



# UML2

## Diagrammi di Attività

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Brevissima introduzione alle Petri Nets

Le reti di Petri sono un formalismo ideato negli anni 60 per modellare sistemi concorrenti, asincroni, distribuiti, paralleli, non deterministici, e/o stocastici.

Formalmente sono definite da una tupla  $\langle \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0 \rangle$ :

- $\mathcal{P}$  è un insieme finito di **piazze**
- $\mathcal{T}$  è un insieme finito di **transizioni**
- $\mathcal{F} \subseteq \{\mathcal{P} \times \mathcal{T}\} \cup \{\mathcal{T} \times \mathcal{P}\}$  è detta **relazione di flusso** della rete di Petri
- $\mathcal{W} : \mathcal{F} \rightarrow \mathbb{N}^+$  è la **funzione peso** che associa valore non nullo agli elementi di  $\mathcal{F}$
- $\mathcal{M}_0 : \mathcal{P} \rightarrow \mathbb{N}$  è la **marcatura iniziale** ed indica lo **stato iniziale** della rete di Petri

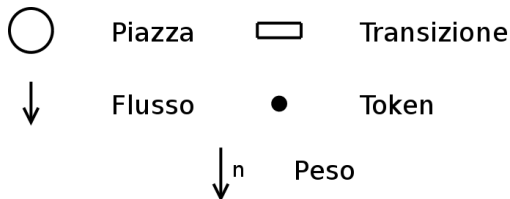
Deve poi valere che:  $\mathcal{P} \cup \mathcal{T} \neq \emptyset$  e  $\mathcal{P} \cap \mathcal{T} = \emptyset$

In ogni momento lo stato di una rete di petri è indicato dalla **funzione di marcatura** che associa ad ogni piazza un numero naturale indicante il numero di “token” (gettoni) presenti nella piazza:

$\mathcal{M} : \mathcal{P} \rightarrow \mathbb{N}$

# Reti di Petri

notazione grafica



Evoluzione di una rete di Petri:

- Data una transizione chiamiamo **piazza di input** (o di output), una piazza collegata ad un flusso entrante (uscente) dalla transizione stessa.
- Una transizione è abilitata, e può dunque “sparare”, se e solo se **TUTTE** le piazze collegate ad un qualsiasi flusso di input alla transizione, contengono un numero di token **maggiore o uguale** al peso dell’elemento flusso che collega ognuna di loro alla transizione.
- Quando una transizione “spara” ogni piazza collegata ad un flusso uscente dalla transizione riceverà un numero di token pari al peso del corrispondente flusso.
- Le transizioni con nessun flusso entrante sono sempre abilitate.

# Petri Net

Tipicamente le **piazze possono rappresentare una risorsa** e l'accesso alla risorsa può essere modellato dalla presenza di Token. Allo stesso modo piazze e transizioni possono essere utilizzate per **rappresentare stati di un processo e l'evoluzione dello stesso**

Politiche di risoluzione dei conflitti **non sono intrinseche nel formalismo** ma devono invece essere implementate. Ad esempio il formalismo non è *fair*. Esempio....

Una rete di Petri si dice in **deadlock** se non ci sono transizioni abilitate. Nel caso in cui la rete di Petri si trovi in **deadlock** la sua marcatura rappresenterà lo **stato finale delle rete**. Esempio...

# Esempi

- Somma di due numeri
- Sottrazione di due numeri
  
- Buffer di due posizioni
- Lettori/Scrittori
- Filosofi a cena
- Semaforo
  
- Prodotto di due numeri
- Divisione intera di due numeri

# Problemi e possibili estensioni al modello

Formalismo che si focalizza sul **controllo**. Comunque presenta alcune lacune in particolare per la gestione dei dati.

- In particolare non è possibile modificare il flusso in **dipendenza dei dati**.
- Non è possibile selezionare una transizione tra più transizioni attive.
- Possibilità di specificare **tempo e deadline**.

Le reti di Petri sono state estese per poter modellare sistemi più complessi:

- Token con associati valori. Token tipati e dunque gestione del contenuto. (**Coloured Petri Nets** - CPN)
- Politiche di priorità.  $\text{pri:T} \rightarrow \mathbb{N}$ .
- Reti di Petri temporizzate. Alle transizioni sono associati valori di tempo minimo e massimo entro i quali la transizione, se abilitata, potrà scattare.
- Reti temporizzate con associate funzioni di distribuzione (**Stochastic Petri Nets** - SPN)

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- classi

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- **casi d'uso**
- classi
- interfacce
- componenti
- collaborazioni
- operazioni



# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- **classi**
- interfacce
- componenti
- collaborazioni
- operazioni

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- classi
- **interfacce**
- componenti
- collaborazioni
- operazioni

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- classi
- interfacce
- **componenti**
- collaborazioni
- operazioni

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- classi
- interfacce
- componenti
- **collaborazioni**
- operazioni

# Diagrammi di Attività

I diagrammi di attività (**Activity Diagrams**) modellano un processo come un'attività costituita da un insieme di nodi connessi da archi. La semantica è descritta tramite il formalismo delle **Reti di Petri**. Servono dunque a modellare **aspetti del comportamento dinamico** di un sistema.

Il contesto di un diagramma di attività può essere dato da:

- casi d'uso
- classi
- interfacce
- componenti
- collaborazioni
- **operazioni**

# Diagrammi di Attività e UP

Nel flusso di lavoro dell'Analisi

- modellazione grafica del flusso di un caso d'uso
- modellazione del flusso tra più casi d'uso (diagramma di interazione generale)

Nel flusso di lavoro della Progettazione

- modellazione dei dettagli di un'operazione
- modellazione di specifici algoritmi

Nella modellazione dei processi di business (BP) - altro linguaggio in voga è BPMN 2.0

# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

- archi di controllo
- archi di oggetto

Azioni ed attività possono essere racchiuse all'interno di pre- e post-condizioni

# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

- flussi di controllo
- flussi di oggetti

Azioni ed attività possono essere racchiuse all'interno di pre- e post-condizioni



# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

- flussi di controllo
- flussi di oggetti

Azioni ed attività possono essere racchiuse all'interno di pre- e post-condizioni

# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

- flussi di controllo
- flussi di oggetti

Azioni ed attività possono essere racchiuse all'interno di pre- e post-condizioni

# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

- flussi di controllo
- flussi di oggetti

Azioni ed attività possono essere racchiuse all'interno di pre- e post-condizioni

# Attività

Tre tipi di nodi:

- nodi azione
- nodi controllo
- nodi oggetto

Due tipi di archi:

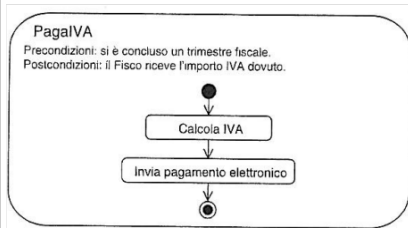
- flussi di controllo
- flussi di oggetti

Azioni ed attività possono essere racchiuse all'interno di **pre- e post-condizioni**

# Casi d'uso e Diagramma delle Attività

Un diagramma delle attività può fornire una rappresentazione grafica compatta di un Caso d'uso.

Caso d'uso: PagaIVA
ID: 1
Breve descrizione: Pagamento dell'IVA alla fine del trimestre fiscale.
Attori primari: Tempo
Attori secondari: Fisco
Precondizioni: 1. Si è concluso un trimestre fiscale.
Sequenza degli eventi principale: 1. Il caso d'uso inizia quando si conclude un trimestre fiscale. 2. Il sistema calcola l'ammontare dell'IVA dovuta al Fisco. 3. Il sistema trasmette un pagamento elettronico al Fisco.
Postcondizioni: 1. Il Fisco riceve l'importo IVA dovuto.
Sequenza degli eventi alternativa: Nessuna.



Azioni verranno nel diagramma verranno poi raffinate nelle attività di progettazione

# Semantica

La semantica è basata sulle Reti di Petri nelle versioni più complesse non approfondite nelle lezioni di modellazione formale di sistemi.

I token possono rappresentare:

- flusso di controllo
- un oggetto
- dati

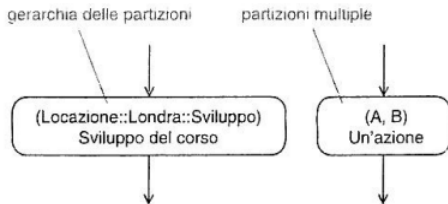
I token sono lo strumento di controllo del flusso che è comunque sottoposto a:

- post-condizioni del nodo sorgente
- condizioni di guardia sull'arco
- pre-condizioni sul nodo di destinazione

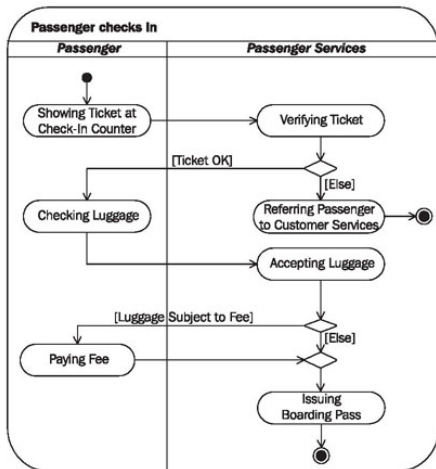
# Partizioni

È possibile raggruppare azioni attraverso uso di partizioni (anche annidate e rappresentanti sistemi esterni):

- casi d'uso
- classi
- componenti
- unità organizzative
- ruoli



# Example - Flight check-in





# Nodi Azione

## Regole di attivazione dell'azione

- esiste un token su ciascun arco entrante
- tutte le precondizioni locali del nodo azione sono soddisfatte

## Regole di uscita

- i token vengono emessi su ogni arco in uscita se la post-condizione viene valutata a vero

## Tipologie di nodi azione:

- azione di chiamata
- invia segnale
- accettazione evento
- espressione temporale

# Nodi Azione

## Regole di attivazione dell'azione

- esiste un token su ciascun arco entrante
- tutte le precondizioni locali del nodo azione sono soddisfatte

## Regole di uscita

- i token vengono emessi su ogni arco in uscita se la post-condizione viene valutata a vero

## Tipologie di nodi azione:

- azione di chiamata
- **invia segnale**
- accettazione evento
- espressione temporale

# Nodi Azione

## Regole di attivazione dell'azione

- esiste un token su ciascun arco entrante
- tutte le precondizioni locali del nodo azione sono soddisfatte

## Regole di uscita

- i token vengono emessi su ogni arco in uscita se la post-condizione viene valutata a vero

## Tipologie di nodi azione:

- azione di chiamata
- invia segnale
- **accettazione evento**
- espressione temporale

# Nodi Azione

## Regole di attivazione dell'azione

- esiste un token su ciascun arco entrante
- tutte le precondizioni locali del nodo azione sono soddisfatte

## Regole di uscita

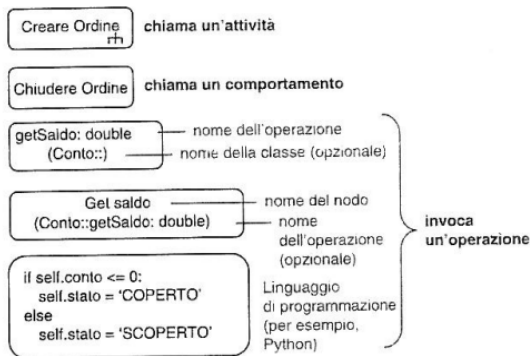
- i token vengono emessi su ogni arco in uscita se la post-condizione viene valutata a vero

## Tipologie di nodi azione:

- azione di chiamata
- invia segnale
- accettazione evento
- espressione temporale

# Nodi Azione di chiamata

- Può attivare:
  - attività
  - comportamento
  - operazione



# Nodi Azione Accettazione Evento Temporale

## Comportamento

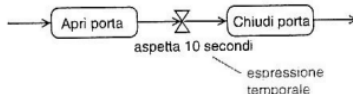
Il nodo ha un'espressione temporale e genera un token quando l'espressione diventa vera.

- ▶ un evento nel tempo (fine del mese)
- ▶ un punto nel tempo (7 novembre 2018)
- ▶ una durata (10 minuti)

- Con nessun flusso in ingresso:



- Con flusso in ingresso:



# Nodi Controllo

Servono a gestire il flusso di controllo:

- **nodo iniziale**
- nodo finale dell'attività
- nodo finale del flusso
- nodo decisione
- nodo fusione
- nodo biforcazione
- nodo ricongiunzione

# Nodi Controllo

Servono a gestire il flusso di controllo:

- nodo iniziale
- nodo finale dell'attività
- nodo finale del flusso
- nodo decisione
- nodo fusione
- nodo biforcazione
- nodo ricongiunzione



# Nodi Controllo

Servono a gestire il flusso di controllo:

- nodo iniziale
- nodo finale dell'attività
- **nodo finale del flusso**
- nodo decisione
- nodo fusione
- nodo biforcazione
- nodo ricongiunzione

# Nodi Controllo

Servono a gestire il flusso di controllo:

- nodo iniziale
- nodo finale dell'attività
- nodo finale del flusso
- **nodo decisione**
- nodo fusione
- nodo biforcazione
- nodo ricongiunzione

# Nodi Controllo

Servono a gestire il flusso di controllo:

- nodo iniziale
- nodo finale dell'attività
- nodo finale del flusso
- nodo decisione
- **nodo fusione**
- nodo biforcazione
- nodo ricongiunzione

# Nodi Controllo

Servono a gestire il flusso di controllo:

- nodo iniziale
- nodo finale dell'attività
- nodo finale del flusso
- nodo decisione
- nodo fusione
- **nodo biforcazione**
- nodo ricongiunzione

# Nodi Controllo

Servono a gestire il flusso di controllo:

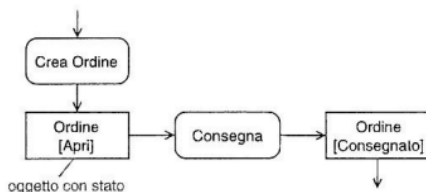
- nodo iniziale
- nodo finale dell'attività
- nodo finale del flusso
- nodo decisione
- nodo fusione
- nodo biforcazione
- **nodo ricongiunzione**

# Nodi oggetto

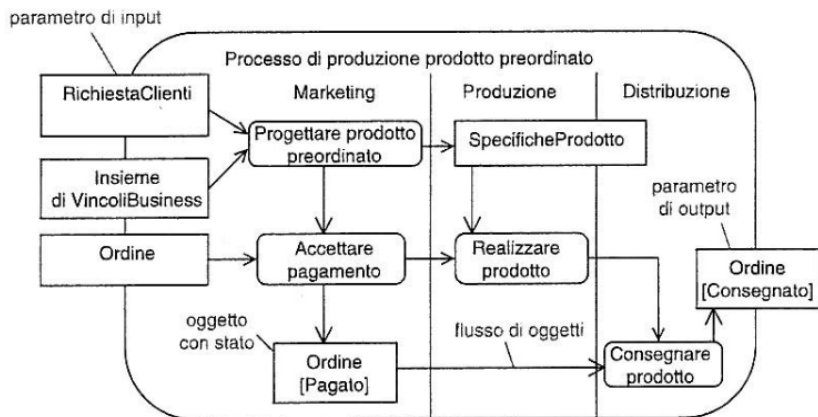
I nodi oggetto rappresentano la disponibilità di istanze di classificatori e fungono da buffer per i dati:

- è possibile specificare dimensione del buffer
- i nodi hanno ordinamento (FIFO default)
- comportamento di selezione - <<selezione>>

Rappresentazione dello stato degli oggetti



# Parametri di attività



# Pin

