



UNIVERSITÀ DI PISA

Business Process Modelling

P 56. Scuola di ballo

Antonia Giovinnazzi (667576)

Alma Stira (658699)

## INDICE

1. Introduzione.....	3
2. modellazione BPMN.....	3
2.1 Pool allievo.....	4
2.2 Pool scuola di ballo.....	5
3. Workflow nets.....	6
3.1 workflow net Allievo.....	6
3.2 workflow net scuola di ballo.....	9
3.3 workflow net completo.....	11
4. Variante processo .....	14

# 1. Introduzione

Il presente report si propone di analizzare e progettare un sistema per la gestione delle richieste degli allievi in una scuola di ballo, che permetta di ottimizzare e rendere efficiente l'intero processo di iscrizione, pianificazione e gestione delle lezioni. Lo scenario descritto implica una serie di interazioni tra allievo, maestro e scuola, che vanno dalla selezione del corso alla programmazione degli appuntamenti, fino alla conclusione del percorso formativo.

## 2. Modellazione BPMN

Lo scenario in oggetto è stato modellato utilizzando la nozione grafica **BPMN**, grazie all'uso della piattaforma online BPMN.io.

Nel nostro caso gli attori individuati sono due:

- Allievo
- Scuola di ballo

Per ciascuno dei quali è stata creata e modellata una **pool** che interagisce con l'altra tramite **message flow**; in particolare all'interno della pool della scuola di danza sono state individuate due **lane**, che descrivono rispettivamente i ruoli specifici della segreteria della scuola di danza e del maestro.

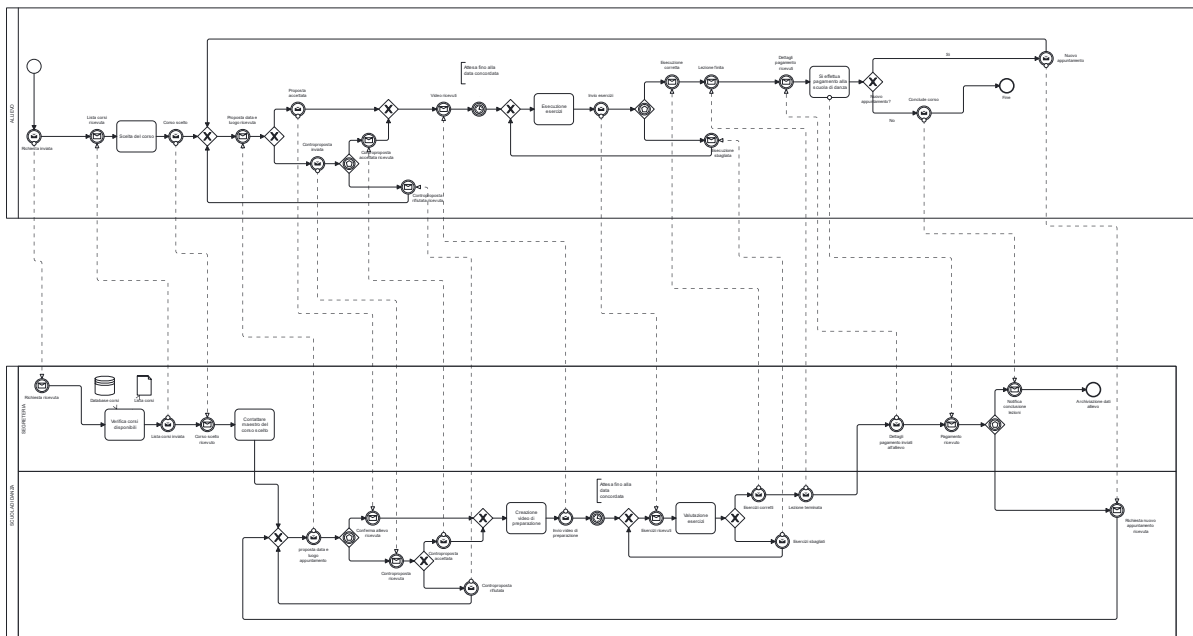


Figura 1: Bpmn completo

## 2.1 Pool allievo

In questa sezione analizziamo le attività dell'allievo; la task iniziale, prevede come il soggetto contatta la scuola inviando una richiesta dei corsi di ballo e rimanendo in attesa degli stessi. Una volta ricevuto il messaggio, esce dallo stato di attesa ed esegue il task di selezione del corso desiderato, inviando una nuova notifica alla scuola e aspettando la proposta per il primo appuntamento da parte del maestro.

Avendo ricevuto la proposta per l'appuntamento l'allievo avrà due scelte, modellate tramite un **XOR split**, che conducono a due strade diverse: nel caso in cui l'allievo accetti la proposta allora il processo genererà un messaggio verso il maestro, altrimenti elaborerà una controproposta e rimarrà in attesa di un nuovo messaggio del maestro. Questa situazione è stata modellata usando un **event based gateway** poiché nel caso in cui la controproposta sia accettata il flusso confluisce verso una porta **XOR di tipo join**, altrimenti ritornerà indietro in attesa che arrivi una nuova proposta dal maestro. Quest'ultima sequenza si ripeterà finché non si troverà un accordo definitivo che stabilirà la data e l'ora dell'appuntamento.

Successivamente l'allievo riceverà dei video da parte del maestro circa l'esecuzione degli esercizi; quindi, rimarrà in attesa dell'appuntamento come mostrato dall'evento intermedio del **timer**. Durante l'incontro l'allievo svolgerà gli esercizi che verranno giudicati dal maestro; per gestire questa fase è stato utilizzato un **event based gateway** per cui: se l'esito è positivo si avrà successivamente la comunicazione della fine della lezione tramite un **catching event**, altrimenti in caso di esito negativo l'allievo dovrà ripetere l'esercizio fino ad una corretta esecuzione.

L'allievo riceverà in seguito i dettagli per il pagamento, che provvederà a effettuare immediatamente. A questo punto l'allievo potrà decidere se prendere un altro appuntamento oppure concludere il corso, il che è stato modellato tramite un **XOR split**. Se l'allievo non desidera un nuovo appuntamento allora si raggiunge l'**end event**, altrimenti viene richiesto una nuova prenotazione e il flusso ritorna verso il primo **XOR join**, in cui l'allievo attenderà una proposta di data e luogo da parte del maestro.

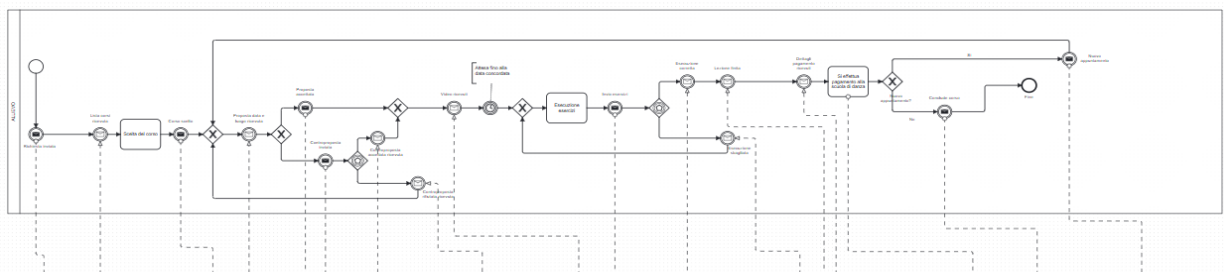


Figura 2: Bpmn della pool allievo

## 2.2 Pool scuola di ballo

In questa sezione analizziamo le attività della scuola di ballo. Questa, come detto precedentemente, è composta da una **lane**, per cui i nostri attori qui sono la segreteria e il maestro.

Il processo inizierà con uno **start message event** attivata nella pool della segreteria su richiesta dell'allievo. Questa, verificherà quindi i corsi disponibili nel database e una volta completata la verifica invierà la lista dei corsi disponibili all'allievo tramite un **throwing intermediate message event**; una volta ricevuta la risposta verrà informato il maestro e il flusso proseguirà nella sua lane.

Tramite un **throwing intermediate message event** il maestro proporrà all'allievo un data e un luogo per la lezione. L'**event based gateway** si propone di modellare due possibili eventi: se si verifica "conferma allievo" allora il maestro preparerà il video dell'esecuzione degli esercizi, altrimenti riceverà la "controproposta" da parte dell'allievo che deciderà se accettare o meno tramite un **XOR split**. Se la controproposta viene accettata il flusso converge verso un **gateway join** che convergerà alla preparazione del video, in caso contrario il maestro provvederà a proporre una nuova data; questa sequenza continuerà finché i due non raggiungeranno un accordo.

Inviato il video di preparazione, tramite un **throwing intermediate message event**, il maestro rimarrà in attesa dell'appuntamento come mostrato dall'evento intermedio del **timer**. Il processo prosegue poi con la ricezione dell'esercitazione dell'allievo, a cui seguirà la valutazione tramite un **XOR split** e la comunicazione dell'esito tramite un **throwing intermediate message event**; in caso di esito negativo l'esercitazione verrà ripetuta, in caso di esito positivo la lezione giungerà al termine e il flusso continuerà verso la lane della segreteria, che si preoccuperà di inviare i dettagli del pagamento all'allievo.

Una volta ricevuto il pagamento da parte dell'allievo, attraverso un **event based gateway** la segreteria potrà ricevere la comunicazione della fine dell'iscrizione e quindi archiviare i dati circa l'allievo, oppure potrà ricevere la richiesta di un nuovo appuntamento, il che convergerà nell'**XOR join** di proposta di data e luogo nella lane del maestro.

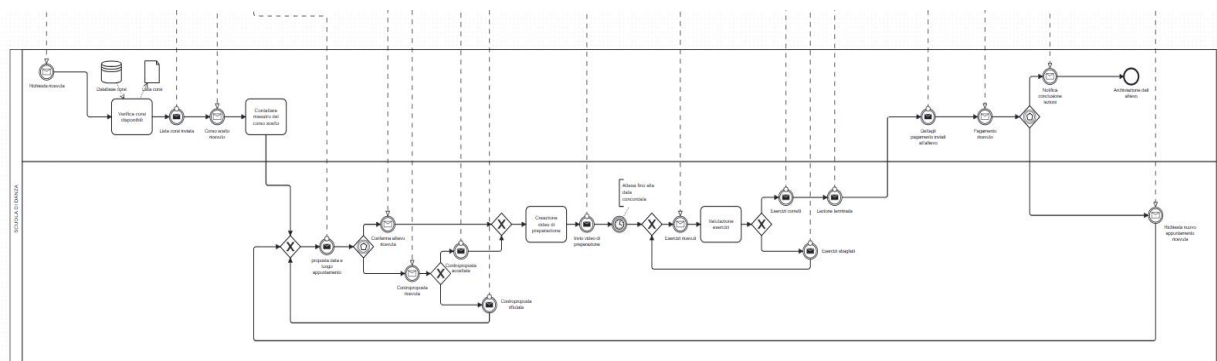


Figura 3: Bpmn scuola di danza

### 3. Workflow nets

Dopo aver modellato i processi con il linguaggio BPMN, è stato possibile convertirli in una particolare tipologia di Reti di Petri, chiamate **workflow nets**. Questa trasformazione permette di utilizzare una semantica più rigorosa e formale, rendendo disponibili diverse tecniche di analisi.

Nel dettaglio, la trasformazione da BPMN a workflow nets segue queste regole:

1. **Eventi BPMN** : sono stati trasformati in transizioni.
2. **Attività (task)** : sono state anch'esse convertite in transizioni.
3. **Sequence flow o message flow** : sono stati rappresentati con dei place.

Per quanto riguarda i gateway, la trasformazione varia in base alla logica dell'operatore:

- **XOR join** (più flussi in ingresso e uno in uscita): è stato rappresentato con più transizioni di input che convergono in un place con un unico arco in uscita.
- **XOR split** (un flusso in ingresso e più in uscita): è stato utilizzato un place, seguito da più transizioni e poi un place per ciascun ramo alternativo.
- **Event-based Gateway**: è stato trasformato in un place, seguito da una transizione per ciascun evento catching che attiva il flusso nel ramo corrispondente.

Per completare le Petri Net è stato necessario aggiungere un place di inizio, senza ingressi, connesso alla prima attività del processo; e un posto finale, senza uscite, connesso all'ultima attività del processo. Per quanto riguarda la connessione tra le transazioni delle diverse pool si è inserito un place intermedio, in modo da garantire una buona sincronizzazione tra i moduli del workflow.

In questo modo, i processi modellati in BPMN sono stati resi più rigorosi e formalizzati, facilitando le analisi e verifiche del comportamento del sistema.

### 3.1 Workflow net Allievo

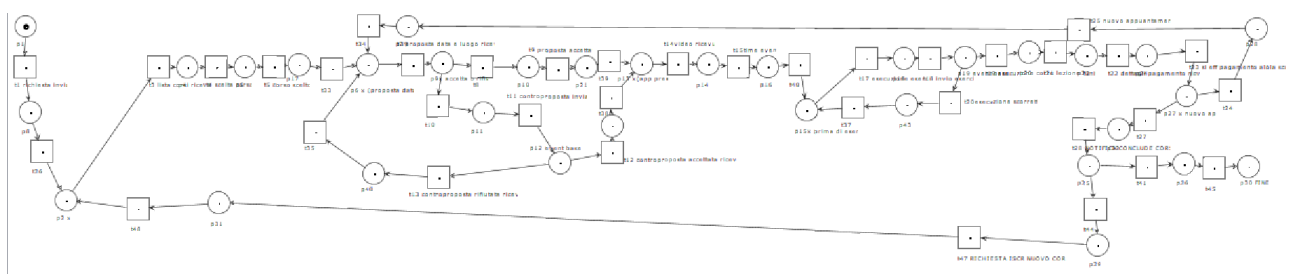


Figura 4: workflow net Allievo

La rete progettata per L'Allievo con Woped è riportata nella figura 5 .

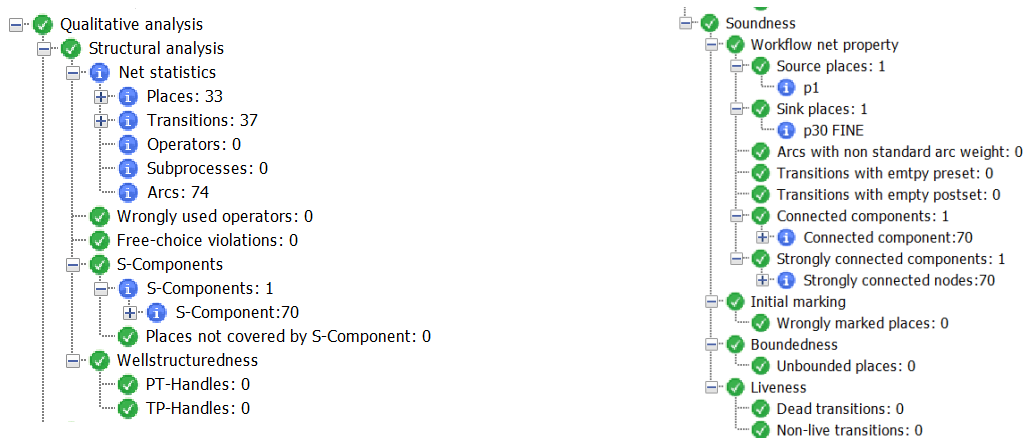


Figura 5: analisi della workflow net Allievo

I risultati “*Wrongly used operators: 0*” e “*Free-choice violations: 0*” mostrano che non ci sono errori o violazioni nei gateway; è stata identificata **1 S-component**, che indica una struttura ben connessa e priva di place non coperti, inoltre, non esistono PT-handles e TP-handles, pertanto, la rete è da definirsi well-structured.

La rete ha un unico luogo di inizio e un unico luogo finale, il che conferma la **Soundness** , inoltre la rete è **bounded** poiché è limitata e non contiene posti illimitati, e **Liveness** in quanto non ci sono transazioni morte o non-live, garantendo che tutte le transazioni siano accessibili durante l’esecuzione del processo.

Il **coverability graph**, nella figura 6 evidenzia la struttura dinamica della rete e le possibili evoluzioni del processo. Ogni luogo rappresenta uno stato raggiungibile, mentre le transizioni mostrano l’attivazione e il flusso tra questi stati. La presenza di ramificazioni e punti di convergenza conferma la capacità della rete di gestire più percorsi e alternative, rispettando le proprietà di *soundness* e correttezza analizzate in precedenza.

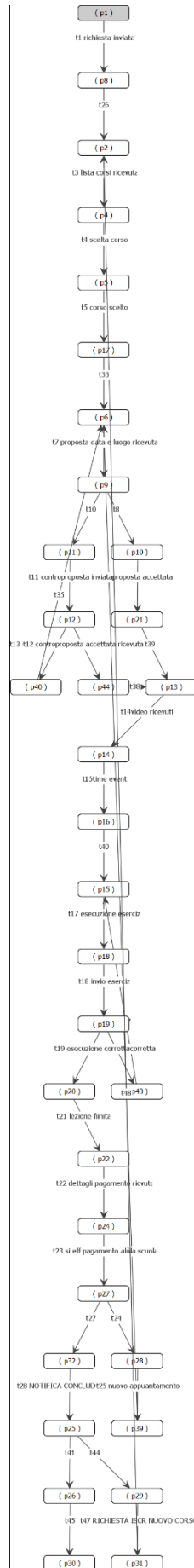


Figura 6: Coverability graph della workflow net Allievo



## 3.2 workflow net scuola di ballo

La workflow net della scuola di ballo come per il caso dell'allievo è stata realizzata con Woped (figura 7).

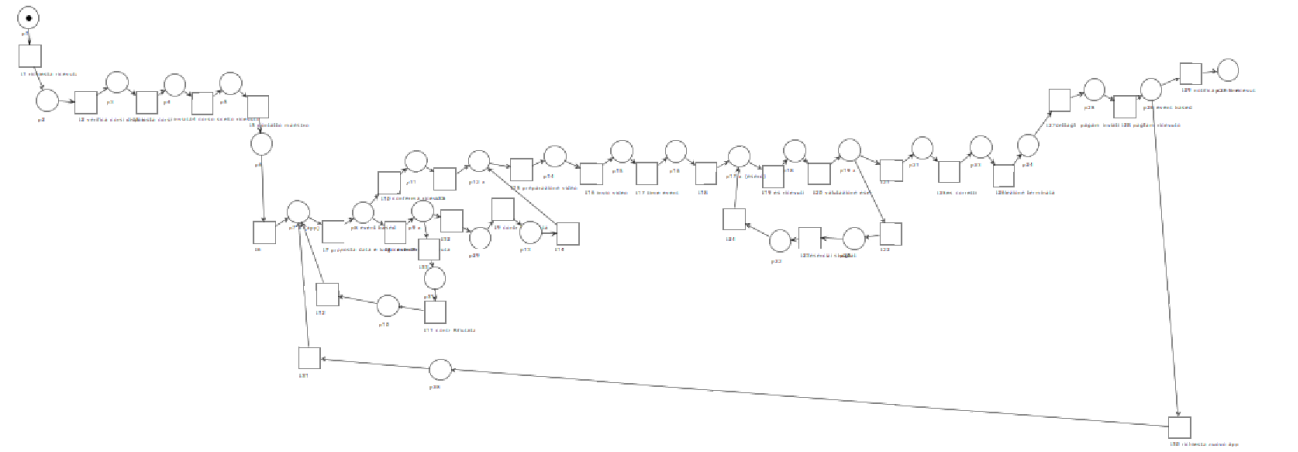


Figura 7: WORKFLOW NET SCUOLA DI BALLO

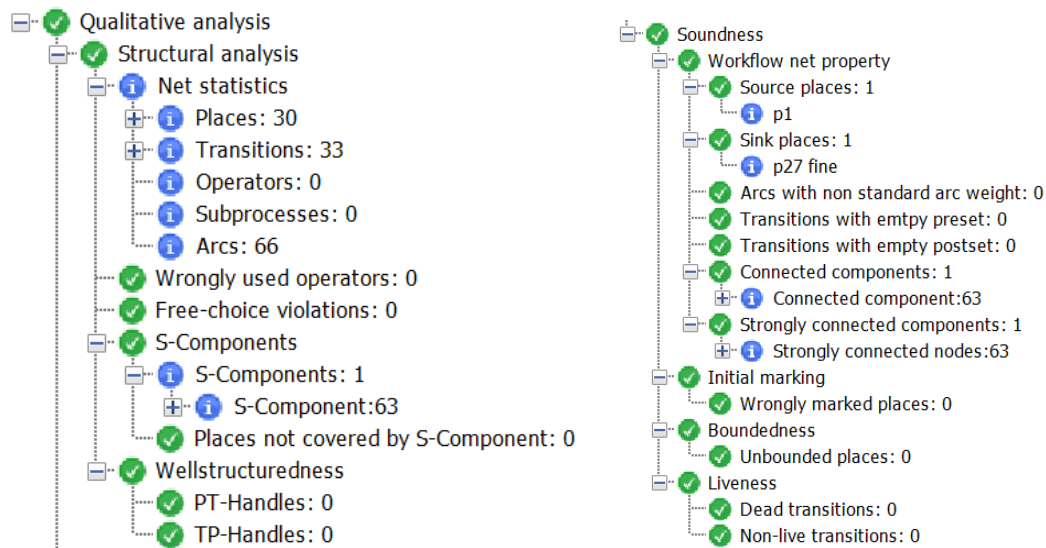


Figura 8: analisi della workflow net della scuola di ballo

Dall'analisi semantica della workflow net della scuola di ballo in figura 8, valgono tutte le considerazioni fatte nella sezione precedente: è una S-net quindi safe, sound e bounded, è strongly-connected, è una Free-choice net, S-coverable con un unico S-component e well-structured. Di seguito si analizza la struttura dinamica della rete e le possibili evoluzioni del processo in figura 9.

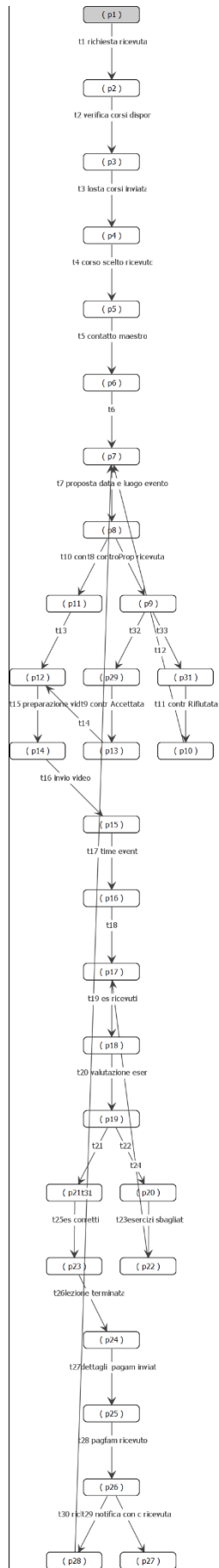


Figura 9: Coverability graph della workflow net scuola di ballo

### 3.3 workflow net completa

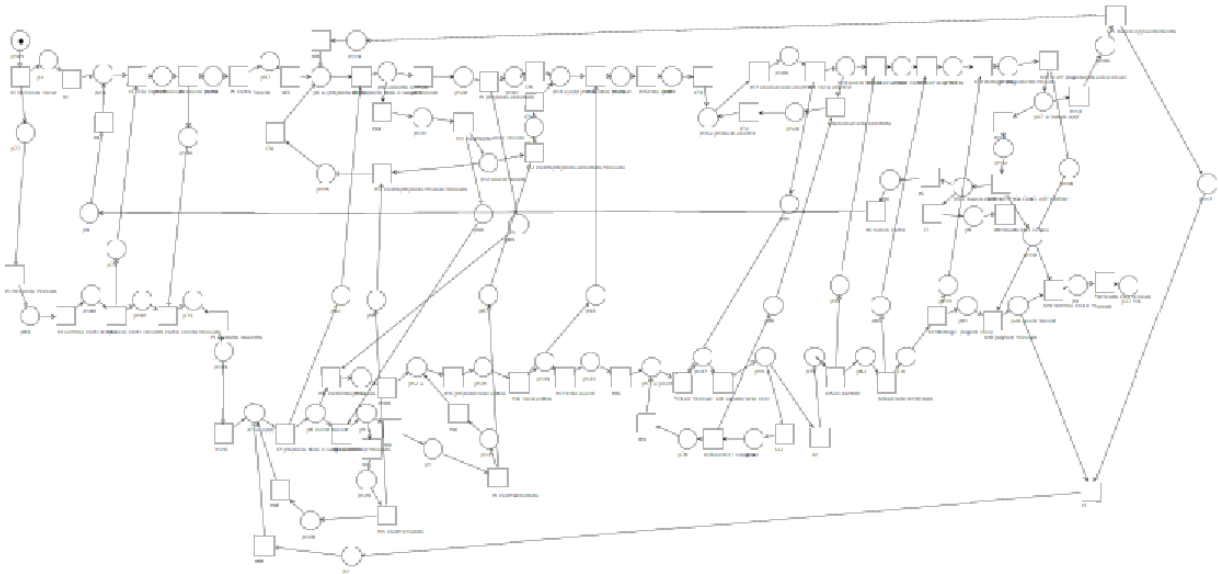


Figura 10: workflow net completo

Dalla figura 10 possiamo vedere come i due moduli del workflow sono stati integrati per ottenere una rappresentazione completa dell'intero processo oggetto di analisi. Il **place iniziale** coincide con quella presente nel modulo "Allievo", che rappresenta l'evento di contatto con la scuola. Questo evento funge da punto di avvio, generando nuove istanze dell'intero processo, mentre il processo si conclude **in segreteria**, che rappresenta lo stato finale in cui vengono completate tutte le attività amministrative e informative previste, chiudendo così il ciclo del processo.

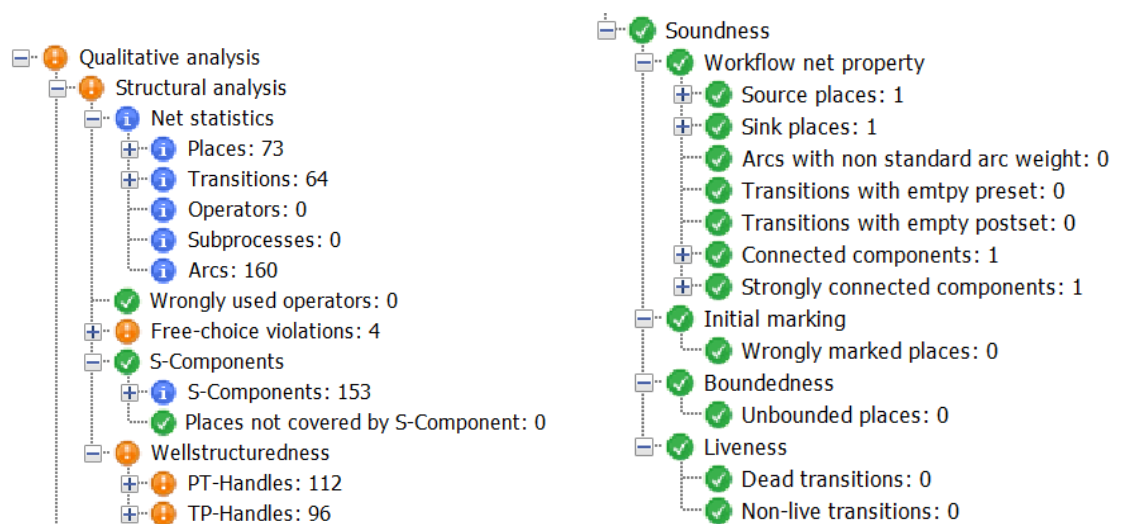


Figura 11: analisi della workflow net completa

Nella figura 11 sono riassunte le caratteristiche della workflow net; dal momento che l'aggiunta dei **place di interfaccia** per modellare il flusso informativo introduce nuove connessioni nella rete, rendendo possibile che più transizioni abbiano **preset non disgiunti** (input condivisi) ma **non uguali**. Questo rompe la proprietà Free-choice e di conseguenza, la rete **non può essere**

**classificata come una Free-choice net.** La rete risulta essere **S-Coverable** perché ogni place è incluso in una componente strettamente connessa (**S-Component**). Questo conferma che la struttura è completa e non presenta parti isolate o scollegate, dimostrando una **buona modellazione** del processo. La presenza di *handles* evidenzia una complessità moderata nella gestione dei nodi (posti e transizioni) e dei collegamenti tra di essi e di conseguenza non è well-structured.

Infine, il modello soddisfa i requisiti di correttezza, in quanto rispetta le proprietà fondamentali di soundness, boundedness e liveness, in conferma di questo abbiamo condotto un' analisi ulteriore presente in figura 12, svolta con Woflan.

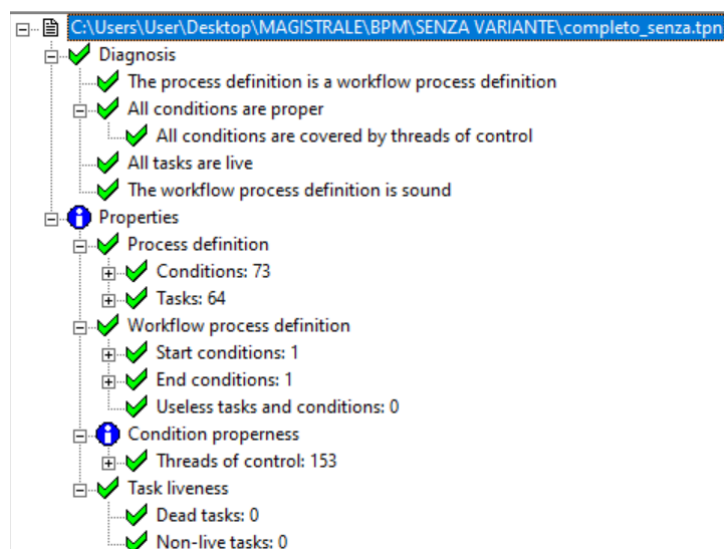


Figura 12: analisi completa condotta su WOFLAN

Il grafo in figura 13 mostra una **struttura complessa**, ma ben organizzata. La presenza di:

- **Percorsi multipli** con condizioni e transizioni interconnesse,
- Flussi chiaramente orientati da una condizione iniziale a una finale,
- Nessun loop improprio o dead tasks,

dimostra che la rete è **completamente funzionante** e rispetta i criteri di **soundness** e **liveness**. Il processo è in grado di gestire tutte le condizioni possibili e garantire il completamento del workflow senza blocchi.

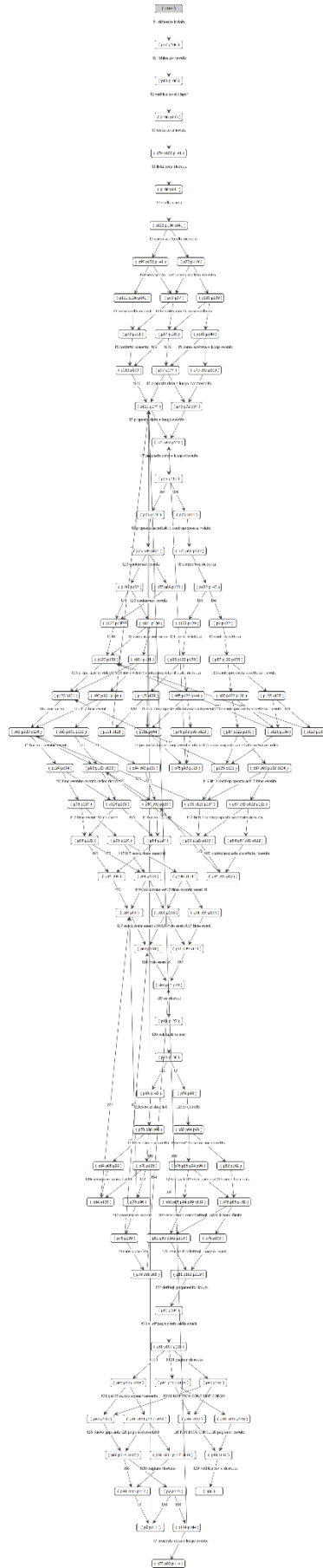


Figura 13: Coverability graph completo

#### 4. Variante del processo

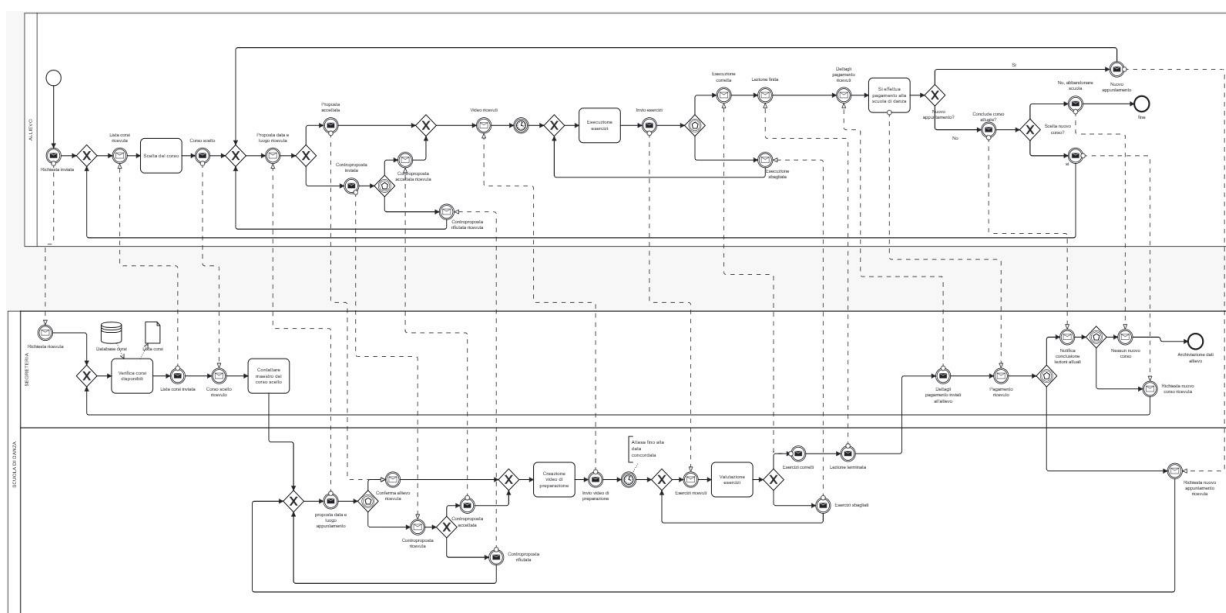


Figura 14: diagramma BPMN del processo con variante

In questa sezione si analizza l'estensione del modello descritto e analizzato finora con la variante che prevede: *al termine del corso per l'allievo, di scegliere se iniziare un nuovo corso di ballo o meno*. Lo sviluppo della variante come si può osservare nella figura 14, nel diagramma BPMN consiste nell'aggiunta di una porta XOR split, per modellare la possibilità di intraprendere un nuovo corso di ballo, qual ora quest'ultimo si verifichi viene comunicato alla scuola con un sequence flow.

Nel poll della scuola di ballo la modifica avviene dopo la notifica della conclusione del corso, è stato inserito un event based gateway che si abilita per modellare la possibilità che l'allievo desideri iniziare un nuovo corso; se quest'ultimo evento si verifica allora il sequence flow si riunisce al primo XOR join della segreteria che si occuperà di inviare nuovamente la lista dei corsi disponibili, altrimenti il processo cessa.

Di seguito la figura 15 mostra il processo tradotto in Petri net attraverso il workflow net, che rispecchia i cambiamenti effettuati in questa sezione.

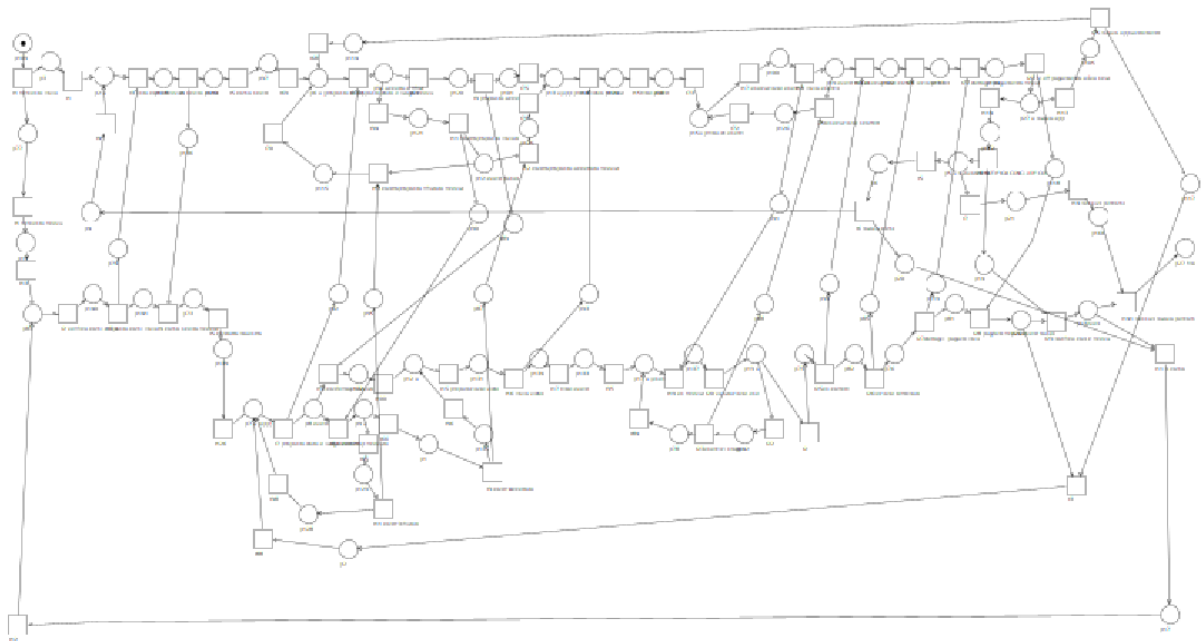


Figura 15: workflow net completo con variante

La figura 16 mostra le analisi semantiche effettuate sul workflow net con la variante, utilizzando sia Woped che Woflan, si giunge alla conclusione che per le stesse motivazioni precedenti la rete non è free-choice, ma è Soundness, Liveness e Boundedness.

Grazie alla presenza delle S-Component

e all'assenza di errori strutturali, possiamo classificare questa rete come S-net (Structured net), garantendo stabilità, correttezza e una chiara esecuzione del workflow.

Infine la figura 16 descrive il Coverability graph completo di questa sezione .

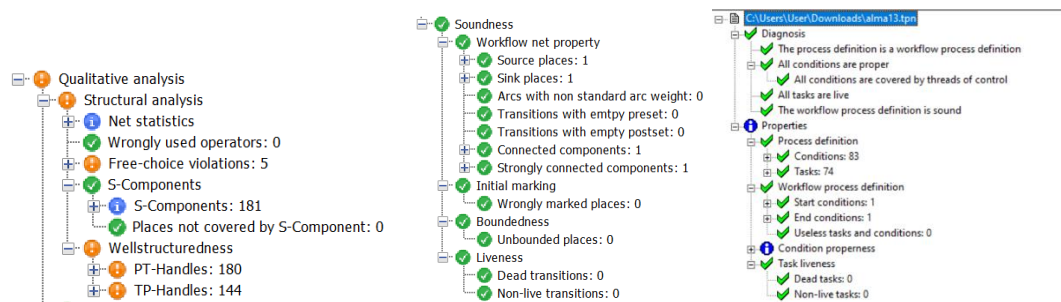


Figura 16: analisi workflow net completo con variante

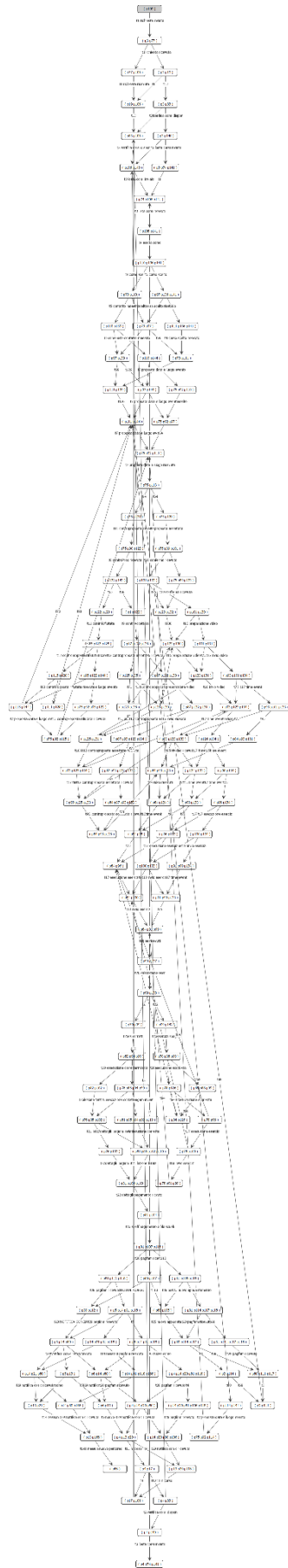


Figura 17: Coverability graph completo