

Università di Pisa – Facoltà di Ingegneria

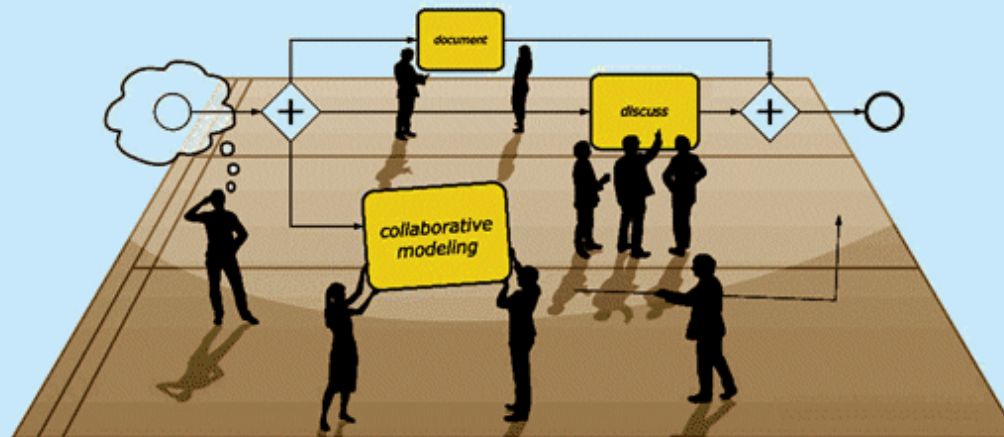
Corso di Laurea in Ingegneria Gestionale

*Laboratorio di Gestione della Qualità (prof. Failli)*

---

*“Le grandi e complesse organizzazioni aziendali sono la manifestazione tangibile della tecnologia avanzata, più delle stesse macchine” ( J.K. Galbraith)*

## **Introduzione alla rappresentazione dei processi organizzativi basata su BPMN**



Mario G. Cimino, Dipartimento di Ingegneria dell'Informazione

*Pisa, 14 Maggio (8.30-11.30 aula C22) e 15 Maggio (13.30-15.30 aula P1), 2012*

# Prologo

La tesi secondo cui per il buon andamento aziendale basti avere i prodotti ed i servizi “giusti” risulta oggi difficilmente condivisibile, poiché i prodotti hanno cicli di vita brevi e anche i migliori diventano ben presto obsoleti. Non sono i prodotti, ma i processi che li creano ad assicurare l’andamento positivo nel lungo periodo, sia per il cliente finale sia per l’azienda. Per poter individuare le azioni di miglioramento occorre sistematicamente monitorare ed analizzare sia i processi che le loro interfacce interne ed esterne. Sviluppare ed analizzare i processi aziendali consente di creare la giusta astrazione per capire il business, procurare una base per creare opportuni requisiti ingegneristici, fornire un supporto per studiare le implicazioni dei cambiamenti, identificare opportunità di outsourcing, e così via. Data la complessità della progettazione e del controllo dei processi delle moderne supply chain, occorrono opportuni linguaggi e tecniche di analisi, che includano possibilità quali simulazione, diagnostica, verifica, valutazione delle prestazioni. Il linguaggio BPMN rappresenta una notazione standardizzata dall’OMG e comprensibile da vari attori: gli analisti che definiscono i processi, gli sviluppatori che ne guidano l’implementazione tecnologica, i responsabili della gestione e controllo dei processi. BPMN è traducibile da e in formati XML per l’esecuzione dei processi, quali lo standard WS-BPEL 2.0 per ambienti service-oriented. Il presente seminario prevede l’illustrazione della notazione BPMN 2.0 attraverso un insieme di esempi, ed esercitazioni pratiche su casi di studio relativi all’analisi dei processi.

## Sommario

Introduzione a BPMN 2.0; caratteristiche generali; flusso di controllo e concetto di token; diagrammi di processo; esempi. Costrutti BPMN di base: processi, sottoprocessi, task; flussi di sequenza e punti decisionali; eventi iniziali, intermedi, finali; partizioni, pool, corsie; flussi di messaggi; artefatti: oggetto dati, gruppo, nota testuale. Consultazione della specifica BPMN 2.0 e costrutti avanzati: tipologie di evento, di processo, di punto decisionale, di flusso. Estendibilità del BPMN 2.0. Strumenti per la modellazione di processo in BPMN 2.0. Laboratorio didattico su BPMN e Business Process Analysis.

Il relativo materiale didattico è pubblicato su <http://tweb.ing.unipi.it/lgq> (user: *business*, pass: *pr0cess*)

# Business Process Model and Notation (BPMN 2.0)

- **BPMN è:**

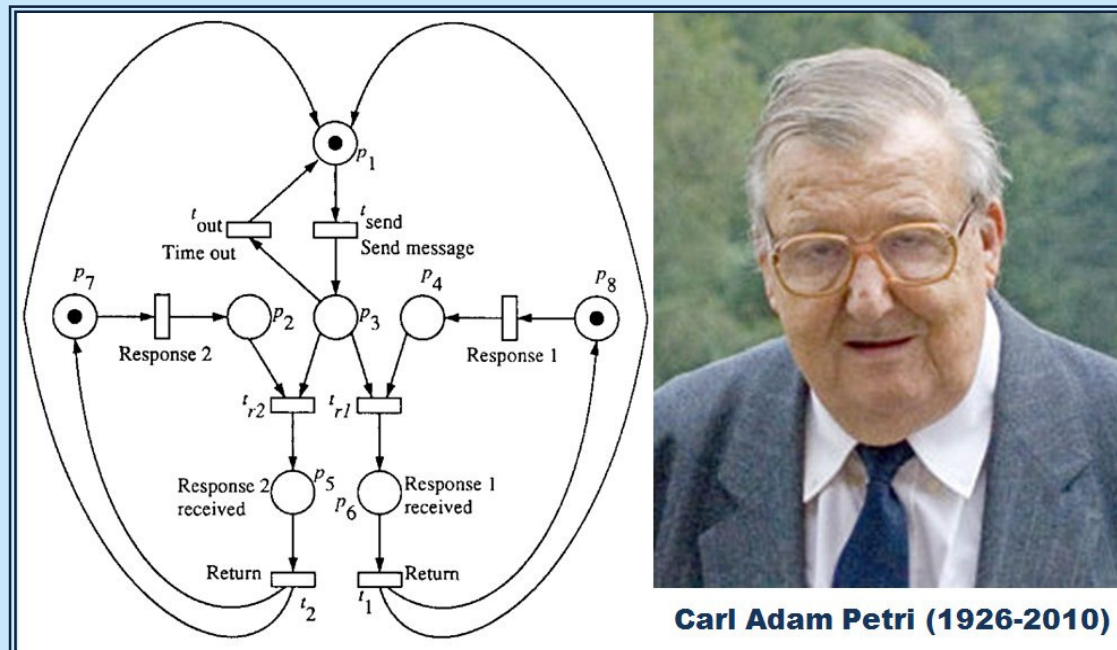
- ✓ *human-readable*: una notazione visuale standard per modellare i processi organizzativi;
- ✓ *accessible*: comprensibile da vari attori: chi analizza o definisce i processi, chi ne guida l'implementazione tecnologica, i responsabili della gestione e controllo dei processi;
- ✓ *machine-readable*: una notazione serializzabile in XML per la simulazione e l'esecuzione dei processi (es. WS-BPEL 2, ambienti SOA).

- **BPMN non è:**

- ✓ un linguaggio per rappresentare flussi di dati e di oggetti, sebbene possa mostrarli ad un certo livello di astrazione;
- ✓ una notazione per rappresentare strutture, scomposizioni funzionali, modelli di dati, strategie dell'organizzazione, regole di business.

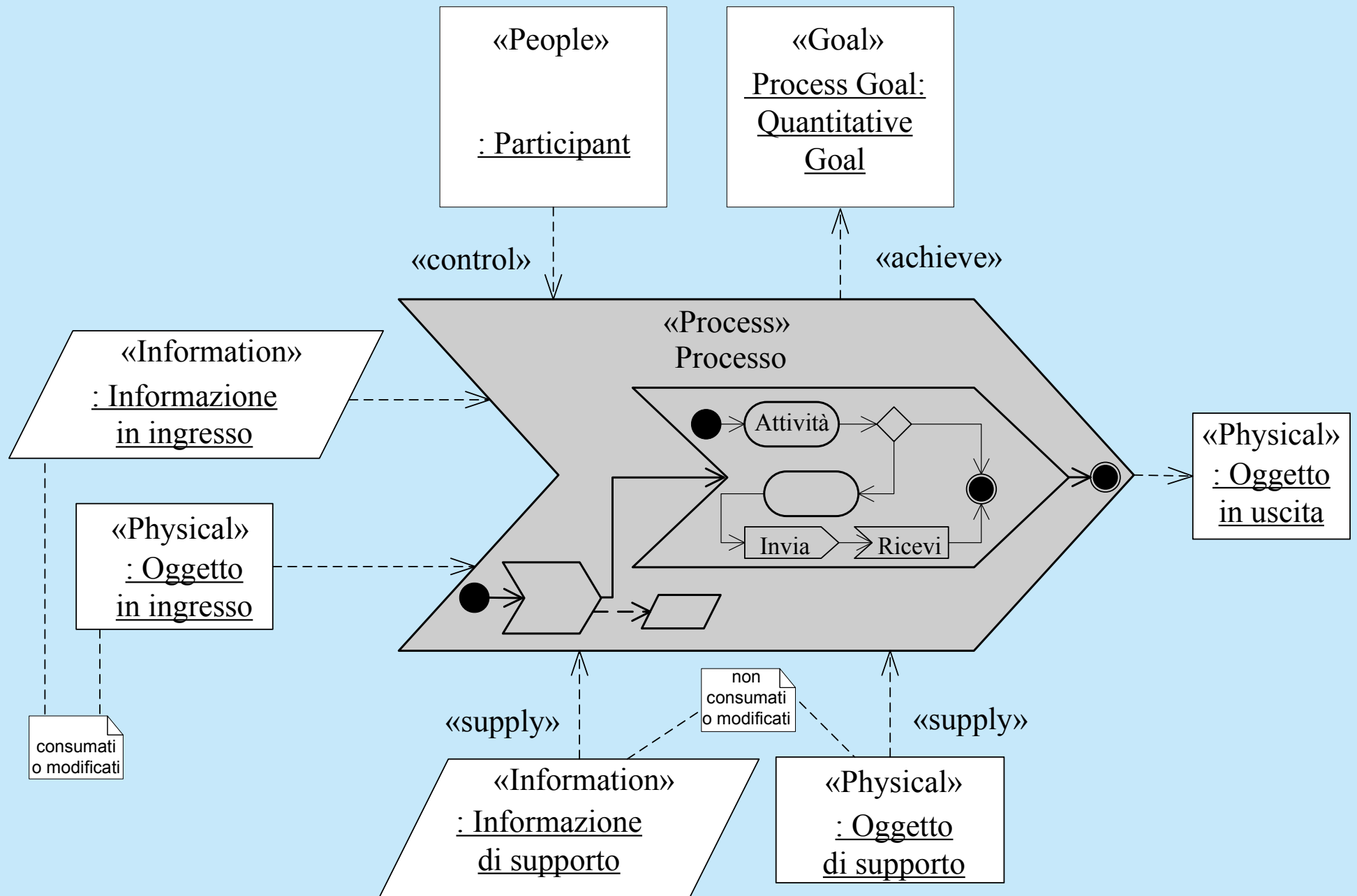
- **“Parenti” di BPMN** (linguaggi per workflow-based analysis)

- ✓ *Petri Nets* (1962): linguaggio formale per la modellazione di sistemi distribuiti, accessibile a computer scientist e progettisti di software specializzato. Possiede una rappresentazione visuale ed una sottostante notazione matematica (grafi), che consente analisi avanzate quali validazione, verifica (es. *soundness* per individuare deadlock, livelock, ...)

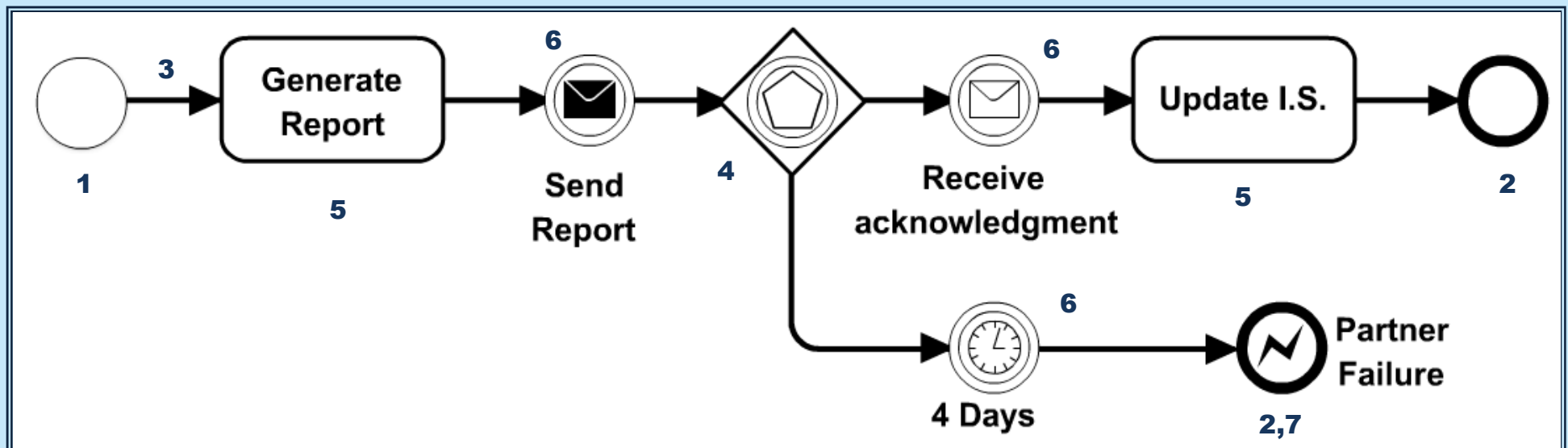


- ✓ *UML Activity Diagram* (OMG, 1997): linguaggio di modellazione visuale per il paradigma ad oggetti, accessibile a ingegneri del software. Nella forma estesa di Eriksson e Penker (2000) è adatto alla modellazione di processo ed accessibile anche ai ruoli non tecnici (di livello business).

# Esempio generico di diagramma di processo in UML esteso di Eriksson-Penker

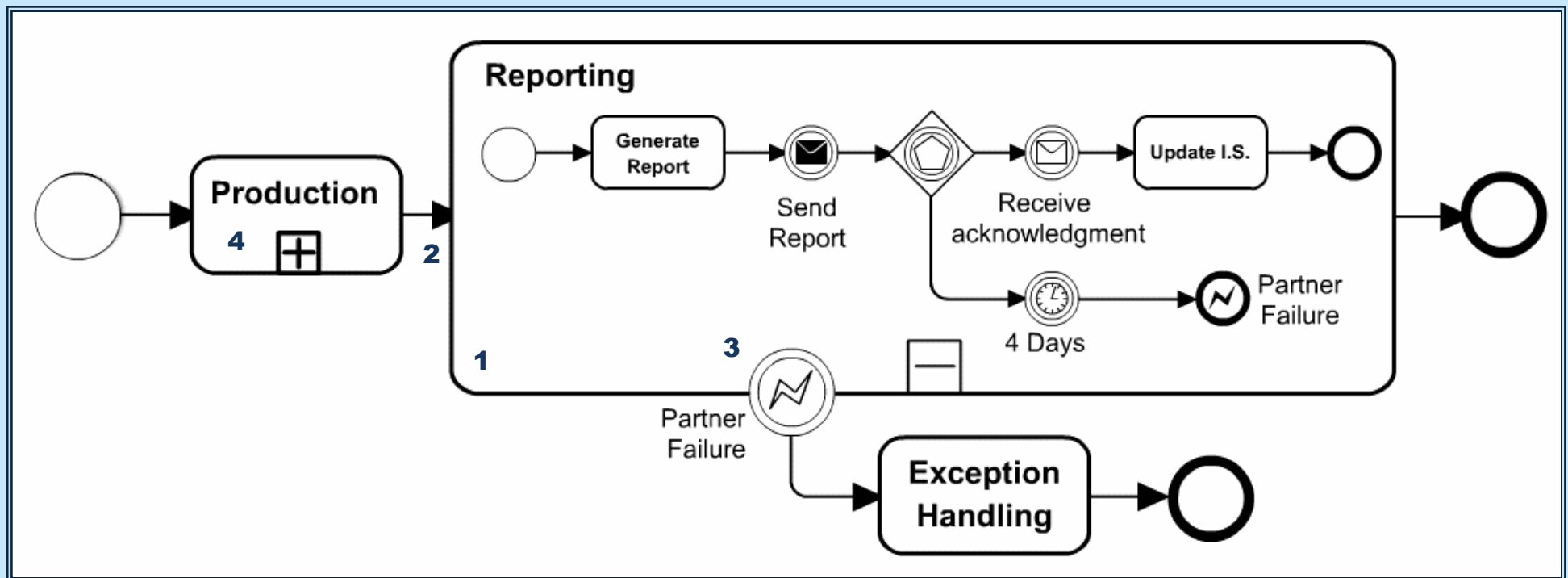


- *BPMN* (OMG, 2005) è specializzato nel rappresentare il comportamento dei processi per ciò che concerne il **flusso di controllo**, attraverso il concetto di *token* (gettone) che attraversa la struttura del processo.
- Uno *Start Event*<sup>1</sup> genera un token che alla fine sarà consumato da uno *End Event*<sup>2</sup>. Il percorso dei token è tracciabile attraverso una rete di *Sequence Flow*<sup>3</sup>, *Gateway*<sup>4</sup>, *Activity*<sup>5</sup> ed *Intermediate Event*<sup>6</sup>, all'interno del processo.
- *Race pattern*: i due eventi intermedi<sup>6</sup> dopo il punto decisionale ad eventi<sup>4</sup> “gareggiano”, attendendo rispettivamente la ricezione di un messaggio e lo scadere del tempo. In quest'ultimo caso il flusso di esecuzione termina con un evento errore con semantica *throw* (lancio)<sup>7</sup>.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov01.swf>

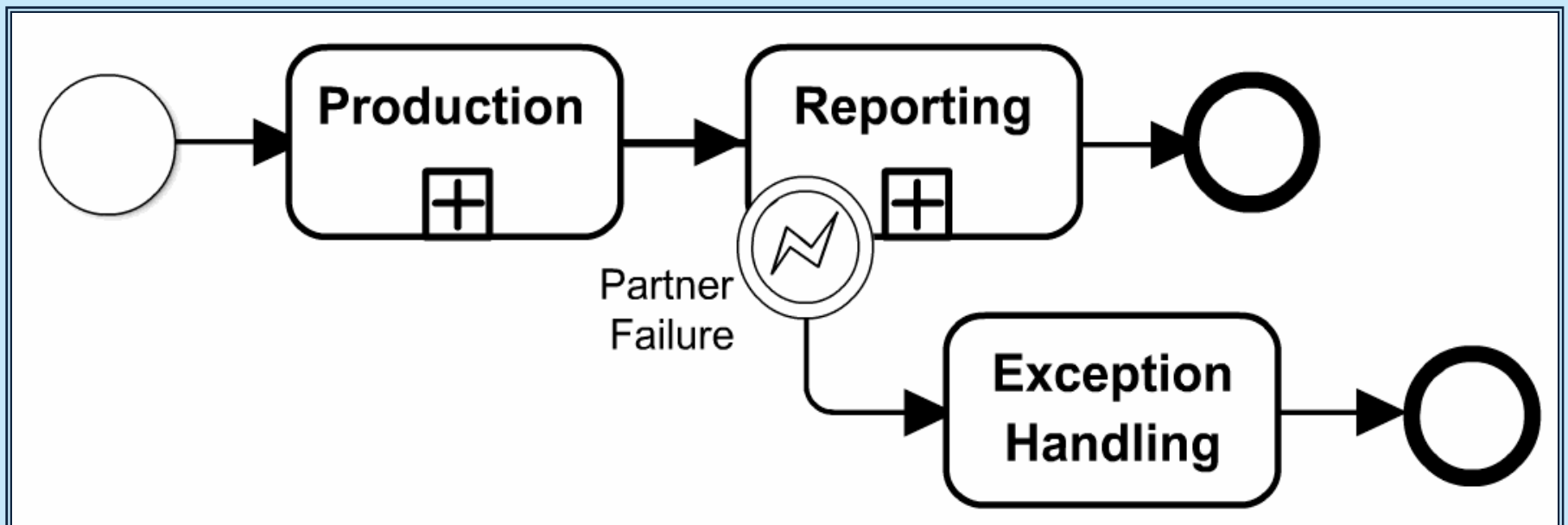
- *Interruption pattern*: il modello precedente viene incluso (embedded) come *sotto-processo*, rappresentato in forma espansa<sup>1</sup>. Il nuovo modello contiene anche un sotto-processo rappresentato in forma contratta<sup>4</sup>.
- L'evento di *Partner failure* con semantica *catch* (cattura) si abilita quando il flusso di esecuzione raggiunge il sotto-processo<sup>2</sup>, finchè non viene raggiunto uno *end event*.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov02.swf>

- Se viene lanciato, l'evento di errore viene catturato dall'omonimo evento di tipo *catch*<sup>3</sup>, e il flusso di esecuzione viene dirottato su quel punto.

- Nota: se un *Sub-Process* viene espanso all'interno di un diagramma, gli oggetti al suo interno non possono essere connessi all'esterno del Sub-Process.
- BPMN consente la *modellazione strutturata* dei processi, visioni a vari livelli di astrazione: dal livello "0" si scompongono i processi in sotto-processi, fino alle attività (livello di dettaglio massimo dell'analisi). Es. nella metodologia IBM l'analisi arriva tipicamente al *terzo* livello di scomposizione.

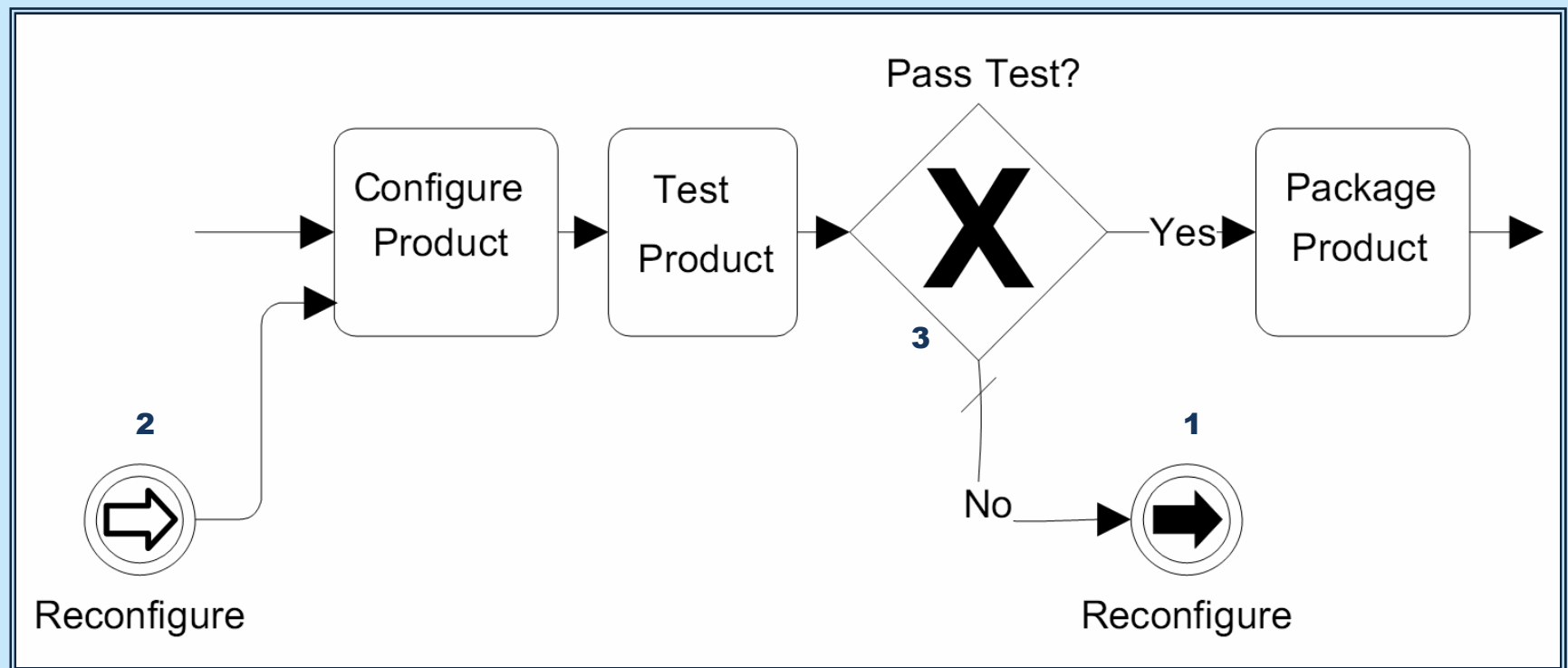


<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov03.swf>

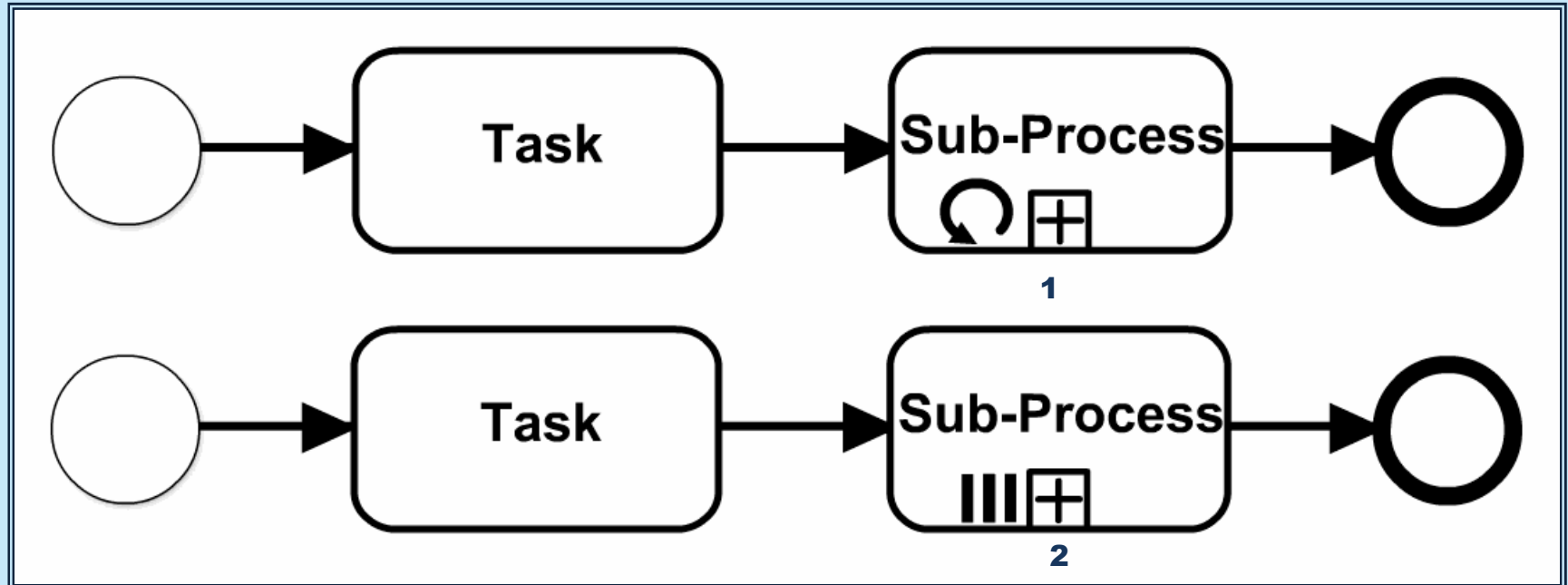
- Un processo descritto ad alto livello documenta un certo comportamento del modello senza i dettagli sufficienti ad eseguirlo. Ad esempio, mancano delle espressioni logiche nei punti decisionali, oppure non è serializzabile in WS-BPEL. In tal caso il processo si dice *non-eseguibile*.



- BPMN consente anche la *segmentazione* dei processi, ad un dato livello, per creare diverse sezioni modulari. Es. nella metodologia IBM si suggerisce di limitarsi a *sei* processilattività in ogni “vista” (stampo), per mantenere gestibile la mole dei dati. Nei formati elettronici (es. pdf) si tende ad includere molti più elementi, data la disponibilità delle funzioni di ingrandimento (zoom).
- I *Link Intermediate Event* di tipo throw<sup>1</sup> e catch<sup>2</sup> permettono di connettere due sezioni di un processo. Nell’esempio, il ramo “No” del gateway di tipo esclusivo<sup>3</sup> conduce a un salto del flusso di esecuzione producendo un ciclo.



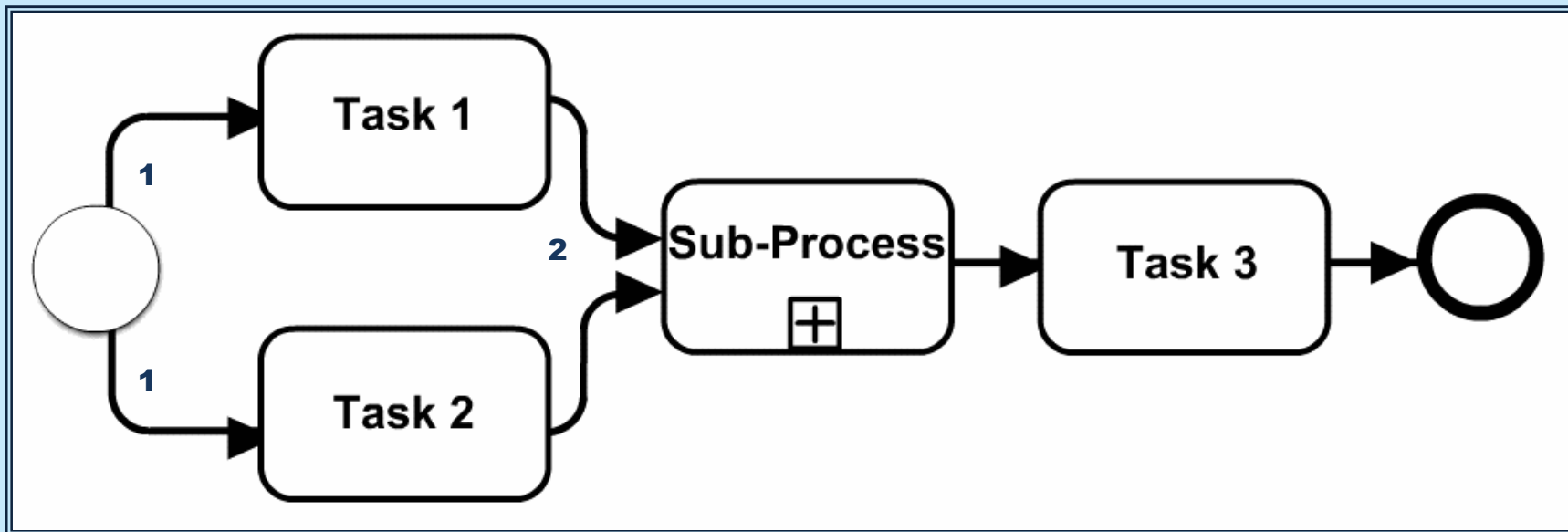
- Una medesima attività o processo può dar luogo a istanze multiple, per esempio reiterate<sup>1</sup> o parallele<sup>2</sup>. Il numero di istanze può essere calcolato a tempo di design o a runtime.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov04.swf>

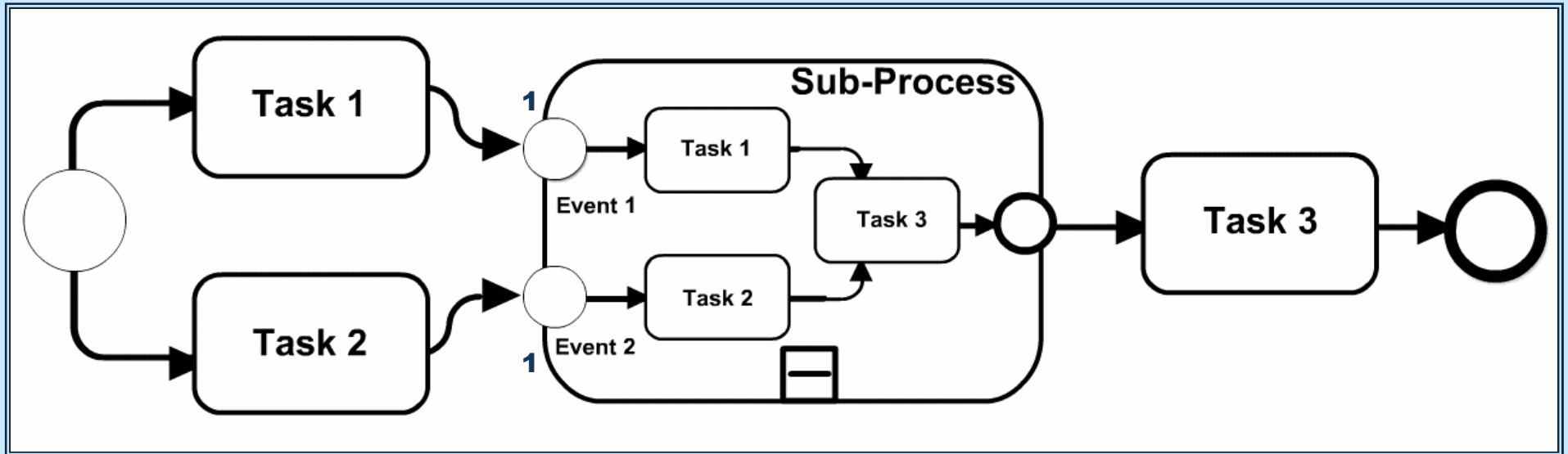
<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov05.swf>

- Possono essere usati *Sequence Flow*<sup>1,2</sup> multipli senza alcun punto decisionale (*uncontrolled flow*) in ingresso o uscita ad attività/processo. In ciascun *uncontrolled flow* fluirà un token.
- Nell'esempio seguente, vengono create due istanze del sotto-processo, ciascuna relativa ad un token distinto.



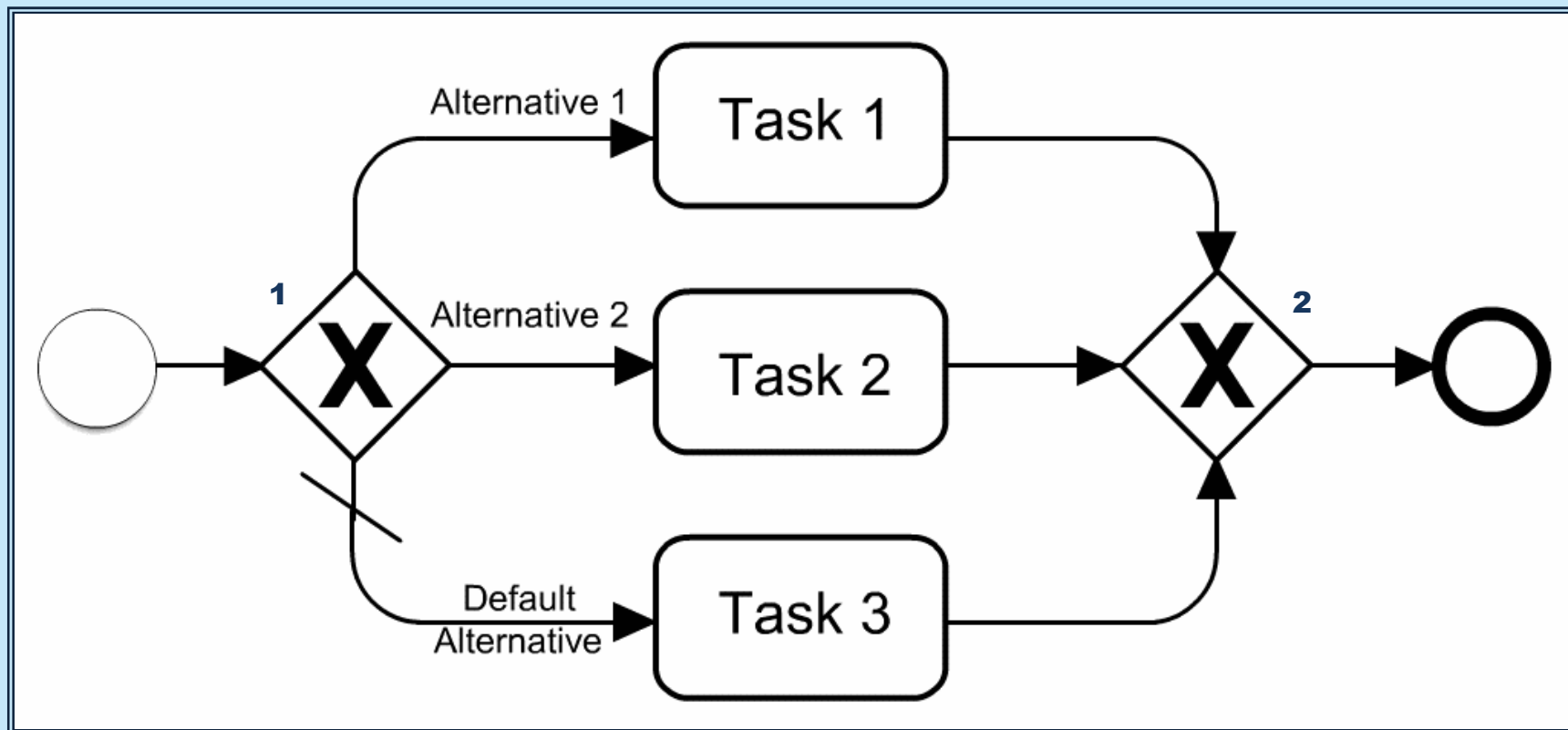
<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov06.swf>

- Se il sotto-processo viene espanso, possiamo immaginare ad esempio che i due flussi in ingresso puntino a diversi *start event*.



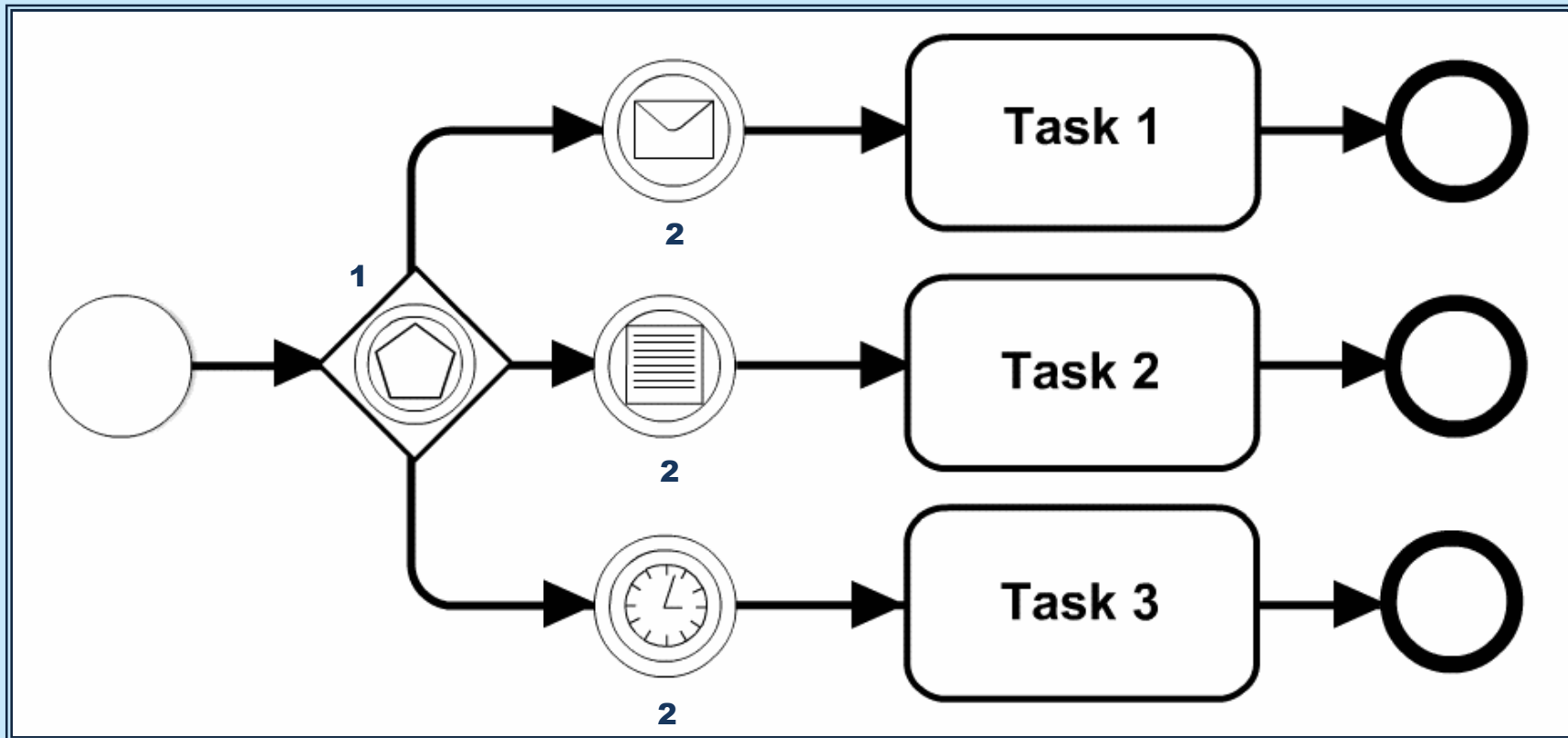
<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov06.swf>

- BPMN mette a disposizione varie tipologie di punto decisionale (o gateway). Nello *exclusive Data-Based gateway*<sup>1</sup> solo una delle alternative di uscita è vera e consente al token di passare.
- Tale gateway può essere adoperato anche per unire più flussi mutuamente esclusivi in ingresso<sup>2</sup>, ed in tal caso l'unico flusso attivo passa direttamente in uscita.



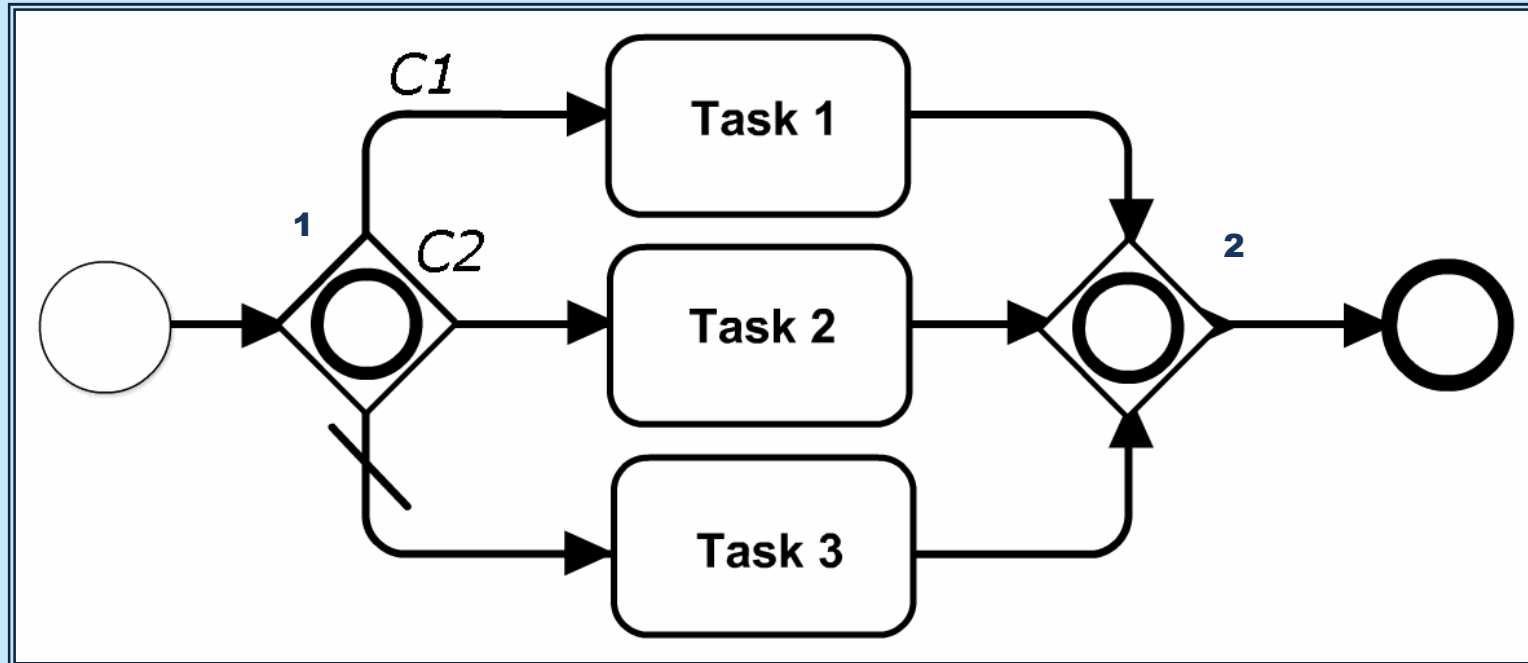
<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov07.swf>

- Nello *Exclusive Event-Based gateway*<sup>1</sup> c'è una gara (race) tra i diversi eventi<sup>2</sup> in cui vince il primo evento che arriva.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov08.swf>

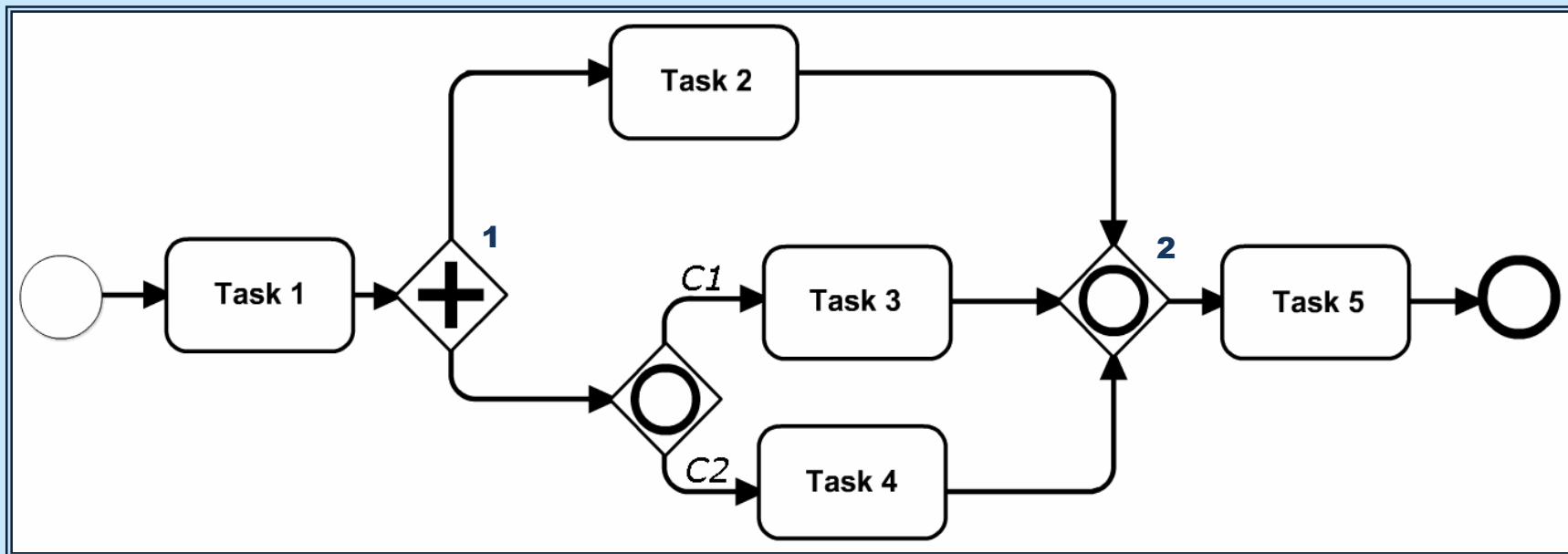
- Nello *Inclusive gateway*<sup>1</sup> più condizioni in uscita potrebbero essere vere. Può anche essere adoperato per unire più flussi in ingresso<sup>2</sup>, ed in tal caso si attendono tutti i token prodotti prima di proseguire in uscita.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov09.swf>

- Un percorso di default (opzionale) viene identificato da un tratto diagonale. Esso viene usato se nessuna delle condizioni è vera. Se non è presente il percorso di default e nessuna condizione è vera, viene lanciata una eccezione a runtime, poiché il token in ingresso sarebbe inaspettatamente consumato.

- Il *Parallel Gateway*<sup>1</sup> non ha condizioni logiche associate. Viene usato per sincronizzare flussi paralleli (in ingresso) e per creare flussi paralleli (in uscita). Si attende che tutti i token in ingresso siano arrivati, e poi si emettono nuovi token su tutti i flussi in uscita.
- Nell'esempio di seguito, si noti come il secondo gateway inclusivo<sup>2</sup> attenda tutti e soli i token prodotti prima di procedere. Se sostituiamo ad esso un gateway parallelo, nel caso di token non prodotto (*C1* o *C2* pari a false) il token in uscita non verrebbe mai emesso.

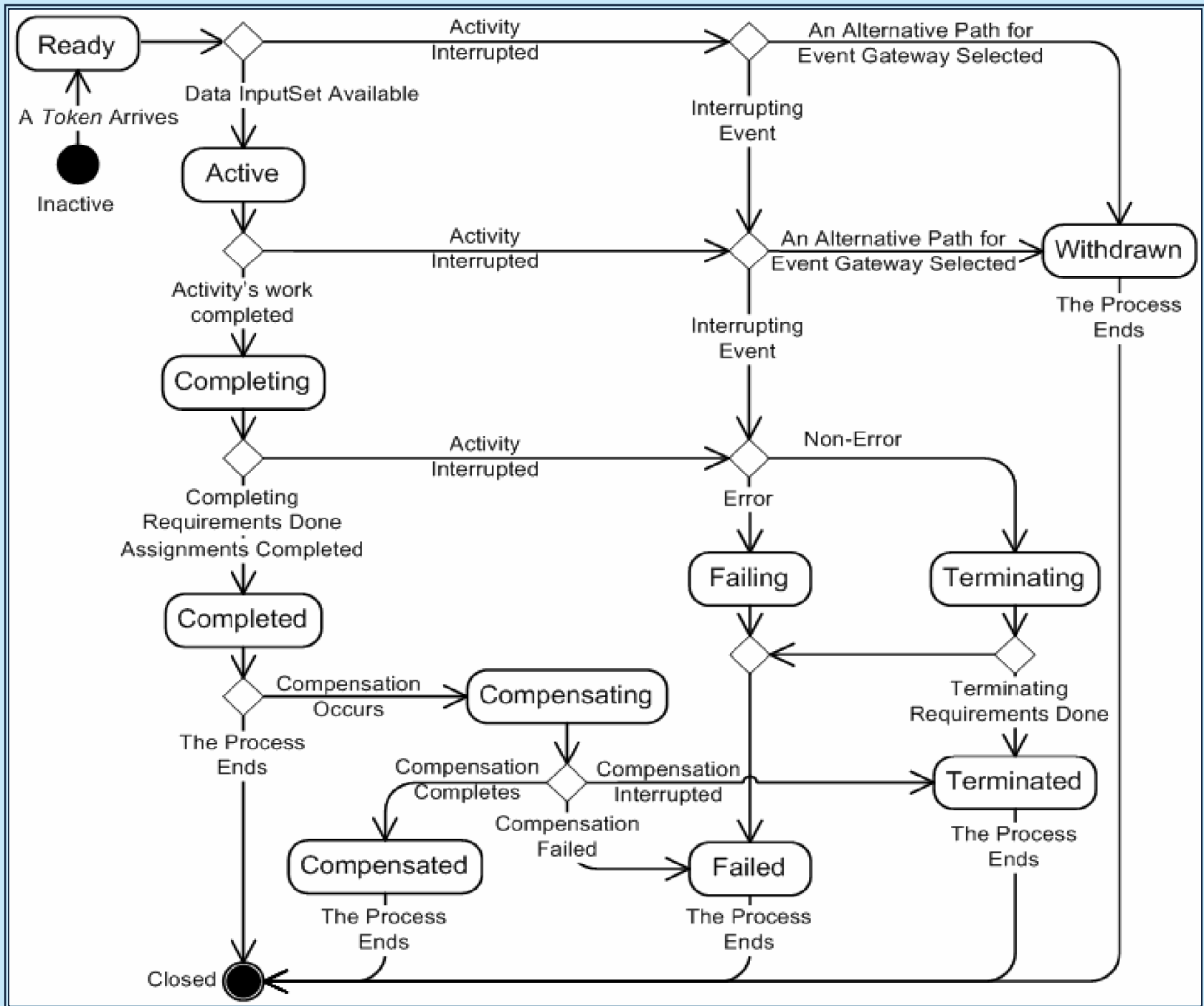


<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov10.swf>



# Ciclo di vita di un Task o Processo in BPMN

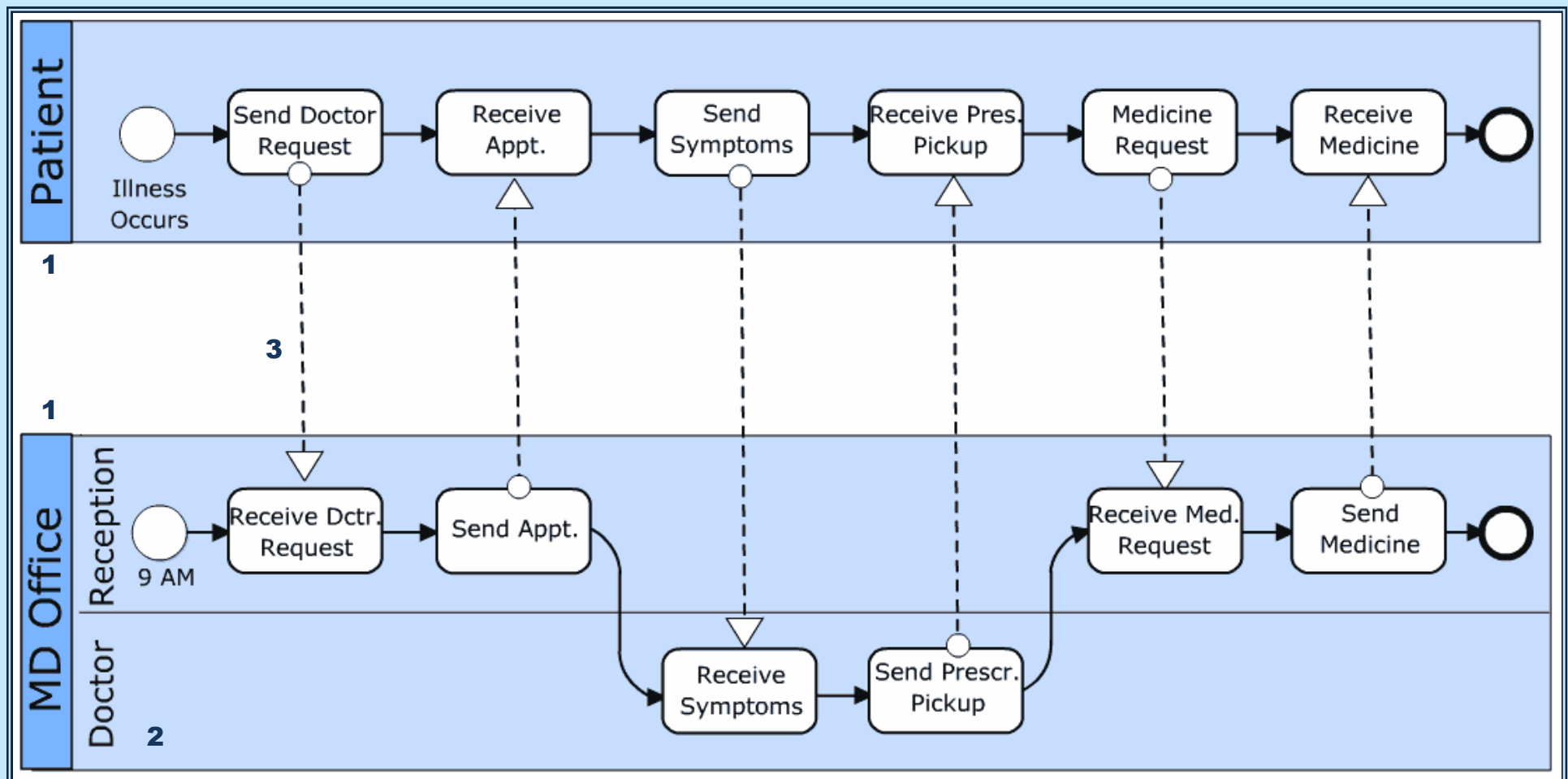
- Una generica *Activity* (uno step di un processo) può essere atomica (task) o composta (Sub-Process), ed eseguita da un sistema (automatico) o persona (manuale).
- Una *Activity* è *Ready* per l'esecuzione se il numero richiesto di token (*InputSet*) è disponibile per attivarla. A quel punto l'*Activity* passa allo stato *Active*, ossia una sua istanza va in esecuzione.
- Una *Activity Ready* o *Active* può essere *Withdrawn* (richiamata) es. se appartiene ad un ramo di un *Event-Based Exclusive Gateway* (race), in cui il primo elemento (task o evento) che va a completamento porta gli altri task ad essere richiamati.
- Una *Activity* può *fallire* e/o *terminare* durante l'esecuzione. Se la esecuzione della *Activity* finisce senza anomalie, lo stato va a *Completing*.
- L'esecuzione di una *Activity* è interrotta se si solleva un *Interrupting Event* (es. un *error*) o inizia un *interrupting Event Sub-Process*.
- Una *Activity* in stato *completed* può essere soggetta a *compensazione*, ossia disfare passi che erano stato completati con successo, riportando indietro lo stato del business (es. ho si fa un versamento, e poi si preleva il medesimo importo). Notare che finchè l'attività è in stato *active* non può essere compensata.
- Tale comportamento viene descritto dal seguente diagramma di stato UML.



# L'interazione tra organizzazioni in BPMN

- Ogni organizzazione ha le proprie interfacce di business verso l'esterno. Ciò significa che vi sono processi *privati* (che non interagiscono con l'esterno) e processi *pubblici* all'organizzazione. In BPMN i processi pubblici sono quelli che contengono attività di comunicazione tra partecipanti, quindi producono un flusso di messaggi.
- BPMN consente di esprimere tre tipi di processo:
  - ✓ *Orchestrazione*: interazione formata dai processi coordinati da un medesimo partecipante. Si compone di processi privati [si pensi al direttore d'orchestra].
  - ✓ *Coreografia*: insieme ordinato di interazioni tra i processi relativi a diversi partecipanti, senza controllo centralizzato. Si compone di processi pubblici [si pensi alla coreografia nella danza].
  - ✓ *Collaborazione*: insieme di interazioni tra i partecipanti medesimi. Può includere anche la coreografia e uno o più orchestrazioni.
- Nell'esempio di seguito si ha una collaborazione tra medico e paziente. I processi di ciascun partecipante sono racchiusi in un *pool*<sup>1</sup> (vasca). Un pool può essere ulteriormente suddiviso in *lane*<sup>2</sup> (corsie), ciascuna rappresentante un sotto-responsabile o una sotto-unità.

- Un *Sequence Flow* non può passare attraverso il confine di un Pool. Le interazioni tra i due partecipanti costituiscono sempre un flusso messaggi<sup>3</sup>, che può essere attraversato solo da un *Message* e non da token. *Message Flow* non possono connettere oggetti che sono all'interno del medesimo Pool.

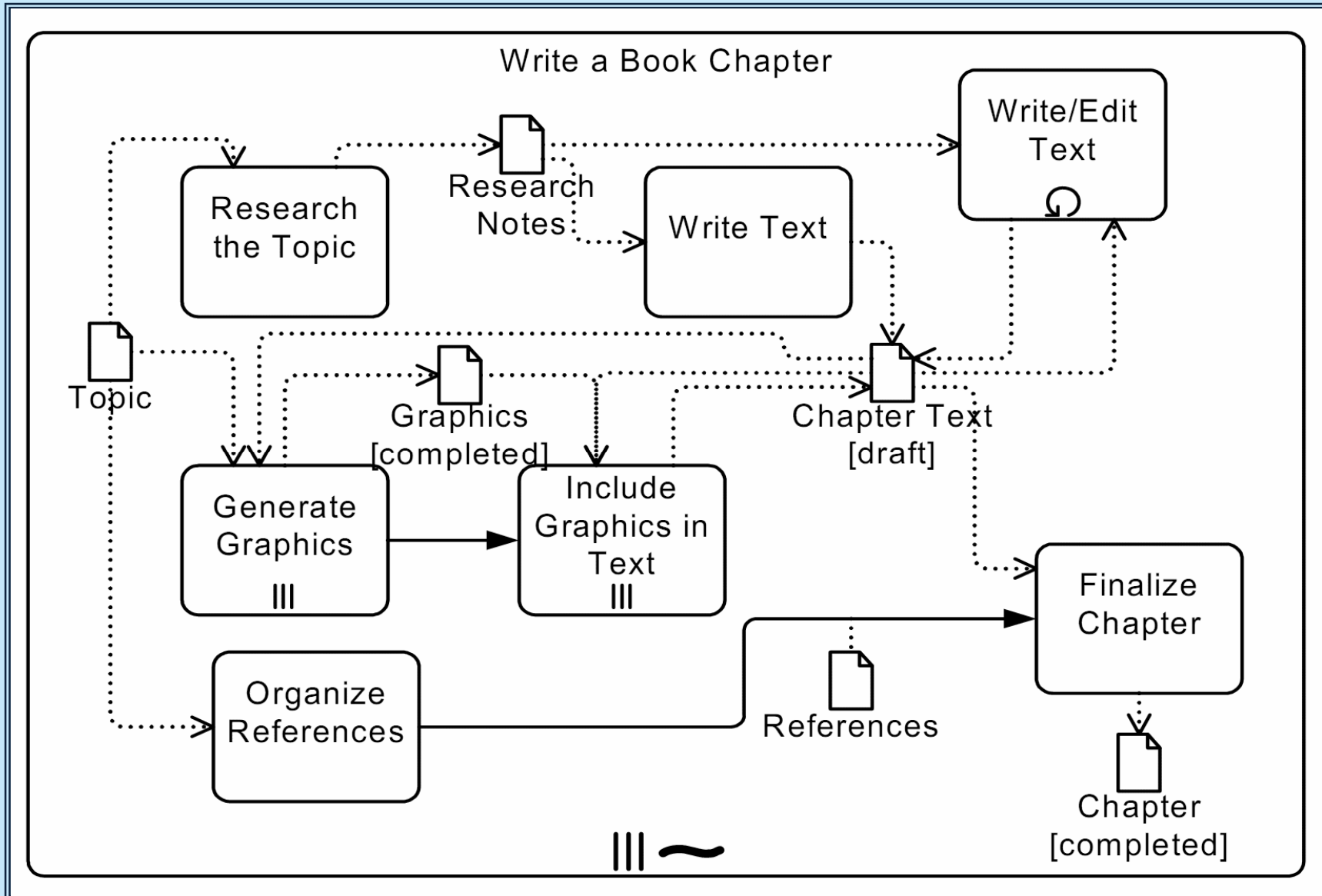


<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov11.swf>

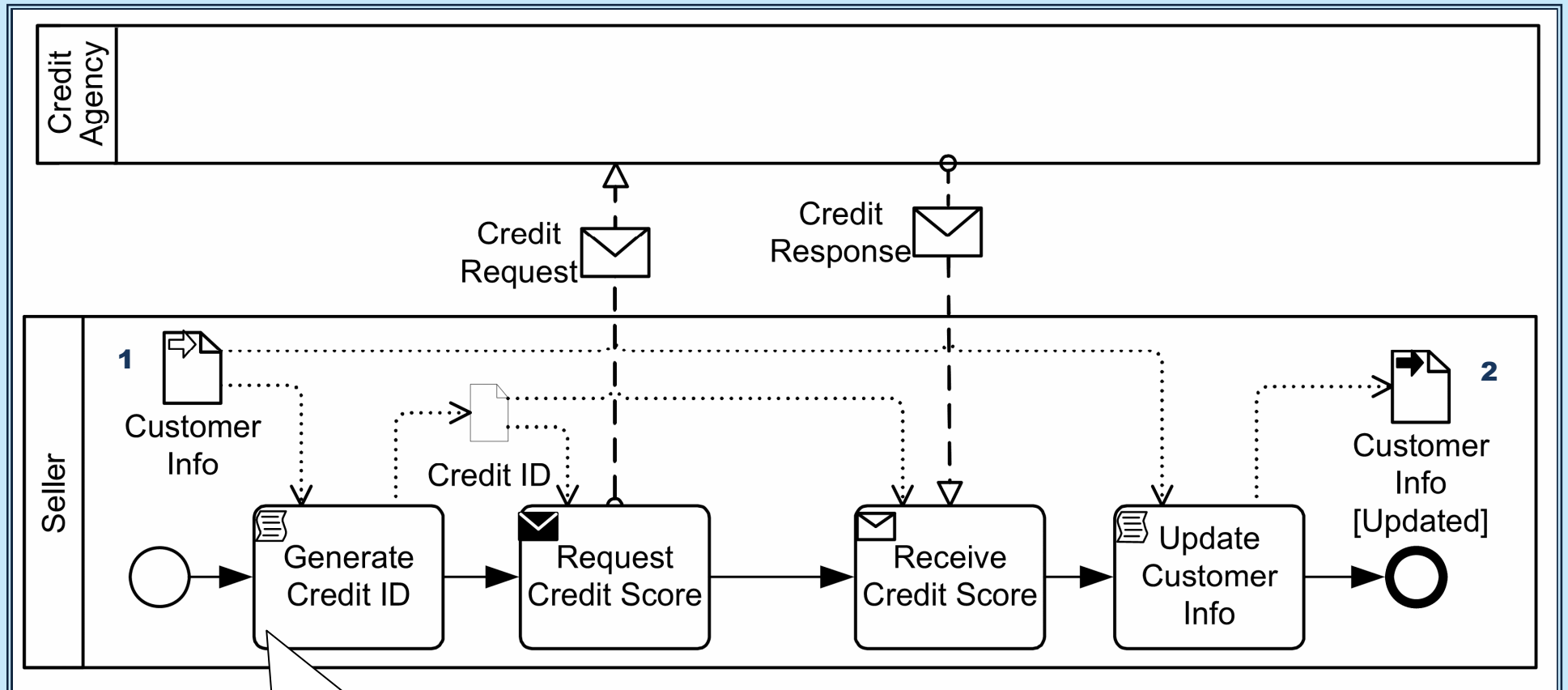
# Flusso dati in BPMN

- In BPMN, i *Data Objects* (📄) e *Data Input/Output* forniscono informazioni sui dati prodotti da una attività e da un processo. Il flusso dati viene indicato da una associazione direzionale, che rappresenta anche una forma di dipendenza.
- Ad esempio, un processo *Ad-Hoc* (indicato dal simbolo ‘~’) contiene un gruppo di attività non strettamente vincolate da relazioni di sequenza. In tal caso la sequenza ed il numero di esecuzioni dipendono ogni volta dall’esecutore. Esempi: sviluppo di codice a basso livello, supporto alle vendite, scrivere un capitolo di libro.
- Anche il messaggio (✉) può rappresentare il contenuto di una comunicazione tra due partecipanti.
- Esempio. Nella scrittura di un capitolo di libro non c’è una sequenza predefinita tra tutte le varie attività, poiché si tratta di un processo creativo. Ad esempio “write text” non precede sempre “edit text”, perché l’editing può avvenire raramente e sulla base di molte istanze di “write text”.
- Tuttavia, i data objects, in qualità di input, determinano un vincolo relativo al flusso di sequenza, in quanto senza il dato di input il processo non può essere eseguito.
- Per cui potremo sicuramente dire che alcune istanze di “write text” verranno prima di istanze di “edit text”, ma che ciò non avviene a livello di processo (su tutte le istanze), per cui non si mette alcun flusso di controllo tra i due processi.

- Invece tra “generate graphics” e “include graphics” possiamo immaginare che l’inclusione di una data immagine avvenga sempre dopo la generazione della medesima, e quindi si rileva un flusso di controllo tra i due processi.



- Altro esempio, la erogazione di credito per un cliente (es. finanziamento per auto o elettrodomestico), basata su una cooperazione tra commerciante e istituto di credito, presenta Data Input<sup>1</sup> e Data Output<sup>2</sup>.



**Cosa rappresenta il simbolo del foglietto ondulato?  
Consultare il BPMN Poster e poi la Specifica.**

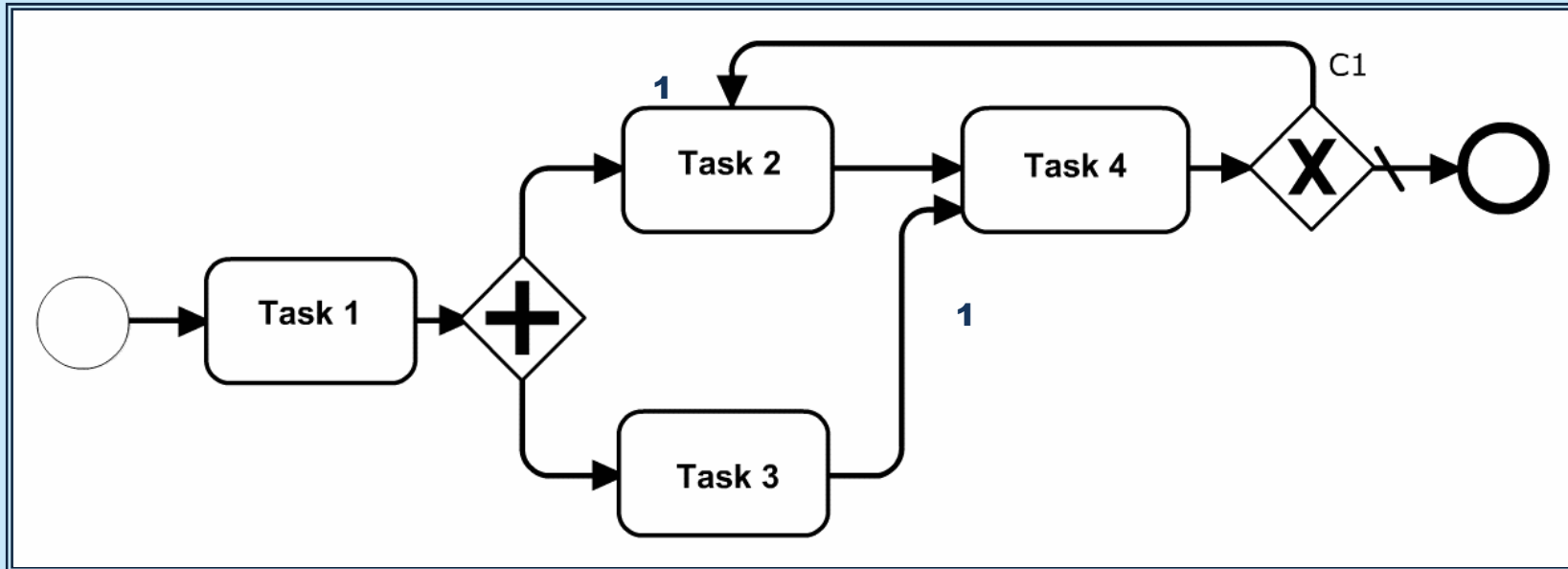
# Workflow Patterns ([workflowpatterns.com](http://workflowpatterns.com))

- I workflow patterns forniscono una base concettuale per la process technology. I Process Aware Information Systems (PAIS) moderni supportano i pattern.
- In un PAIS possono essere distinti pattern secondo quattro prospettive. Ogni prospettiva contiene decine di pattern:
  - ✓ *control-flow perspective*: cattura aspetti relativi alle dipendenze tra vari task quali parallelismo, scelta, sincronizzazione, ecc...
  - ✓ *data perspective* riguarda il passaggio dell'informazione, l'ambito di vita delle variabili, ecc...
  - ✓ *resource perspective* tratta l'allocazione, delegazione, ecc... di risorse al task
  - ✓ *exception handling perspective* tratta le varie cause di eccezione del business, e le azioni che occorre intraprendere.



## Pattern WCP-10: Arbitrary Cicles

- Rappresenta cicli in un modello di processo che ha più di un entry o exit point, quindi non è strutturato a blocchi (come ad es. il while ed il for) e quindi non prevede restrizioni sul formato del processo. Nell'esempio abbiamo due entry point<sup>1</sup>.

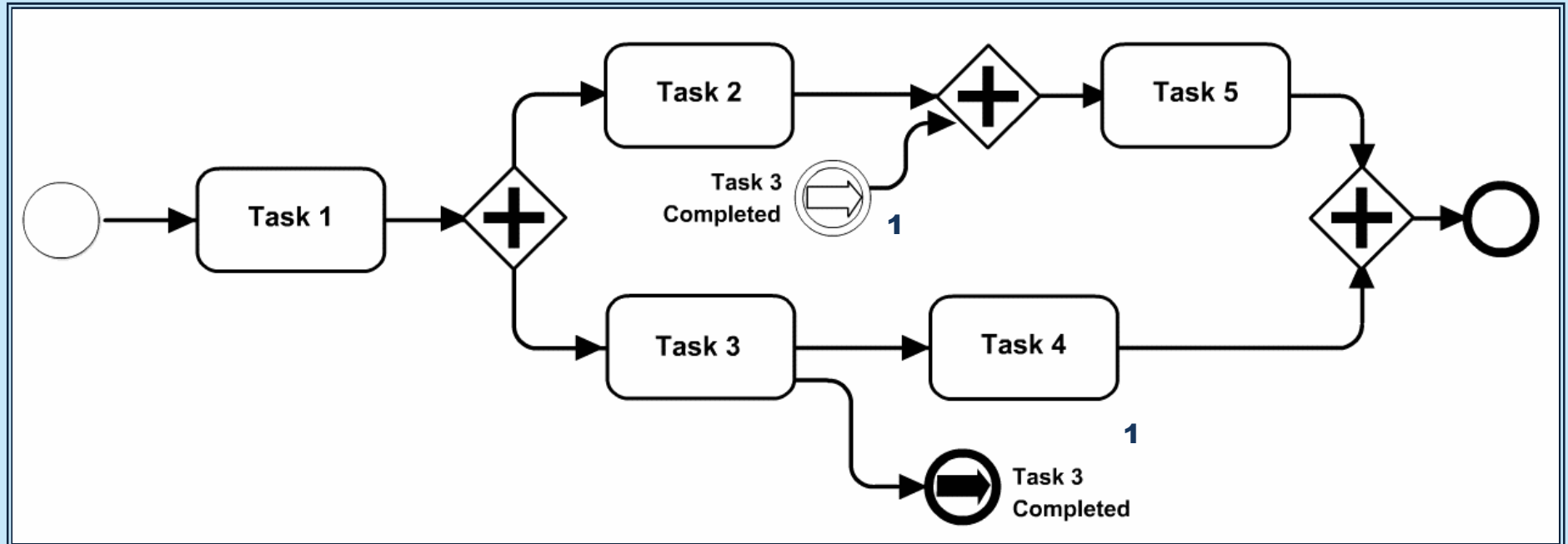


<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov12.swf>

- **Esempi** – Le situazioni inaspettate (es. mancanza di materie prime, disastro naturale, sciopero) possono impedire che il venditore spedisca in tempo utile al compratore l'esatto numero di prodotti. In tal caso, il venditore dovrebbe informare il cliente e chiedergli se (a) accettare l'ordine; (b) modificarlo; (c) cancellarlo. Nel secondo caso si ricomincia da capo con l'ordine (ciclo). Poiché questo ciclo ha tre punti di uscita, si modella con un arbitrary cycle pattern.

## Pattern WCP-18: Milestone

- Rappresenta un modello in cui una attività viene abilitata solo quando la istanza del processo di cui fa parte è in uno stato specifico, caratterizzato da un particolare punto di esecuzione detto milestone.

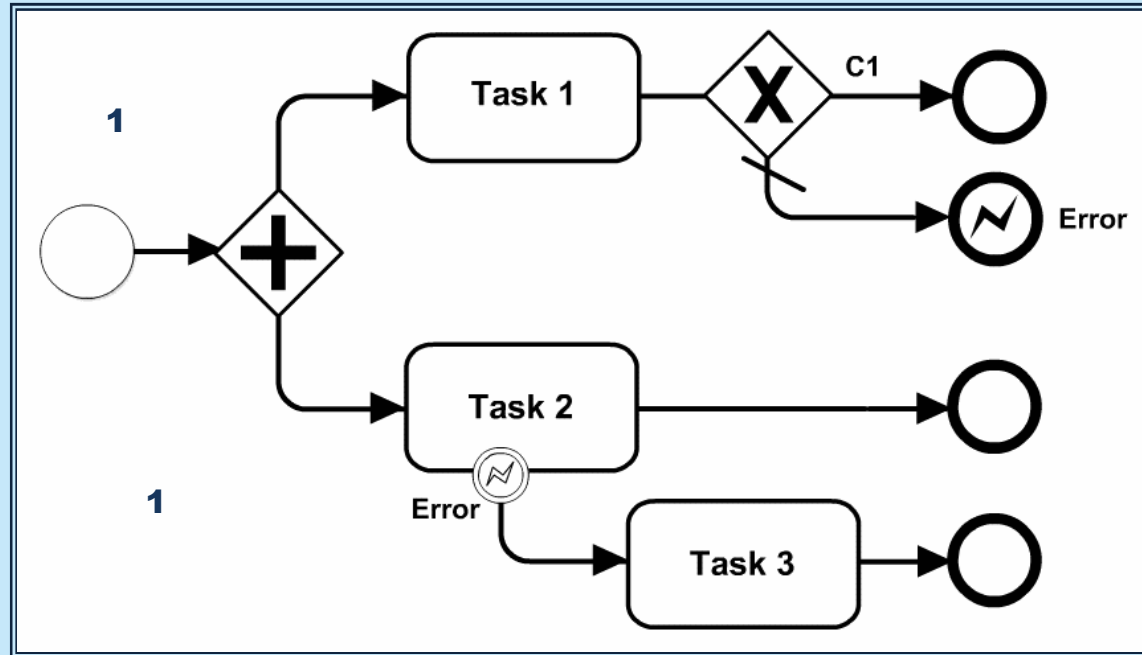


<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov13.swf>

- **Esempi** – Molte compagnie aeree consentono il cambio di una prenotazione se il biglietto non è stato ancora emesso. All'università, la registrazione di uno studente può avere luogo dopo l'apertura delle registrazioni e prima della loro chiusura.

## Pattern WCP-19: Cancel Activity

- Una attività abilitata viene ritirata prima che inizi la sua esecuzione. Se l'attività è già partita, essa viene disabilitata e, se possibile, l'istanza corrente viene arrestata e ritirata.



<http://www.iet.unipi.it/m.cimino/gpa/res/movie/mov14.swf>

- **Esempi** – L'attività di stima del danno viene intrapresa da due periti assicurativi. Una volta che il primo perito ha completato l'attività, il secondo viene arrestato. Gli acquirenti possono fermare la loro attività di ispezione del fabbricato in qualunque momento prima che essa inizi.

# BPMN Core: le definizioni

- Ogni elemento BPMN ha una struttura fatta da attributi e associazioni di elementi, definita nella specifica, tramite classi UML e XML Schema.
- Es. Text Annotation, nota del progettista per il lettore del diagramma.

Text Annotation allows a modeler to provide additional information

## Text Annotation attributes

Attributes	Description
<b>text:</b> string	Text is an attribute that is text that the modeler wishes to communicate to the reader of the Diagram.
<b>textFormat:</b> string	This attribute identifies the format of the text. It MUST follow the mime-type format. The default is "text/plain."

## Text Annotation XML schema

```
<xsd:element name="textAnnotation"
  type="tTextAnnotation" substitutionGroup="artifact"/>
<xsd:complexType name="tTextAnnotation">
  <xsd:complexContent>
    <xsd:extension base="tArtifact">
      <xsd:sequence>
        <xsd:element ref="text"
          minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="textFormat"
        type="xsd:string" default="text/plain"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="text" type="tText"/>
<xsd:complexType name="tText" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##any"
      processContents="lax" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

- Es.Timer Event.

### Timer Event



The `TimerEventDefinition` element inherits the attributes and model associations of `BaseElement`

#### TimerEventDefinition model associations

Attribute Name	Description/Usage
<b>timeDate:</b> Expression [0..1]	If the <i>trigger</i> is a Timer, then a <code>timeDate</code> MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, <code>timeDate</code> MUST NOT be set (if the <code>isExecutable</code> attribute of the <b>Process</b> is set to <i>true</i> ). The return type of the attribute <code>timeDate</code> MUST conform to the ISO-8601 format for date and time representations.
<b>timeCycle:</b> Expression [0..1]	If the <i>trigger</i> is a Timer, then a <code>timeCycle</code> MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, <code>timeCycle</code> MUST NOT be set (if the <code>isExecutable</code> attribute of the <b>Process</b> is set to <i>true</i> ). The return type of the attribute <code>timeCycle</code> MUST conform to the ISO-8601 format for recurring time interval representations.
<b>timeDuration:</b> Expression [0..1]	If the <i>trigger</i> is a Timer, then a <code>timeDuration</code> MAY be entered. Timer attributes are mutually exclusive and if any of the other Timer attributes is set, <code>timeDuration</code> MUST NOT be set (if the <code>isExecutable</code> attribute of the <b>Process</b> is set to <i>true</i> ). The return type of the attribute <code>timeDuration</code> MUST conform to the ISO-8601 format for time interval representations.

#### TimerEventDefinition XML schema

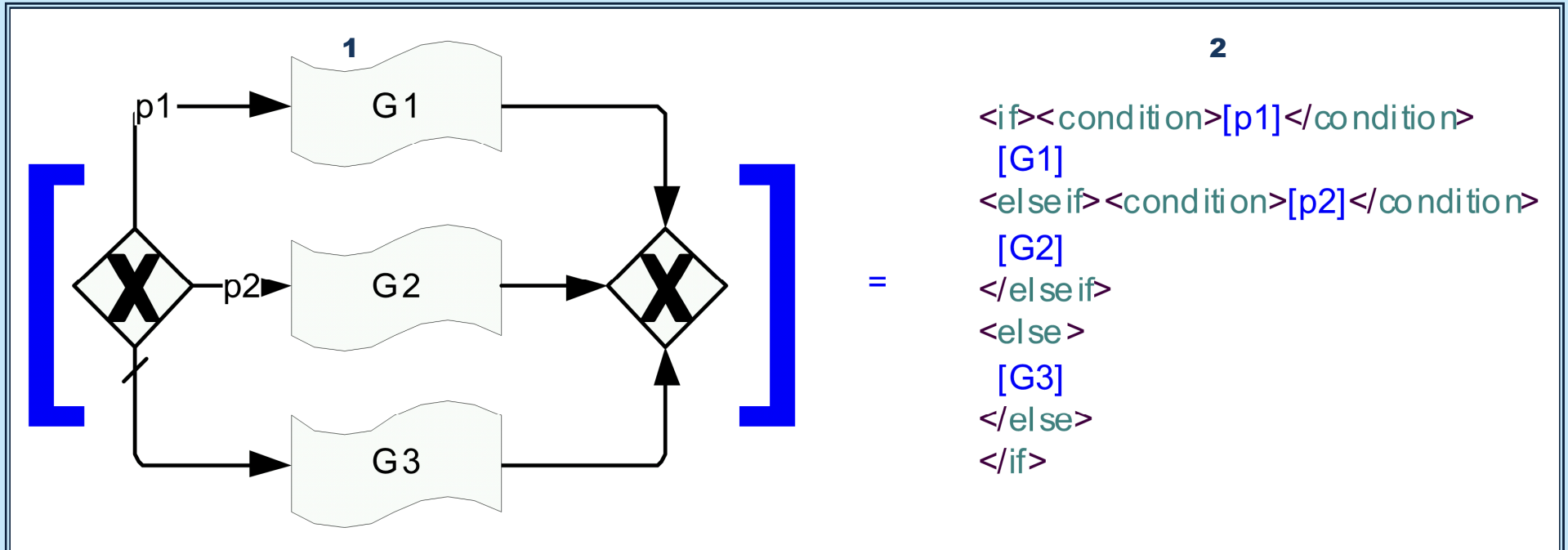
```
<xsd:element name="timerEventDefinition" type="tTimerEventDefinition" substitutionGroup="eventDefinition"/>
<xsd:complexType name="tTimerEventDefinition">
  <xsd:complexContent>
    <xsd:extension base="tEventDefinition">
      <xsd:choice>
        <xsd:element name="timeDate" type="tExpression" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="timeDuration" type="tExpression" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="timeCycle" type="tExpression" minOccurs="0" maxOccurs="1"/>
      </xsd:choice>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

# Serializzabilità di BPMN in WS-BPEL

- La conversione di processi BPMN in WS-BPEL avviene per Pool, e non per un intero diagramma.
- Non tutti i processi BPMN possono essere tradotti adeguatamente, perché BPMN consente di disegnare percorsi arbitrari, mentre WS-BPEL contiene diverse restrizioni.
- Il processo BPMN, per essere mappabile in WS-BPEL deve essere *sound* (van der Aalst 2003).
- Un workflow è *sound* se è solo se: (i) non contiene parti inerti, ossia task che non possono mai essere eseguiti; (ii) è sempre possibile terminare da un qualsiasi stato raggiungibile; (iii) nel momento in cui il workflow termina, vi sono token solo nei punti di terminazione;
- La *soundness* consente di escludere errori logici, quali deadlock e livelock. Il *deadlock* è una situazione di stallo del processo in cui non può essere eseguita nessun'altra attività. Un *livelock* è una situazione in cui alcuni loop di task sono eseguiti continuamente, e non c'è un percorso che porta ad una terminazione corretta.
- Tali proprietà si studiano tramite la matematica dei grafi. Infatti *gateway* e *sequence flows* del processo formano grafi e sottografi diretti. Si dice blocco (*block*) del

diagramma un sottografo che è connesso al resto del grafo solo attraverso esattamente due sequence flow: uno di ingresso ed uno di uscita.

- Esempio di data-based exclusive choice, controllata da due predicati p1 e p2, con tre blocchi (G1,G2,G3)<sup>1</sup> e relativo mapping a WS-BPEL<sup>2</sup>.



- La *soundness* non garantisce che il workflow funzioni come inteso. Molti errori possono essere trovati solamente usando conoscenza del contesto. Ad esempio, in un ordine di centinaia di beni è accettabile che la spedizione di beni sia interrotta se qualcuno dei beni non è disponibile? Basandoci sul contesto si potrebbe decidere se ciò è accettabile o meno. → Context-based validation.

# Cenni alla estendibilità di BPMN 2.0

- BPMN 2.0 introduce un meccanismo di estendibilità che permette l'estensione dei costrutti standard con attributi addizionali.
- Gli attributi estesi non devono porre in contraddizione la semantica di elementi BPMN core, e dovrebbero mantenerne il look-and-feel.
- La specifica BPMN definisce gli elementi che devono definire una estensione BPMN, con tanto di XML schema della estensione.
- Ad oggi, sono state proposte estensioni con caratteristiche addizionali di modellazione quali (Saeedi et al. 2010):
  - ✓ business goal e misura delle prestazioni;
  - ✓ requisiti di costo ed affidabilità;
  - ✓ requisiti di sicurezza;
  - ✓ modellazione dati;
  - ✓ visualizzazione dei business process;
  - ✓ vincoli temporali;
  - ✓ allocazione di risorse umane e vincoli di autorizzazione;
  - ✓ dettagli implementativi e tecnici della coreografia.

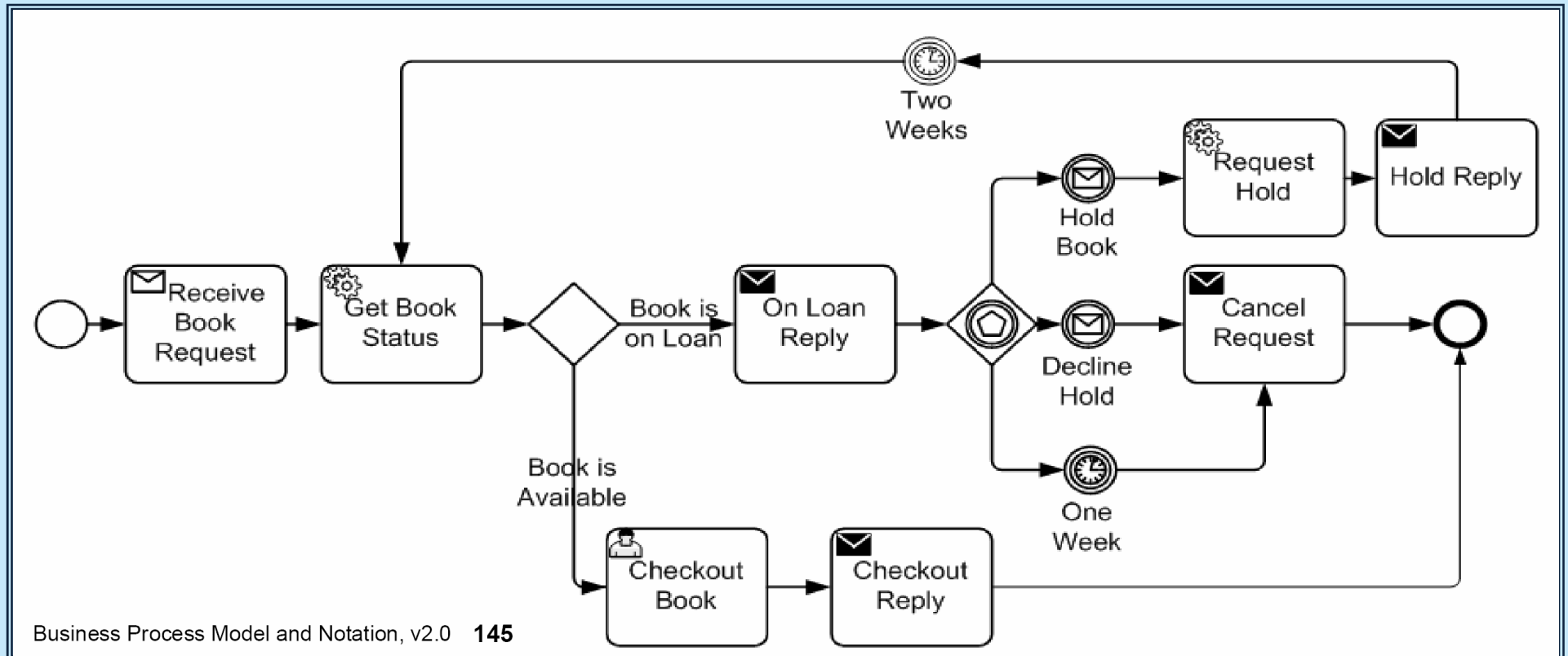


# Casi di studio

- Nel seguito vengono illustrati dei diagrammi BPMN appartenenti a vari ambiti, da usare come casi di studio di laboratorio.
- Non sono modelli di processo eseguibili, sono orientati esclusivamente agli aspetti organizzativi.
- Procedura suggerita:
  - Individuare un caso di studio (scelta libera).
  - Descriverlo in linguaggio naturale.
  - Rappresentare il diagramma su uno strumento per la modellazione di processo.
- Suggerimenti:
  - ✓ Nella descrizione in linguaggio naturale, inserire come note a piè di pagina i riferimenti (documento e numero di pagina esatto) che sono stati consultati.
  - ✓ Per avere altre informazioni sul diagramma, consultare il riferimento indicato in basso al diagramma stesso;
  - ✓ Per avere informazioni su uno specifico costrutto, consultare il BPMN 2.0 poster come indice visuale, identificare le keyword adatte e fare poi una ricerca nella specifica BPMN 2, consultando gli indici (generale e analitico) oppure tramite la funzione “trova” di acrobat reader;
  - ✓ Cercare di adoperare uno stile di scrittura essenziale, con vocabolario minimo, senza sinonimi, per evitare ambiguità. Dare al testo una struttura narrativa, ossia senza elenchi o tabelle.

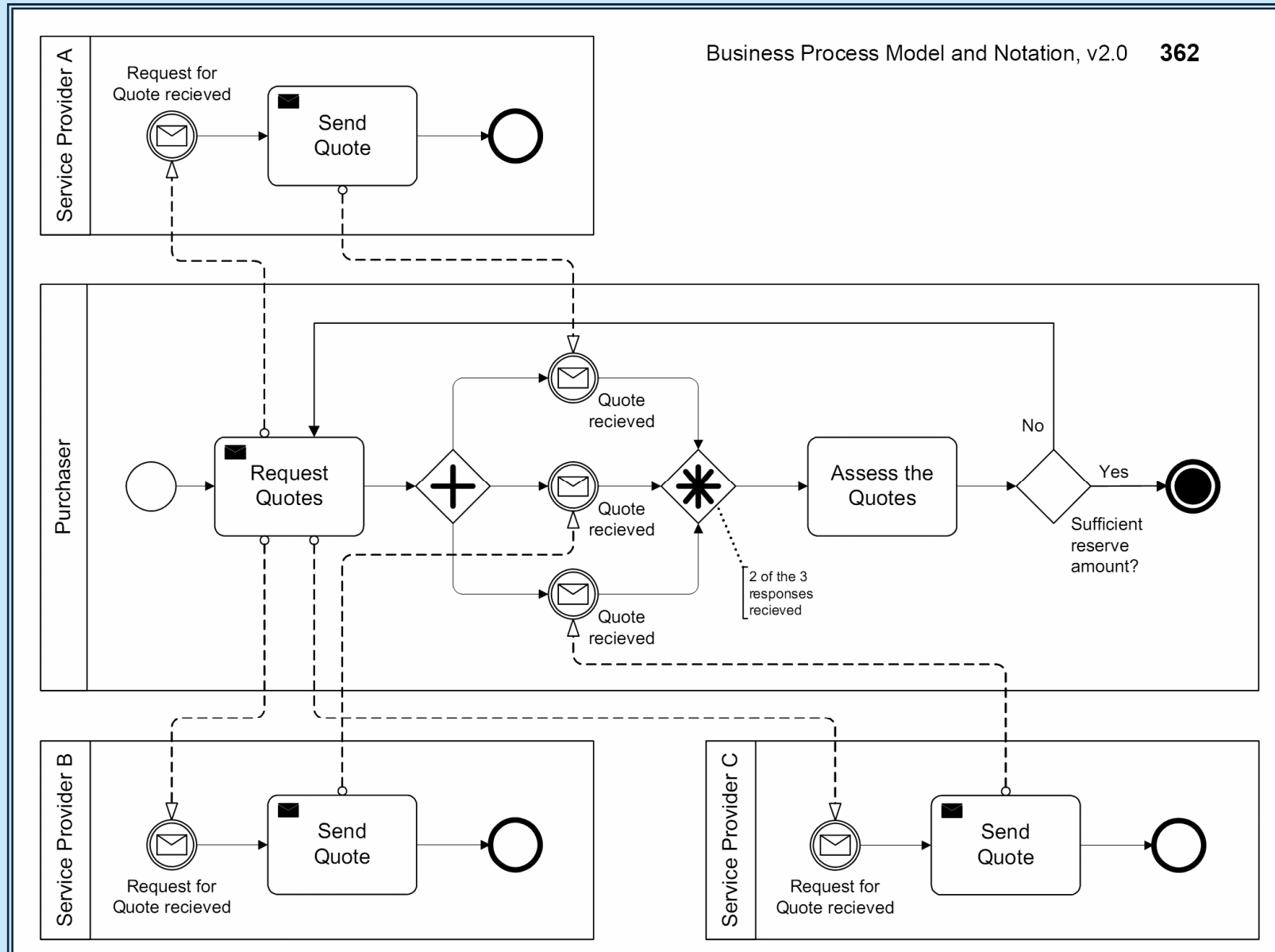
# I) Prestito bibliotecario

- Il seguente processo (orchestrazione) riguarda la gestione del prestito bibliotecario.



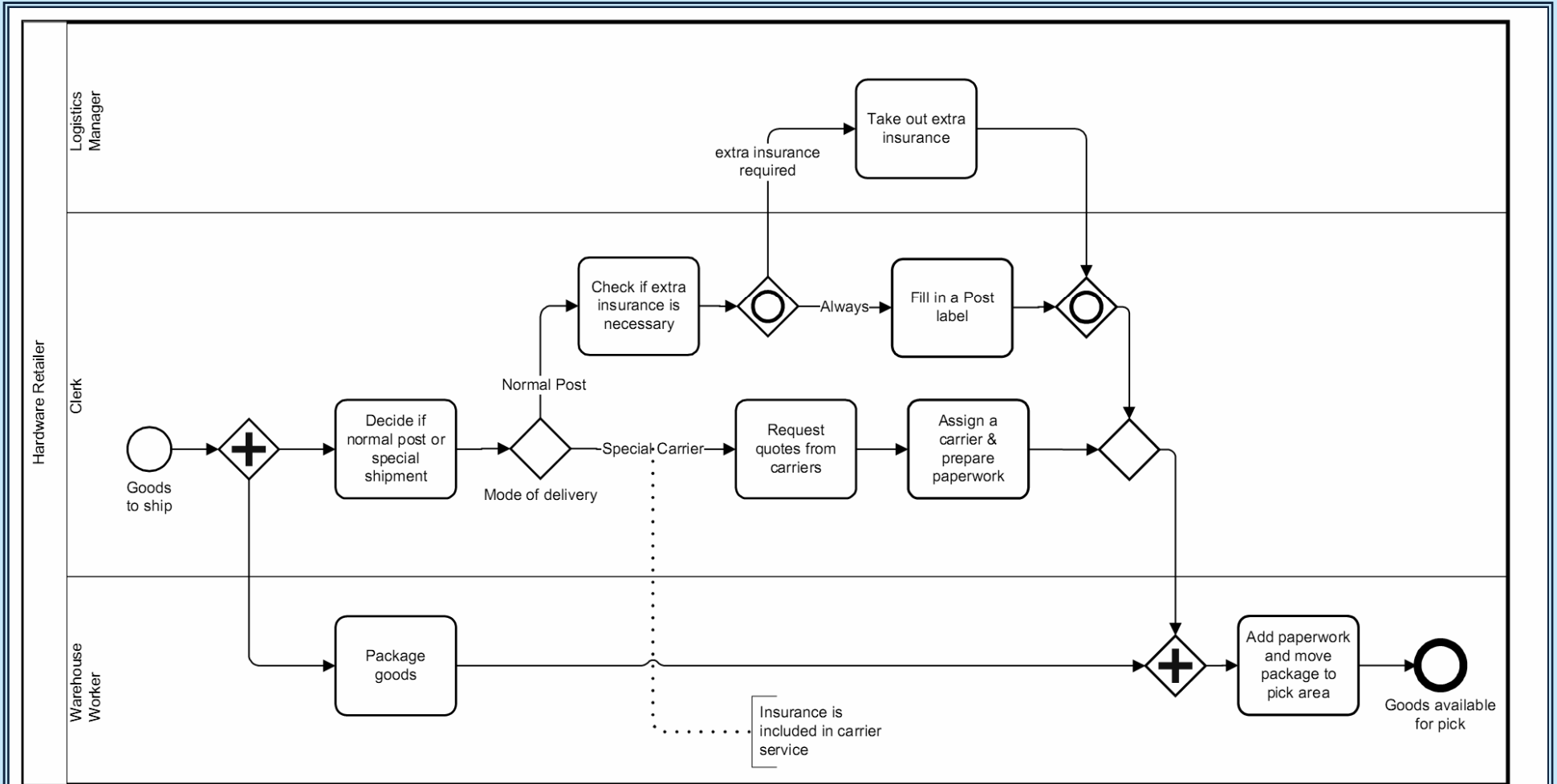
## II) E-tendering

- Il termine e-tendering (offerta elettronica) viene usato per designare l'uso di internet per fare delle richieste di preventivo da una parte e ricevere le offerte dall'altra.



### III) Spedizioni di un rivenditore di HW

- La modellazione riguarda i passi che un rivenditore di hardware al dettaglio deve compiere prima di spedire i beni ordinati al cliente.

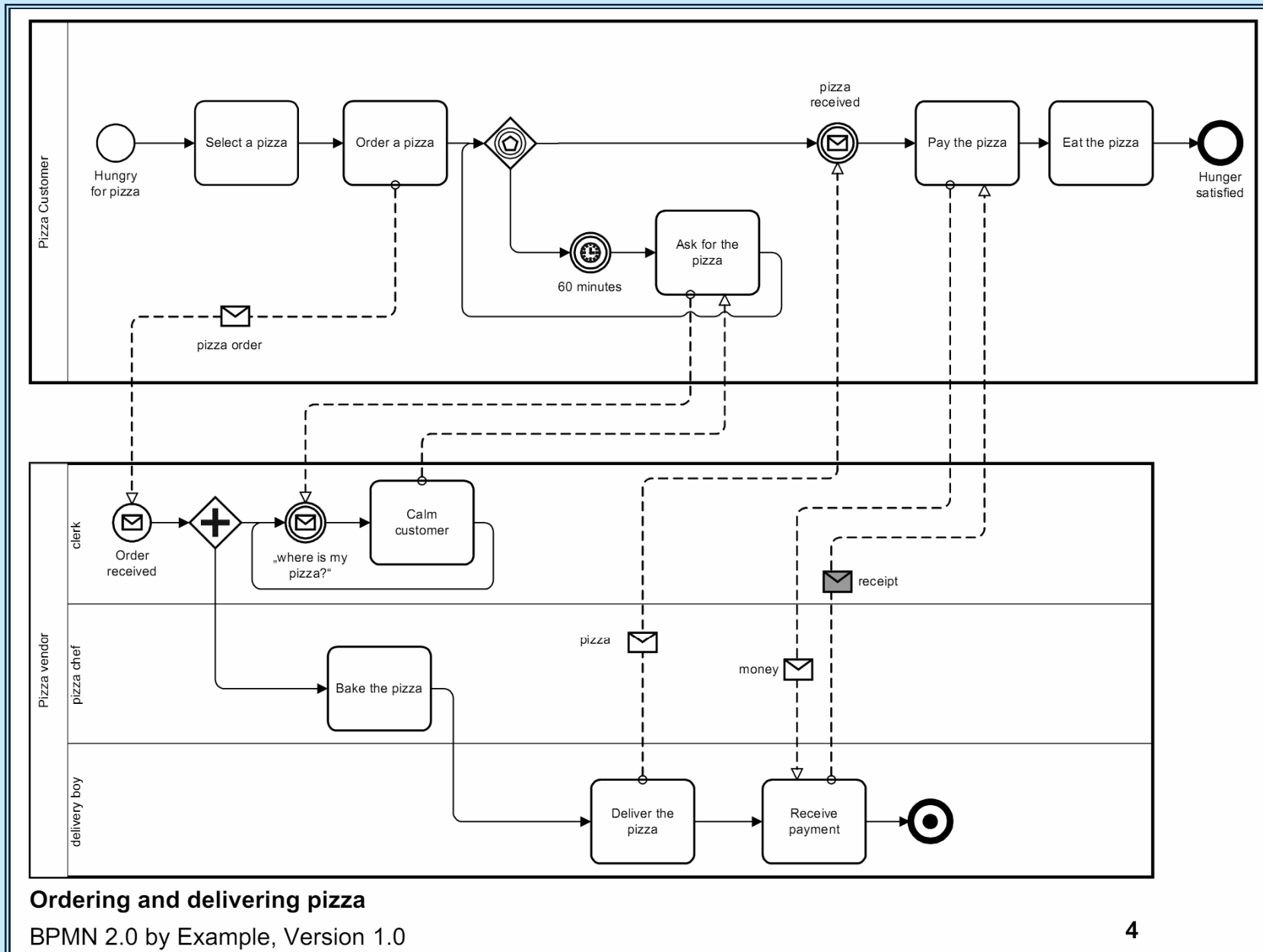


Shipment Process of a hardware retailer

BPMN 2.0 by Example, Version 1.0

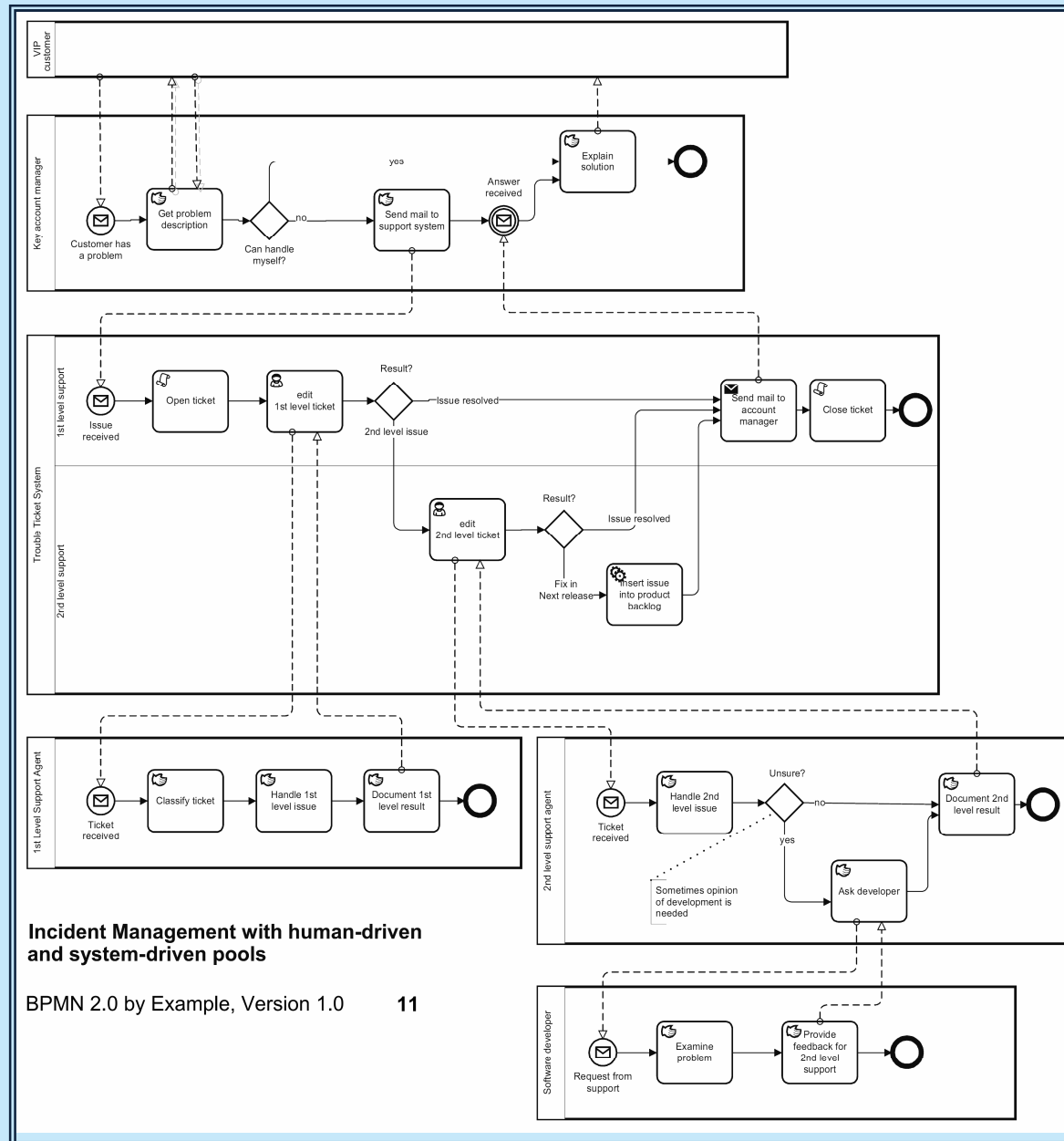
# IV) Pizza collaboration

- Modello di interazione tra un cliente, che ordina una pizza, e un venditore.



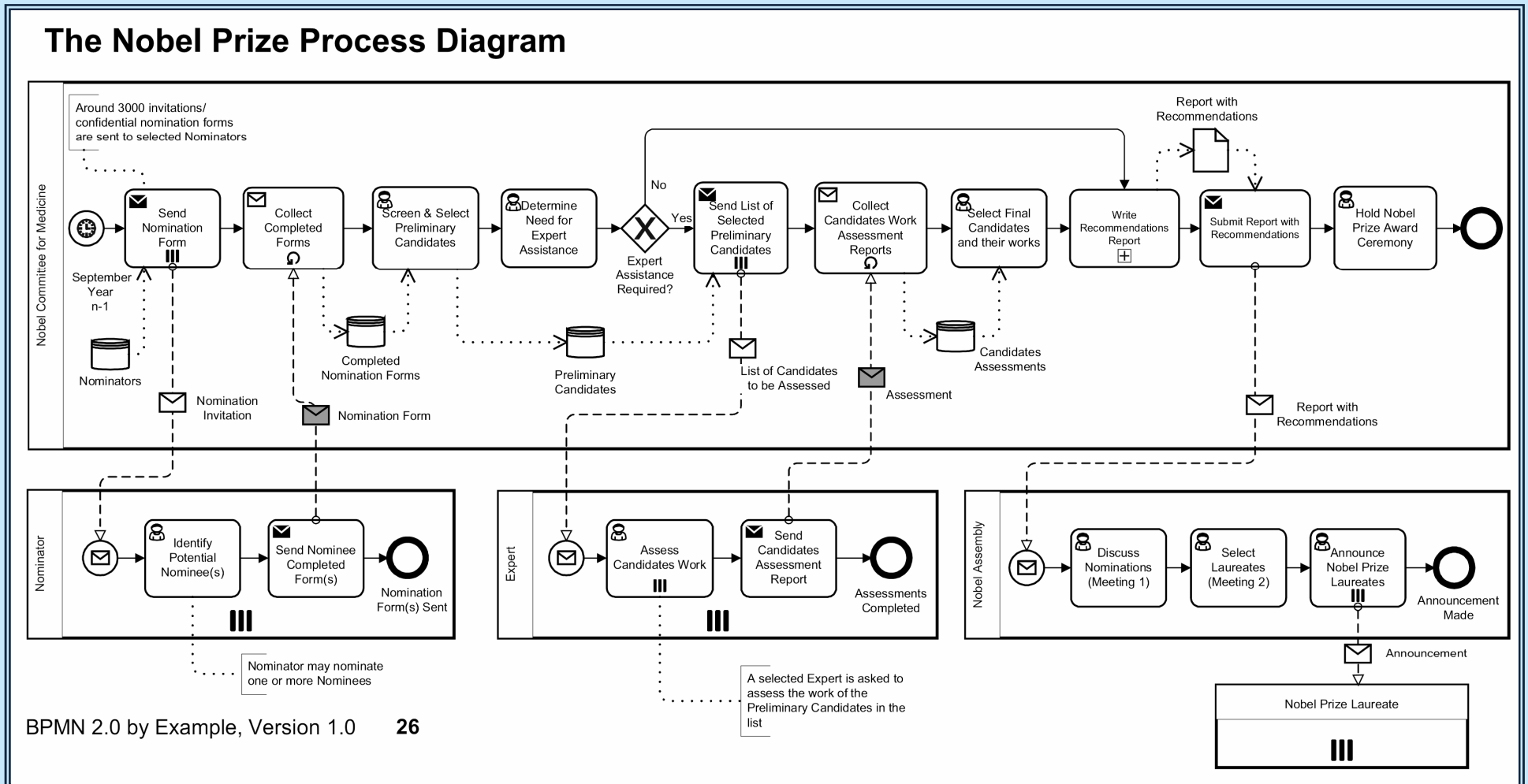
# V) Supporto in problemi software

- Come un produttore di software gestisce un problema segnalato dal cliente.



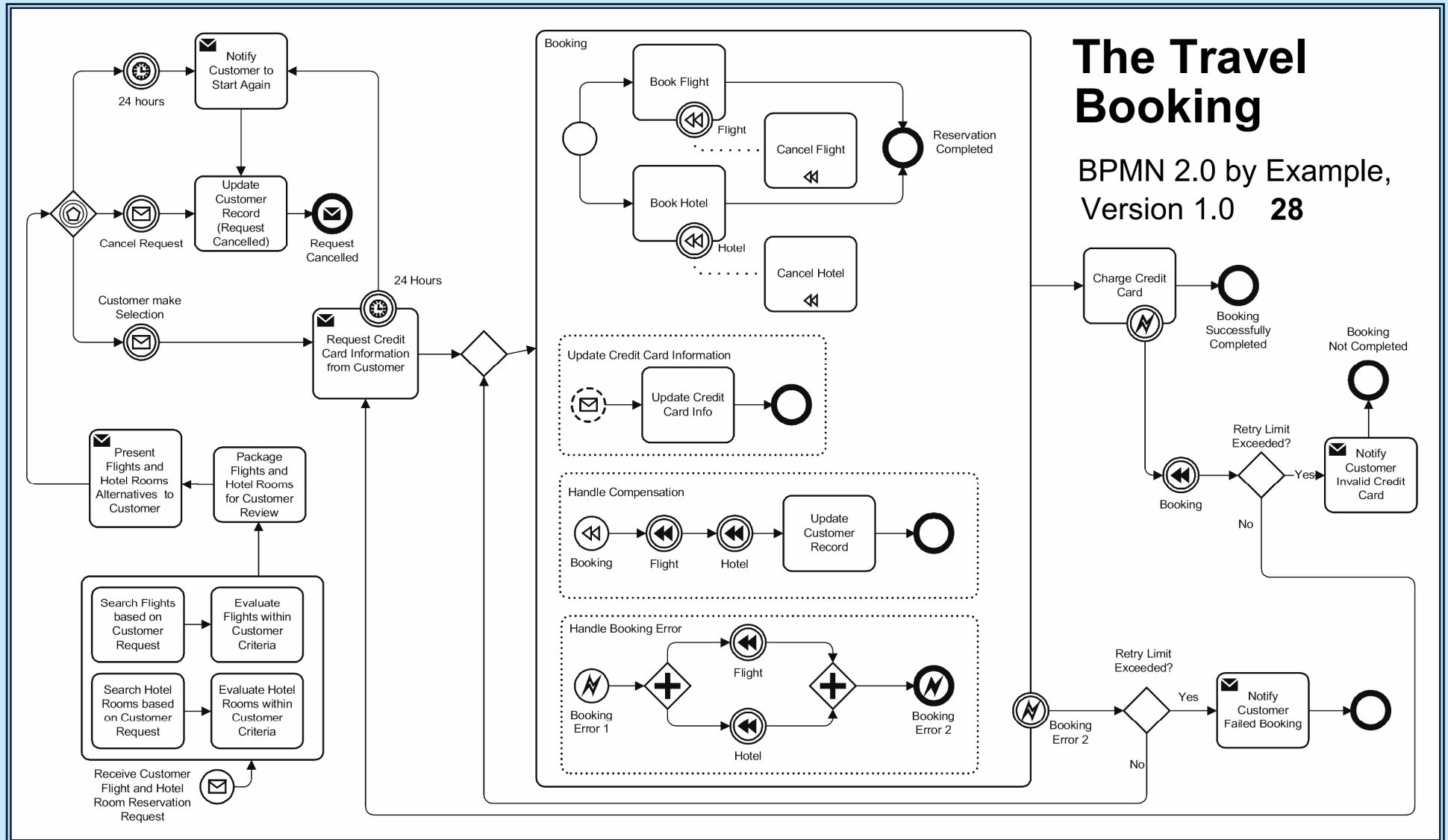
# VI) Assegnazione del Premio Nobel

- L'assegnazione di un Premio Nobel è un processo lento e delicato, che ogni anno coinvolge circa 3000 candidati, solo nella sezione Medicina.



# VII) Travel Booking

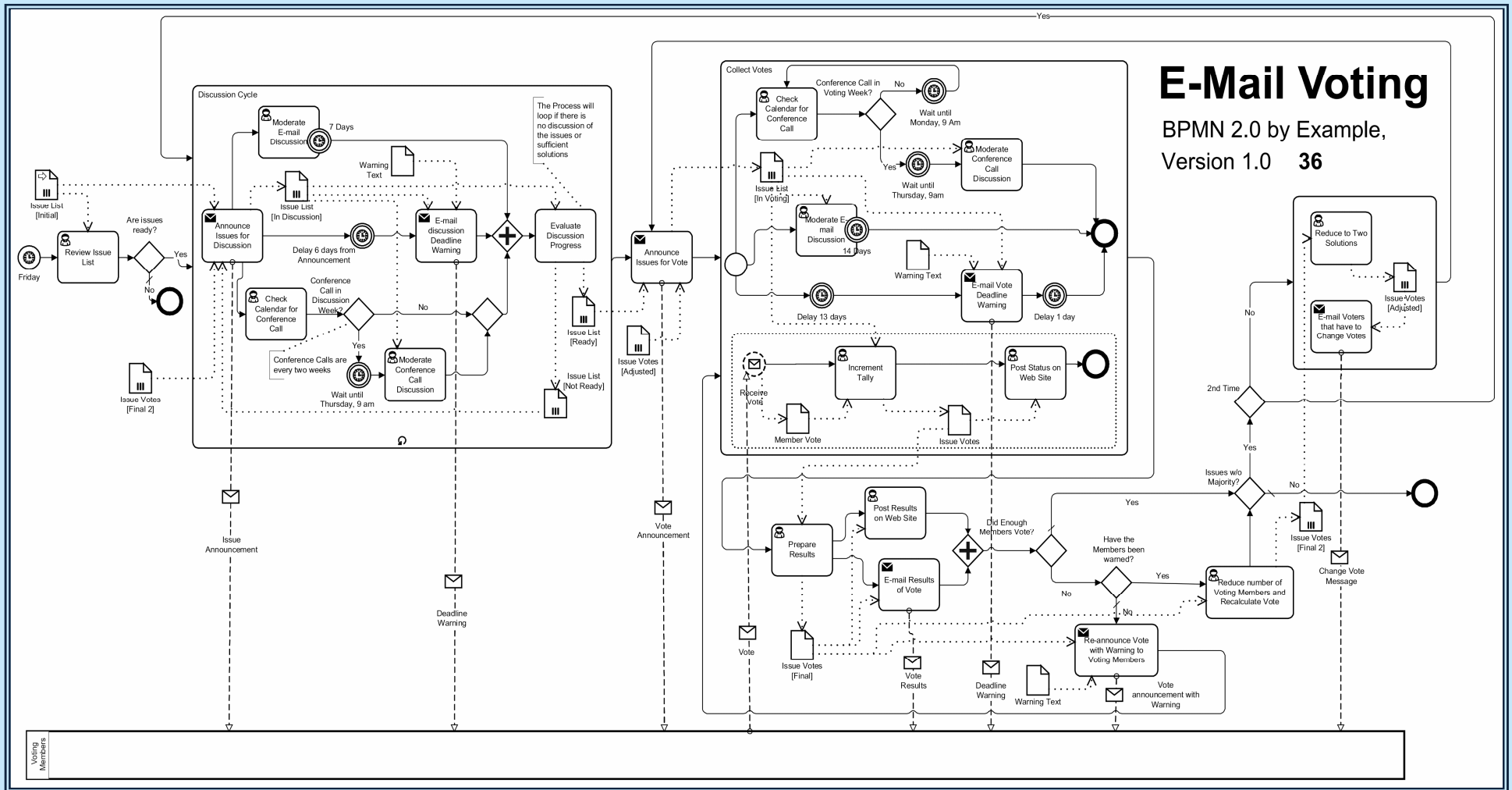
- Una agenzia viaggi gestisce la richiesta di un cliente di una prenotazione di viaggio, che include la prenotazione di volo ed hotel.





# VIII) E-mail voting

- Gestione della votazione tramite messaggi, da parte di membri di un gruppo di lavoro



# Riferimenti

- [1] BPM Offensive Berlin, *BPMN 2.0 Poster*;
- [2] OMG, *BPMN 2.0 Specification*, January 2011;
- [3] OMG, *BPMN 2.0 by Example*, June 2010.