

Introduzione a BPMN

versione 9 ottobre 2011

©Adriano Comai

<http://www.analisi-disegno.com>

Obiettivo di questa introduzione

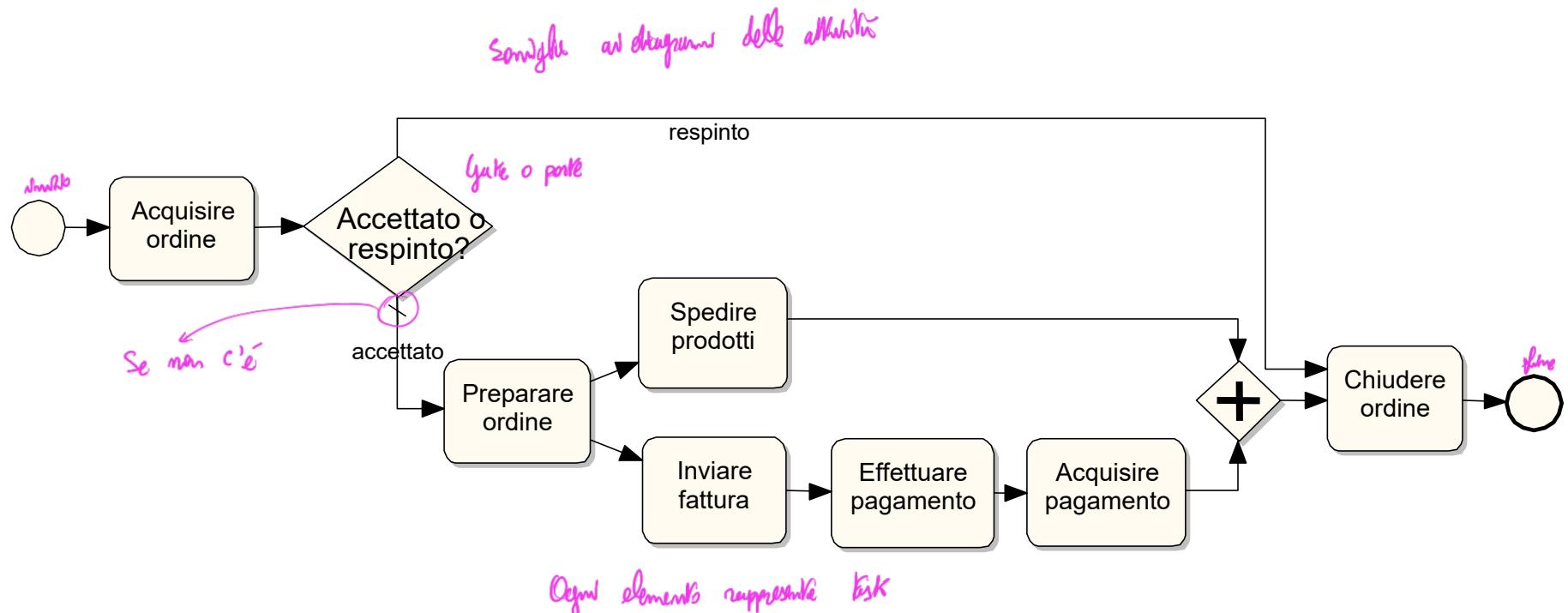
- fornire elementi di base su BPMN
- il tema è trattato in modo approfondito, con esercitazioni, nel corso "Analisi dei processi di business":

http://www.analisi-disegno.com/a_comai/corsi/sk_bm.htm

Business Process Model and Notation (BPMN)

- creato dalla Business Process Management Initiative, poi confluita in OMG (Object Management Group)
- versione 1.0 OMG: 2006
- versione 1.1: 2007
- versione 1.2: 2009
- **versione 2.0: 2011**
- documenti:
 - <http://www.omg.org>

BPMN - esempio



Obiettivi di BPMN

- Fornire una notazione comprensibile da:
 - analisti che definiscono i processi (*analisi dei requisiti*)
 - sviluppatori responsabili dell’implementazione tecnologica dei processi
 - “persone del business” che gestiranno e terranno sotto controllo i processi (*per esempio processi interni dell’azienda; dimensione collaborazione dello stakeholder*)
- Far sì che i linguaggi nati per l’esecuzione dei processi di business (es. BPEL) possano essere visualizzati con una notazione “non tecnica”

BPEL

- Business Process Execution Language for Web Services (IBM / Microsoft / Bea / SAP / Siebel nel 2002)
- Ideato per definire i processi di business e per permettere la loro interoperabilità a livello software
- Poco adatto al diretto utilizzo umano per progettare, gestire e tenere sotto controllo i processi di business (BPMN, invece, serve proprio a questo)

Vantaggi di BPMN

- Permette alle organizzazioni di rappresentare i propri processi con una notazione intuitiva (flow chart)
- La standardizzazione agevola la comunicazione (anche nei confronti di organizzazioni esterne)
- Permette di rappresentare in modo comprensibile anche costrutti definiti nel linguaggio di esecuzione software

BPMN è mirato ai processi

BPMN serve a modellare solo i processi. Non a rappresentare altri aspetti importanti:

- Strutture organizzative, ruoli, responsabilità
- Scomposizioni funzionali
- Modelli di dati e informazioni
- Regole di business
- Strategie dell'organizzazione

Diagrammi BPMN

Sistema di gestione bancaria della carta

3 modelli diversi

- Possono rappresentare diversi tipi di modelli di processo:

– privato non vedo cosa accade al loro interno: arrivano messaggi e vedo risposte

Modo in cui richiede di fare qualcosa
che viene gestita

– pubblico vedo tutto lo interno
tra me ed il sistema che sta usando.
Es. scommessa

– collaborazione

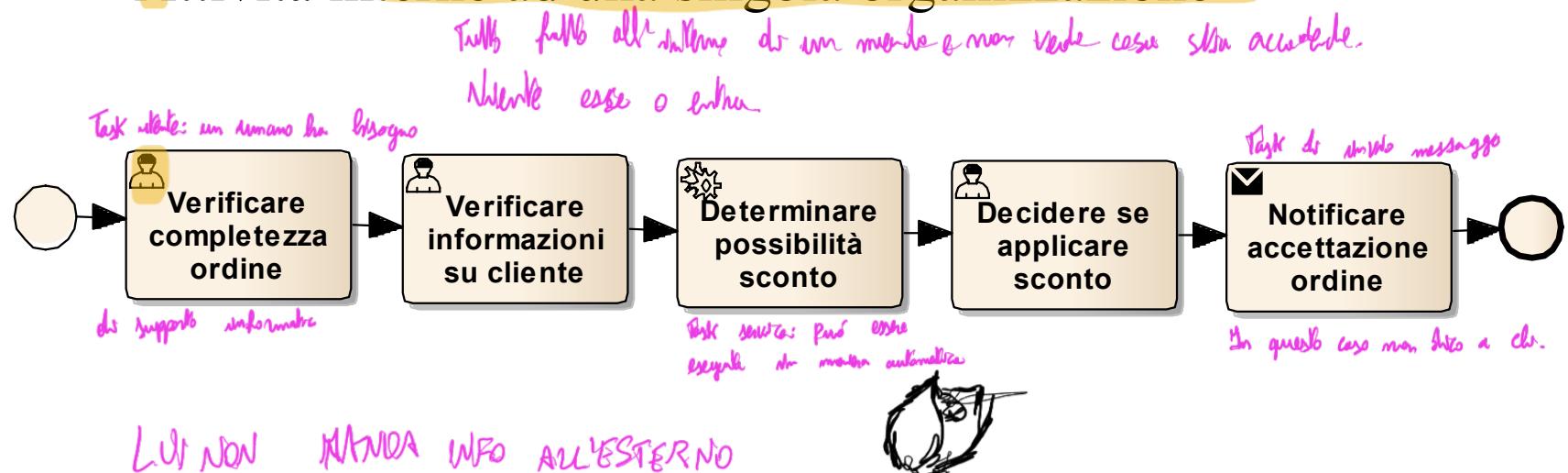
• coreografia → i vari oggetti lavorano comunicando fra loro senza controllo specifico su cosa accade. Riceve un messaggio su quale è folla.

• conversazione → sta continuamente ad aspettare una risposta dell'altra parte.

Es: per ogni cosa che fa la scommessa sono avvisato e dà feedback.
Sono in genere lenti e complessi da imparare.

Private (Internal) Process

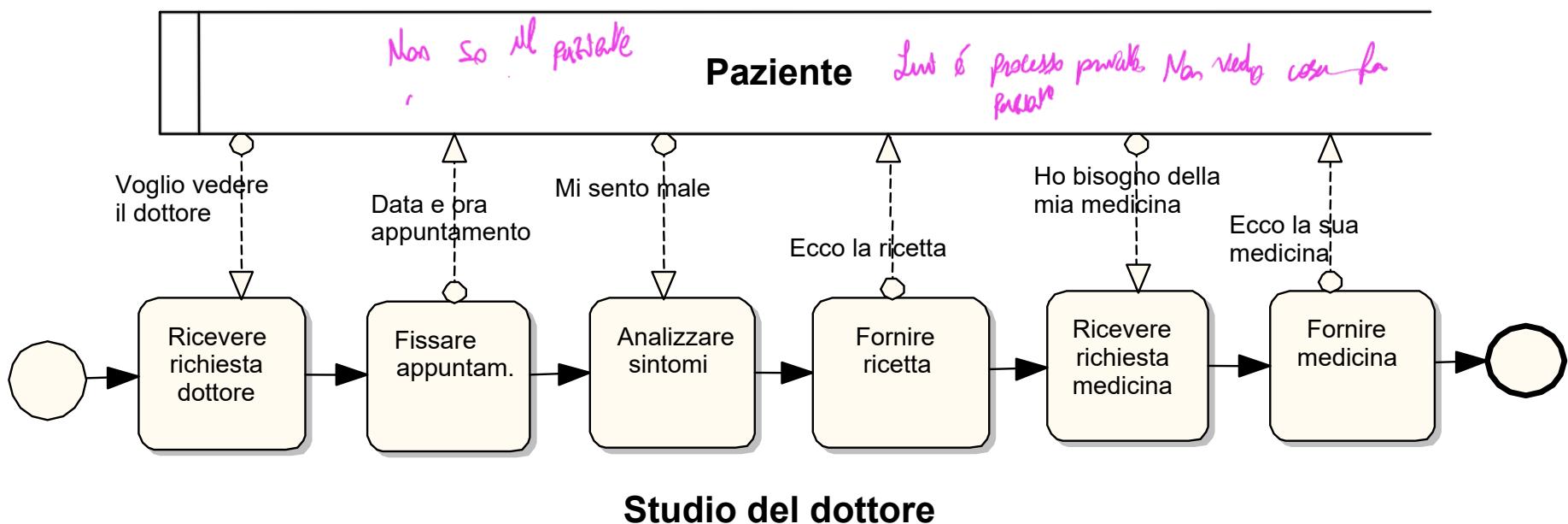
- Attività interne ad una singola organizzazione



Public Process

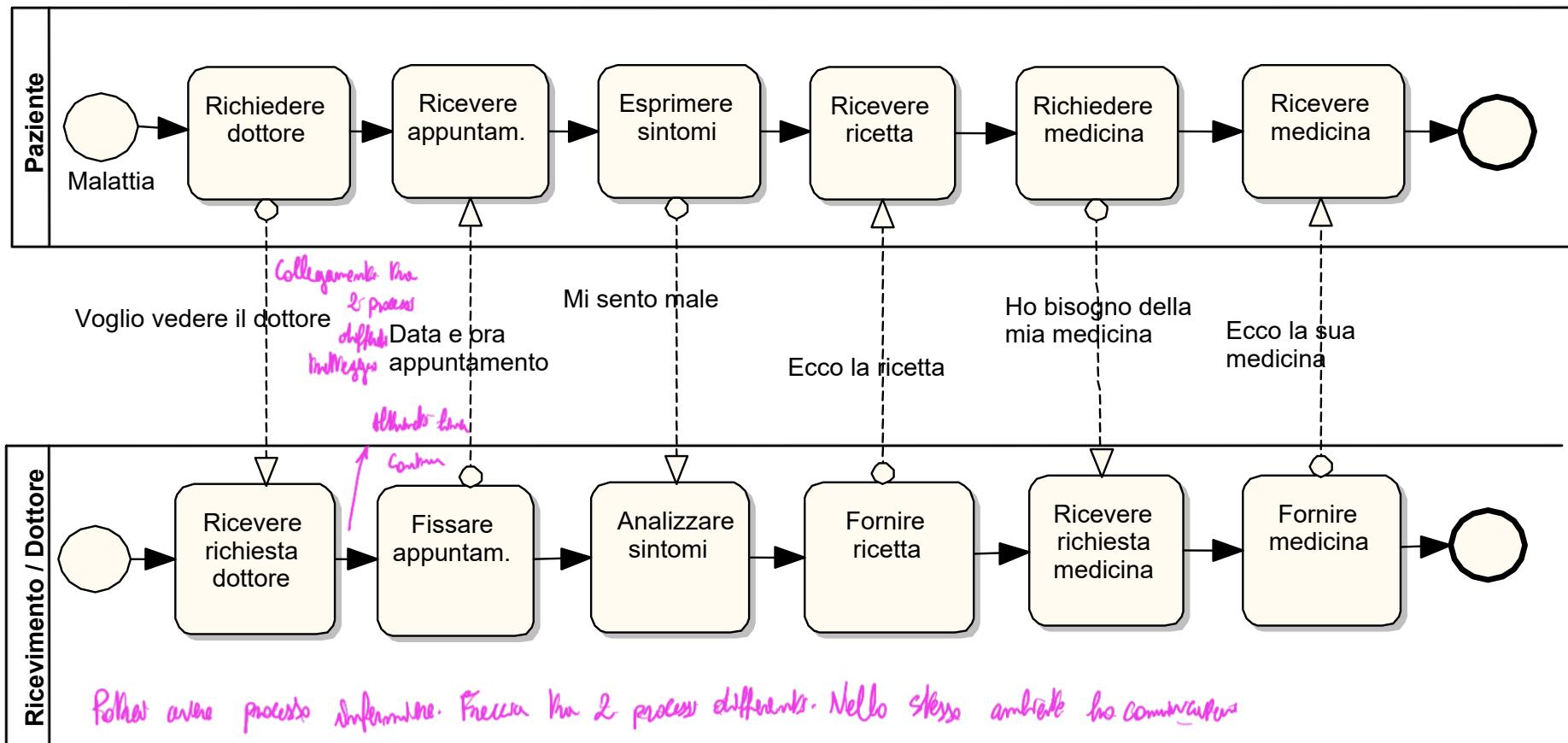
Quello che avviene esse fai confini del processo.

- Vengono evidenziate solo le attività necessarie a comunicare verso entità esterne, ed i relativi messaggi



Collaboration Process

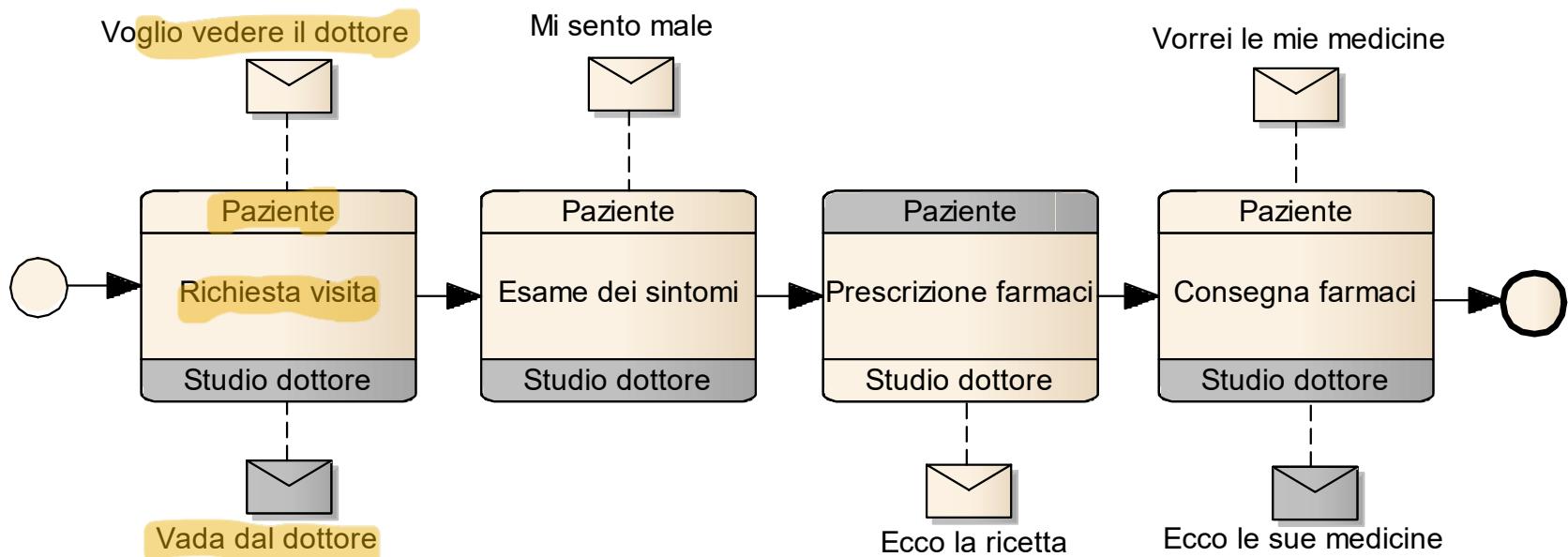
- Rappresenta le interazioni tra due o più processi pubblici



*Affusta' non risposta
domanda. Fa'ci' niente
e qualcosa fa'ci' o per' aiutandoci*

Coreografia

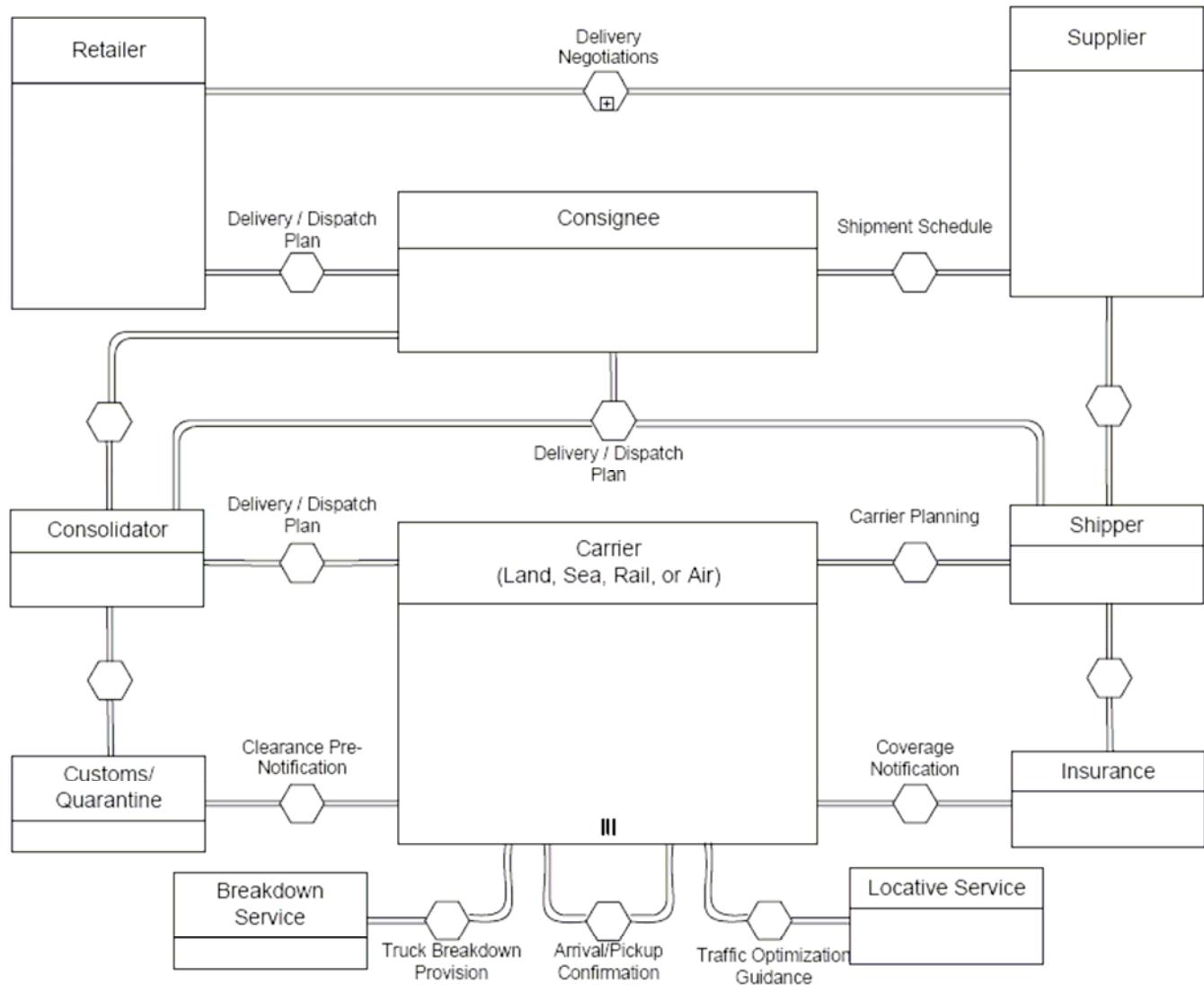
- Rappresentazione speciale di una collaborazione.
- Ogni attività rappresenta un'interazione tra più partecipanti, basata sullo scambio di messaggi. Non esiste un coordinamento centrale.



Conversazione

Rettangoli:
partecipanti

Esagoni:
collaborazioni



- Flow Objects (oggetti del flusso)
 - Events (low watermark)
 - Activities
 - Gateways (connette 2 attività dello stesso flusso)
- Data Objects (oggetti attaccati ad un messaggio (es. oggetto fattura))
- Connecting Objects (connettori)
 - Sequence Flow } Nelle step processi
 - Message Flow } Le frecce (scambi tra processi differenti) [chiavi che sta inviando o ricevendo messaggio]
 - Association } Collegamento tra processi differenti per spiegare che tipo di associazione
- Swimlanes (partizioni)
 - Pool (un pool processi)
 - Lane Attivita a cui è associato per un processo
- Artifacts
 - Group Disegni da utilizzare per spiegare come fare attività
 - Annotation

Elementi base (core set)

Eventi

- Gli eventi “accadono” nell’ambito di un processo (e hanno una rilevanza per la sua esecuzione). 3 tipologie:
 - Start event (inizio) – indica il punto di partenza di un processo
 - Intermediate event (intermedio) – può avvenire tra l’evento iniziale e quello finale *Rimangendo bloccato finché non si verifica*
 - End event – indica la conclusione di un processo

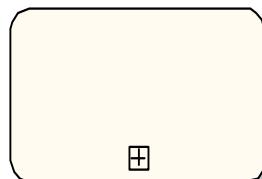


Attività

- Può essere **atomica** o **composta**:
 - Task – un'attività atomica, non scomposta

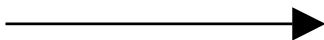


- Sub-Process – un'attività composta. Può venirne evidenziato il contenuto, oppure essere rappresentata in modo sintetico (+)



Connettori

- **flusso di sequenza** – indica l'ordine di svolgimento delle attività



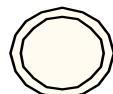
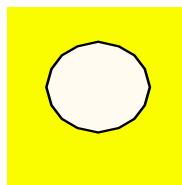
- **flusso di messaggio** – comunicazione tra due partecipanti (pools)
due pool diverse. Non sarebbero messagg, lo scambio di info avviene con messagg di flusso.
evento che invia messaggio
- **associazione** – utilizzata per tutti gli altri legami tra gli oggetti del diagramma



Start event

Evento di inizio

- Opzionale
- Se non c'è, tutte le attività che non hanno un sequence flow in ingresso partono insieme
- Necessario se c'è un End event
- Può essercene più di uno (ma il modello si complica)



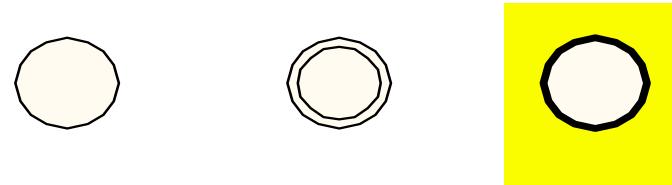
con più di uno partono contemporaneamente!

Start event può ricevere messaggio in ingresso ma non può mandare un output
↓ solo dopo ricezione di un messaggio

End event

Evento finale

- Opzionale
- Se non c'è, tutte le attività che non hanno un sequence flow in uscita devono terminare per concludere il processo
- Necessario se c'è uno Start event *Mettendo SEMPRE*
- Può essercene più di uno (corrispondono a esiti diversi del processo)
- Può essere la destinazione di più sequence flow
- Non può essere destinazione di un message flow

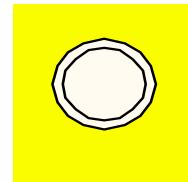
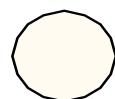


La fine può inviare un messaggio

*Nessun messaggio
in ingresso
può comunque all'end event*

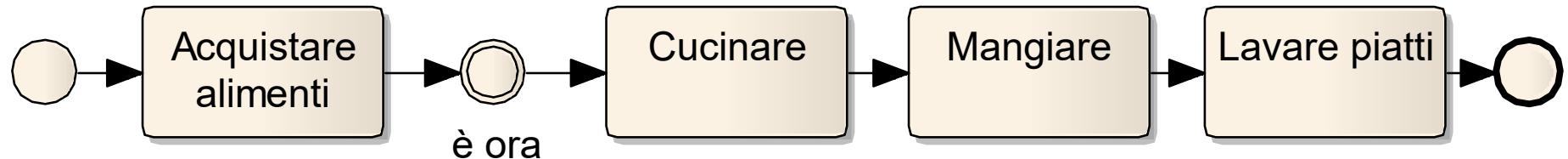
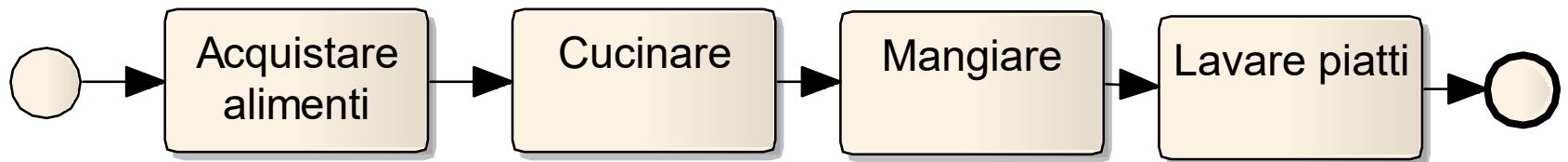
Intermediate event

Evento intermedio



- messaggi
- interruzioni e ritardi
- gestione eccezioni
- gestione compensazioni

Eventi e attività

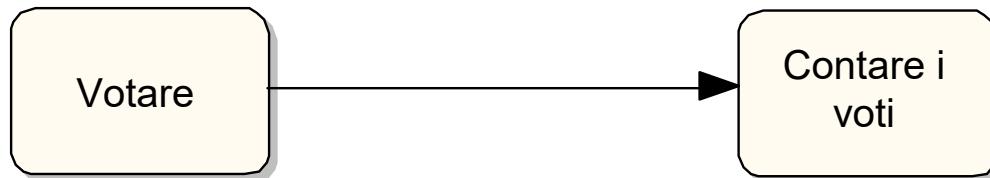


è ora

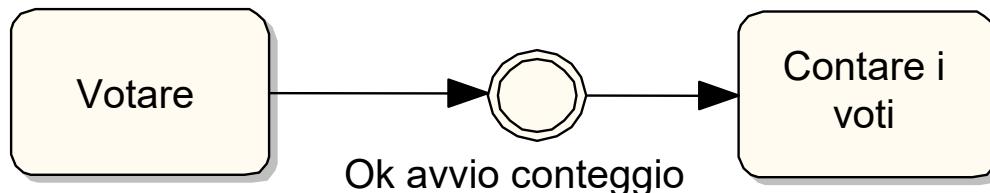
eventi intermedi. Quando l'attività si blocca su eventi finiti
a quando non si ripete.

Evento intermedio - pausa

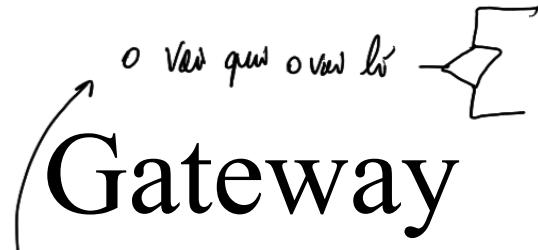
- Quando due attività sono collegate direttamente, la seconda può iniziare dopo la conclusione della prima



- Si può separarle con un evento intermedio: il processo viene sospeso e riprenderà in seguito al manifestarsi dell'evento

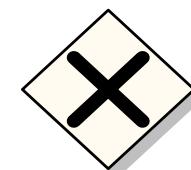
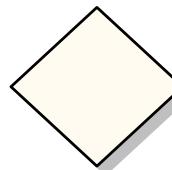


- esprimono la convergenza e la divergenza dei flussi di sequenza

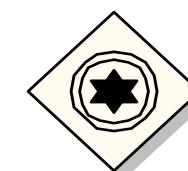


Esclusivo (Menge: ne basta uno) Ne affia sempre 1 solo
(XOR)

basato sui dati

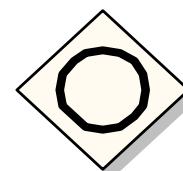


basato su eventi



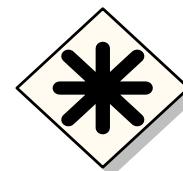
Ho associato un evento, le uscite alternate.
Solo se si realizza un certo evento. Solo 1 uscita.
Voglio almeno 1 evento per poter proseguire.

Inclusivo
(OR)



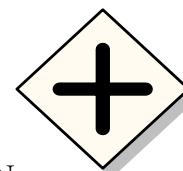
Ne può affiancare più di uno volendo

Complesso



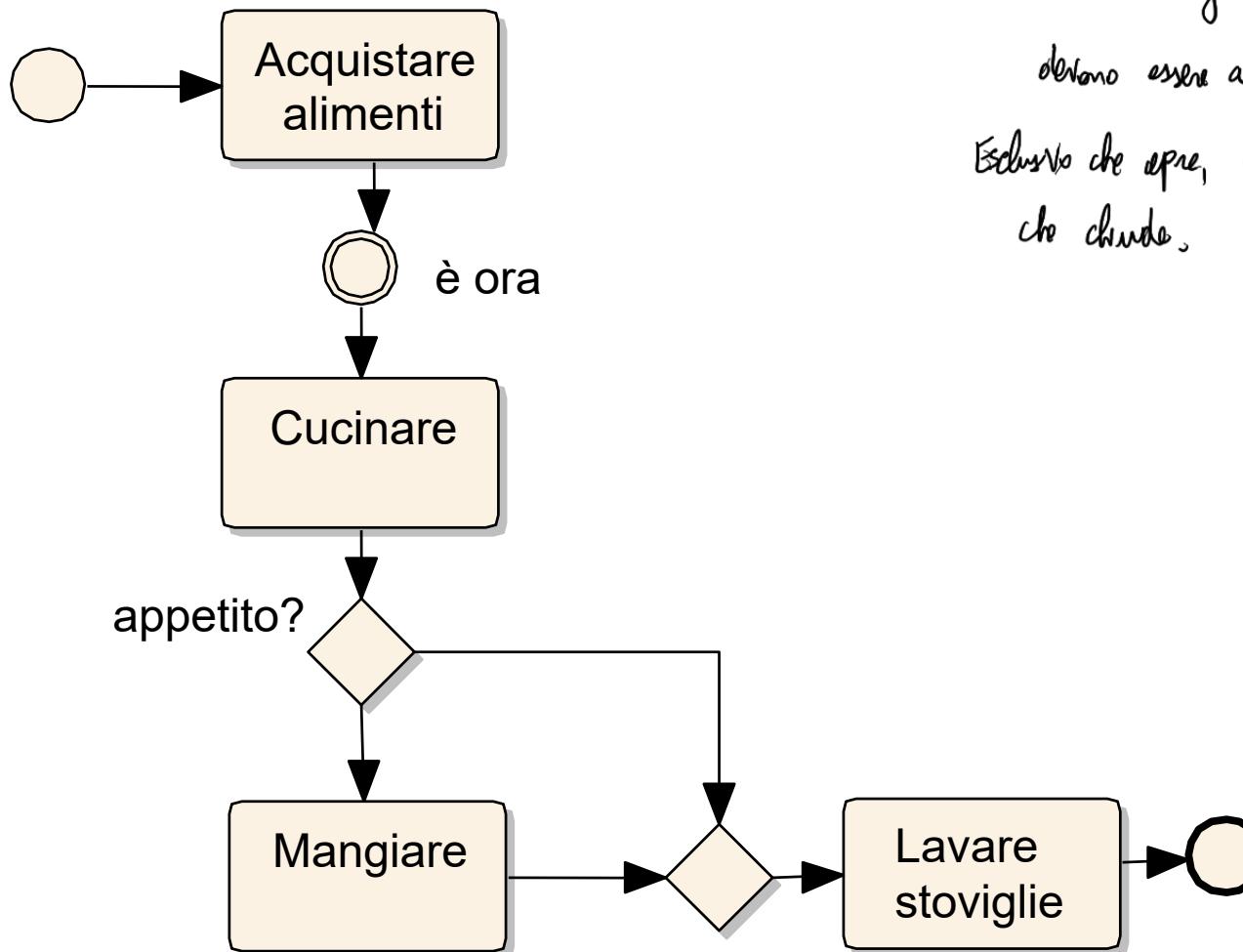
Mix: si attiva per un evento, per un segnale ecc.

Parallelo (Sohm)
(AND)



Si attivano tutti insieme sempre, con tutti in parallelo

Gateway



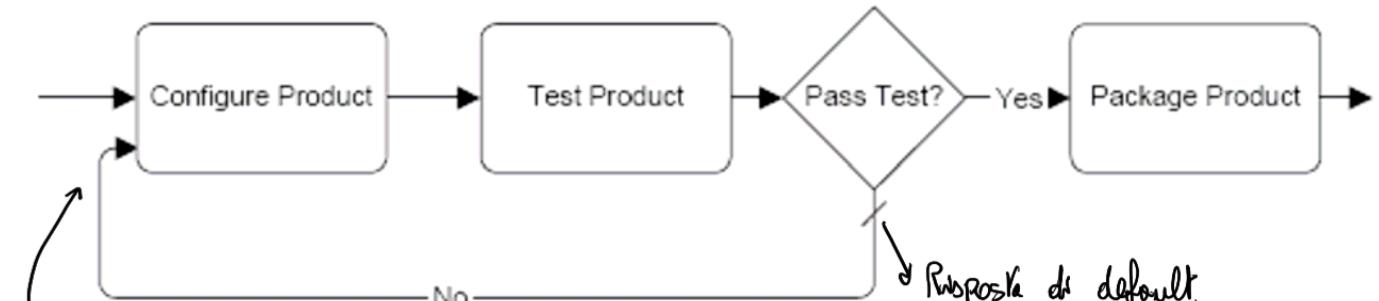
Note: gateway

dovono essere esclusivi

Esclusivo che apre, esclusivo
che chiude,

Looping

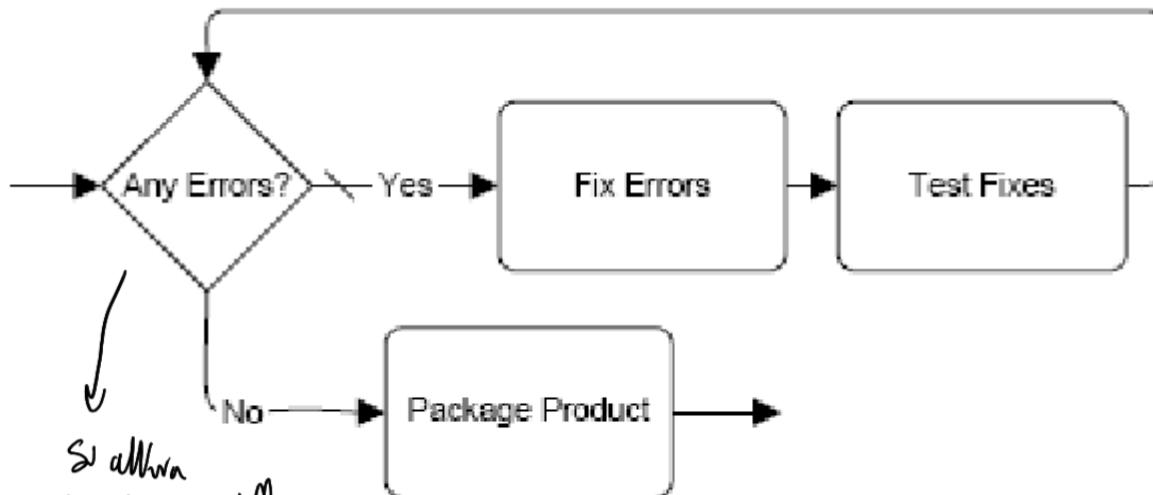
Loop until



No gate. Consente di allontanare si accade
o con una fallita o un'altra. Non gate \Rightarrow fa non conferendoti
esclusivo con 2 input e 1 output mette la gate

Risposta di default,
magari non ha senso che
donna risposta scava.

Loop while



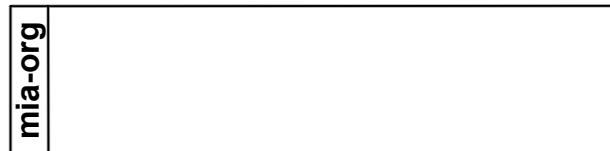
Sì allora
prendo una delle 2
si allora e dà una risposta esclusiva. \Rightarrow Qui gate si richiede su sé stessa.

2 input 1 result, 1 input 2 result

Swimlane (partizioni)

Funziona le attività all'interno del pool e permette l'uso di sequence flow per fare comunicare le attività

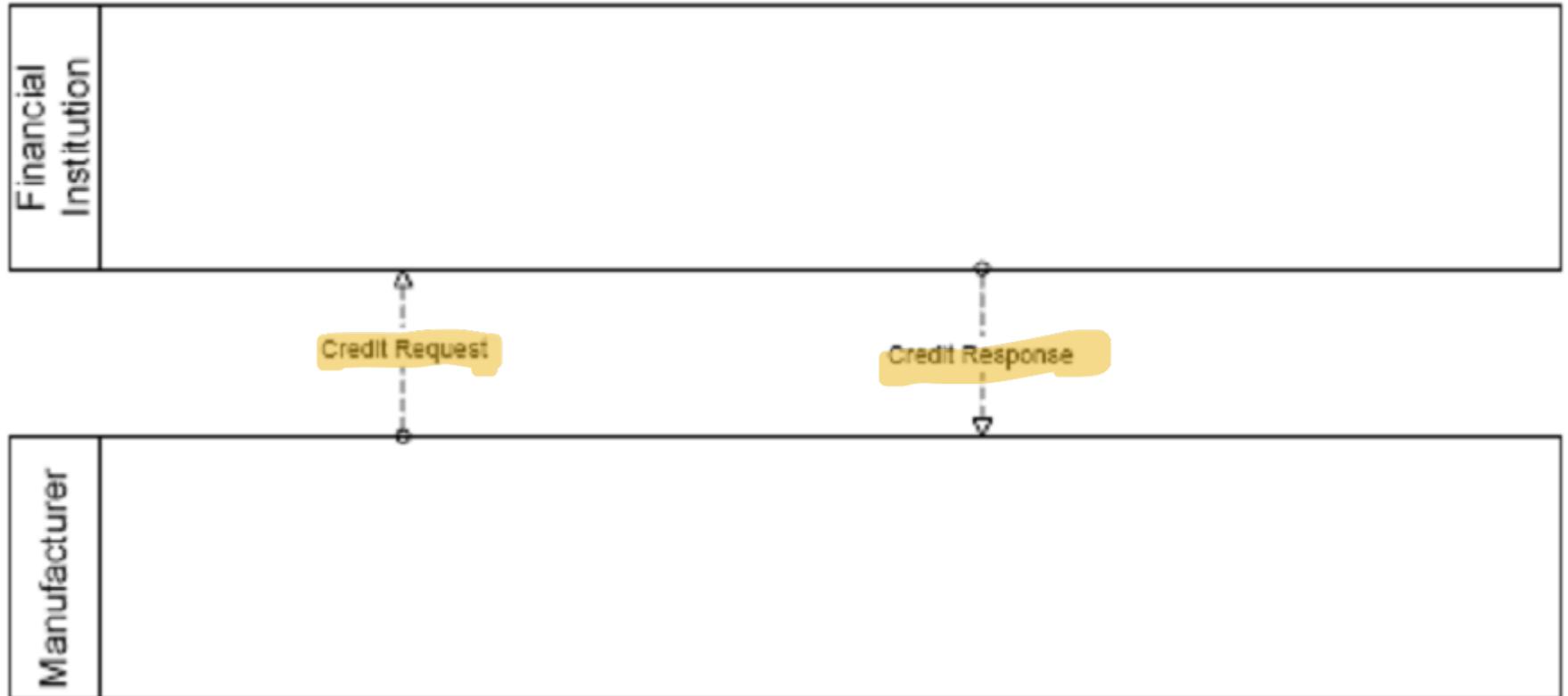
- Pool – rappresenta un partecipante al processo, e può contenere un insieme di attività



- Lanes – partizioni all'interno di un pool per attribuirne le attività con maggiore precisione



Messaggi

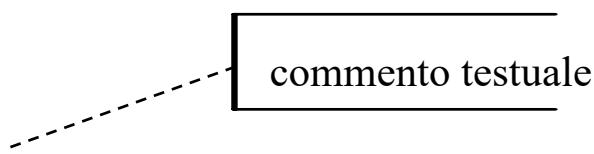


Artifacts

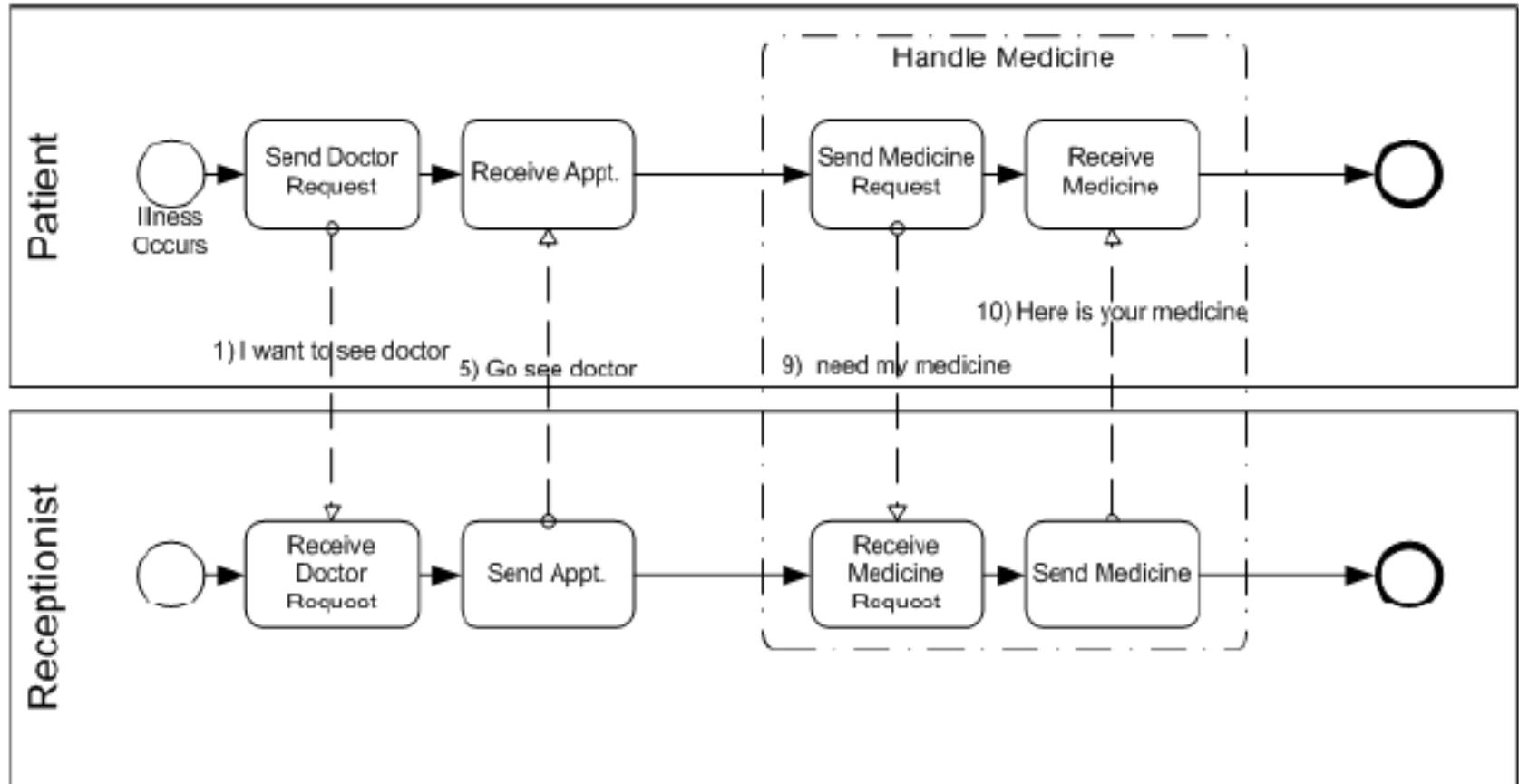
- Group – raggruppamento utilizzabile per evidenziare un insieme di attività



- Annotazione

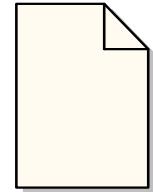


Gruppi



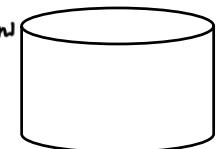
Data objects

- Data Object – un input o un output per una attività

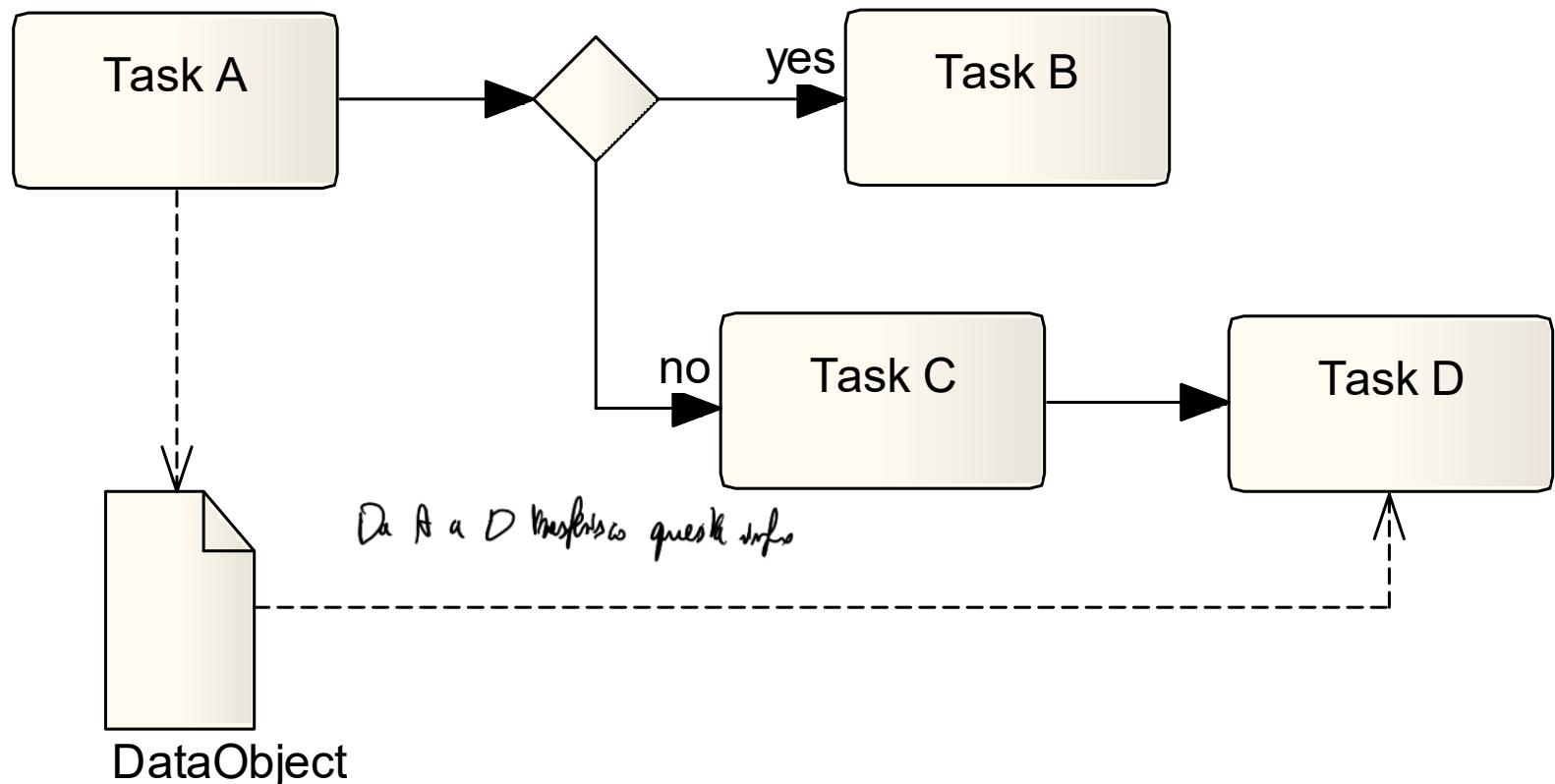


- Data Store – un deposito di dati (o un sistema) che persiste oltre la conclusione del processo

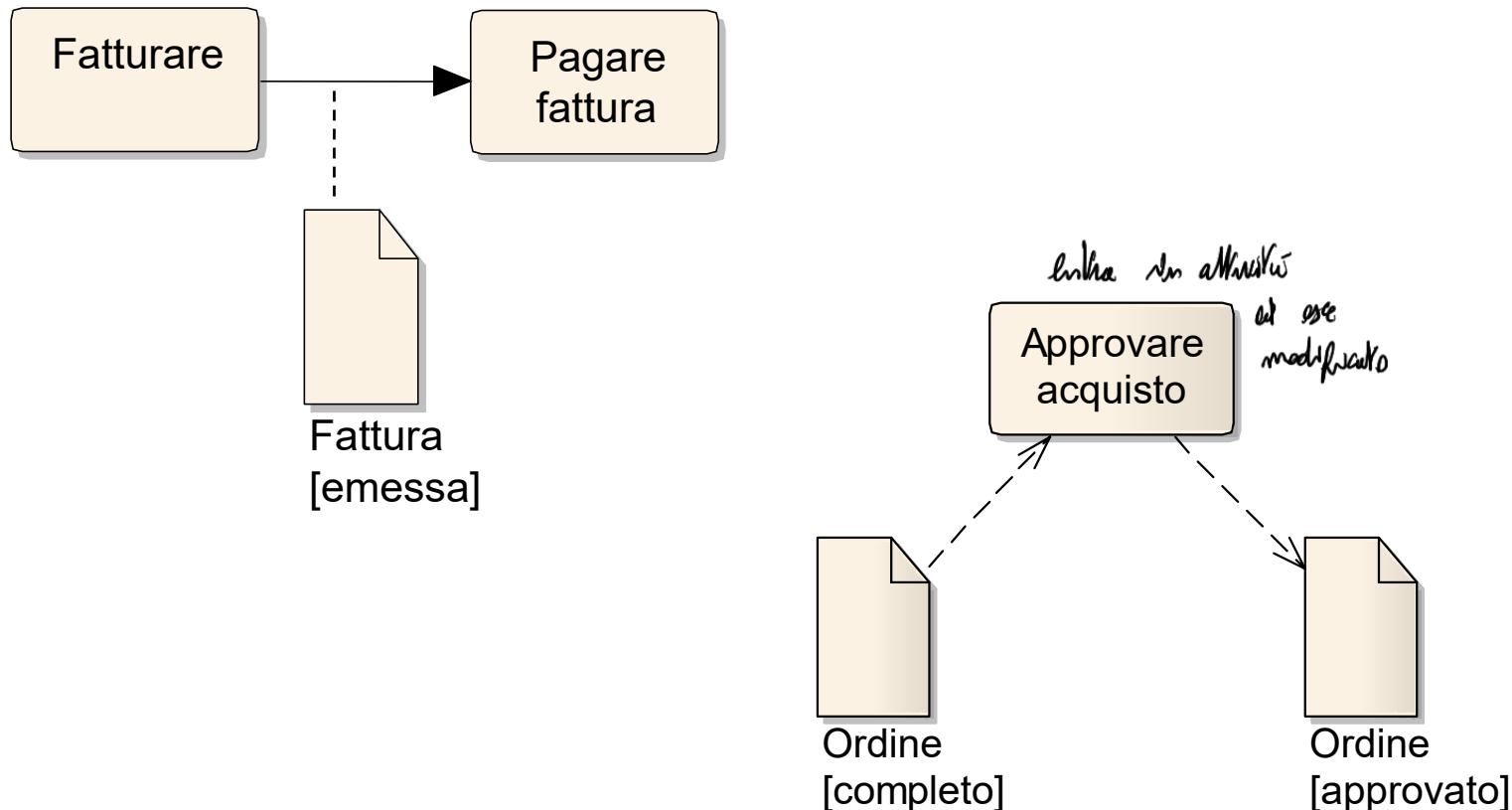
Da indicare come
magazzino per i
dati con database



Dati e attività



Dati e attività



Costrutti ulteriori – extended set

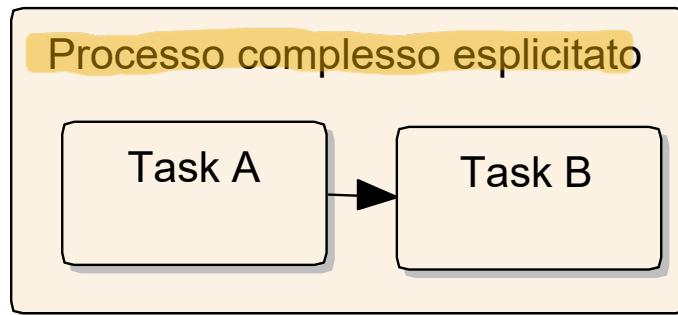
- La notazione basata sugli elementi base di BPMN (“core set”) è semplice e intuitiva
- Permette di modellare adeguatamente la maggioranza dei processi
- Per esigenze di modellazione più precise, è disponibile una serie di costrutti avanzati (“extended set”), che specializzano quelli base

Tipi di attività

Se ho un processo non esplicitato, posso definirlo da qualche altra parte più specificando.



elements
non abitati
che puoi
essere
ulteriormente
esplicitati

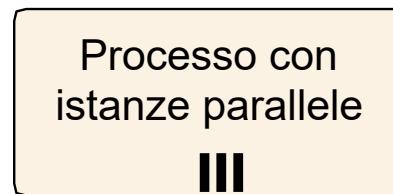


↓ Non ho eventi (se non attivo)

né messaggi: non puoi comunicare
all'esterno



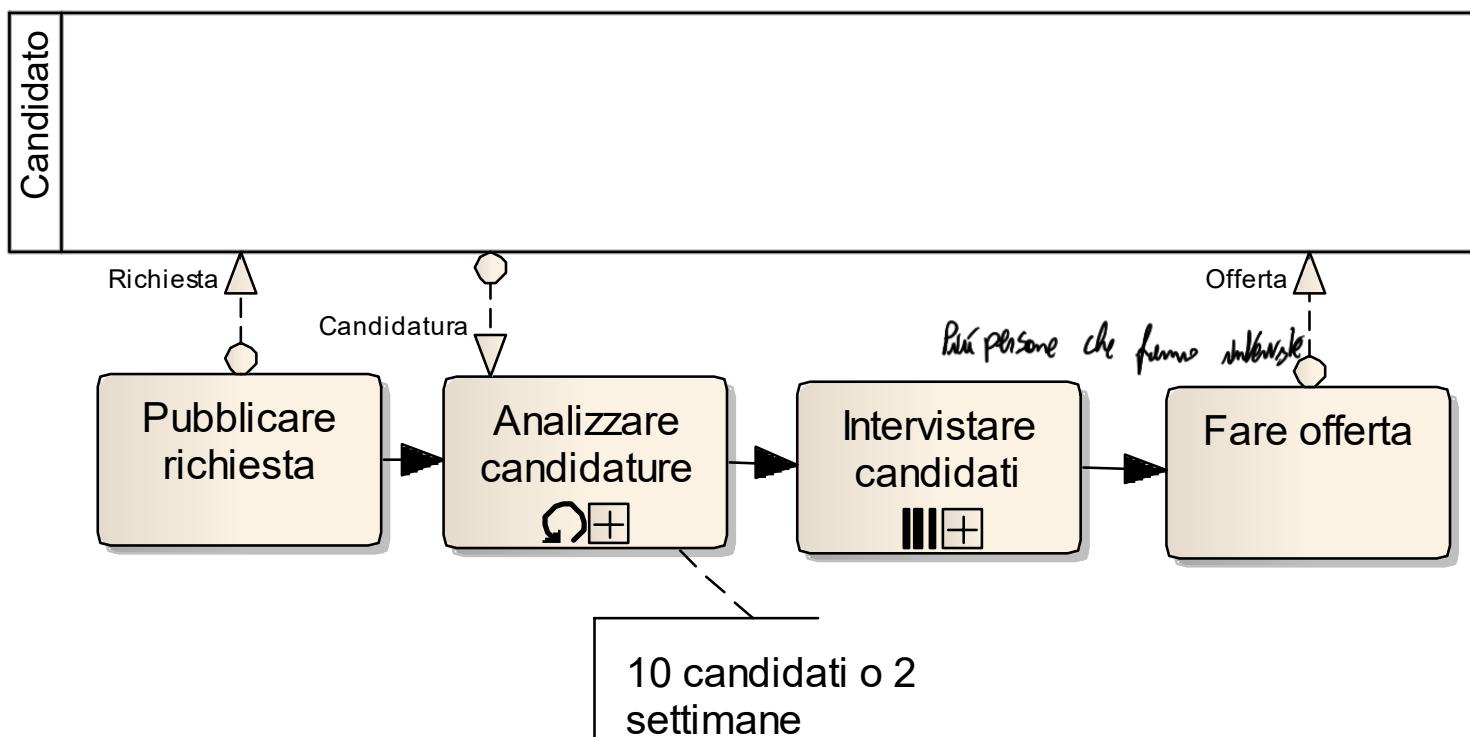
Qualcosa che viene ricoperto.
Indico anche queste volte, volte.



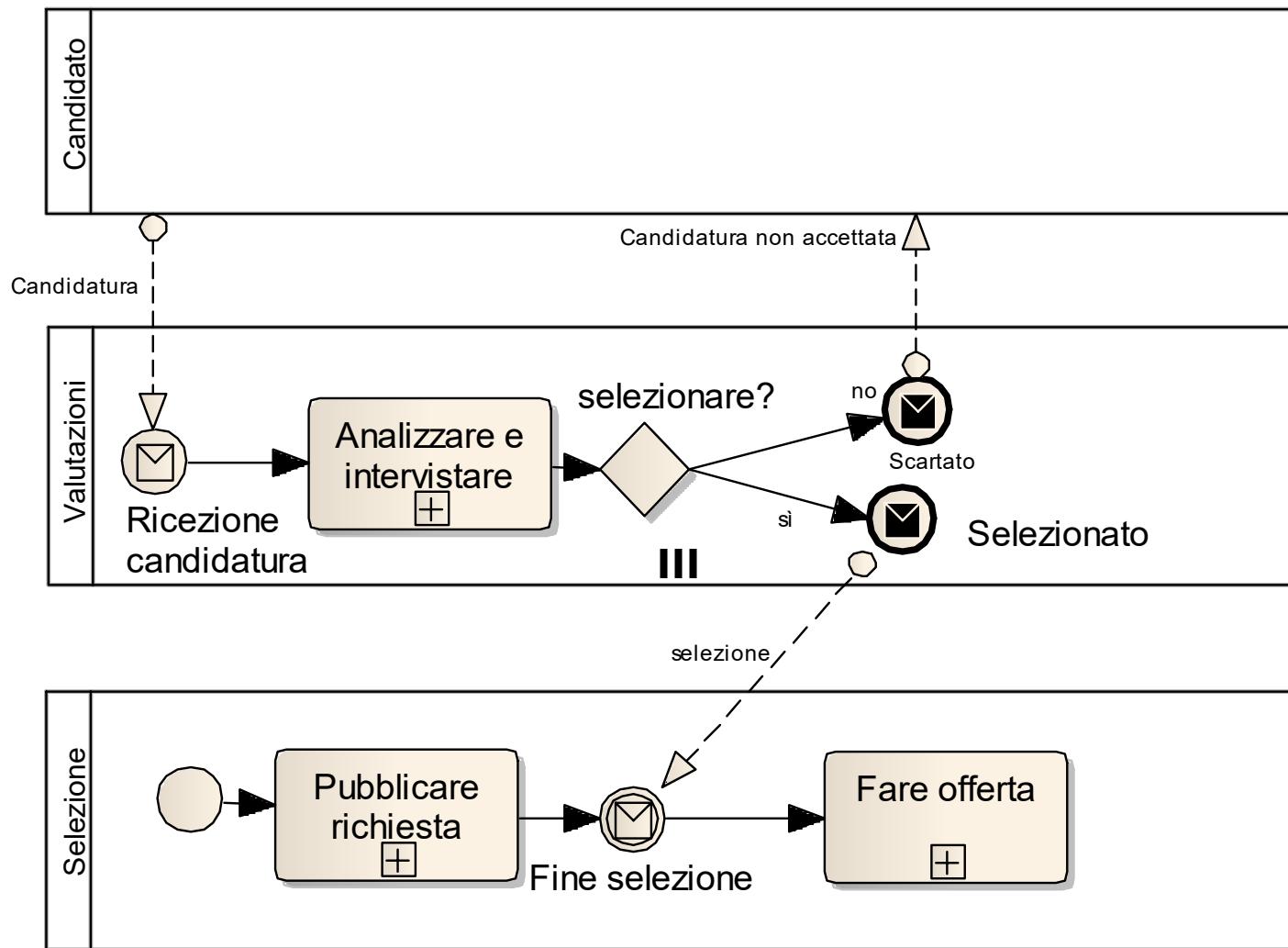
↑
Stesso processo
con più istanze
parallele (3 pc per mail)

Tutto questo anche sovrapposto

Sequenza e multiistanza

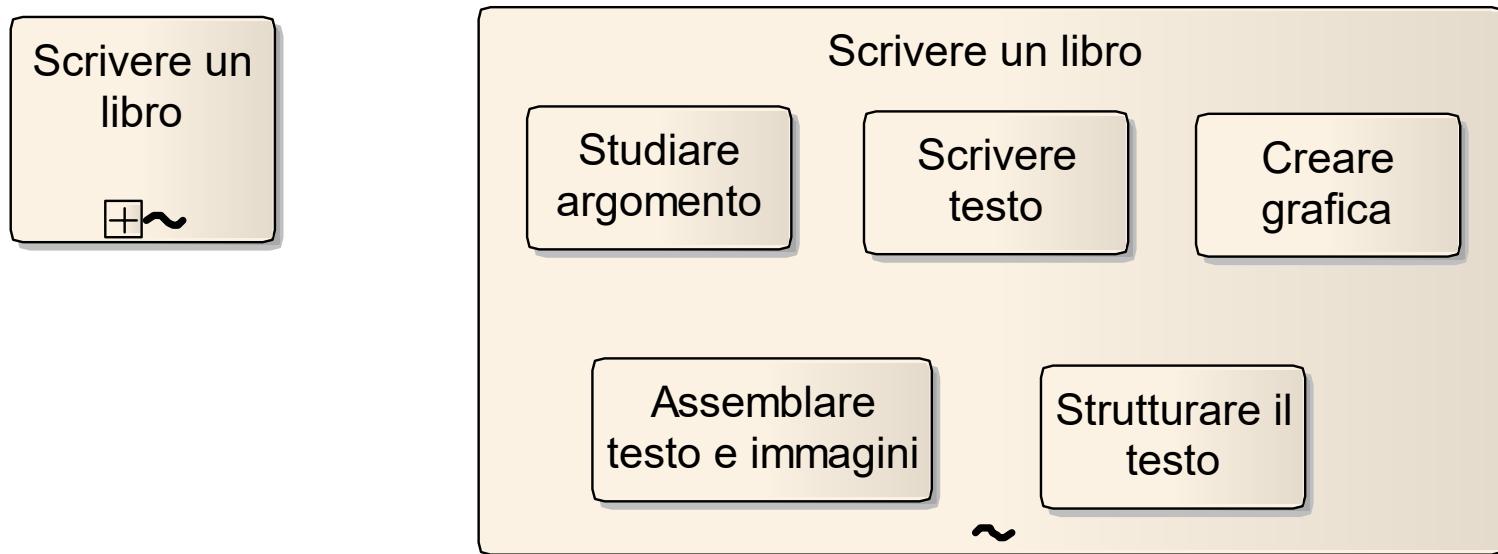


Pool multiistanza



Processo ad hoc

- È formato da attività prive di una sequenza predefinita

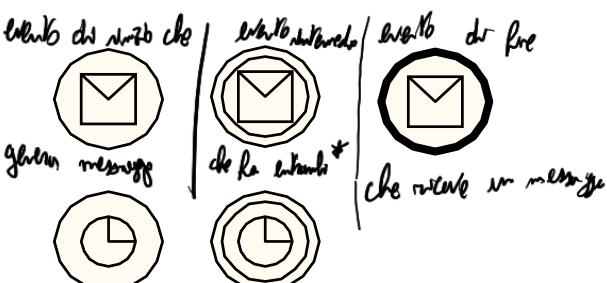


Eventi - tipologie

- Gli eventi di inizio e intermedi hanno un “trigger”, che ne definisce la causa
- Gli eventi di fine possono avere un “risultato”, cioè una conseguenza

eventi basati su Message

Timer



Error

Escalation

Cancel

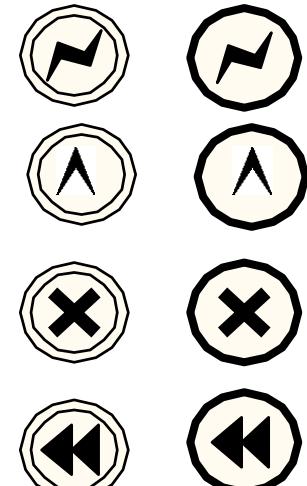
Compensation

Rule

Link

Multiple

Terminate



* Bush non se riceve, finisce n.

Start event: tipologie

| | Icona | Significato |
|-------------|---|--|
| None |  | Non specificato. Il diagramma di un sottoprocesso può avere solo questo tipo di start event. |
| Message |  | L'inizio è conseguenza dell'arrivo di un messaggio. |
| Timer |  | L'inizio avviene con periodicità predefinita. |
| Conditional |  | L'inizio avviene a fronte di una condizione (es. "temperatura > 300°") <small>hai parlato raffreddamento</small> |
| Signal |  | L'inizio avviene a fronte della ricezione di un segnale (che a differenza di un messaggio, è pubblico). |
| Multiple |  | L'inizio può avvenire a fronte di eventi diversi. |

End event: tipologie (risultato)

| | Icona | Significato |
|---|-------|--|
| None | | Non specificato. |
| Message | | A conclusione del processo viene inviato un messaggio. |
| Error | | A conclusione del processo viene generato un errore. |
| Escalation | | A conclusione del processo viene generato una escalation. |
| Cancel | | Solo nell'ambito di una transazione, cancella le variazioni. |
| Compensation | | Indica la necessità di una compensazione. |
| Signal | | A conclusione del processo viene inviato un segnale. |
| Terminate <small>Blocca VML, molti fatti</small> | | Termina ogni attività, anche parallele o multiistanza. |
| Multiple | | A conclusione del processo vi sono più conseguenze. |

* Possono essere messi dentro all'attività.
Se l'evento viene eseguito prima della
conclusione dell'attività segue flusso alternativo

Intermediate event: tipologie (1)

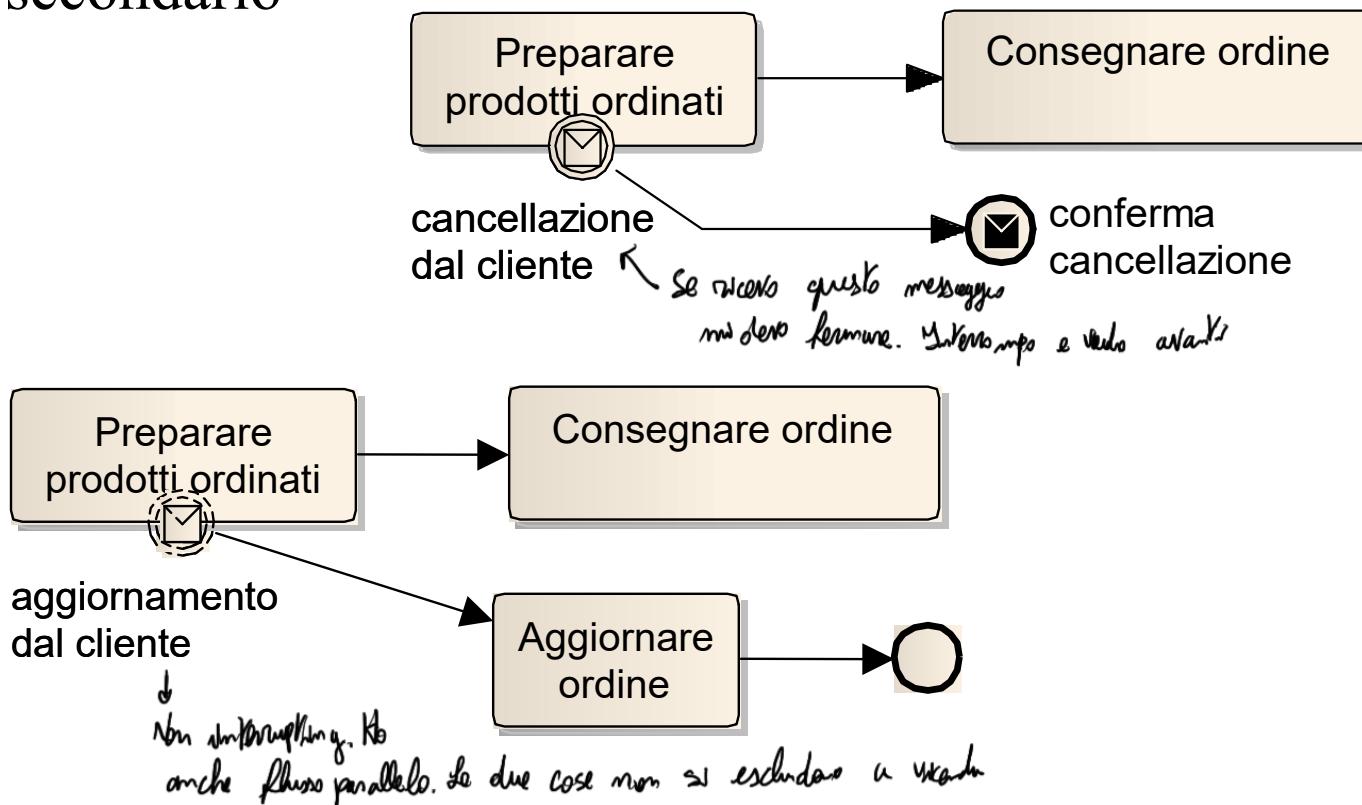
| | Icona | Significato | N.I. |
|--------------|-------|--|---------------|
| None | | Non specificato. es: arrivo/non arrivo guest? | MANDATO |
| Message | | Riceve (vuoto) o manda (pieno) un messaggio. qualcosa che viene eseguito al termine di un'attività | SOLO RICEVUTO |
| Timer | | Evento a periodicità predefinita. | |
| Error | | Attaccato ad un'attività che deve gestire un errore. | |
| Escalation | | Riceve (vuoto) o manda (pieno) una richiesta di escalation | |
| Cancel | | Solo nell'ambito di una transazione, cancella le variazioni. | |
| Compensation | | Effettua (vuoto) o attiva (pieno) una compensazione. | |

Intermediate event: tipologie (2)

| | Icona | Significato | N.I. |
|-------------|--------------|--|-------------|
| Conditional | | Evento legato al verificarsi di una condizione. | |
| Link | | Invia (pieno) o riceve (vuoto) un riferimento. | |
| Signal | | Invia (pieno) o riceve (vuoto) un segnale. | |
| Multiple | | Invia (pieno) o riceve (vuoto) più stimoli / eventi. | |

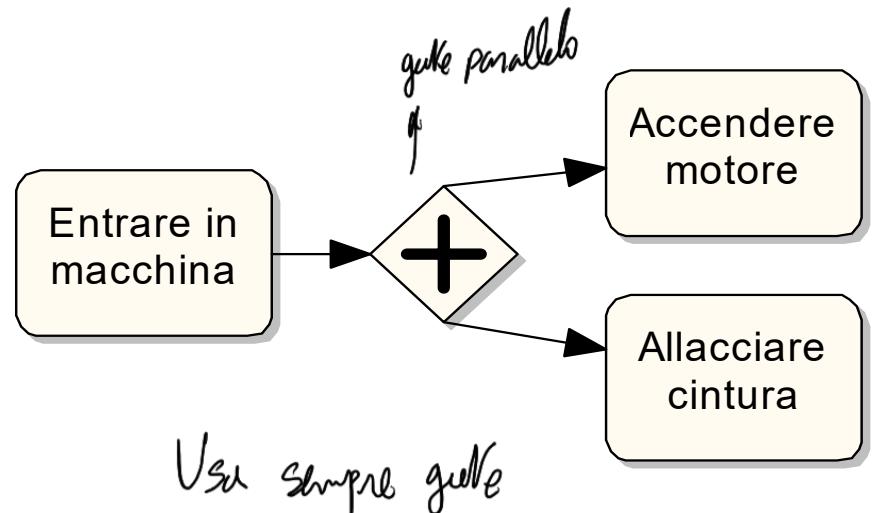
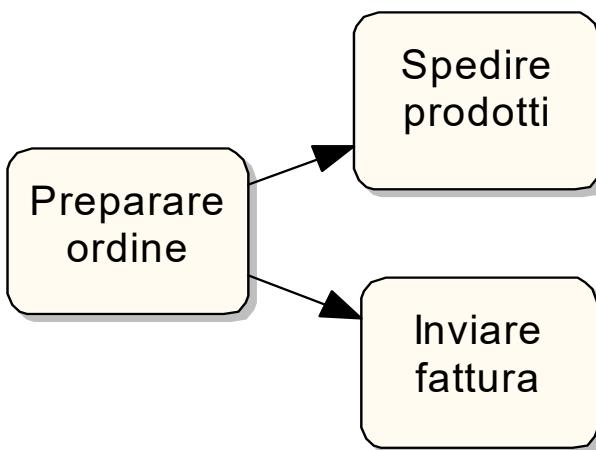
Evento intermedio “boundary”

- Un evento intermedio al bordo dell'attività la interrompe
- Oppure, se “non-interrupting”, attiva un flusso di attività secondario



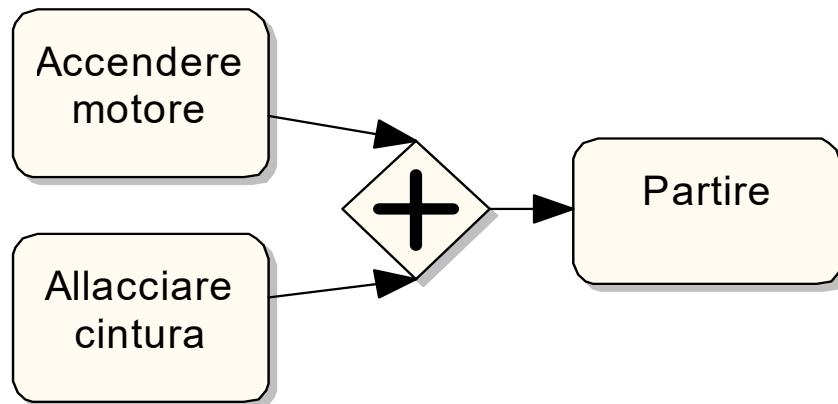
Fork (And-Split)

- Divisione di un cammino in due o più cammini paralleli
- Due opzioni di rappresentazione:

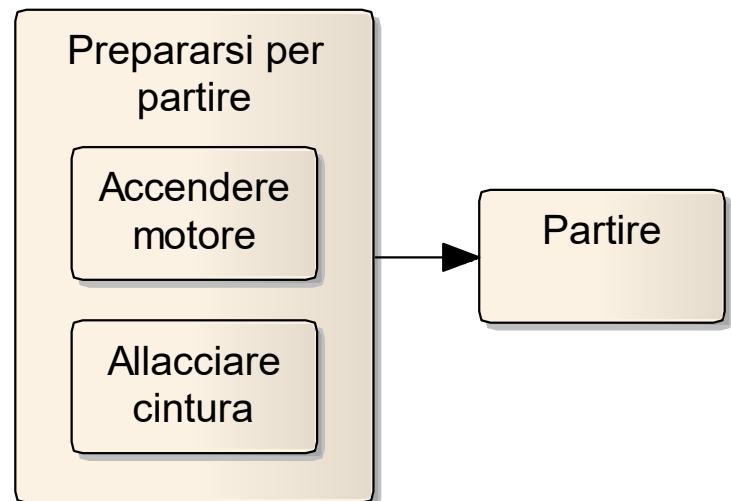


Join (And-Join)

- Congiunzione di due o più cammini paralleli in uno unico

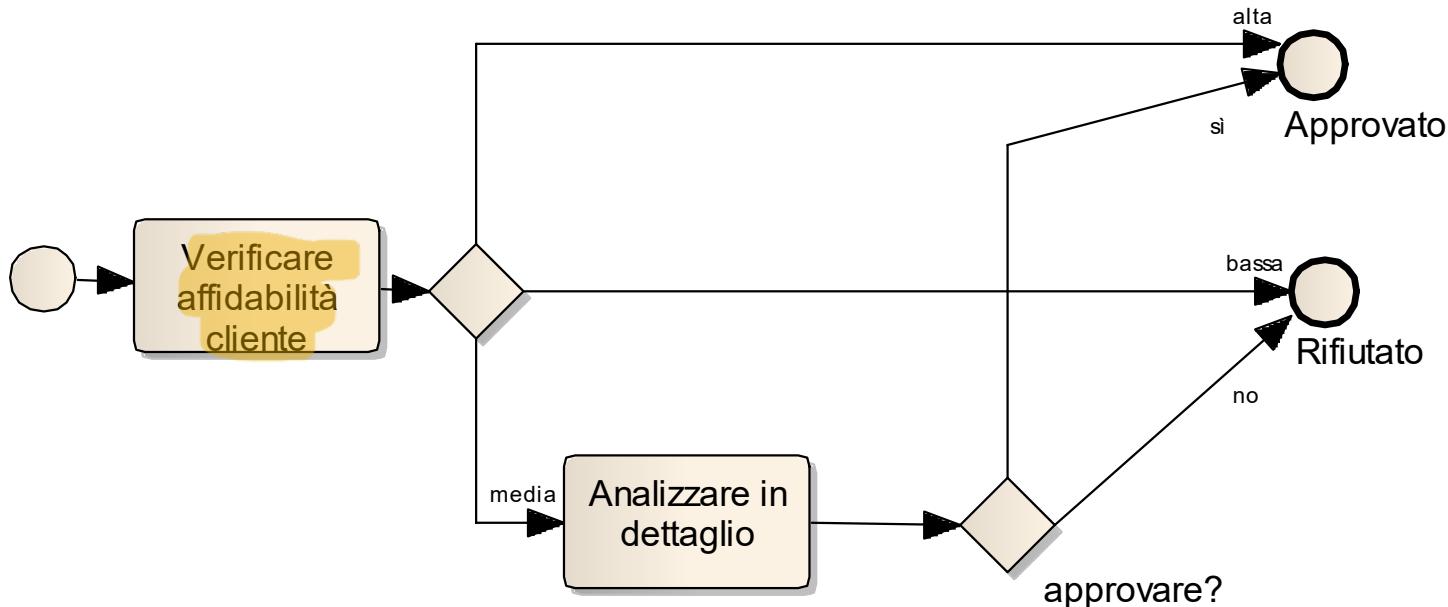


Se mettessi 2 frecce da ingresso non era equivalente



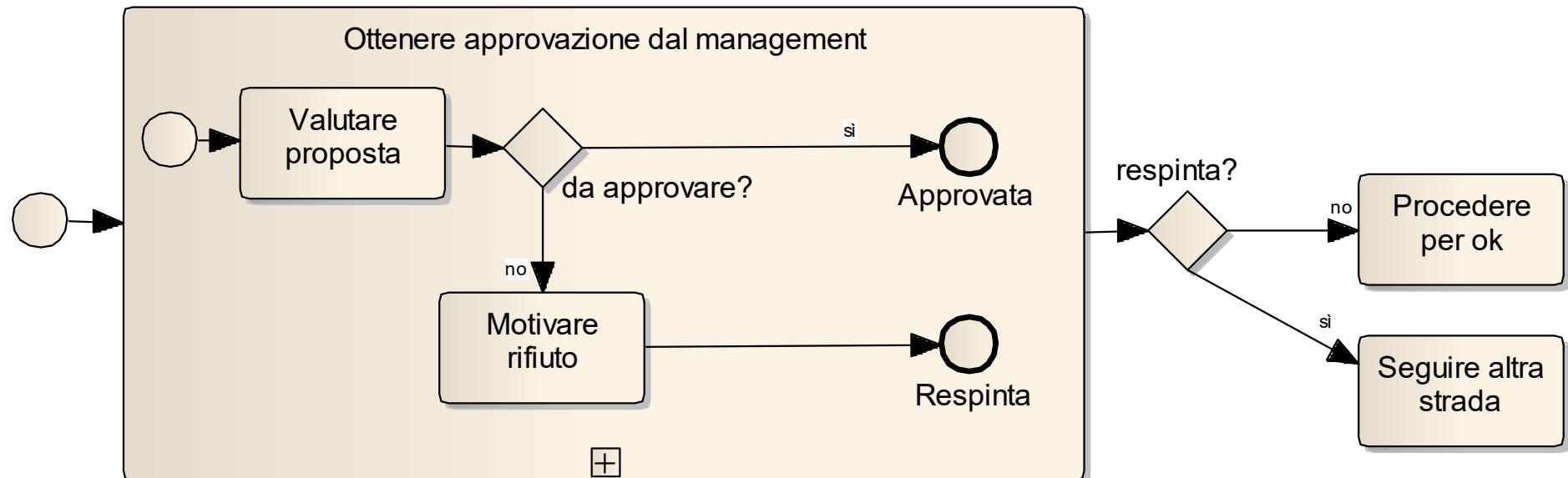
XOR basato sui dati

- Attenzione: il gateway “non decide”, ramifica solo, la decisione è conseguenza delle attività precedenti!
- Quando è possibile, etichettare il gateway con una domanda, e i flussi di sequenza in output con le risposte



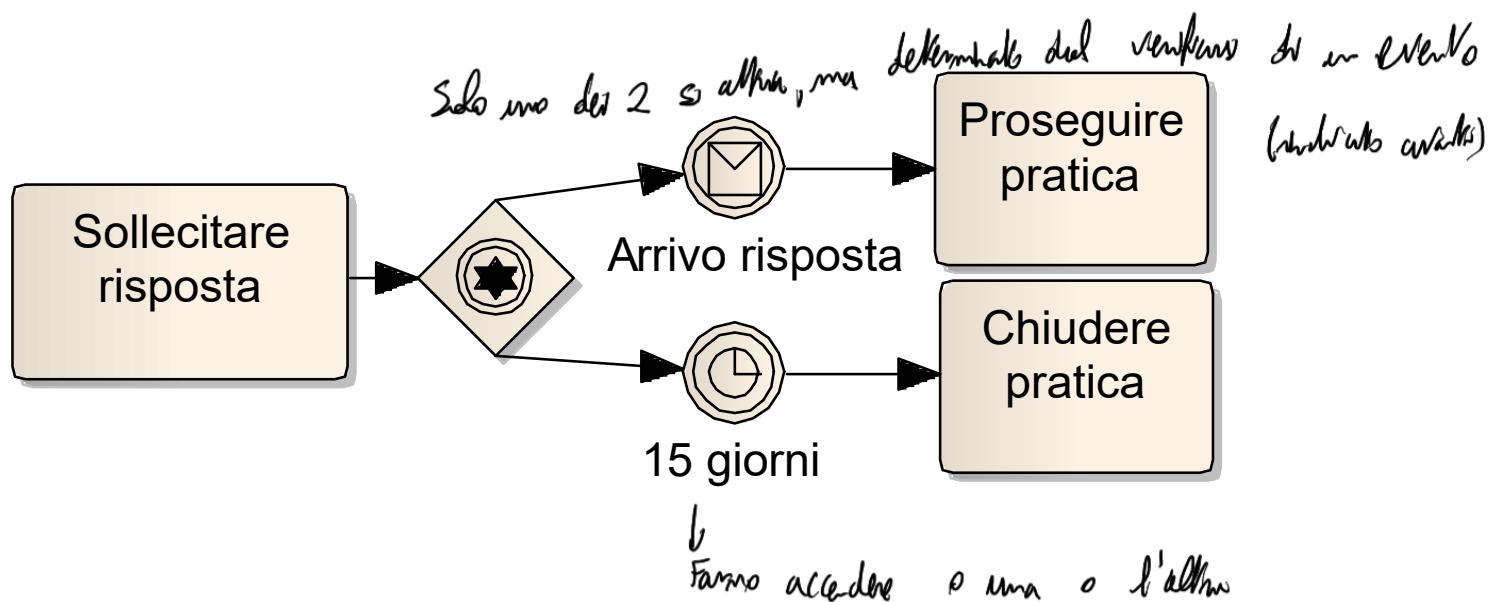
XOR basato sui dati

- La condizione può essere basata sul risultato di un'attività precedente (espresso con eventi finali distinti)



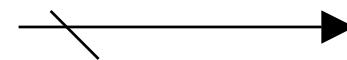
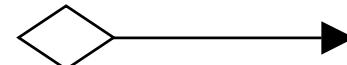
XOR basato su eventi

- **Esclusivo – basato su eventi** (tipicamente, la ricezione di un messaggio)
- il controllo passa tramite un evento intermedio, per poi proseguire con le attività successive



Flussi sequenza - tipologie

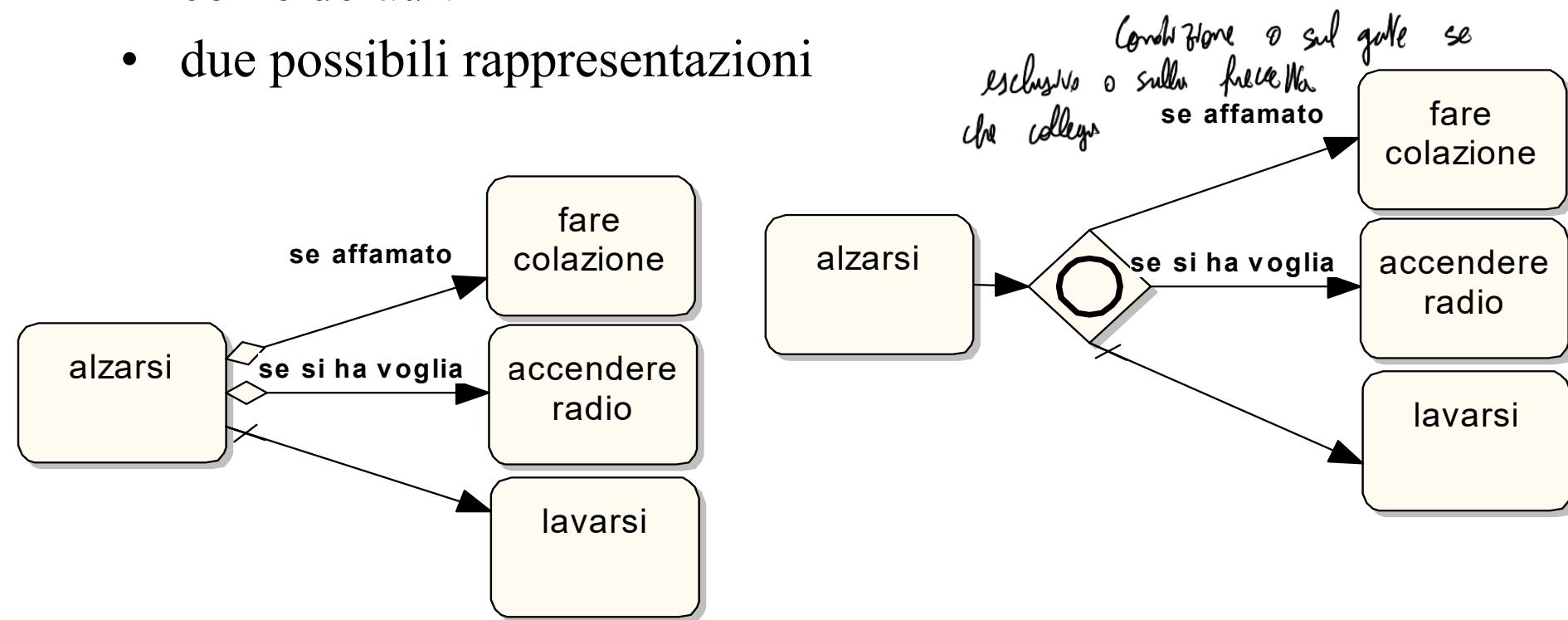
- **Uncontrolled** – non passante per un gateway, né condizionale
- **Condizionale** – usato solo se la condizione a cui è legato risulta vera
- **Default** – usato se altri flussi condizionali non vengono attivati



Negli esclusivi, o si attiva una o l'altra.
In quelli inclusivi dove non è detto che si attiverà
qualcosa, possono andare da 0 a N.
Meglio se metto default.

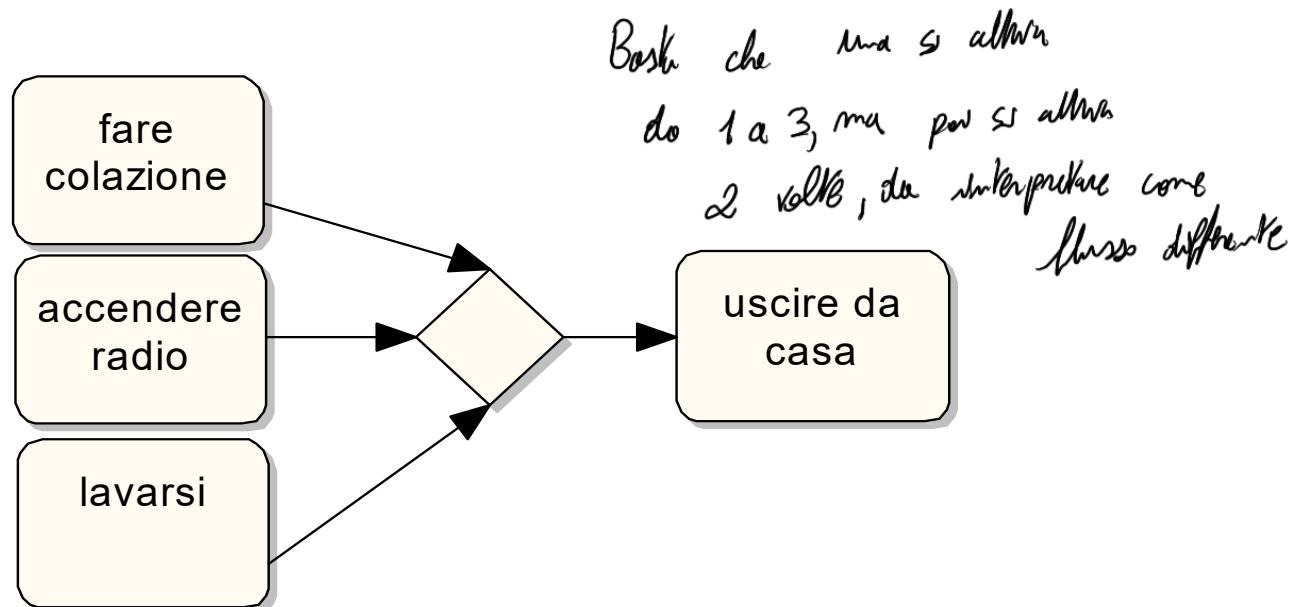
Branching (Or-Split) - OR

- non esclusivo – possono essere percorsi più cammini
- può essere opportuno indicare uno dei possibili cammini come default
- due possibili rappresentazioni



Merging (Or-Join)

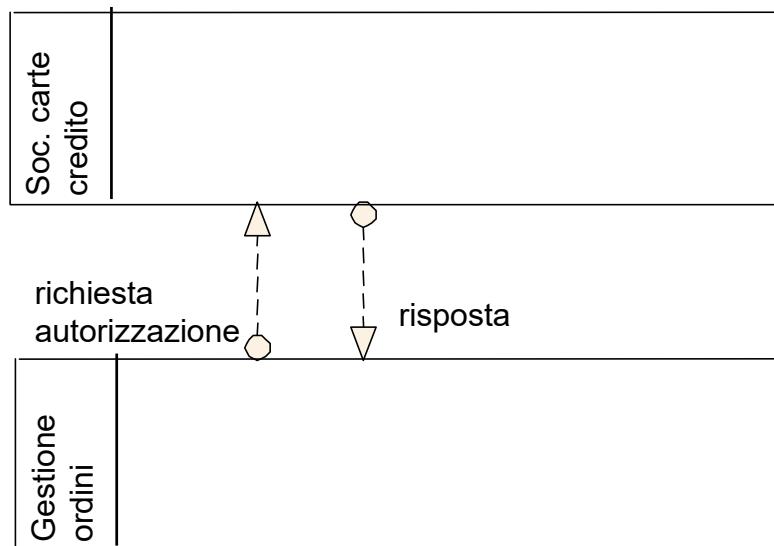
- congiunzione di due o più cammini in un cammino unico



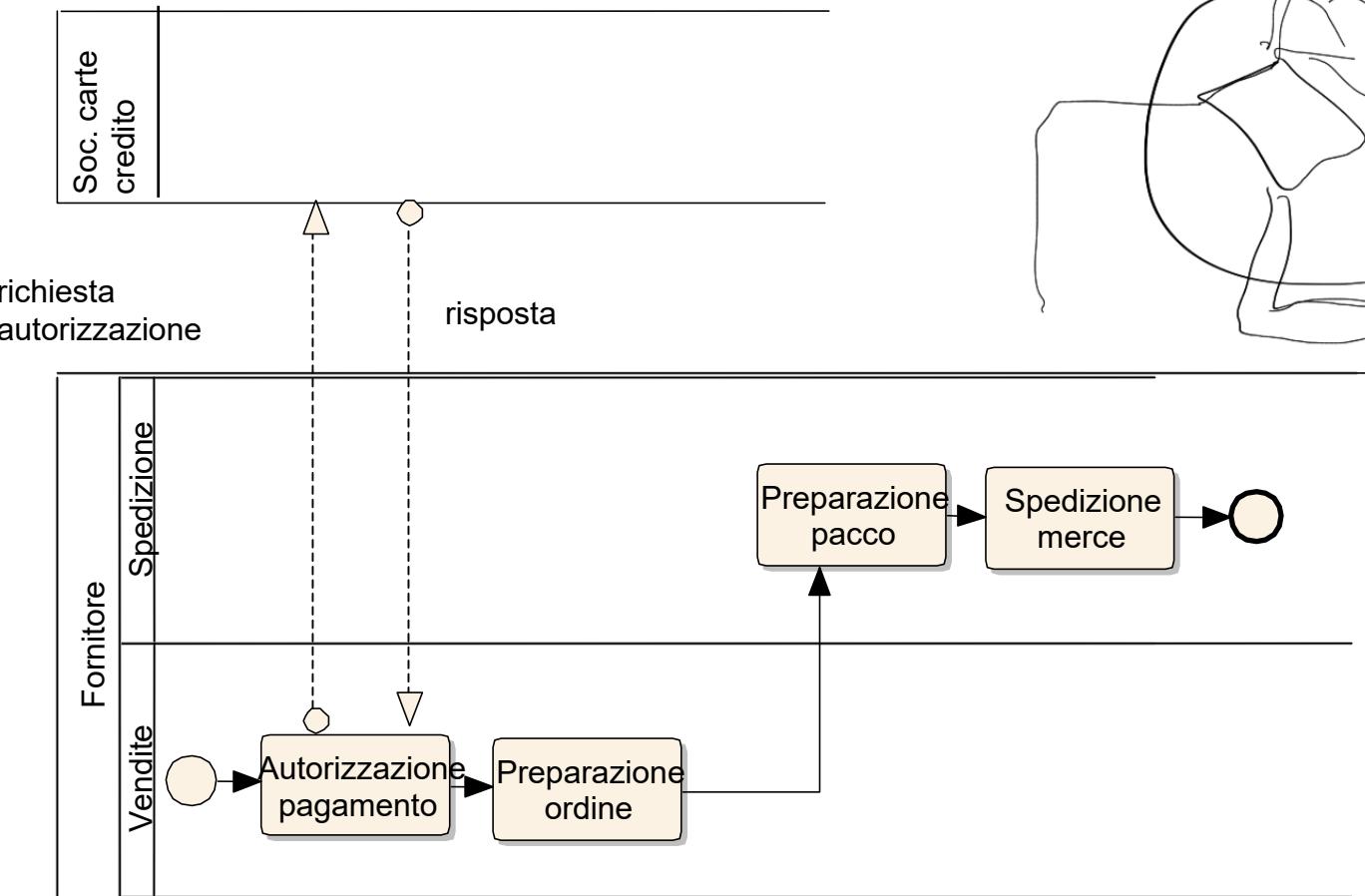
Pool e processi

Se un processo coinvolge più partecipanti:

- Ognuno può essere rappresentato con un pool distinto
- I pool possono essere “black box” (senza attività) se non ci interessano i processi interni dei partecipanti



Pool strutturato in lanes

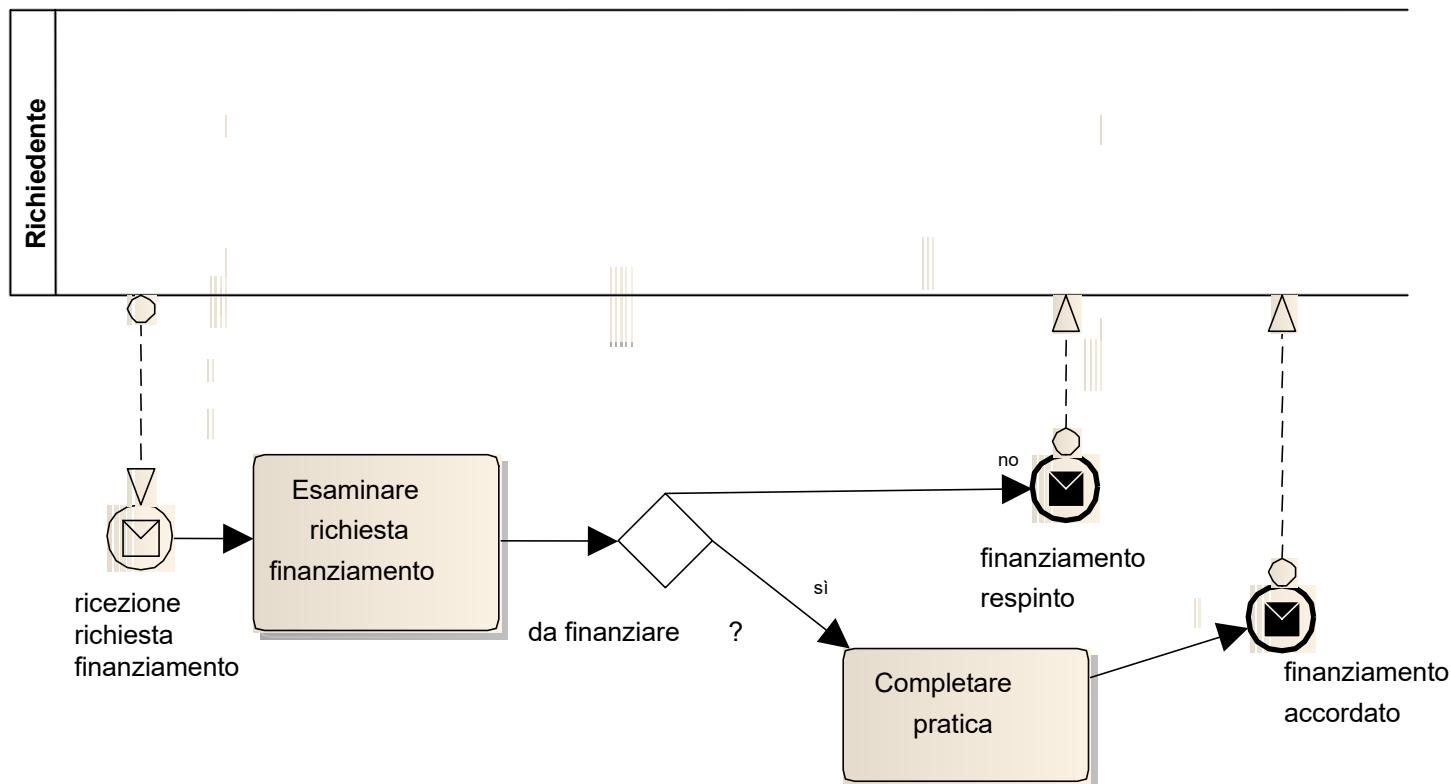


Flussi (sequenza, messaggio) e Pool

- Pool diversi possono essere connessi solo da messaggi, non da flussi di sequenza
- Non si possono inviare e ricevere messaggi tra attività ed eventi che appartengono al medesimo pool

Interazioni con un cliente

- Spesso un processo inizia con la ricezione di un messaggio da un cliente, e termina con uno o più messaggi di risposta



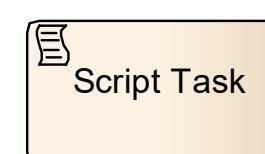
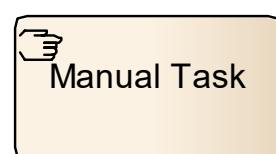
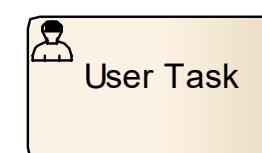
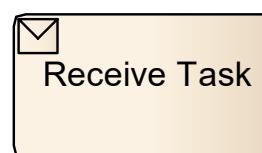
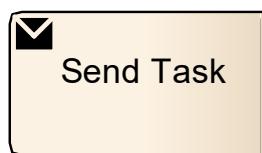
Fino a che livello di dettaglio?

Criteri possibili:

- Individuare attività o sottoattività automatizzabili e non
- Attribuire responsabilità univoche ad ogni attività

Tipi di task

- Service – servizio automatizzato (es. web service)
- User – attività umana svolta con uso di computer
- Manual – attività umana senza uso di computer
- Script – tipo di service task che corrisponde a codice eseguito dal motore di processi
- Business Rule – tipo di service task che valuta una regola di business e precede un gateway
- Send e Receive – invio e ricezione di messaggi



dove fare qualcosa
Per ricevere messaggi
devo fare qualcosa

RECEIVE: A Nessa è
parte dell'attività

BPMN.10

Per approfondimenti e altri materiali:

<http://www.analisi-disegno.com>