IBM Business Process Management

Corso di sviluppo su BPM Standard 8.5

06 Giugno 2016

Salvatore Traina - traina@netsitech.com

IBM Business Process Management

Struttura del Corso

- **Unità 1**: Panoramica sul BPM
- **Unità 2**: Process Designer e Applicazioni BPM
- **Unità 3**: Componenti BPM
- **Unità 4**: Integrazione Servizi

Unità 1: Panoramica sul BPM

- Uso del BPM e i vantaggi che ne conseguono
- Struttura e architettura del prodotto
- Presentazione Process Center e Process Server
- Presentazione delle console di gestione (Process Admin, Process Portal);
- BPM e meccanismi di sicurezza/autenticazione.

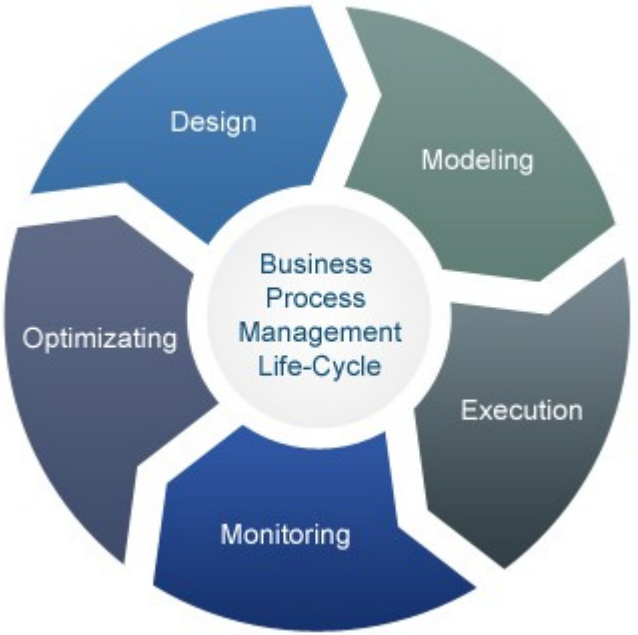
Uso del BPM e i vantaggi che ne conseguono

- Ottimizzare le operazioni di business tramite il monitoraggio e analisi continui dei processi
- Accelerare il completamento delle attività tramite valide funzionalità di collaborazione (Collaboration Teams)
- Sicurezza e governance intuitiva
- Offre ambiente altamente integrato, scalabile, aumentando la produttività degli utenti.

Dall'esigenza reale al BPM...

- Individuare fasi processo → Attività manuali e automatiche (Tasks, Services)
- Individuare gli attori (teams)
- Struttura per iterazioni → Evoluzione continua
- Trasformazione modello (BPMN)
- Design → Esecuzione → Monitoraggio → Ottimizzazione

BPM Life Cycle



I prodotti del BPM 8.5.5

- **Process Center** → Repository BPM artifacts (Apps, BPD, Toolkits, Tasks, Services) used to deploy application to Process Server
- **Process Designer** → GUI (eclipse based) used to design and build BPM artifacts to Process Center
- **Process Servers** → Execution envirnoments to run Business Process Application

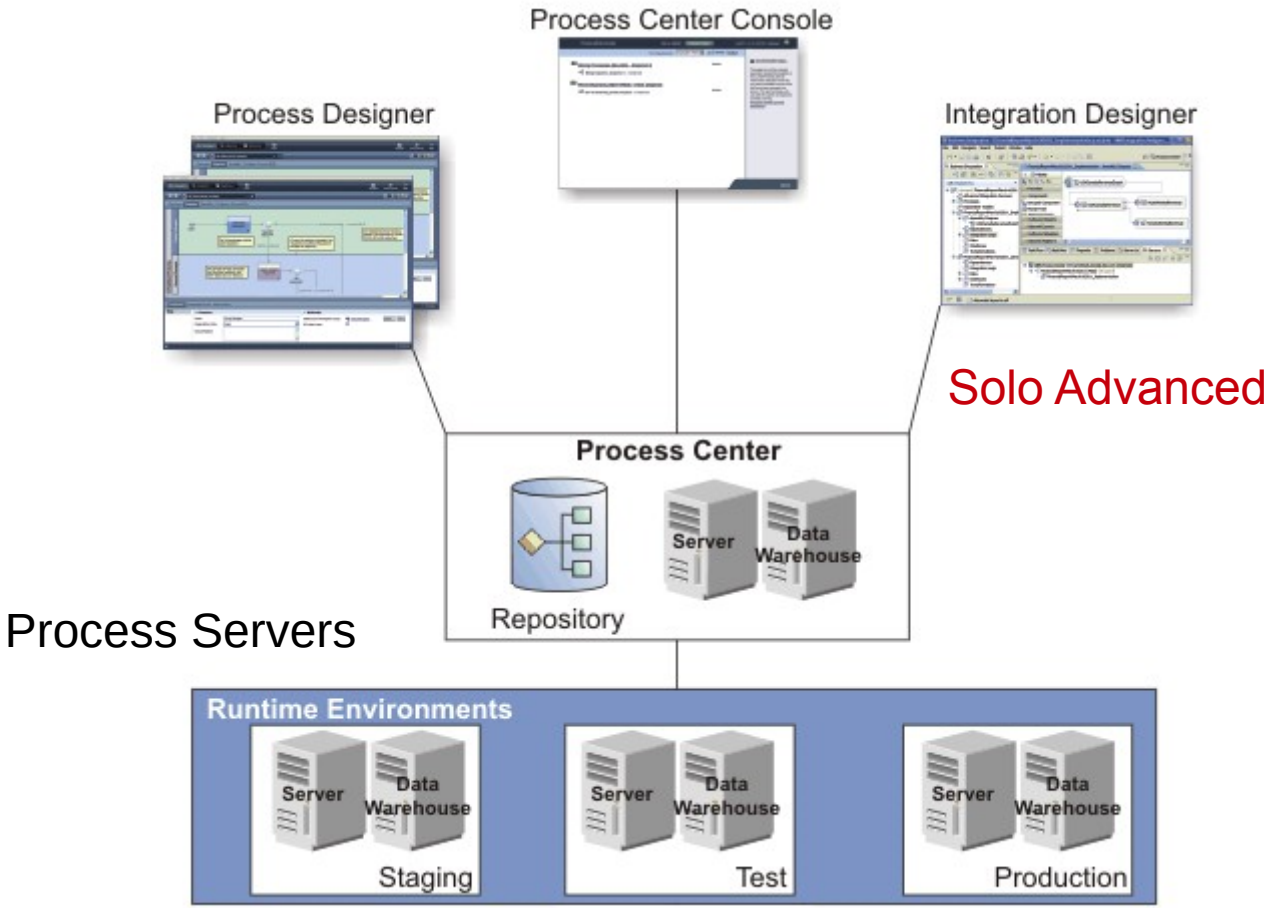
Process Center (Development)

- Process Designer Interface
- Process Center Console
- Process Server
- Performance data warehouse

Process Servers (Test, staging, prod)

- Process Server
- Performance data warehouse

Architettura Generale del BPM 8.5.5



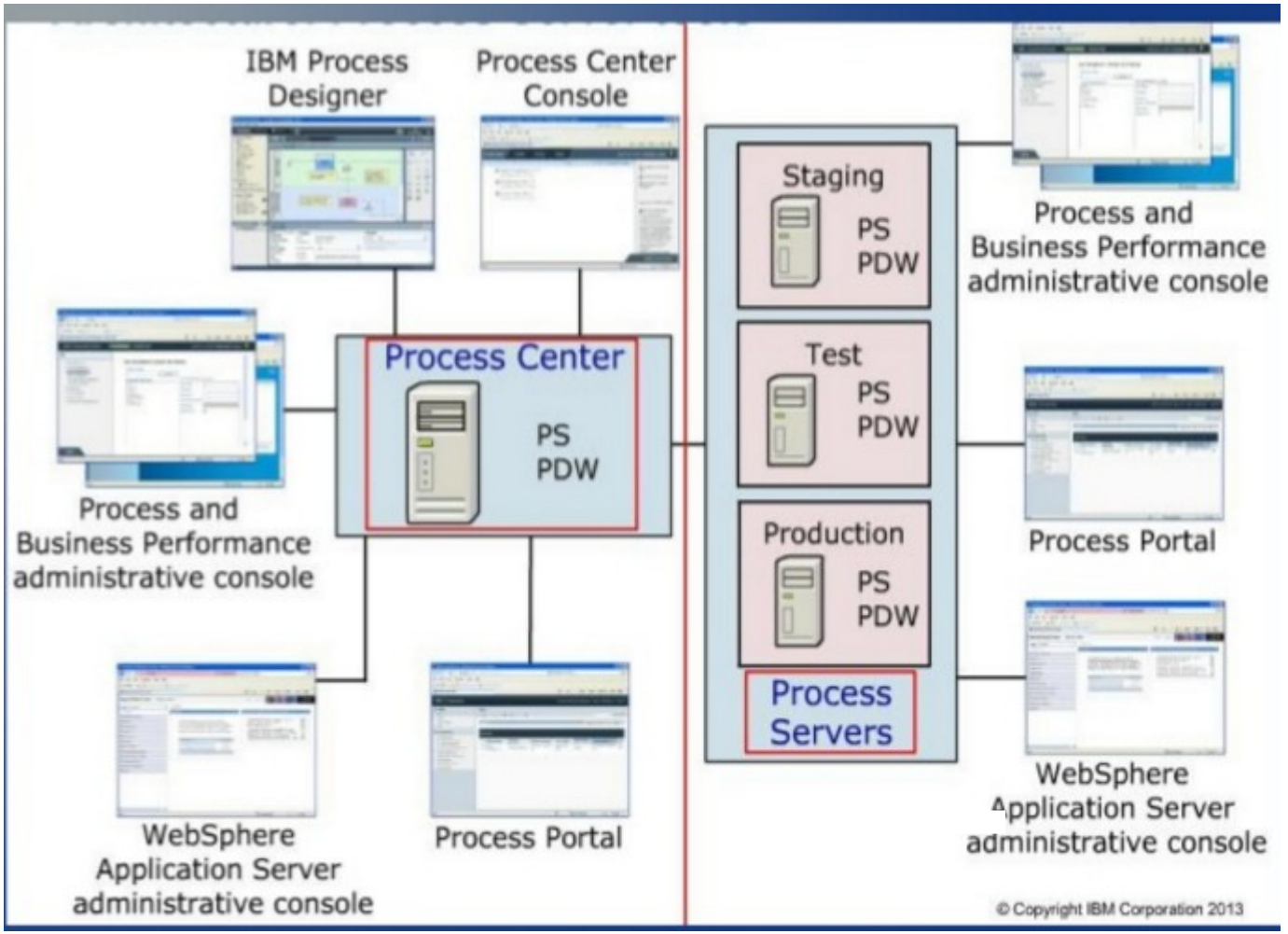
WebSphere Application Server

Applicazioni del BPM 8.5.5 (Process Center vs Process Servers)

Process Center

- Process Center Console
- Process Designer
- Process Admin Console
- Process Portal
- WAS Admin Console

PC e PS



Versioni del BPM 8.5.5: Express / Standard / Advanced

Principali caratteristiche

Per Dettagli: http://www.ibm.com/support/knowledgecenter/SS964W/com.ibm.wbpm.main.doc/topics/ibmbmp_overview.html

Capability	Express	Standard	Advanced
WebSphere Lombardi Edition compatible execution	X	X	X
Process Designer (BPMN)	X	X	X
Collaborative editing / immediate playback	X	X	X
Interactive "process coach" user interfaces	X	X	X
ODM-based process rules	X	X	X
Process Portal	X	X	X
Real-time monitoring and reporting	X	X	X
Performance analytics & optimizer	X	X	X
Performance Data Warehouse	X	X	X
Process Center / shared asset repository	X	X	X
Unlimited process authors and end-users	200 users / 3 authors	X	X
High availability: clustering and unlimited cores	• 4 cores production • 2 cores development • No cluster	X	X
Process Federation Server		X	X
WebSphere Process Server compatible execution			X
Integration Designer (BPEL / SOA)			X
Built-in enterprise service bus (ESB)			X
Transaction support			X
Integration adapters			X
Flexible Heritage Process Portal spaces			X
Advanced platform support (Linux on System z®, Linux on Power, IBM AIX®, Solaris)		X	X
Basic Case Management feature	X	X	X

Process Center (Development)

Repository centralizzato dove sono create e salvate le **Business Process Application**.

Si compone di:

- **Process Server (PS)** → ambiente di esecuzione delle Business Process Application
- **Perfomance Data Warehouse** (**PDW)** → Raccoglie dati di performance

Process Center Console (Development)

Applicazione di gestione del **Process Center**.
Permette il deploy delle Business Process Application (BPA) sui **Process Servers**

Permette la gestione delle Snapshot di una BPA

Snapshot: una specifica versione di una BPA → Craere per ogni deploy

Possono esistere più snapshot attive per una stessa BPA.

Migrazione Snapshot delle istanze di processo attive

Esploriamo la Process Center Console ...

https://ibmbpm.netsitech.com/ProcessCenter

Process Designer Interface (Development)

Applicazione desktop (eclipse based) per la gestione di tutti gli artifacts BPM.

Salva tutti gli artifacts su **Process Center**.

Consente agli sviluppatori di creare, gestire, testare, e ottimizzare le
Business Process Definition (BPD)

BPD: Rappresenta il flusso (modello logico) di processo → notazione BPMN

- **Designer** → Permette la creazione, modifice e implementazione dei modelli BPM
- **Inspector** → Permette il test dei modelli durante la loro implementazione
- **Optimizer** → Scenari di simulazione

Dettaglio in Unità 2

Process Server

Ambiente di esecuzione delle Business Process Application.

Esegue su **WebSphere Application Server (WAS 8.5)**

Si compone di:

- **BPD Engine**
- **Service Engine**
- **Event Manager**

3 Process Server (Test, Staging, Production) + PS del Process Center (Development)

Process Administrative Console

Applicazione che consente la gestione del Process Server.

- **Gestione Server** → Utenti, Gruppi, Eventi
- **Process Inspector** → Interrogazione processi
- **Applicazioni** → Gestione snapshot

Esploriamo l'Administrative Console...

https://ibmbpm.netsitech.com/ProcessAdmin

Process Portal

Applicazione che si interfaccia al **Process Server** e consente agli utenti finali l'utilizzo delle **Business Process Application**.

- Elenco delle attività (human task) disponibili per utente
- Presa in carico di una attività
- Completamento di una attività
- Collaboration team
- Stato di avanzamento di un processo

Utilizzo Business Process Application mediante:

- Process Portal Standard
- Process Portal Customizzato

} Run Process Server

- Portale Esterno (GUI + Integrazione API REST) → External Env.

Esploriamo il Process Portal ...

https://ibpmbpm.netsitech.com/ProcessPortal

API REST Tester Console

Applicazione collegata al **Process Server** per il testing mediante le **API REST** Standard fornite dal BPM.

Esclusivo uso per sviluppatori / sistemisti

Esploriamo le API REST ...
https://ibmbpm.netsitech.com/bpmrest-ui/BPMRestAPITester

Autenticazione e Sicurezza

Utenti e gruppi possono essere definiti

- In ambito BPM (Process Admin)
- In ambito WAS
- Sistema terzo → Estensione User Registry (**CUR**)

Sincronizzazione Registri federati

Somma

Utenti e gruppi definiti in ambito BPM non possono essere modificati a livello WAS e viceversa

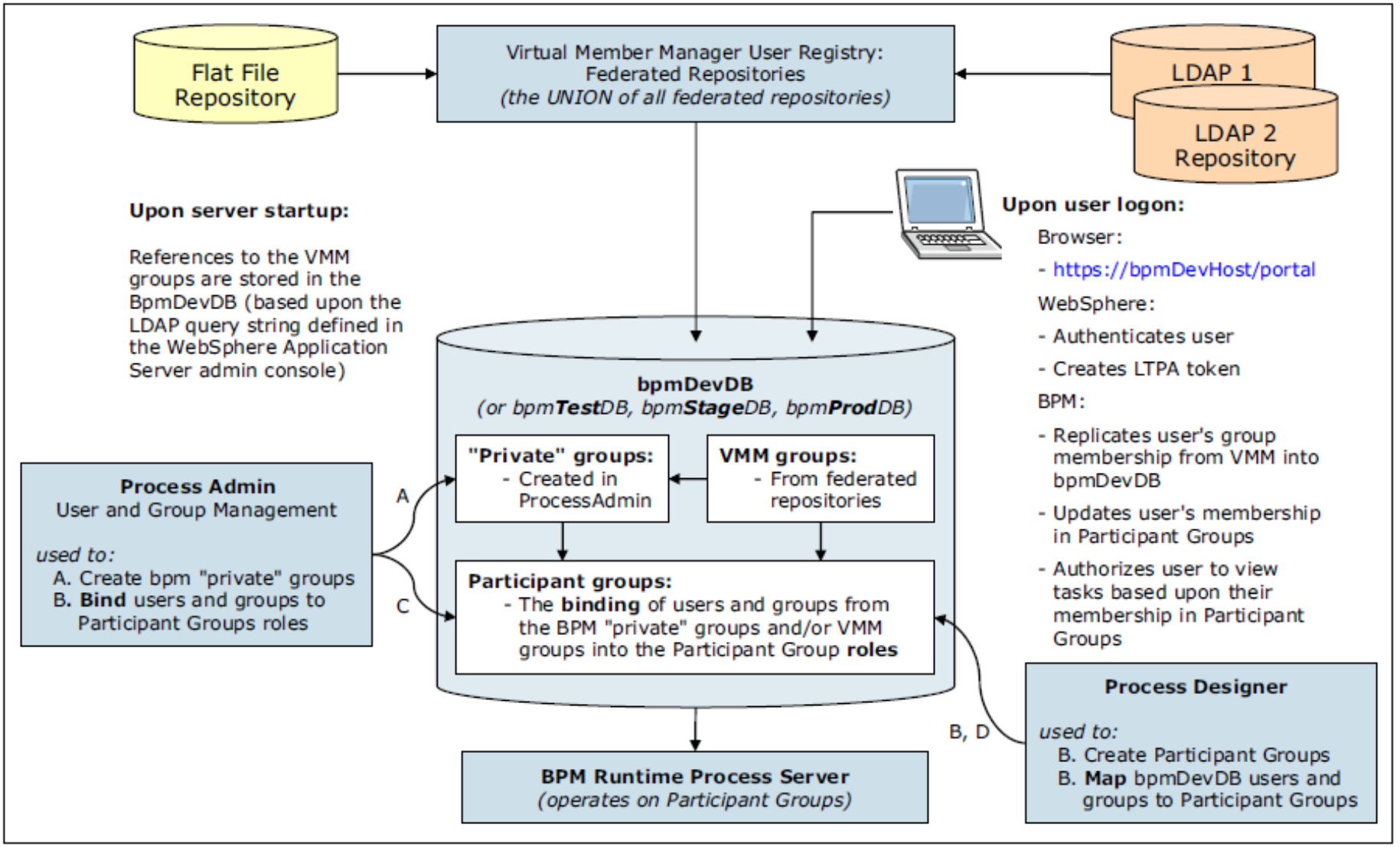
Team: Aggregazione di utenti, gruppi e altri team

Sicurezza legata ai Teams (Human Task, Swimline) → binding dinamico di utenti e gruppi (Admin Console)

Gruppi predefiniti BPM (tw_*)

Autenticazione esterna → **LTPA** → Custom **TAI**

Autenticazione e Sicurezza

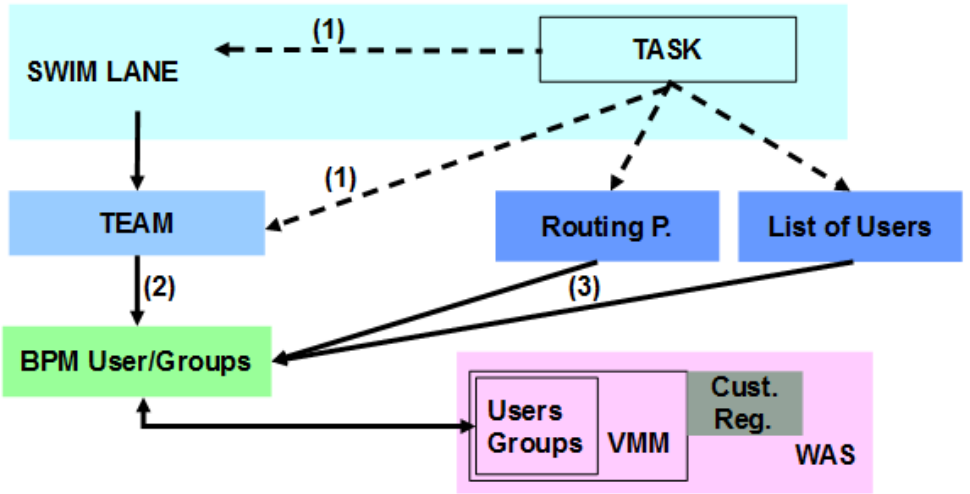


Autenticazione e Sicurezza

I task possono essere assegnati a dei team sulla base di una assegnazione diretta oppure in virtù del team associato alla "swimlane" a cui appartiene il task nella mappa di processo.

In produzione, la raccomandazione generale è quella di NON usare il Process Admin del BPM per creare utenti, ma definirli in WAS e/o nell'LDAP e/o nel custom registry aziendale, e renderli disponibili al BPM.

E' invece prassi creare "gruppi" specifici in BPM (che vengono conservati sul DB del BPM), chiamati anche BPM "**private groups**".



Unità 2: Process Designer e Applicazioni BPM

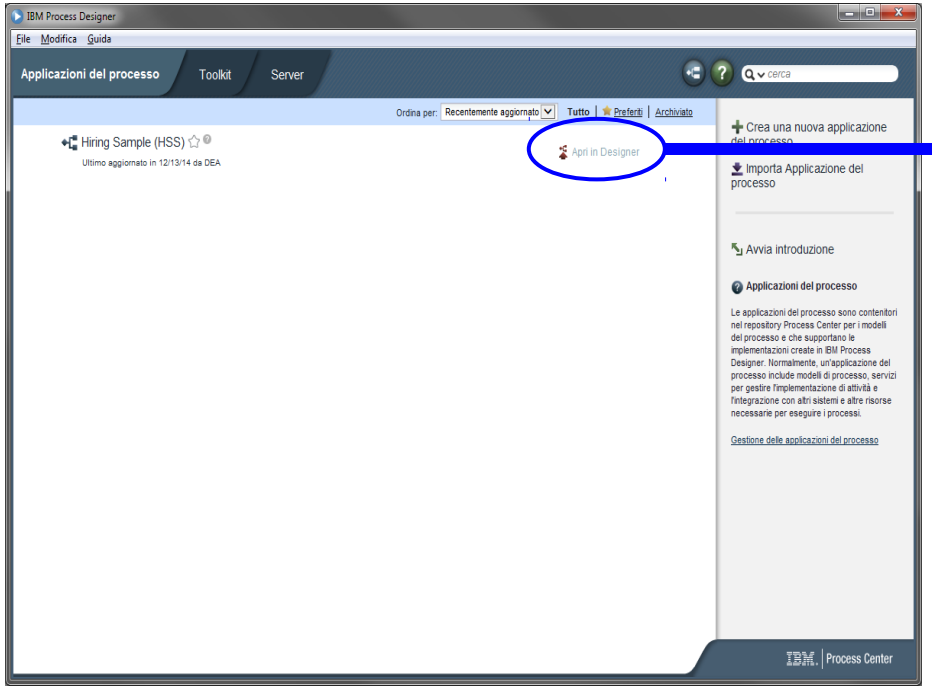
- Presentazione del Process Designer
- Struttura di una Process Application / BPD
- Ciclo di vita delle Process Instance
- Organizzazione in toolkit, snapshot, configurazione (variables environment) e deploy di una Process Application
- Definizione di utenti, gruppi e teams
- Interazione mediante API Rest
- Esempi pratici

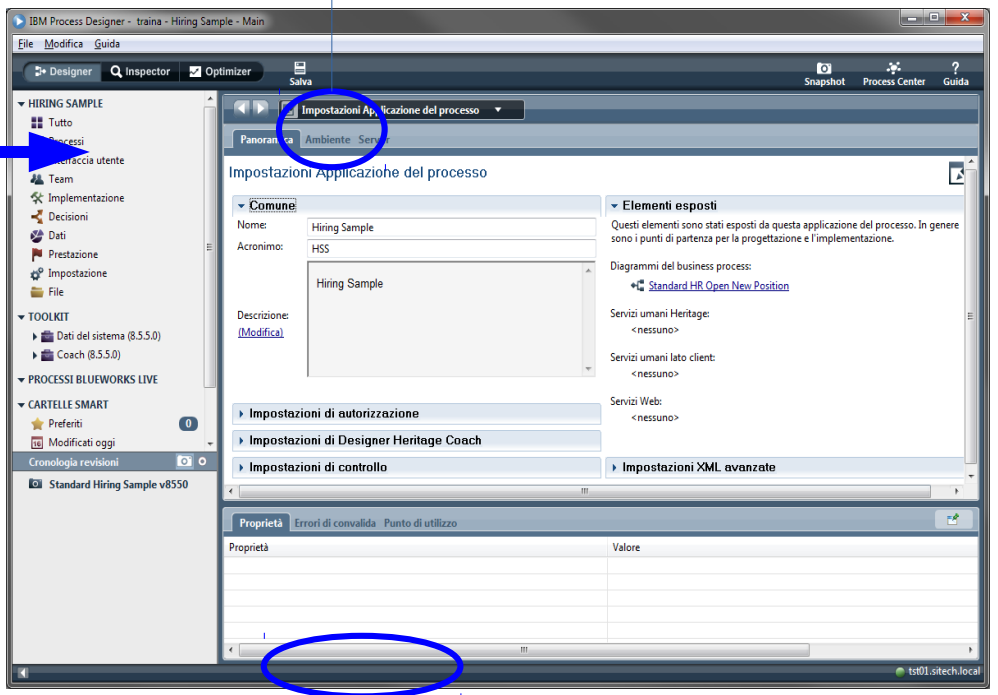
Process Designer

Applicazione desktop (eclipse based) per la gestione di tutti gli artifacts BPM.

Salva tutti gli artifacts su **Process Center**.

- Vista sulla Process Center Console
- Tool di Desing





Var env

Sviluppatori e work sharing

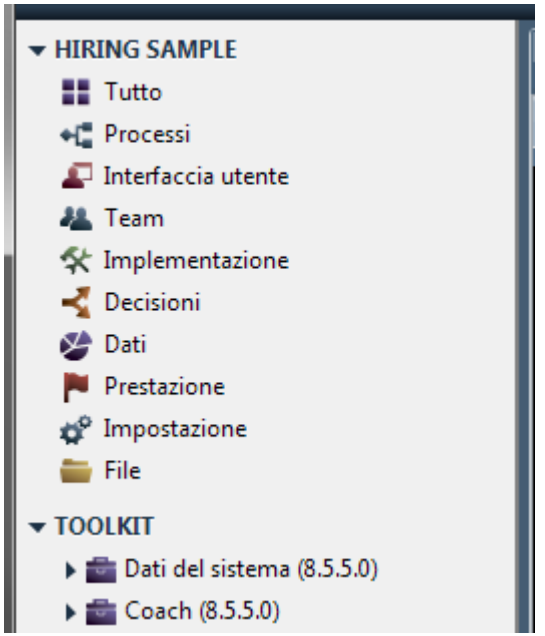
Esploriamo il process designer....

Process Application (Definizione)

Contenitore di tutti gli elementi necessari per l'implementazione di una soluzione. Astrazione di progetto. Contenitore di artifacts sul **Process Center**.

Tipicamente si compone di:

- Uno o più processi → **Business Process Definition (BPD)**
- Implementazioni di Servizi e Attività
- Interfacce grafiche
- Modello Dati
- Integrazioni verso sistemi terzi
- Teams
- Dipendenze → **Toolkits** (vedi in seguito)



Process Application (Struttura)

- Process Application (PA)
  - Business Process Definition (BPD)
    - Activity (uno “step” del processo)

Cosa

Come

- Task → Implementazione di un' Activity (Es. User, System, External)
  - Service (relativo diagramma) (Es. Human, Integration, Sys..)
    - Service Component (nel diagramma del service)

Cosa

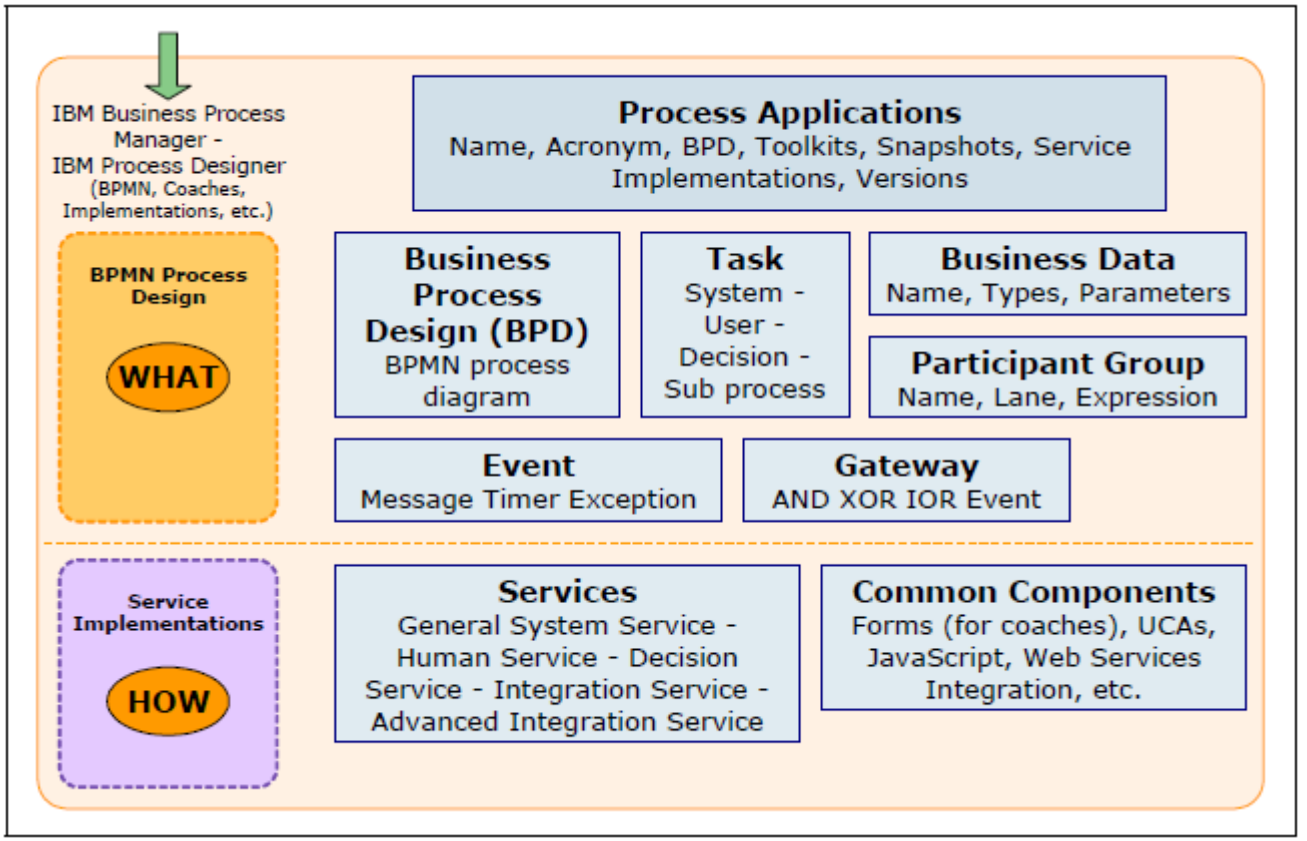
- Sub-Process
- Linked-Process

Come

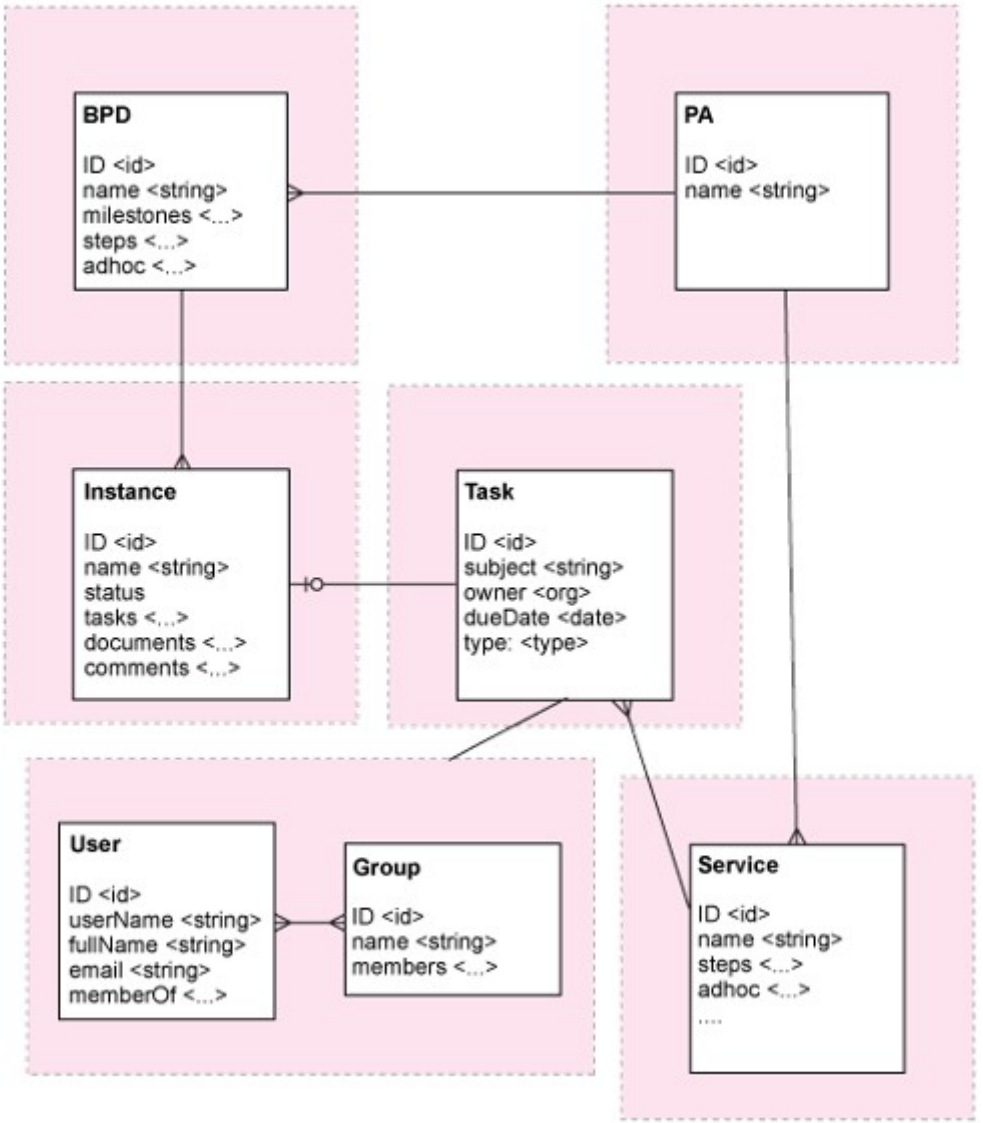
- Script

    - Altri elementi BPD (Es: Gateways, events...)
  - Altri elementi PA (Es. Data model, Teams....)

Process Application (Struttura)

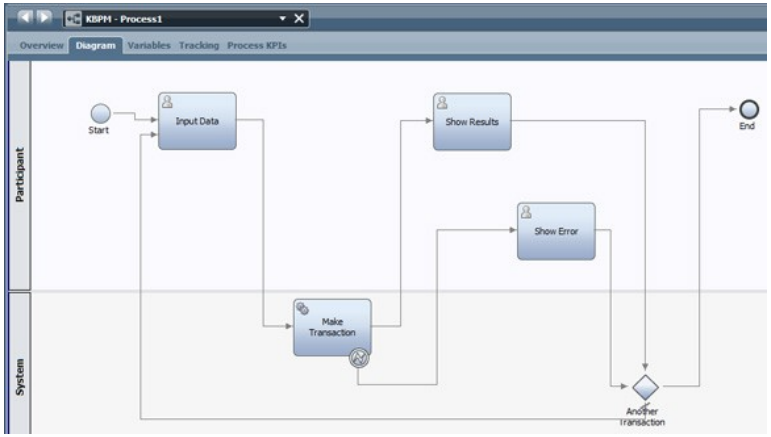


Process Application (Run Time)

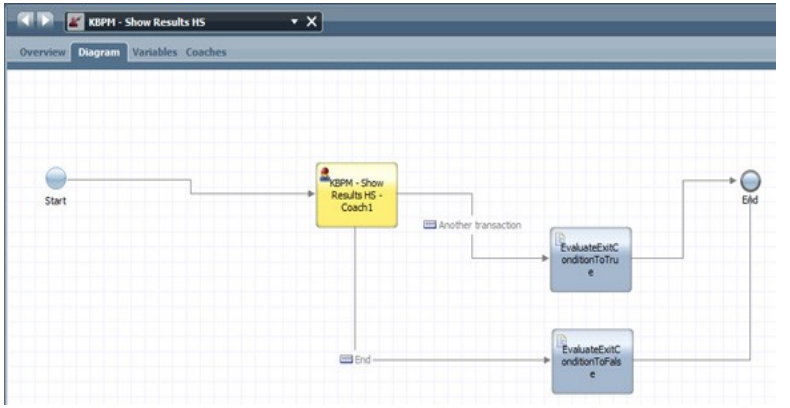


Livelli di Modellazione

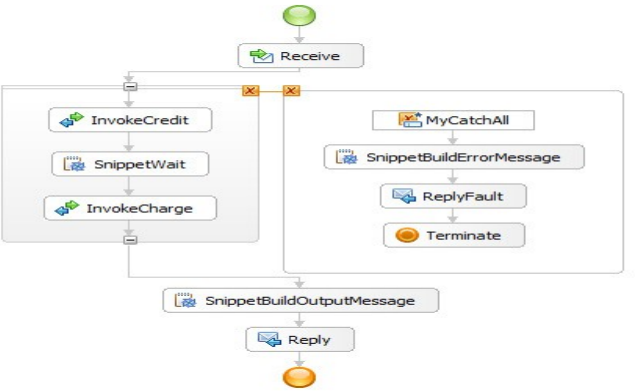
1. **BPMN** → Per Business Process Definition (BPD)
   Process Designer



2. **FLOW** → UML “like” Per implementazione Activity
   Process Designer



3. **BPEL** → Integrazioni WS / SOA (**Solo Advanced**).
   Integration Designer



Business Process Definition vs Process Instance

BPD: E' la definizione del processo, ne rappresenta il flusso logico

Process Instance: Rappresenta l'esecuzione (run time) di una specifica BPD.
E' identificata da un id univoco e può assumere i seguenti stati:

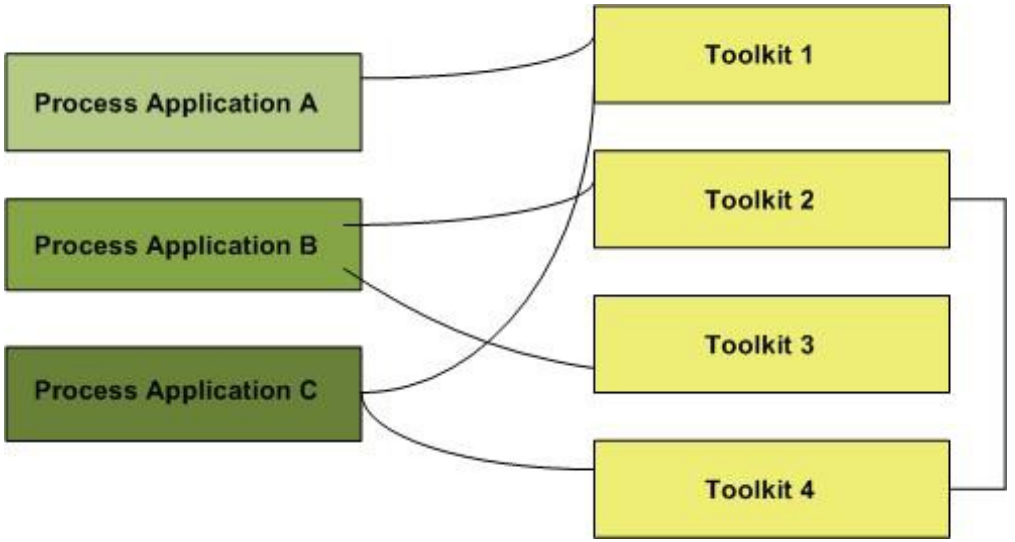
- **Active** → Istanza in esecuzione
- **Completed** → Istanza completata con esito positivo (senza errori)
- **Failed** → Istanza terminata con errori
- **Terminated** → Istanza terminata forzatamente (end event) e cancellazione timer
- **Suspended** → Esecuzione temporaneamente sospesa
- **Did_not_started** → Processo non avviabile (condizione start event non verificata)

Toolkits

I toolkits sono raccolte di asset che possono essere condivise tra più Process Applications.
Un PA dipende dai toolkits di sistema più eventuali toolkits custom.

Rappresentano delle vere e proprie librerie di “elementi” riutilizzabili trasversalmente.

Gli elementi contenuti in un toolikit sono gli stessi della PA ma non possono essere deployati separatamente.
Un toolkit può a sua volta dipendere da altri toolkits.



Strutturare una Business Process Definition

Linee guida su come strutturare una process application (best practice)

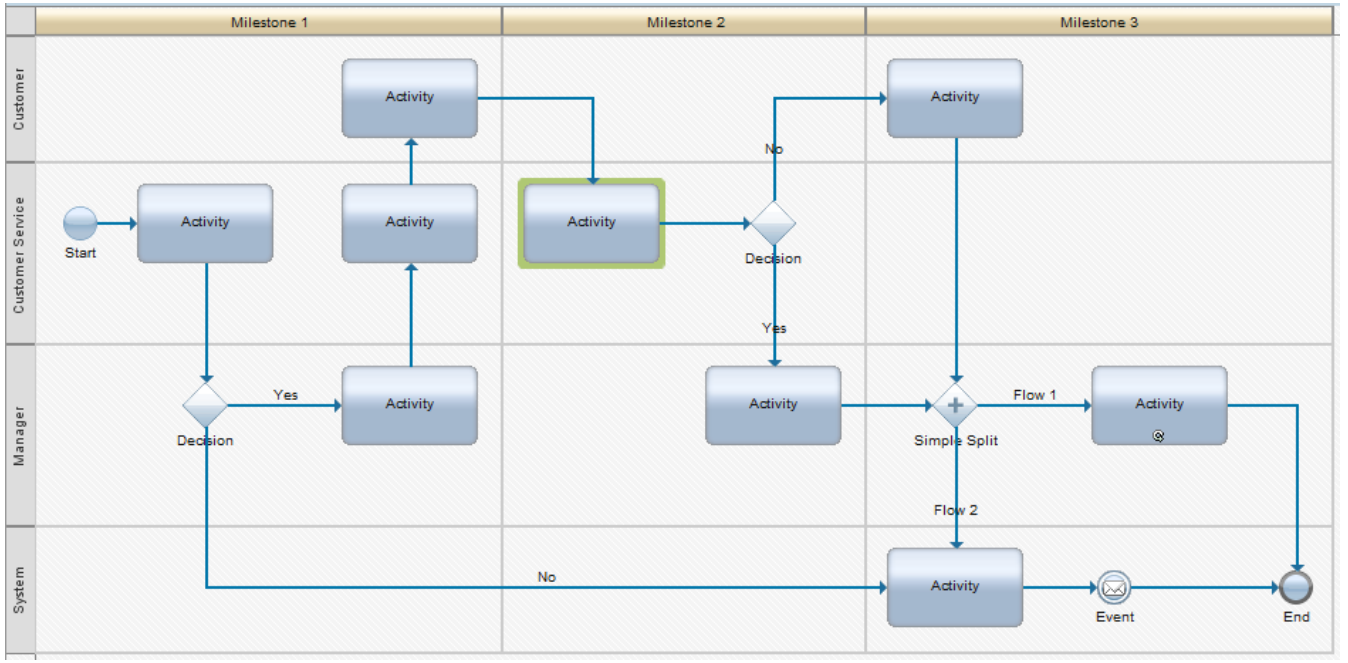
- **Rule of seven**
- **Activity granularity**
- **Activity description**
- **Inputs and outputs**
- **The system lane**

Individuare gli elementi riusabili e pacchettizzarli in toolkits.

Applicare le linee guida anche ai toolkits

Strutturare una Business Process Definition: Rule of Seven

Mantenere il numero di attività (task) del diagramma di processo (BPD) con un livello di granularità di sette (...circa sette!, nella pratica anche 10!)
Al di là di sette (10) distinte attività su un diagramma di processo,
un utente perde la capacità di comprendere lo schema o il suo intento.
Se sono necessarie più di sette (10) attività utuilizzare i sottoprocessi
per raggruppare attività logicamente coerenti.



9 OK!

Strutturare una Business Process Definition: Activity Granularity

Una attività rappresenta uno step di un processo che:

- Ha lo scopo di produrre un risultato in uscita
- Può essere implementato da un task (system o human) o da un sub-process
- Può essere uno human task se soltanto un Team inizia e completa l'attività
- Contiene degli step non identificabili con step di processo
- Può essere un sottoprocesso → applicare le stesse regole del processo padre

Ad ogni livello di astrazione tutte le attività dovrebbero essere simili per scopo e importanza.

Bad Pattern → **Best Practice**

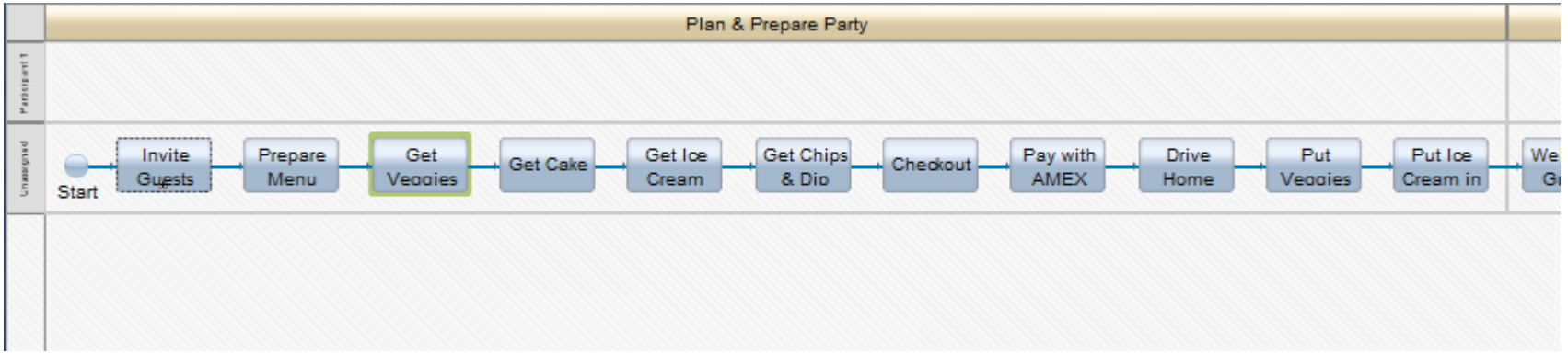
- String of pearls
- Constellation
- Flow lines

Strutturare una Business Process Definition: Activity Granularity

String of Pearls Pattern

Bad Pattern

Consiste in una serie di operazioni sequenziali nella stessa swimlane (stesso team!) che non attinenti a livello di processo ma di servizio.



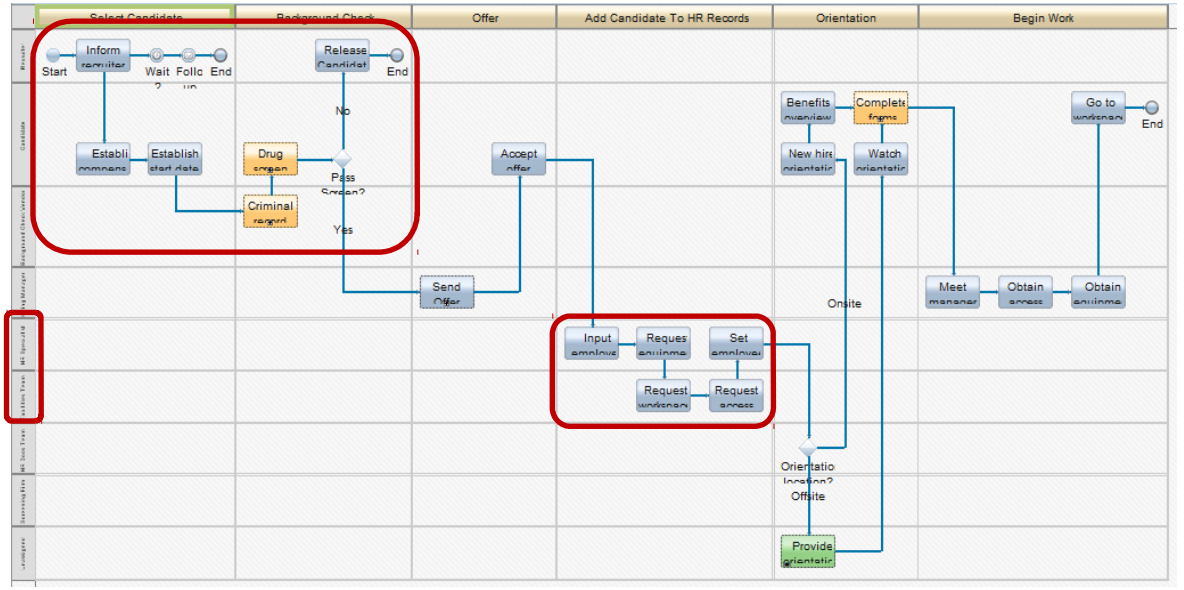
Queste attività vanno raggruppate all'interno di un unica attività

Strutturare una Business Process Definition: Activity Granularity

Constellation Pattern

Bad Pattern

Consiste in una serie di operazioni relative ad un'unica linea di input un'unica linea di output (anche raggruppati in swimlane non più usate nel proseguo del processo)



Queste attività dovrebbero essere svolte all'interno di un sotto processo

Strutturare una Business Process Definition: Activity Granularity

Flow Lines Pattern

Una linea di flusso

- Generalmente non dovrebbe fluire all'indietro in una milestone precedente
- Generalmente non dovrebbe saltare una milestone
- Dovrebbe evitare di tornare a una fase precedente per ripetere un'attività (o sottoprocesso) più avanti nel processo

Se possibile rivedere il flusso di processo e/o generalizzare alcuni asset

Strutturare una Business Process Definition: Activity Description

I nomi delle attività dovrebbero sottostare a una delle seguenti naming-convention

- ✔ **Activity name = action + entity**
- ✔ **Activity name = [action verb] + [business object]**

Scrivere la documentazione dell'attività nell'apposito box in modalità **User Story.**

Vantaggi **User Story**

- Il formato **User Story** rende più facile la comprensione e l'individuazione di importanti dettagli sul processo
- Si presta ad uno sviluppo **Agile**
- Aiuta a concentrarsi su cosa fare (analisi requisiti) astraendosi dai dettagli implementativi

Strutturare una Business Process Definition: Inputs and Outputs

In fase di analisi modellare gli Inputs e gli Outputs

- Come oggetti di Business astratti senza specificare i singoli campi di cui si compongono
- Senza fare riferimento ad un possibile stato dell'entità (es: contratto firmato) ma usare attributi per definirne lo stato (fase di sviluppo)

Strutturare una Business Process Definition: System Lane

- La system lane rappresenta tutte le attività automatiche e le interazioni verso sistemi esterni con cui il BPM deve interagire.
- Non deve mai includere attività manuali (human task).
- Se c'è molta attività tra la system lane e una partecipant lane (user / team) , potrebbe essere perché c'è una granularità di basso livello che non attiene al livello di processo ma di servizio .
- Definire una sola system lane

Strutturare una Business Process Definition

Scomposizione attività di processo e implementazione di servizi

[...The fundamental difference between process models and services is that BPD activities are asynchronous. They are placed on the event manager for later execution or assignment. If you are expecting two units of work to occur immediately, you should use a service. In typical cases, reserve BPDs for task assignment and high-level process logic. The table provides high-level guidelines about when to use a business process definition (BPD) and when to use a service in IBM Business Process Manager ….]

Situation	BPD	Either/ both	Service
Display UI to a user.			X
Integrate to an external system (DB, rules engine, web service, and so on).			X
Background processing, scheduled processing.			X
Series of UIs directed at the same user.		X	X
Simple Javascript.		X	
Event correlation.	X		
Exception handling.		X	
Task handed from one user to another.	X		
Split to parallel paths.	X		

Process Application e BPD: Esercizi

Iniziamo ad utilizzare il **Process Designer...**

Obiettivi

- Creare semplici BPD di test
- Gestione snapshot e deploy
- Prendere familiarità con i vari componenti del designer
- Prendere familiarità con il linguaggio di scripting
- Definire utenti / gruppi / team e swimlane
- Interazione con process inspector, process portal e API Rest
- Monitoraggio da process admin console
- Utilizzo variabili di ambiente

Unità 3: Componenti BPM

- Flusso di processo (BPD): Process, Sub-Process, Linked-Process, Script, Gateways, Multi Instanze Process
- Variabili di processo e relativi scope
- Activity: Task, Services ed External
- System Task e Human Task
- Data Object
- Realizzazione di interfacce grafiche, coach e componenti nativi
- Gestione degli eventi: I vari tipi di eventi e il loro utilizzo
- Gestione degli errori
- Esempi pratici

Process Application (Struttura)

Riprendiamo dalla struttura PA per passare ai componenti BPM

- Process Application (PA)
  - Business Process Definition (BPD)
    - Activity (uno “step” del processo)

Cosa

Come

- Task → Implementazione di un' Activity (Es. User, System, External)
  - Service (relativo diagramma) (Es. Human, Integration, Sys..)
    - Service Component (nel diagramma del service)
- Sub-Process
- Linked-Process

Cosa

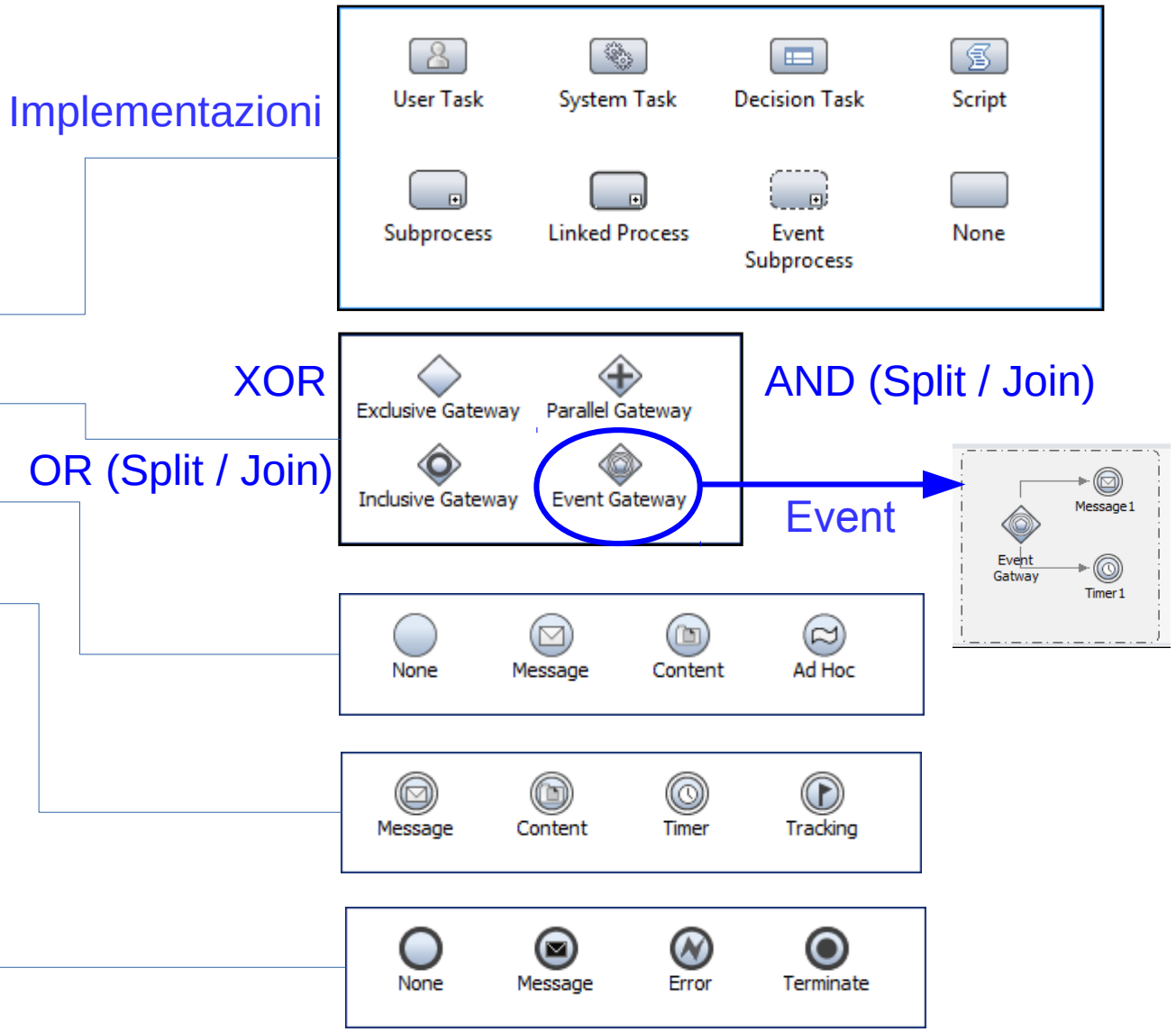
Come

- Script

    - Altri elementi BPD (Es: Gateways, events...)
  - Altri elementi PA (Es. Data model, Teams....)

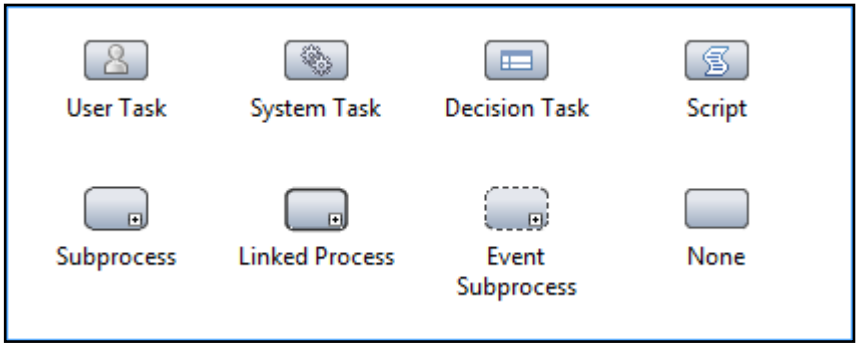
Elementi di una BPD

- Activity (Task)
- Gateway
- Start Event
- Intermediate Event
- End Event
- Note



Activity

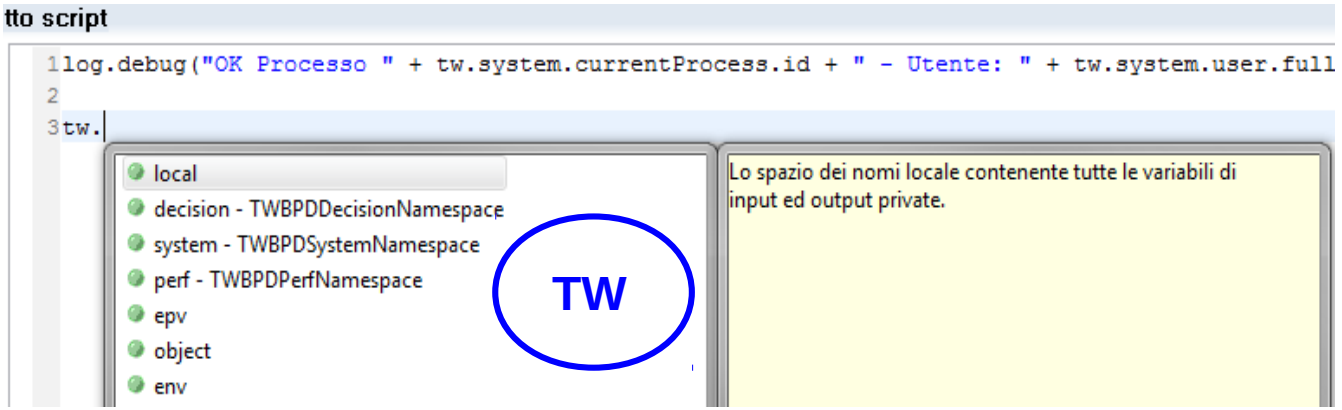
Una Activity (Task) rappresenta uno step all'interno di una BPD.
Può avere le seguenti implementazioni

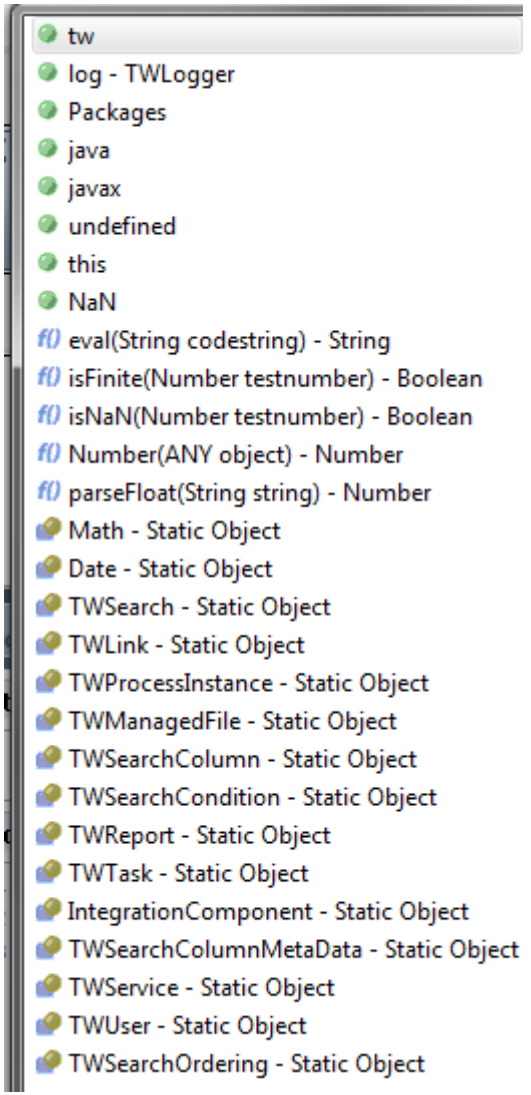


Activity Implementation:	Default service type	Possible service types
User Task	Human Service	External Implementation, Human Service, Integration Service
System Task	Integration Service	Ajax Service, External Implementation, General System Service, Integration Service
Decision Task	Decision Service	Decision Service
Script		
Subprocess		
Linked Process	BPD	BPD
Event Subprocess		
None		

Script

- A livello di BPD / Activity / Service
- Linguaggio javascript 'like' server side
- Evitare l'uso di scripting per logica di business
- Usare per mappature dati, inizializzaizoni e logiche semplici relative al flusso di processo
- Accesso ai dati dell'istanza, dell'utente, dati di sistema e variabili di ambiente.
- Sono richiamabili API di sistema (Es. avviare una istanza, lanciare un evento, etc...)





Esploriamo un po' gli oggetti disponibili mediante scripting...

Sub Process

Sottoprocesso ("embedded sub process" in notazione BPMN)

Il sottoprocesso rappresenta l'incapsulamento degli step logicamente correlati all'interno di un processo principale. Gli step di un sottoprocesso possono accedere direttamente gli oggetti di business (variabili) dal processo padre.

Non è richiesta alcuna mappatura dei dati.

A differenza di un processo collegato (Linked Process), un sottoprocesso può essere istanziata solo dal processo padre e non è riutilizzabile mediante da altri processi o sottoprocesso.

È possibile visualizzare un sottoprocesso come una singola attività, fornendo una vista semplificata di alto livello del processo padre, oppure si può essere espanso per una visualizzazione più dettagliata del suo contenuto.

Un sottoprocesso può essere incorporato all'interno di un altro sottoprocesso.

Linked Process

Processo Collegato ("resusable sub process" in BPMN notazione)

I processi collegati incapsulano step logicamente correlati all'interno di un processo, pur mantenendo la visione di alto livello del processo padre. I processi collegati differiscono da sottoprocessi perché possono essere riutilizzabili da più processi padre.

Nelle versioni precedenti erano noti come *processi nidificati*.

Event Sub Process

Gli **Event Sub Process** sono sottoprocessi innescati da un evento che si verifica nel processo padre. Sono simili ad altri sottoprocessi nel senso che sono contenuti e utilizzabili **solo** all'interno al processo padre (no linked process).

Non sono collegati ad altre activity del processo (no connection in/out) → attivati dal catching di un evento (start event) o da un timer.

Ha accesso agli oggetti di business (variabili) del processo padre ed è quindi in grado di incapsulare step che utilizzano tali variabili .

Un event sub process può o interrompere l'esecuzione del processo padre e può avere una esecuzione parallela.

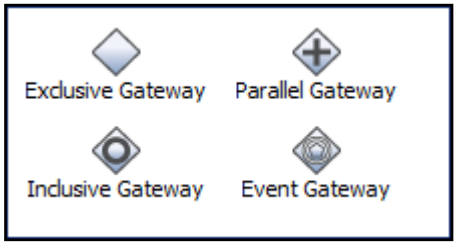
Avvio mediante **Error Event** → gestione errori nel processo padre

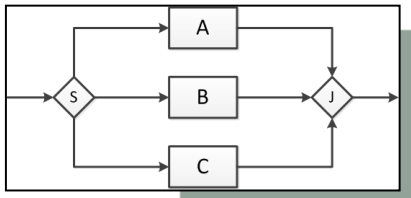
Avvio mediante **Message Event** → da altro task del processo padre

Avvio mediante **Timer** → Attività da ritardare rispetto al processo padre oppure da eseguire solo allo scadere di un timeout

Gateway (Tipologie)

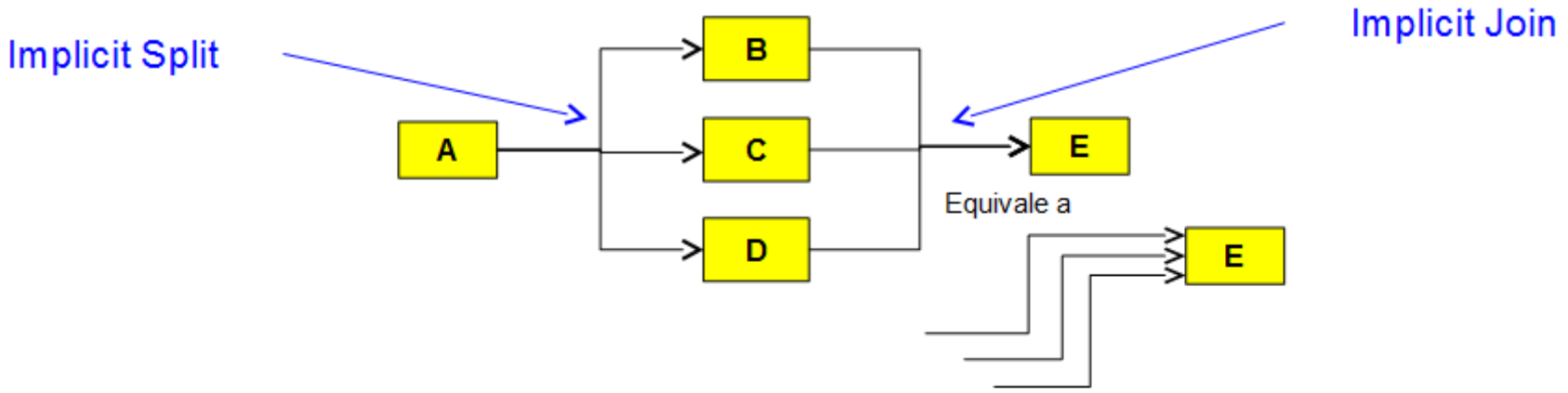
Permettono lo **spilt & join** tra due o più branches di una **BPD (No Service!!!)**



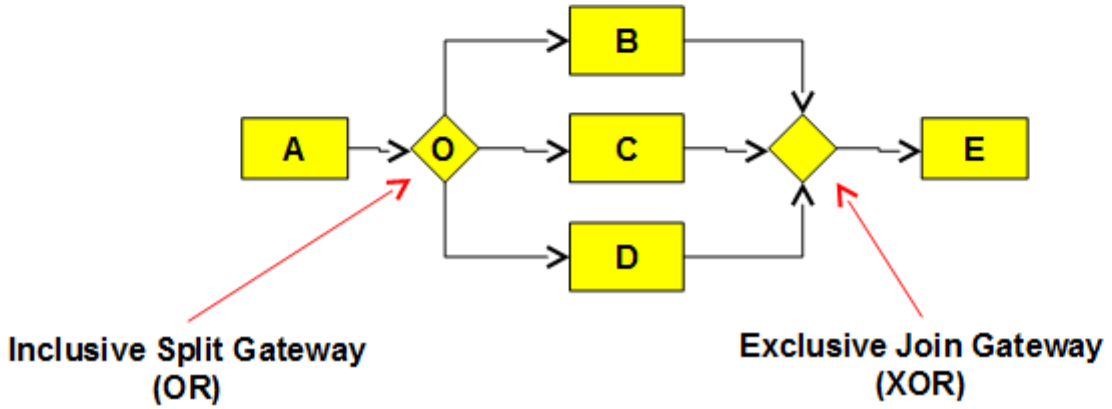


Split: \ Join:	Exclusive Gateway (XOR)	Parallel Gateway (AND)	Inclusive Gateway (OR)
Exclusive Gateway (XOR)	• "No synchronization" • 1 token in • Execute A xor B xor C • 1 token out	• "Synchronization" • **Invalid Diagram** • 1 token in • Execute A xor B xor C • 0 token out	• "Wait for all possible active paths" • 1 token in • Execute A xor B xor C • 1 token out
Parallel Gateway (AND)	• "No synchronization" • 1 token in • Execute A and B and C • **3 token out**	• "Synchronization" • 1 token in • Execute A and B and C • 1 token out	• "Wait for all possible active paths" • 1 token in • Execute A and B and C • 1 token out
Inclusive Gateway (OR) (see also "Note" below)	• "No synchronization" • 1 token in • Execute A or B or C • 1 or 2 token out	• "Synchronization" • **Invalid Diagram** • 1 token in • Execute A or B or C • 0 token out	• "Wait for all possible active paths" • 1 token in • Execute A or B or C • 1 token out

Gateway (Join & Split Implicito)



Semanticamente uguale a:



Gateway (Regole)

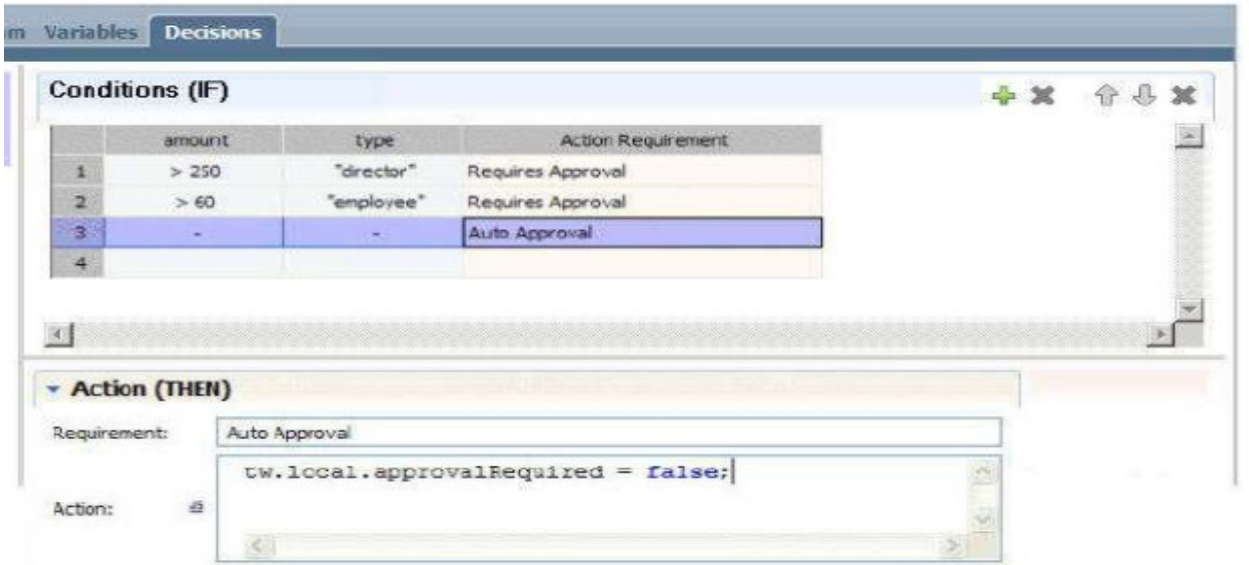
Vari modi di impostare le regole sui gateway

- Regole (espressioni) definite sui rami (Implementation=>Decisions)

  dei decision gateway OR e XOR

- Regole definite in ambito Decision Service (DS), tramite componenti Decision Table:
  - DS usati per implementare delle activity con un implementation type di tipo Decision Task (pattern "*Implementing an activity using a Decision service*")
  - DS usati nei gateway OR e XOR (pattern "*Attaching a Decision service to a decision gateway*") [in questo caso gli output del DS sono accessibili nel namespace tw.decision]
  - DS usati in altri servizi

Esempio
Decision Table



Start Events

None: Il *None Start Event* (o semplicemente "Start Event ") è utilizzato per indicare il punto di partenza di una istanza di un processo . Per impostazione predefinita, quando si crea una nuova BPD, viene automaticamente aggiunto al diagramma.

Message: Lo *Start Event Message* è un nodo di partenza che può essere utilizzato per avviare una istanza di un processo a seguito della notifica di un evento (Es: avvio di un processo a fronte di un evento esterno) . L'associazione tra il processo e l'evento esterno avviene mediante il componente **UCA (Under Covered Agent)**. Quando un messaggio viene ricevuto da uno start event message viene creata una nuova istanza di processo avente un nuovo process instance id.

Content: Evento ECM per dare il via un processo

Ad Hoc: Permette l'avvio di una sequenza di attività in una istanza di processo già avviata.

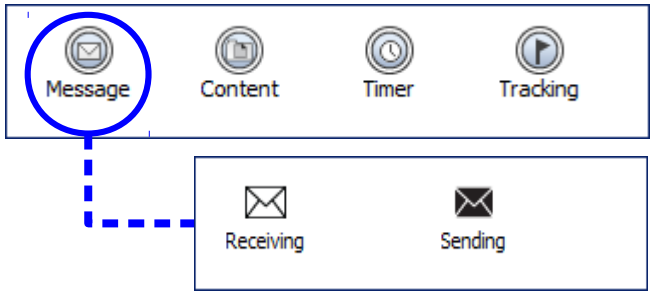
Esempio

```
var ProcInstanceID = tw.system.currentProcessInstance.id;

var AdhocEventName = "AdHoc1";
var procI = tw.system.findProcessInstanceByID(ProcInstanceID);
procI.findAdhocStartingPointByName(AdhocEventName).startNew();
```

Intermediate / Boundary Events

Intermediate → Flusso BPD



Boundary → Attach Activity

Message: Usato per inviare o ricevere notifiche (Tipologia Receving / Sending). Può essere usato sia nel flusso di processo come elemento singolo oppure collegato a un'activity (in questo caso può solo ricevere).

Timer: Usato per modellare percorsi di escalation o ritardi nelle BPDS. E' possibile specificare un intervallo di tempo dopo o prima di effettuare un'attività. Può essere usato sia nel flusso di processo come elemento singolo oppure collegato a un'activity.

Tracking: Usato per il monitoraggio. Indica il punto di un processo in cui si desidera che il BPM per catturi i dati di runtime per scopi di reporting . Si può usare solo nel flusso di processo.

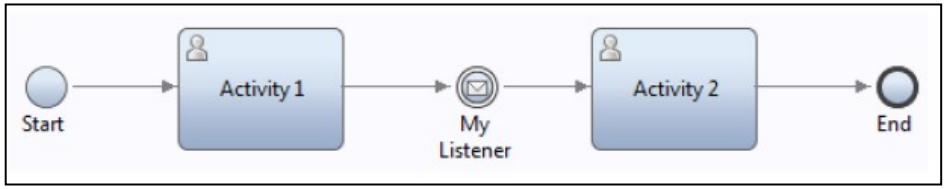
Error: Usato per intercettare (catch) gli errori nel flusso di processo . Può essere usato sia nel flusso di processo come elemento singolo oppure collegato a un'activity

Content: Usato nella gestione dei contenuti per modellare la ricezione di un evento ECM. Può essere usato sia nel flusso di processo come elemento singolo oppure collegato a un'activity

Intermediate Message Event (Dettagli)

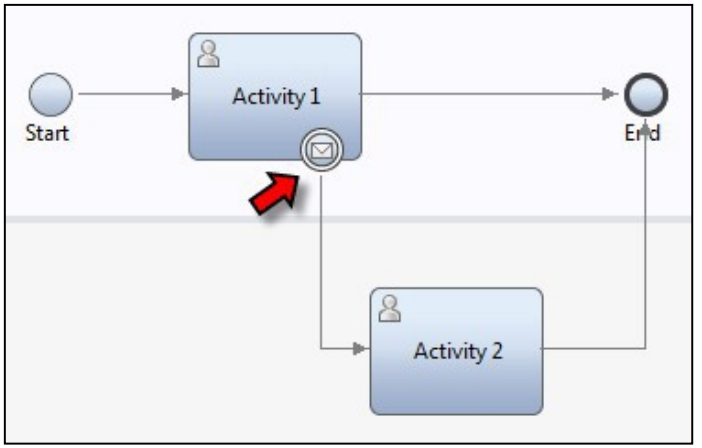
Esempio Intermediate Message Event in modalità receiving

Intermediate standalone



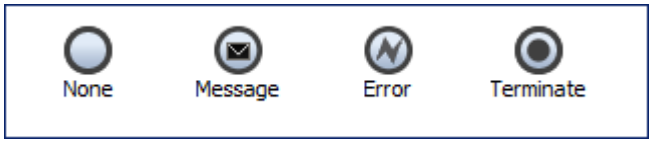
Sospende l'esecuzione del processo finché non riceve un evento o il processo viene terminato forzatamente

Attached to Activity (sempre receiving)



Mentre Activity 1 è in esecuzione se arriva l'evento si attiva parallelamente l'Activity 2.
Impostazione per interrompere Activity 1.

End Events



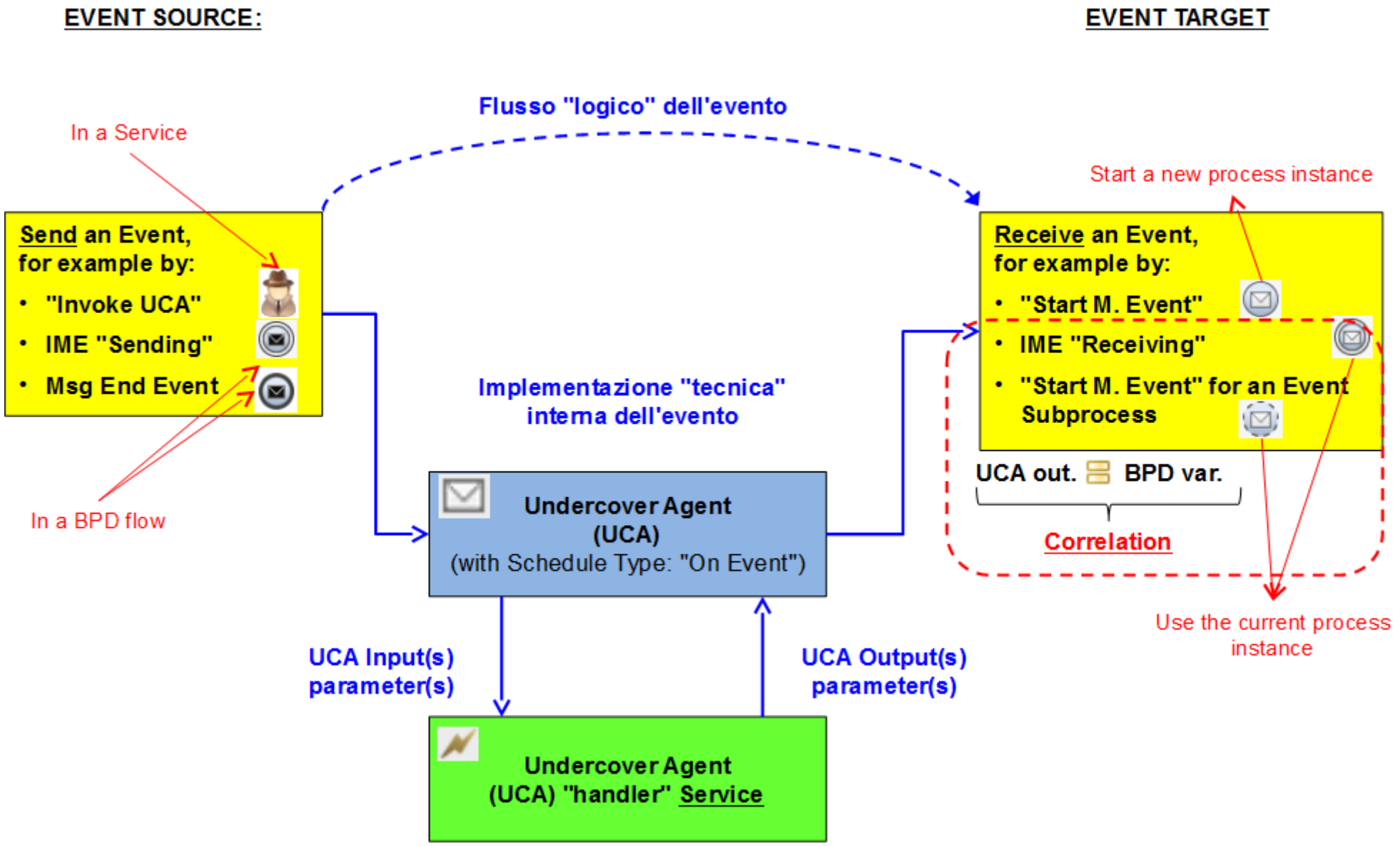
None: Usato per terminare una attività, un processo o un singolo ramo di un processo.

Error: Usato per lanciare un errore (exception) al processo / suottoprocesso / task padre. Vedi più avanti in gestione errore.

Terminate: Usato per terminare una attività in esecuzione associata a un processo annullando gli eventuali timer in sospeso. È possibile indicare che si desidera terminare l'intera istanza di processo, oppure è possibile indicare se si desidera terminare solo il processo che contiene l'evento ei suoi sottoprocessi . Se un'intera istanza di processo viene terminato, il processo mostra lo stato di Terminated.

Message: Usato per inviare un messaggio . Ad esempio, si potrebbe voler inviare un messaggio a conclusione di una istanza di processo per permettere l'avvio di un altro processo (start message event) [processi correlati]. Vedi più avanti uso dei messaggi.

Eventi ed Undercover Agents (UCA)



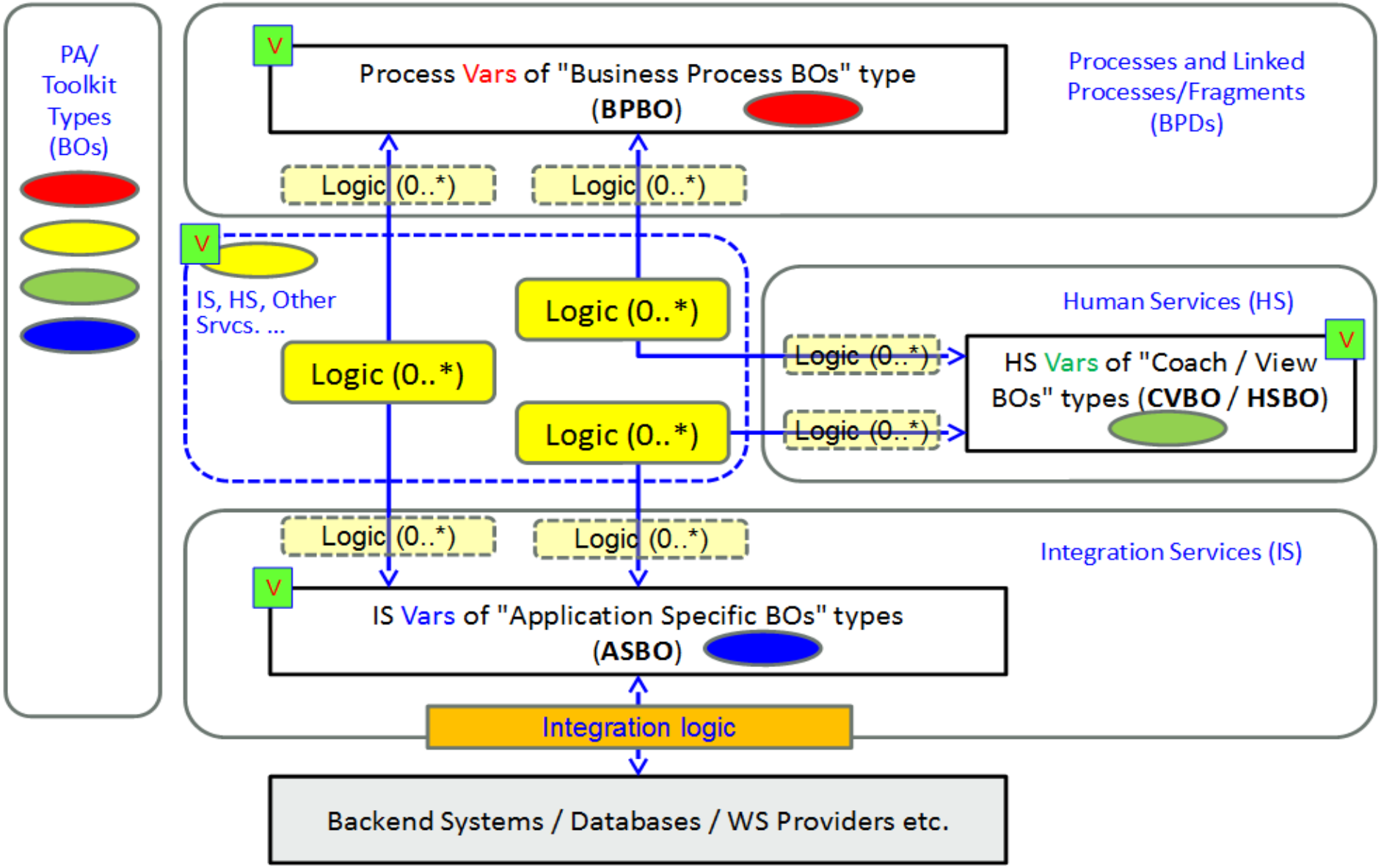
Activity Service Components

Componenti utilizzabili all'interno delle implementazioni (service) di una activity

	Integration Service	Human Service	Rule Service	Other Service Types
Coach		•		
Heritage Coach		•		
Server Script	•	•	•	•
Server Scriptlet	•	•	•	•
Modify Task		•		
Postpone Task		•		
Content Integration	•	•		
Decision Gateway	•	•	•	•
End Event	•	•	•	•
Note	•	•	•	•
Error End Event	•	•	•	•
Invoke UCA	•	•	•	•
Error Intermediate Event	•	•	•	•
Intermediate Tacking Event	•	•	•	•
Stay On Page Event		•		
Nested Service	•	•	•	•
Send Alert	•	•	•	•
Web Service Integration	•			

Business Object (Data Type)

Variables



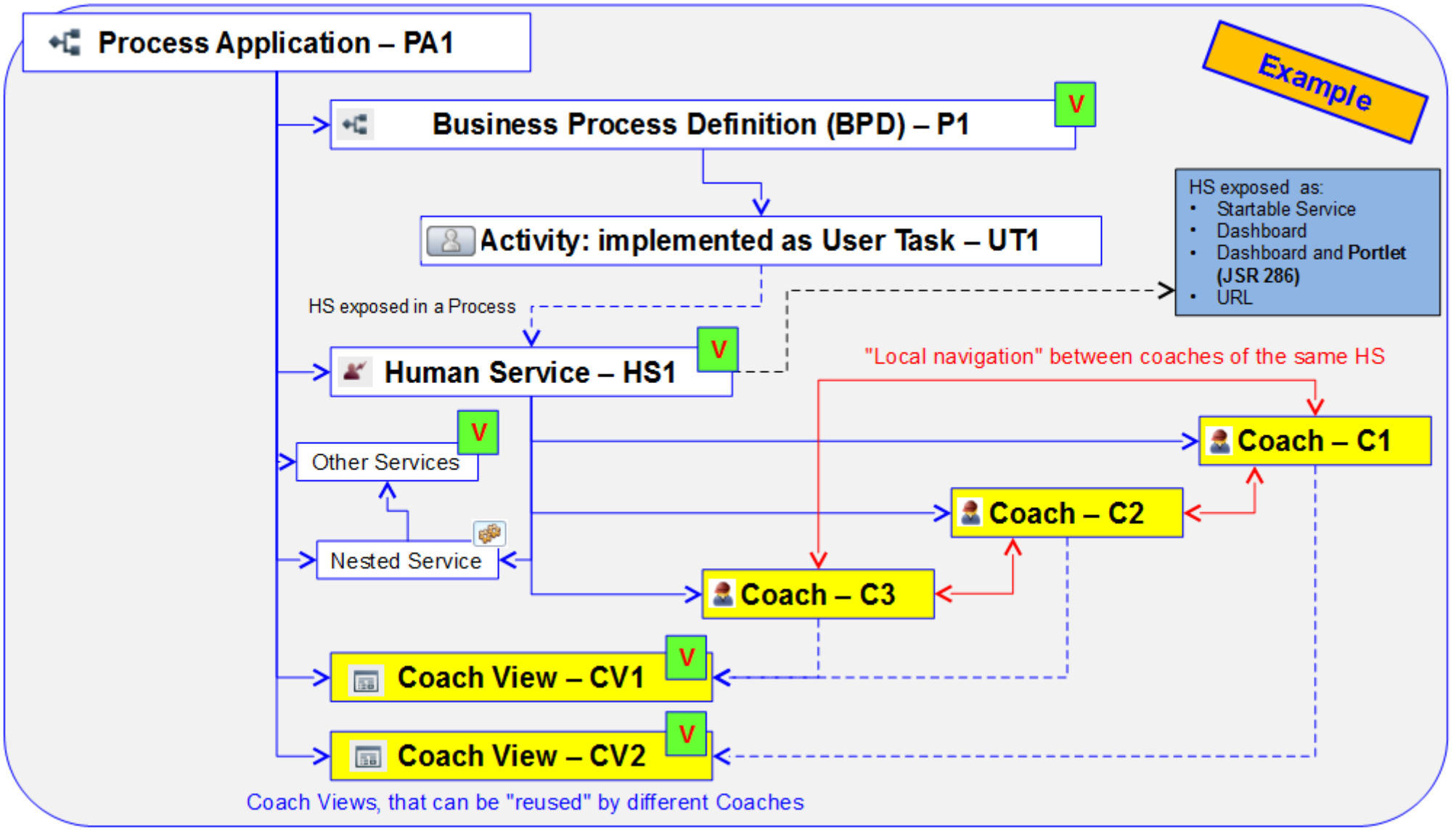
Business Object (Data Type)

Passaggio di variabili tra BPD e Servizi

How variables are passed			
From	**To**	**Type**	**Pass by**
Business process definition (BPD) activity	Service	Simple	Value
BPD activity	Service	Complex	Value (the BPD and Service have separate copies of the business object)
BPD activity	Linked BPD	Simple	Value
BPD activity	Linked BPD	Complex	Reference to the same business object
Service	Nested service	Simple	Value
Service	Nested service	Complex	Reference to the same business object

Nel caso di tipi semplici e di tipi complessi (da BPD a Service) dichiarare la stessa Variabile sia in input e output se il chiamato può modificarne il valore

Interfacce Grafiche: Coach e Coach View



Vedremo casi pratici...

Gestione Errori

Quando si progetta una BPD o un servizio, occorre pensare fin da subito alla logica per gestire eventuali errori che potrebbero essere generati a runtime.

Ci sono tre scenari:

- End Error Event in BPDS e servizi (terminazione con errore)
- Intermediate Error Event in BPDS e servizi che catturano gli errori (try/catch)
- Start Error Event che avviano sottoprocessi in BPD che catturano gli errori.

È possibile assegnare i codici di errore e i dati di dettaglio dell'errore

Gli errori sono catturati nel seguente ordine:

1. Boundary event error intercetta gli errori che vengono generati dall'attività collegata
2. Se non c'è alcun boundary event che gestisce l'errore, e c'è un sottoprocesso con error event, gli errori sono catturati da tale processo
3. Se non ci sono error event sub process, gli errori sono propagati al livello successivo.

Gestione Errori

Regole Error Catching

Specify	Errors caught
error code and error data	Only errors with the same code and data type
error code	Errors with the same code, unless they are already caught by an event, as specified by the preceding rule
error data	Errors with the same data type, unless they are already caught by an event, as specified by the preceding rules
neither code nor error data or the **Catch All Errors** option on error properties	All errors that are not already caught by an event, as specified by the preceding rules

Gestione Errori

In Business Process Definition

BPD event	Description
error intermediate event at the boundary of an activity	• Catches specified errors or all errors • Provides error handling logic for errors raised by the activity that it is attached to
error event subprocess that starts with an error start event	• Catches specified or all errors • Provides error handling logic for errors raised by activities of the BPD, subprocess, or event subprocess that directly contains the error event subprocess
error end event	Throws an error

In Services

Service event	Description
error intermediate event attached to the boundary of a step	Catches errors from the step
error intermediate event as part of the service flow	Catches all errors raised by steps of the service flow that are not handled by an error intermediate event at the boundary of a step. This event can have only outbound links.
error end event	Throws an error and ends the processing of this service. You might, for example, use an error end event when you want a particular result from a Coach to end a human service.

Gestione Errori

Modelli di gestione

	User Task / Human Service (Scenario Interattivo)	System Task / not Human Service (Scenario di sistema)
In una BPD: `(tw.system.step.error)`	Gestione generalizzata, via BPD "proxy", verso una mappa di errore (User Task e HS con Coach) nel flusso	Gestione generalizzata via BPD "proxy" con lancio evento verso una BPD di gestione errore, che crea uno user task per l' Admin
	Gestione "ad-hoc" (custom): • Gestione con uno User Task/HS/Coach • Semplice Rethrow errore	Gestione generalizzata tramite linked process con BPD di gestione errore. che crea uno user task per l'Admin
In un Service: `(tw.system.error)`	Gestione generalizzata, via Service "proxy", verso uno HS con Coach nel flusso	Gestione generalizzata via Service "proxy" con lancio evento verso una BPD di gestione errore, che crea uno user task per l' Admin
	Gestione "ad-hoc" (custom): • Gestione con uno HS/Coach • Semplice Rethrow dell'errore	Gestione generalizzata via Service "proxy" con chiamata a tw.system.startProcessByName, che lancia una BPD di gestione errore, e tale BPD crea uno user task per l' Admin.
		Semplice Rethrow dell'errore

Notifica Utente (modello interattivo)

Genera Email/Log/User Task solo per Admin

Gestione Errori

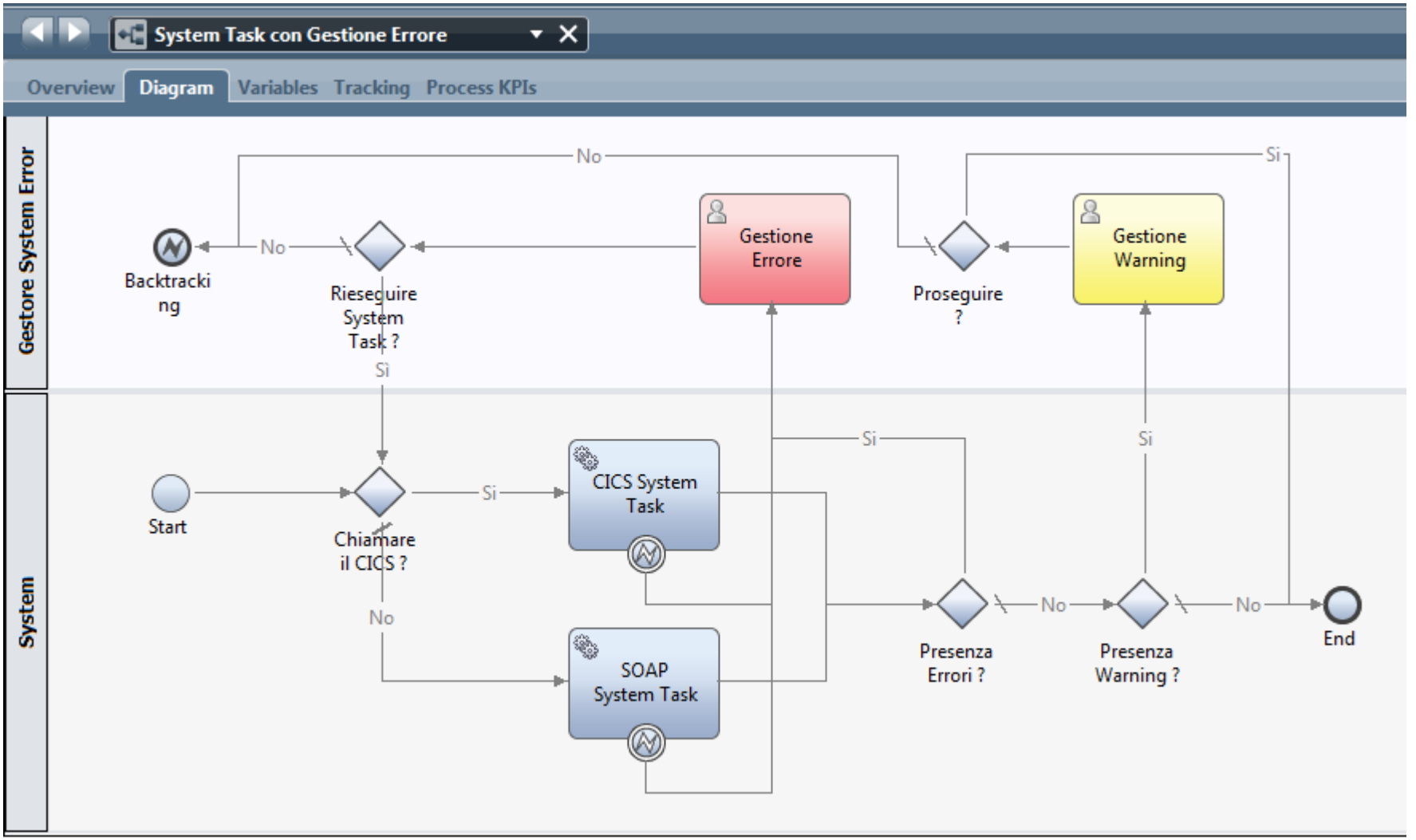
Le casistiche tipicamente gestibili in un coach di "error/exception notification & handlings" (sia a livello di utente interattivo che di notifica all' Admin per errori in un system task) sono le seguenti:

- **Riprova/Ritenta**
- **Salta**
- **Annulla**: rimanda il task al team (togliendolo all'utente corrente)
- **Sospendi**: "postpone" del task

In particolare le opzioni "Riprova/Ritenta" e "Salta" vanno gestite dal programmatore nell'ambito della BPD o Servizio che "lancia" il proxy di gestione dell'eccezione, in modo tale che le eventuali azioni di "Riprova/Ritenta" e/o "Salta" siano collegate in modo opportuno, coerentemente allo "use case" in corso di implementazione.

Gestione Errori

Esempio con notifica utente



Esercizi

Obiettivi

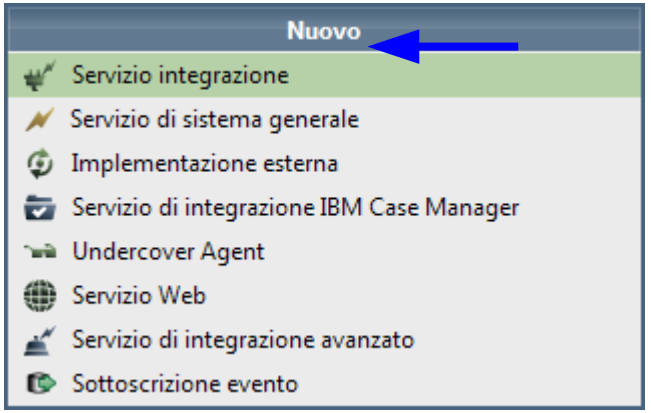
- Creare BPD più complesse
- Implementazione Human Task e semplici interfacce grafice
- Interazione con GUI dal process portal
- Sperimentare le metodologie di gestione errore

Unità 4: Integrazione Servizi

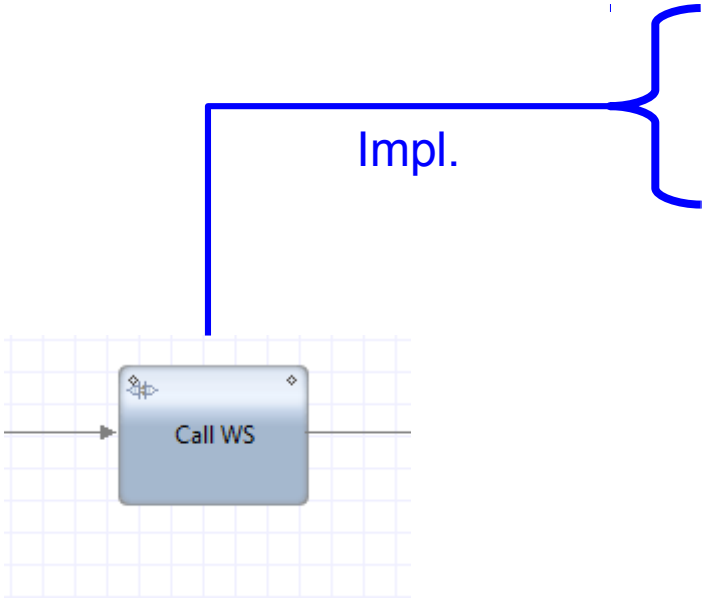
- Gli integration service;
- Integrazione Web Service;
- Integrazione servizi Java generici.
- Integrazione database (GSS)
- Esempi pratici.

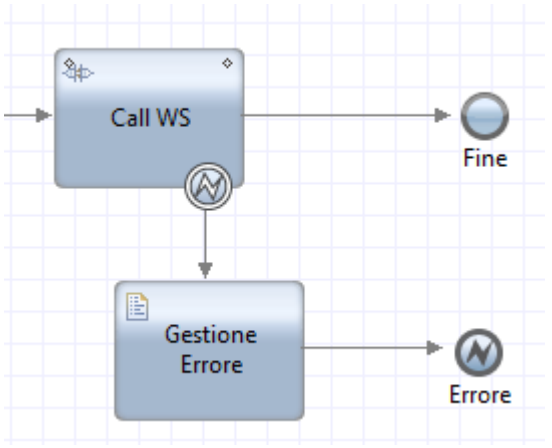
Integration Service

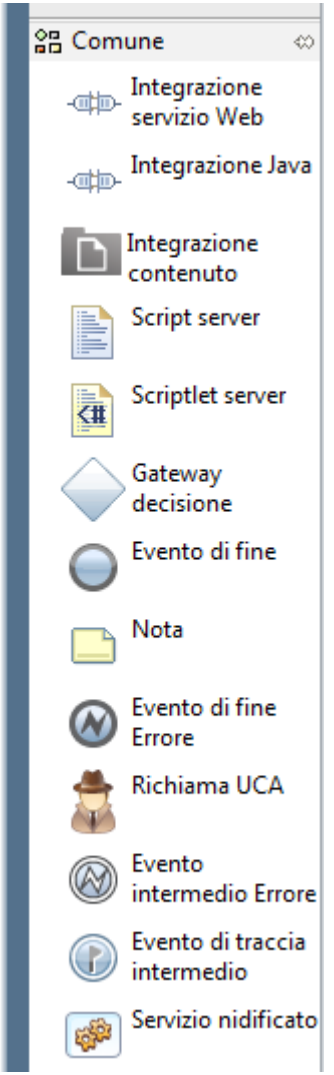
- Tipicamente raccolti in un toolkit
- Semplici componenti riutilizzabili
- Chiamati all'interno di GSS



- Intercettano l'errore nativo, lo gestiscono e lo convertono in un error model applicativo che viene poi lanciato al GSS chiamante

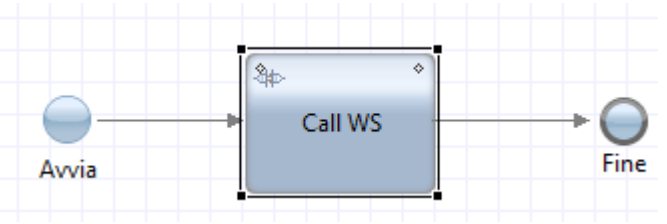




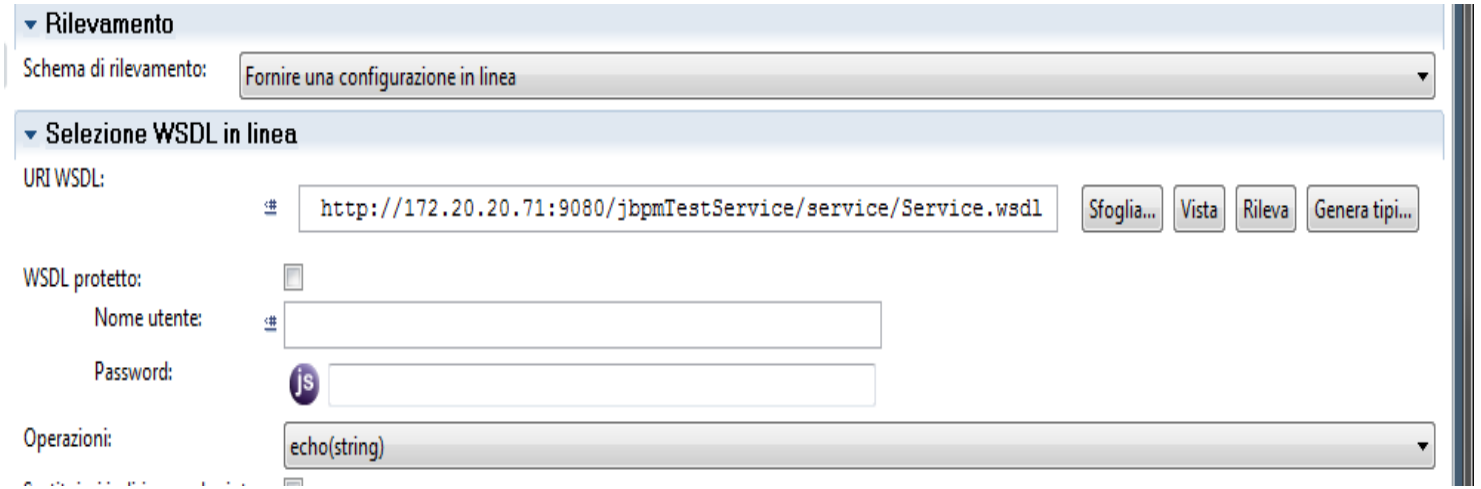


Integration Service

Web Service Integration



Esempio pratico http://172.20.20.71:9080/jbpmTestService/service/Service.wsdl



Tipicamente usata per servizi semplici e/o consolidati sviluppati internamente nell'almbito applicativo.

- Configurare *host:port* in var env
- Discovery
- Generazione input/output
- Eventuale soap header

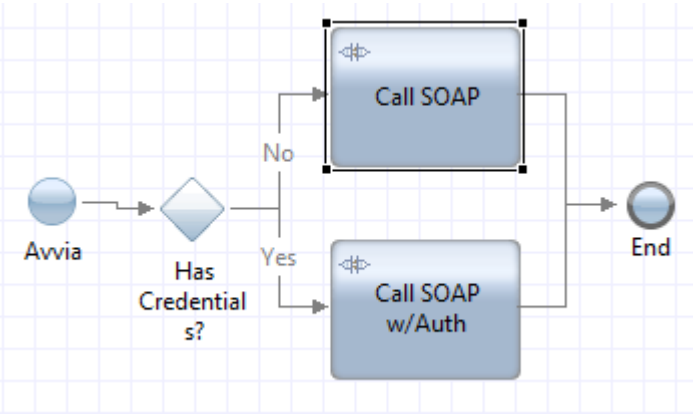
Vantaggi: Integrazione semplice

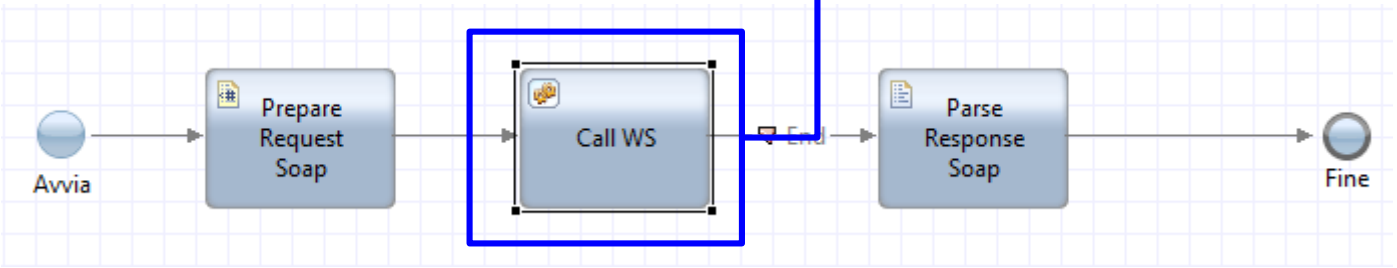
Svantaggi: Rigenerazione in/out e Gestione versionamento

Integration Service

Web Service Integration SOAP

- Disponibile in toolkit di sistema
- Da integrare in GSS
- Gestione request/response SOAP





Tipicamente usata per servizi complessi e per integrazioni con WS di terze parti esterni all'ambito applicativo.

Vantaggi: No generazione input/output

Svantaggi: Menu intuitivo → gestione SOAP

Integration Service

Web Service Integration SOAP (Esempio Pratico)

```
"http://172.20.20.71:9080/jbpmTestService/service/Service
"http://www.jbpm.test/service/"
"service"
"http://172.20.20.71:9080/jbpmTestService/service"
tw.local.soapInput
"echo"
```

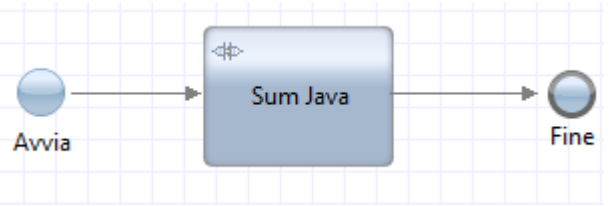
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://www.jbpm.test/service/">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:echo>
         <in>Mario Rossi</in>
      </ser:echo>
   </soapenv:Body>
</soapenv:Envelope>
```

Scriptlet binding Input XMLElement

Script gestione output XMLElement (recupero dati Xpath)

```
if(tw.local.soapOutput != null) {
   tw.local.echoOut = tw.local.soapOutput.xpath("//out")[0].getText();
}
```

Integration Service

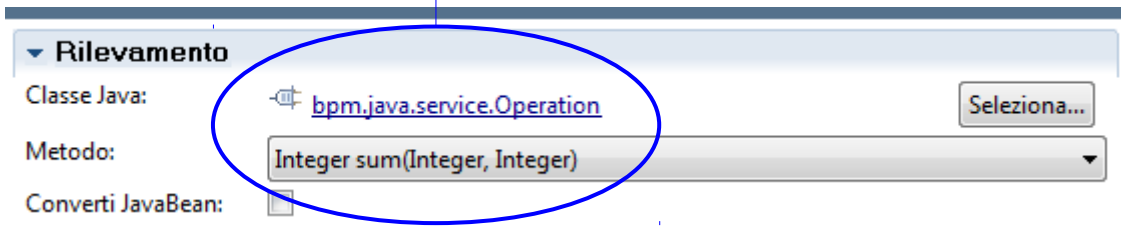


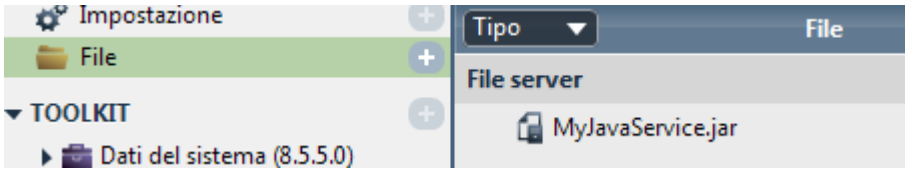
Java Integration

- Consente la chiamata ad un metodo di una classe java
- Realizzazione JAR (complilazione livello 1.6)
- Caricamento JAR sul server
- Logica Java anche molto complessa → Sviluppo esterno (RAD)

Esempio pratico

```
public class Operation
{
      public Integer sum(Integer add1, Integer add2) {
            return new Integer(add1.intValue() + add2.intValue());
      }
}
```





Realizzazione di Integration Service Custom

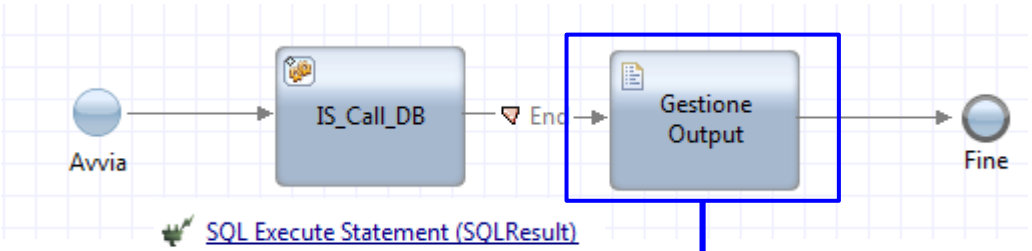
Mediante java integration

(Es. chiamate CICS / integrazione altri sosttosistemi)

Integration Service

Integrazione Database

- Disponibili una serie di Integration Service nel toolkit di sistema
- Utilizzabili all'interno di GSS
- NO Transaction



Esempi pratico

```
Select current date as TS from sysibm.sysdummy1
```

```
Select NAME,TYPE from sysibm.systables
```

```
if(tw.local.result != null &&
   tw.local.result.listLength !=0) {

        tw.local.tableList = new tw.object.listOf.NameValuePair();
        for(var i = 0; i < tw.local.result[0].rows.listLength; ++i)
        {
            tw.local.tableList[i] = new tw.object.NameValuePair();
            tw.local.tableList[i].name = tw.local.result[0].rows[i].NAME;
            tw.local.tableList[i].value = tw.local.result[0].rows[i].TYPE;
        }
   }
```

Esercitazioni Pratiche

Realizzare BPD con

- System task realativi a integration service (DB, Web Service, Java)
- Gestione Input e Output mediante Human Task
- Simulazione Errori con gestione eventi e sub process dedicati

Un possibile esercizio

Realizzare un processo che mostri a video un elenco contenente i nomi e il tipo di tutte le tabelle di sistema. L'utente dovrà digitare il nome di una tabella e dare conferma mediante apposito pulsante. Se il nome della tabella non è tra quelle in elenco visualizzare un messaggio di errore permettendone il reinserimento. Se il nome è corretto invocare il web service echo. Alla prima invocazione dovrà essere simulato un errore con relativa gestione a livello utente mediante apposita view che consente di ritentare l'operazione. Al secondo tentativo l'operazione dovrà avere esito positivo mostrando in una view il buon esito dell'operazione (risposta del web service)

Variante: gestione errore mediante sottoprocesso dedicato per l'amministratore...

Nello svolgimento si dovrà creare un process context con (data creazione processo, utente, stato processo, nome tabella selezionata dall'utente) e un opportuno error model.