



UNIVERSITÀ DI PISA

RELAZIONE PROGETTO DI BUSINESS PROCESS MODELING

p20: Permesso

Biviano Matteo 543933

Anno Accademico 2021/2022

Indice

1	Introduzione	1
2	BPMN	1
2.1	Cittadino	1
2.2	Ufficio permessi	3
2.3	Analisi Processo Complessivo	4
3	Processo Modificato	5
3.1	Analisi processo modificato	6
A	Traduzione da BPMN a Petri Net	8
A.1	XOR split e join	9
A.2	Event-based Exclusive Gateway	9

1 Introduzione

In questo report viene rappresentato lo scenario della richiesta di permesso di parcheggio in una zona a traffico limitato. In particolar modo, vengono modellate le relazioni che intercorrono tra il cittadino e l'ufficio preposto. Il report è articolato come segue: in Sezione [2] viene presentata la modellazione BPMN del problema, la traduzione in Petri net e la sua analisi; in Sezione [3] viene invece presentata la versione modificata richiesta. I passaggi effettuati per tradurre i modelli BPMN in Petri net *desugarized*, sono riportati nell'Appendice [A].

2 BPMN

Per modellare lo scenario richiesto, considerandone la natura collaborativa, è stato ritenuto più utile utilizzare la notazione grafica BPMN (attraverso il programma **BPMN.io**), rispetto al diagramma EPC. Nel problema, infatti, sono presenti due attori (Cittadino e Ufficio Permessi) che comunicano tra di loro tramite *message flows*. Il diagramma dell'intero processo, costituito da un pool per ogni attore, è visibile in Figura [1].

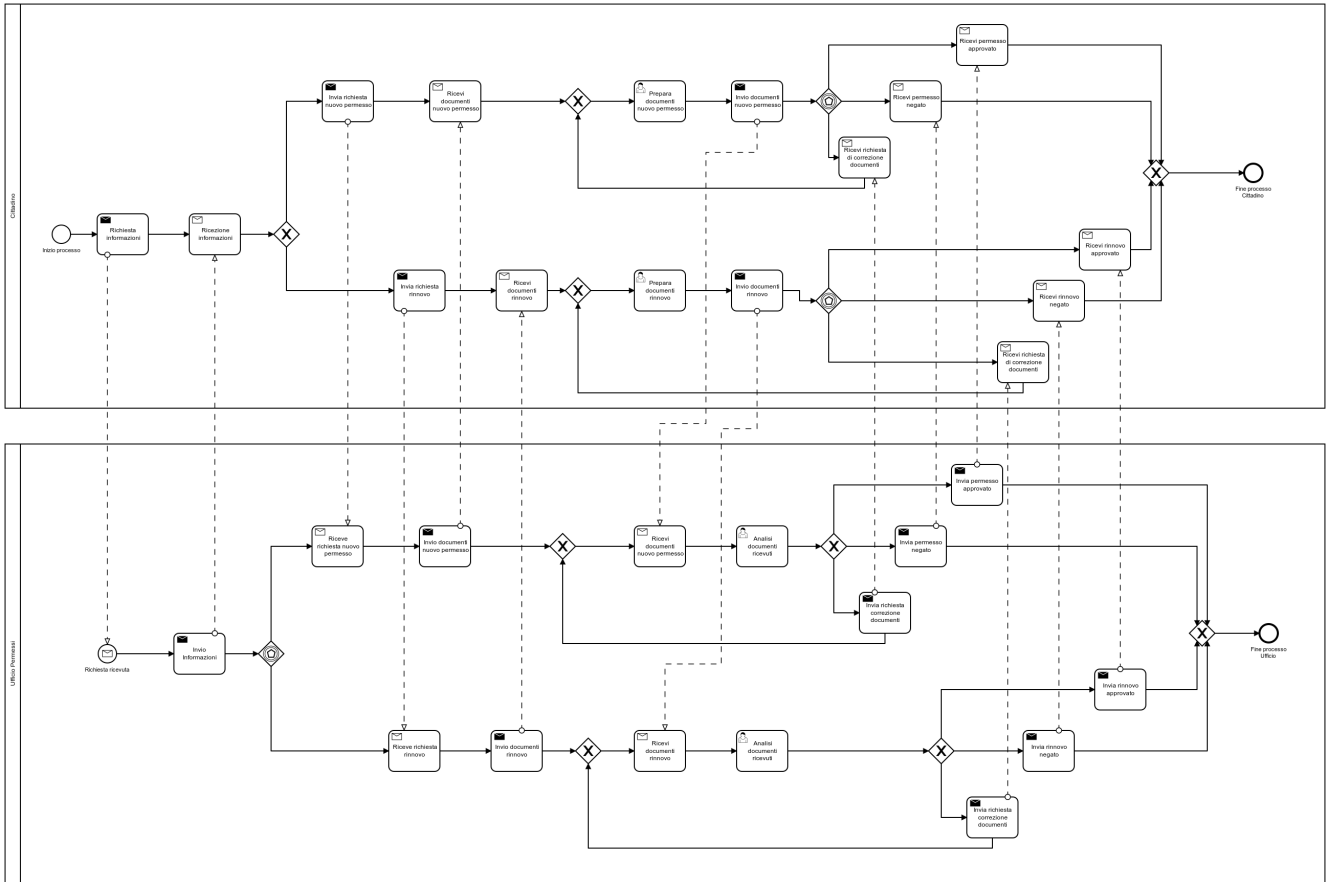


Figura 1: Rappresentazione processo come BPMN

2.1 Cittadino

Il processo inizia con una richiesta di informazioni da parte del **Cittadino** verso l'**Ufficio permessi** (come visibile in Figura [2]), fase comune per entrambi i tipi di richieste (**richiesta di nuovo permesso** e **richiesta di rinnovo**). La richiesta delle documentazione specifica è stata considerata in un momento successivo alla richiesta di informazioni, in quanto i documenti da ricevere e compilare erano differenti per

le due tipologie di attività. Dopo aver ricevuto le informazioni del caso, il cittadino, decide se chiedere un rinnovo o un nuovo permesso (attraverso attività di tipo *send* o *receive*) all'ufficio. La decisione, di tipo esclusivo, è rappresentata nel diagramma da un *exclusive gateway*. La continuazione dei due flussi, dopo aver ricevuto i documenti, risulta praticamente identica, poichè non sono state specificate operazioni che differenziassero le tipologie di documenti e moduli. Il cittadino, dopo aver preparato i documenti ed averli inviati all'ufficio permessi, rimane in attesa del responso dell'ufficio attraverso un *event-based exclusive gateway* collegato a 3 diversi task di ricezione, uno per ogni tipologia di responso (**permesso accettato**, **permesso negato** e **documenti da correggere**). Nel caso di risposta positiva o negativa, il processo si conclude, utilizzando un *exclusive gateway* per poter riunire i 4 casi di fine processo (2 per ogni tipologia di permesso) in un unico output finale. Nel caso di richiesta di correzione di documenti, il flusso ritorna alle fasi precedenti entrando in un ciclo compiuto un numero imprecisato di volte.

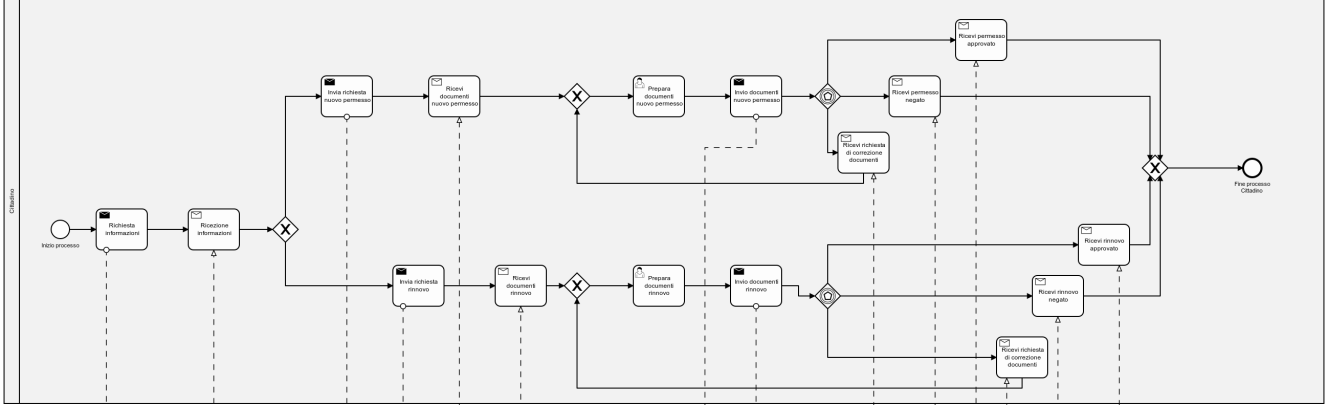


Figura 2: Rappresentazione processo del cittadino

Il BPMN del cittadino è stato tradotto in una rete di workflow (riportata in Figura [3] ed analizzato su **WoPeD** (Figura [4]). La rete risultante ha 12 places, 16 transitions e 32 archi. È **s-coverable** avendo un unico s-component che comprende tutta la rete, ed è **strongly connected**. Avendo un unico s-component, ed essendo i preset e postset delle transizioni composti da un unico place, la rete è una **S-net** (quindi è *safe* e *sound*). Date queste considerazioni sarebbe quindi possibile trovare una S-invariant positiva. La rete non è una **T-net**, in quanto sono presenti piazze con più archi entranti/uscenti. La rete è **free-choice** (essendo una S-net), in quanto ogni coppia di transizioni ha preset equivalenti o disgiunti. La rete (essendo free-choice) è **live** e **bounded**. In particolare, per la proprietà fondamentale delle S-net, risulta essere **1-bounded**, iniziando con un unico token. Non essendo presenti **PT handles** e **TP handles**, la rete è **well-structured**. Non sono presenti **deadlocks** poichè ogni nodo del grafo ha un arco uscente (tranne quello relativo alla fine del processo). Il *coverability graph*, riportato da WoPeD, è finito e contiene 12 vertici e 16 archi.

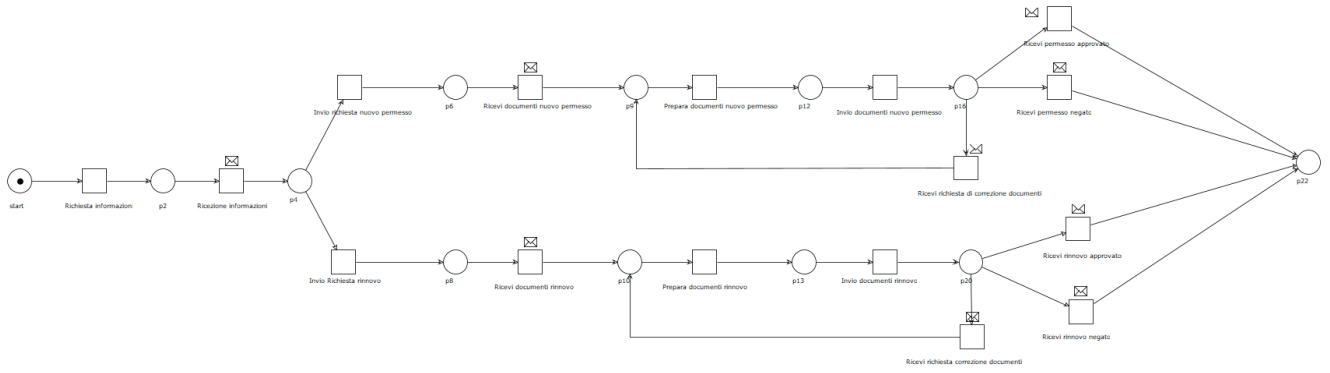


Figura 3: Traduzione Cittadino

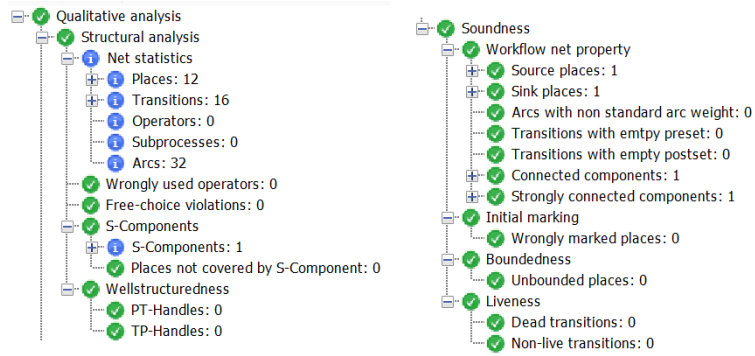


Figura 4: Semantical Analysis - cittadino

2.2 Ufficio permessi

Il pool relativo all'Ufficio permessi (mostrato in Figura [5]) inizia con un *catching message start event* che si attiva quando il cittadino invia la richiesta di informazioni all'ufficio. Questa fase, insieme all'invio di informazioni, è comune ai due tipi di permessi. L'ufficio, attraverso un *event-based exclusive gateway*, attende la richiesta del cittadino dividendo il processo in due flussi, uno per ogni tipo di richiesta. In entrambi i casi, una volta ricevuta la richiesta, viene inviata al cittadino la documentazione specifica per quella richiesta. Una volta ricevuti i documenti compilati, vengono analizzati ed emanato un responso (attraverso un *exclusive gateway* collegato a 3 *send task*) nel quale si accetta la richiesta, si nega la richiesta o si richiede la correzione dei documenti. In quest'ultimo caso, dopo aver inviato la richiesta di correzione, il flusso si ricongiunge alle fasi precedenti in attesa di un nuovo invio.

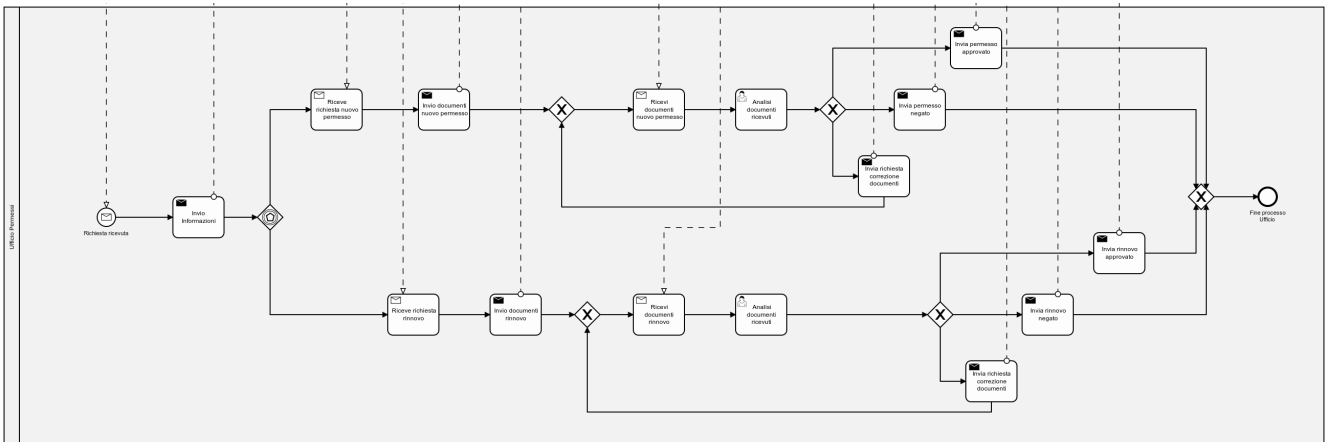


Figura 5: Rappresentazione processo dell'ufficio permessi

La traduzione in Petri net dell'attore **Ufficio Permessi** è riportata in Figura [6]. Per la tipologia di processo rappresentato, la rete risultante ha le stesse caratteristiche di quella riportata in Sezione [2.1] per l'attore **Cittadino**, avendo infatti stesso numero di piazze, transizioni, archi e stessa collocazione degli stessi nella rete risultante.

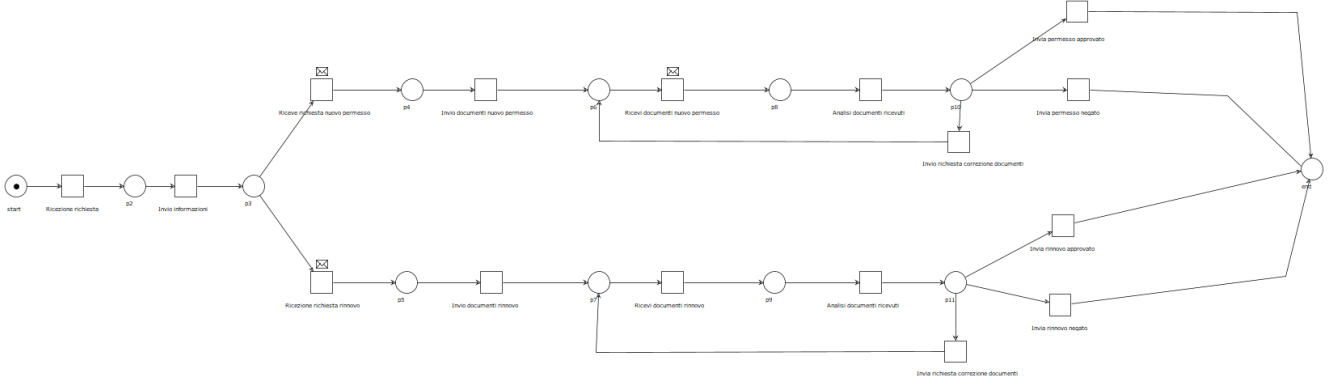


Figura 6: Traduzione Ufficio

2.3 Analisi Processo Complessivo

La traduzione da BPMN a Petri Net dell'intero sistema è mostrata in Figura [7] ed è il risultato della combinazione dei due workflow modules descritti nelle sezioni precedenti, attraverso le piazze di interfaccia che permettono la comunicazione e lo scambio di messaggi tra essi. Queste piazze di interfaccia sono collegate attraverso archi alle transizioni corrispondenti in modo tale da tradurre le attività di ricezioni ed invio inserite nel BPMN (in Sezione [2]). Per garantire un'unica piazza iniziale è stato considerato come punto di inizio quello della Petri net del Cittadino, mentre, per garantire un'unica piazza finale è stata aggiunta una transizione finale che rappresenta un AND join, il cui preset è formato dalle piazze finali dei singoli processi a cui è collegata una piazza finale priva di archi uscenti. Questo poichè il sistema si conclude quando entrambi i processi terminano.

La rete risultante ha 38 piazze, 33 transizioni e 94 archi. Come visibile anche dalle analisi riportate in Figura [8] la rete risulta essere **sound**, tuttavia si riscontrano delle violazioni delle proprietà *free-choice*, infatti, creando i collegamenti tra i pool ci sono transizioni che condividono piazze di input. Inoltre, sono presenti 46 *PT-handles* e 52 *TP-handles*, non essendo quindi *well-structured*. La rete risulta, inoltre, **bounded**, **strongly-connected** e **live** (perciò è **Place-Live** e **Deadlock-Free**). Infine, la rete risulta **S-coverable** poichè tutte le piazze sono coperte da **S-component**. Il Coverability graph, che coincide con il Reachability graph dato che la rete è bounded, presenta 29 vertici e 33 archi.

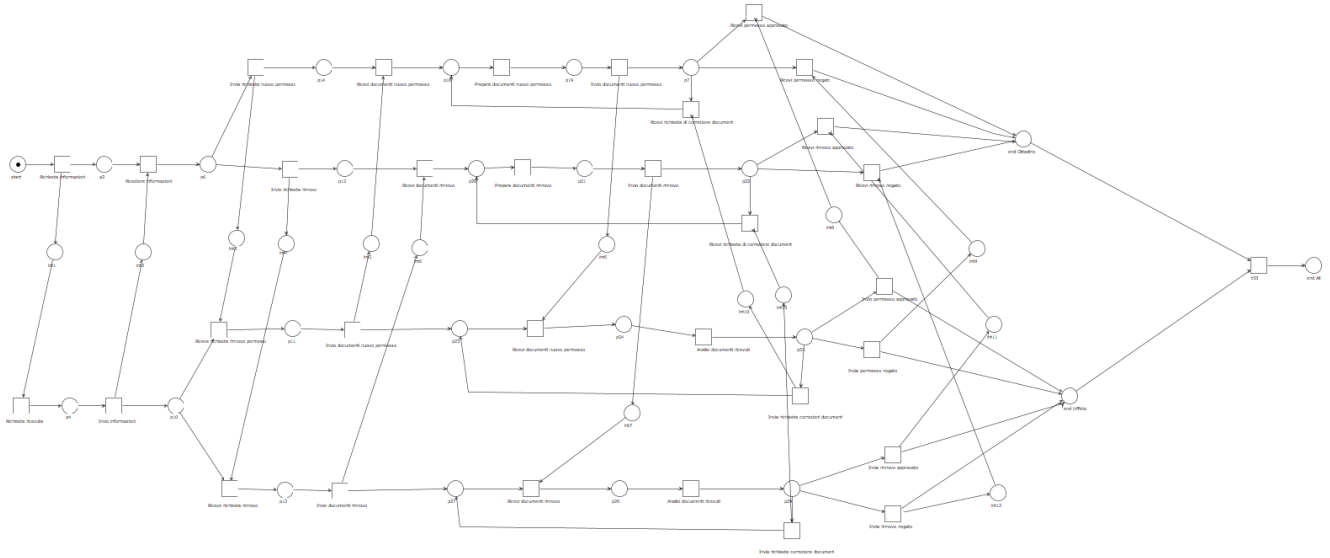


Figura 7: Traduzione workflow net

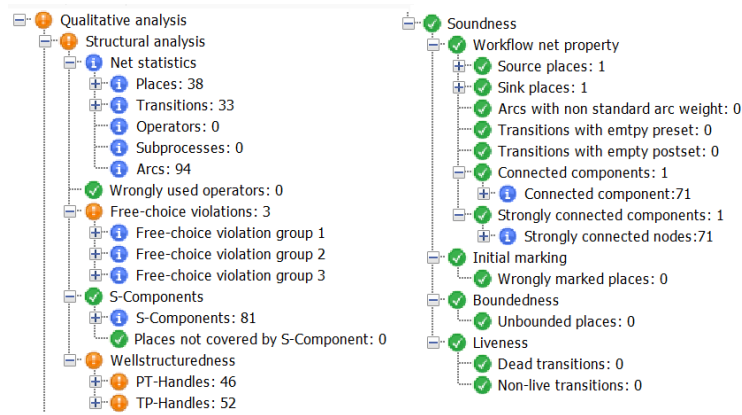
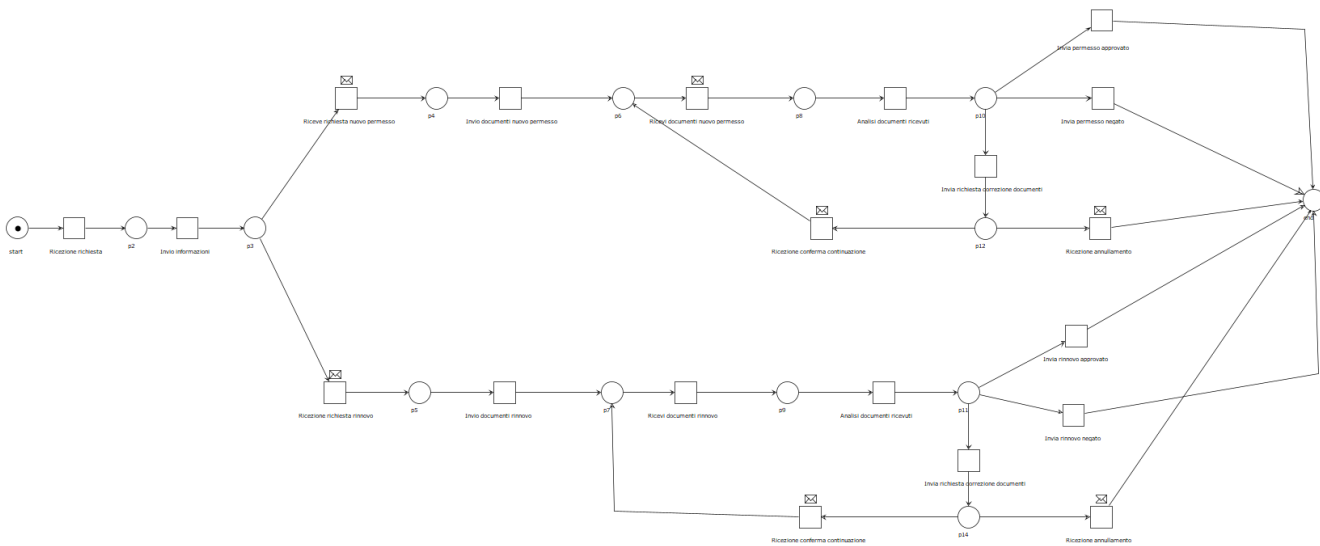
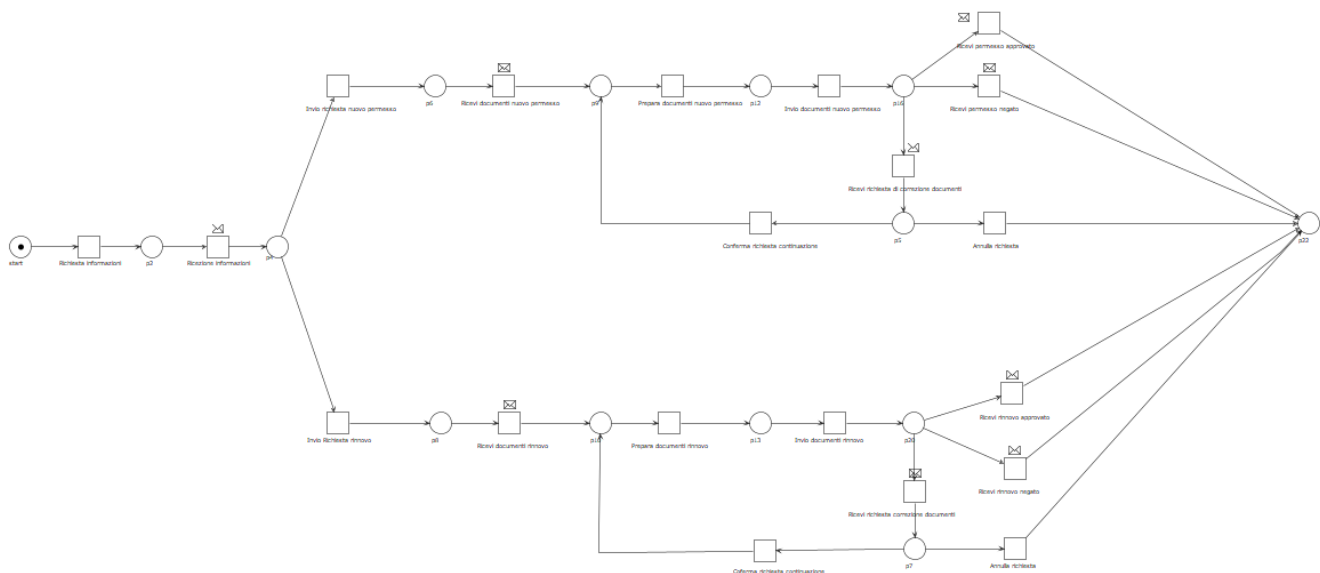


Figura 8: Semantical Analysis - Workflow net

3 Processo Modificato

La consegna del progetto prevede una modifica al processo, nella quale il cittadino ha la possibilità di annullare la procedura in caso di ricezione della richiesta di correzione dei documenti errati. Il diagramma BPMN riportato in Sezione [2] è stato opportunamente modificato, come riportato in Figura [9], aggiungendo nel pool del cittadino, dopo la ricezione della richiesta di correzione della documentazione, le attività *Annulla richiesta* e **Conferma richiesta continuazione**, rispettivamente per annullare la procedura (e terminare il processo) o per tornare nel loop, riconsegnando i documenti dopo averli corretti. Questo passaggio (aggiunto per ogni tipologia di richiesta) è stato modellato nel diagramma attraverso un *exclusive gateway* essendo due attività esclusive. Nel caso in cui il cittadino scelga di continuare la procedura, notifica infatti la sua decisione all'ufficio, il quale attenderà (all'interno del loop) la ricezione della documentazione corretta. Nel pool dell'ufficio è stato quindi aggiunto, dopo l'invio della richiesta di correzione dei documenti, un *event-based exclusive gateway* per poter attendere la decisione del cittadino e continuare il processo.



La traduzione della rete complessiva, la quale riporta 46 piazze, 41 transizioni e 118 archi, è riportata in Figura [12]. Dall’analisi (Figura [13]), la rete risultate, come la versione non originale (Sezione [2.3]), risulta essere **sound**, ma si riscontrano 5 violazioni della proprietà free-choice. Come in precedenza vi sono TP-handles e PT-handles (64 e 90 rispettivamente). La rete risulta però **s-coverable**, quindi è possibile trovare una **positive s-invariant**.

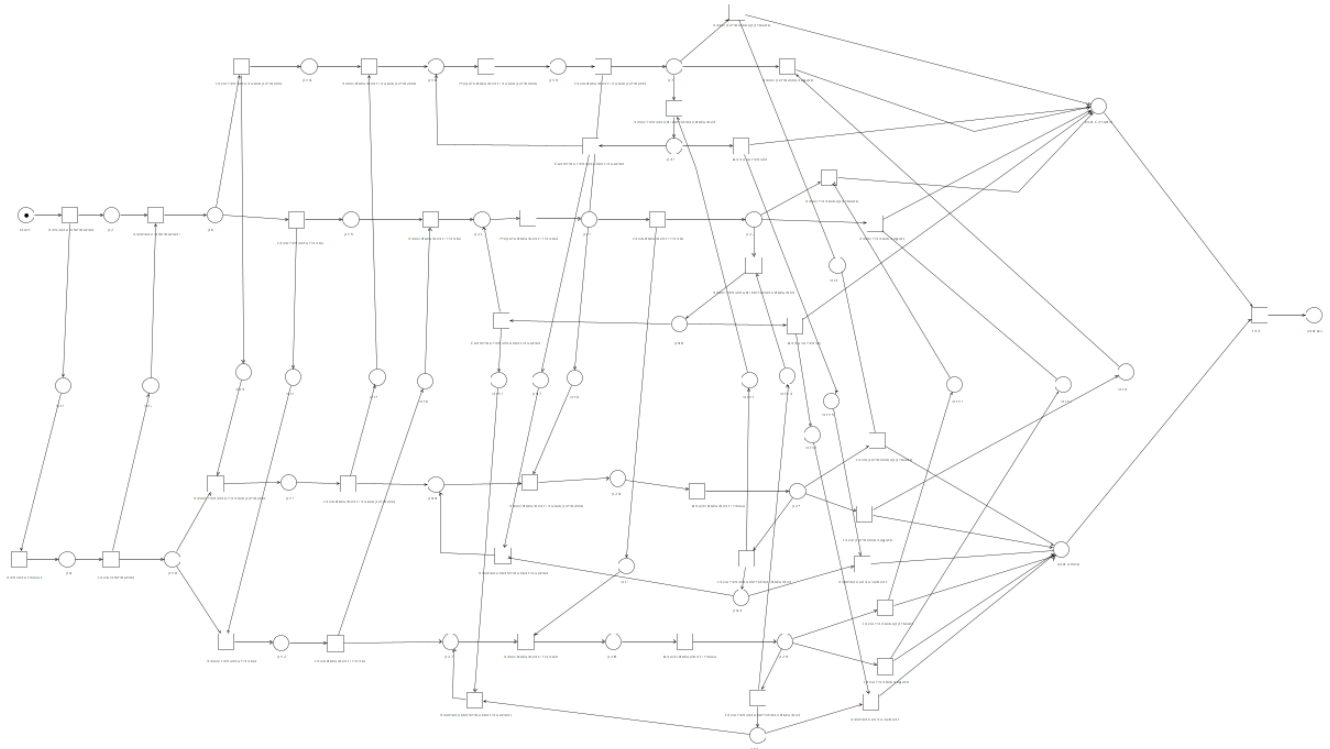


Figura 12: Traduzione workflow net variante

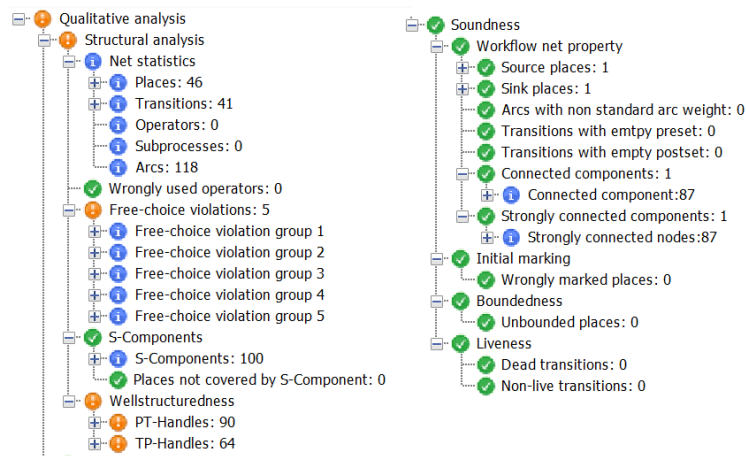


Figura 13: Semantical Analysis - Workflow net variante

A Traduzione da BPMN a Petri Net

I passaggi principali effettuati per tradurre i modelli BPMN in Petri Net sono stati i seguenti:

- inserire una piazza per ciascun arco;
- sostituzione di attività ed eventi in transizioni;
- traduzione dei gateway;
- ottimizzazione della rete.

In particolare, la traduzione dei gateway, con conseguente ottimizzazione della rete viene riportata nelle sezioni successive.

A.1 XOR split e join

Lo XOR split (Figura [14](a)), utilizzato per modellare i casi in cui gli attori dovevano effettuare la scelta, viene tradotto inserendo una piazza seguita da tante transizioni quante sono le scelte, collegando queste ad un'altra piazza che rappresenta l'input per la transizione che identifica l'attività da svolgere, come mostrato in Figura [14] (b). È stato possibile semplificare la struttura della rete, racchiudendo sequenze come $t_4, p_5, \text{Invio richiesta rinnovo}$, costituenti building blocks (safe e sound) in un'unica transizione (Figura [14] (c)).

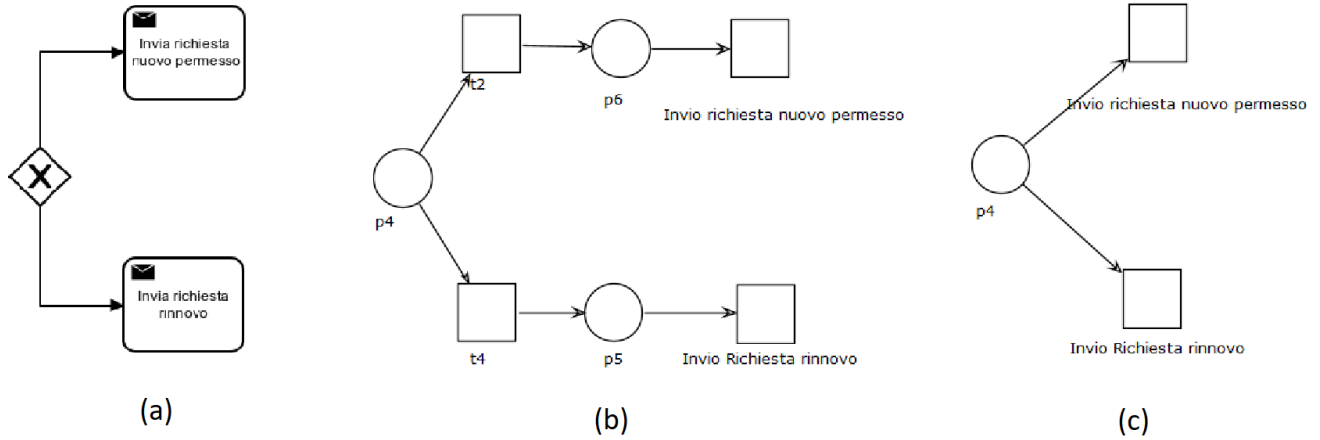


Figura 14: XOR Split - Traduzione e ottimizzazione

Analogamente a quanto fatto per lo XOR split, lo XOR join verrebbe tradotto inserendo delle piazze come output delle transizioni rappresentanti le attività coinvolte nel join, seguiti da due transizioni che convergono in un'unica piazza. Tuttavia, anche in questo caso il frammento di rete è stato semplificato, utilizzando i building blocks, quindi sostituendo le sequenze *transizione, piazza, transizione* come un'unica transizione.

A.2 Event-based Exclusive Gateway

Per tradurre l'*event-based exclusive gateway* in Petri net è stata utilizzata un'unica piazza di input, la quale avrà nel postset un numero di transizioni pari al numero di alternative possibili. La differenza con lo XOR split è che la decisione dell'alternativa, in questo caso, dipende dall'esterno, poichè ogni transizione avrà nel suo preset anche una piazza dell'interfaccia, utilizzata (come riportato in Sezione [2.3]) per ricevere la comunicazione dell'altro attore.