Sistemi Informativi:

modellazione dinamica

Sommario

modellazione della dinamica di un sistema:

diagrammi di stato

di attivita’

di sequenza

Di interazione/collaborazione (cenni)

Specificare gli stati

Lo stato di un oggetto e’ determinato dai valori dei suoi attributi e delle associazioni

La specifica dello stato include il modello delle strutture dati

Modello dinamico

Definizione: una collezione di diagrammi dinamici, uno per ogni classe con rilevante comportamento dinamico

Scopo: rilevare e fornire metodi al modello degli oggetti

Come: partire da caso d'uso o scenario

- modellare interazione tra oggetti: diagramma di sequenza
- modellare il comportamento di un singolo oggetto: diagramma di stato

Flusso di eventi in un caso d’uso: Telefonata

Chiamante solleva cornetta

Inizia il suono di chiamata

Chiamante compone il numero

Telefono squilla

Il chiamato risponde

Lo squillo si interrompe

….

Che cos'e' un evento?

Qualcosa che avviene in un istante definito di tempo

Relazioni:

- causato da: prima,dopo
- concorrenti: indipendenti

mandano informazione da un oggetto ad un altro tramite messaggi

Gerarchia di eventi:

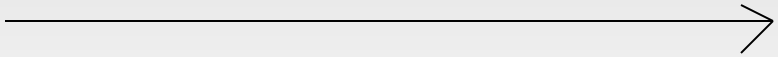
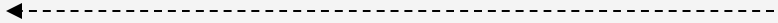
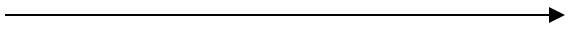
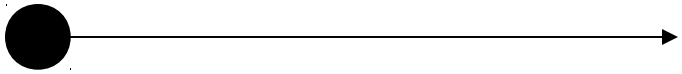
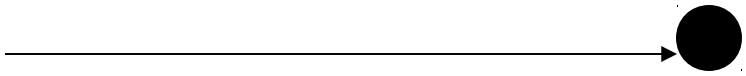
- Input->Mouse->MouseMotion

Messaggio

Tipi di messaggio:

- Segnale
  - Comunicazione asincrona
  - Il mittente continua senza aspettare risposta
- Chiamata
  - Comunicazione sincrona
  - Il mittente aspetta una risposta
- In linea di principio, un messaggio sincrono puo’essere modellato con due asincroni (occorre pero’ correlarli)
- Creazione/Distruzione

Notazione per i messaggi

aMessage(aParameter) 	sincrono
aMessage(aParameter) 	asincrono
	messaggio di ritorno
<<create>> aMess()  :A	creazione
<<destroy>> 	distruzione
	messaggio trovato
	messaggio perso

scambio messaggi

se un oggetto o1 manda un messaggio ad oN, deve esserci un cammino dalla classe o1 ad oN, attraverso associazioni o attributi derivati

ogni oggetto intermedio fa da tramite per recuperare l'identificatore dell'oggetto successivo nel cammino

la navigazione si puo' fare tramite attributi o metodi

{o1.o2.o3.getO4()……..}.ciao()

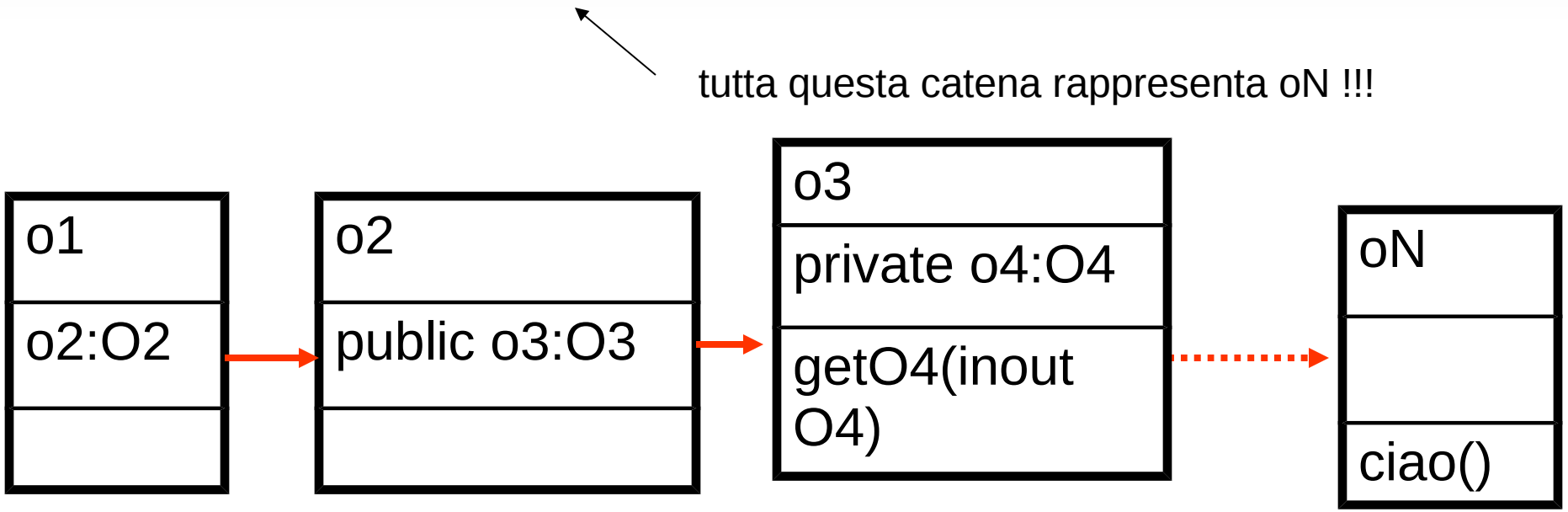


Diagramma di sequenza

dal flusso eventi dello scenario si procede al diagramma di sequenza

e’ un grafico che rappresenta il procedere nel tempo dello scambio di messaggi nel caso d’uso

premessa: gli oggetti/classi sono stati gia’ parzialmente identificati in fase precedente

Euristica: un messaggio ha sempre mittente e destinatario. Sono due oggetti del caso d’uso

Esempio di flusso eventi

Stabilire connessione tra “tesseraIntelligente” e “computerDiBordo”

Stabilire connessione tra “computerDiBordo” e “sensoreSedile”

recupera posizione sedile e memorizzala su “tesseraIntelligente”

Oggetti ?

Sequence Diagram for "Get SeatPosition"

1. Establish connection between smart card and onboard computer

2. Establish connection between onboard computer and sensor for seat

3. Get current seat position and store on smart card

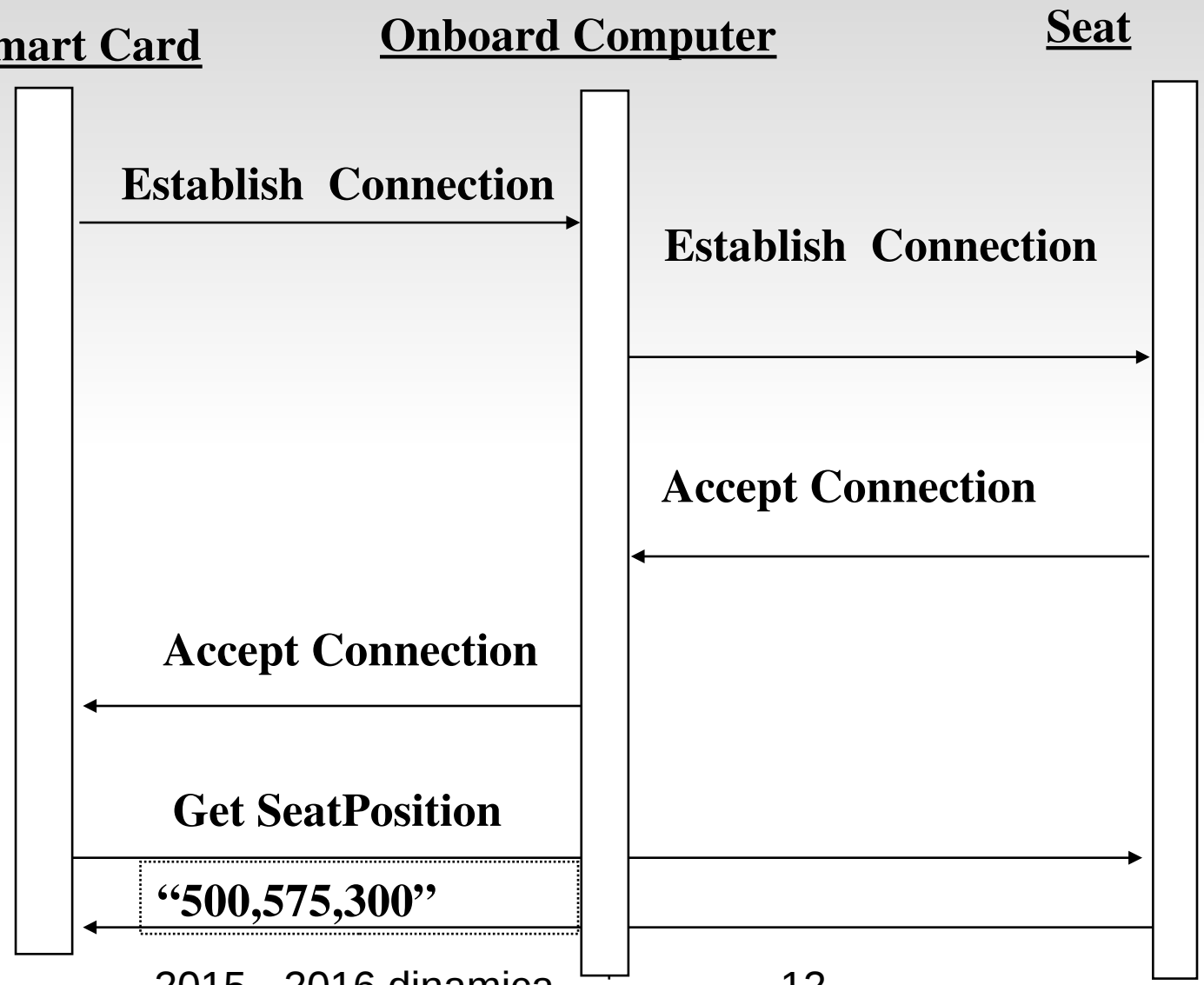


Diagramma di sequenza

Ogni colonna del diagramma descrive un oggetto

I rettangoli verticali rappresentano il periodo di attivita' dell'oggetto

Il tempo passa dalla cima al fondo del diagramma

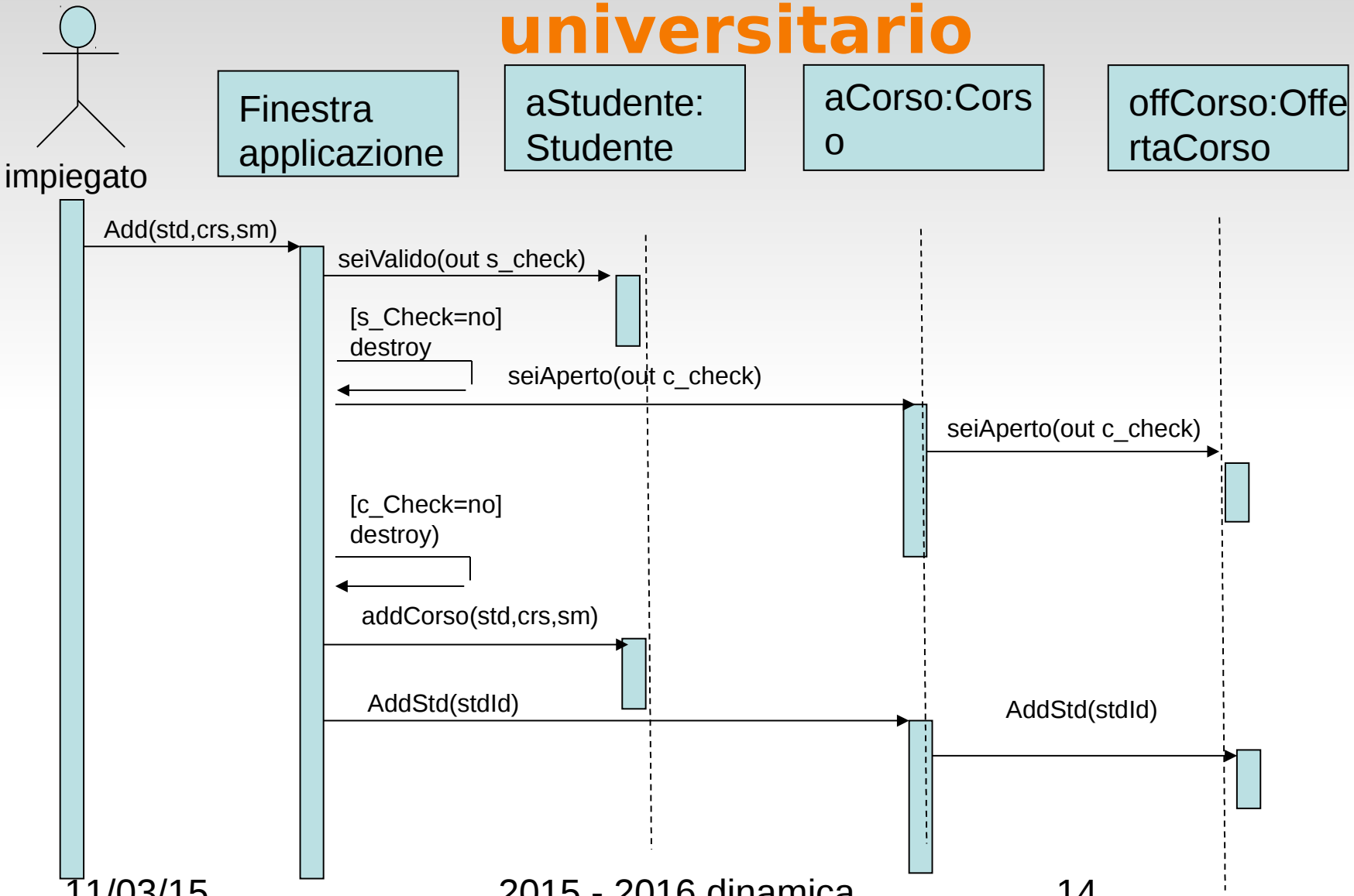
Le linee orizzontali rappresentano I messaggi

Due tipi di argomenti

- Ingresso (input) da mittente a destinatario
- Uscita (output) da destinatario a mittente

Uno * indica messaggio mandato ad un insieme di oggetti

Esempio: iscrizione corso universitario



Consuetudini per il diagramma

Disposizione:

- Colonna 1: attore che inizia il caso d'uso
- colonna 2: oggetto di confine del sistema
- colonna 3: oggetto di controllo che governa il resto del caso d'uso

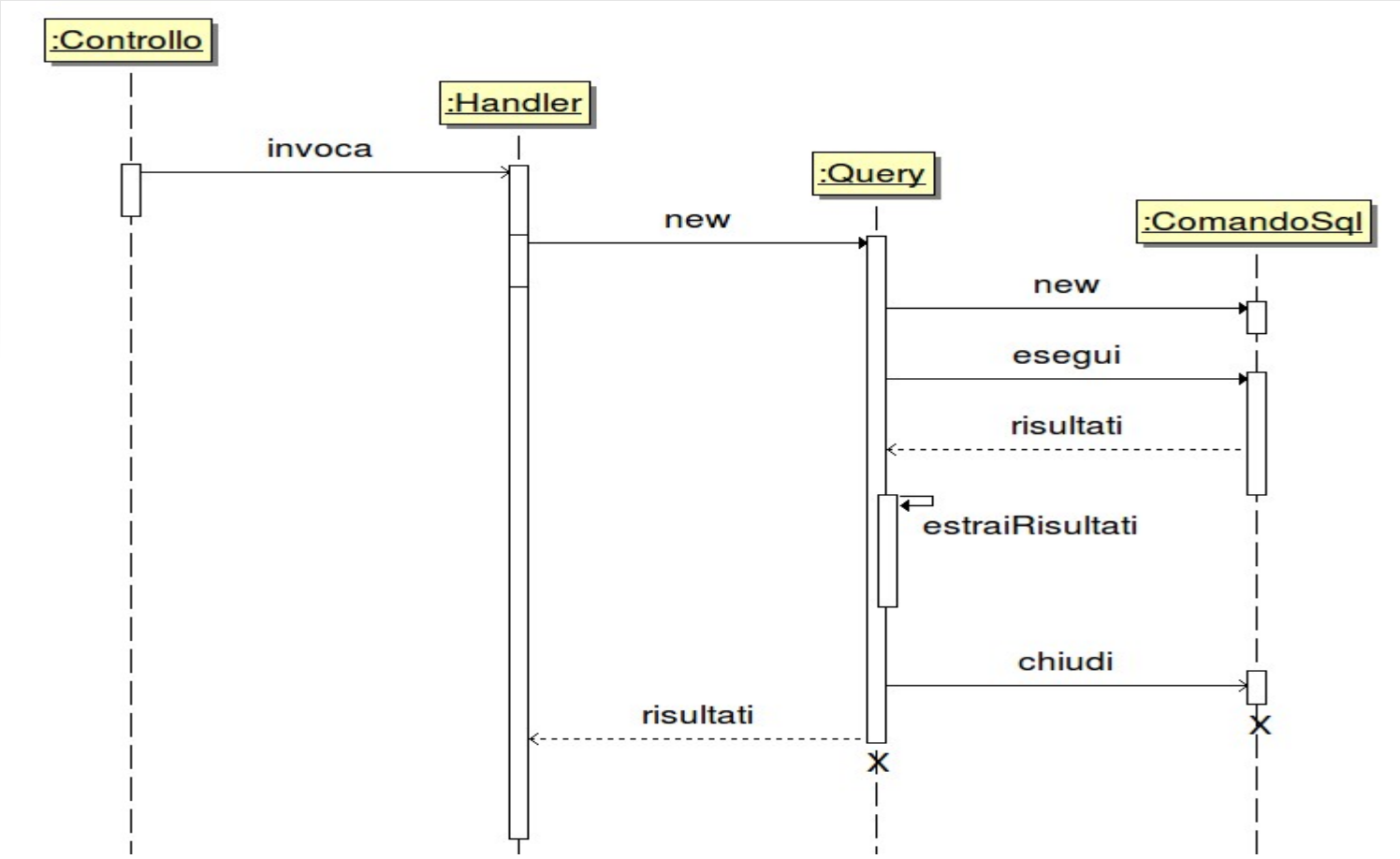
Creazione:

- gli oggetti controllo sono creati all'inizio del caso d'uso
- quelli di boundary da quelli di controllo

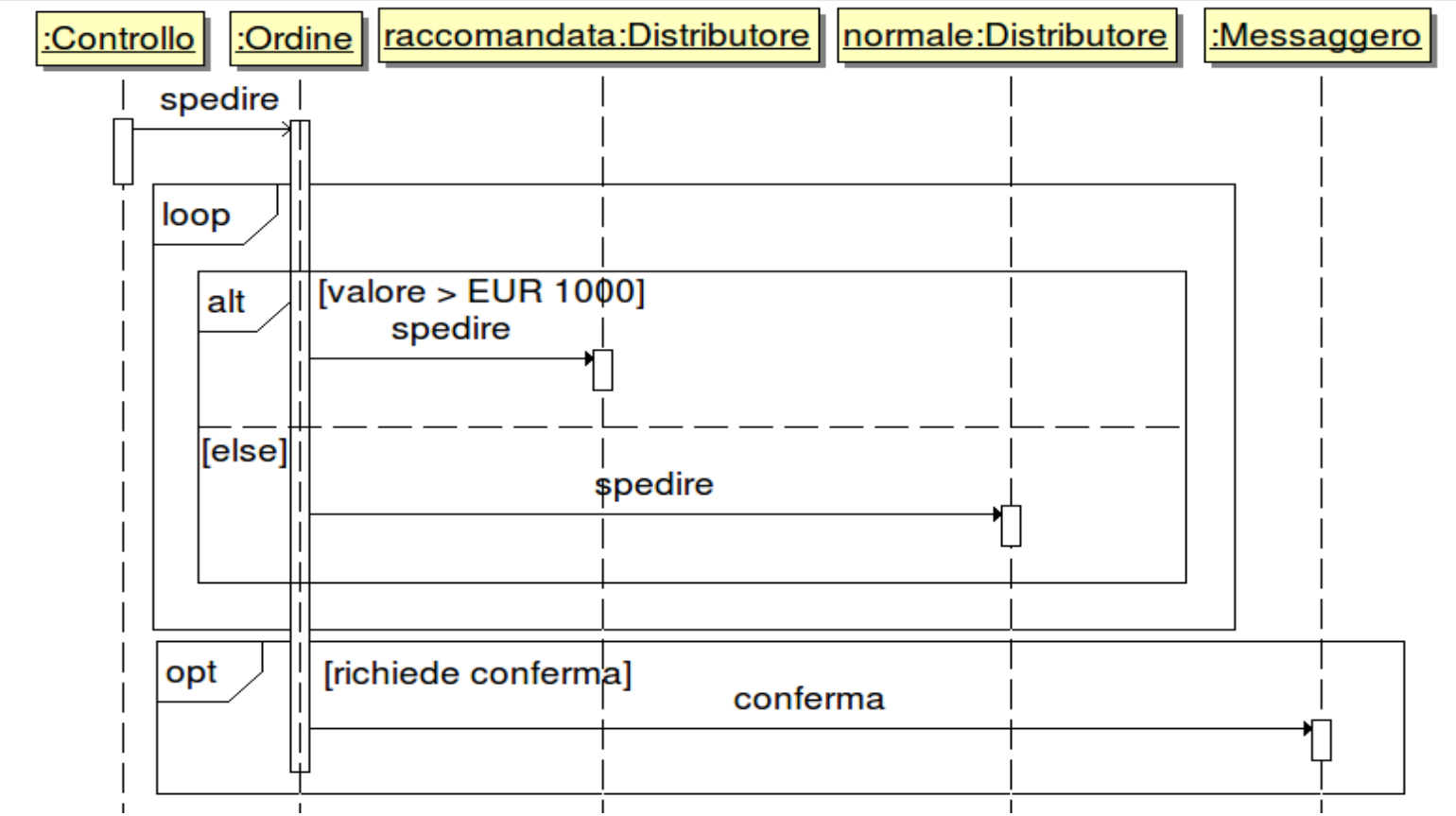
Accesso

- Gli oggetti entita' sono acceduti da boundary e controllo
- gli oggetti entita' non accedono agli altri. cio' li rende utilizzabili in diversi casi d'uso e tolleranti in caso di variazioni tecnologiche

Creazione/Distruzione



Frammenti


:Controllo
:Ordine
raccomandata:Distributore
normale:Distributore
:Messaggero
spedire
loop
alt
[valore > EUR 1000]
spedire
[else]
spedire
opt
[richiede conferma]
conferma

messaggi, classi, parametri (1)

a volte una determinata operazione puo' essere arbitrariamente realizzata da diverse classi, quale scegliere?

Es. vogliamo inviare il messaggio ad un cliente che l'ordine e' stato spedito:

- invia(cliente, messaggio, ordine)
- possibilita':
  - ordine.invia(cliente, messaggio)
  - messaggio.invia(cliente,ordine)
  - cliente.invia(messaggio,ordine)

messaggi, classi, parametri (2)

ci si chiede: quale classe ha piu' conoscenza per eseguire l'operazione?

sia ordine che cliente dispongono di dati indispensabili, (es indirizzo di spedizione) ma tutto sommato non sono tenuti a sapere dettagli di come spedire, soprattutto se il tipo di messaggio e' variabile (lettera, SMS…)

- lettera.invia(cliente,ordine)
- sms.invia(cliente,ordine)

modellare

Diagramma di stato

Sono dei grafcii I cui nodi sono degli stati ed i cui archi sono delle transizioni etichettate da nomi di eventi

due tipi di operazione:

- attivita': richiede tempo per il completamento
  - associata allo stato, interrompibile
- azione: operazione istantanea
  - associata ad eventi. non interrompibile

il diagramma modella eventi riferiti ad oggetti di **una singola** classe

Stato

Una astrazione degli attributi di una classe

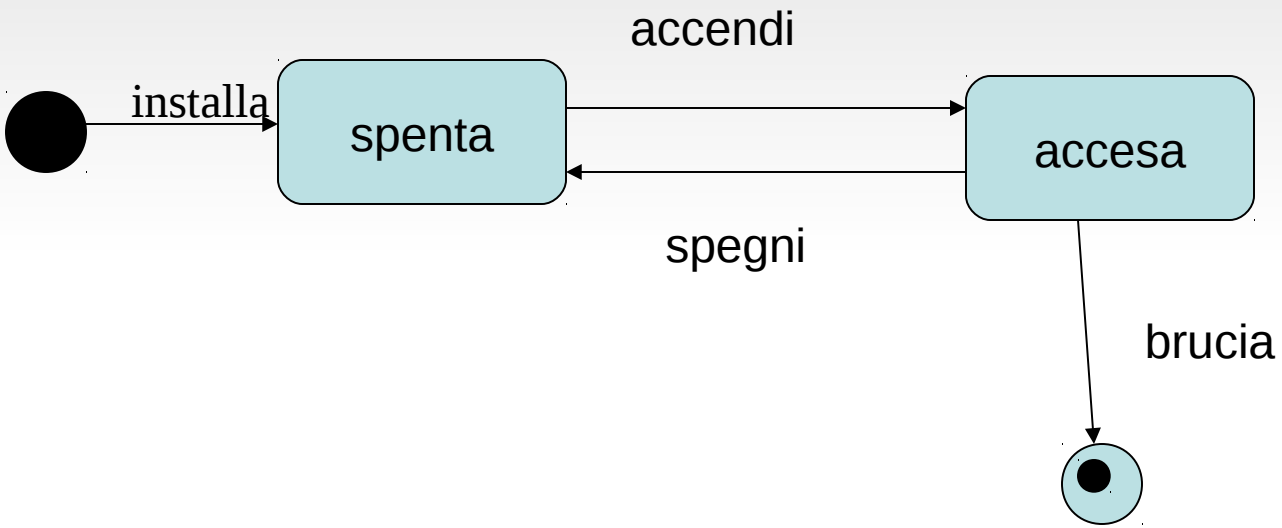
- e' l'aggregazione di diversi attributi della classe

raggruppa tutti quei valori di attributi che sono equivalenti per il diagramma.

- es conto in rosso: [disponibilita' < 0]

ha una durata

cambiamenti di stato: una lampadina



Specifica dei cambiamenti di stato

Importanti nei sistemi di controllo real-time

Diagrammi di stato

Per le classi che hanno comportamenti dinamico interessante

Cambiamenti del valore di attributi indicano cambiamento di stato

Non tutti i cambiamenti sono modellati esplicitamente: **solo quelli significativi**

Descrivono il comportamento di un oggetto durante la sua vita, comprendendo **piu' casi d'uso**

Transizione di un oggetto da uno stato all'altro

Ogni transizione deve avere un'etichetta, altrimenti non ci fermeremmo nello stato ma passeremmo subito a quello successivo

La transizione e' modellata da:

- evento (parametri) [condizione] azione (ECA)

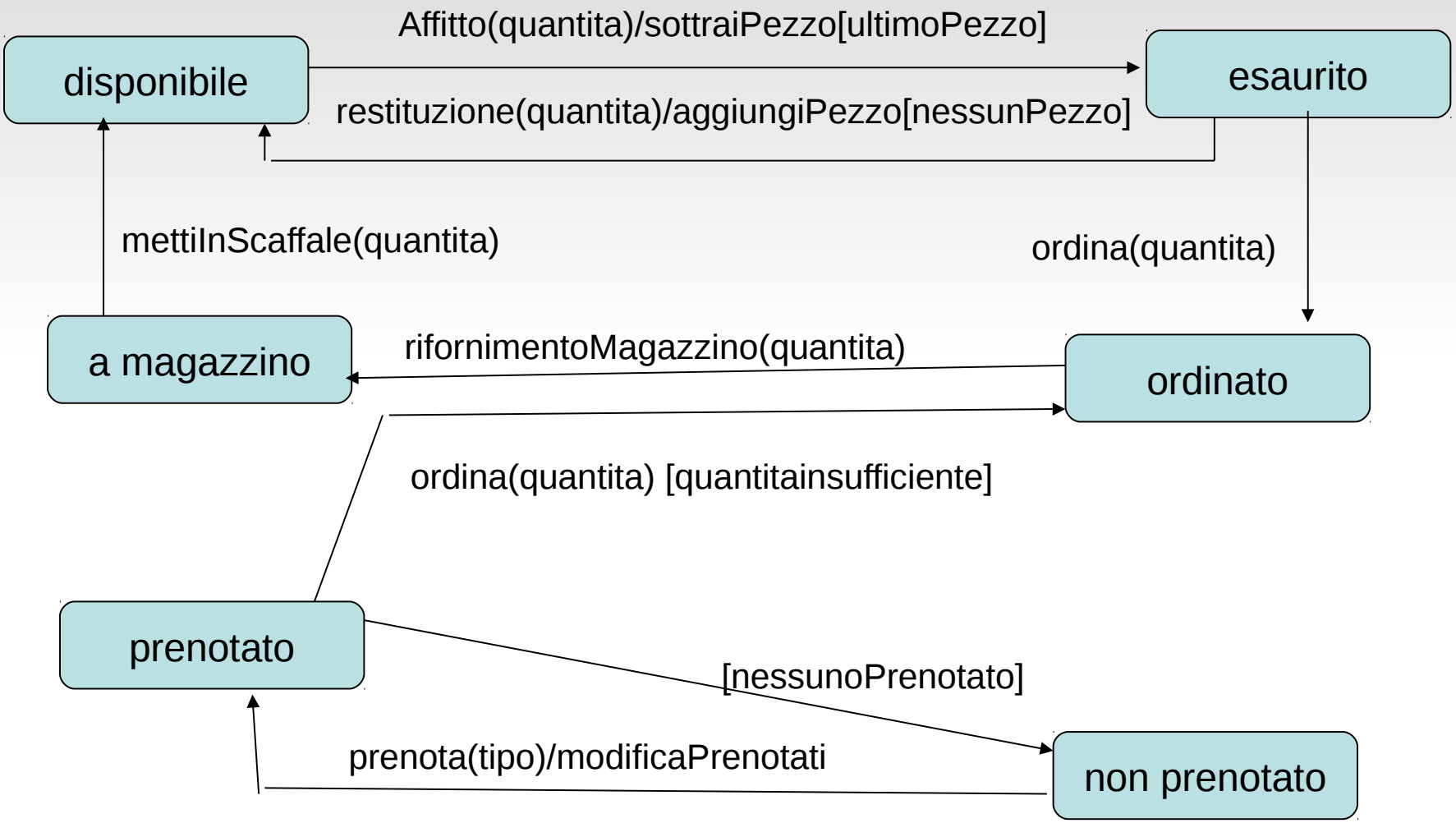
Puo' avvenire all'occorrenza dei seguenti eventi

- Ricezione di un messaggio (sincrono/asincrono)
- Un cambiamento di qualche attributo
- Scorrere del tempo

o al verificarsi di una certa condizione

- Non e' necessario etichettare con tutto l'arco della transizione
- Una condizione verificata (scritta tra []) puo' attivarla

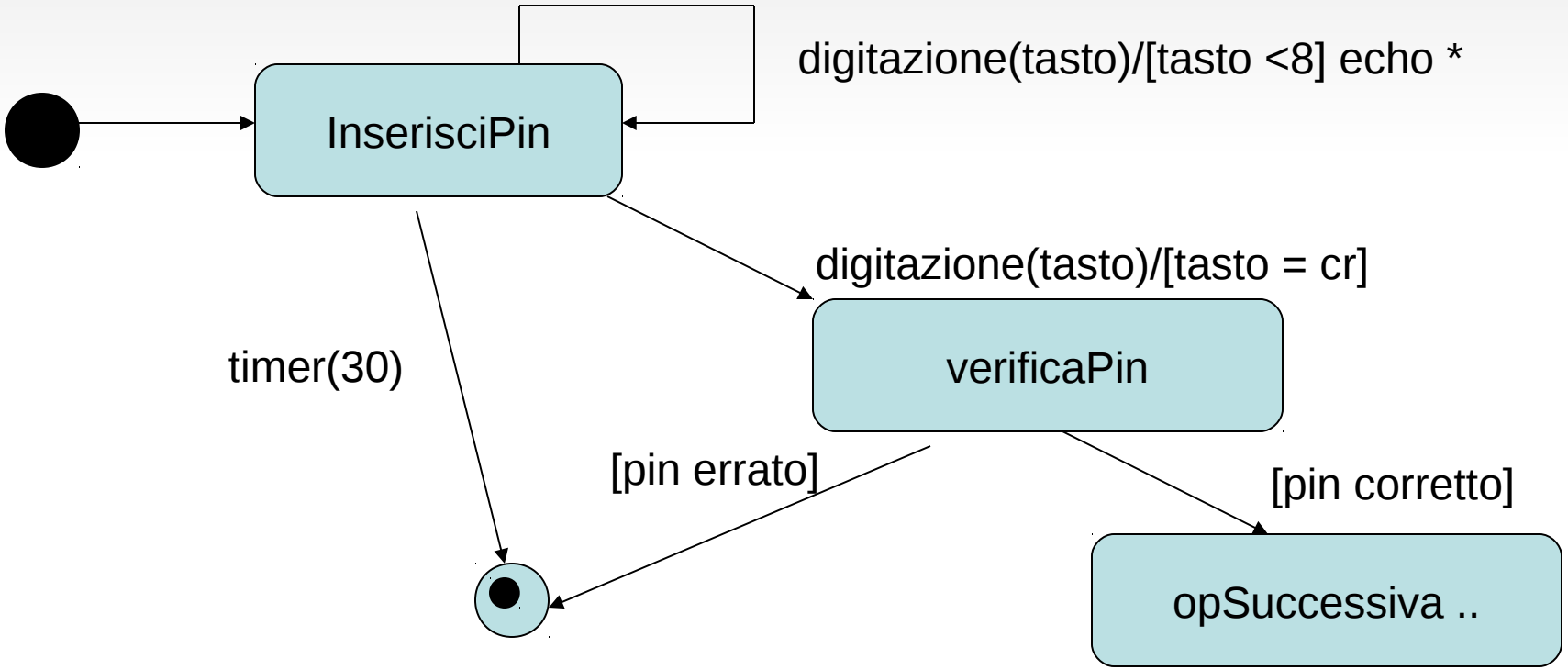
Esempio: affitto video



Transizioni interne ad uno stato

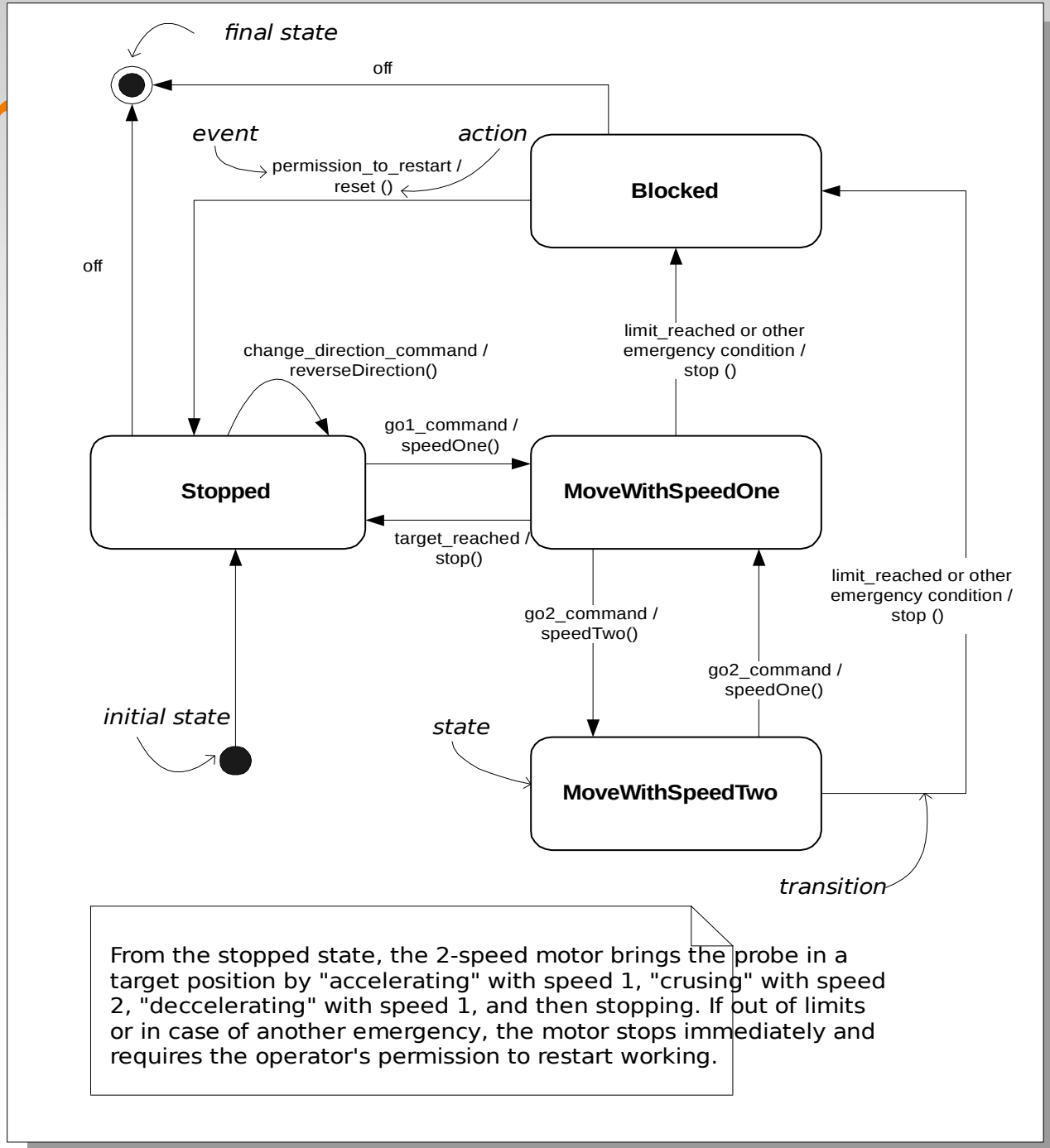
quando un evento non fa cambiare stato

a volte e' complicato specificare correttamente tutti i casi



Basic Statechar ts: Review

Example of statechart [SD, 2001] (Arlow..)



Modellare le attivita’

Il diagramma delle attivita’ descrive come sono organizzati I vari casi d'uso che costituiscono tutto il sistema

Puo’ essere usato per descrivere una procedura ad alto livello

Descrive i passi della computazione

- Ogni stato rappresenta l’esecuzione di caso d'uso
- Descrive quali operazioni possono esser **parallele** e quali **sequenziali** o comunque ordinate
- Gli archi rappresentano le transizioni da uno stato all’altro
- I nomi hanno significato dal punto di vista del sistema e non dell’ attore
- Le attivita’ non sono associate ad un oggetto specifico

Diagramma di attivita’

Mostra le transizioni tra le attivita’

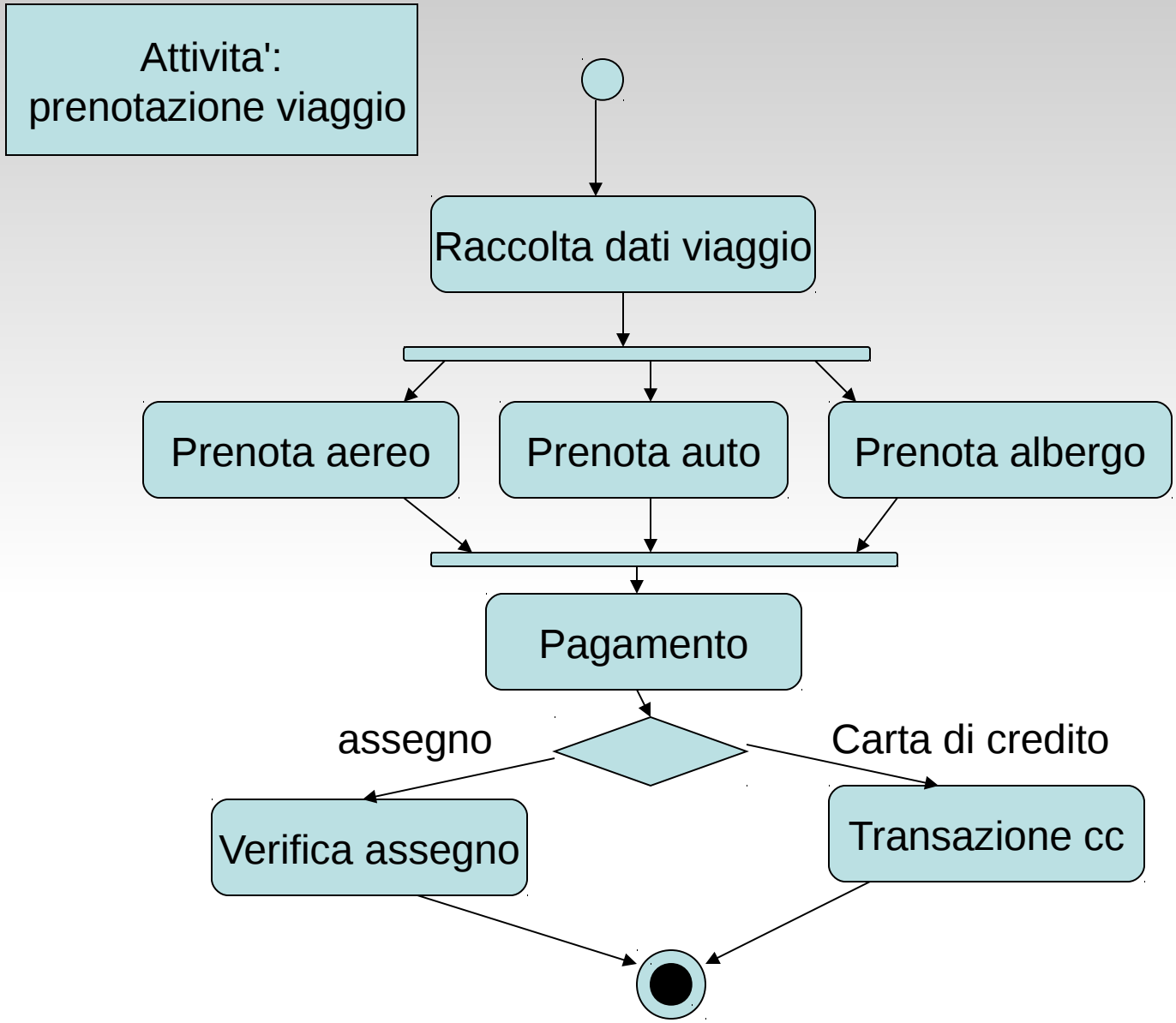
Rappresentate con rettangoli arrotondati

Lo stato iniziale e’ un disco colorato, quello finale ha un cerchio aggiuntivo

Rombi rappresentano esecuzioni alternative

Barre rappresentano esecuzioni parallele

Simile ad un flowchart


Attivita':
prenotazione viaggio
Raccolta dati viaggio
Prenota aereo
Prenota auto
Prenota albergo
Pagamento
assegno
Carta di credito
Verifica assegno
Transazione cc

Quando usarli

devono essere semplici, al corretto livello di astrazione

servono per evidenziare i parallelismi delle principali attivita’

utili per comunicare con personale non tecnico piu' che per dettagli implementativi

Non sono in genere delle descrizioni tramite flow chart di un caso d'uso

utili per chiarirsi le idee

quando si entra nei dettagli, spesso il codice sorgente e’ l’unico modo

- problema generale a tutti i diagrammi dinamici

altro uso : modellare i workflow (fine del corso)

I nodi di parallelismo sono sempre a coppie:

- start paralellismo , attivita', end parallelismo

A chi assegnare le operazioni?

GRASP

- General Responsibility Assignment Software Patterns

  http://en.wikipedia.org/wiki/GRASP_%28object-oriented_design%29

Schemi generali per assegnare le rersponsabilita' (operazioni) alle varie classi

GRASP

Creator: A crea un oggetto B

Ad esempio se:

- A contiene o aggrega B
- A registra B
- A usa parecchio B
- A possiede i dati per inizializzare B

Es. scacchiera e caselle

GRASP

Information Expert: A conosce le informazioni su B

Es, data una casella della scacchiera, cercarla e referenziarla

GRASP

Low coupling: come ridurre impatto modifiche

Stabilire tra gli oggetti il minimo numero di relazioni, per ridurre le modifiche in caso di cambiamenti

Expert supporta Low coupling

GRASP

Le classi di tipo Controller

- Il primo oggetto fuori dalla UI che riceve e coordina una richiesta di operazione
- Trasforma gli eventi UI in operazioni della logica applicativa
- Puo' essere un controller generale o di un singolo caso d'uso

GRASP

High Coesion

Per mantenere gli oggetti focalizzati, comprensibili , sostenere basso accoppiamento

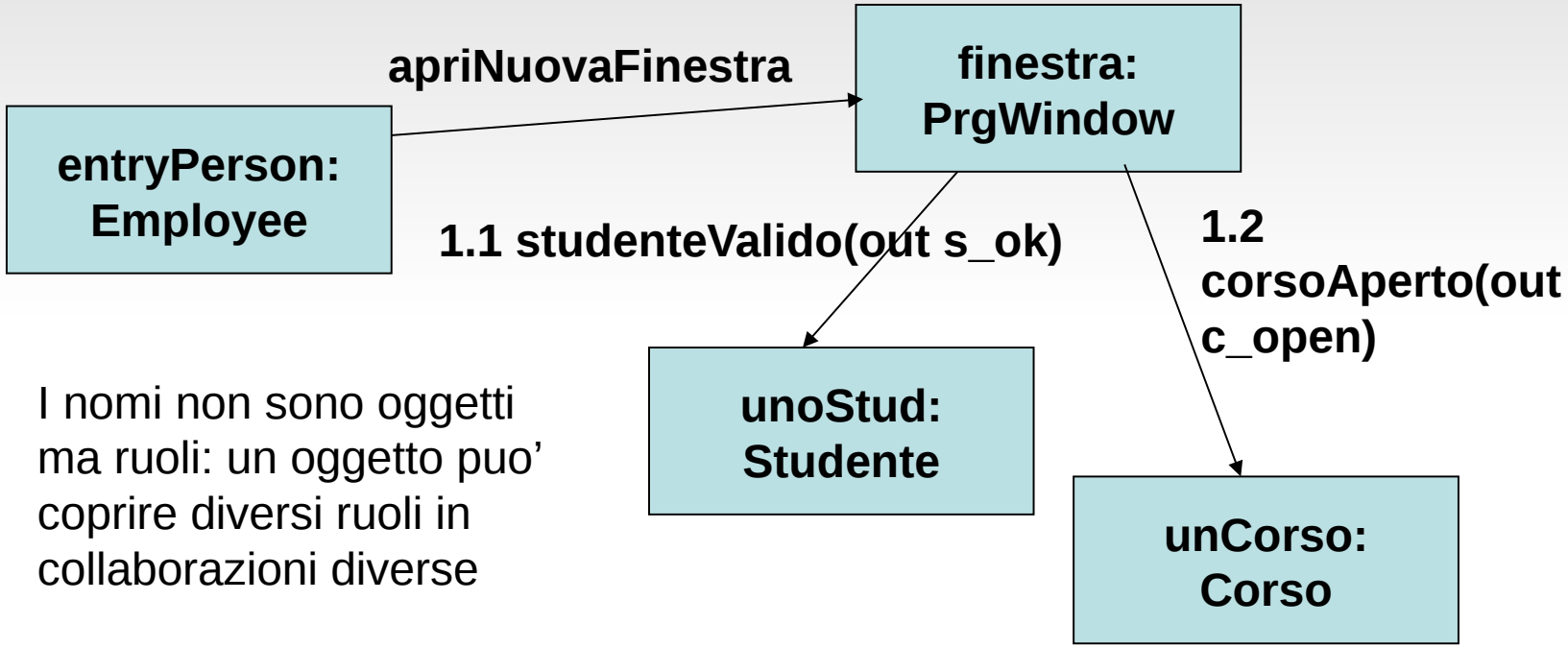
Spezzare a volte un oggetto complesso in pezzi piu' coesi (es messaggio)

In genere ce ne si accorge dopo averlo scritto

Altri pattern trattati altrove nel corso:

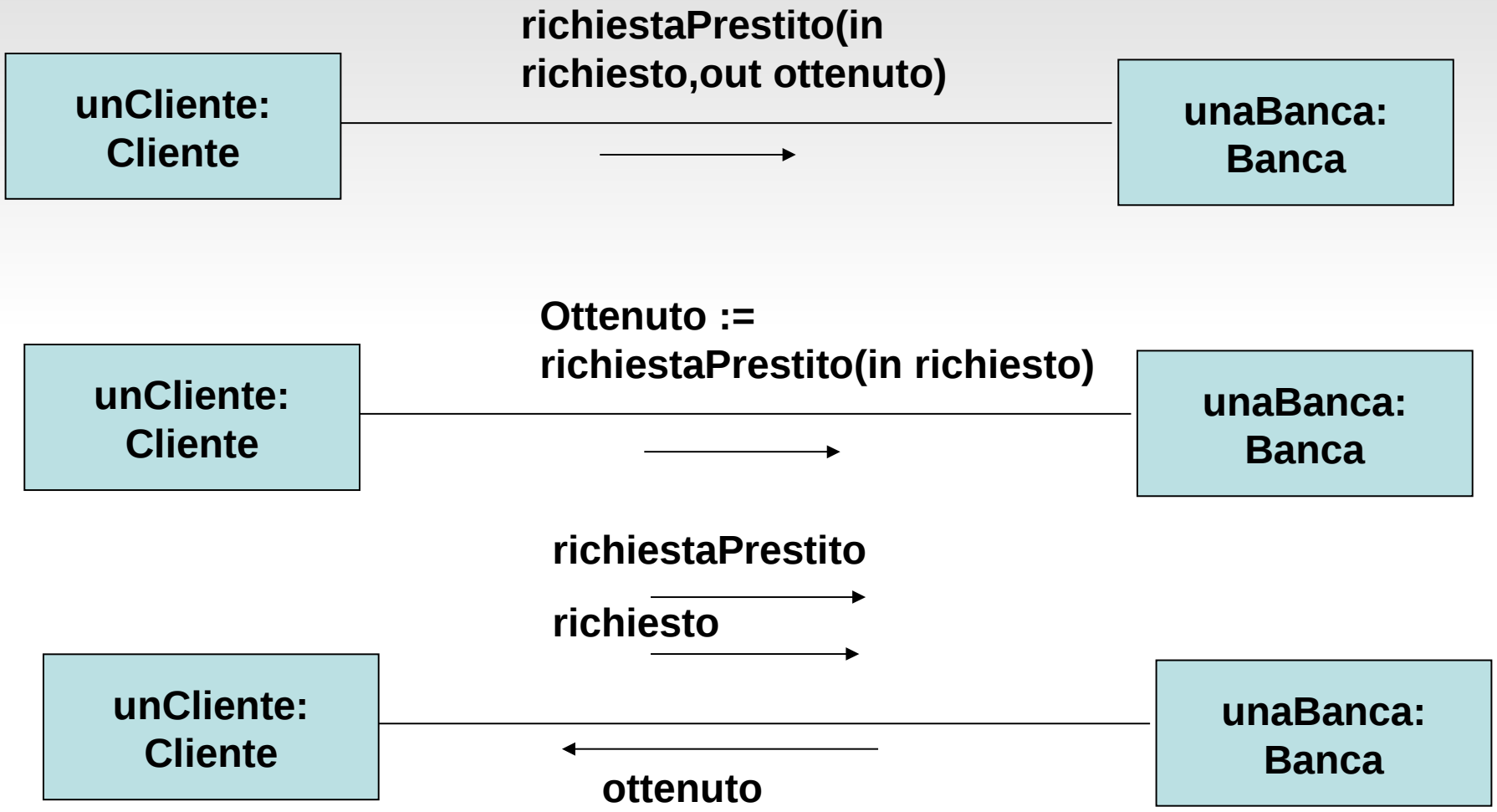
Polimorfismo, Pura fabbricazione, Indirezione, Variazioni protette

Il Diagramma di collaborazione



I nomi non sono oggetti ma ruoli: un oggetto puo' coprire diversi ruoli in collaborazioni diverse

notazioni alternative per i messaggi



Tipi di messaggio

Costruttori/distruttori

Lettura (dati attuali)

Aggiornamento (dati gia’ noti)

Collaborazione (orientato al futuro)

Messaggi iterativi

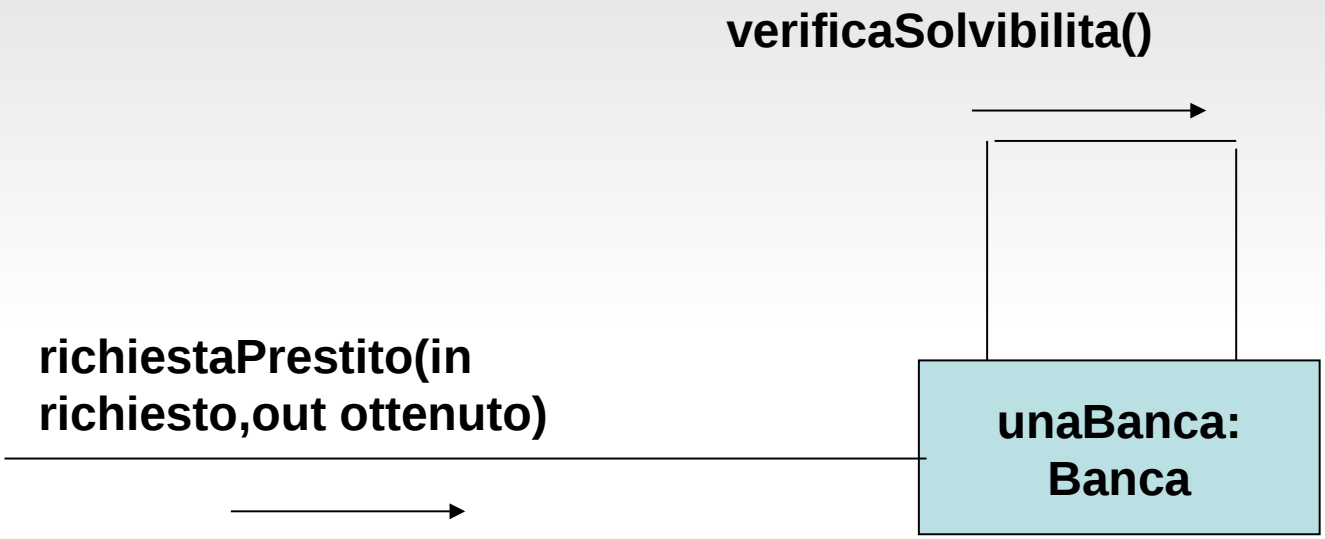
Mandati ad un insieme di oggetti

In UML denotati con “*”

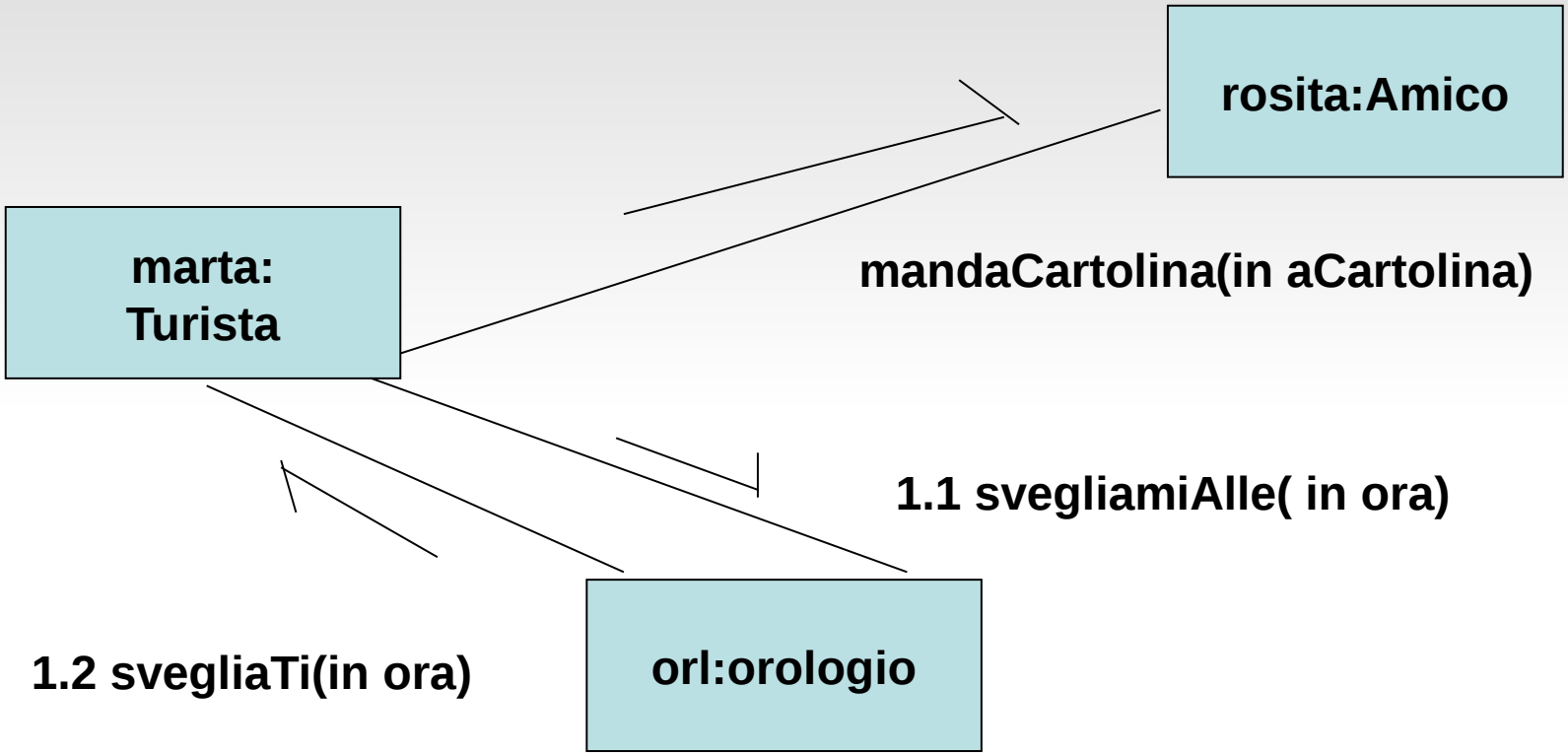
Gli oggetti destinatari sono contenuti in un oggetto di tipo ad es List.

List ha come parametro il tipo degli oggetti contenuti : List(Date)

Messaggi a self



Messaggi asincroni



Diagrammi di sequenza e collaborazione

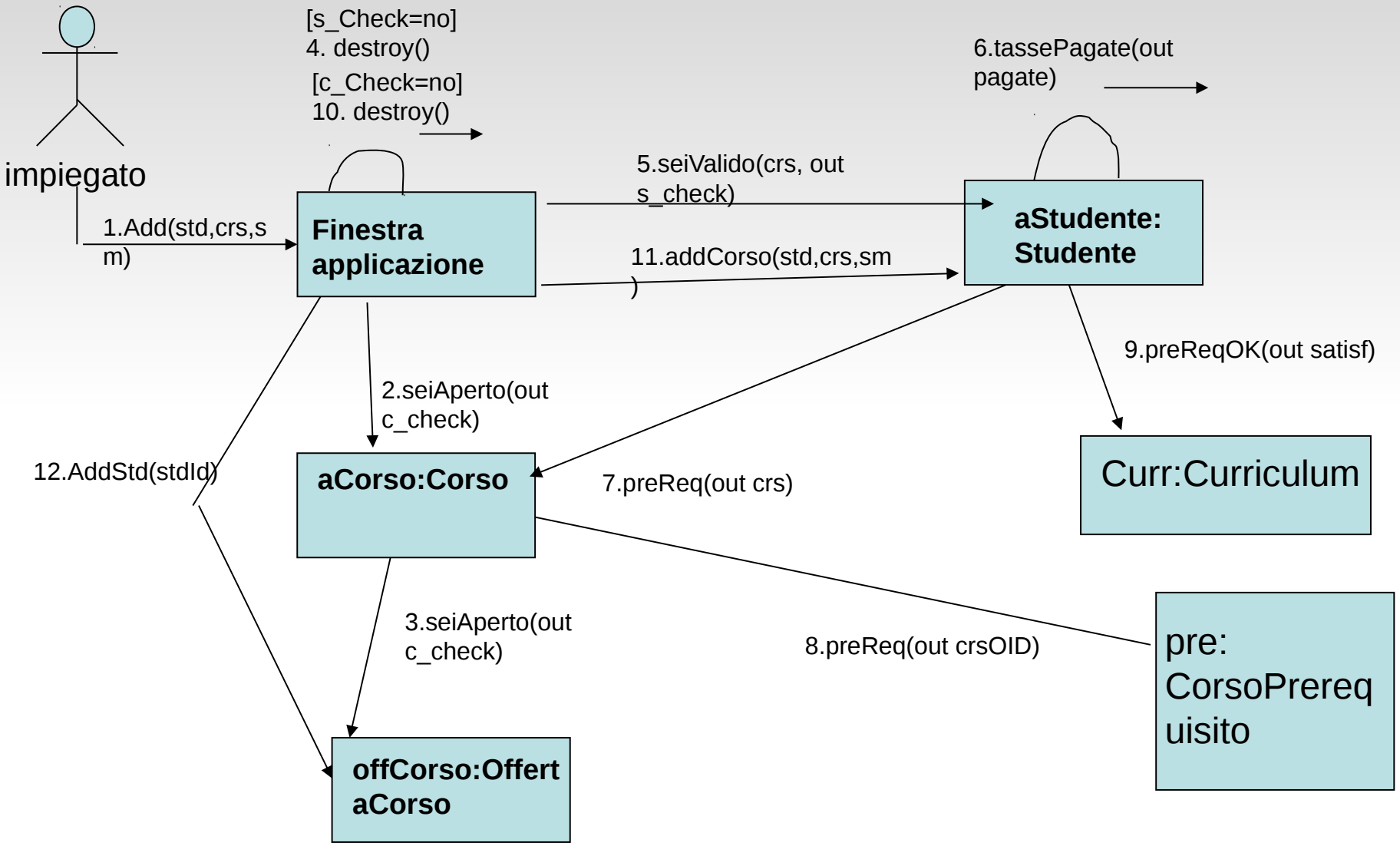
Il diagramma di sequenza privilegia l'aspetto temporale dello scambio di messaggi tra oggetti

- Scomodo per rappresentare cammini alternativi: meglio il diagramma di attivita'
- Pesante da utilizzare quando si mostra l'interazione tra un gran numero di oggetti

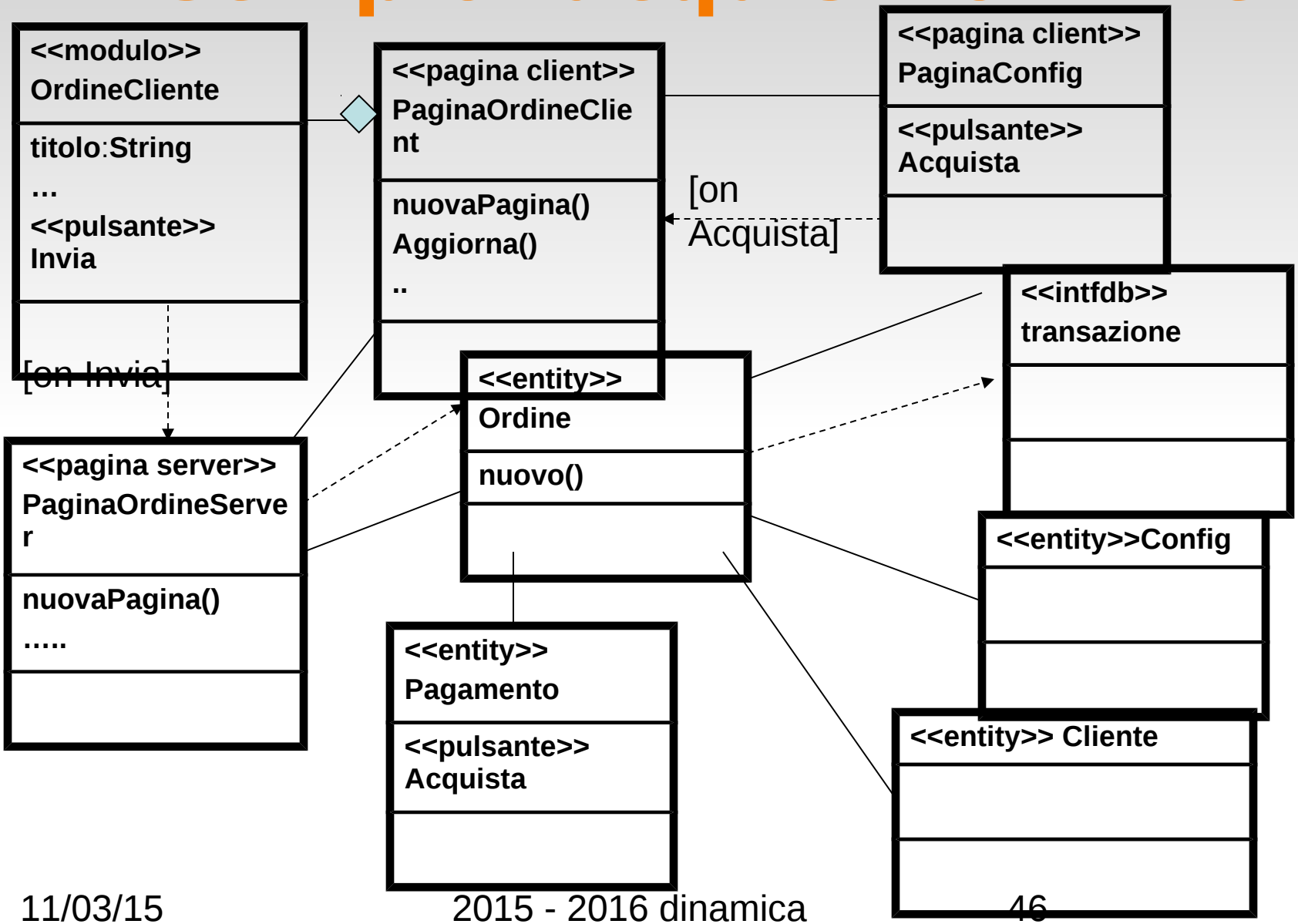
Il diagramma di collaborazione (interazione) mostra le stesse informazioni. Mostra staticamente le linee di interazione tra gli oggetti attraverso cui si scambiano i messaggi

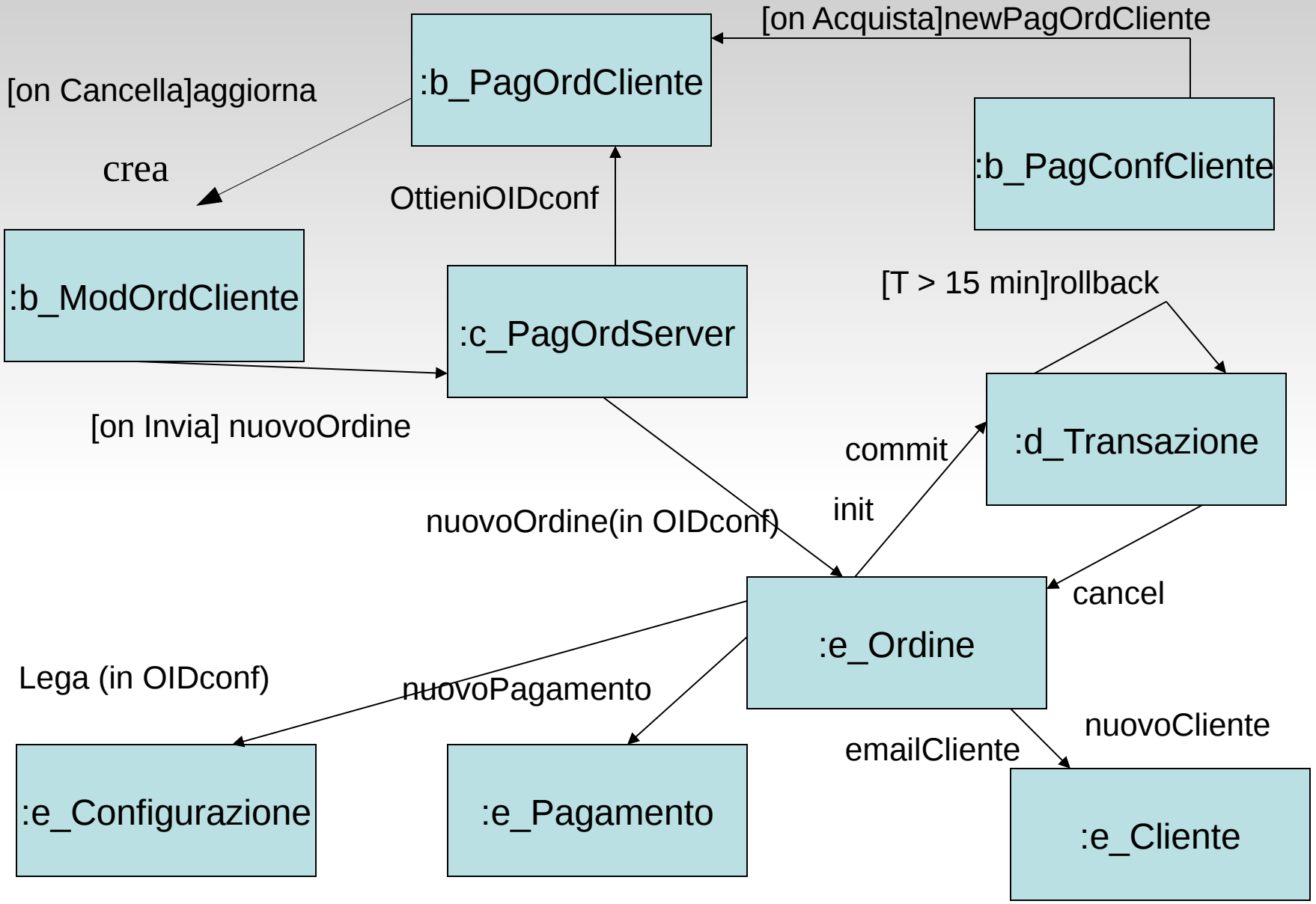
- Permette di mostrare piu' oggetti nello stesso spazio
- Il messaggio puo' essere specificato ed annotato meglio

Comportamento in collaborazione

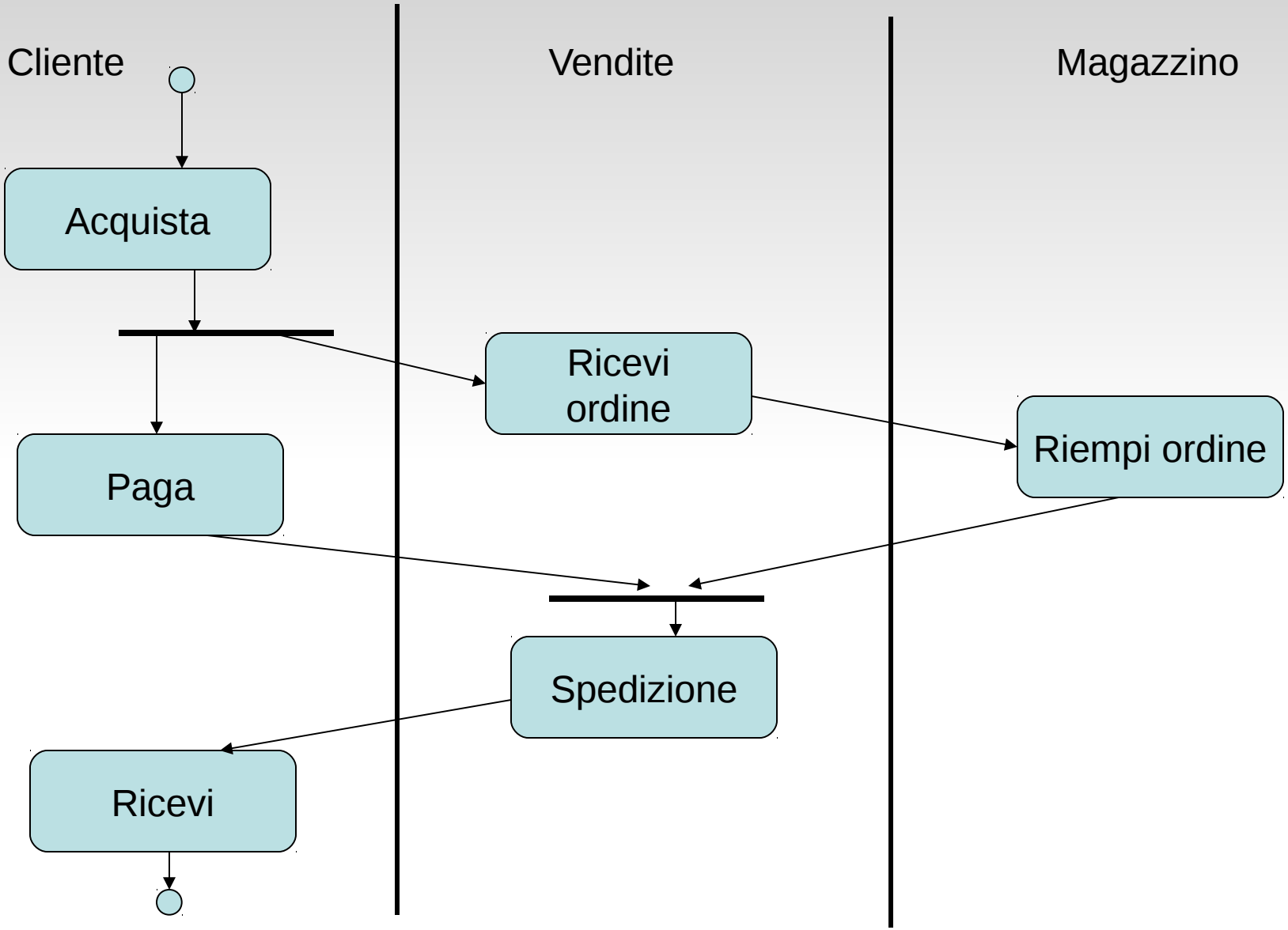


Esempio :acquisti on line

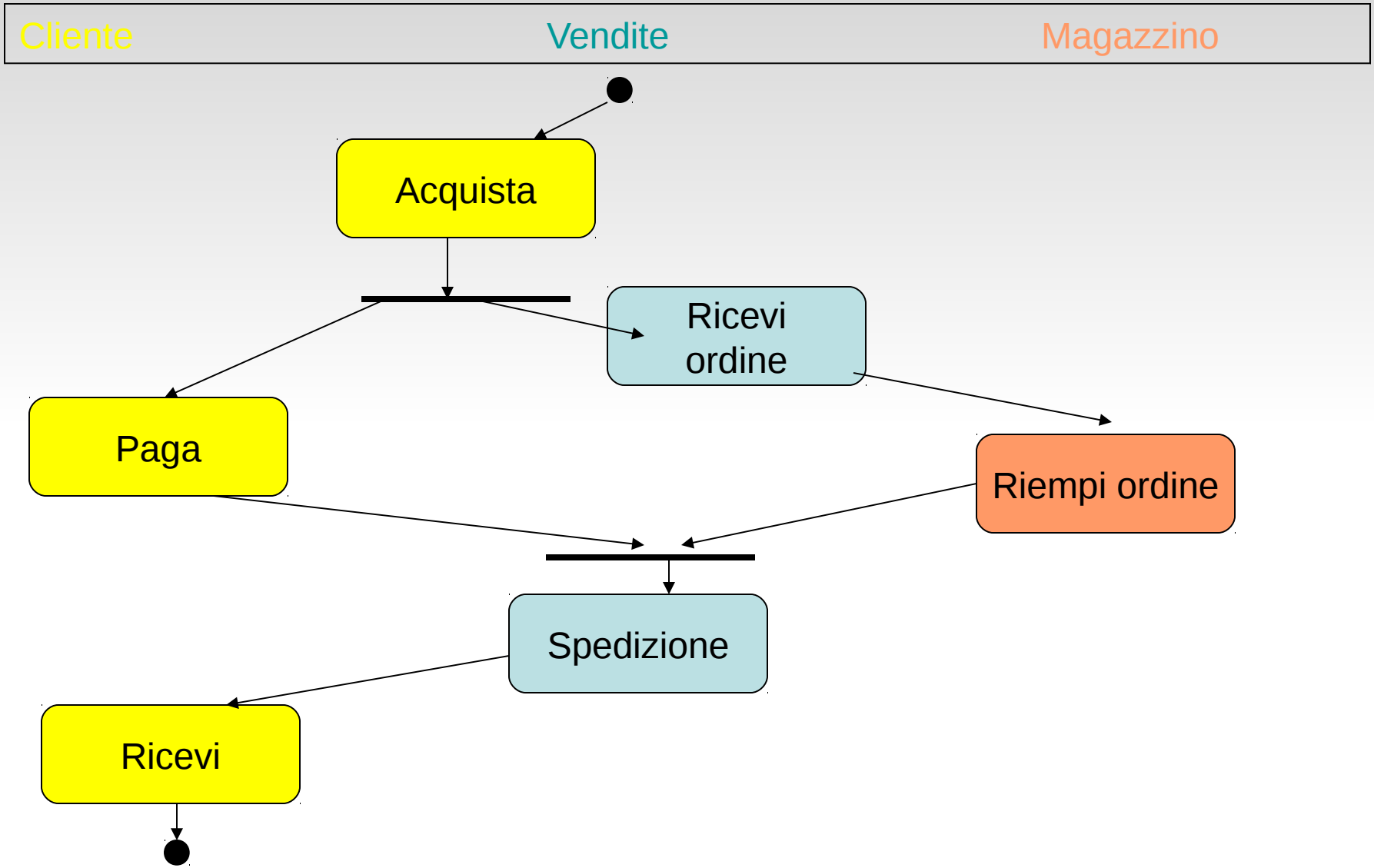



[on Acquista]newPagOrdCliente
:b_PagOrdCliente
[on Cancella]aggiorna
crea
:b_PagConfCliente
OttieniOIDconf
:b_ModOrdCliente
:c_PagOrdServer
[T > 15 min]rollback
[on Invia] nuovoOrdine
commit
:d_Transazione
init
nuovoOrdine(in OIDconf)
cancel
:e_Ordine
Lega (in OIDconf)
nuovoPagamento
nuovoCliente
emailCliente
:e_Configurazione
:e_Pagamento
:e_Cliente

Swimlanes (corsie) in diagramma di attivita'



Colorazione delle attivita' in base agli oggetti



le varie fasi

1. Trasformazioni
   - Creare *scenari e casi d'uso*
     - parlare al cliente, osservare , leggere documenti etc
2. struttura del sistema
   - creare diagrammi di classe
     - identificare oggetti relazioni molteplicita'
     - attributi, operazioni
3. struttura di controllo
   - diagrammi di sequenza
     - mittenti e riceventi, parametri operazioni
     - dipendenze e concorrenze tra eventi
   - diagramma di stato
     - solo per oggetti interessanti
       - diagramma attivita'
       - diagramma collaborazione

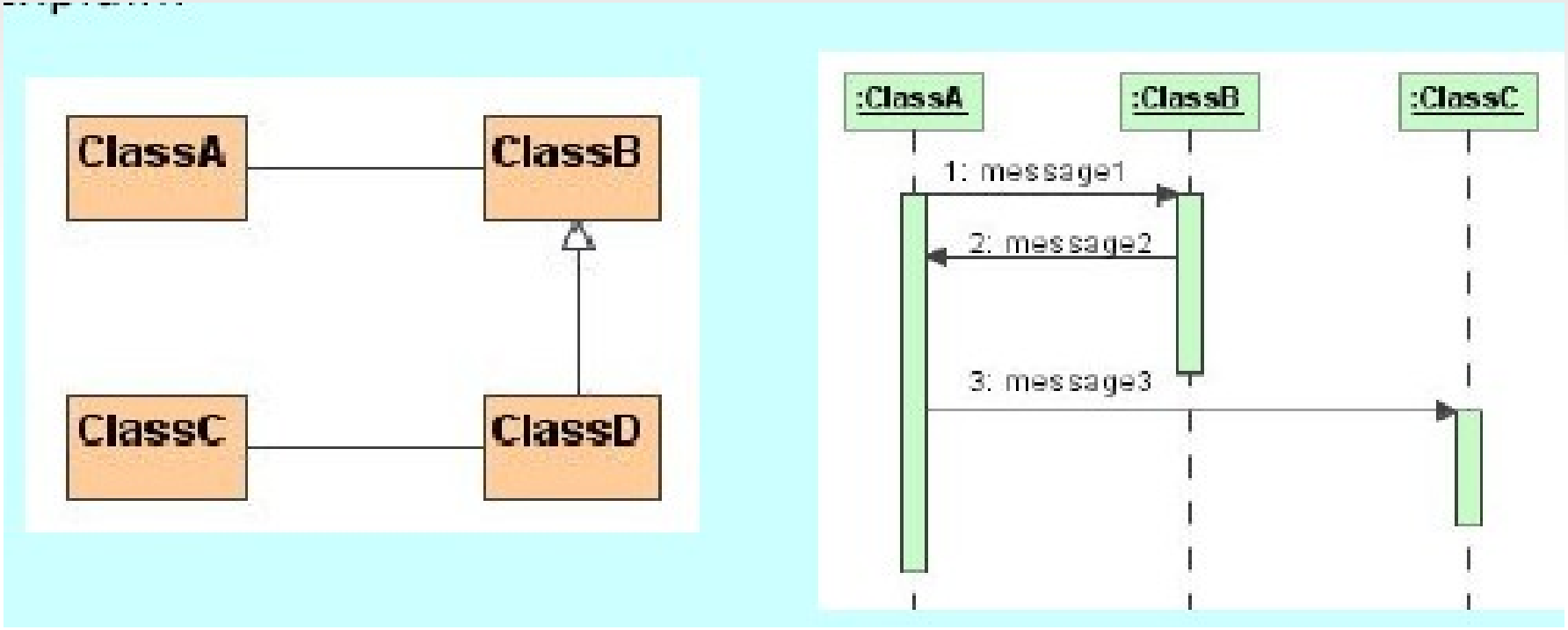
Modello funzionale

modello oggetti

modello dinamico

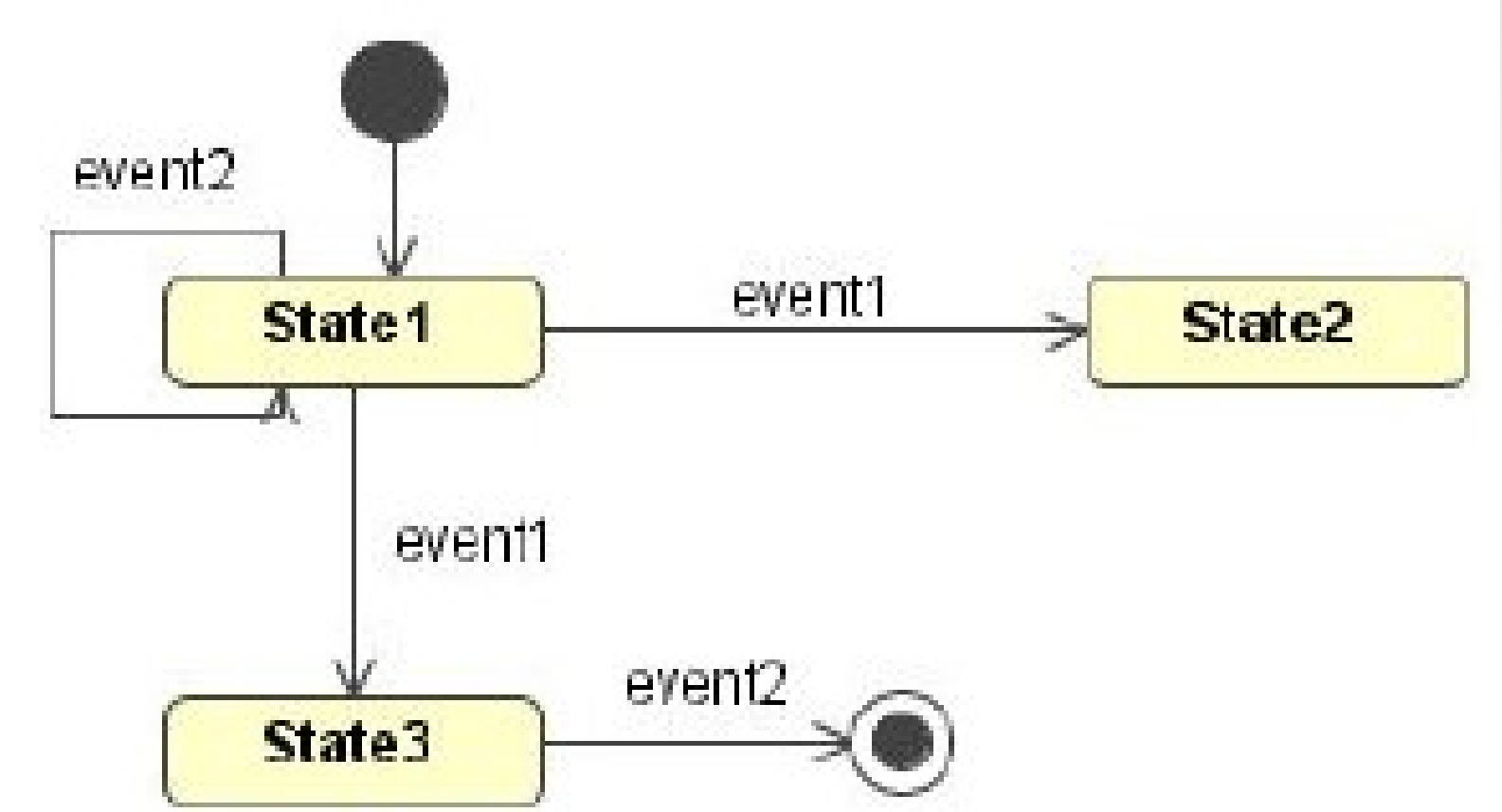
domanda

problema nel seguente diagramma di sequenza?

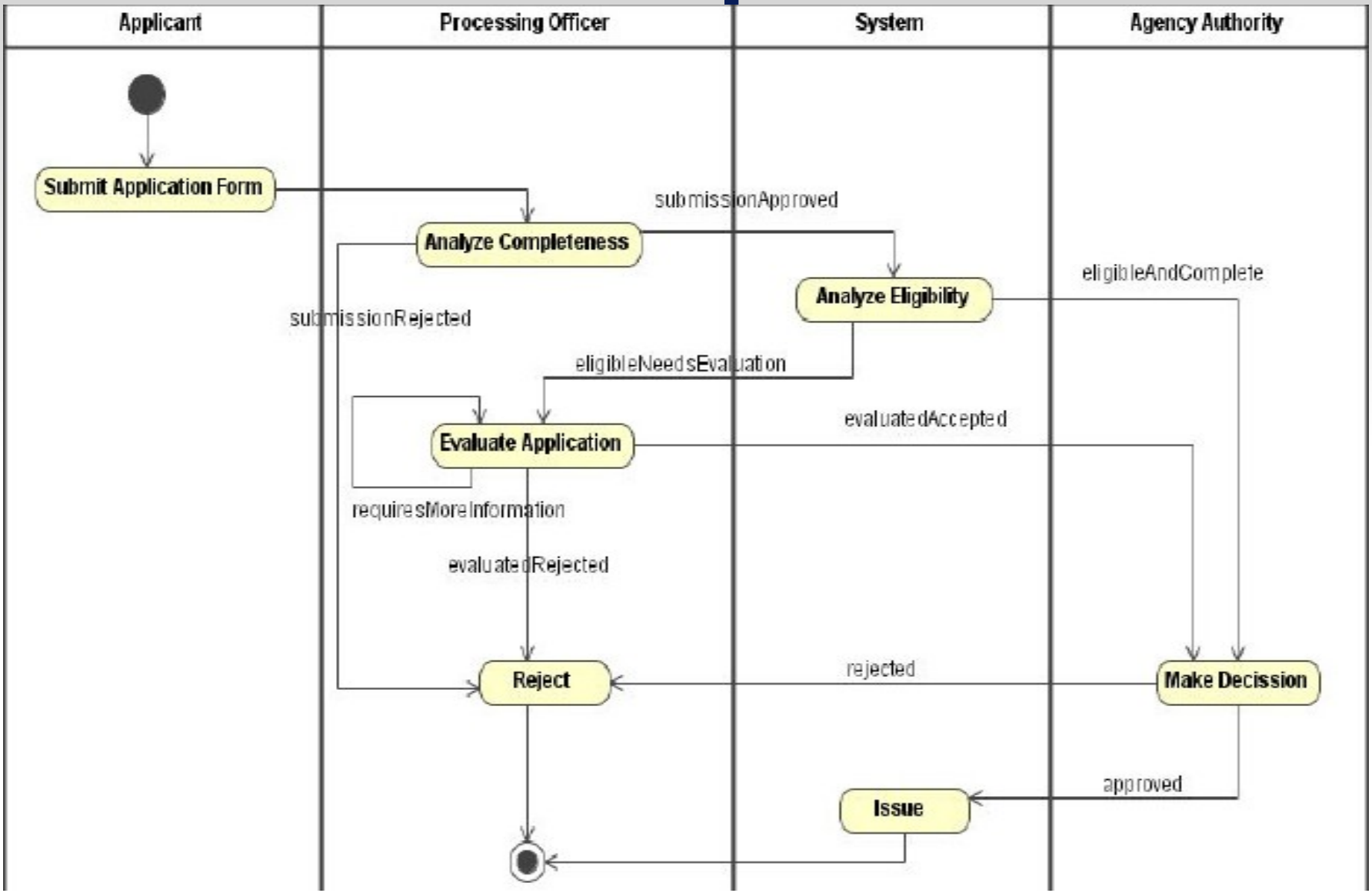


domanda

errore nel seguente diagramma di stato?



esempio


Applicant
Processing Officer
System
Agency Authority
Submit Application Form
Analyze Completeness
submissionApproved
Analyze Eligibility
eligibleAndComplete
submissionRejected
eligibleNeedsEvaluation
Evaluate Application
evaluatedAccepted
requiresMoreInformation
evaluatedRejected
Reject
rejected
Make Decission
approved
Issue

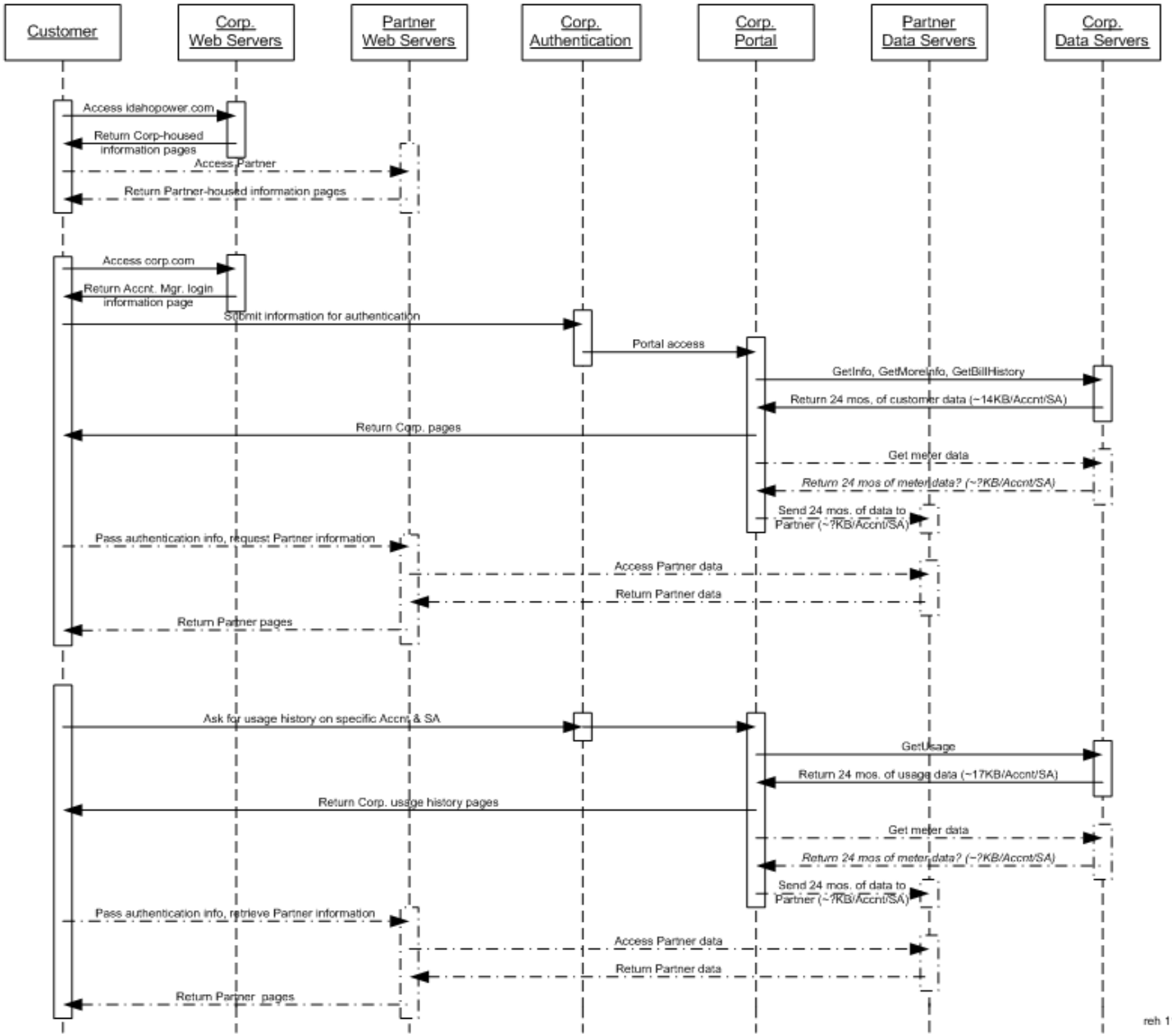
Customer Care Interaction

Customer Scenario

Part A:
Customer accesses corp.com site to get public, corporate-related information -- some from Corp., some from Partner.

Part B:
Customer accesses corp.com to retrieve account-based use information. Account Manager login info. is submitted. Customer is authenticated. Portal application automatically retrieves core customer account data for Portal use. Get meter data as appropriate. Initiate data transfer to Partner. Display core customer account pages from Corp. site. Display ? pages from Partner site.

Retrieve usage history for a specific account and service agreement, as requested by customer. Get meter data as appropriate. Initiate data transfer to Partner. Display usage history from Corp. site. Display ? pages from Partner site.



reh 11/04

devo usare tutti questi diagrammi?

i vari tipi di diagrammi sono utili per dare una descrizione **sintetica** del comportamento del sistema

servono a comunicare tra gli implementatori e gli altri gruppi interessati e ad introdurre al progetto nuovo personale

se fossero un modello completo:

- tenderebbero a diventare complicati
- avrebbero la stessa quantita' di informazioni del codice, scritta a volte in modo involuto

inoltre:

- se non sono **sempre** aggiornati, sono **fuorvianti** →
- diventano un costo aggiuntivo nella manutenzione

riassumendo

La specifica dello stato descrive il sistema attraverso classi attributi e relazioni

- Diversi metodi di scoperta delle classi
- I diagrammi di classe mostrano le classi e le loro relazioni: associazioni,aggregazioni,generalizzazioni

La specifica del comportamento descrive il sistema dal punto di vista funzionale

- Il diagramma dei casi d'uso fornisce una visualizzazione semplice cui e' associata una descrizione testuale
- diagrammi dinamici sono quelli di attivita' e di sequenza

La specifica dei cambiamenti di stato descrive il comportamento dinamico del sistema

- Si utilizzano i diagrammi di stato

esercizio

diagramma attivita’ (workflow) ristrutturazione casa