

UNIVERSITA DI PISA

ANNO ACCADEMICO 2018/2019

TURNI

BUSINESS PROCESS MODELING



CINQUINI MARTINA
MORINI VIRGINIA

Introduzione

Lo scenario su cui si focalizza l'analisi è incentrato sulle interazioni tra due impiegati che devono concordare il piano di ferie per l'anno venturo in modo da garantire la disponibilità di almeno una persona alla postazione in ogni giorno lavorativo. Ciascuno dei due impiegati può accedere al sistema elettronico e cominciare a prenotare i giorni di ferie, selezionando, deselectionando o azzerando il piano per cominciare nuovamente. Il piano di ciascun impiegato non è visibile all'altro.

Il problema può essere suddiviso in due scenari.

Nel primo scenario, una volta sottoposto il piano ad approvazione, il sistema elettronico rileva automaticamente i conflitti che lascerebbero sguarnita la postazione e se segnalati gli impiegati possono attivare una conversazione dove, a turno, si scambiano dei messaggi. Al termine della conversazione, ciascun impiegato può modificare il proprio piano di ferie per sottoporlo nuovamente ad approvazione.

Nel secondo scenario invece, i conflitti possono essere segnalati dal sistema anche prima che i piani vengano sottoposti ad approvazione.

Per la modellazione di entrambi gli scenari viene illustrato di seguito un modello astratto con rappresentazione grafica BPMN e la corrispondente conversione in Rete di Petri.

Progettazione BPMN

Per l'astrazione del problema è stata scelta la notazione BPMN che ha l'obiettivo di fornire uno standard di rappresentazione di facile utilizzo e comprensione per tutti gli utenti interessati al problema della modellazione, progettazione ed eventuale informatizzazione dei processi aziendali¹.

Primo scenario

Pool Impiegato A e Impiegato B

Le pool dei due impiegati coinvolti nel processo di prenotazione ferie saranno inizialmente descritte insieme in quanto presentano numerose affinità.

Gli impiegati avviano il processo accedendo al sistema elettronico. L'inserimento delle credenziali è modellato utilizzando una *send task* alla pool del sistema elettronico (Figura 1).

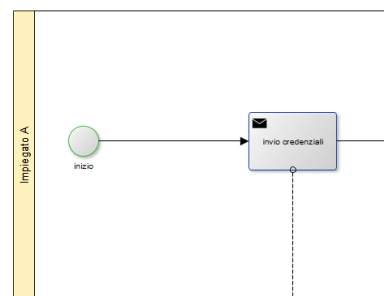


Figura 1. Pool impiegato A e B accesso

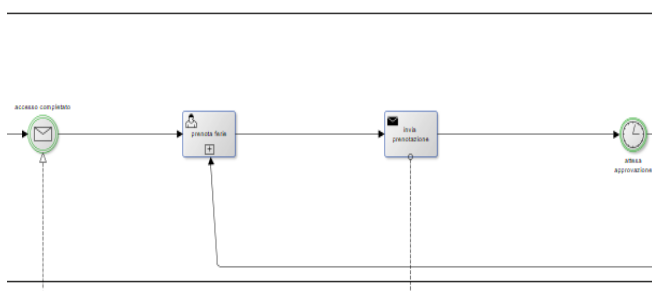


Figura 2. Pool impiegato A e B prenotazione ferie

Dopo aver ricevuto un messaggio di comunicazione di avvenuto accesso da parte del sistema elettronico, gli impiegati iniziano a prenotare il proprio piano di ferie.

Si è scelto di modellare il processo di prenotazione come un sottoprocesso, ossia come un'attività costituita a sua volta da altre attività minori, con il fine di aumentare il livello di comprensione del diagramma (Figura 2 e Figura 3).

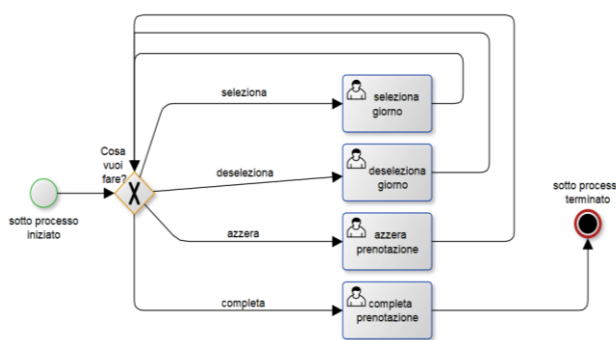


Figura 3. Pool impiegato A e B sottoprocesso prenota ferie

All'interno del sottoprocesso gli utenti, iterativamente, possono selezionare o delezionare un giorno di ferie tramite gli *user task* 'selezione giorno' e 'deselezione giorno' oppure possono azzerare il piano e ricominciare a prenotare (*user task* 'azzerà prenotazione'). Inoltre, in qualsiasi momento, possono terminare il sottoprocesso, decidendo di completare la prenotazione tramite l'*user task* 'completa prenotazione'. La scelta di svolgere una delle azioni sovraccitate è stata modellata con l'uso dell'*exclusive gateway* 'Cosa vuoi fare?'. Infine dopo la terminazione del sottoprocesso gli impiegati tramite una *send task* inviano la propria prenotazione al sistema elettronico e attendono

¹ <http://static.gest.unipd.it/labtesi/eb-didattica/GIAR/2011/bpmn-appunti.pdf>

di sapere l'esito dell'approvazione.

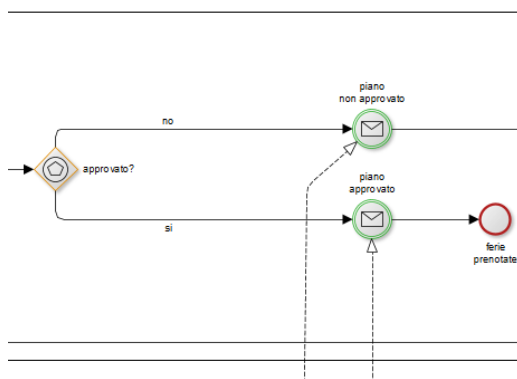


Figura 4. Pool impiegato A e B approvazione

Il sistema elettronico può approvare o meno il piano presentato dai due impiegati. Tali opzioni sono implementate nella pool degli impiegati tramite l'uso dell'*event based gateway* 'approvato?'. In caso di esito positivo gli impiegati ricevono un messaggio di approvazione e il processo termina con la prenotazione definitiva delle ferie.

In caso di esito negativo, i due impiegati iniziano una chat dove a turno si scambiano messaggi. Di seguito le interazioni tra i due utenti, mediate dal sistema elettronico, saranno illustrate separatamente.

Chat impiegato A

Arbitrariamente si è deciso di far inviare il primo messaggio all'impiegato A tramite una *send task*. Dopo aver atteso e ricevuto la risposta, l'impiegato A ha due possibilità (*event based gateway* 'chat finita?'): terminare la chat, comunicarlo al sistema e a B tramite l'invio di un messaggio e iniziare nuovamente a prenotare le ferie; continuare la conversazione tramite la *send task* precedente.

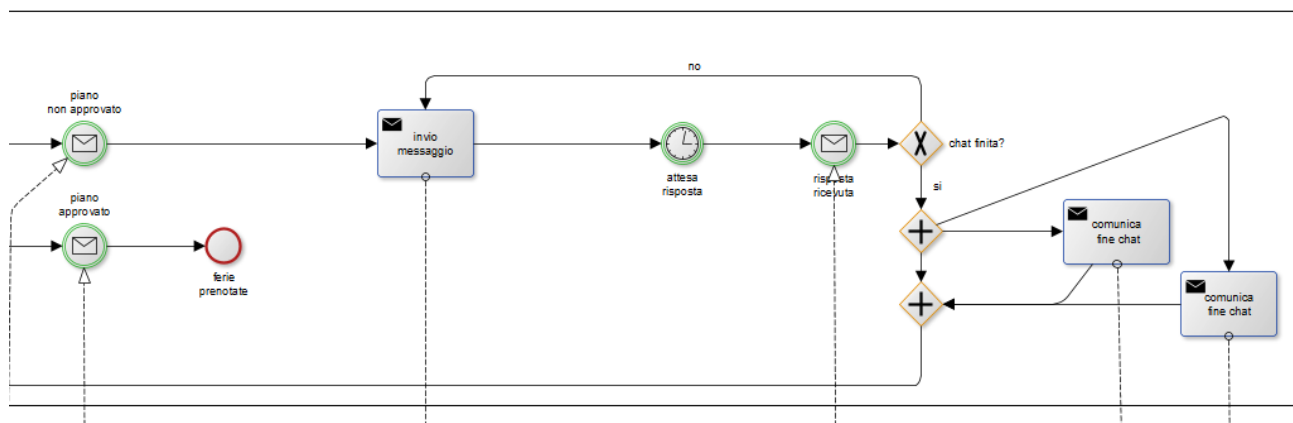


Figura 5. Pool impiegato A chat

Chat impiegato B

In caso di non approvazione del piano, l'impiegato B attende di ricevere un messaggio da A. Una volta ricevuto, invia la risposta ad A tramite una *send task* e attende a sua volta di ricevere un messaggio da A. Se B riceve da A il messaggio di fine chat può tornare al sotto processo 'prenota ferie' e iniziare una nuova prenotazione, altrimenti dopo la ricezione del messaggio di A continua la conversazione.

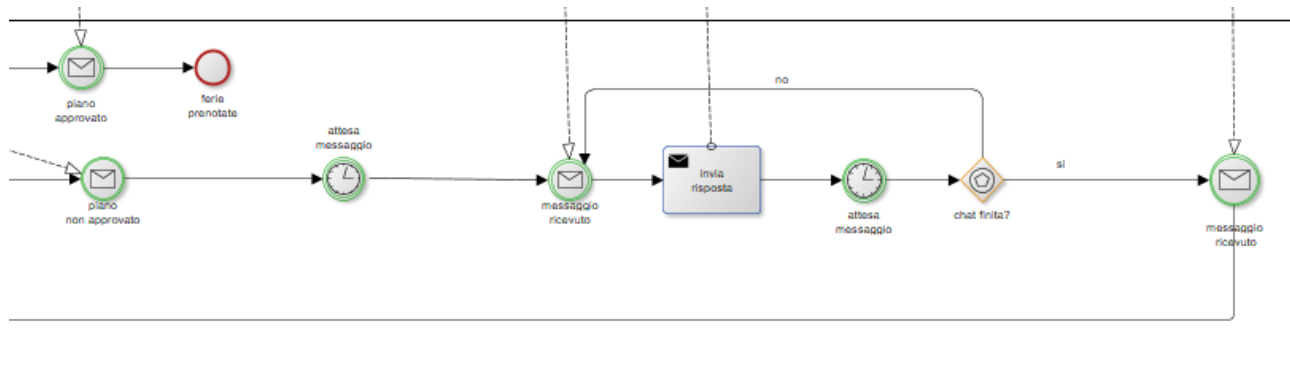


Figura 6. Pool impiegato B chat

Pool Sistema elettronico

Il Sistema elettronico inizia il processo attendendo le credenziali degli impiegati intenzionati ad effettuare l'accesso. Dopo aver ricevuto le credenziali sia da A che da B può completare l'accesso. Si è scelto di modellare tali parti del processo attraverso un *parallel gateway* per motivi di efficienza (Figura 7).

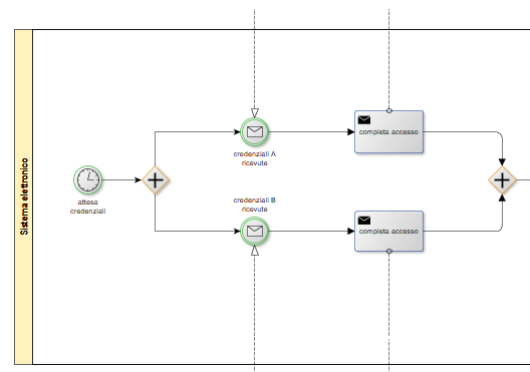


Figura 7. Pool Sistema Elettronico accesso

Successivamente, dopo aver atteso e ricevuto il piano di ferie dei due impiegati (modellato nuovamente con un *parallel gateway*) il sistema elettronico, tramite il *service task* 'approva richiesta', valuta se i piani presentano dei conflitti attraverso l'*event based gateway* 'approvato?' (Figura 8). In caso positivo i piani non vengono approvati, altrimenti il processo termina (Figura 9).

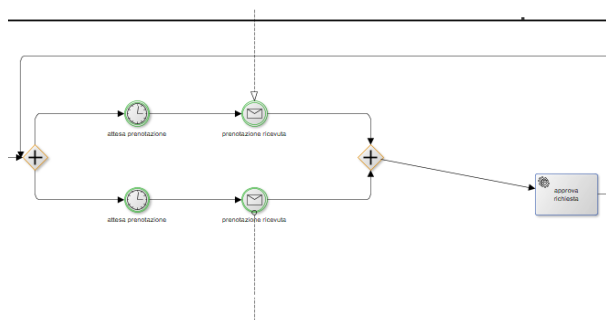


Figura 9. Pool sistema elettronico approvazione

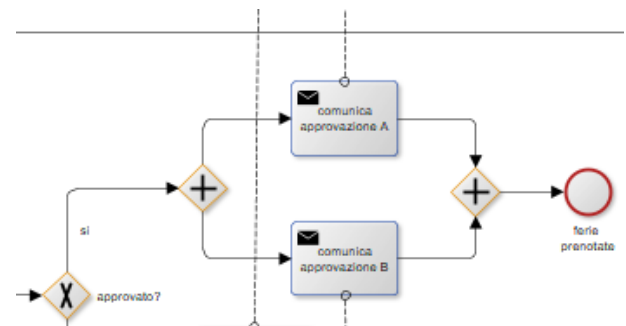


Figura 8. Pool sistema elettronico gestione prenotazione

Se il sistema rileva conflitti invia, in parallelo a entrambi gli impiegati, un messaggio in cui comunica la mancata approvazione. A questo punto, il sistema attende che l'impiegato A invii il primo messaggio per attivare la chat. Come spiegato in precedenza, il sistema svolge durante la chat un ruolo di mediazione: riceve il messaggio dall'impiegato A e lo spedisce all'impiegato B; lo stesso avviene anche per le interazioni successive. Infine, se il sistema riceve da A il messaggio di fine chat può tornare al *parallel gateway* di attesa e ricezione delle prenotazioni (Figura 8), altrimenti torna all'evento intermedio 'attesa messaggio', continuando la mediazione.

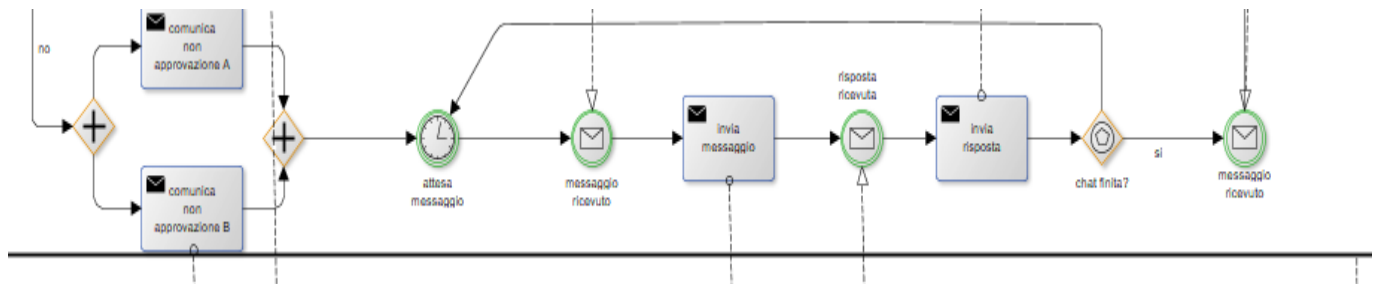


Figura 10. Pool sistema elettronico chat

Processo collaborativo

In figura 11 viene presentato il processo collaborativo tra le tre pool precedentemente descritte. L'interazione tra le pool è resa possibile dai *message flow*.

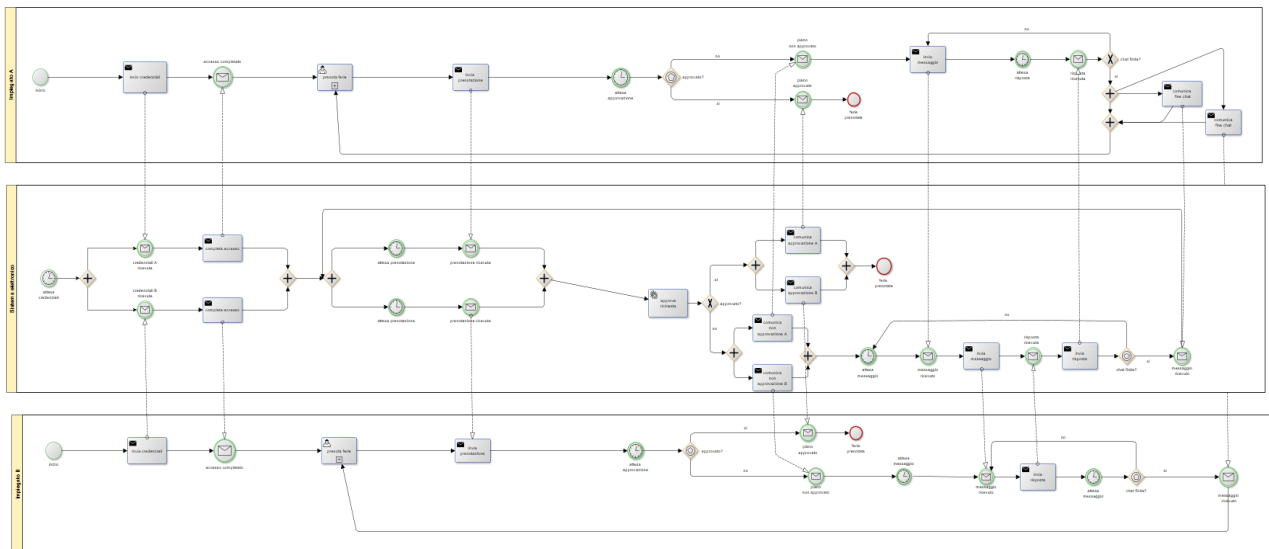


Figura 11. Processo collaborativo primo scenario

Secondo Scenario

Il secondo scenario richiede di modificare i processi in modo che i conflitti possano essere segnalati dal sistema anche prima che i piani vengano sottoposti ad approvazione.

Al fine di modellare la segnalazione dei conflitti seguendo le specifiche del progetto, si è scelto di inserire nella pool del sistema elettronico la task per prenotare le ferie. A differenza del primo scenario, quindi, dopo che il sistema ha ricevuto la richiesta di accesso e di prenotazione da parte di entrambi gli utenti (Figura 12), coordina la scelta dei giorni di ferie. Quest'ultima è stata implementata utilizzando un sotto processo (descritto precedentemente nel primo scenario) per ciascuno dei due impiegati (Figura 13).

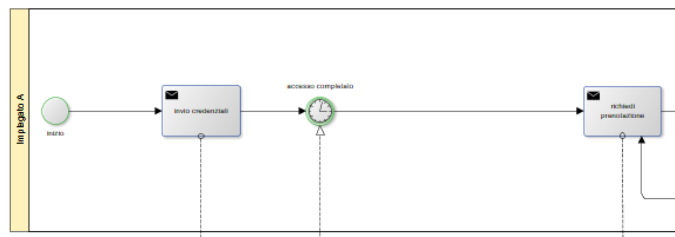


Figura 12. Pool impiegato A e B prenotazione

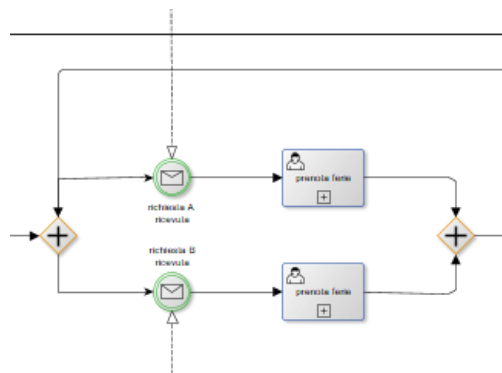


Figura 13. Pool sistema elettronico prenotazione

Vengono quindi prese in considerazione due opzioni tramite l'utilizzo di un *exclusive gateway*: nessun conflitto generato e la successiva approvazione del piano oppure la segnalazione di conflitti e l'apertura della conversazione tra gli utenti. La prima possibilità è stata modellata come nel precedente scenario mentre la seconda è stata gestita attraverso un parallel gateway per comunicare i conflitti agli impiegati con il successivo avvio della chat in modo che i conflitti possano essere risolti (Figura 14).

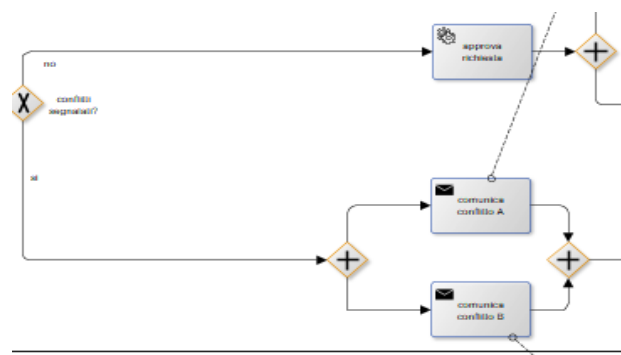


Figura 14. Pool sistema elettronico comunicazione conflitti

Processo collaborativo

In figura 15 viene presentato il processo collaborativo tra le tre pool precedentemente descritte. Anche in questo caso l'interazione tra le pool è resa possibile dai *message flow*.

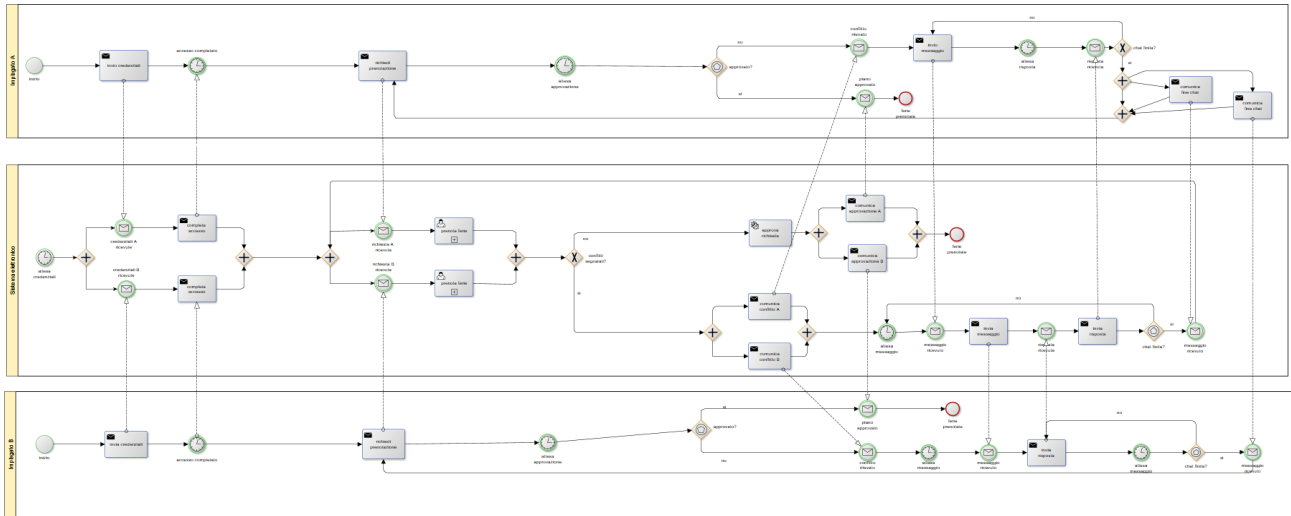


Figura 15. Processo collaborativo secondo scenario

Mapping da BPMN a rete di Petri

Per poter analizzare e verificare i processi implementati con la notazione BPM, è necessario convertirli nel modello formale della rete di Petri.

La conversione è stata effettuata mediante l'utilizzo di WoPeD (Workflow Petri net Designer), in particolare utilizzando il tool Semantical Analysis.

Di seguito, verranno descritti nel dettaglio i risultati ottenuti dall'implementazione delle reti di Petri relative agli impiegati A e B, al sistema elettronico e al processo collaborativo di entrambi gli scenari. Ciascuna analisi sarà illustrata mediante il grafo della rete, la sintesi delle caratteristiche semantiche fornite dal tool e alcune osservazioni sulle proprietà della rete.

Primo scenario

Impiegato A

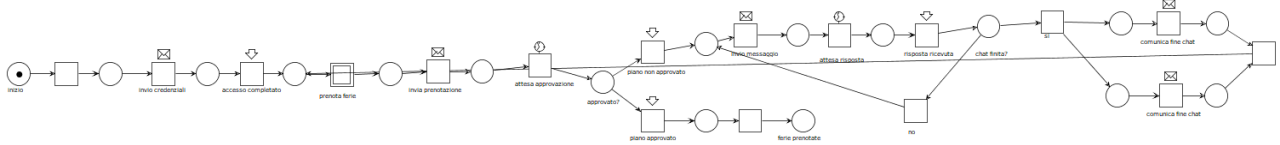


Figura 16: rete di Petri impiegato A

La rete dell'impiegato (Figura 16) A è composta da 17 places e transitions e da 35 archi. Come si evidenzia dall'analisi semantica (Figura 17), è *sound* quindi rispetta la proprietà di workflow net ed è *live* e *bounded*. Poiché *live* è anche *deadlock-free* e, dato che è *bounded*, il *grafo di raggiungibilità* è finito ma non coincide con quello di copertura: esso contiene 17 vertici e 19 archi.

Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

Non è *S-net* dato che esiste almeno una transizione con più di un place nel suo pre-set e/o post-set. Non è *T-net* dato che esiste almeno un place con più di una transizione nel suo pre-set e/o post-set. Ne consegue che non è possibile trovare né *S-invariant* né *T-invariant* del tipo $[k \dots k]$ dato un valore k .

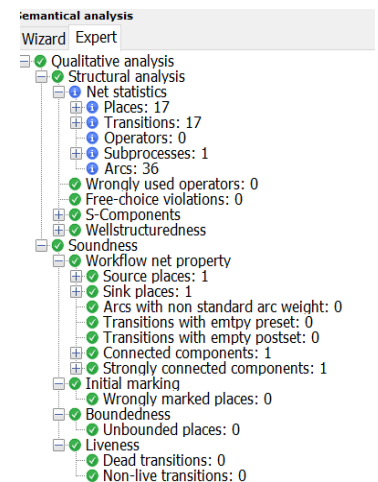


Figura 17: Analisi semantica impiegato A

Impiegato B

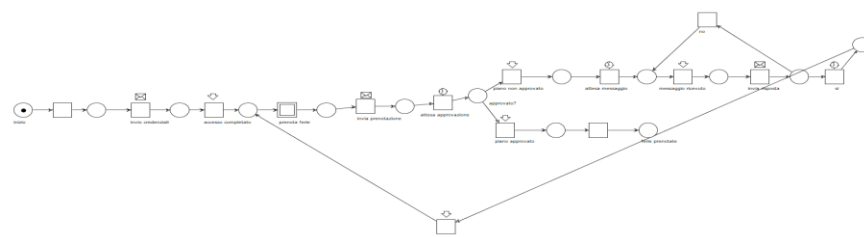


Figura 18: rete di Petri impiegato B

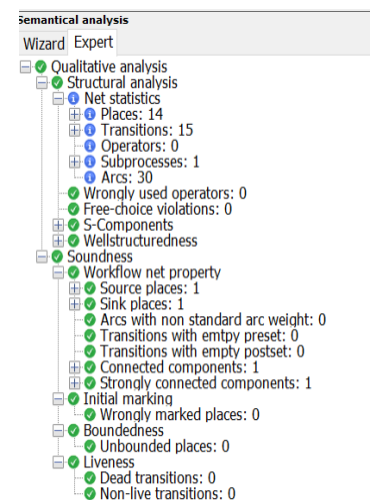


Figura 19: Analisi semantica impiegato B

La rete dell'impiegato B (Figura 18) è composta da 14 places, 15 transitions e da 30 archi. Come si evidenzia dall'analisi semantica, (Figura 19) è *sound* quindi rispetta la proprietà di workflow net ed è *live* e *bounded*. Poiché *live* è anche *deadlock-free* e, dato che è *bounded*, il *grafo di raggiungibilità* è finito e coincide con quello di copertura: esso contiene 14 vertici e 15 archi.

Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

La rete è *S-net* dato che ogni transizione ha un unico input place e un unico output place. Ne consegue che la rete è *S-System* e ammette tutti gli *S-invariant* del tipo $I=[k...k]$ e è quindi *safe*.

Infine, la rete non è *T-net* in quanto alcuni place hanno più di una transizione di input e/o output.

Sistema elettronico

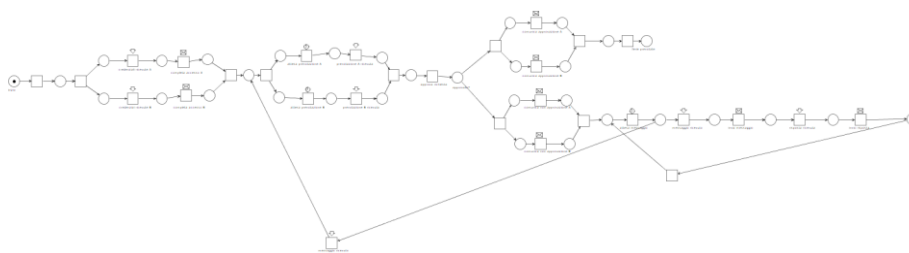


Figura 20: rete di petri sistema elettronico

La rete del sistema elettronico (Figura 20) è composta da 33 places, 30 transitions e da 68 archi. Come si evince dall'analisi semantica (Figura 21), è *sound* quindi rispetta la proprietà di workflow net ed è *live* e *bounded*. Poiché *live* è anche *deadlock-free* e, dato che è *bounded*, il *grafo di raggiungibilità* è finito ma non coincide con quello di copertura: esso contiene 39 vertici e 50 archi. Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

Non è *S-net* dato che esiste almeno una transizione con più di un place nel suo pre-set e/o post-set. Non è *T-net* dato che esiste almeno un place con più di una transizione nel suo pre-set e/o post-set. Ne consegue che non è possibile trovare né *S-invariant* né *T-invariant* del tipo $[k .. k]$ dato un valore k .

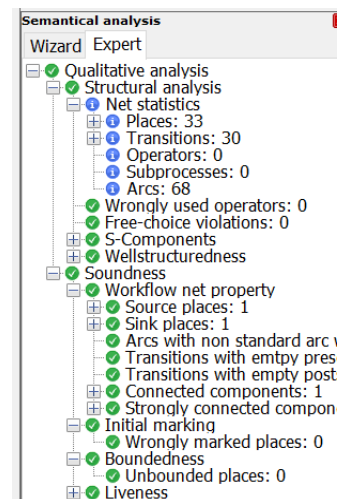


Figura 21: Analisi semantica del sistema elettronico

Processo collaborativo

Al fine di far interagire i workflow net precedentemente descritti e di rispecchiare il diagramma collaborativo realizzato in BPMN, è stata implementato un workflow net collaborativo (Figura 22).

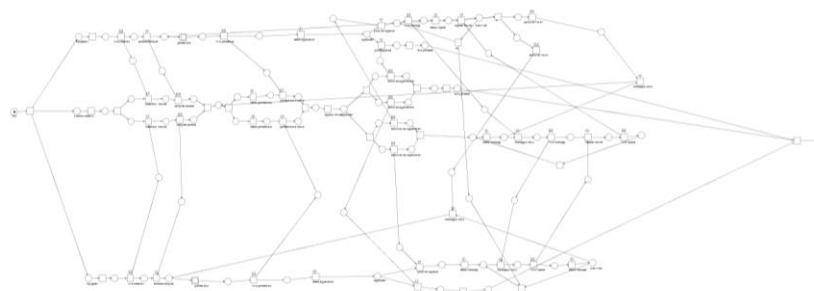


Figura 22: Processo collaborativo primo scenario

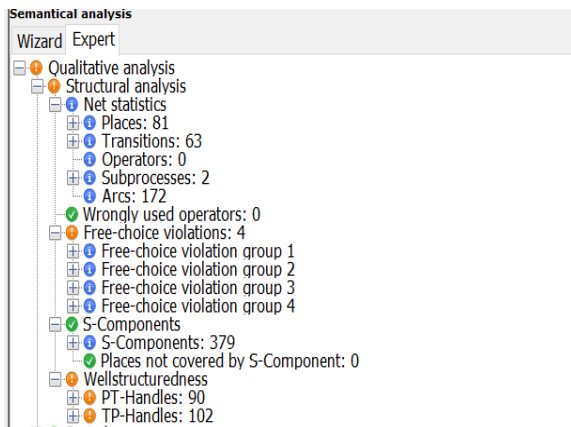


Figura 23 A: Analisi semantica processo collaborativo

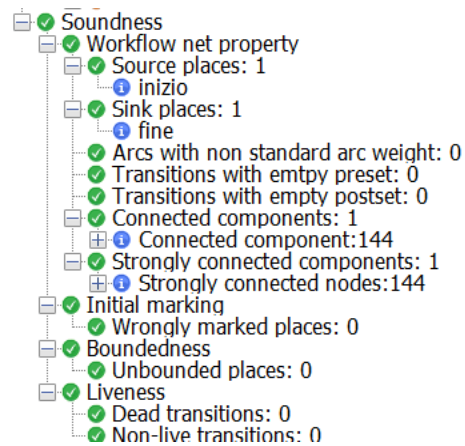


Figura23 B: Analisi semantica processo collaborativo

La rete collaborativa è composta da 81 places, 63 transitions e da 172 archi. Come mostrato nell'analisi semantica (Figura 23 A/B) è *sound* quindi rispetta la proprietà di workflow net ed è live e *bounded*. Considerando che è *live* è anche *deadlock-free*. Per quanto riguarda la struttura, la rete è sempre *s-coverable* ma non preserva le proprietà *well-structured* e *free-choice*. Quest'ultima risulta non rispettata a causa dell'aggiunta di un place al momento dell'interazione tra i moduli.

Secondo scenario

Impiegato A

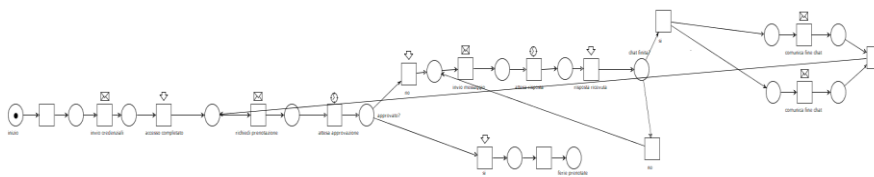


Figura 24: rete di Petri impiegato A Il scenario

La rete dell'impiegato (Figura 24) A è composta da 16 places e transitions e da 34 archi. Come si evidenzia dall'analisi semantica (Figura 25), è *sound* quindi rispetta la proprietà di workflow net ed è live e *bounded*. Poiché live è anche *deadlock-free* e, dato che è *bounded*, il *grafo di raggiungibilità* è finito ma non coincide con quello di copertura: esso contiene 16 vertici e 18 archi.

Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

Non è *S-net* dato che esiste almeno una transizione con più di un place nel suo pre-set e/o post-set. Non è *T-net* dato che esiste almeno un place con più di una transizione nel suo pre-set e/o post-set. Ne consegue che non è possibile trovare né *S-invariant* né *T-invariant* del tipo $[k \dots k]$ dato un valore k .

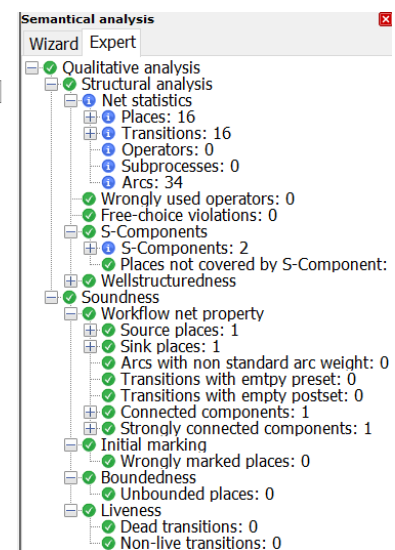


Figura 25: Analisi semantica impiegato A

Impiegato B

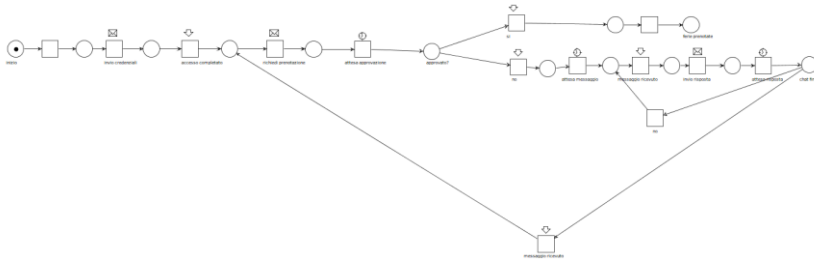


Figura 26: rete di Petri impiegato B II scenario

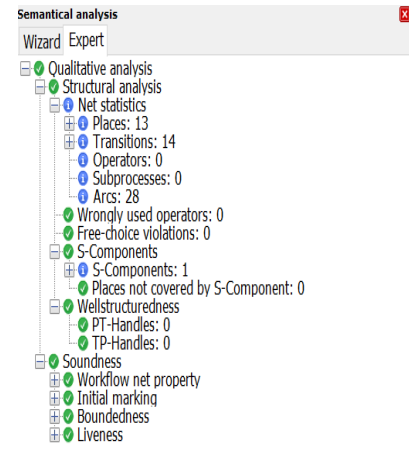


Figura 27: Analisi semantica impiegato B

La rete dell'impiegato B (Figura 26) è composta da 13 places, 14 transitions e da 28 archi. Poiché live è anche *deadlock-free* e, dato che è bounded, il *grafo di raggiungibilità* è finito e coincide con quello di copertura: esso contiene 13 vertici e 14 archi.

Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

La rete è *S-net* dato che ogni transizione ha un unico input place e un unico output place. Ne consegue che la rete è *S-System* e ammette tutti gli *S-invariant* del tipo $I=[k...k]$ e è quindi *safe*.

Infine, la rete non è *T-net* in quanto alcuni place hanno più di una transizione di input e/o output.

Sistema elettronico

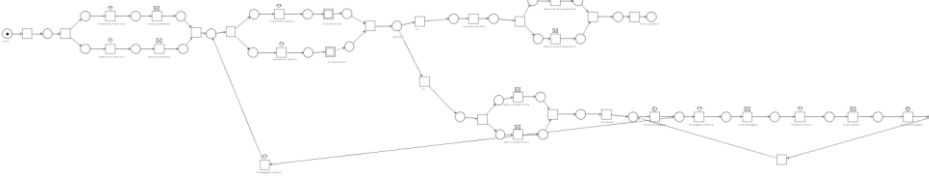


Figura 27: rete di Petri sistema elettronico

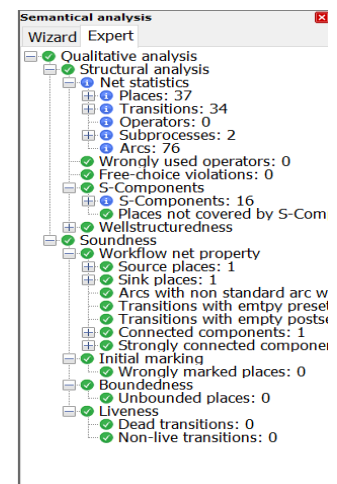


Figura 28: Analisi semantica sistema elettronico

La rete del sistema elettronico (Figura 27) è composta da 37 places, 34 transitions e da 76 archi. Come si evidenzia dall'analisi semantica (Figura 28), è sound quindi rispetta la proprietà di workflow net ed è live e bounded. Poiché live è anche *deadlock-free* e, dato che è bounded, il *grafo di raggiungibilità* è finito ma non coincide con quello di copertura: esso contiene 43 vertici e 54 archi. Per quanto riguarda la struttura è *free-choice*, *well-structured* e *s-coverable*.

Non è *S-net* dato che esiste almeno una transizione con più di un place nel suo pre-set e/o post-set. Non è *T-net* dato che esiste almeno un place con più di una transizione nel suo pre-set e/o post-set. Ne consegue che non è possibile trovare né *S-invariant* né *T-invariant* del tipo $[k .. k]$ dato un valore k .

Processo collaborativo

Anche per questo scenario è stato implementato un workflow net collaborativo tramite l'interazione dei workflow precedentemente descritte.

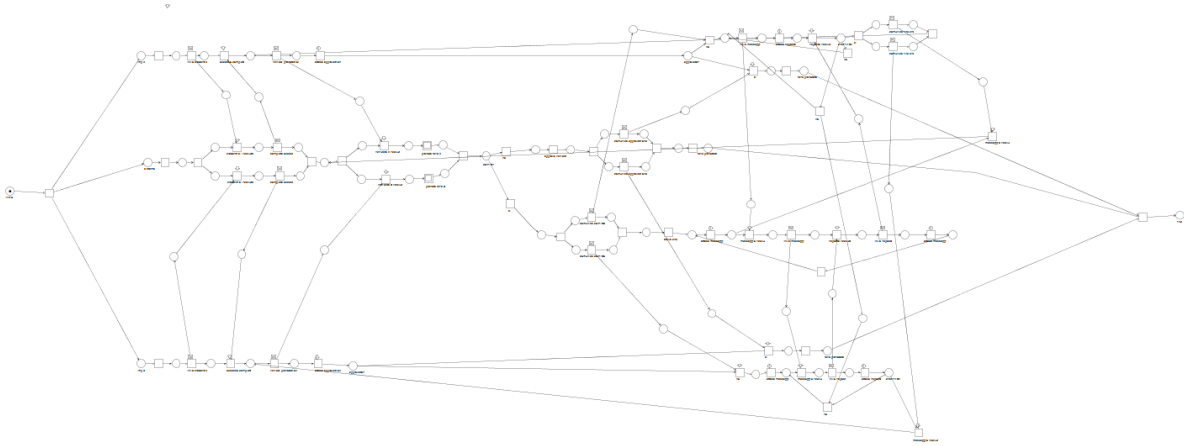


Figura 29: rete di petri processo collaborativo

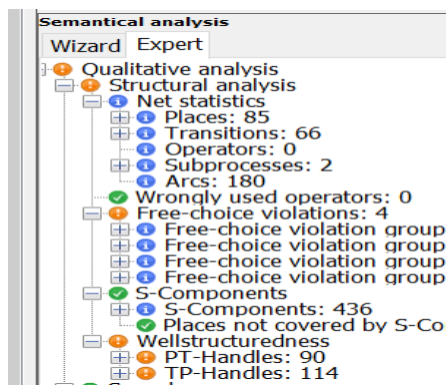


Figura 30 A: analisi semantica processo collaborativo

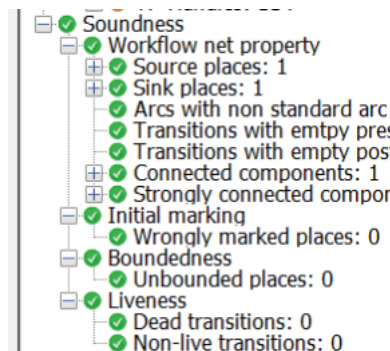


Figura 30 B: analisi semantica processo collaborativo

La rete del processo collaborativo (Figura 29) è composta da 85 places, 66 transitions e da 180 archi. Come mostrato nell'analisi semantica (Figura 30 A/B), è *sound* quindi rispetta la proprietà di workflow net ed è *live* e *bounded*. Considerando che è *live* è anche *deadlock-free*. Per quanto riguarda la struttura, la rete è sempre *s-coverable* ma non preserva le proprietà *well-structured* e *free-choice*. Quest'ultima risulta non rispettata a causa dell'aggiunta di un place al momento dell'interazione tra i moduli.

Conclusioni

La differenza significativa rilevata nel confronto tra i due scenari è la gestione della task 'prenota ferie' tra le tre pool coinvolte. Le altre parti del processo rimangono invece sostanzialmente invariate.

