

Università degli Studi di Pisa



BUSINESS PROCESS MODELING

(Professore Roberto Bruni)

PROGETTO: NEGOZIAZIONE

Riccardo Figliozzi
609652

A.A. 2019/2020

Indice

[Introduzione](#)

[Modelli](#)

[Variante dello scenario](#)

[Collaboration diagram](#)

[Conversione in Petri-net con Woped](#)

[Acquirente](#)

[Analisi Acquirente](#)

[Fornitore](#)

[Analisi Fornitore](#)

[Collaboration diagram in Petri-Net](#)

[Analisi Negoziazione](#)

[Conclusioni](#)

INTRODUZIONE

Il progetto si basa sulla realizzazione di diverse **workflow net** impiegando una serie di **modelli astratti**, in questo caso **BPMN**. Lo scenario preso in considerazione è quello di un processo di **negoziazione** in cui gli attori sono: l'Acquirente ed il Fornitore.

La negoziazione inizia quando l'Acquirente decide di fare una **prima offerta** al Fornitore. Entrambi gli attori possono **accettare** l'ultima offerta fatta dall'altro oppure **rilanciare** con una nuova. Nel momento in cui uno dei due attori accetta l'offerta dell'altro, l'Acquirente può **inviare i dati** per effettuare il pagamento **oppure** cambiare idea e **rinunciare** all'acquisto.

Se l'Acquirente decide di inviare i dati, conferma il pagamento, il quale deve essere anche accettato dalla banca. Successivamente, solamente dopo che il Fornitore avrà ricevuto il pagamento, potrà **inviare la chiave** di attivazione del servizio, in modo tale che entrambi gli attori possano terminare il processo.

Oltre al modello dello scenario appena descritto viene richiesto anche una **variante** nella quale, dopo aver ricevuto la chiave, l'Acquirente può scegliere se negoziare un **nuovo servizio** oppure **terminare**.

Per la rappresentazione degli scenari proposti sono stati utilizzati i modelli di astrazione **BPMN** (*Business Process Modeling Notation*) e per le analisi del processo le corrispondenti **Petri-Net**.

MODELLI

Innanzitutto è necessario creare il **modello BPMN del processo di negoziazione**¹ per entrambi gli attori. Come si può vedere sia dall'immagine Fig.1 sia dal modello intero (Fig.4), sono presenti **due pool** separate per attore (Acquirente e Fornitore).

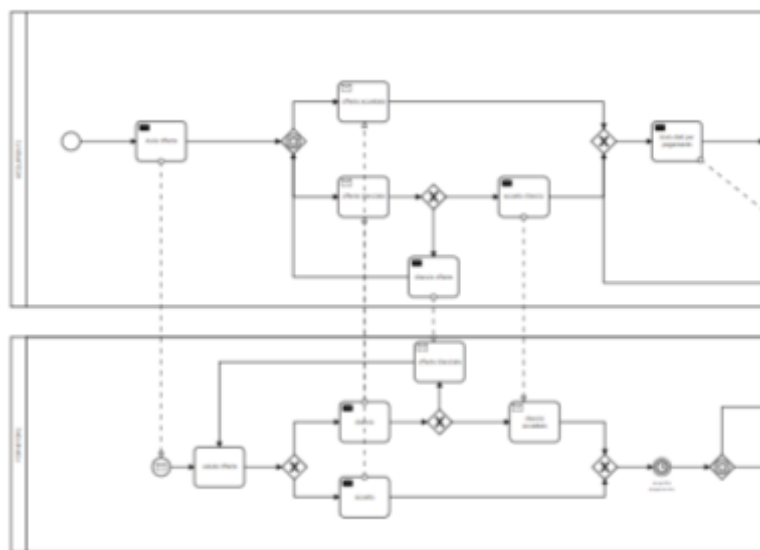


Fig.1: parte iniziale del processo in BPMN

La negoziazione **inizia** quando l'Acquirente invia l'offerta al Fornitore. Quest'ultimo, dopo averla ricevuta (*message intermediate task*), valuta l'offerta decidendo se accettarla o rilanciare (*XOR-split gateway*).

Da notare che all'interno della pool dell'Acquirente è stato inserito un **event base gateway**, poiché le *task* successive dipendono dalla risposta del Fornitore.

¹ Lo strumento utilizzato per lo sviluppo dei modelli in BPMN è bpmn.io.

Se l'offerta viene **accettata** dal Fornitore allora l'Acquirente riceverà l'accettazione dell'offerta, se invece il Fornitore decide di **rilanciare**, l'Acquirente riceverà un rilancio. A questo punto potrà scegliere se **accettare** il rilancio del Fornitore o **rilanciare** a sua volta con un'altra offerta; in entrambi i casi il Fornitore dovrà ricevere la decisione dell'Acquirente.

Questo scambio di informazioni appena descritto è stato rappresentato nel modello come un **loop** per entrambi i processi degli attori.

Dopo che uno degli attori decide di accettare l'offerta, l'Acquirente dovrà prendere un'altra decisione, in questo caso sono presenti degli **XOR-join**, in particolare uno **XOR-join/split**, poiché l'Acquirente può decidere se **rinunciare** definitivamente all'acquisto del servizio, e quindi terminare il processo, o **continuare** con la negoziazione.

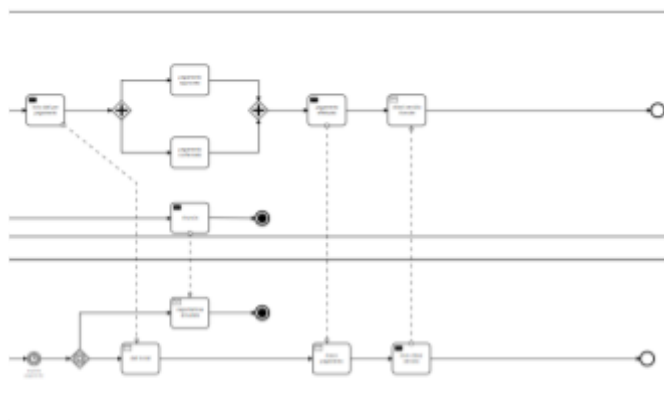


Fig.2: continuazione del processo in BPMN

Anche in questa fase, per quanto riguarda il Fornitore, sono stati inseriti un *timer intermediate catch event* ("aspetta acquirente") ed un *event base gateway* dato che in base alle decisioni dell'Acquirente, il Fornitore dovrà ricevere o un messaggio di **rinuncia** all'acquisto (in questo caso entrambi i processi degli attori terminano con un *terminate end event*) o di **conferma** dell'invio dei dati.

Una volta confermato il pagamento ed accettato dalla banca (questo processo avviene all'interno di un **parallel path**, AND-split + AND-join), il Fornitore invia una chiave di attivazione del servizio all'Acquirente per terminare così il processo.

Variante dello scenario

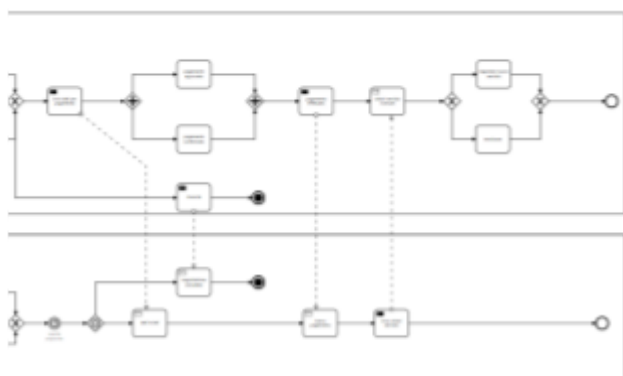


Fig.3: Variante dello scenario

Nel caso della **variante** del processo sopra descritto, dopo che l'Acquirente ha ricevuto le chiavi del servizio, è stato aggiunto uno **XOR-split** in modo tale che possa scegliere se **terminare** il processo o **negoziare un nuovo servizio**.

COLLABORATION DIAGRAM

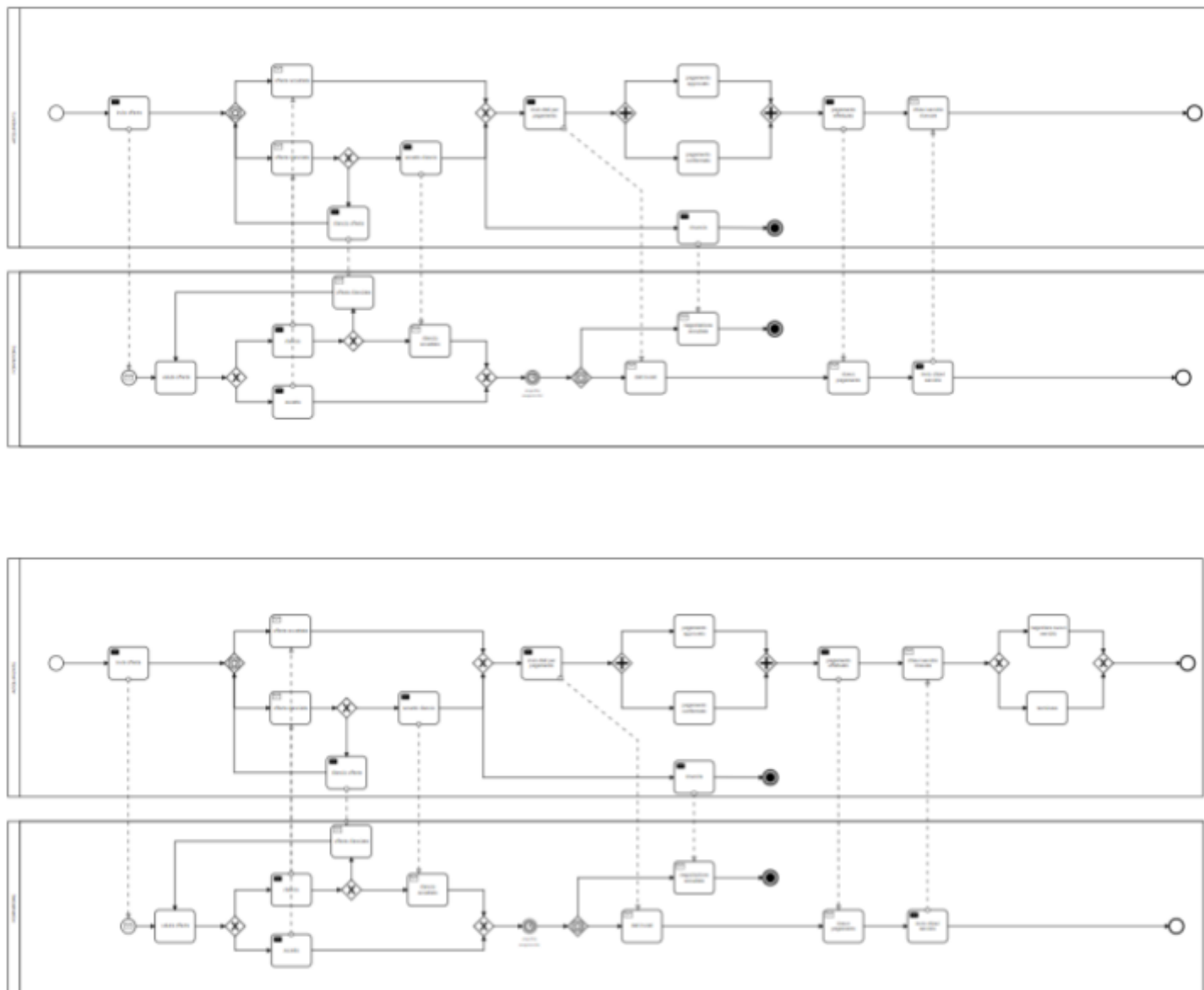


Fig.4: Processo di negoziazione senza variante (sopra) con variante (sotto)

CONVERSIONE IN PETRI-NET CON WOPED

Dopo aver sviluppato i modelli in *BPMN*, è necessario convertirli in **Petri-Net** per analizzare le varie proprietà e verificare la correttezza delle reti. Per queste verifiche sarà importante capire se le reti siano **sound** o meno.

Acquirente

Per la creazione della *Petri-net* dell'Acquirente è necessario aggiungere una **piazza di inizio** senza flussi in entrata ed aggiungendo un *token*, ed una **piazza finale** senza flussi in uscita. Da notare che la transizione "rinuncia" è stata collegata alla piazza finale, per rispettare le proprietà della rete (*unique end-event place*).

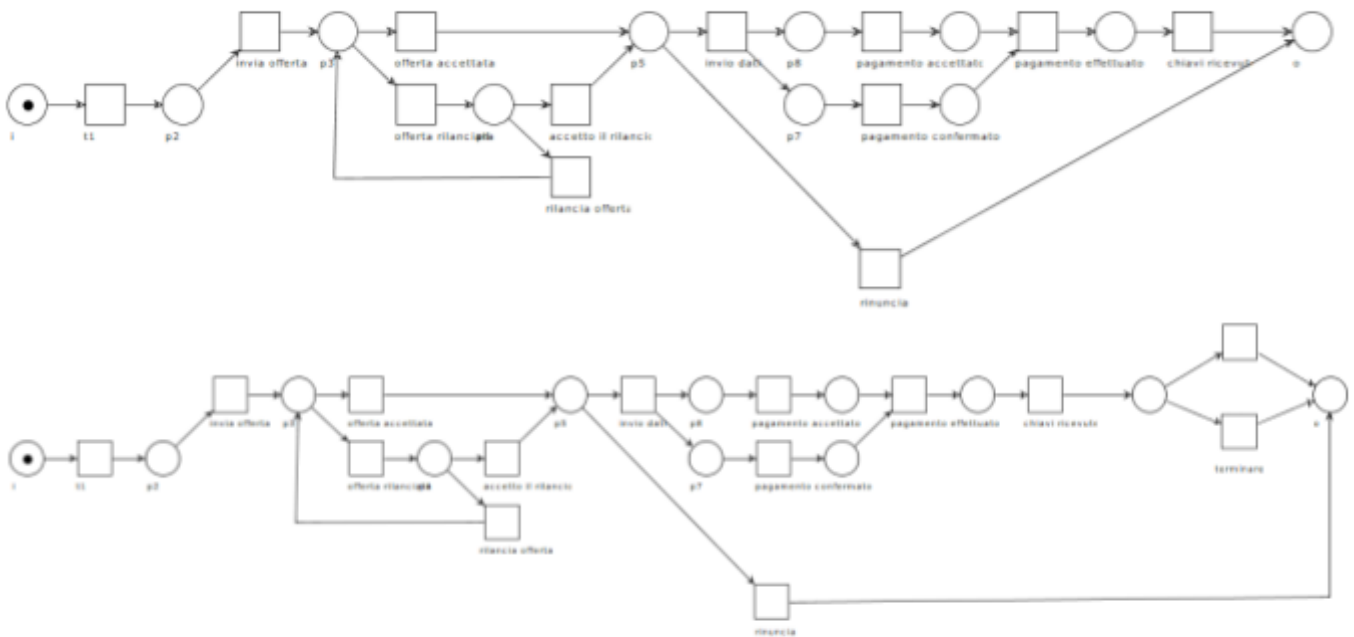


Fig.5: Petri-Net dell'Acquirente con e senza la variante dello scenario

Come si può vedere dalla Fig.5 le reti rappresentano in modo speculare il processo visto su *BPMN*. Per i vari **gateway** sono state utilizzate le espressioni delle *Petri-Net*. Questo lo si può vedere nei seguenti casi:

- **all'inizio** della rete, quando viene inviata l'offerta e l'Acquirente deve sapere se sia stata accettata o rilanciata (*event based gateway*);
- se il Fornitore ha rilanciato l'offerta ed allora l'Acquirente deve decidere se **accettare** il rilancio o **rilanciare** a sua volta con un'altra offerta (*XOR-split*);
- quando l'Acquirente deve decidere se cambiare idea e **rinunciare** (in questo caso va direttamente alla piazza finale) o **proseguire** inviando i dati (*XOR-join/split*);
- un **parallel path** presente quando il pagamento viene sia confermato sia accettato dalla banca (*AND-split + AND-join*);
- nel caso della **variante** dello scenario si ha la possibilità di decidere se negoziare un nuovo servizio o terminare la negoziazione (*implicit XOR*).

Analisi Acquirente

- Entrambe le reti sono **sound**. Questo lo si può vedere sia dal **grafo di raggiungibilità** (11 piazze e 12 transizioni, senza variante, 12 piazze e 14 transizioni, con variante) sia dall'analisi fatta da Woped;
- Sono **S-systems**, dato che ogni transizione ha un solo *pre-set* ed un solo *post-set*;
- Non sono **T-systems**, perché alcune piazze posseggono più *post-set*;
- Sono **Free-choice**, dato che gli *input place* delle transizioni e gli *output transition* delle piazze appartengono agli stessi archi;
- Le reti sono ben strutturate, ossia non presentano alcun tipo di **PT** o **TP handle**;
- Sono **strongly connected**;
- Sono entrambe **bounded** e **live**;
- Sono **deadlockfree**;
- Sono **S-coverable**, entrambe hanno 2 *S-components*;
- Sono **safe**.

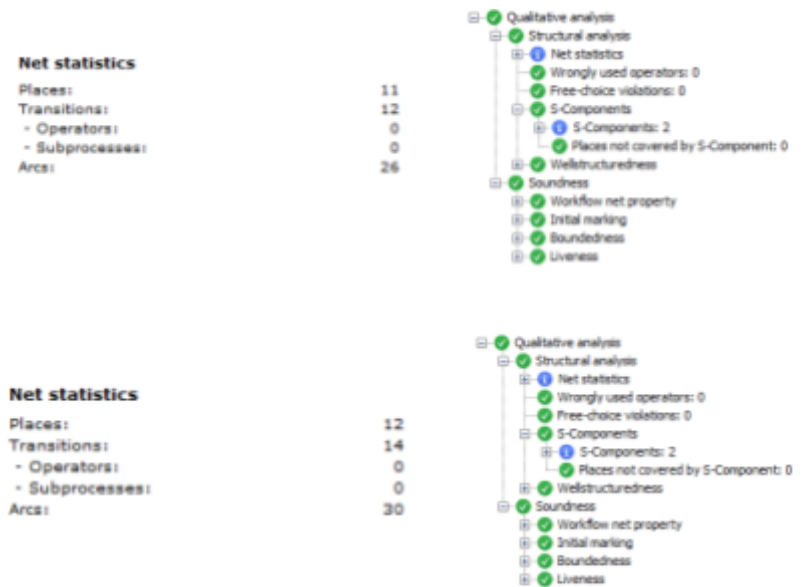


Fig.6: Analisi delle reti dell'Acquirente, (dall'alto) senza variante e con variante

Fornitore

Lo stesso processo di conversione visto precedentemente per la creazione della *Petri-net* dell'Acquirente è stato utilizzato anche per la **rete del Fornitore**, rispettando sempre le proprietà delle *workflow net*.



Fig.7: Petri-Net del Fornitore

Anche qui sono presenti dei **gateway** come nelle reti dell'Acquirente:

- **all'inizio della rete**, quando viene ricevuta l'offerta, il Fornitore può decidere se rilanciare o accettare (*XOR split*);
- se il Fornitore **rilancia**, il *token* ritorna allo *XOR split* iniziale, generando così un *loop*;
- solo se il rilancio è stato accettato dall'Acquirente o dal Fornitore stesso si può chiudere con uno *XOR-join*;
- in seguito alla decisione dell'acquirente di proseguire o rinunciare al pagamento è stato inserito uno *XOR-split*, se la negoziazione è stata annullata termina il processo, altrimenti prosegue;
- nella **parte finale della rete** vediamo che è presente solamente una sequenza di processi che dovranno avvenire in seguito alle decisioni di entrambi gli attori (invio del pagamento e delle chiavi del servizio), terminando con la piazza finale.

Analisi Fornitore

- La rete è **sound**. Questo lo si può vedere sia dal grafo di raggiungibilità (8 piazze e 10 transizioni) sia dall'analisi fatta da Woped;
- È un **S-system**, dato che ogni transizione ha un solo pre-set e post-set;
- Non è un **T-system**, perché alcune piazze posseggono più post-set;
- È **Free-choice**, perché le piazze e le transizioni condividono gli stessi archi;
- La rete è ben strutturata dato che non presenta alcun tipo di **PT** o **TP handle**;
- È **strongly connected**;
- È **bounded** e **live**;
- È **deadlockfree**;
- È **S-coverable**, ha 1 *S-component*;
- È **safe**.



Fig.8: Analisi delle reti del Fornitore

Collaboration diagram in Petri-Net

Adesso che sono stati convertiti in *Petri-Net* i processi di entrambi gli attori possiamo analizzare il diagramma di collaborazione in cui si scambiano le varie informazioni.

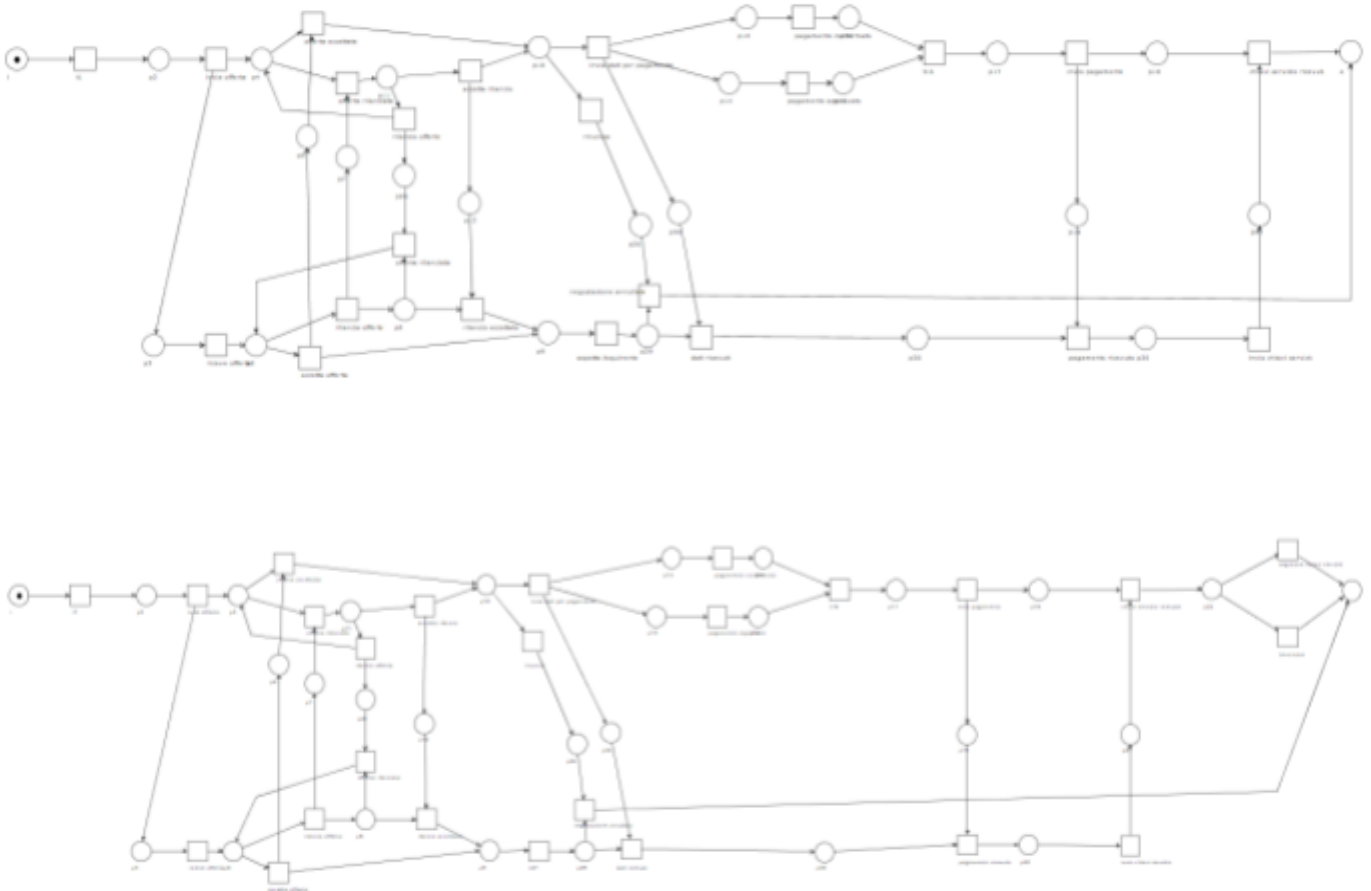


Fig.9: Petri-net senza variante (sopra) e con variante (sotto)

Analisi Negoziazione

- Entrambe le reti sono **sound**. Questo lo si può vedere sia dal **grafo di raggiungibilità** (27 piazze e 23 transizioni, senza variante, 28 piazze e 25 transizioni, con variante) sia dall'analisi fatta da Woped;
- Non sono **S-systems**, dato che alcune transizioni hanno più *pre-set* e *post-set*;
- Non sono **T-systems**, perché alcune piazze posseggono più *post-set*;



Fig.10: Analisi delle proprietà delle reti

- Non sono **Free-choice**, dato che gli input place delle transizioni e gli output transition delle piazze non condividono gli stessi archi;
- Non sono ben strutturate, presentano dei **PT** e **TP handle**;
- Sono **strongly connected**;
- Sono entrambe **bounded** e **live**;
- Sono **deadlockfree**;
- Sono **S-coverable**, entrambe hanno 12 *S-components*;
- Sono safe.

CONCLUSIONI

Analizzando le reti separatamente tutte le proprietà vengono soddisfatte e, come si è visto, unendo le due reti è possibile ottenere una rete sound anche se non rispetta a pieno tutte le proprietà.