



Project Report
BUSINESS PROCESS MODELING
Data Science and Business Informatics

Mario Proia	616679
Giada Traina	616682

Anno Accademico 2021/2022

ABSTRACT

Lo scenario presentato riguarda la creazione di un business process relativo ad uno stagista ed al suo periodo di prova presso un'azienda. Prima di iniziare l'analisi, è importante identificare tutti gli attori del processo, che dovranno interagire tra di loro al fine di portare a termine le rispettive attività. Gli attori identificati sono:

- Stagista
- Supervisore
- Project Leader

Il processo di modellazione è stato realizzato usando la **notazione BPMN**, mentre le **Petri Nets** hanno consentito di svolgere un'analisi sulla struttura e sulla soundness della rete.

Il diagramma BPMN è stato creato con **Signavio**, mentre l'analisi relativa alle Petri Nets è stata eseguita tramite **WoPed** e **Woflan**.

Per quanto riguarda le scelte relative al design del processo, poiché l'interazione tra i vari attori rappresenta un elemento cruciale, è stato deciso di costruire un **diagramma di collaborazione**. Come sarà possibile vedere nelle pagine seguenti, sono state create **tre pools** (una per ogni attore) nelle quali saranno analizzate le varie attività portate a termine dagli attori.

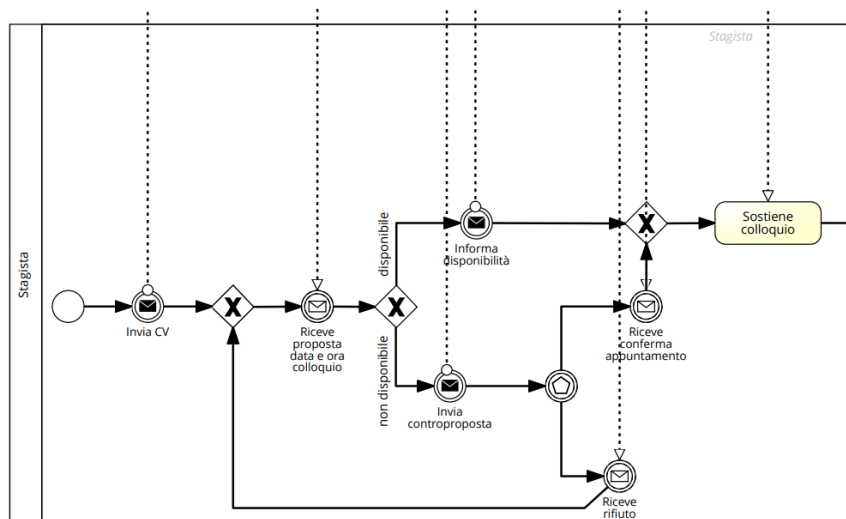
Per rappresentare le varie interazioni fra gli attori sono state utilizzate delle **frecce di message flow**. Inoltre, è stato utilizzato un artefatto, che rappresenta il **database dei dipendenti**, dove è possibile trovare ed aggiornare le informazioni relative a questi, una volta esaminata la scheda di valutazione. Successivamente, sono stati inseriti anche dei **gateway basati sulle decisioni**, i quali mettono in evidenza come le varie attività delle rispettive pool siano fortemente collegate fra loro. Oltre a questi gateways sono stati inseriti degli XOR split e XOR join.

Infine, poiché le attività e fasi relative al colloquio non sono state definite in modo specifico nella consegna, è stata predisposta l'assunzione secondo cui lo stagista abbia inviato il curriculum prima di sottoporsi al colloquio e, successivamente, siano stati identificati dei passaggi per definire come trovare un punto di incontro tra lo stagista e il supervisore, al fine di fissare data e ora del colloquio.

MODELLO BPMN

1. Stagista

La pool principale è quella dello Stagista, che rappresenta il punto di inizio dell'intero processo di business. Il processo inizia con l'invio, da parte di quest'ultimo, del curriculum al Supervisore. In seguito, lo stagista riceverà una notifica relativa alla data e ora proposta dal Supervisore; per poi, scegliere di accettare o inviare una controproposta. Quest'ultima condurrà ad un **event based gateway**, durante il quale lo Stagista attenderà una risposta dal Supervisore. Se la controproposta verrà rifiutata, lo Stagista attenderà un'altra proposta dal Supervisore (tramite uno **XOR join** si ritorna ad uno step precedente); in caso contrario, riceverà una conferma dell'appuntamento per poi sostenere il colloquio (ciò viene rappresentato tramite l'utilizzo di uno **XOR join** prima dell'attività "Sostiene Colloquio").



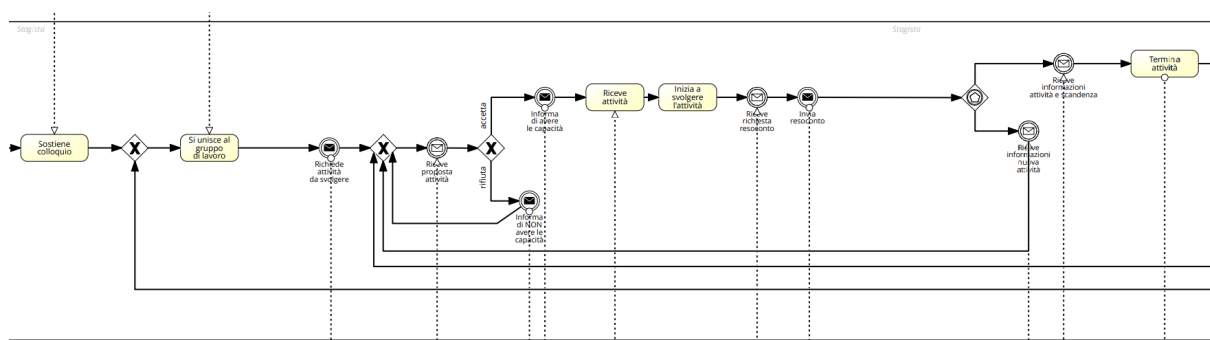
Una volta terminato il colloquio, lo stagista si unirà ad un gruppo di lavoro, per poi richiedere al Project Leader quale attività svolgere all'interno del gruppo.

Successivamente, dopo aver ricevuto la proposta di attività, lo stagista potrà comunicare di avere le capacità oppure no (rappresentato tramite uno **XOR split**), in quest'ultimo caso sarà ricondotto indietro all'evento "riceve proposta attività", tramite uno **XOR join**.

Qualora abbia le capacità, riceverà la descrizione dell'attività e inizierà a svolgerla.

Ad un certo punto, dopo una determinata scadenza fissata dal Project Leader, dovrà inviare un resoconto a quest'ultimo, per poi attendere una risposta (**event based gateway**).

La risposta ricevuta potrà coincidere con nuove informazioni su una nuova attività, e ciò porterebbe allo **XOR join** precedente, ricominciando il ciclo sopra descritto; oppure potrebbe ricevere informazioni relative alla stessa attività, ma con una nuova scadenza (la quale coincide con la fine dell'attività).



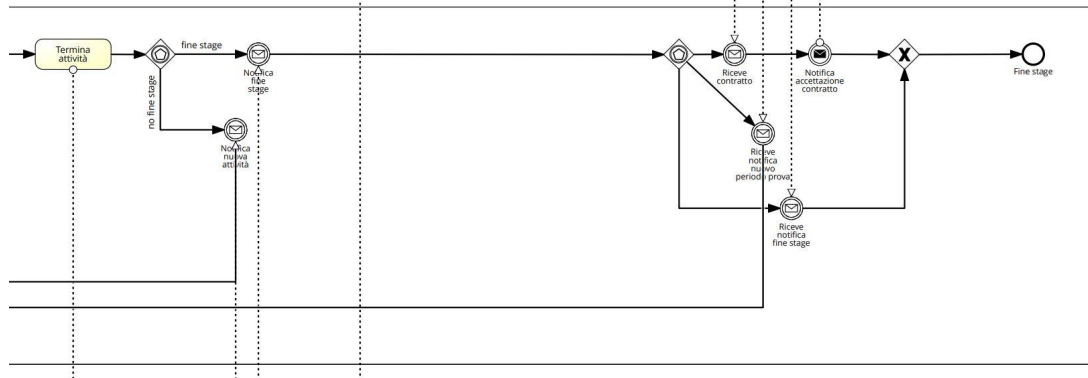
Al termine dell'attività, un **event based gateway**, pone lo Stagista in una condizione di attesa, durante la quale attenderà informazioni sull'attività da svolgere successivamente.

Potrà ricevere una notifica relativa ad una nuova attività, e ciò lo condurrebbe ad uno **XOR join**, per ritornare al ciclo precedente dove stabilirà se ha le capacità o meno; oppure potrà ricevere una notifica di fine stage dal Project Leader.

A quel punto attenderà una comunicazione da parte del Supervisore (**event based gateway**).

Successivamente potrà: 1) ricevere un contratto, accettarlo (si suppone che lo stagista possa solo accettare il contratto) e terminare lo stage; 2) ricevere una notifica per iniziare un nuovo periodo di prova e quindi ritornare allo **XOR join** che precede l'attività "si

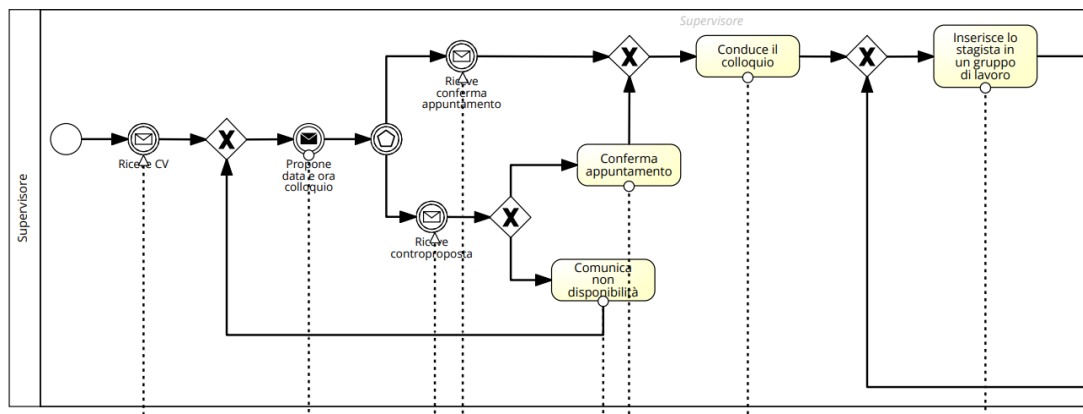
inserisce in un gruppo di lavoro”, con cui sarà inserito in un nuovo gruppo e ricomincerà il ciclo descritto sopra; 3) ricevere una notifica di fine stage, ma senza ricevere un contratto e terminare lo stage.



2. Supervisore

Il processo di business inizia con la ricezione del curriculum da parte dello Stagista e con uno **XOR join** che definisce il processo di scelta della data e ora del colloquio.

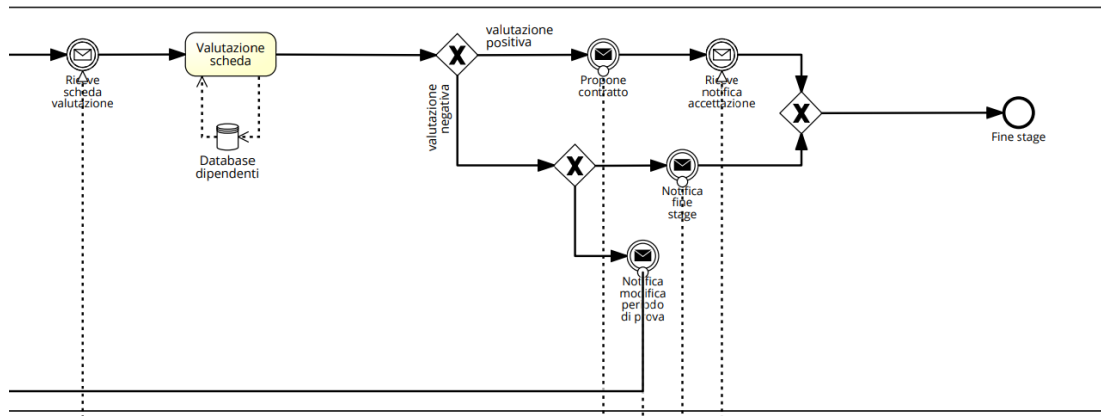
Alla fine del colloquio il Supervisore inserirà lo Stagista in un gruppo di lavoro e rimarrà in attesa fino a quando non riceverà la scheda di valutazione dello Stagista da parte del Project Leader.



Al momento della ricezione della scheda di valutazione, il Supervisore effettuerà una valutazione, confrontando le schede degli altri dipendenti contenute nel database (rappresentato tramite l'artefatto “**database**”).

Se la valutazione risulterà negativa (secondo output dello **XOR split**) potrà decidere (altro **XOR split**): 1) di proporre un nuovo periodo di prova e ciò porterebbe all'inserimento in un nuovo gruppo di lavoro (**XOR join**); 2) di notificare la fine dello stage e terminarlo.

Se la valutazione risulterà positiva (primo output dello **XOR split**), proporrà un contratto allo Stagista e attenderà la sua accettazione, così da portare a termine lo stage.

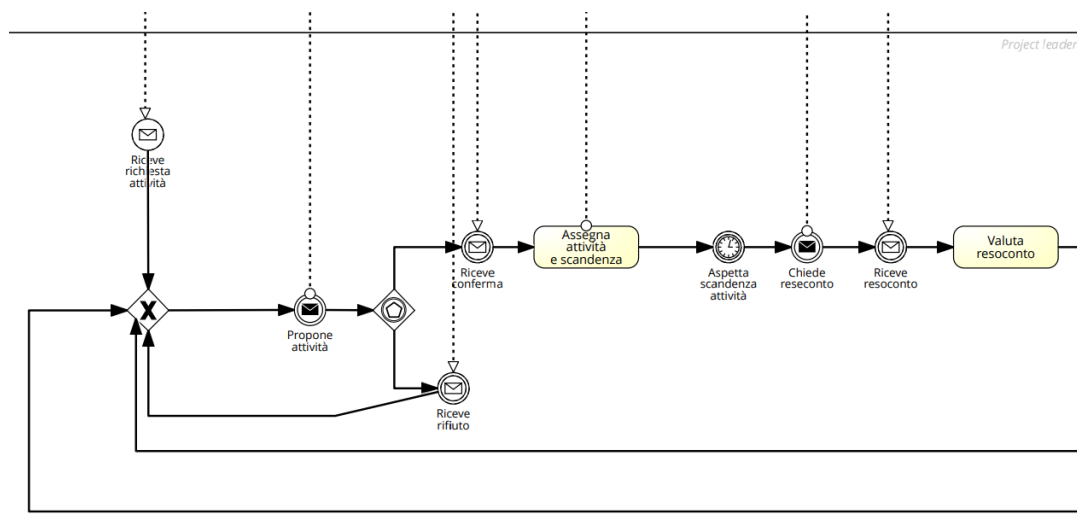


3. Project Leader

Il processo di business del Project Leader inizia con una richiesta di attività proveniente dallo Stagista. Successivamente il Project Leader proporrà un'attività e attenderà (**event based gateway**) che lo Stagista comunichi di avere (o meno) le capacità necessarie per svolgerla.

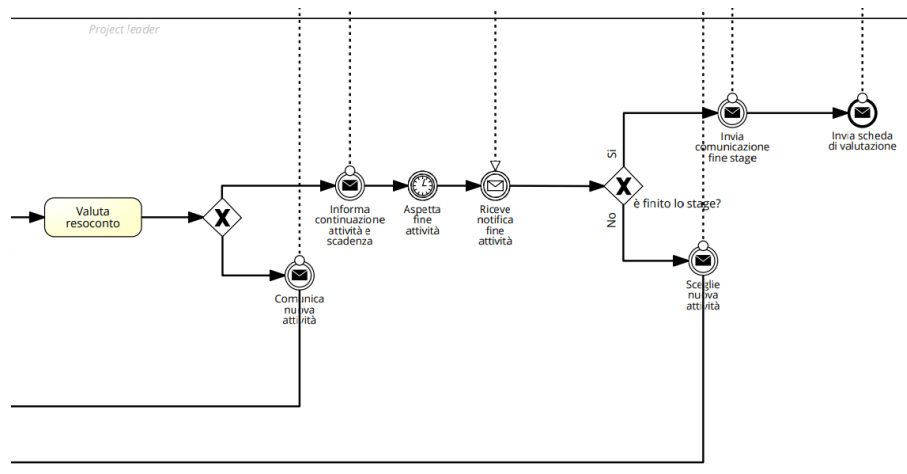
Se riceverà un rifiuto, il ciclo ricomincerà tramite uno **XOR join**; altrimenti l'attività verrà confermata, per poi assegnarla ufficialmente, insieme ad una scadenza.

Tramite un **intermediate message event** è stato rappresentato il periodo di attesa del Project Leader, al termine del quale chiederà un resoconto allo Stagista, per poi valutarlo.

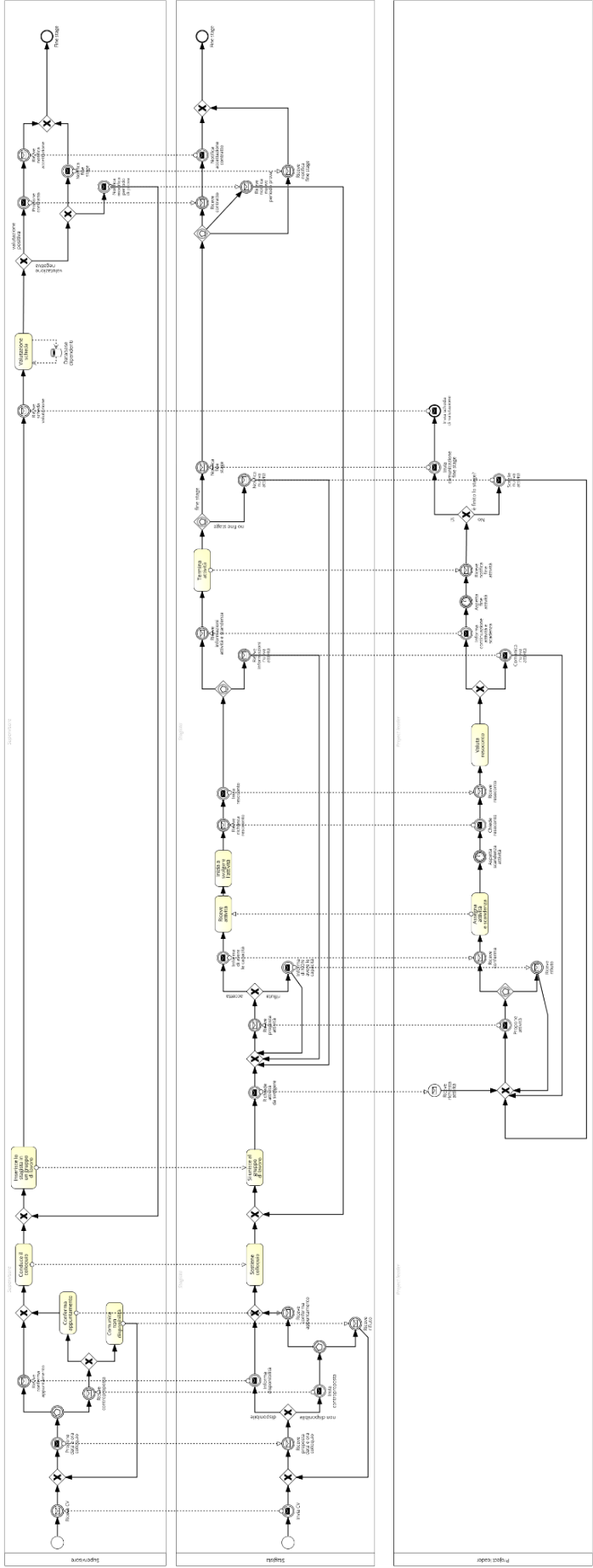


In seguito, tramite uno **XOR split** è rappresentata la possibilità del Project Leader di decidere se comunicare una nuova attività (e tornare indietro allo **XOR join** iniziale per riproporre una nuova attività) oppure se stabilire una nuova scadenza (che coincide con la fine dell'attività) e attendere la fine dell'attività, che sarà verificata tramite la ricezione di una notifica.

Infine, il Project Leader decide (**XOR split**) se terminare il periodo di stage (relativo alla suddetta attività del suddetto gruppo): qualora lo stage non fosse finito, sceglierà una nuova attività e tramite uno **XOR join** sarà ricondotto all'inizio del processo; altrimenti, invierà una notifica di fine stage, per poi concludere il suo processo inviando una scheda di valutazione al Supervisore.



Nella pagina seguente è mostrato l'intero processo tramite notazione BPMN.

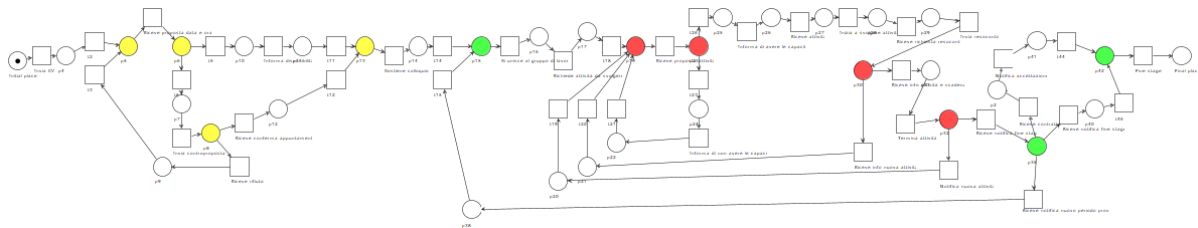


PETRI NETS - ANALISI

I **diagrammi BPMN** sono stati tradotti in **workflow nets** al fine di effettuare un'analisi semantica. Il processo di traduzione include tre step: prima sono state aggiunti i **places** al posto degli archi e le **transizioni** al posto delle attività e degli eventi; poi sono stati codificati i **gateways**; in seguito il net è stato **"de-sugarized"** ed infine sono stati aggiunti i **places iniziali e finali**.

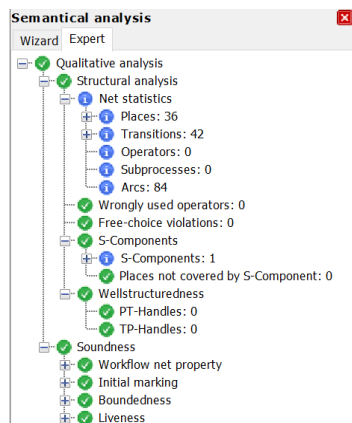
Per tutti i net analizzati, poiché la "finiteness" del processo è garantita, il coverability graph coinciderà con il reachability graph.

1. Stagista

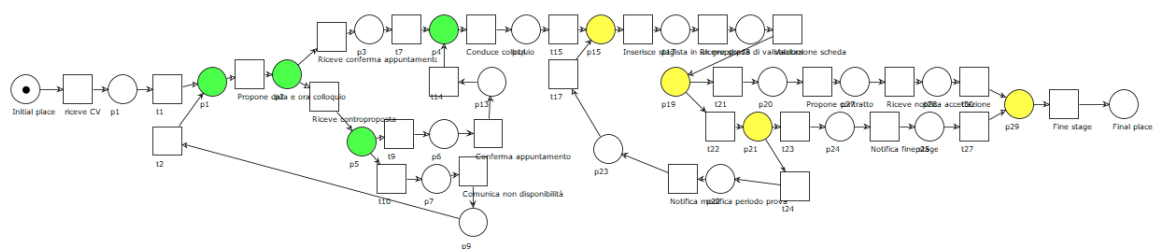


Analisi soundness e analisi strutturale:

- La net ha: 36 **places**, 42 **transizioni** e 84 **archi**
- Si tratta di un **Workflow net** perché:
 - 1) È presente un **initial place** il cui pre-set è vuoto
 - 2) È presente un **final place** il cui post-set è vuoto
 - 3) Esiste un **path** dal place iniziale fino al place finale
- Si tratta di una net **deadlock free**, **bounded** e **free-choice**
- N è una **S-net**, quindi sarà sound
- La net è **S-Coverable**: ogni place appartiene ad un S-component
- La net non ha PT-Handles o TP-Handles, quindi N è **Well Structured**
- Il **coverability graph** ha 36 vertici e 42 archi

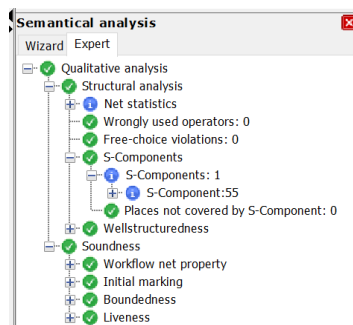


2. Supervisore

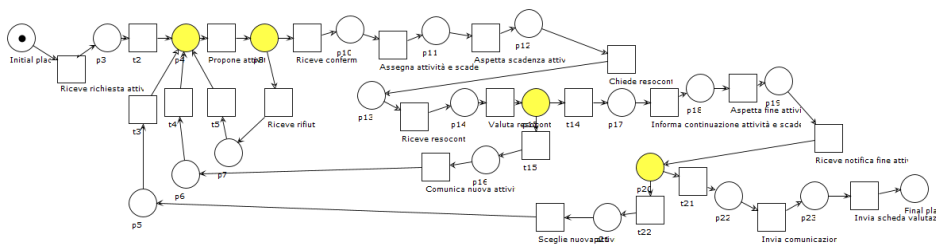


Analisi soundness e analisi strutturale:

- La net ha: 26 **places**, 29 **transizioni** e 58 **archi**
- Si tratta di un **Workflow net** perché:
 - 1) È presente un **initial place** il cui pre-set è vuoto
 - 2) È presente un **final place** il cui post-set è vuoto
 - 3) Esiste un **path** dal place iniziale fino al place finale
- Si tratta di una net **deadlock free, bounded e free-choice**
- N è **safe and sound** (perché N e N* sono S-nets, ma **non T-nets**)
- N* è live and bounded, poiché N è sound.
- La net è **S-Coverable** grazie ad un S-component che include 55 elementi
- La net non ha PT-Handles o TP-Handles, quindi N è **Well Structured** (poiché N* è Well-Handled)
- È facile dimostrare che la net è **safe e sound by construction** perché è formata da sotto-blocchi che sono sound
- Il **coverability graph** ha 26 vertici e 29 archi



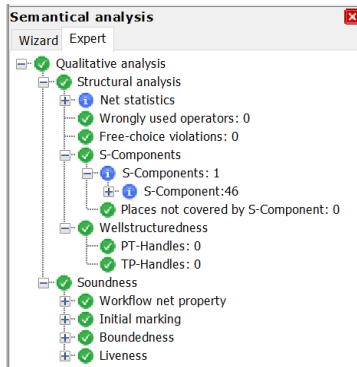
3. Project Leader



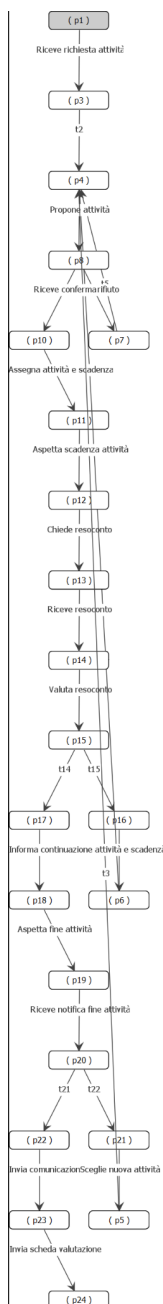
Analisi soundness e analisi strutturale:

- La net ha: 22 **places**, 24 **transizioni** e 48 **archi**
- Si tratta di un **Workflow net** perché:
 - 1) È presente un **initial place** il cui pre-set è vuoto
 - 2) È presente un **final place** il cui post-set è vuoto
 - 3) Esiste un **path** dal place iniziale fino al place finale
- Si tratta di una net **deadlock free, bounded e free-choice**
- Sia N che N* sono **S-net** (ma non T-net), poiché ogni transizione ha un solo arco entrante e un solo arco uscente. Per tale ragione N può essere definita anche safe and sound.
- La net N è **S-Coverable** grazie ad un S-component che include 46 elementi
- N è **Well Structured** (no PT-Handles o TP-Handles)
- Poiché la net è live e bounded allora avrà un positive S-Invariant
- Poiché la net è live e bounded allora avrà un positive T-invariant

- Il **coverability graph** ha 22 vertici e 24 archi



4. Coverability graph per il Project Leader



Il coverability graph coincide con il reachability graph. Inoltre, poiché si tratta di un finite graph, allora sarà bounded.

Per ciò che riguarda la proprietà della liveness, essa non può essere soddisfatta per N. Per esempio, dopo aver eseguito la transizione t2, il place p3 diventa dead.

Nonostante questo, analizzando N^* , tutte le proprietà della liveness sono soddisfatte.

Ci sono tre cicli che partono da p4:

- Il **primo**, nel momento in cui il token arriva al place p8, permette di raggiungere il place p7, e successivamente di tornare al place p4, dopo aver eseguito la transizione t5; oppure, raggiunto il place p8, continua a scendere lungo il path.
- Il **secondo**, dopo che il token è arrivato al place p15, permette di tornare nuovamente a p4 passando per p16 e p6; oppure, raggiunto il place p15, continua a scendere lungo il path.
- Il **terzo**, invece, permette al token di raggiungere il place p20 dove sarà possibile eseguire le transizioni t22 e “riceve nuova attività” per tornare al place p4; altrimenti dopo avere eseguito le transizioni “invia comunicazione” e “invia scheda valutazione” permetterà di completare il processo.

Questo grafico mostra anche che sono soddisfatte le proprietà **option to complete** e **proper completion**.

Una volta che il token raggiunge il place p24 non ci sono altri token in altri places precedenti.

Inoltre, l'unico modo per portare il token al place iniziale consiste nell'esecuzione della reset transition.

In tal caso non sarebbero presenti dead tasks perché esisterebbe almeno un arco per ogni transition della net.

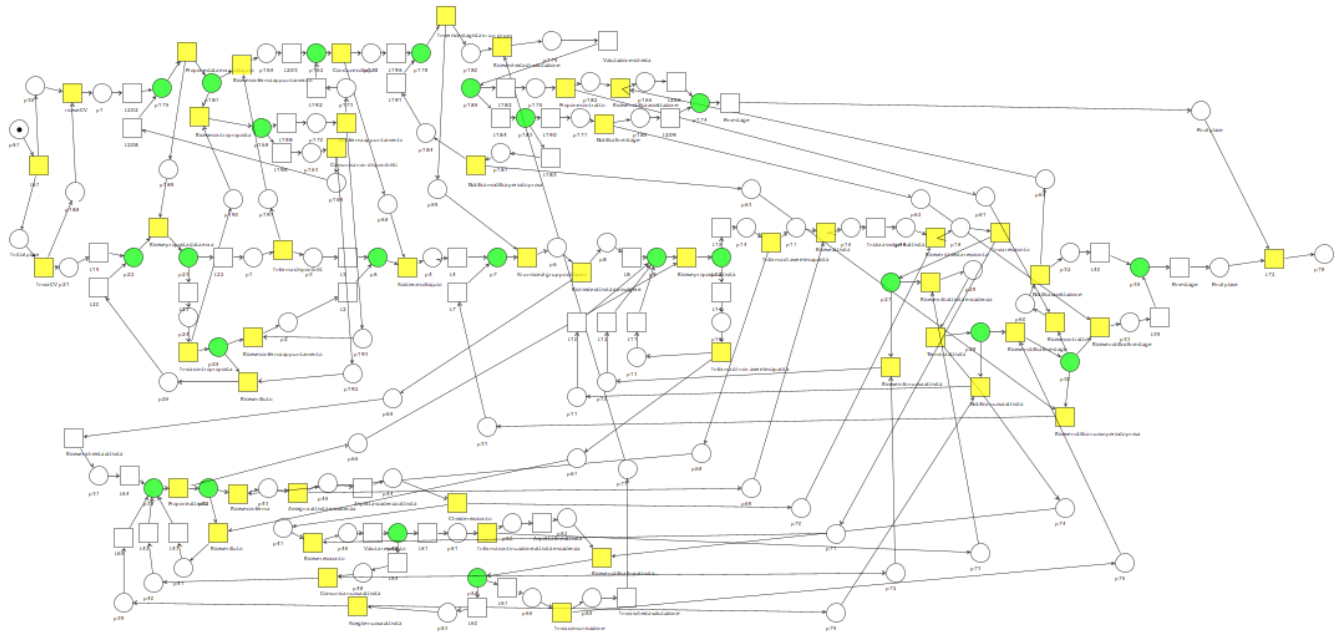
Quindi N è sound.

Conclusione

I workflow nets sono stati trasformati in un **workflow module**, aggiungendo dei **places** nei rispettivi “**message flow**”. I singoli workflow net, come illustrato nelle sezioni sopra, sono risultati sound, perciò è stato possibile unirli al fine di analizzare la soundness nel workflow module.

Inoltre, poiché vengono rispettate le proprietà “proper completion” e “option to complete”, è facilmente visibile come, qualora venga aggiunta una reset transition, la net possa essere considerata anche priva di dead task. Per tale motivo, N sarà sound.

Nell'immagine sottostante è rappresentato il workflow system risultante dall'analisi su Woped, insieme ai risultati dell'analisi realizzata tramite **Woflan**.



- ✓ Diagnosis
 - ✓ The process definition is a workflow process definition
- ✓ All conditions are proper
 - ✓ All conditions are covered by threads of control
- ✓ All tasks are live
- ✓ The workflow process definition is sound
- ⓘ Properties
 - ✓ Process definition
 - ✓ Conditions: 109
 - ✓ Tasks: 97
 - ✓ Workflow process definition
 - ✓ Start conditions: 1
 - ✓ End conditions: 1
 - ✓ Useless tasks and conditions: 0
 - ⓘ Condition properness
 - ✓ Threads of control: 1353
 - ✓ Task liveness
 - ✓ Dead tasks: 0
 - ✓ Non-live tasks: 0