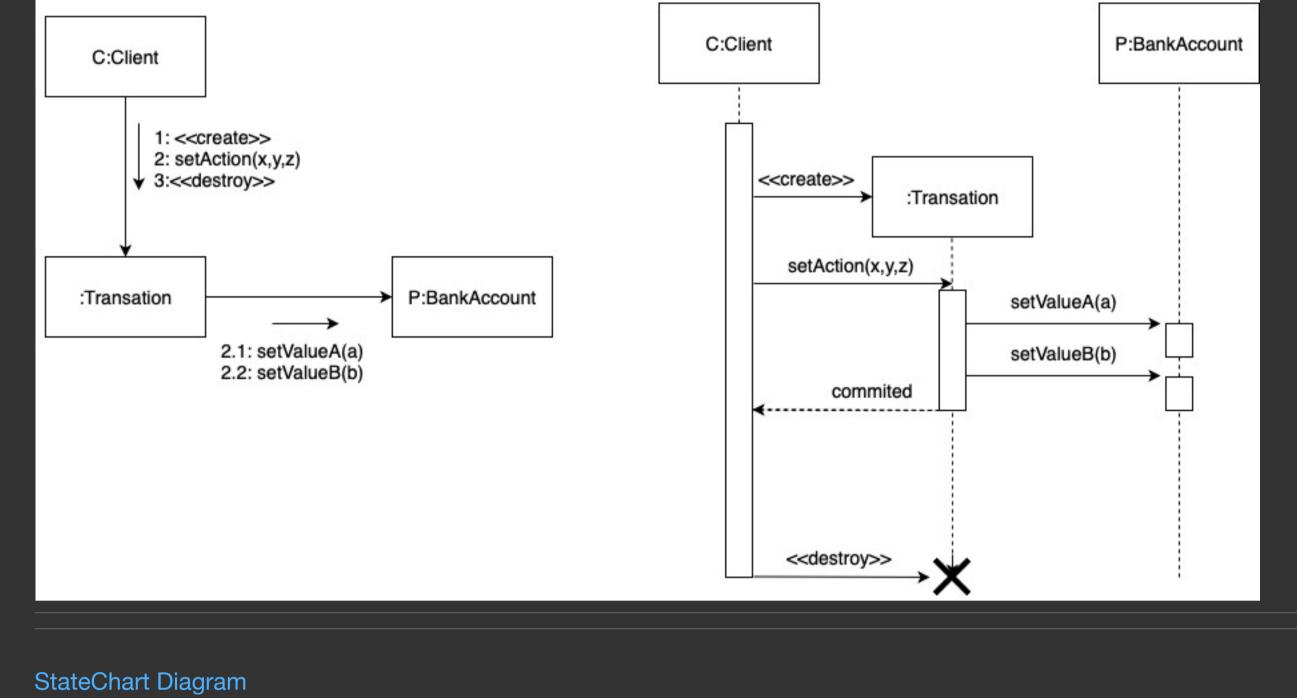
### nella sequenza.

Quando si utilizza? Questi diagrammi possono essere utilizzati quando le relazioni tra gli oggetti sono fondamentali per la visualizzazione. Di seguito, alcuni esempi casi di uso di collaboration diagram: 1. Modulazione di collaborazioni, meccanismi o organizzazione strutturale all'interno di un progetto di sistema.

2. Fornire una panoramica degli oggetti che collaborano all'interno di un sistema orientato agli oggetti. 3. Esporre molti scenari alternativi per lo stesso caso d'uso.

- 4. Visualizzazione della logica complessa dietro un'operazione.
- Non è utile nei casi in cui il numero di oggetti e messaggi è elevata per il motivo che diventa difficile da leggere e da utilizzare in modo efficiente.

SequenceDiagram CollaborationDiagram



# difficoltà a capirli. Quindi per comprendere meglio tale oggetto, si approfitta il modello diagrammi a stati in UML.

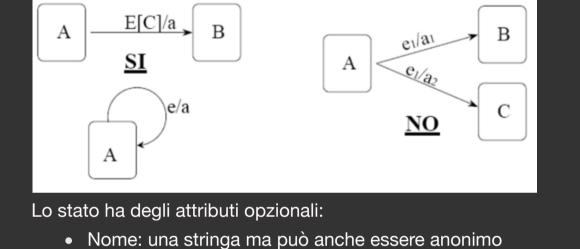
Questo tipo di diagramma mostra quali azioni sono valide per un determinato oggetto, a seconda dello stato in cui si trova attualmente.

Tali diagrammi mostrano vari stati in cui può trovarsi un oggetto e le transizioni tra tali stati. Uno **stato** è una situazione nella vita dell'oggetto in questione in cui esso soddisfa una qualche condizione, esegue una qualche attività o è in attesa di un qualche evento. Ci sono due tipi di stati particolari: Stato iniziale: una freccia non etichettata che parte dal "vuoto" ed entra in uno stato.

Gli oggetti hanno sia comportamento che stato quindi fanno cose e conoscono cose. C'è la possibilità che vi siano degli oggetti cosi complicati che gli sviluppatori possono riscontrare delle

Tali due stati possono anche essere denotati da appositi simboli: stato iniziale

stato finale



Stato finale: una freccia non etichettata che esce da uno stato e finisce nel "vuoto".

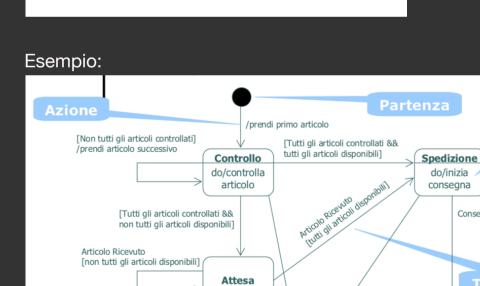
#### • Entry actions: eseguite all'ingresso/uscita dallo stato Azione eseguita ogni volta che si entra nelllo stato ( cioè in risposta all'evento entry). Exit actions:

- Azione eseguita ogni volta che si lascia lo stato (cioè in risposta all'evento exit).
- Transizioni interne: Transizione che gestisce un evento senza cambiare stato. Attività:
- Corrisponde all'esecuzione di un'attività, la cui interno è composta da un insieme di azioni. Può prevedere un lasso di tempo considerevole e può essere interrotta da un evento ( non • Sottostati: struttura innestata di dati; disgiunti ( sequenzialmente attivi) o concorrenti (concordantemente attivi).
- Tracking entry / setMode(on) exit / setMode(off) exit action do / followTarget

Consegnato

Consegnato

**Transizione** 



Cancellato

Cancellato

name

entry action

activity

self-transition

composto.

all'oggetto.

• Interno:

• Evento:

Quando sono usati:

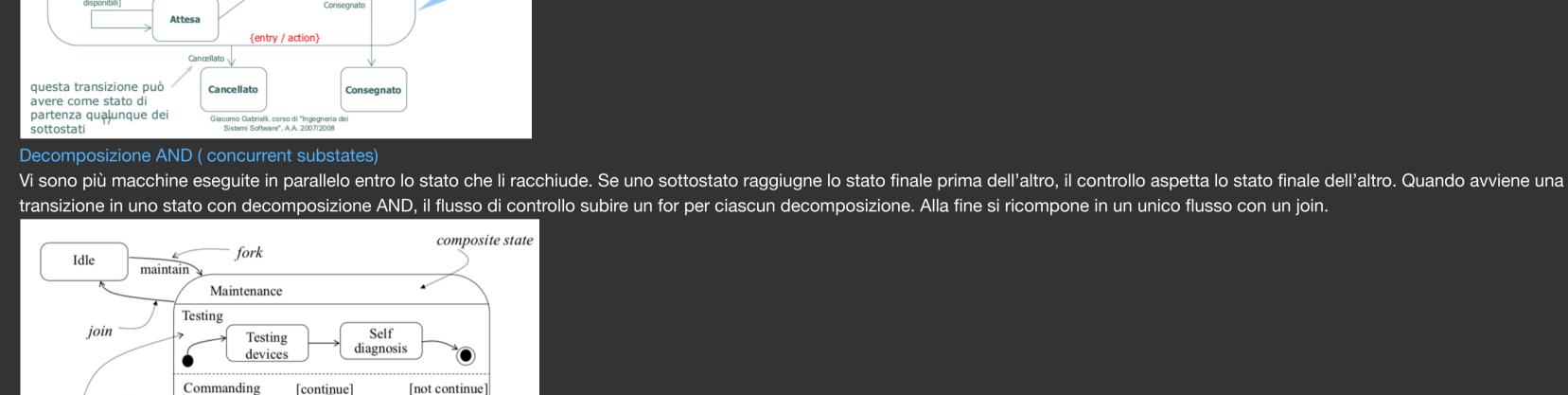
Stato

[Tutti gli articoli controllati && non tutti gli articoli disponibili] [non tutti gli articoli disponibili]

Stato composto o macro-stato o superato

Ordine [Non tutti gli articoli controllati] [Tutti gli articoli controllati && Controllo tutti gli articoli disponibili] Spedizione do/controlla do/inizia articolo Superstato

Uno stato può anche essere composto (detto anche macro-stato) in cui ha un nome e che contiene a sua volta un diagramma di stato. I sottostati ereditano le transizioni in uscita dello stato



## Concurrent substate keyPress

Waiting

Command

Transizione Una transizione ,connessa da due stati, modifica permanentemente uno stato di un oggetto in un altro stato e tale modifica ha effetto quando verrà innescata da un evento interno o esterno

Il diagramma corrisponde ad un "automa determinismo", dove deterministico si riferisce che ad ogni coppia di stato in ingresso c'è una ed una sola transizione allo stato successivo quindi non ci

possono essere che due transizioni diverse portino allo stato eguale. La condizione è detta anche guardia e l'evento è "quasi" sempre presente mentre condizione e azione sono "opzionali". CI sono diversi tipi di transizioni:

Tale transizione è denotata da: Evento[Condizione]/Azione->E[C]/A

 Transizione che viene subito all'interno dello stato poichè lo stato finale è uguale allo stato iniziale. Esterno: Transizione che porta da uno stato iniziale a uno stato finale diverso.

Al termine dell'attività\_A si passa automaticamente nello stato B, si può dire cioè che è il termine dell'attività ad innescare la transizione fra stati.

Se l'oggetto si trova nello stato S1, e si verifica l'evento E, e se la condizione C è verificata allora viene eseguita l'azione A e l'oggetto passa nello stato S2.

 Una espressione logica da valutare. La transizione avviene se l'evento accade e la condizione di guardia è vera Azione:

Condizione di guardia:

Riguardo agli attributi della transizione:

• E' associata alla transizione, considera un processo rapido, non interrompibile da un evento. (atomica). Transizioni senza eventi scattano o Con guardia: se la condizione di guardia è vera. • Senza guardia: se l'attività interna allo stato iniziale è completata.

• E' l'occorrenza di uno stimo che può innescare una transizione tra stati.

- Una transizione che non abbia nessun evento ad innescarla è detta triggerless
  - В do / Attività\_B do / Attività\_A

triggerless trans.

Può aiutare a identificare importanti attributi di oggetto

• Può aiutare a perfezionare la descrizione del comportamento di un oggetto • Dovrebbe concentrarsi sugli attributi importanti che influenzano il comportamento dell'oggetto

Utilizzato per descrivere una sequenza di transizione di stato in un oggetto in risposta a eventi

 Dovrebbe essere usato per rappresentare un comportamento non banale di un determinato tipo di oggetto **Activity Diagram** 

> Document Incident

Un tipo speciale di diagrammi di stato dove tutti gli stati sono stati di azioni. Rappresentano il sequenziamento e il coordinamento dei comportamenti di livello inferiore. Un diagramma di attività indica come viene realizzato in termini di una o più sequenze di attività e i flussi di oggetti necessari per coordinare l'attività.

Archive

Incident

Stesso caso della figura elencata sopra ma in questo caso modellato per mezzo di activity diagram.

- All'implementazione di un'operazione Ad uno Use Case A tutto il sistema

A una classe

- A processi di business Quando sono usati?

Handle

Incident

A quale ambito sono associati?

- Utili per modellare: • comportamenti sequenziali
  - operazioni
- business workflow
- concorrenza sistemi distribuiti

- non determinismo