

## Lezione 4 modulo 1

In questo modulo, vedremo alcuni degli elementi essenziali di BPMN, riprendendo quelli che abbiamo già introdotto e aggiungendone altri che fanno parte dell'insieme degli elementi essenziali di questa notazione. Abbiamo già introdotto il concetto di attività, di gateway e di eventi. Per le attività, abbiamo visto semplici attività, abbiamo detto che nelle attività dovremo inserire un nome che caratterizza l'attività. Abbiamo visto i gateway di tipo XOR e parallelo (o AND). Abbiamo visto gli eventi di inizio, di fine, come eventi semplici, e abbiamo detto che possiamo avere anche eventi legati a messaggi che possono essere inseriti all'interno di un processo come eventi intermedi, oppure possono iniziare un processo, quindi avremo degli eventi di start che sono con dei messaggi, quindi un messaggio fa iniziare il processo, o degli eventi di fine, quindi posso anche terminare un processo inviando un messaggio. Abbiamo anche già visto che esistono dei connettori di flusso, quindi i Process flow, oppure di invio di messaggi, Message flow, che consentono di collegare tra di loro diversi attori nel processo che saranno in pool differenti. Adesso andremo a introdurre alcuni altri simboli che vediamo in questa figura. Cominciamo inizialmente a vedere degli elementi che sono legati al tempo, saranno dei timer che potremmo mettere all'inizio di un processo oppure come eventi intermedi. Vediamo come possiamo utilizzare questi eventi che tipicamente, come altri elementi di BPMN saranno associati a un'etichetta, ad esempio per i messaggi abbiamo visto che qua vorremmo sempre etichettare anche il messaggio con il nome del messaggio. Anche per il timer avremo un'etichetta legata al tempo e il timer ci potrà dare, o una durata, vorremmo fare passare un intervallo di tempo, oppure ci dirà che avremo un evento che accade in un certo momento, quindi possiamo, ad esempio, dare una data con l'ora e il minuto che è l'inizio, ad esempio, di un certo processo. Vediamo l'uso del timer all'interno del processo. In un primo uso che possiamo vedere qui, è qualcosa che si interpone fra due attività e nel flusso che collega queste due attività, possiamo inserire questo evento intermedio timer all'interno del flusso fra A e B, ad esempio due ore, e questo è l'intervallo che dovrà passare dalla fine di A prima di poter iniziare B. Un altro modo di utilizzare il timer che viene usato piuttosto frequentemente, è quello di associare il timer allo svolgimento in un'attività. Ad esempio, in questo caso, noi vediamo un processo che ha due attività A e B. Questo processo normalmente verrà svolto nel seguente modo: inizia il processo, viene svolta A e termina il processo. Inserire un timer qui, ad esempio, se scrivo un'ora, vorrà dire che dopo un'ora dall'inizio di A, andrò a verificare una condizione che è un evento intermedio che è appunto lo scattare di queste timer. In questo caso, A viene terminata, quindi non proseguirò per il percorso normale che vediamo qui, ma andrò a seguire il nuovo percorso che ci porta a B, quindi questo è tipicamente un caso di timeout: abbiamo un timeout associato all'attività A che porta a eseguire B qualora l'attività A si prolunghi oltre un certo tempo prestabilito. L'inserimento, appunto degli eventi di tipo timer in un processo consente di gestire degli intervalli di attesa, oppure delle scadenze che si possono verificare quando vogliamo interrompere un'attività che si sta prolungando troppo, come nel caso che abbiamo appena visto. Vediamo l'altro simbolo che abbiamo introdotto nella prima figura di questo modulo che è quello del sotto processo. Il sotto processo viene disegnato indicando per una certa attività A un + al suo bordo che indica che questa attività, in realtà, essa stessa può essere descritta come un processo. Idealmente, quando noi parliamo di sotto processi, possiamo immaginare, ad esempio, che un'attività A è scomposta in altre attività, ad esempio, nel nostro esempio qua, A1, A2, questo non vuol dire che mi fermo ad A1, A2 anche A1 potrebbe essere a sua volta scomposta in altre attività A11, A12 e così via. Quindi noi abbiamo un raffinamento sostanzialmente di quella che è l'attività A che viene descritta in un processo separato che ne dà i dettagli. Questo si può fare in diversi modi. Vediamo qua un primo esempio: facciamo vedere all'interno di un flusso normale, quindi all'interno di un processo che stiamo già disegnando, la nostra attività e facciamo vedere all'interno del simbolo di A un processo che inizia e finisce. L'inizio e la fine qua sono l'inizio e la fine delle attività che compongono A. Quindi A, in realtà, sarà una sequenza di A1, A2. Questi sono, in generale, eventi di tipo semplice, soprattutto l'evento d'inizio corrisponde al fatto che io sto attivando A, arrivando qua con il flusso. Ovviamente se un processo è complesso, il disegno contiene molti elementi. Disegnare un'attività, soprattutto se essa stessa è complessa, potrebbe risultare un diagramma troppo complesso. Quindi



possiamo, in alternativa, adottare un altro modo di rappresentare questo sotto processo. Possiamo disegnare all'interno di un processo la nostra attività come attività sotto processo e poi separatamente, quindi in un altro diagramma che possiamo, ad esempio, etichettare con lo stesso nome dell'attività, possiamo avere la descrizione del processo separatamente. Questo è assolutamente equivalente al diagramma precedente. Quindi possiamo descrivere il nostro sotto processo all'interno del simbolo di A, oppure possiamo avere più diagrammi che uniti rappresenteranno esattamente la stessa situazione. Il sotto processo viene quindi utilizzato per poter descrivere dei processi complessi in dettaglio ma senza farvi vedere subito tutti i dettagli, ma raffinando man mano le attività in sotto attività in sotto- sotto attività con processi relativi, per poter descrivere tutti i dettagli del processo. Vediamo adesso anche un altro simbolo che inseriamo nella nostra notazione di base e che ci può essere utile. Abbiamo già parlato dei loop, dei cicli. Abbiamo visto che un loop può essere rappresentato indicando delle condizioni, ad esempio, in questo caso, entriamo nel ciclo, eseguiamo A, verifichiamo una certa condizione, se la condizione 2 è verificata eseguiamo B e riprenderemo il nostro ciclo, fino a che non avremo, abbiamo detto che questa è una condizione esclusiva, la condizione 2 negata, che ci consentirà di uscire dal ciclo. Spesso in un processo, i cicli possono comprendere diverse attività e possono anche essere abbastanza complessi nella loro struttura. Quindi, analogamente a quello che abbiamo fatto con i sotto processi, potremmo adottare una notazione che ci dice che una certa attività è un ciclo. Questo come lo potremmo indicare? Potremmo indicare che la nostra attività, chiamiamola L, per ciclo, è un loop indicandola con un simbolo di questo tipo. Questo vuol dire che noi continueremo a svolgere una certa attività finché non verrà effettivamente terminato il nostro ciclo. Questo è un ciclo che ha una natura di questo tipo: noi andremo, ad esempio, in un caso di questo tipo, sostanzialmente ad eseguire un'attività L testeremo una condizione che ci dirà se continuare oppure no, oppure se uscire dal ciclo e quindi andremo sostanzialmente ad invocare l'attività L una serie di volte in sequenza. Per i loop, quindi, avremo la possibilità di disegnare delle attività di questo tipo, di tipo ciclo, ma in realtà quello che può essere utile a volte, è iniziare questa attività in parallelo, perché può essere che io voglia, ad esempio, mandare molte richieste, ad esempio, per procurarmi delle voci che voglio ordinare oppure possono fare dei controlli che voglio fare in parallelo e così via. Quindi introdurremo un concetto che è quello di Multi Instance, in cui noi vogliamo sostanzialmente cominciare delle attività in parallelo e lo potremo fare in due modi diversi: potremmo avere un Multi Instance, in cui effettivamente avremo un Multi Instance parallelo, in cui, appunto, inizieremo questa attività in tante istanze, quindi, analogamente a quello che abbiamo fatto prima, abbiamo detto che A veniva svolta più volte, ci sarà un a1, seguito da un a2, seguita da un a3. Nel caso del Multi Instance parallelo, sostanzialmente quello che sto dicendo, è che avrò ancora tante istanze ma le inizierò in parallelo. Nel mentre, rimangono i vari discorsi che abbiamo detto su quello che è il significato del parallelo all'interno di una notazione di questo tipo, ma vogliamo dire che sostanzialmente le vogliamo eseguire tutte e non ci interessa, in particolare, il loro ordine. In altri casi, però, vogliamo fare qualcosa che un po' intermedio rispetto a queste due situazioni e che rappresenteremo con un simbolo simile, sempre un Multi Instance ma di tipo sequenziale. Qui sostanzialmente vorremmo dare un ordine a questa attività che svolgiamo in parallelo, quindi faremo prima a1, poi a2, poi a3, consentiamo di averli in parallelo, quindi non abbiamo il caso che abbiamo indicato qui, ma possiamo avere delle attività che si svolgono contemporaneamente ma il loro inizio avverrà in sequenza. Tramite questi costrutti in loop, noi possiamo andare a rappresentare dei processi anche abbastanza complessi, nascondendo quella che è la complessità del fatto di iterare dell'operazioni all'interno di un processo, che se utilizziamo i gateway di tipo XOR, ne cambia un po' la natura, perché vediamo che torniamo indietro nel processo, anziché proseguire. Quindi avremo processi di questo tipo, ad esempio per il nostro ordine potremo inserire i dettagli dell'ordine, puoi inviare delle richieste al magazzino e abbiamo più richieste in parallelo, riceviamo quei prodotti e, un'altra cosa che possiamo fare come loop, è imballare i prodotti uno dopo l'altro, quindi abbiamo un ciclo in questo caso di tipo loop semplice e poi spedire l'ordine e così via. Quindi un'altra cosa da aggiungere è che nulla ci impedisce di avere situazioni di questo tipo: presa un'attività A e uno dei loop che abbiamo visto, dire che al suo interno abbiamo un sotto processo. In questo caso, possiamo rappresentare anche delle situazioni che



non sono semplici attività, ma in realtà insiemi di attività svolte all'interno del sotto processo in modo ciclico. Per concludere, andiamo a vedere qualche dettaglio in più riguardo l'uso dei gateway che abbiamo visto, soprattutto per quanto riguarda l'utilizzo del gateway di OR esclusivo. Abbiamo visto che lo utilizziamo normalmente in questo modo: cominciamo dei percorsi A e B e tipicamente ci aspettiamo di avere un join nel processo per proseguire all'interno del processo. Può essere in un processo che questo possa, in realtà, essere invece di A una serie di attività, un percorso completamente separato da quello che sarà il percorso che invece abbiamo nell'altro ramo. Un'altra cosa che si può fare all'interno di questa notazione è dire che si termina il processo in questo modo: ad esempio, una notazione di questo tipo è sostanzialmente equivalente a quella che abbiamo visto prima, la useremo soprattutto quando poi il processo non continua con altre attività comuni che comunque andranno svolte dopo A e B, ma semplicemente vogliamo poi andare a terminare il processo e quindi non abbiamo l'esigenza di continuare a descriverlo e soprattutto, dal punto di vista grafico, è più semplice una descrizione che ci va a dire che i vari rami terminano. Vorremmo anche fare qualche osservazione sull'utilizzo dei gateway. Una situazione che si può verificare è la seguente: A e B vengono svolte in alternativa e poi voglio svolgere in alternativa C e D. Ovviamente non sto mettendo le condizioni, ma ci aspettiamo sempre di avere delle condizioni di questo tipo, esclusive per i nostri percorsi. Questo ovviamente è un join e questo è uno split che mi divide il flusso. Non posso fare operazioni di questo tipo: più ingressi e più uscite da uno stesso gateway perché, in questo caso, utilizzo in realtà due gateway in uno solo e creo un'ambiguità nella rappresentazione.

