

Programmazione a Oggetti

Modulo B

Lezione 22

Dott. Alessandro Roncato

22/04/2013

Ultima

- Codice Parser
- Diagramma stati

Oggi

- Design Pattern State
- Diagramma Stati (cont.)
- Diagramma Attività
- Proxy

Implementazione Esempio

```
public class Macchinetta {  
    enum Stato={Wait,Credito,Selezione, Credito_Selezione,...  
    Stato stato=Stato.Wait;  
    double credito=0.0; double prezzo=0.0;  
    int selezione;  
  
    public void moneta(){  
        switch(stato){  
            case Stato.Wait:  
                stato=Stato.Credito;  
                credito+=valore_moneta;  
                break;  
            case Stato.Credito:  
                //stato=Stato.Credito;  
                credito+=valore_moneta;  
                break;  
            ...  
        }  
    }  
}
```

Implementazione Esempio

```
...
public void tasto(){
    switch(stato){
        case Stato.Wait:
            stato=Stato.Selezione;
            selezione = posizione(tasto);
            prezzo = selezione;
            break;
        case Stato.Credito:
            selezione = posizione(tasto);
            prezzo = valore[selezione];
            if (credito>=prezzo)
                stato =Stato.Erogazione;
            else
                stato = Stato.Credito_Selezione;
            break;
        ...
    }
}
```

Pro e contro Switch

- + Implementazione diretta;
- Tanti casi: Stati x Eventi
- Difficile da gestire

Si può usare la Table State avendo a disposizione un'interprete:

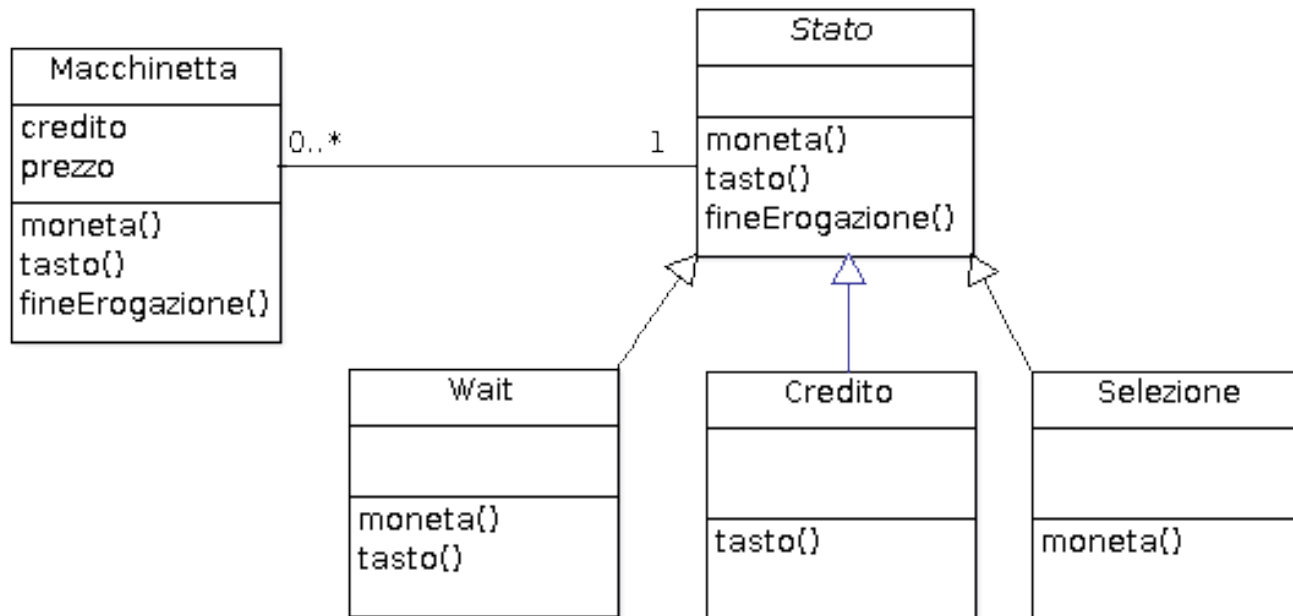
Sorgente	Destinazione	Evento	Guardia (*)	Attività (*)
Wait	Credito	moneta		
Wait	Selezione	tasto		
Credito	Credito	moneta		
Credito	Credito/Selezione	tasto	credito<prezzo	
Credito	Erogazione	tasto	credito>=prezzo	
...				

Pattern State

Alternativamente si può usare il Design Pattern State.

Si definisce una nuova classe Astratta in cui ogni evento corrisponde a un metodo e il metodo non fa niente nella classe astratta. Per ogni stato si definisce una nuova classe che estende la classe astratta e sovrascrive i soli metodi per i quali l'evento ha un effetto nello stato

Pattern State



Pattern State

```
public class Macchinetta {  
    Stato stato = new Wait();  
    double credito; double prezzo;  
    int selezione;  
  
    public void setStato(Stato s){  
        stato=s; }  
  
    public void tasto(){  
        stato.tasto(this);}  
    public void moneta(){  
        stato.moneta(this);}  
    public void fineErogazione(){  
        stato.fineErogazione(this);}  
    ...  
}
```

Pattern State

```
public abstract class Stato {  
  
    //tutti metodi implementati in modo da ignorare  
    //tutti gli eventi  
    public void tasto(Macchinetta m){  
    }  
    public void moneta(Macchinetta m){  
    }  
    public void fineErogazione(Macchinentta m){  
    }  
    ...  
}
```

Pattern State

```
public class Wait extends Stato {  
    private static final Wait singleton = new Wait();  
    private Wait(){}  
    public static Wait getInstance(){  
        return singleton; }  
  
    public void tasto(Macchinetta m){  
        m.setStato(Selezione.getInstance());  
    }  
    public void moneta(){  
        m.setStato(Credito.getInstance());  
        m.credito=valore_moneta;  
    }  
}
```

Pattern State

```
public class Credito extends Stato{
    private static final Credito single = new Credito();
    private Credito(){ }
    public static Credito getInstance(){
        return single; }

    public void tasto(Macchinetta m){
        if(m.credito>=m.prezzo)
            m.setStato(Erogazione.getInstance());
        else
            m.setStato(CreditoSelezione.getInstance());
        ...
    }
    public void moneta(Macchinetta m){
        m.credito+=valore_moneta;
    }
}
```

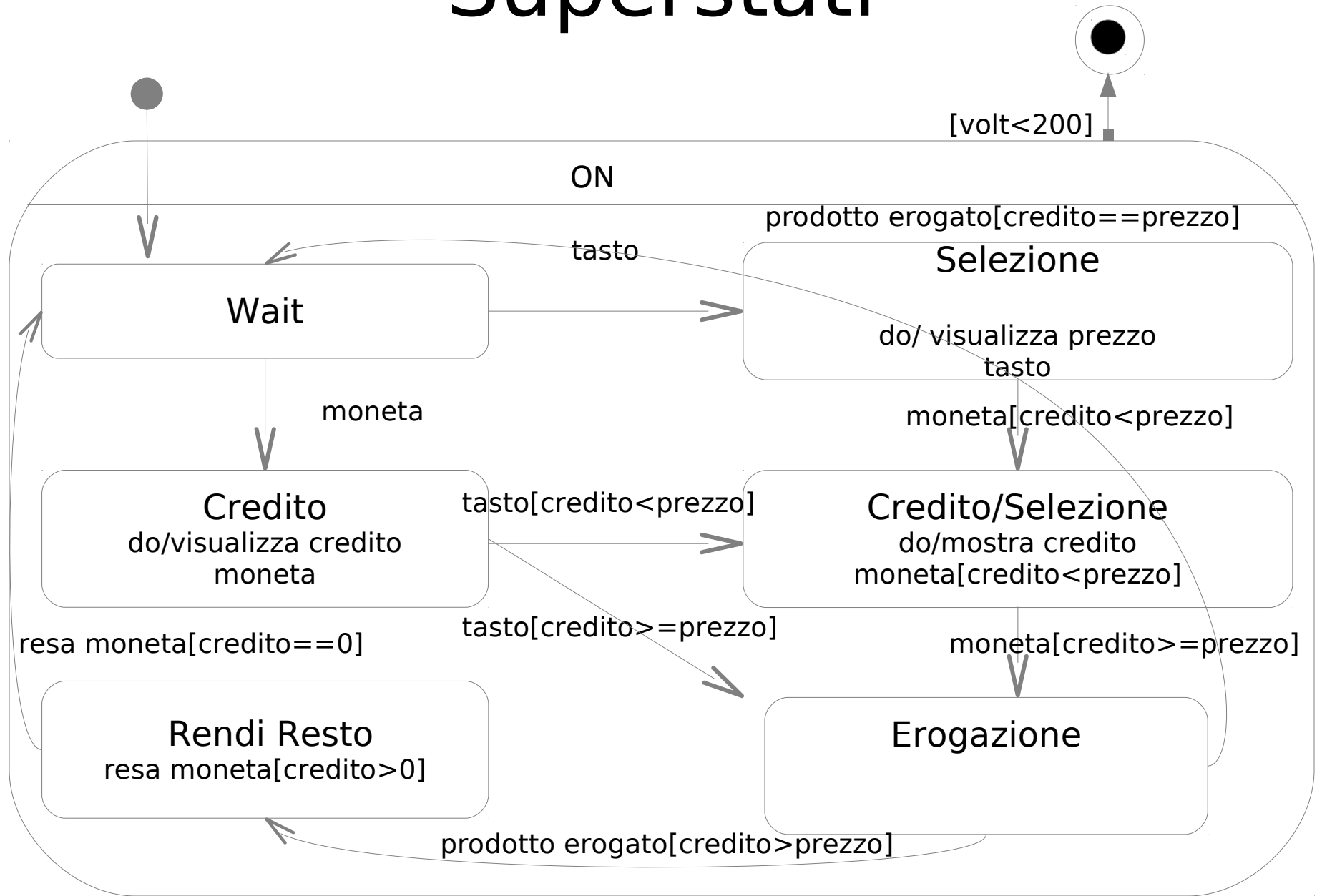
Pattern State

Il comportamento di un oggetto dipende dal suo stato. Quale alternativa alla logica condizionale?

Crea delle classi stato per ciascuno stato che implementano un'interfaccia comune.

L'implementazione polimorfa elimina la logica condizionale.

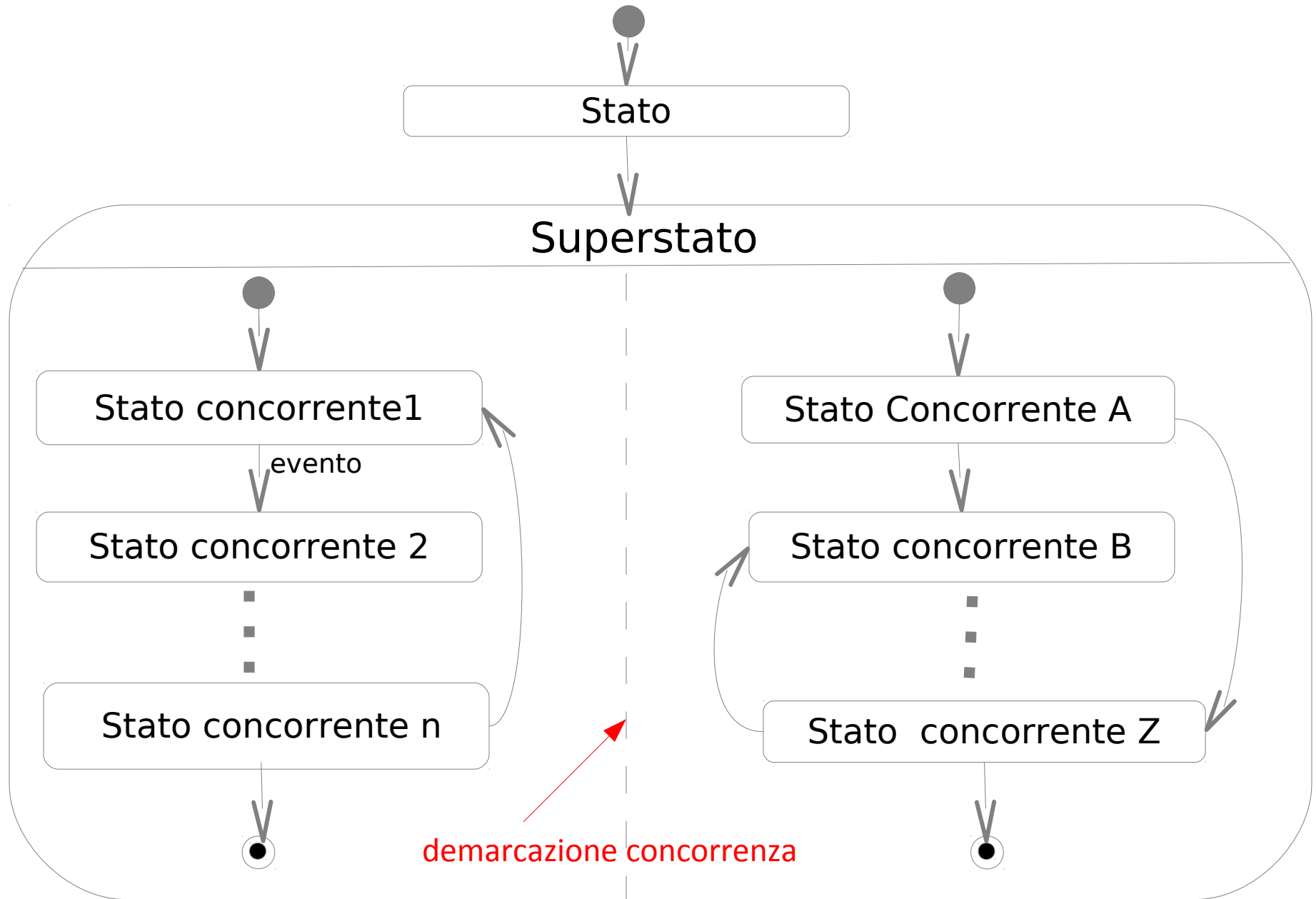
Superstati



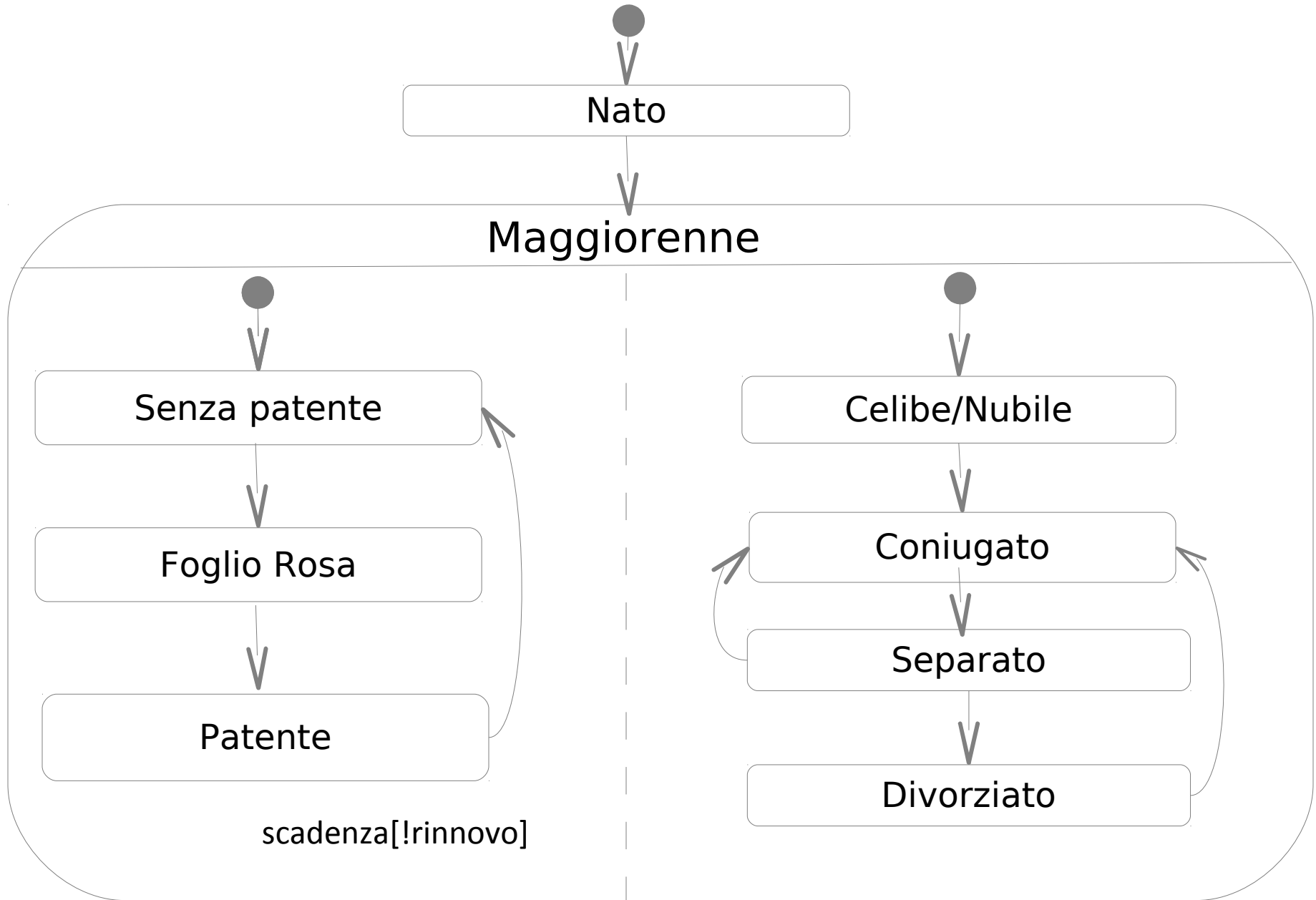
Superstati

Il superstato permette di rappresentare un comportamento comune a tutti gli stati appartenenti al superstato senza bisogno di dover riportarlo per ogni singolo stato.

Stati concorrenti



Esempio stati concorrenti



Stati concorrenti

Gli stati concorrenti rappresentano due comportamenti indipendenti e concorrenti del superstato.

In pratica:

- non possono esserci transizioni tra un insieme di stati concorrenti e l'altro
- e ogni combinazione di stati appartenenti zone diverse è lecita

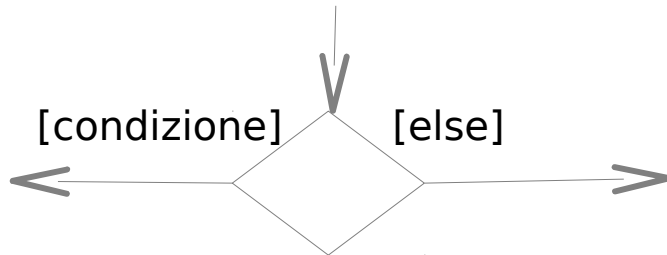
Diagramma attività

La notazione del diagramma delle attività è in parte condivisa con quella del diagramma di stato appena vista. Lo scopo dei diagrammi di attività è però molto diverso.

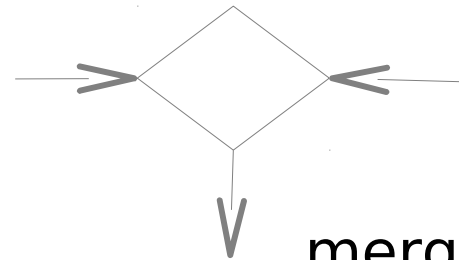
- 1) descrivere algoritmi
- 2) descrivere logica di business
- 3) descrivere casi d'uso
- 4) evidenziare parallelismo
- 5) workflow

Notazione

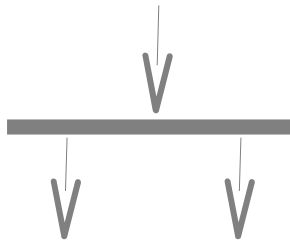
Diagrammi di stato +



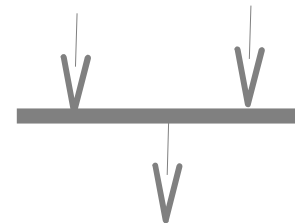
branch



merge



fork



join

Notazione

Stato → Activity

Transizione → Flusso

Transizioni senza eventi, guardie e activity.

Partizioni

Molta altra notazione...

Esempio

Lavagna

Proxy

Accesso diretto a un oggetto (chiamato Soggetto) non è possibile.

Aggiungere un livello di indirizzione con un oggetto Proxy che implementa la stessa interfaccia dell'oggetto Soggetto ed è responsabile del controllo o del potenziamento dell'oggetto Soggetto.

Remote Proxy

Accesso diretto al Soggetto non e' possibile perché risiede su un altro host.

Si accede a un oggetto Proxy che implementa la stessa interfaccia dell'oggetto Soggetto remoto ed è responsabile della comunicazione con l'oggetto remoto.

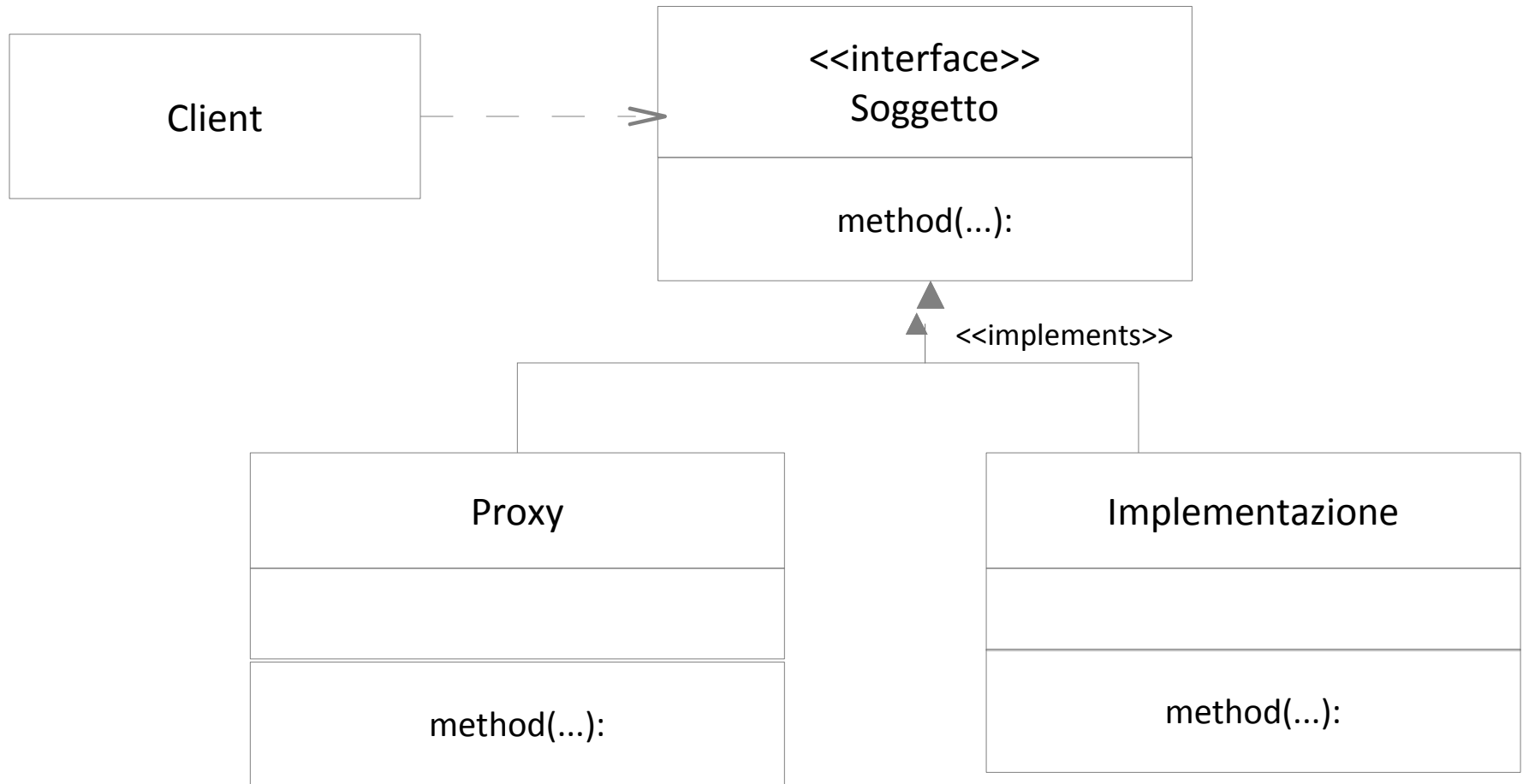
(RMI l'oggetto Proxy è chiamato Stub)

Failover Proxy

Accesso diretto al Soggetto non e' affidabile.

Si accede a un oggetto Proxy che implementa la stessa interfaccia del Soggetto ed è responsabile di supplire al Soggetto non affidabile quando Soggetto non è (temporaneamente) disponibile

Proxy



Varie

Dall'id all'oggetto

Passare oggetti come parametri piuttosto che le sole informazioni di interesse per il metodo

Mapping degli oggetti: mapping diretto vs Database Mapper (pag 644)