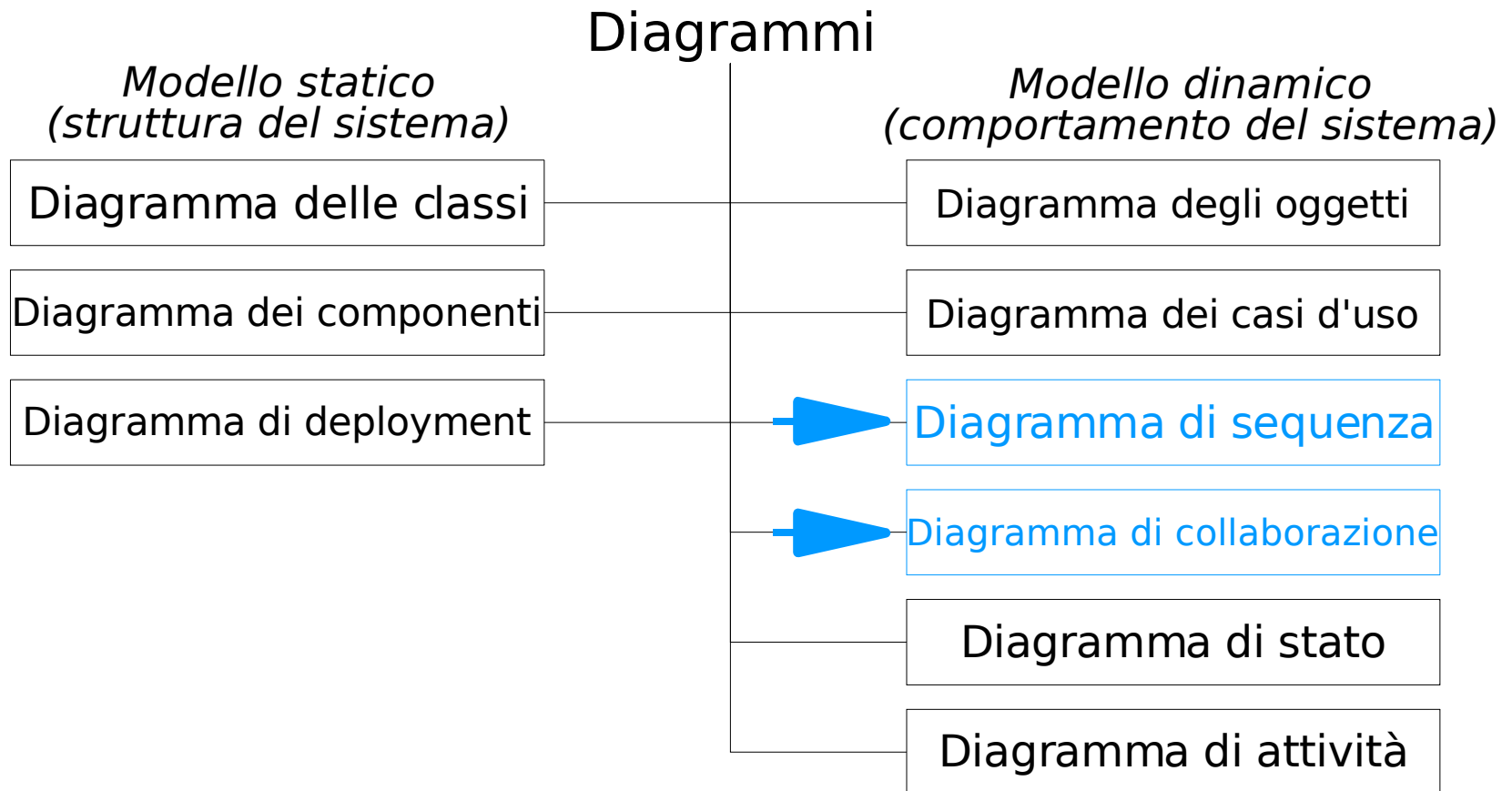


Elementi di UML (5): Diagrammi di interazione

Università degli Studi di Bologna
Facoltà di Scienze MM. FF. NN.
Corso di Laurea in Scienze di Internet
Anno Accademico 2004-2005

Laboratorio di Sistemi e Processi Organizzativi

Diagrammi di Interazione



Realizzazioni di casi d'uso (1/2)

- Dopo l'individuazione delle classi, un aspetto cruciale dell'analisi consiste nell'individuare le realizzazioni di casi d'uso
- Si trasforma un caso d'uso in un insieme di diagrammi delle classi e di diagrammi di interazione
- I casi d'uso specificano i requisiti funzionali
- I diagrammi delle classi più i diagrammi di interazione costituiscono una specifica ad alto livello del sistema

Realizzazioni di casi d'uso (2/2)



Diagrammi di interazione (1/4)

- Diagrammi delle classi d'analisi
 - Mostrano le classi di analisi che inetragiscono per realizzare il caso d'uso
- Diagrammi di interazione
 - Mostrano le collaborazioni e le interazioni tra istanze specifiche che realizzano il caso d'uso
 - Sono delle istanze del sistema in esecuzione

Diagrammi di interazione (2/4)

- Diagrammi di interazione
 - Esplicitano come oggetti specifici delle classi collaborano e interagiscono tra di loro per realizzare una parte, o tutto, del comportamento di un caso d'uso
 - Ne esistono due tipi:
 - Diagrammi di collaborazione
 - Diagrammi di sequenza

Diagrammi di interazione (3/4)

- Diagrammi di collaborazione:
 - Enfatizzano le relazioni strutturali tra gli oggetti e sono utilissimi per l'analisi, soprattutto per creare una bozza di una collaborazione tra oggetti
- Diagrammi di sequenza:
 - Enfatizzano la sequenza temporale degli scambi di messaggi tra diversi oggetti.

Diagrammi di interazione (4/4)

- I diagrammi di collaborazione e i diagrammi di sequenza sono isomorfi.
- Entrambi possiedono due forme:
 - *Forma descrittore*: descrive le collaborazioni e le interazioni tra i **ruoli** che le istanze dei classificatori possono svolgere nel sistema
 - *Forma istanza*: descrive le collaborazioni e le interazioni tra le effettive istanze dei classificatori

Collaborazioni e interazioni

- Una **collaborazione** descrive un insieme statico di relazioni tra istanze, e i ruoli che queste istanze svolgono in queste relazioni
- Un'**interazione** descrive come le istanze interagiscano tra loro in modo dinamico; descrive i messaggi che le istanze si scambiano
- Un **ruolo** è un particolare comportamento o una modalità di utilizzo di qualcosa

Diagrammi di Collaborazione

- In *forma descrittore*:
 - offre una vista molto generica della collaborazione, specificando i ruoli svolti dalle istanze e le relazioni tra questi
 - non include la presenza di istanze
- In *forma istanza*:
 - più concreto
 - mostra le effettive istanze di classificatori e i collegamenti tra queste istanze

Forma descrittore vs istanza

- Non si possono mostrare ramificazioni e iterazioni sui diagrammi di collaborazione in forma descrittore
- Perché?
- Perché la forma descrittore rappresenta esclusivamente ruoli e non include istanze

Esempio forma descrittore

Diagramma delle classi

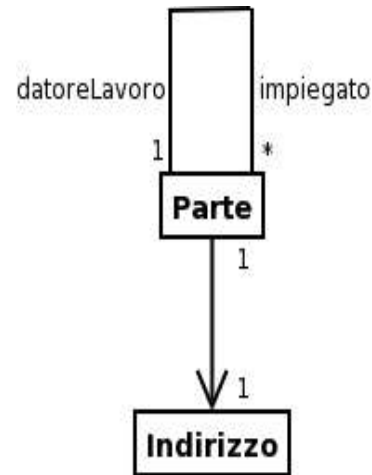
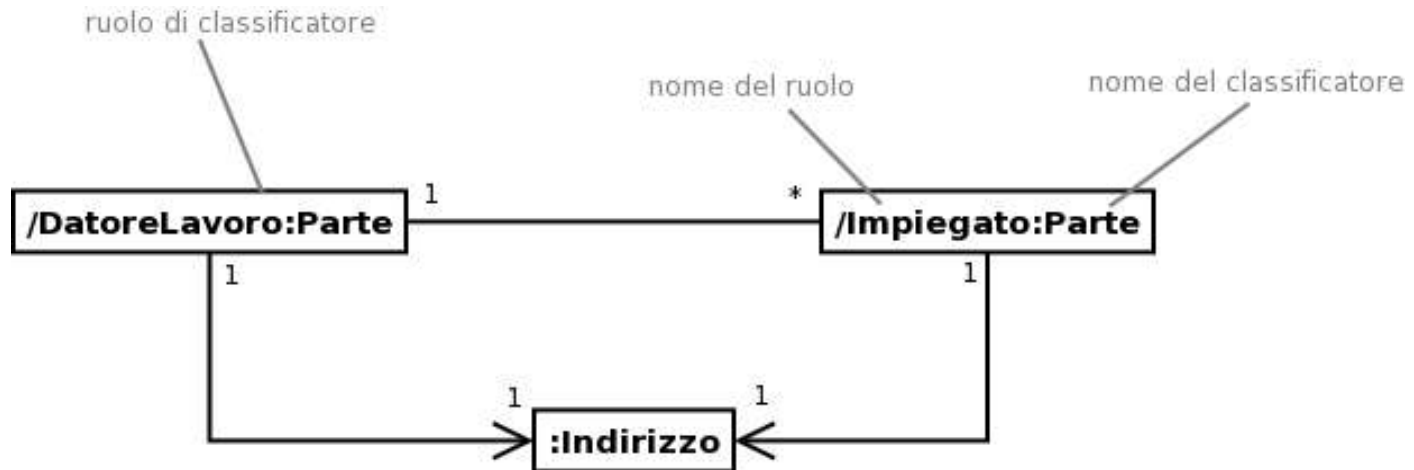


Diagramma di collaborazione in forma descrittore:



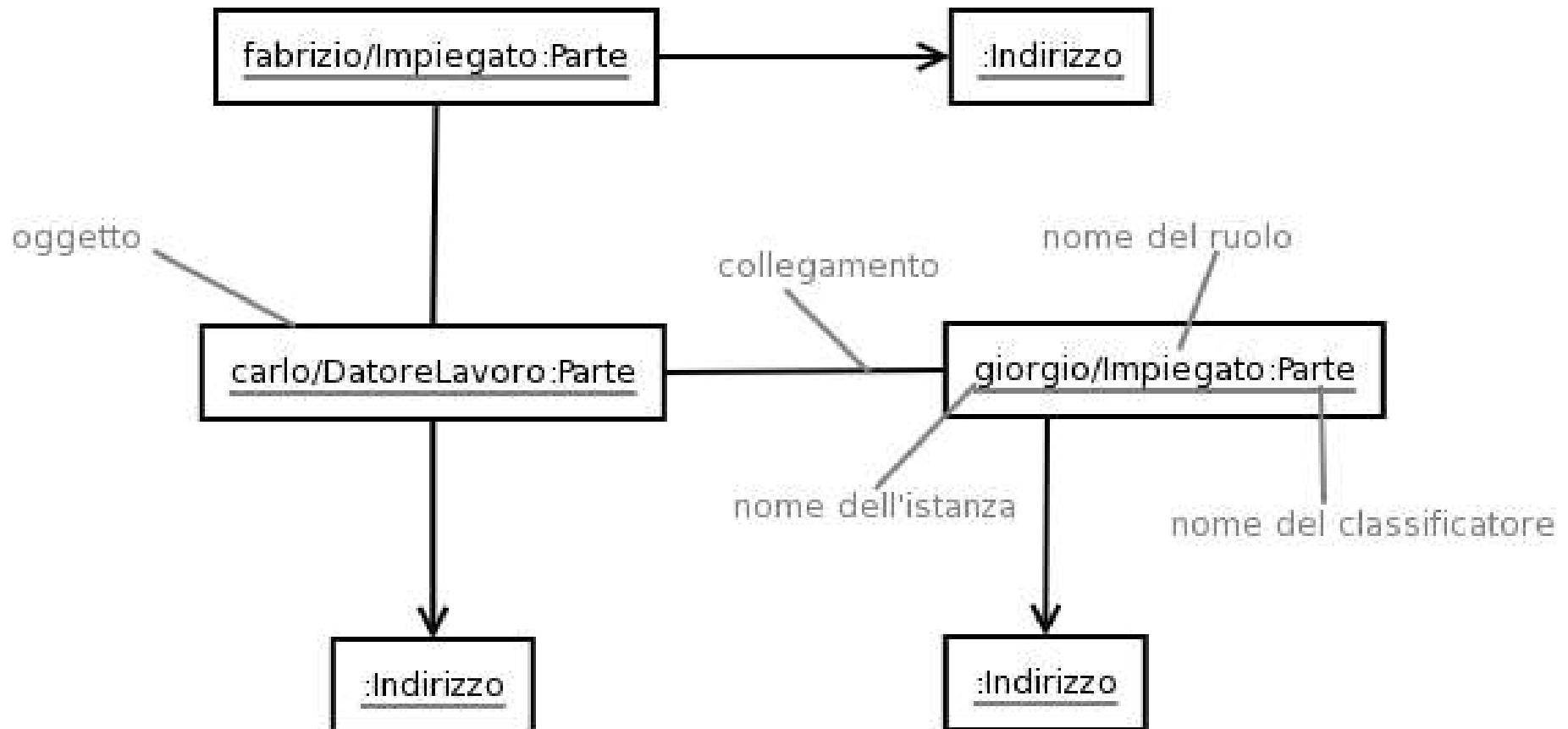
Nomi dei ruoli

- I nomi dei ruoli di classificatore hanno la forma:

`/NomeRuolo:NomeClassificatore`

- Il nome del ruolo deve sempre iniziare con uno slash, mentre il nome del classificatore deve sempre essere preceduto dai due punti
- Entrambi sono facoltativi, ma ne deve essere specificato almeno uno
- Come potremmo modellare in UML questo requisito?

Esempio forma istanza



Nomi delle istanze

- I nomi delle istanze vengono indicati come segue:

nomeIstanza/NomeRuoloClassificatore:NomeClassificatore

- Il nome del ruolo di classificatore inizia sempre con lo *slash*, il nome del classificatore inizia sempre con i due punti
- Tutte le parti sono facoltative, ma deve esserne presente almeno una

Esempio (1/3)

- Illustriamo come gli oggetti del sistema realizzino effettivamente il comportamento specificato nel caso d'uso

AggiungiCorso

Caso d'uso: AggiungiCorso
ID: UC2
Attori: Registrante
Precondizioni: 1. Il Registrante è stato autenticato dal sistema
Sequenza degli eventi: 1. Il Registrante seleziona “aggiungi corso” 2. Il sistema accetta il nome del nuovo corso 3. Il sistema crea il nuovo corso
Postcondizioni: 1. Un nuovo corso è stato aggiunto al sistema

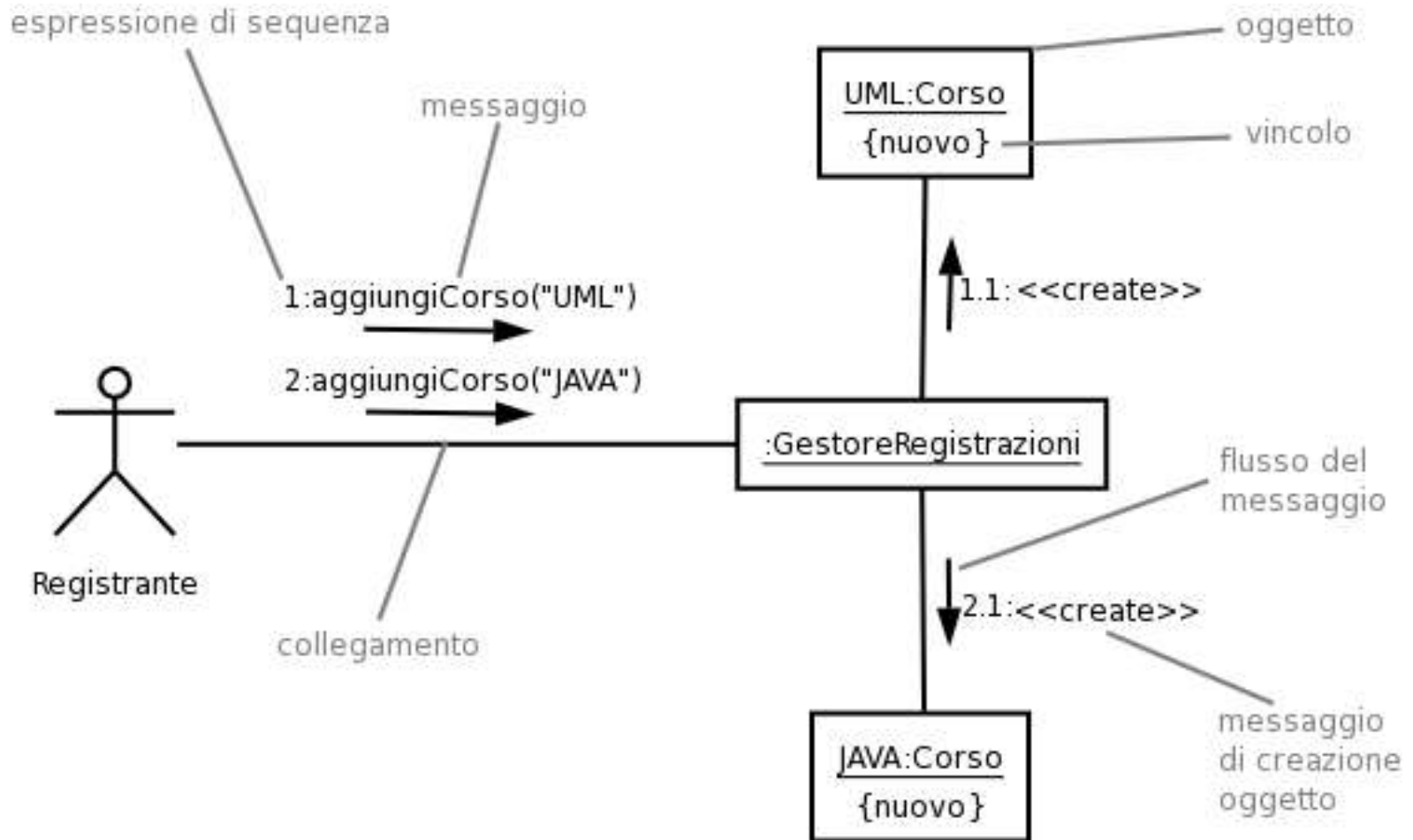
- Quali classi individuate?

Esempio (2/3)

- Classi:
 - Registrante (Attore): responsabile per il mantenimento dell'informazione sui corsi nel sistema di registrazione dei corsi
 - Corso (Classe): contiene i dettagli di un corso
 - GestoreRegistrazioni (Classe): responsabile per il mantenimento di una collezione di corsi
- Diagramma delle classi



Esempio (3/3)



Sintassi dei messaggi

- I messaggi devono essere indicati nel seguente modo:

EspressioneSequenza valoreRestituito:=nomeMessaggio(arg1, arg2,...)

- L'espressione di sequenza indica l'ordine in cui verrà inviato il messaggio
- Il valore restituito è il nome di una variabile a cui assegnare un eventuale valore, normalmente è un attributo dell'oggetto che invia il messaggio

Notazione per i messaggi



Chiamata di procedura

(*sincrona*): il mittente attende fintantoché il ricevente non ha finito.



Comunicazione asincrona: il mittente prosegue dopo l'invio del messaggio, senza attendere che il ricevente abbia finito.



Ritorno da una chiamata di procedura.

Vincoli per i messaggi

- {nuovo} L'istanza o il collegamento viene creato durante l'interazione
- {distrutto} L'istanza o il collegamento viene distrutto durante l'interazione
- {transiente} L'istanza o il collegamento viene creato e poi distrutto durante l'interazione

Focus di controllo

- Quando un oggetto sta eseguendo un'operazione si dice che esso ha il **focus di controllo**
- L'evolversi nel tempo della collaborazione provoca lo spostamento del focus di controllo tra i diversi oggetti modellati
- Questo viene detto **flusso del controllo**

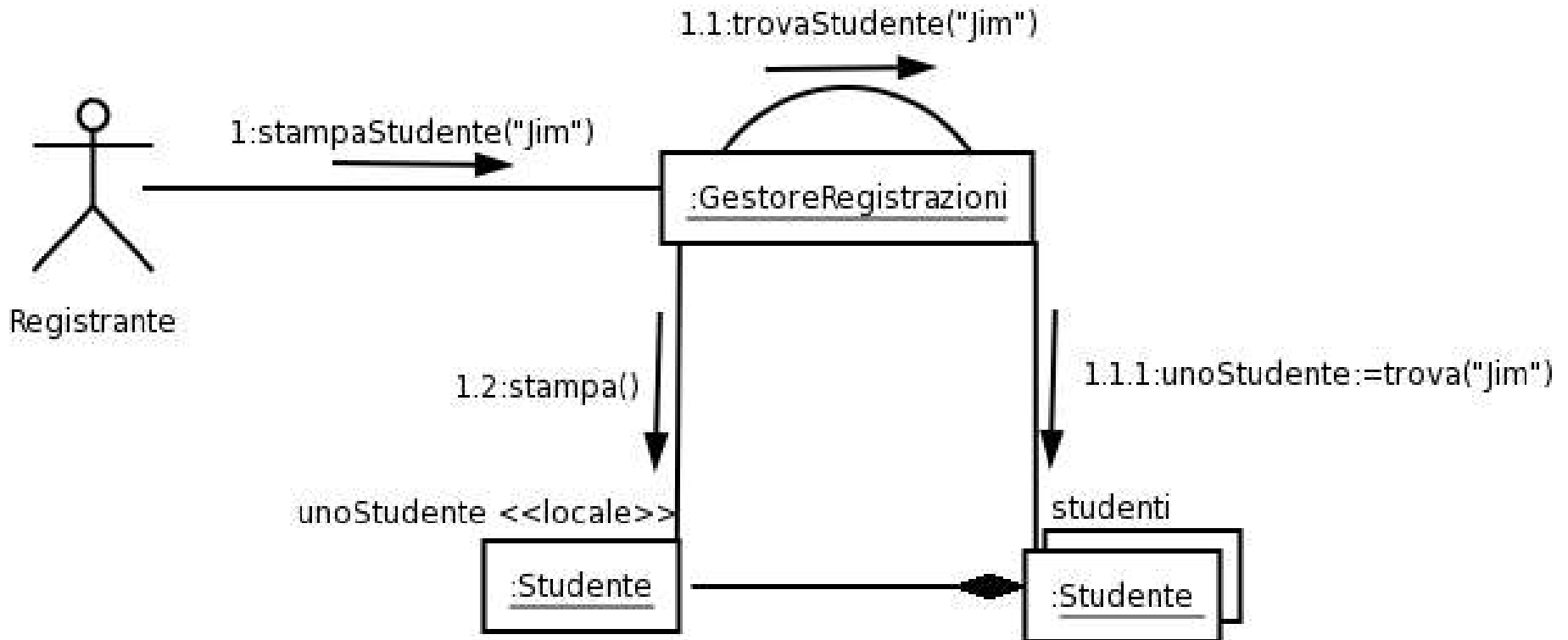
Multioggetti (1/2)

- Un **multioggetto** rappresenta un insieme di oggetti
- È un metodo per rappresentare le collezioni di oggetti nei diagrammi di collaborazione
- I messaggi inviati a un multioggetto vengono ricevuti e processati dall'insieme e non da un suo singolo oggetto
- *UML* non dice quali siano i metodi supportati da un mutlioggetto, perché in fase di *progettazione* ogni multioggetto sarà sostituito con un'istanza di una classe contenitore.

Multioggetti (2/2)

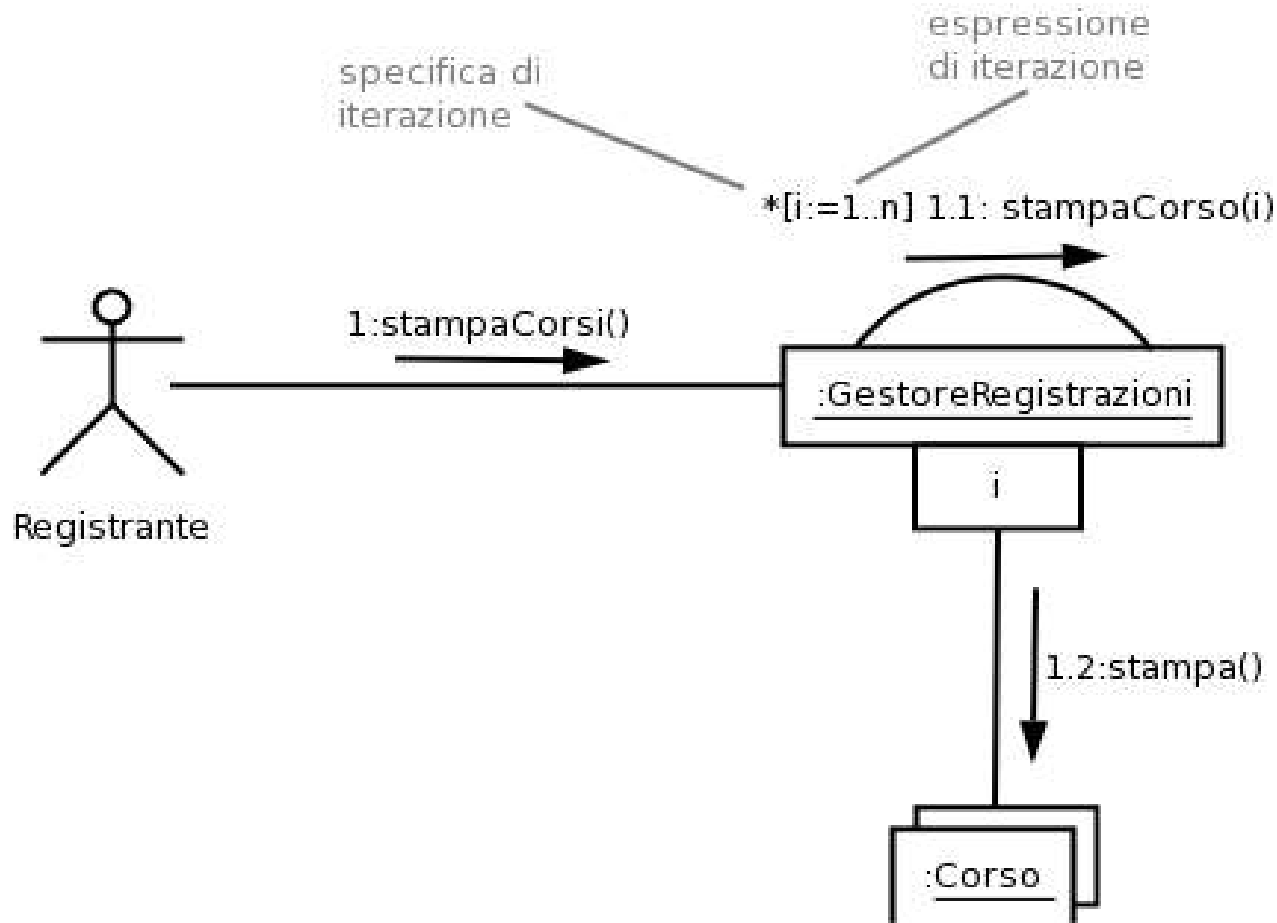
- Durante l'analisi si suppone che i multioggetti possano ricevere i seguenti messaggi:
 - Trova(idUnivoco): restituisce un oggetto specifico
 - include(unOggetto): restituisce *vero* se il multioggetto contiene unOggetto
 - Conteggio(): ritorna il numero di elmementi
- Per inviare un messaggio ad una specifica istanza presente nel multioggetto occorre
 - individuare l'istanza specifica e
 - quindi inviare il messaggio a quell'istanza.
 - Per farlo, quale relazione adoperereste?

Multioggetti: esempio



Iterazione (1/2)

Si indica un'iterazione premettendo al numero di sequenza l'indicatore di iterazione *



Iterazione (2/2)

- L'indicatore di iterazione * indica sempre l'elaborazione sequenziale delle istanze del multioggetto
- Per indicare che le istanze sono elaborate in parallelo si fa seguire l'indicatore * da due slash: *//

Ramificazioni e auto-delegazione

- Le **ramificazioni** possono essere modellate semplicemente aggiungendo delle condizioni davanti ai messaggi
- Il messaggio viene inviato solo se la condizione risulta vera
- L'**auto-delegazione** si ha quando un oggetto chiama sè stesso
- Oltre alle *operazioni pubbliche*, gli oggetti sfruttano *operazioni ausiliarie* **private**

Esercizio (1/2)

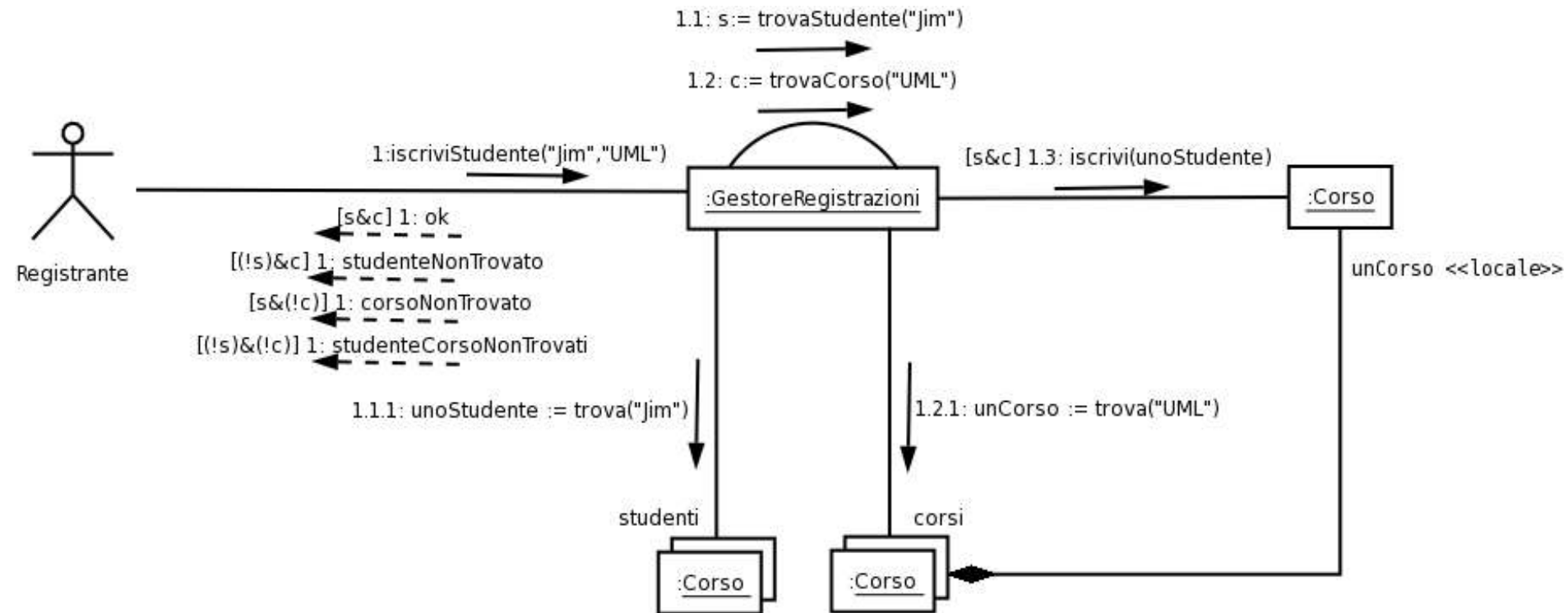
Scrivere il diagramma di collaborazione che realizza il caso d'uso

IscriviStudenteAlCorso dove:

1. L'attore Registrante invia il messaggio `iscriviStudente("Jim","UML")` all'oggetto `:GestoreRegistrazioni`.
- 1.1 Il `:GestoreRegistrazioni` invia a sè stesso il messaggio `trovaStudente("Jim")` e salva il valore di ritorno nella variabile booleana `s`
 - 1.1.1 Il `:GestoreRegistrazioni` invia il messaggio `trova("Jim")` al multioggetto di nome studenti e salva il valore di ritorno nella variabile `unoStudente`
- 1.2 Il `:GestoreRegistrazioni` invia a sè stesso il messaggio `trovaCorso("UML")` e salva il valore di ritorno nella variabile booleana `c`
 - 1.2.1 Il `:GestoreRegistrazioni` invia il messaggio `trova("UML")` al multioggetto di nome corsi e salva il valore di ritorno nella variabile `unCorso`
- 1.3 Se sia `c` che `s` sono vere il `:GestoreRegistrazioni` invia il messaggio `iscrivi(unoStudente)` all'oggetto di nome Corso.

`IscriviStudente(...)` restituisce `ok` se lo Studente è stato iscritto al Corso (ossia quando sia `c` che `s` sono vere)

Esercizio: soluzione



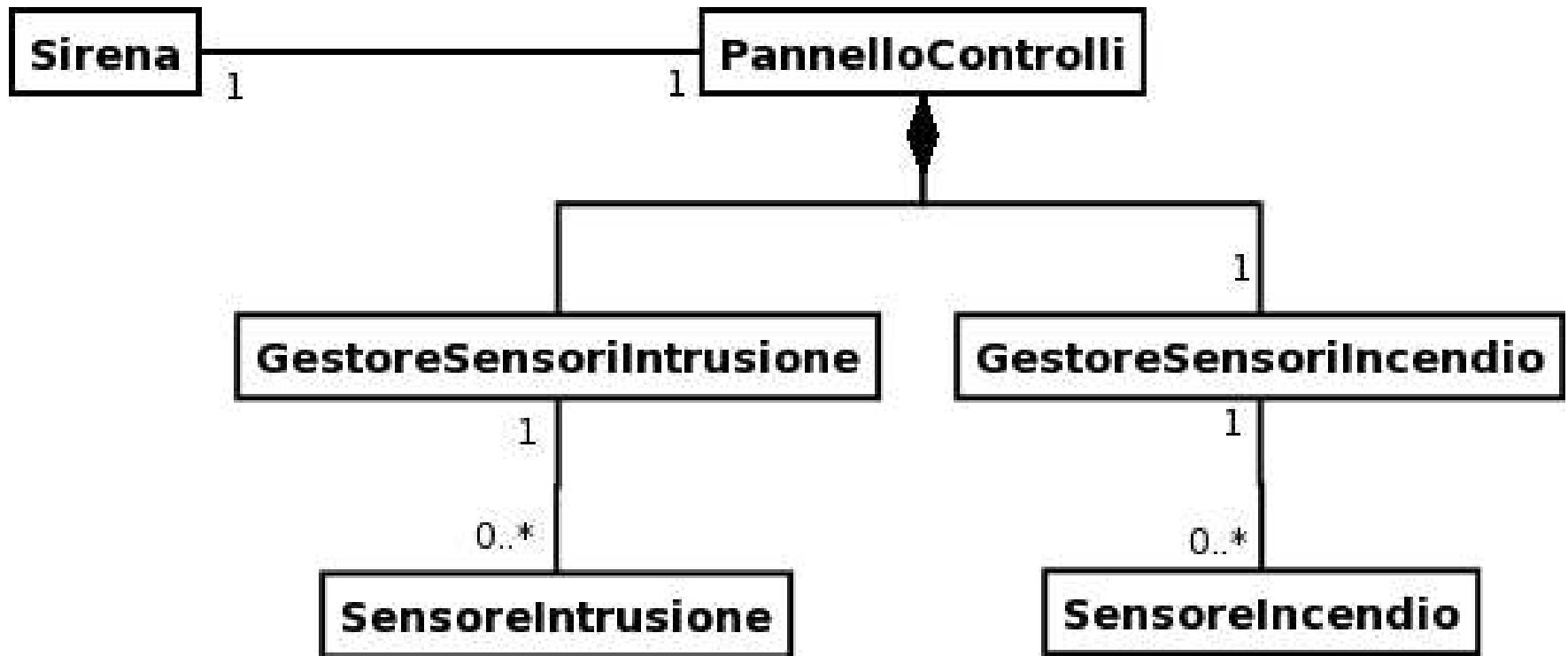
Oggetti attivi

- La **concorrenza** può essere facilmente modellata nei *diagrammi di collaborazione*.
- Ogni **thread** o **processo concorrente** viene modellato come un **oggetto attivo**, che incapsula un proprio flusso di controllo
- Gli oggetti attivi eseguono in concorrenza e hanno ciascuno un proprio *focus di controllo*
- *Notazione*: hanno il bordo disegnato più spesso

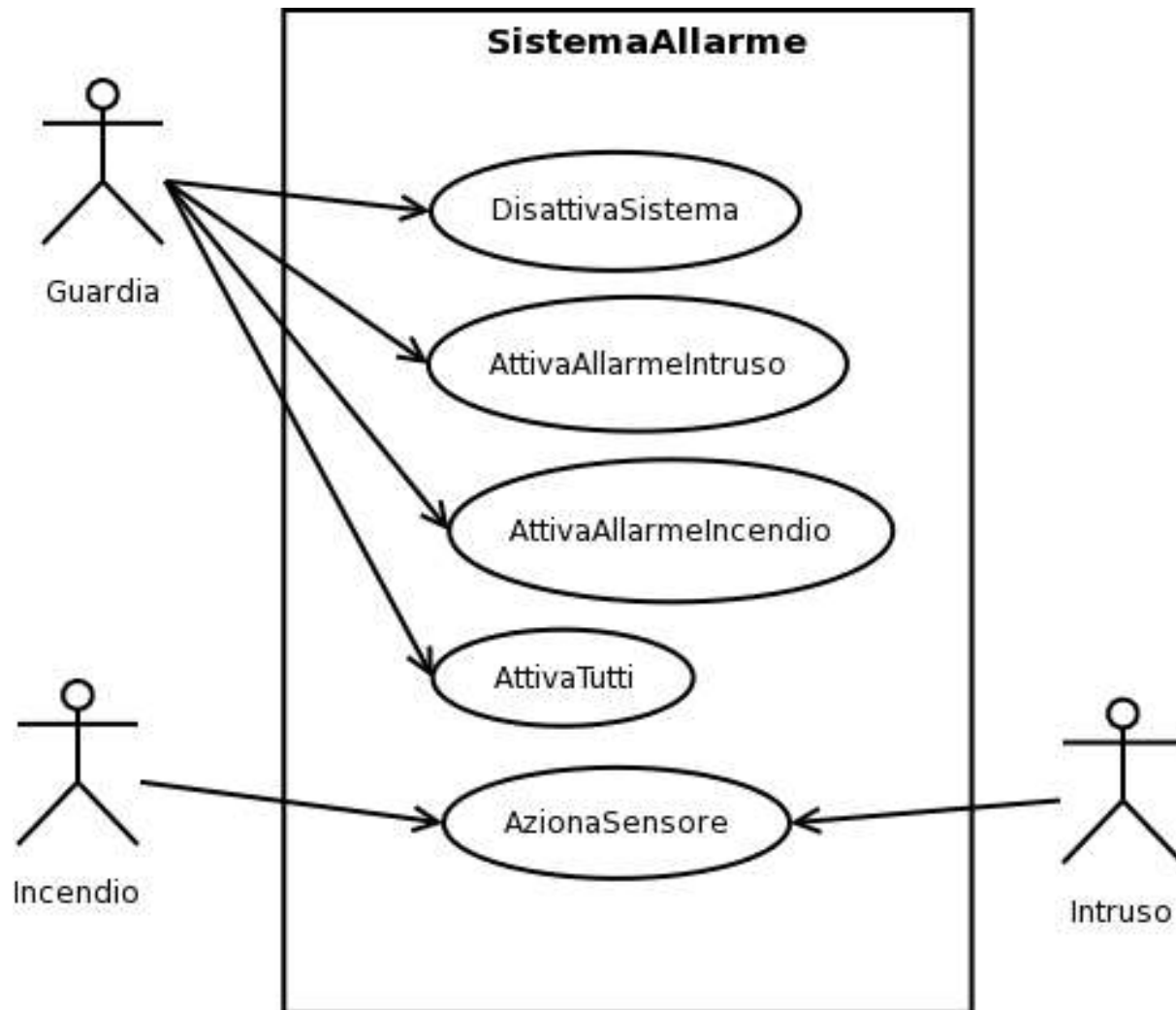
Esercizio sistema allarme (1/3)

- Il sistema di sicurezza che si vuole modellare è costituito da quattro componenti: il pannello dei controlli, la sirena, i sensori di incendio e i sensori di intrusione
- Il pannello dei controlli si compone di gestori per ciascun tipo di sensore
- Il pannello dei controlli fa suonare la sirena se si attiva l'allarme
- Si scriva il diagramma delle classi opportuno

Esercizio sistema allarme (1/3) (soluzione)



Esercizio sistema allarme (2/3)

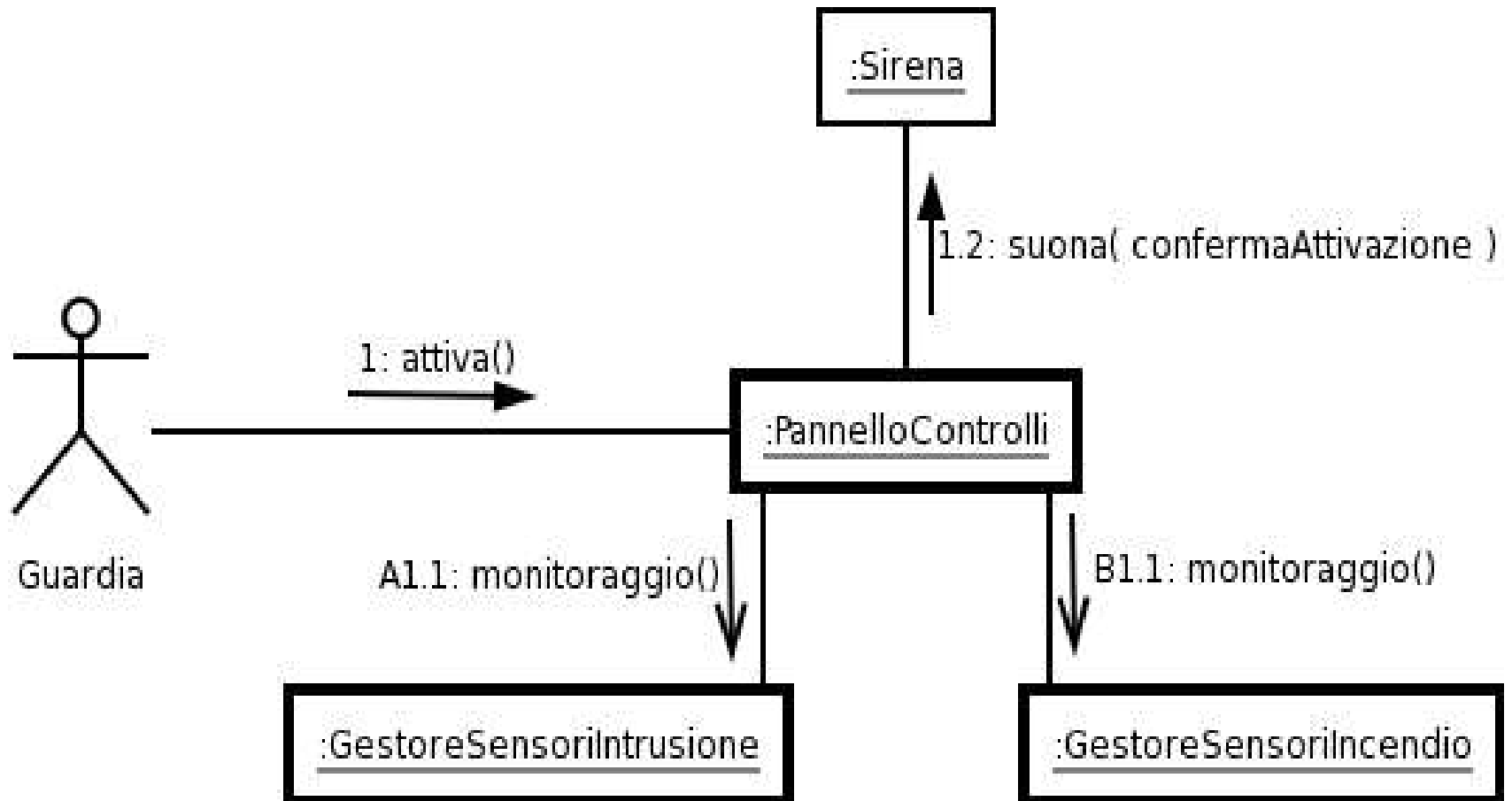


Esercizio sistema allarme (3/3)

Caso d'uso: AttivaTutti
ID: UC1
Attori: Guardia
Precondizioni: 1. La Guardia dispone della chiave di attivazione
Sequenza degli eventi: 1. La Guardia usa la chiave di attivazione per attivare il sistema 2. Il sistema inizia il monitoraggio dei sensori antifurto ed incendio 3. Il sistema emette un segnale di conferma con la sirena pe rindicare che è attivo
Postcondizioni: 1. Il sistema è attivato 2. Il sistema effettua il monitoraggio dei sensori

Disegnate il diagramma di collaborazione in forma istanza relativo a questo caso d'uso

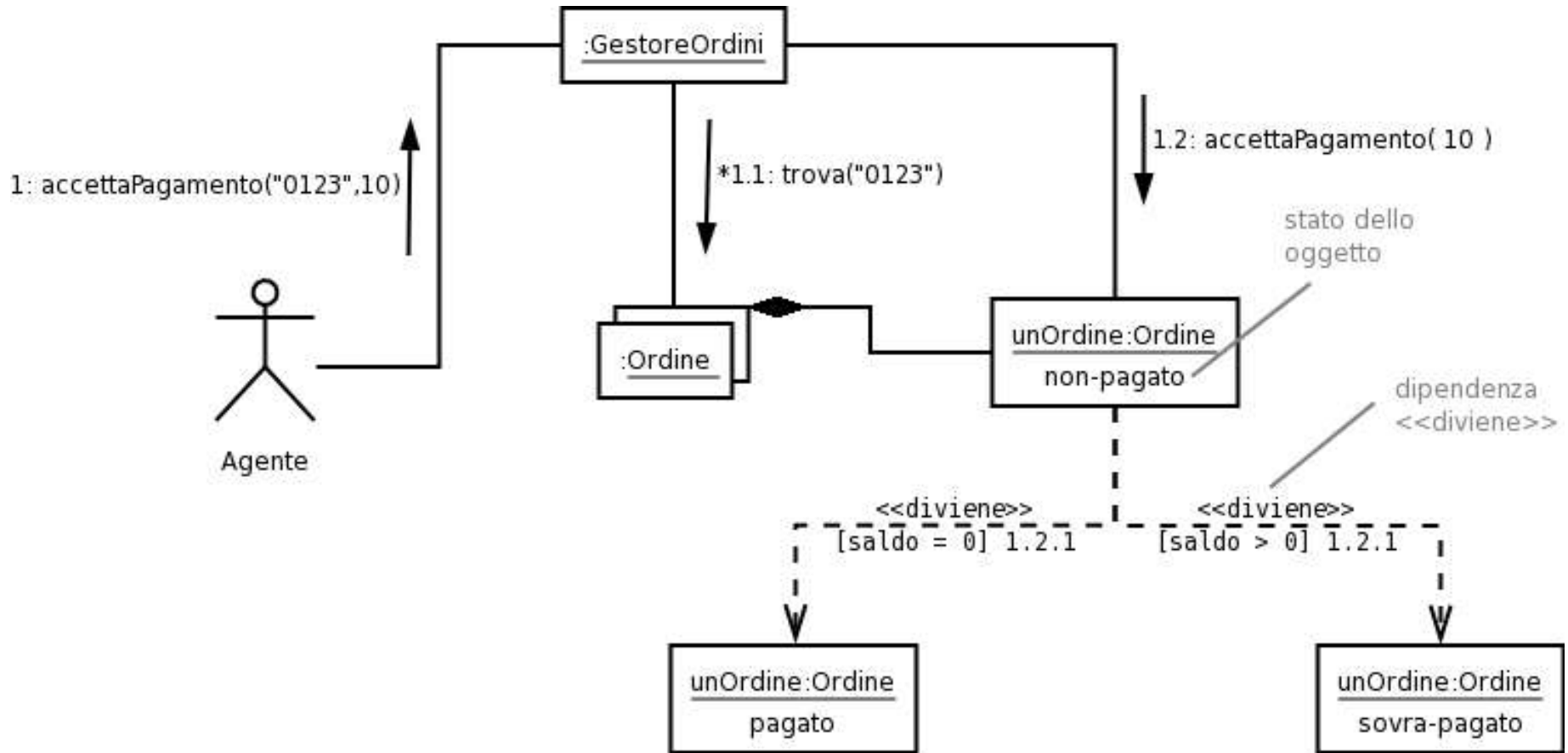
Esercizio sistema allarme (3/3) (soluzione)



Stato di un oggetto

- Lo **stato** di un oggetto è una condizione durante la vita dell'oggetto in cui esso soddisfa una qualche condizione, esegue una qualche attività o attende un qualche evento
- Gli oggetti mentre interagiscono e collaborano tra loro, passano attraverso una successione di stati distinti
- Un messaggio può spesso provocare una transizione di stato sull'oggetto che lo riceve

Stato di un oggetto (esempio)



Diagrammi di Sequenza

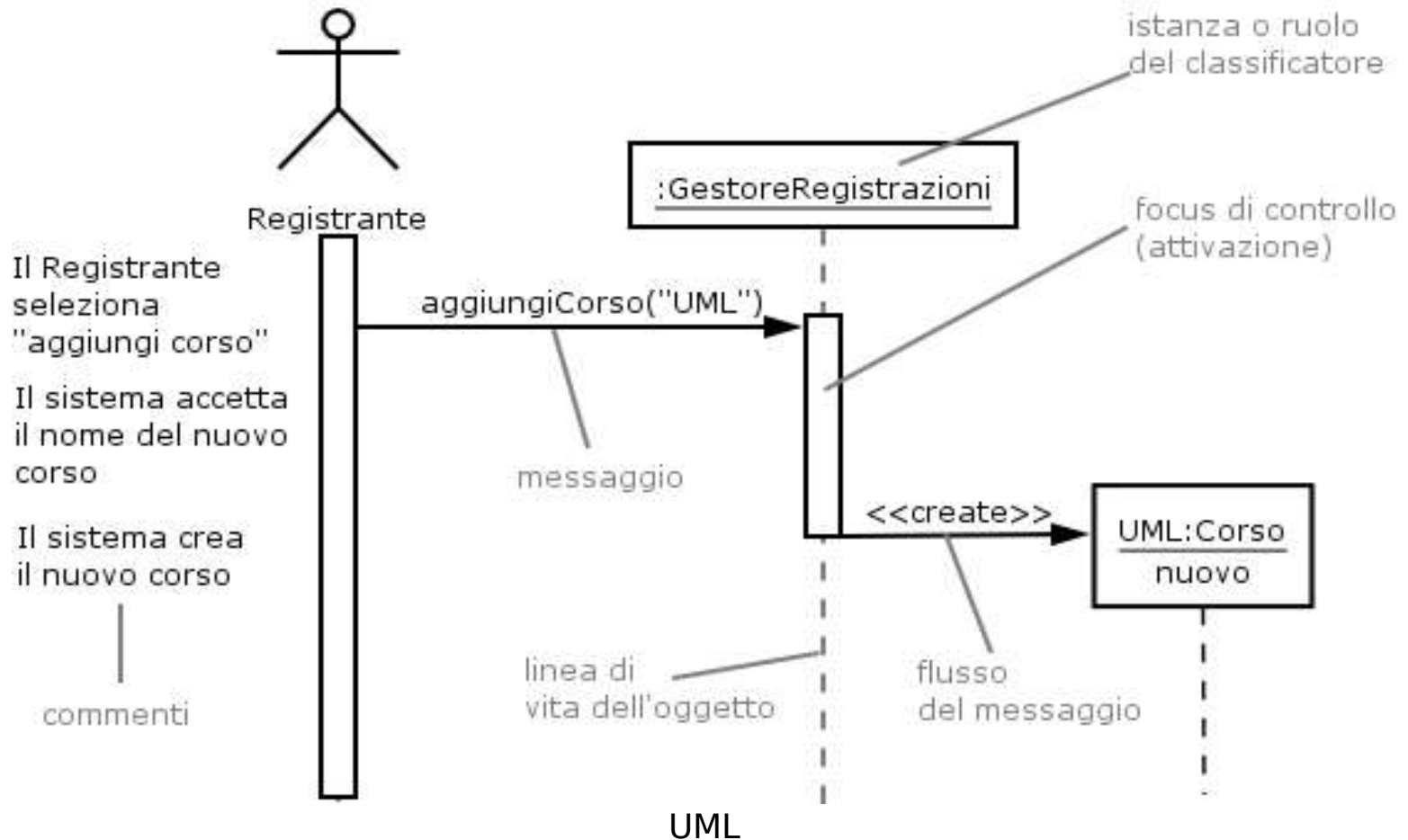
- I *diagrammi di sequenza* mostrano le interazioni tra oggetti ordinate temporalmente
- Contengono gli stessi elementi di modellazione dei *diagrammi di collaborazione* più:
 - *Le linee di vita degli oggetti*
 - *I focus di controllo*

Diagrammi di Sequenza

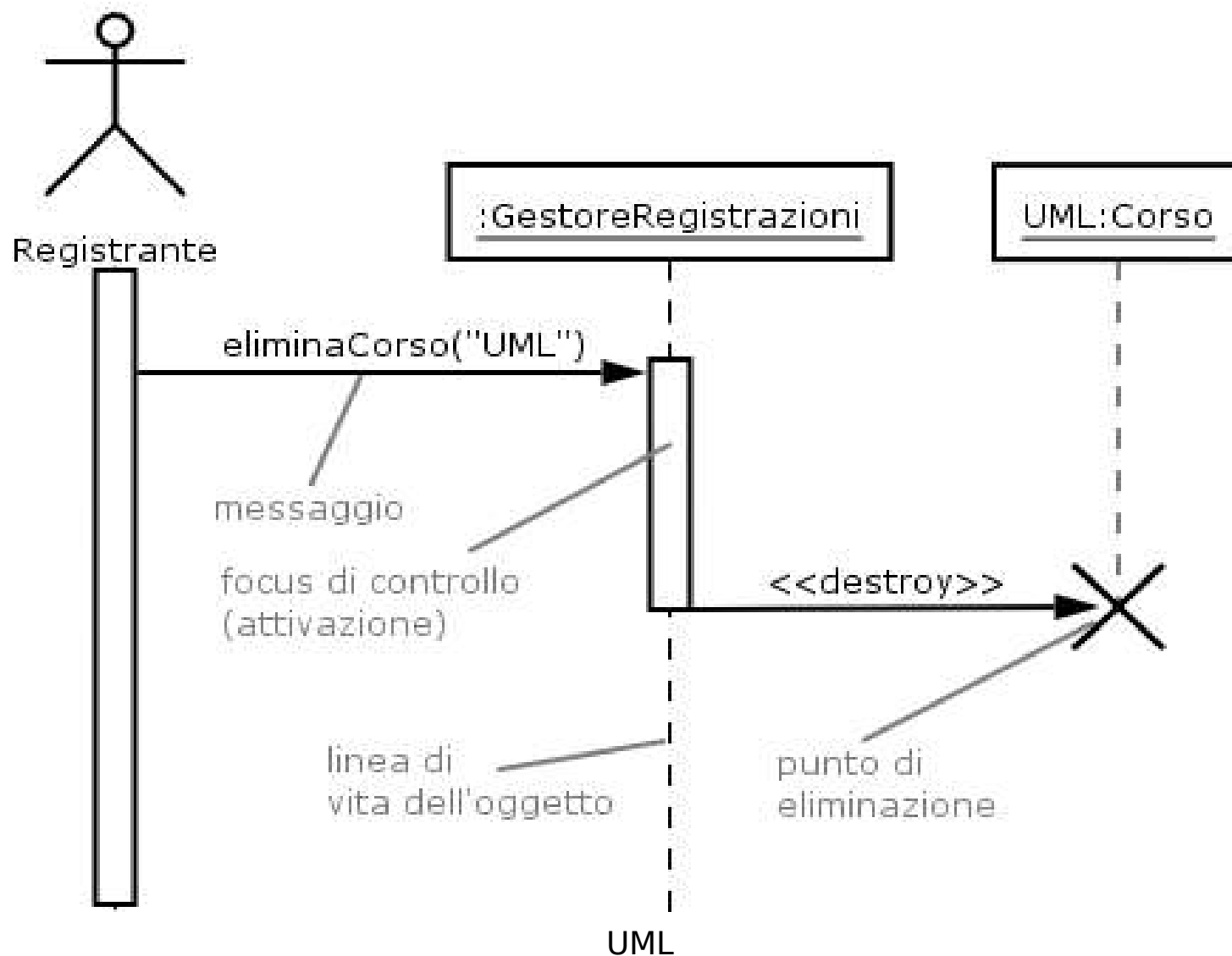
- I *diagrammi di collaborazione* sono indicati per:
 - mostrare gli oggetti e le loro relazioni strutturali (la collaborazione)
- I *diagrammi di sequenza* sono indicati per
 - esplicitare le interazioni tra gli oggetti come sequenza di eventi ordinata temporalmente

Diagrammi di Sequenza (esempio)

Consideriamo il diagramma di sequenza relativo alla slide a pag 17



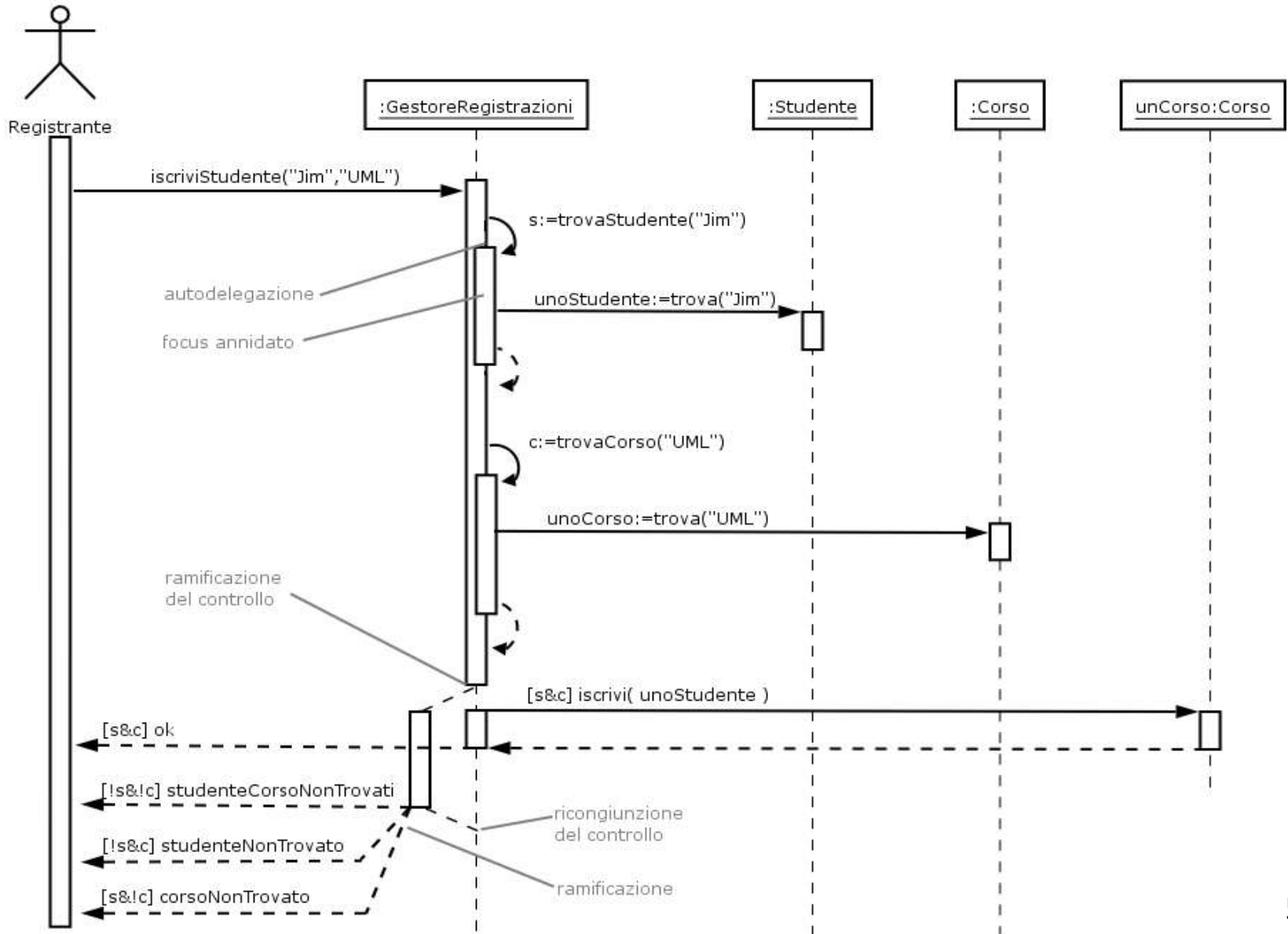
Diag. di Sequenza (esempio eliminazione)



Esercizio 1 (1/2)

- Si scriva il *diagramma di sequenza* corrispondente al *diagramma di collaborazione* della slide a pag 31
- L'auto-delegazione si rappresenta come visto nella slide precedente per l'iterazione
- La ramificazione si rappresenta biforcando la linea di vita del corrispondente oggetto.

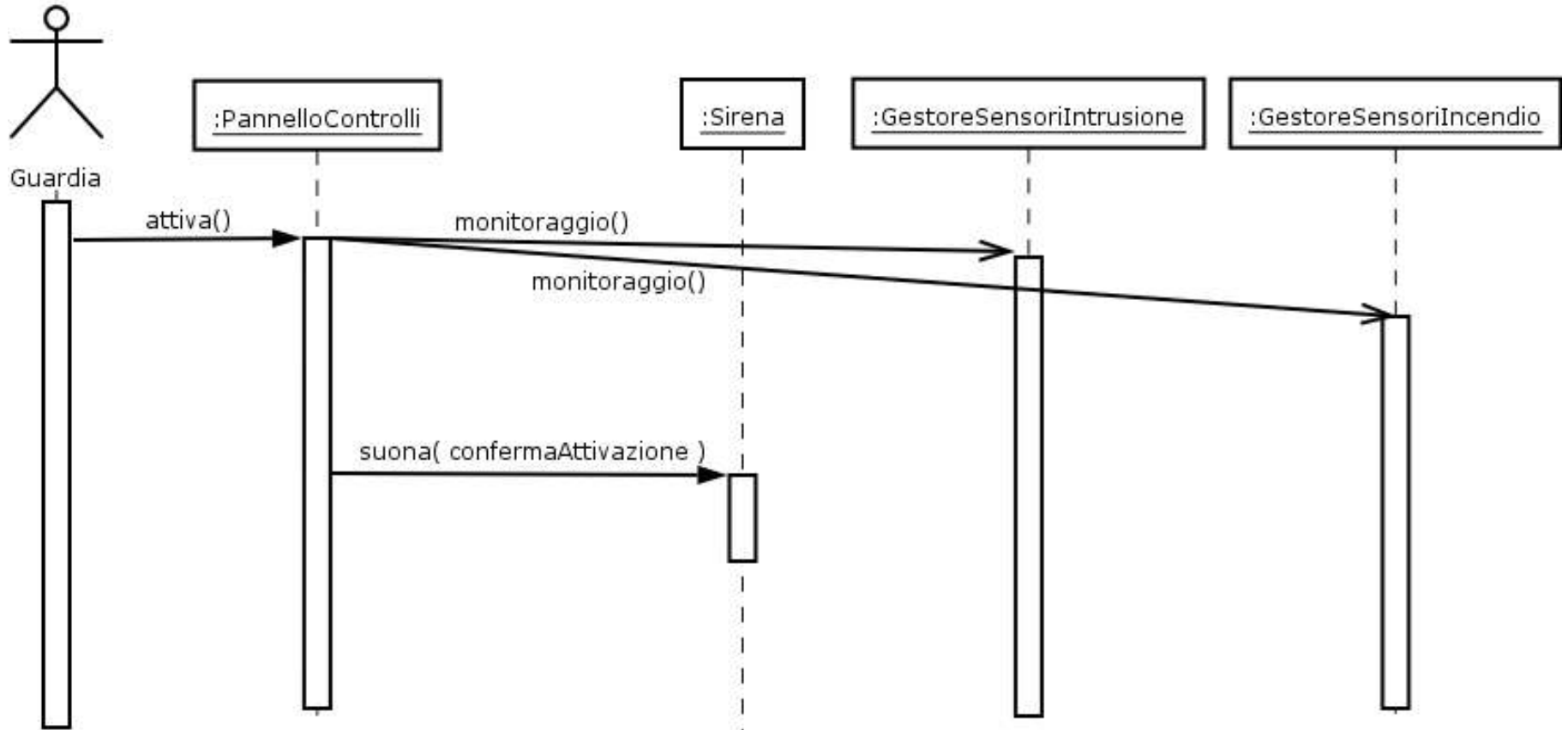
Esercizio 1 (2/2) Soluzione



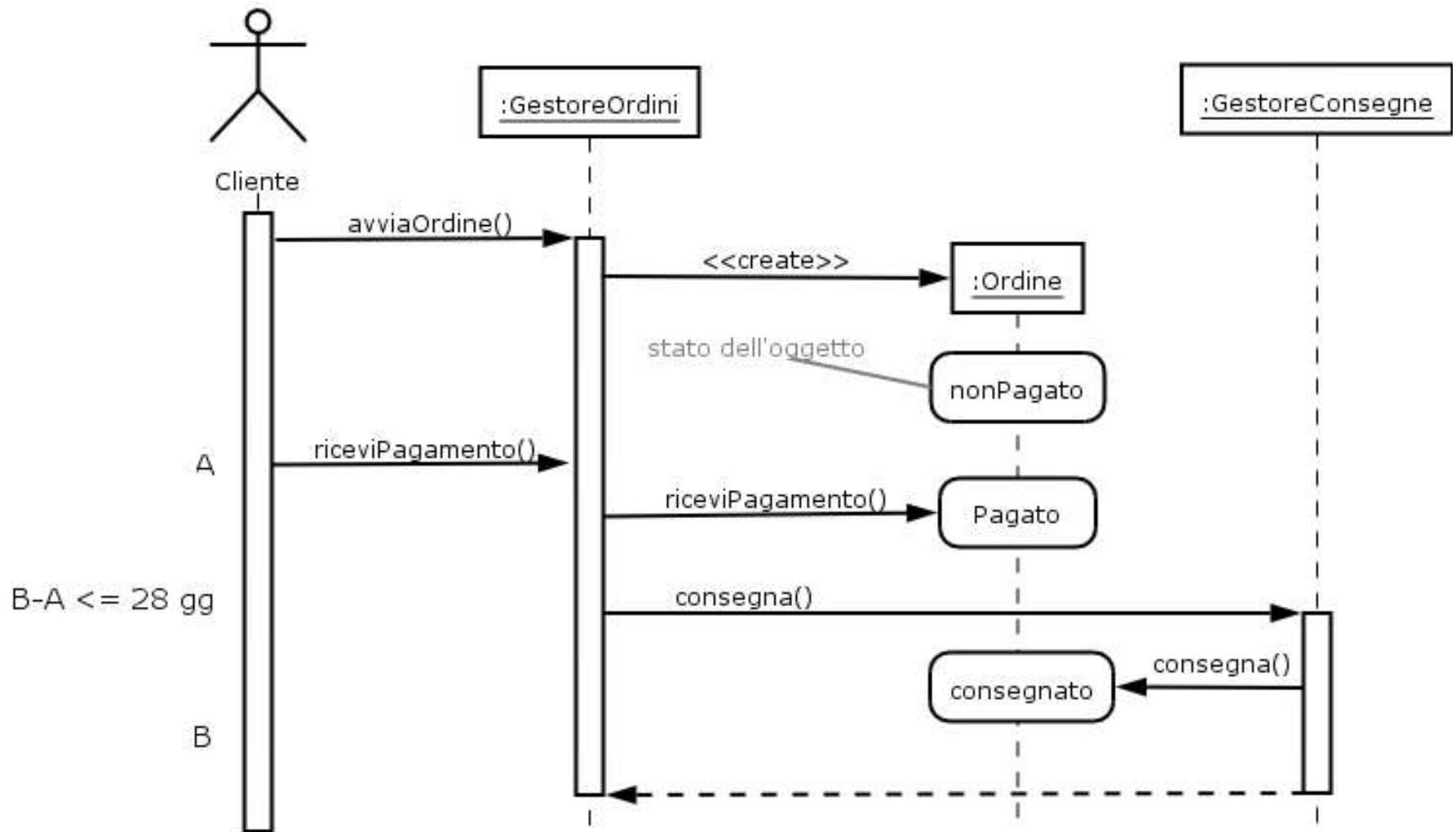
Esercizio 2 (1/2)

- Si scriva il *diagramma di sequenza* corrispondente al *diagramma di collaborazione* della slide a pag 37

Esercizio 2 (2/2) Soluzione



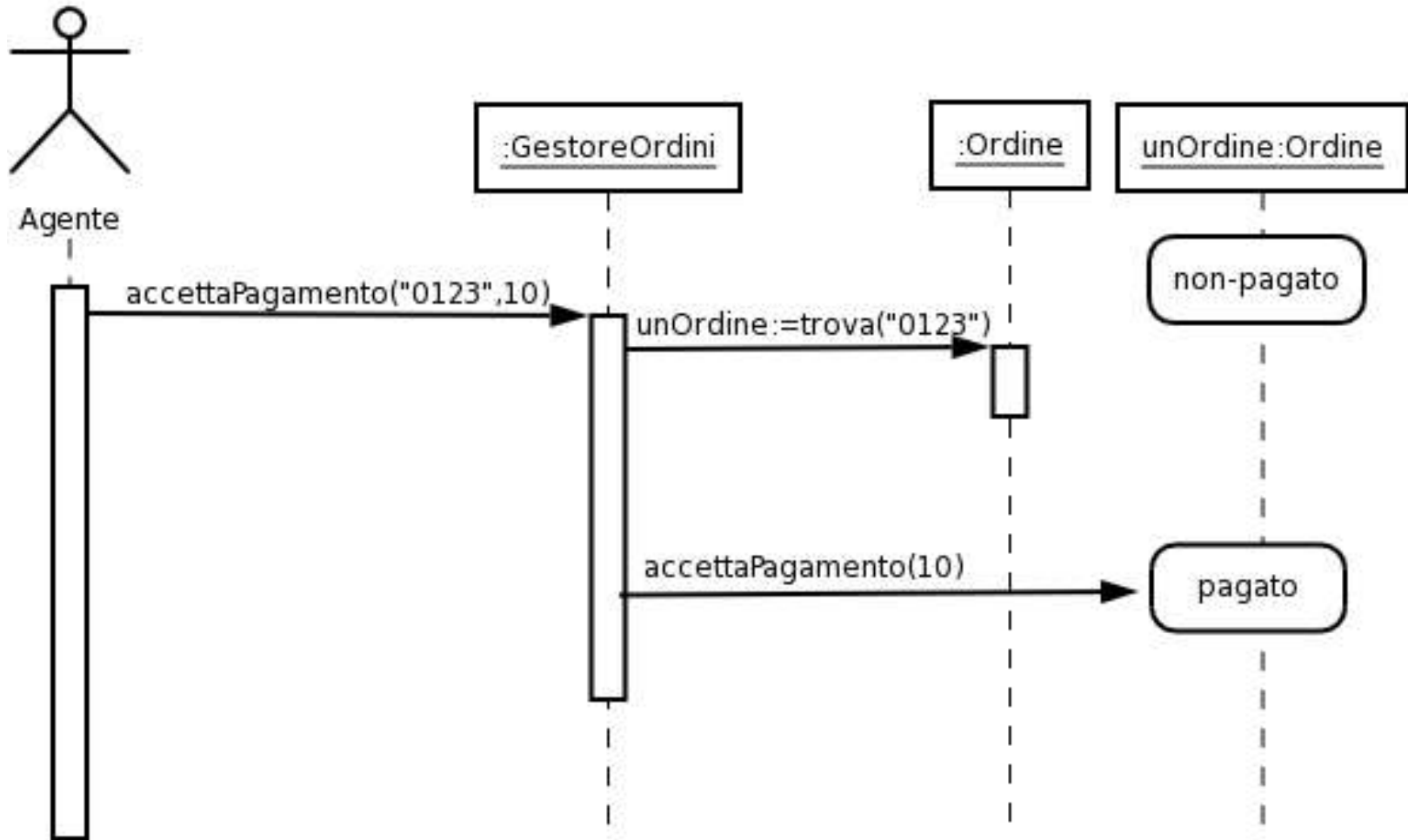
Esempio: stati di un oggetto



Esercizio 3 (1/2)

- Si scriva il *diagramma di sequenza* corrispondente al *diagramma di collaborazione* della slide a pag 39
- Non si consideri lo stato “sovra-pagato” relativo all'espressione *[saldo>0]*

Esercizio 3 (2/2) Soluzione

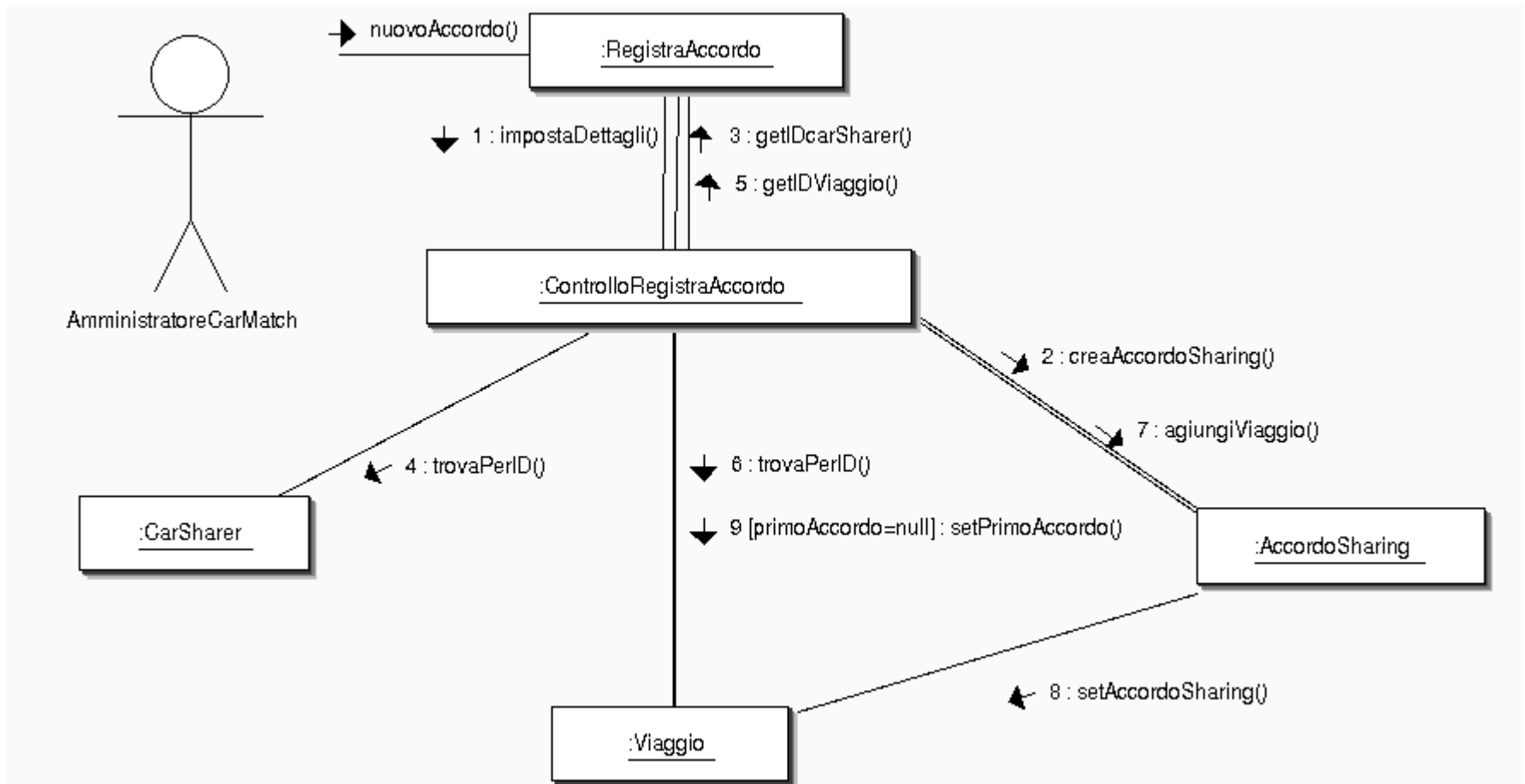


Diag. di Sequenza vs Diag. Di Collaborazione

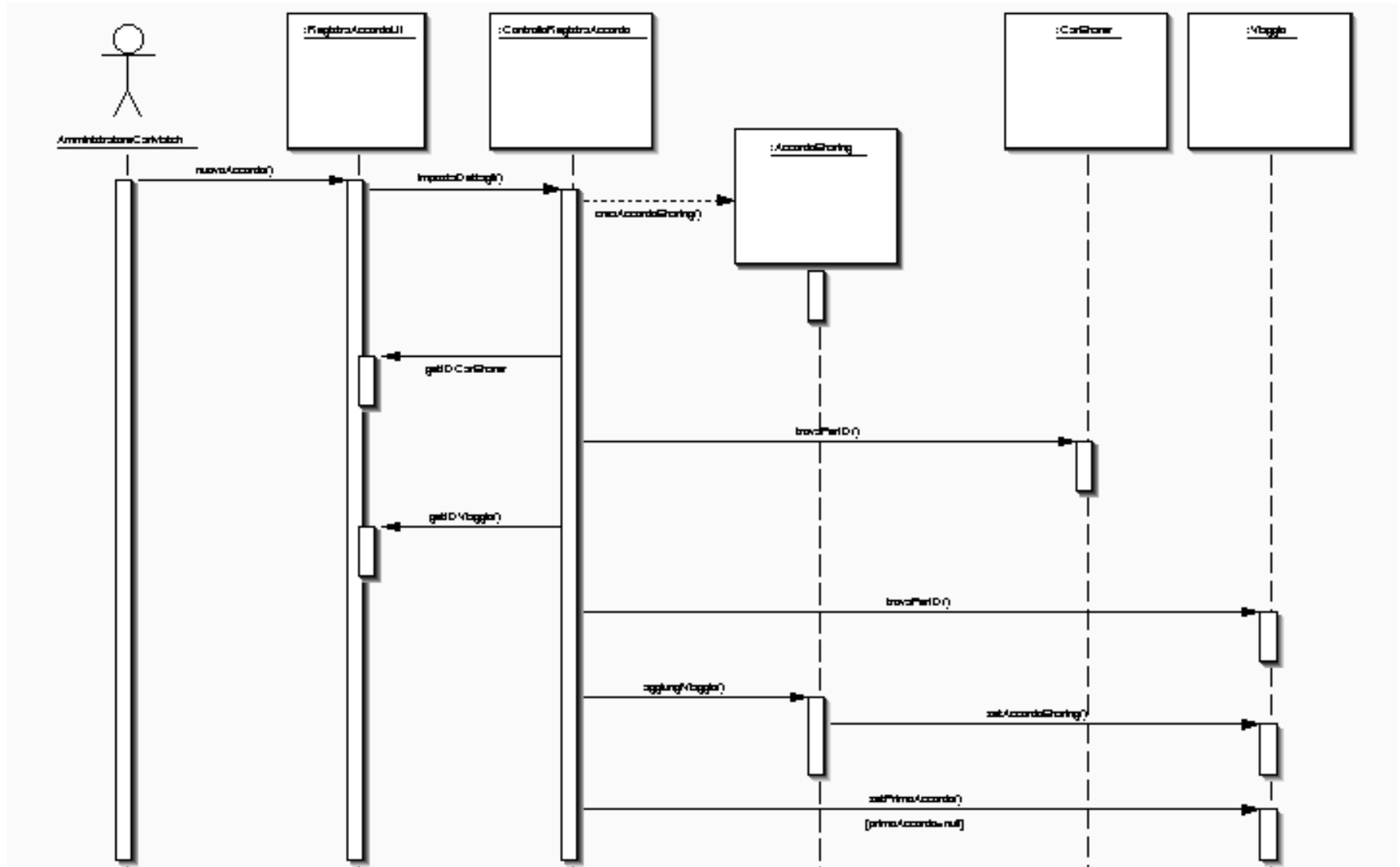
- **Diagrammi di Collaborazione** sono migliori nel rappresentare
 - multioggetti
 - iterazioni
- **Diagrammi di Sequenza** sono migliori nel rappresentare
 - passaggio del focus di controllo
 - ramificazioni

Esercizio 4 (1/2)

Si scriva il *diagramma di sequenza* corrispondente al *diagramma di collaborazione* seguente:



Esercizio 4 (2/2) Soluzione



Riferimenti

- [UML 1.5] *OMG UML Specification v. 1.5.*
- [AN02] Arlow, Neustadt, *UML e Unified Process*, McGraw-Hill, 2002
- [BSL02] Bennett, Skelton, Lunn, *Introduzione a UML*, McGraw-Hill collana Schaum's, 2002
- I diagrammi sono stati realizzati con *Dia* ed *Eclipse*
<http://www.gnome.org/projects/dia/>
<http://www.eclipseuml.com>