



UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in Data Science and Business Informatics

Business Process Modeling **Relazione Progetto 47: Commissione**

Ilaria Ritelli - 579271

A.A. 2024/2025

1 Introduzione

L'obiettivo di questo progetto è modellare processi che rappresentano uno scenario in cui un candidato sottopone il resoconto di un proprio lavoro a una commissione di tre membri. Uno dei membri è il presidente, che si occupa di coordinare la valutazione della commissione. Al ricevimento del resoconto, ciascun membro della commissione esprime un parere (positivo o negativo), oppure richiede ulteriori chiarimenti. Se uno o più membri richiedono chiarimenti, il candidato deve aggiornare il resoconto e reinviarlo, ripetendo la fase di valutazione della commissione. Se tutti i membri esprimono un parere, il presidente propone una valutazione complessiva, la quale può essere approvata o respinta dagli altri due membri. Se uno o entrambi i membri rifiutano la valutazione finale, il presidente elabora una nuova proposta e la sottopone alla commissione. Se invece entrambi i membri approvano la valutazione, questa viene comunicata al candidato e il processo si conclude.

È stata inoltre modellata una variante del processo che prevede un'opzione aggiuntiva per il candidato. In caso di valutazione negativa, il candidato può scegliere se accettarla, terminando così il processo, oppure presentare un nuovo resoconto per una nuova valutazione da parte della commissione. Nel caso in cui il candidato opti per il rifiuto, i task legati al processo di valutazione vengono ripetuti.

Questa relazione descrive come sono stati modellati i processi e le caratteristiche delle workflow net corrispondenti. Poiché sono coinvolti più attori, è stata scelta la rappresentazione grafica Business Process Model and Notation (BPMN), che permette di specificare sia *orchestration* che *collaboration* tra processi. I diagrammi BPMN sono stati progettati con il tool online bpmn.io; le reti di Petri sono state realizzate e analizzate con WoPeD.

2 Diagrammi BPMN

Nello scenario è possibile identificare due attori distinti: il candidato e la commissione. Sono stati creati due diagrammi di *orchestration*, i quali comunicano tra di loro in un diagramma di *collaboration*.

2.1 Candidato

Come da specifiche, il candidato scrive un resoconto (“Produci resoconto”) e lo invia alla commissione (“Invia resoconto”) con un *send task*. Rimane quindi in attesa di ricevere un riscontro dalla commissione: se essa chiede ulteriori chiarimenti, il resoconto viene aggiornato e reinviato (“Aggiorna resoconto” e “Invia resoconto aggiornato”), quindi il candidato ritorna in attesa; altrimenti, il candidato riceve una valutazione finale e il processo termina. Poiché la scelta dipende da un evento esterno, questa situazione viene modellata con un *event-based gateway*, collegando tramite archi uscenti da esso due *message event*: “Valutazione finale ricevuta” e “Richiesta di chiarimenti ricevuta”. Se il candidato deve modificare il resoconto, il flusso procede nel ramo corrispondente, il quale ritorna nell’arco entrante dell’*event-based gateway* tramite un join *XOR*. Se altrimenti riceve la valutazione, il processo termina. Il diagramma del candidato è mostrato in Fig. 1.

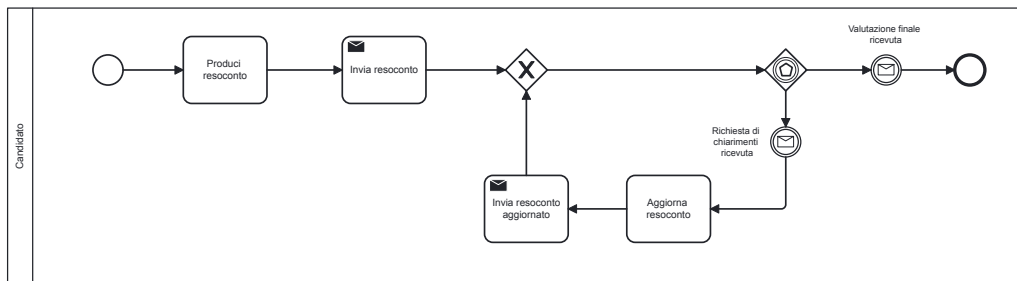


Figura 1: Diagramma BPMN del candidato.

2.2 Commissione

La commissione è stata modellata come una *pool* con tre *lane*, una per membro. Il processo comincia nel momento in cui il presidente riceve il resoconto con un *message event* iniziale (“Resoconto ricevuto”). Il flusso viene propagato ai membri con uno split *AND*, e ciascuno legge il lavoro ed esprime un parere (“Esprimi parere positivo”, “Esprimi parere negativo” o “Richiedi chiarimenti”). La scelta fatta dal membro viene modellata con uno split *XOR* con tre archi uscenti collegati ai rispettivi task, e

un join *XOR* che riunisce gli archi. I flussi delle tre *lane* vengono riuniti con un join *AND*, seguito da uno split *XOR* che permette di richiedere chiarimenti al candidato oppure procedere con la valutazione finale. Se sono richiesti chiarimenti, il flusso ritorna all’inizio, in un join *XOR* posto subito dopo lo *start event*.

Quando il presidente propone una valutazione, essa viene inviata agli altri due membri. Anche in questo caso viene usato uno split *AND* per duplicare il flusso, e uno split *XOR* per permettere ai membri di esprimere approvazione o disapprovazione. I flussi ritornano al presidente, ricongiunti con un join *AND*, ed egli verifica che entrambi i membri abbiano approvato la sua valutazione, con uno split *XOR*. In caso negativo, il flusso procede nell’arco che porta a ripetere il task “Proponi valutazione”, altrimenti procede nell’arco che porta a comunicare la valutazione con un *send task*, e infine all’evento finale. Il diagramma è mostrato in Fig 2.

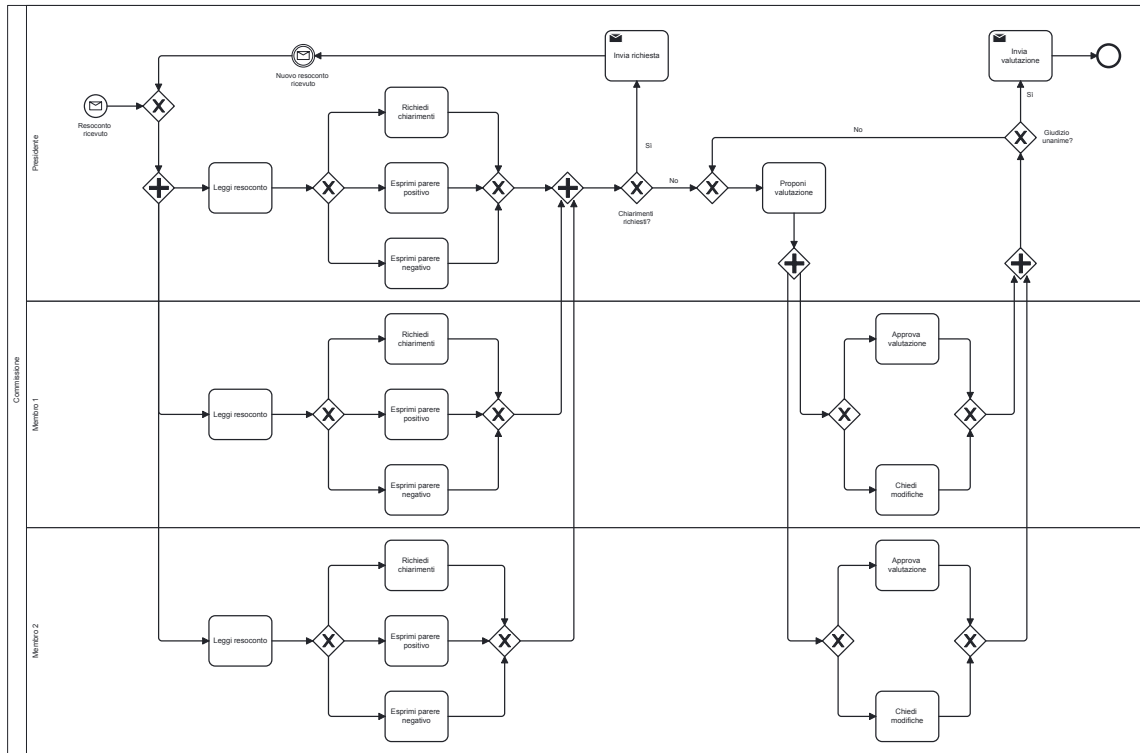


Figura 2: Diagramma BPMN della commissione.

2.3 Collaboration

Il diagramma di *collaboration* è composto dai diagrammi dei singoli attori, collegati tra loro tramite *message flows* tra i *send task/event* e *receive task/event* corrispondenti. Il diagramma è mostrato in Fig. 3.

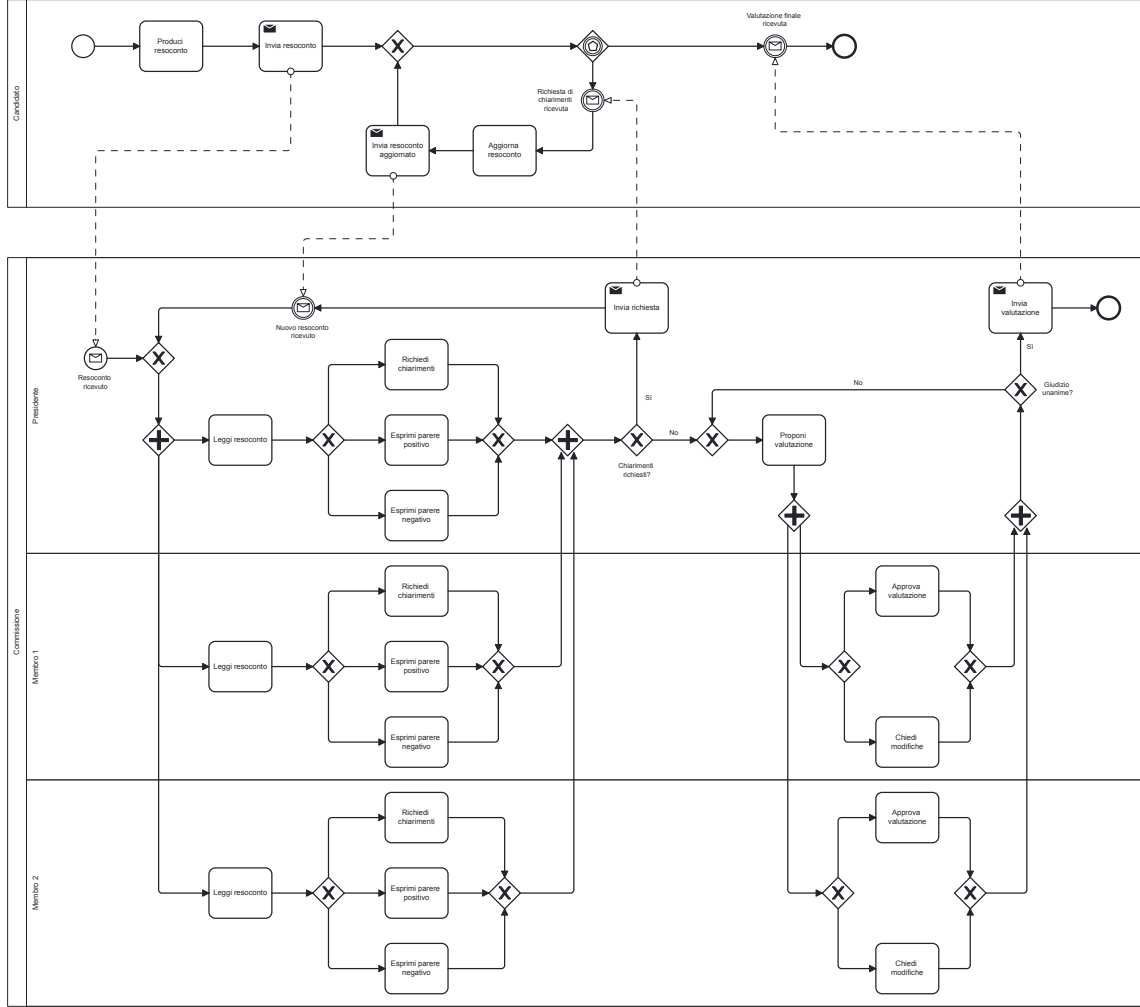


Figura 3: Diagramma BPMN della collaborazione tra candidato e commissione.

2.4 Variante

Per adattare i processi alla variante richiesta, sono stati modificati i diagrammi di entrambi. Nel diagramma del candidato sono stati aggiunti due split *XOR*; il primo per distinguere tra il caso in cui la valutazione è positiva (e quindi viene accettata automaticamente) e il caso in cui è negativa, e il secondo per consentire al candidato di decidere se accettarla comunque o inviare un nuovo resoconto. Se sceglie di rifiutare, notifica la commissione con un *message event* (“Rifiuto inviato”) e il flusso raggiunge il task “Aggiorna resoconto” con un join *XOR*. Se invece accetta, viene inviata la conferma alla commissione con un apposito *message event* (“Conferma inviata”), e il processo termina.

Nel diagramma della commissione è stato aggiunto un *event-based gateway* subito dopo il task “Invia valutazione”, così che il presidente rimanga in attesa della scelta

del candidato. Se egli rifiuta la valutazione, allora la commissione attende l'arrivo di un nuovo resoconto. È stato aggiunto un join *XOR* che ha come archi entranti quello uscente dal *message event* “Rifiuto ricevuto” e quello uscente da “Invia richiesta”, usata per chiedere ulteriori chiarimenti. Quando il candidato risponde inviando la nuova versione del suo lavoro vengono quindi ripetuti gli stessi task che la commissione svolge per valutarlo. Se il candidato invece accetta, la sua conferma viene ricevuta con l'evento “Conferma ricevuta”, e il processo termina. Il diagramma della variante è mostrato in Fig. 4.

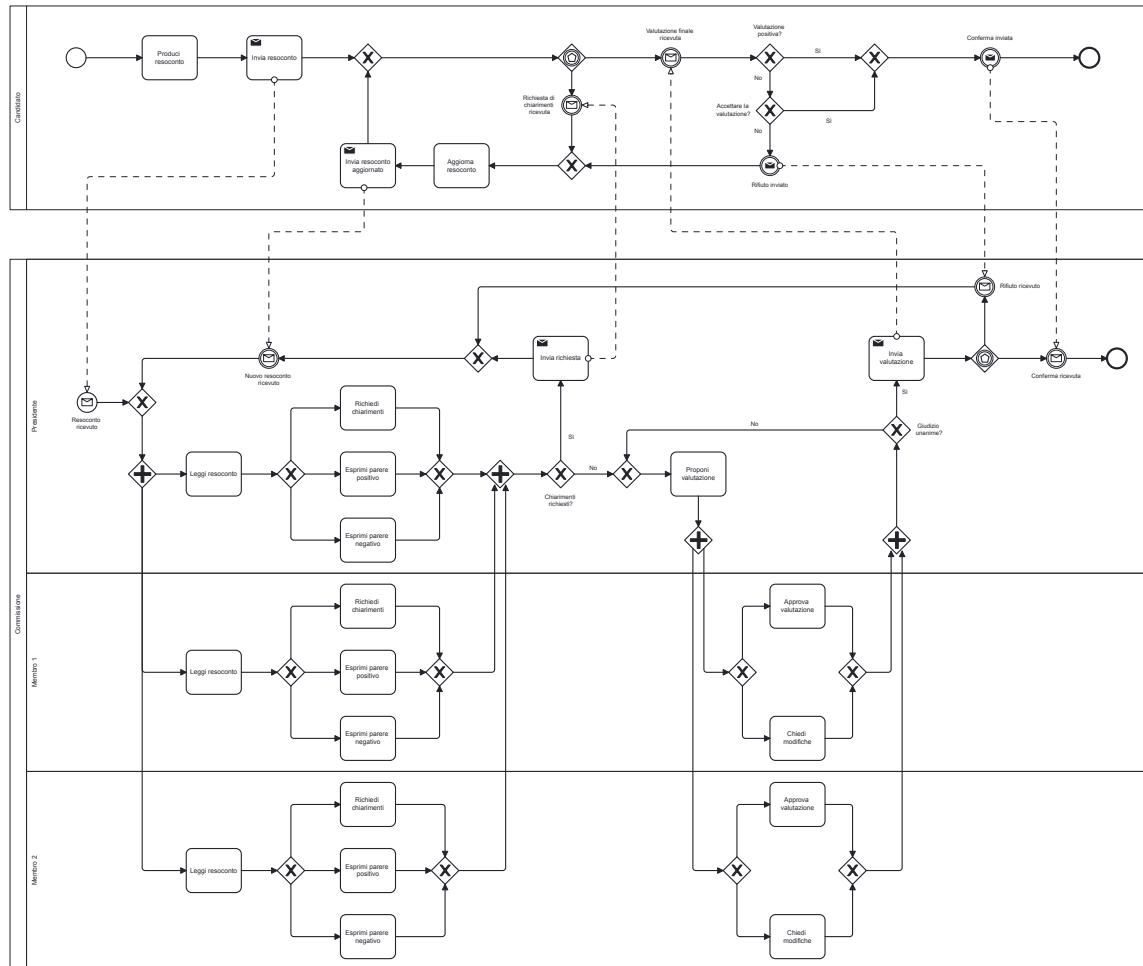


Figura 4: Diagramma BPMN della collaborazione tra candidato e commissione (variante).

3 Workflow Net

I diagrammi BPMN sono stati tradotti in workflow net, in modo da analizzarli formalmente e verificare le loro proprietà. Ciascun diagramma di *orchestration* è stato trasformato in un workflow module e analizzato individualmente. I due module di ogni versione sono stati poi collegati in un workflow system, definendo un'apposita interfaccia.

I passi principali seguiti per la traduzione sono i seguenti:

- Ogni task ed evento è stato trasformato in una transizione, e ogni arco di flusso di sequenza è stato trasformato in un posto.
- Split e join *AND/XOR* sono stati inizialmente trasformati in transizioni annotate con il tipo corretto, e successivamente sono stati sostituiti con la corretta combinazione di posti e transizioni rimuovendo lo “zucchero sintattico” delle annotazioni.
- Ove opportuno, per rendere più leggibile la rete, sono stati rimossi posti e transizioni aggiunte nel passo precedente, assicurandosi che la rete rimanesse equivalente.
- Split *event-based* sono stati direttamente trasformati in posti.

3.1 Candidato

La rete di Petri del candidato è mostrata in Fig. 5. La rete ha 10 posti, 10 transizioni e 20 archi. Rispetta le proprietà di workflow net: esiste un solo posto iniziale i , $\bullet i = \emptyset$ e un solo posto finale o , $o \bullet = \emptyset$, e ogni altro posto o transizione giace su un cammino da i a o . La rete è free-choice e well-structured (non ha PT-handles o TP-handles). Inoltre è un S-system, quindi è safe e sound, e un qualsiasi vettore $I = [k, \dots, k] \in \mathbb{Q}^{10}$ è S-invariante della workflow net. Poiché la rete è sound, allora N^* è bounded e live, e quindi ha un T-invariante positivo. N^* non è un T-system.

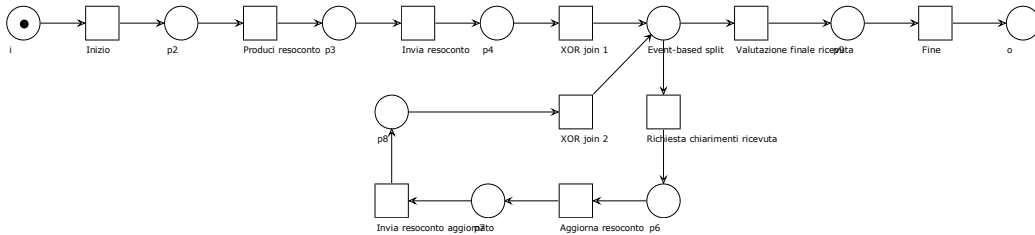


Figura 5: Workflow net del diagramma Candidato.

Il coverability graph ha 10 vertici e 10 archi. La rete è safe (1-bounded), quindi il suo reachability graph è finito e coincide con il coverability graph.

3.2 Commissione

La rete è mostrata in Fig. 6. La rete ha 25 posti, 31 transizioni e 68 archi. È una workflow net, è sound, è free-choice, ed è well-structured. Non è un S-system, e N^* non è un T-system. Siccome la rete è sound, N^* è bounded e live, quindi ha un T-invariante positivo, ed è S-coverable, quindi ha un S-invariante positivo. Sono presenti 6 S-components diverse, quindi un S-invariante è ottenibile sommando gli invarianti indotti dalle S-component.

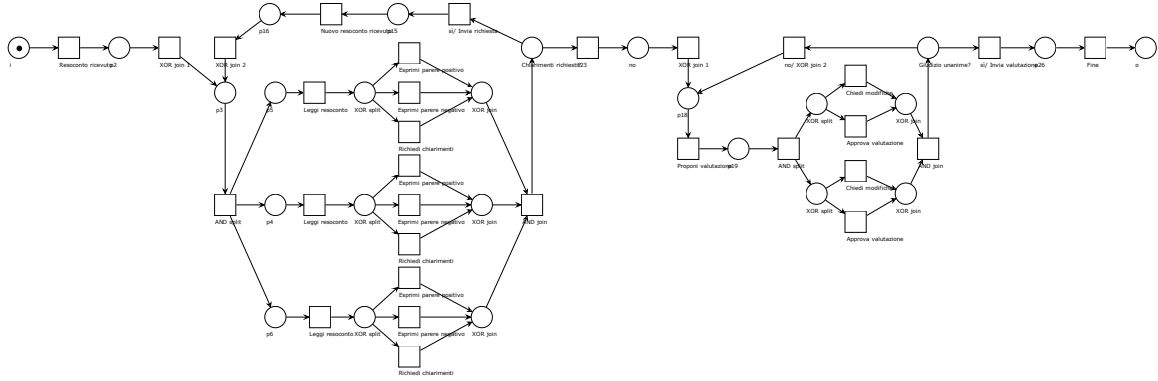


Figura 6: Workflow net del diagramma Commissione.

Il coverability graph ha 43 vertici e 73 archi. Come la precedente, la rete è bounded, quindi il reachability graph è finito e coincide con il coverability graph.

3.3 Workflow System

Il workflow system è mostrato in Fig. 7. Sono stati aggiunti 4 posti per definire l'interfaccia tra i due moduli. Inoltre, sono stati aggiunti un posto iniziale e un posto finale univoci, collegati rispettivamente ai posti iniziali e finali dei due moduli con una transizione. In totale, il sistema ha 41 posti, 43 transizioni e 102 archi. È una workflow net ed è sound, ma a differenza dei suoi moduli, non è free-choice e non è well-structured. È presente una *free-choice violation* dovuta alla presenza dell'*event-based split* nel modulo Candidato, le cui transizioni nel post-set hanno pre-set disgiunti. Sono presenti 8 PT-handles e 2 TP-handles. N^* è S-coverable, con 49 S-components distinte. È bounded e live, quindi ha un T-invariante positivo.

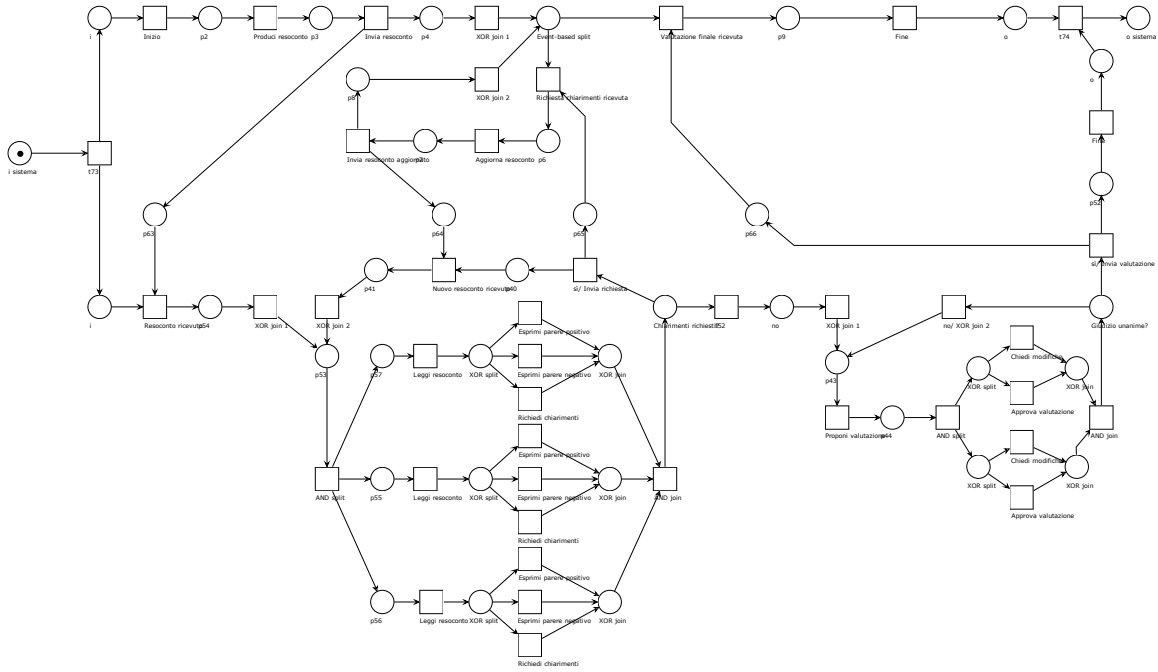


Figura 7: Workflow system.

Il coverability graph ha 139 vertici e 313 archi. Il reachability graph è finito e coincide con il coverability graph, poiché la rete è bounded.

3.4 Variante

Le workflow net delle varianti dei diagrammi sono state ottenute modificando le reti precedenti. La variante della rete del candidato ha 16 posti, 18 transizioni e 36 archi; quella della commissione ha 27 posti, 34 transizioni e 74 archi. Mantengono le stesse proprietà osservate nelle rispettive versioni precedenti delle workflow net. Il coverability graph del primo ha 16 vertici e 18 archi, mentre il secondo ha 45 vertici e 76 archi. In entrambi i casi, il reachability graph rimane finito e coincide con il coverability graph.

Fig. 8 mostra il workflow system della variante. La rete ha 51 posti (di cui 6 parte dell'interfaccia, e 2 aggiunti come posto iniziale/finale), 54 transizioni e 128 archi. Anche in questo caso la rete è sound ma non è free-choice e non è well-structured. Sono infatti presenti due *free-choice violations*, una dovuta all'*event-based split* nel modulo Candidato (già presente nell'altra rete), e l'altra all'*event-based split* nel modulo Commissione aggiunto nella variante. La rete ha 30 PT-handles e 24 TP-handles. È S-coverable, con 62 S-components. Il coverability graph ha 152 vertici e 332 archi, e siccome la rete è bounded, il reachability graph è finito e coincide con il coverability graph.

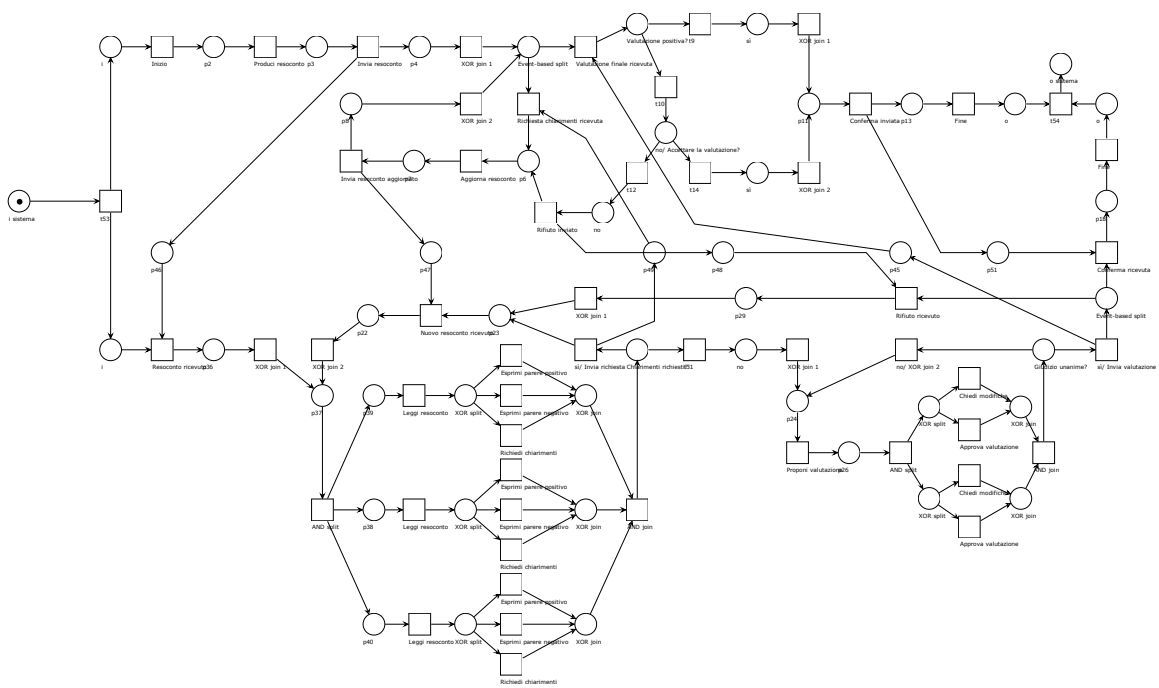


Figura 8: Workflow system della variante.