

# Misurare Processi di Business

R.Bruni   A.Corradini   G.Ferrari   T.Flagella   R.Guanciale  
G.O.Spagnolo

10 novembre 2011

**Conferenza AICA 2011 - Torino**

## Business Process Management

- BPM affronta la modellazione, l'organizzazione, l'applicazione e l'ottimizzazione delle attività necessarie per raggiungere un determinato obiettivo (es. offrire un determinato servizio, oppure produrre un certo manufatto).
- In BPM, i processi vengono rappresentati attraverso formalismi grafici, permettendo di comunicare in modo non ambiguo le regole di business, e quindi discuterle o modificarle, tra gli svariati ruoli coinvolti che vanno dagli esperti del dominio di business o del settore, agli architetti software e sviluppatori.

## Obiettivi

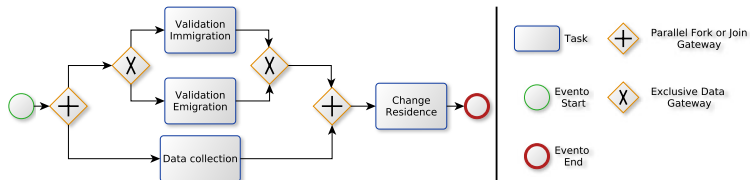
- In questo contributo ci focalizziamo su una specifica fase del BPM, che comprende il monitoraggio e la valutazione. L'obiettivo di questa fase consiste nel verificare la corretta esecuzione dei processi e misurarne le prestazioni dopo il loro deployment.

## Strategia

- Adottare ed estendere esistenti metodi formali (Petri Nets)
- Integrare ed estendere esistenti infrastrutture software (ProM)
- Metodologia dei work-flow
  - ① I processi sono descritti con BPMN diagrams
  - ② Il BPMN diagram viene trasformato in una Petri Net
  - ③ I log delle istanze di processo sono processati usando tecniche disponibili per le Petri Net
  - ④ I risultati delle analisi sono proiettati indietro sul modello di partenza BPMN.

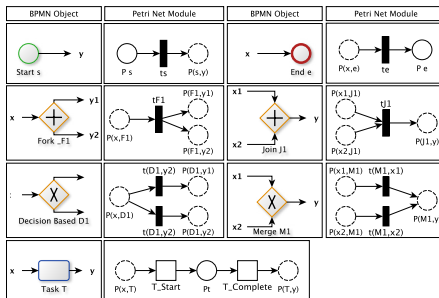
## Esempio di modello BPMN

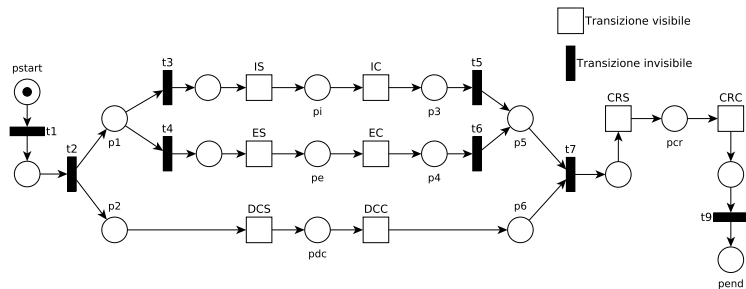
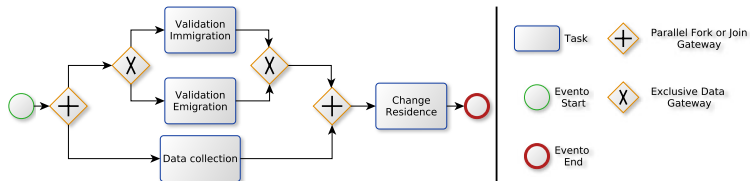
- Permette di modellare i processi ad un alto livello di astrazione, questo modello può essere compreso o creato anche dai non addetti ai lavori.



## From BPMN to Petri Net

- Sfruttiamo una metodologia di trasformazione esistente (Dijkman, R.M., Dumas, M., Ouyang, C.) estesa
- Successivamente affrontiamo il problema di riportare i risultati di queste analisi sul modello BPMN di partenza.





## Analisi basata su Petri Nets

- Gli eventi delle istanze di processo del log sono ordinati (e.s. timestamp)
- Gli eventi sono mappati sulle transizioni della rete
- **Log Replay**: replay delle istanze di processo del log (non-blocking way)
  - 1 L'algoritmo parte con un token nella piazza iniziale delle rete
  - 2 Estrae dalla testa del log l'evento
  - 3 Viene effettuato il fire della corrispondente transizione
    - Se la transizione non è abilitata i token mancanti vengono creati artificialmente e chiamati **missing token**
- Metriche
  - Il numero di missing/remaining token per ogni piazza/transizione
  - Il numero di archi attraversati
  - Il tempo di soggiorno/attesa/sincronizzazione per ogni piazza.

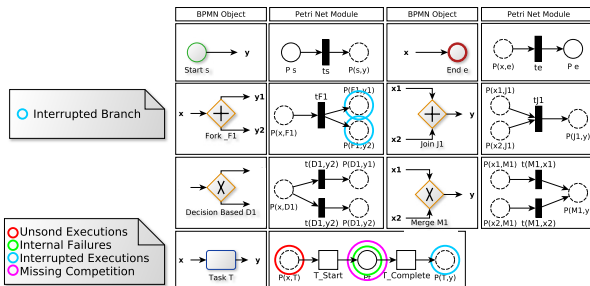
## Analisi di Performance

- L'idea è di calcolare l'intervallo di tempo tra produzione e consumo di token in ogni piazza della rete.
- Fruttando il log-replay si possono calcolare le metriche di performance.
  - Si parte dall'algoritmo standard di log replay che produce la lista delle transizioni
  - Si trasforma la lista di transizioni in una sequenza "eager"  $R = [tr_1, \dots, tr_n]$  per ogni  $tr_i$  transizione invisibile
    - Sia  $tr_p$  l'ultima ( $p < i$ ) transizione visibile
    - $\bullet tr_i \cap tr_p \bullet \neq \emptyset$
  - Per ogni transizione invisibile  $tr_i$ 
    - 1 spostato a sinistra la transizione fino a quando non trovo una transizione visibile  $\bullet tr_i \cap tr_p \bullet \neq \emptyset$

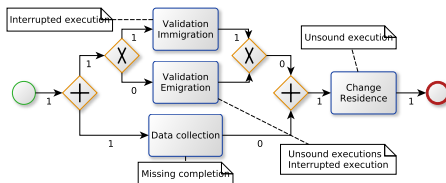
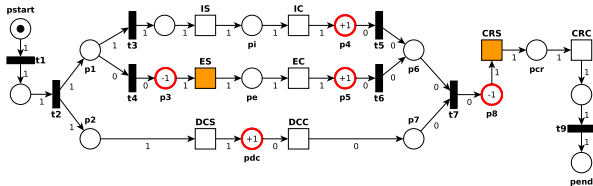


## Dall'analisi al BPMN (Conformance)

- **Missing tokens:** Log replay produce missing tokens solo per il fire delle transizioni visibili  $\Rightarrow$  pre-set di una transizione visibile
- **Remaining tokens** Le transizioni invisibili sono fired solo se richiesto da una successiva transizione visibile  $\Rightarrow$  nei place post-set di una transizione visibile o di una transizione invisibile che generano più di un token.

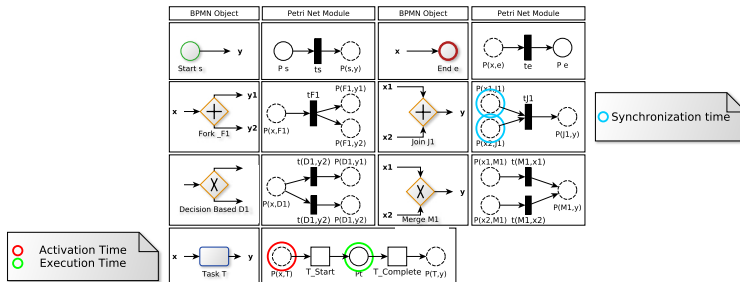


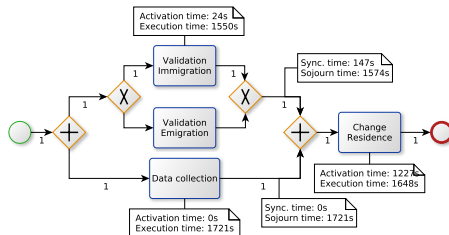
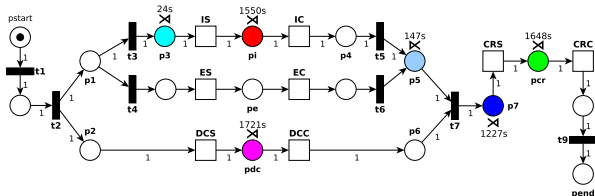
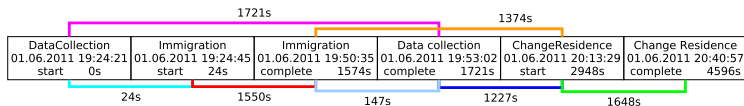
Immigration #1 start 01.06.2011 17:14:56	Immigration #2 complete 01.06.2011 17:31:19	Emigration #3 start 01.06.2011 18:24:33	Emigration #4 complete 01.06.2011 18:37:10	Data collection #5 start 01.06.2011 18:25:27	Change Residence #6 start 01.06.2011 18:35:21	Change Residence #7 complete 01.06.2011 18:42:41
--	---	---	--	--	---	--



## Dall'analisi al BPMN (Performance)

- **Tempo di Attesa**: transizione invisibile è fired immediatamente  $\Rightarrow$  nei place pre-set delle transizioni visibili
- **Tempo di Sincronizzazione** nei place che hanno almeno una transizione nel loro post-set che dipende da un altro place.





## Risultati teorici

- Affinate le tecniche per gestire le transizioni invisibili sulle Petri Net
- Proiezioni delle misure sul BPMN

## Risultati di sviluppo

- Nuovi plug-ins per ProM
  - Trasformazione del modello BPMN in una Petri Net
  - Analisi di Performance e Conformance su Petri Net
  - Proiezioni delle misure di analisi sul modello BPMN di partenza.

