UNIVERSITÀ DI PISA

Bussines process modelling
Project 58

Rustemi Ajla 598227
Bardazzi Carlo 603930

May 16, 2025

Contents

1 Introduction

This report presents the analysis and modeling of the business process for managing interactions between a painting school/its instructors and the student. The type of workflow its a **collaboration** involving Student and School/Instructor pools. The primary objective of the process is to facilitate course enrollment, lesson scheduling, learning activities, and course completion in a structured and efficient manner.

The analysis is structured from two main perspectives:

- Student in a distinct pool.
- Painting School/Instructor as swimlanes within that pool. Swimlanes are subdivisions within a pool, used to represent different participants or roles involved in a process.

followed by an analysis of the communication and synchronization between them.

2 BPMN

The BPMN model is designed to reflect real-world interactions, using appropriate gateways, events, and message flows to represent communication. The document further analyzes the impact of a variant where the student is given the option to start a new learning journey upon course completion. To create the BPMNs we used *BPMN.io* tools. The model here represents a detailed workflow of the painting school, managing student requests, lesson scheduling, and course completion. The process is designed to reflect the interactive nature of a learning journey, where communication and decision-making between the student and instructor are frequent and iterative. The process is divided into distinct phases that logically transition from course selection to lesson execution and finally to course completion.

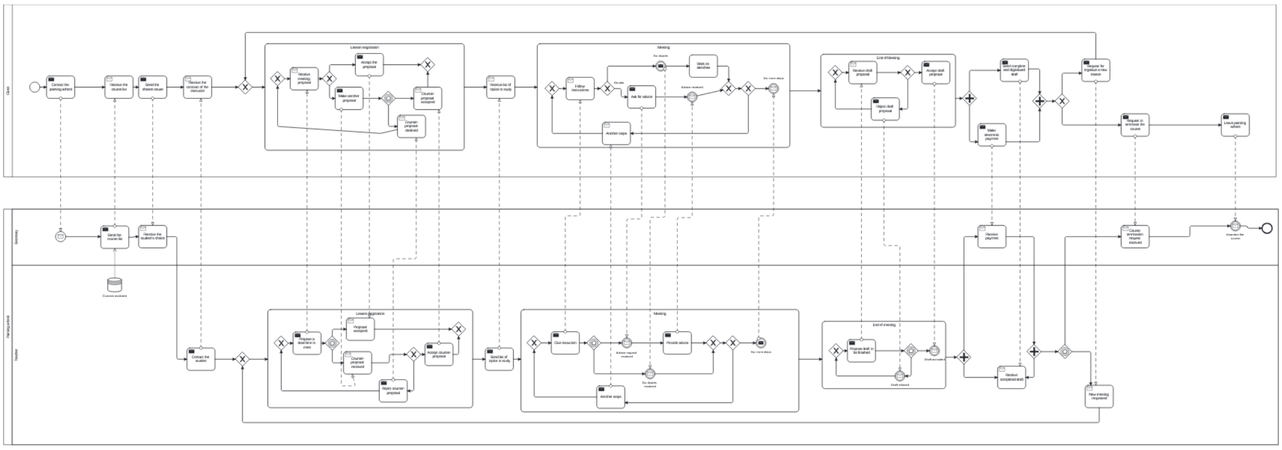


Figure 1: BPMN original

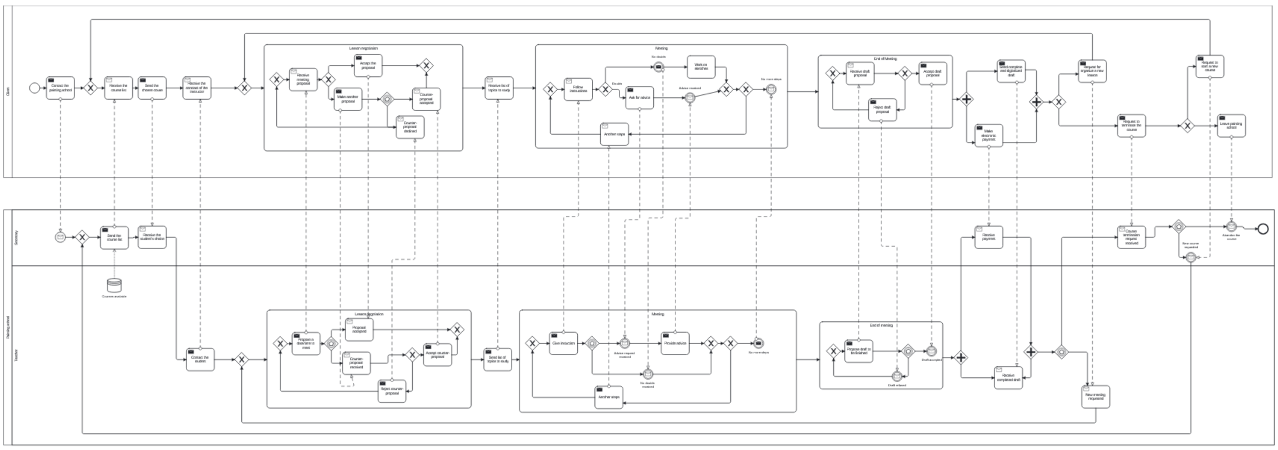


Figure 2: BPMN with the variant

2.1 Student's Pool

The student's process begins with contacting the painting school for enrollment. This interaction triggers a *Receive Task* in the BPMN model, where the student receives a list of available courses from the school. This information exchange allows the student to make an informed decision about the course. The separation of the reception of the course list and the decision to select a course is intentional. It allows the student to take time to review the options without stalling the process.
Upon reviewing the list, the student selects a course, represented by a *Send Task* labeled 'Choose new course.' This decision triggers the scheduling phase, where the student awaits a proposal for the first lesson's date and location from the instructor.

Scheduling the First Lesson

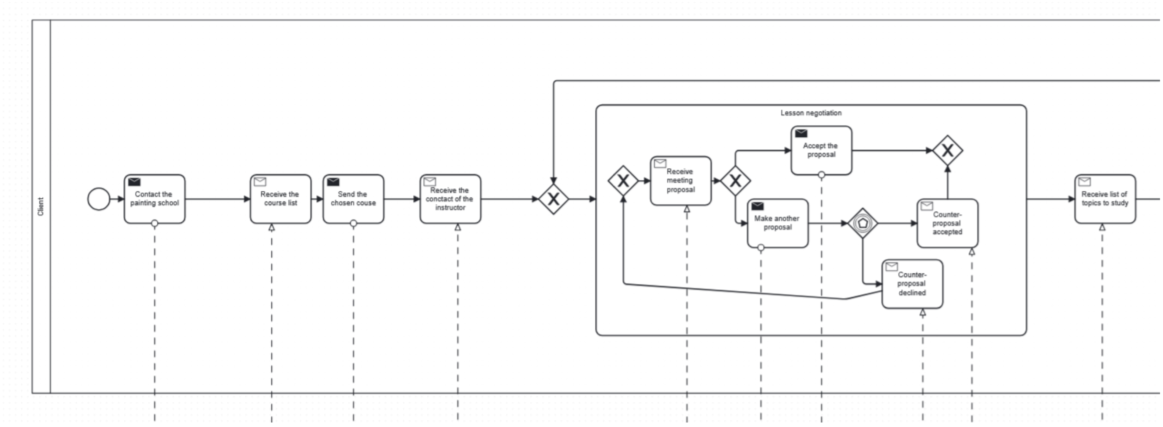


Figure 3: Starting process for the student pool

The scheduling phase begins when the instructor proposes a date and location for the lesson (fig. 3). In the BPMN model, this is represented by an *Exclusive Gateway*, which handles the decision logic. If the student agrees with the proposed schedule, the process continues to the lesson preparation phase. If not, a loop is initiated where the student proposes new dates until an agreement is reached.
The choice of using an *Exclusive Gateway* is justified by its ability to efficiently manage decision points with mutually exclusive paths. This representation reflects real-world negotiation scenarios where back-and-forth communication is necessary to find common availability.
The During the Lesson phase represents the core of the learning journey for the student. Once the student and instructor agree on the scheduled lesson, the student arrives prepared, having reviewed the list of techniques and tools provided beforehand. This preparation ensures that the student is ready to participate actively during the meeting, maximizing the value of the instructor's guidance.

Start of the lesson

The instructor provides step-by-step guidance as the student works on sketches and drafts. This is modeled through Provide Instructions, where the instructor explains each technique, maybe corrects mistakes, and answers questions in real-time.

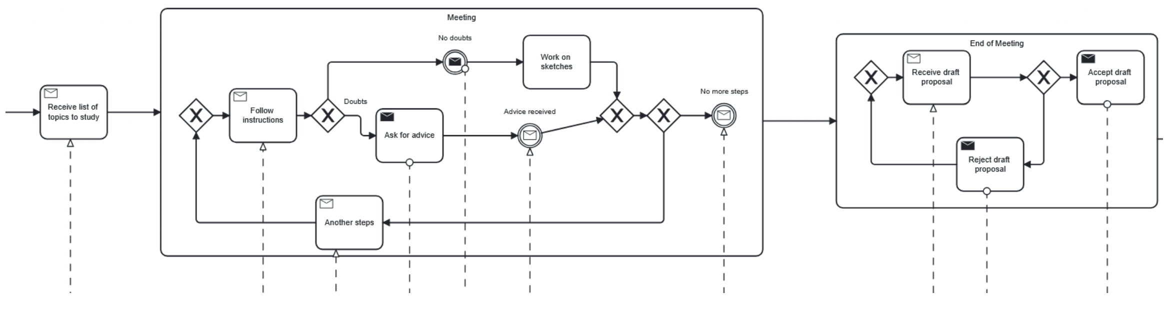


Figure 4: During and after lesson for the student pool

If the student encounters difficulties or needs clarification, a *Request* for Assistance is initiated. This is represented as a *Send Task* to the instructor, who responds with immediate feedback or further explanations. The *Exclusive Gateway* here handles the decision-making flow:

If the student is satisfied with the guidance and completes the sketch, the process continues.(fig. 4)
If not, the loop goes back to Provide Instructions until the student is confident with the work.

After the Lesson

After the practical session, the process transitions into After the Lesson activities. The instructor evaluates the drafts created during the session and proposes one for the student to complete as homework (fig. 4). This is modeled as a *Send Task* called Propose Draft for Completion. The student then reviews the proposal:
If the student agrees with the instructor's choice, the process continues.
If the student wants to propose a different draft, it loops back to the instructor's evaluation.
Once agreed, the student completes the draft independently, digitalizes it, and submits it back to the instructor. This submission is represented as a *Send Task* in the BPMN model. The instructor reviews the digital submission and provides feedback, ensuring that the student's work meets the required standards.

Payment and Process Continuation

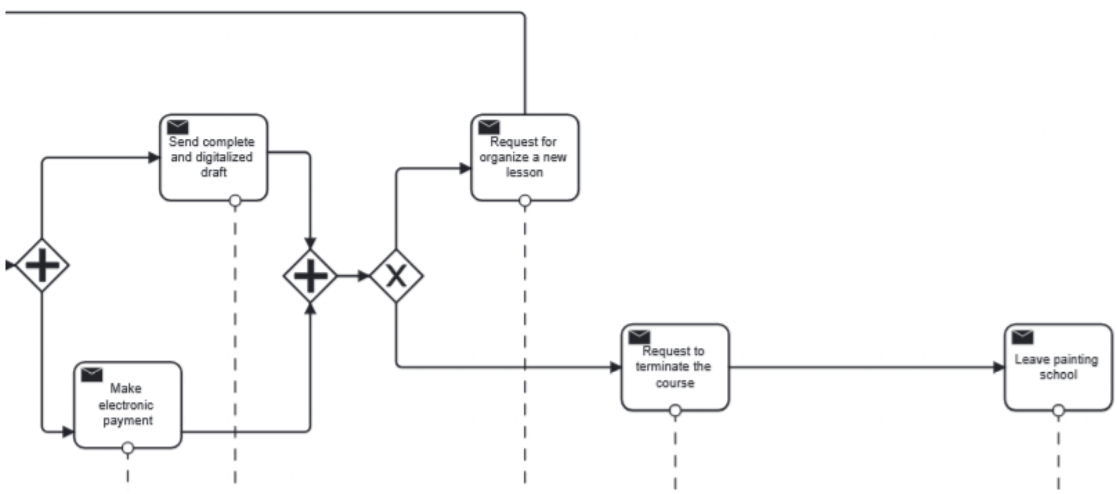


Figure 5: Ending process without the variant for the student pool

Upon submission of the digitalized draft, the student might proceeds to make an Electronic Payment to the school, represented as a *Send Task*. This step is critical for maintaining the flow of the process:
Once payment is confirmed, the process splits using a *Exclusive Gateway*.
One path allows the student to schedule a new lesson, looping back to the Lesson Preparation phase (fig. 5.
The other path terminates the course if the student does not wish to continue.

Course Termination or Renewal

The final phase models the student's decision to either end the course or continue with further lessons:
If the student chooses to end the course, a *Terminate Event* is triggered, representing the official completion of the learning journey.(fig. 5)
If the student opts for a new session, the process loops back to Lesson Preparation with a new scheduling negotiation.
The choice is managed by an *Exclusive Gateway* to ensure only one path is followed: either the student terminates the course or initiates a new learning cycle.

2.2 School's Pool

Course Offering and Scheduling

From the perspective of the school, the process initiates upon receiving a contact request from the student. This is modeled as a *Receive Task*, after which the school sends a list of available courses to the student. Upon receiving the student's course selection, the process transitions to the instructor, who is responsible for proposing the first lesson's schedule.

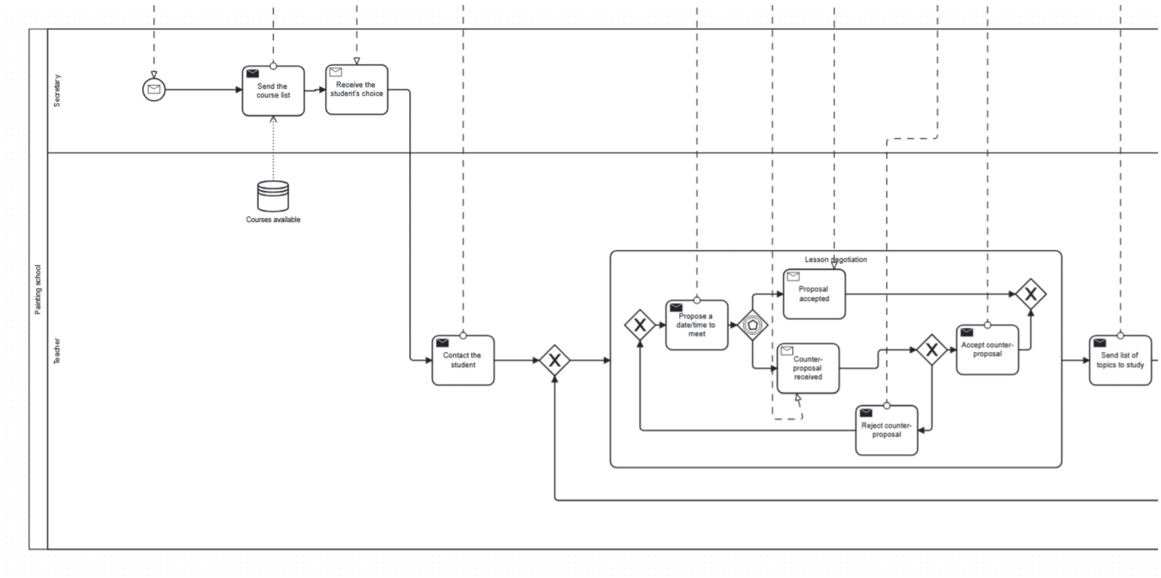


Figure 6: Starting process for the instructor/school pool

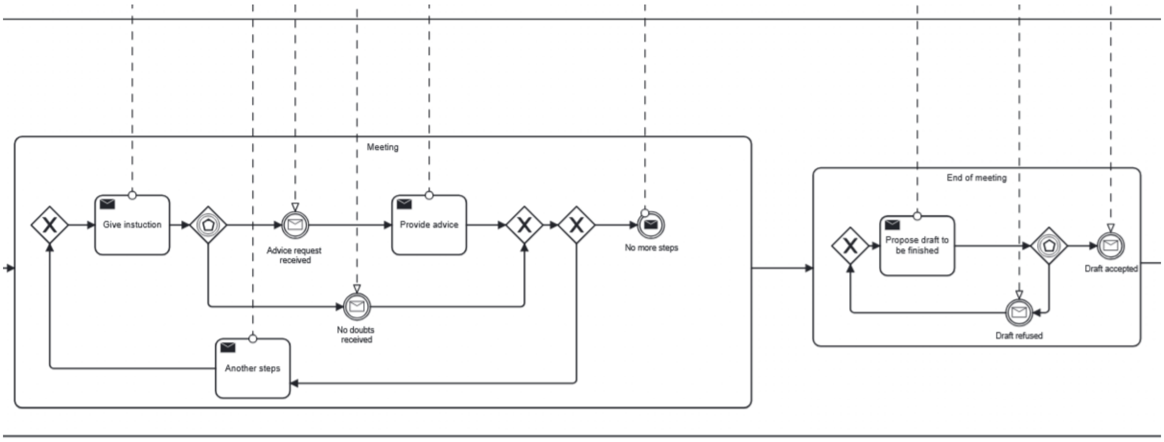


Figure 7: During the lesson and after for the school/instructor pool

The instructor's proposal is modeled as a *Send Task* to the student, representing the suggested date and location. The use of an *Event-based Gateway* allows the process to wait for the student's response. If the student proposes a different date, the gateway loops back to the scheduling phase (fig. 6).

Lesson Preparation and Instruction

We divide the lesson in three main phases: before the lesson, during the lesson and after the lesson.
As the student works on sketches and drafts, the instructor provides step-by-step explanations and demonstrations(fig. 7). This is represented as:

- Provide Instructions: The instructor explains techniques, answers questions, and guides the student through each phase of the process.
- Follow Instructions: The student follows the instructor's guidelines, applying the learned techniques in real-time.

If the student finds difficulties, the instructor will:

Clarify Doubts: This is shown as a *Receive Task* for the instructor, where the student can express any confusion.

Provide Additional Explanations: If necessary, the instructor re-explains the steps and ensures the student understands before moving forward.

This is handled through a loop, represented by an *Exclusive Gateway* that checks if the student understood every step before moving forward. If not, it returns to Provide Instructions until the student is ready to move on.

One of the main features in the diagram is the Ask for Advice interaction, where the student, during the lesson, can request clarification. This is reflected as a *Message Flow* from the student to the instructor. The instructor responds with real-time feedback:

If the issue is resolved, the process moves forward.

If more clarification is needed, it loops back, ensuring no uncertainty remains.

End of Lesson and Draft Evaluation

At the conclusion of the lesson, the instructor evaluates the drafts prepared by the student(fig. 7). This is modeled as:

Propose Draft for Completion: The instructor selects the draft that the student should complete independently.

A Message Flow is used to communicate the proposal to the student.

Here, another *Event-based Gateway* manages the student's response:

If the student agrees, the process continues.

If the student suggests another draft, it loops back to the instructor to Evaluate and Propose Again.

Payment and draft evaluation

After the student completes the draft and submits it digitally, the instructor receives it for evaluation. This is represented as a *Receive Task* in the BPMN model:

The instructor reviews the draft and provides feedback.

If corrections are needed, the student is informed, and the loop continues until it is fully acceptable.

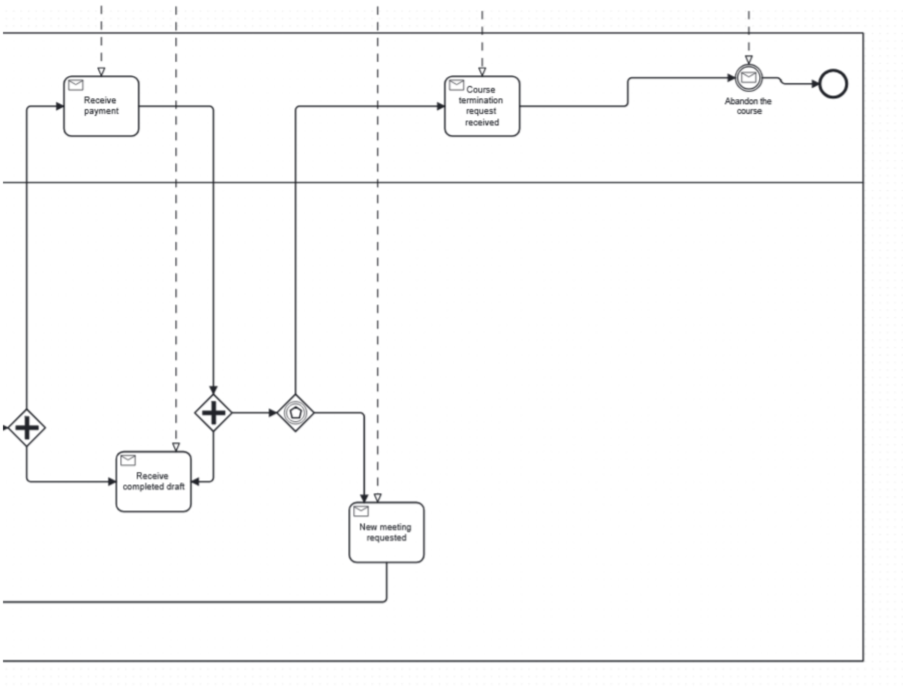


Figure 8: Ending without the variant for the school/instructor pool

Once the final draft is approved, the School Administrative Office manages the payment:

This is modeled as a *Receive Task* labeled "Receive Payment."

Upon confirmation of payment, the instructor is notified to continue scheduling future lessons or terminate the course(fig. 8).

Course Continuation or Termination

At this point, the instructor initiates a choice:

If the student wishes to continue, a *Message Flow* triggers the scheduling of the next lesson.(fig. 8)

If the student decides to terminate, the instructor finalizes the process and closes the learning cycle.

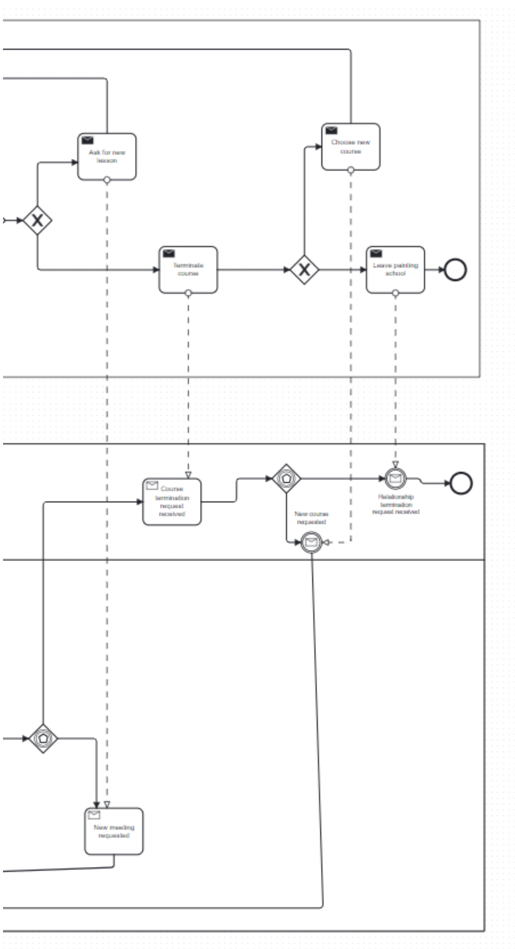
This decision is managed by an *Exclusive Gateway*, allowing only one path to be followed—either renewal of the cycle or termination.

2.3 Variant discussion

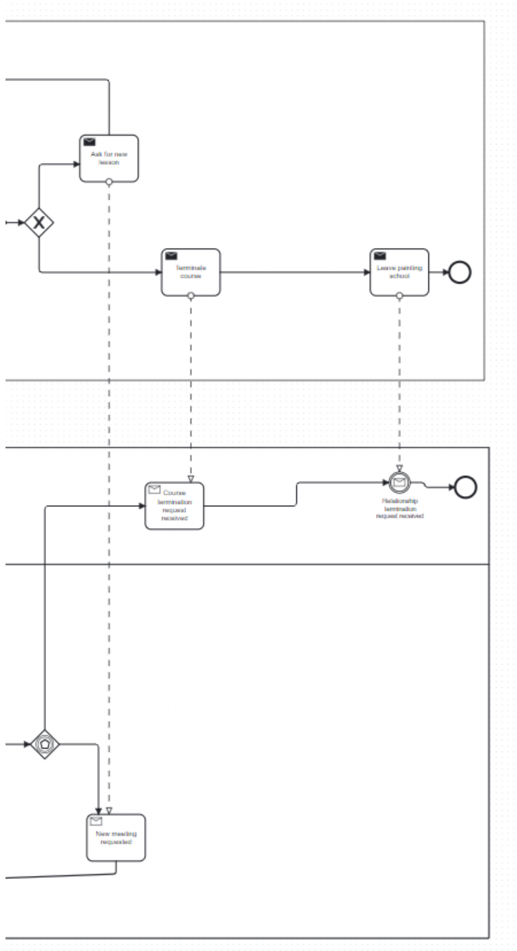
The proposed variant introduces an Immediate Re-enrollment Option at the end of the student's journey. This enhancement allows the student to decide, directly after completing the current course, whether they wish to begin a new learning cycle without exiting the process flow.
This is modeled with an additional *Exclusive Gateway* placed after the Payment Confirmation and Course Termination phases. The gateway represents a decision:

End the Course – The process terminates as in the standard model.

Choose a New Course – The process loops back to the Course Selection phase, bypassing the need for re-registration or additional contact with the school.



(a) Ending Process with the Variant



(b) Ending Process without the Variant

Figure 9: Comparison of the Ending Process with and without the Variant

In the BPMN model, the Exclusive Gateway is added at the end of the student's process path creating so two outgoing flows:

One leading to a Terminate Event, symbolizing the end of the learning journey.

One looping back to the Course Selection task, reinitializing the process.

The Message Flow between the student and the instructor is maintained, ensuring that re-enrollment is communicated, and the instructor is ready to propose a new schedule.

3 Transitioning into a Petri Net

The transformation of a BPMN diagram into a Workflow Net, represented as a Petri Net, in which various BPMN elements are mapped to specific Petri Net components. For this transformation we used *Woped 3.7.1*. Tasks, user or service tasks, are represented as transitions, each connected to an input place (preconditions) and an output place (post-conditions). Events are handled as follows: Start Events map to an initial place containing a token, Intermediate Events become places that can pause or branch token flow, and End Events correspond to final places that consume tokens upon process completion. Gateways are translated into specific Petri Net structures: AND Gateways (Parallel Gateways) are represented by a split (one incoming place to multiple simultaneous transitions) and a join (multiple transitions converging to one place requiring all to fire). XOR Gateways (Exclusive Gateways) are modeled as a split where a token flows to exactly one path and a join where one of the preceding paths must complete. OR Gateways (Inclusive Gateways) are depicted as a split allowing token flow to one or more paths and a join that completes when all active incoming paths have finished. Message Flows, representing inter-process communication, are modeled as synchronized transitions linked with places in separate Petri Net components. The overall transformation process involves identifying start and end events for initial and final places, mapping each task to a transition with appropriate input and output places, replacing gateways with their corresponding Petri Net structures, and synchronizing message flows across nets. Finally, the validity of the transformation is assessed by ensuring reachability from the start place to the end place through all possible transition firings.

4 Workflow Nets

This section presents the transformation of the BPMN process for the Painting School scenario into Workflow Nets, represented as Petri Nets. Workflow Nets are a special kind of Petri Net that are particularly suited for modeling business processes with formal semantics, for properties such as soundness, liveness, and boundedness to be analyzed precisely. For the analysis we used programms such as *Woped*, *Woflan* and *Prom*.

4.1 Student's workflow

Starting off with the Student's Workflow; (fig. 10). The Student Workflow Net provides a detailed representation of student interactions with the school, commencing at the initial place (*p1*) where student contact initiates the workflow. The Student Workflow Net comprises 41 **places**, 45 **transitions**, and 92 **arcs**, demonstrating a structured representation of the student's interaction with the school. Analysis of the net reveals several key structural properties:

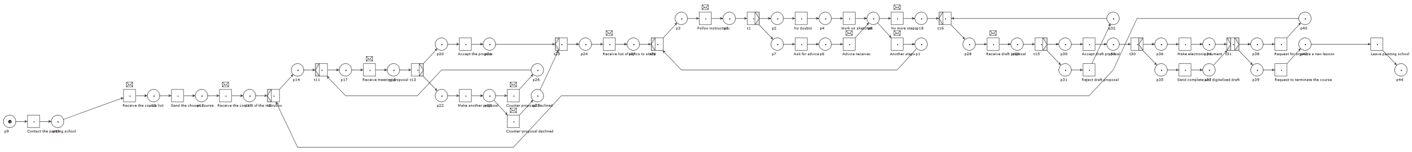


Figure 10: Students Petri Net

Analysis of the net reveals several structural properties (fig. 11):

- Well-Structured and Sound: The net exhibits a d design, free from wrongly used operators and free-choice violations. It is sound, characterized by a single source place (*p1*) representing the student's initial contact, and a single sink place (*p41*) indicating the completion of the workflow.
- S-Components and Boundedness: The presence of exactly two S-Components within the net indicates its capacity to model both cyclic and acyclic processes, such as iterative date negotiations and linear progression through course selection. The net is also bounded, ensuring that no place accumulates an excessive number of tokens, thus guaranteeing controlled execution.
- Connectivity and Liveness: All transitions in the net are connected, with no empty presets or postsets, confirming a complete and functional model. Furthermore, all transitions are reachable from the initial marking, confirming liveness. This guarantees that for any valid action taken by the student, the corresponding state change in the net is always achievable, and it will lead to the next expected state without stalling. The net is deadlock-free, with proper synchronization mechanisms in place for activities such as meeting negotiations, lesson execution, and draft submission. Every cycle, such as those involving date proposals, is managed through S-Components, maintaining overall process integrity.

4.2 School's workflow

This Petri Net model effectively captures the School's Workflow (fig. 12) , tracing the student's progression from the initial contact. The School Workflow Net is composed of: 38 **places**, 42 **transitions** and 86 **arcs**,

The source place (*p1*) represents the school's reception of the student's initial contact, triggering the workflow. The school manages the proposal cycle, which is handled through the S-Components, allowing flexible negotiation and agreement on meeting dates. Feedback loops are modeled for draft reviews, where acceptance or suggestions for modifications are cycled until the student completes the task satisfactorily. Boundedness is

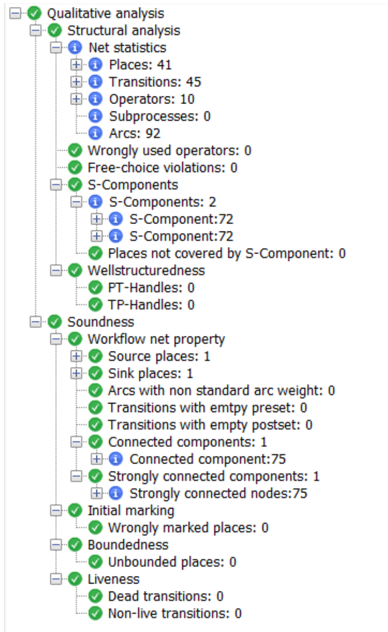


Figure 11: Semantic Analysis of the Student

preserved through careful synchronization, preventing token overflow and ensuring every transition is executed predictably. Similar to the student perspective, all transitions are live, ensuring full execution. Synchronization is achieved during course proposal and feedback loops. No deadlocks or infinite loops are present, confirmed through Coverability Graph Analysis.

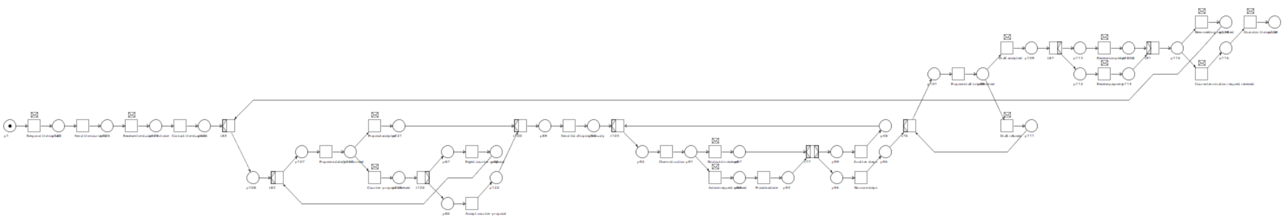


Figure 12: School/Instructor Petri Net

The net maintains well-structuredness, confirmed through the semantic analysis. (fig. 13) It also has 2 S-Components, similar to the Student Workflow Net. One source place (p1) and one sink place (p41) are defined, indicating a clear start and termination of the process. All arcs maintain standard weights, and there are no transitions with empty presets or postsets. All components are connected and support a seamless flow of execution.

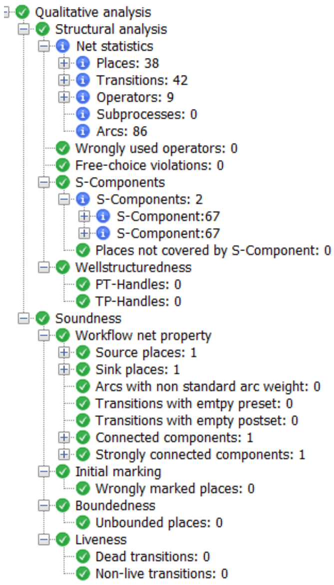


Figure 13: Semantic Analysis of the School

4.3 Full Petri Net

Here we will look into the full version of the Painting School Petri Net, with and without the variant.

Complete Workflow Net Analysis (Without Variant)

The Complete Workflow Net (without variant) comprises 100 **places**, 69 **transitions**, and 205 **arcs**. The net is well-structured and contains 4 S-Components, effectively representing both cyclic and acyclic paths within the workflow. These S-Components are responsible for modeling loops such as date negotiations and draft resubmissions, which are crucial elements of the business process. Notably, these cycles are controlled by exit conditions, preventing infinite loops and maintaining the soundness of the model.

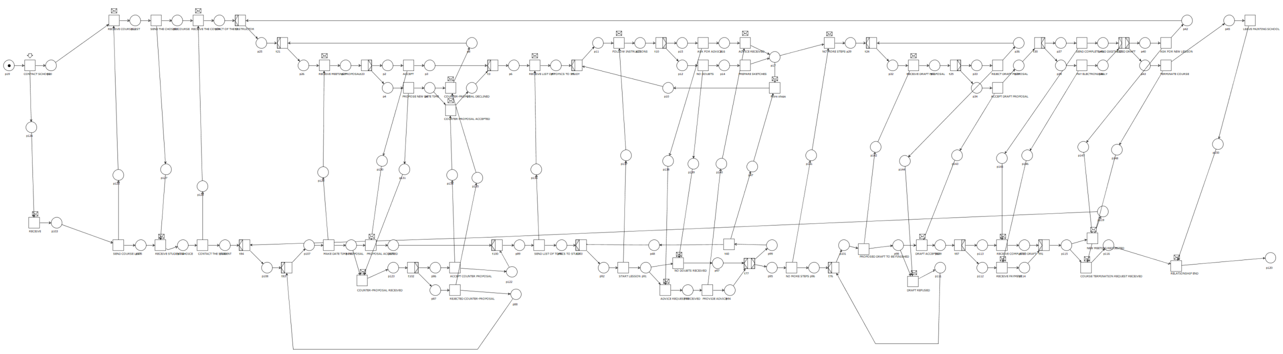


Figure 14: Full Petri Net without the variant

The net is confirmed to be sound, possessing a single source place (*p1*) and a single sink place (*p120*). This structure guarantees a clear start and termination point for the workflow, ensuring that all transitions are properly executed and the process reaches its intended completion without deadlocks or livelocks.

The Petri Net is also bounded, which guarantees that no place can exceed its maximum token capacity during execution. This is an essential property for workflow reliability, as it prevents overflow conditions that could disrupt the process. Boundedness also contributes to controlled resource management, ensuring that transitions are only triggered when required conditions are met.

Furthermore, the model is live, indicating that all transitions are reachable and can potentially fire during the process lifecycle. This eliminates the presence of dead tasks or non-live transitions, ensuring that every step in the workflow remains active and can be executed under the right conditions.

Communication between the student and the school is represented by synchronized transitions, modeling message exchanges and handshakes. These handshakes are critical for activities such as:

Proposal Acknowledgments: Ensuring that a date proposal or counter-proposal is confirmed.

Feedback Exchanges: Capturing the flow of information during lesson preparation or draft reviews.

Lesson Scheduling: Reflecting mutual agreements on scheduling and rescheduling lessons.

These synchronized transitions ensure proper alignment between the student's actions and the school's responses, maintaining a cohesive workflow.

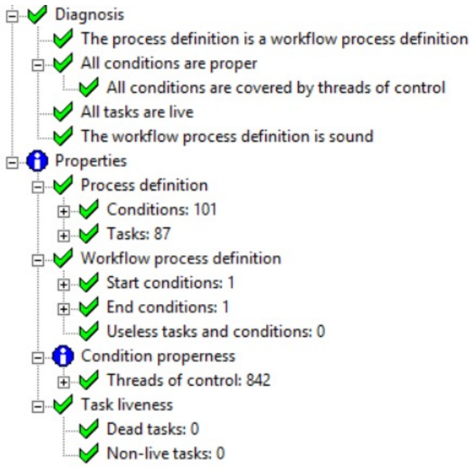


Figure 15: Semantic Analysis for painting school original

Despite its strong adherence to soundness and boundedness, the Petri Net structure does exhibit some structural concerns. There are rigid paths that could be further optimized for efficiency, particularly in the negotiation and feedback loops. Although bounded, the structure can be sensitive to token flow delays if synchronization between the student and school is not well-coordinated. Additionally, there are no empty presets or postsets, indicating that all transitions have proper input and output connections, which is a sign of structural completeness.

Complete Workflow Net Analysis (With Variant)

The Petri Net for this variant comprises 106 **places**, 73 **transitions**, and 218 **arcs**, reflecting the inclusion of additional decision points and synchronization paths that allow for the continuation of the student-institution relationship or its termination.

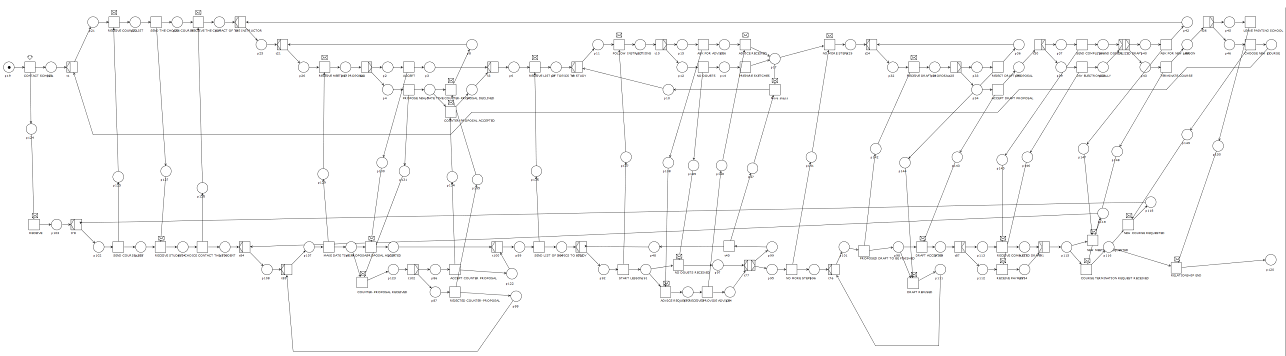


Figure 16: Full Petri Net with variant

The addition of two new decision points enables:

New Course Enrollment: A path that allows the student to select another course after completing the current one, extending their interaction with the institution.

Relationship Termination: A separate path to formally end the association with the school upon course completion.

These extensions result in an increase in S-Components from 4 to 5, reflecting the extra cyclic behavior introduced by the re-enrollment loop. This additional S-Component models the repetition of course selection, registration, and lesson preparation for the new session, effectively capturing the iterative nature of the extended workflow..

Despite the increased complexity, the Workflow Net remains sound. It preserves its single source place (*p1*) and single sink place (*p120*), guaranteeing a well-defined start and end to the process. Token flow logic

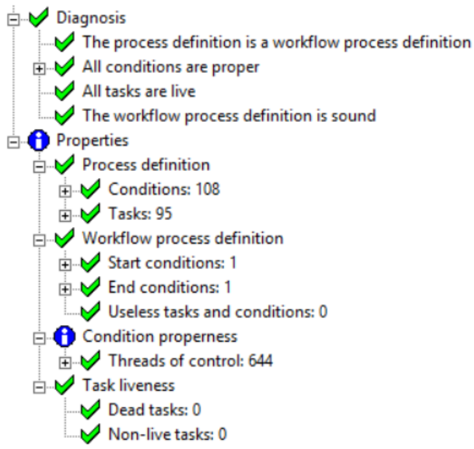


Figure 17: Semantic Analysis for Painting school with variant

is maintained across both the original and the variant path, ensuring that the workflow progresses correctly through re-enrollment or termination without causing deadlocks or livelocks.

The model is also bounded, demonstrating that all places maintain a controlled number of tokens during execution, even with the additional looping structure. This boundedness is crucial for managing system resources effectively and preventing overflow scenarios that could destabilize the workflow.

The variant introduces:

- New Synchronization Path: Specifically for the re-enrollment process, ensuring that both the student and the school acknowledge the decision to continue the relationship.
- Mutual Acknowledgment Mechanisms: At the decision point for course continuation, the system requires synchronized agreement between the student and the school before moving forward with re-enrollment. This handshake mechanism guarantees that both parties are aligned in their expectations.

Free-Choice and Well-Structured Analysis

The Workflow Net is not classified as a **Free-Choice Net**, as it contains transitions that share input places without sharing the entire preset. This violates the mutual exclusivity required for Free-Choice properties, impacting concurrency in certain execution paths.

Additionally, the net is **not well-structured** due to the presence of both PT-handles and TP-handles, as identified in our semantic analysis. These structural components prevent the net from maintaining a clean, architecturally sound layout, which affects its maintainability and modularity.

S-Net Classification and S-Coverability

The Workflow Net does not satisfy the requirements of an **S-Net**, as many transitions do not have a single input and output place. Furthermore, the analysis confirms that the net is **S-Coverable**, indicating that all transitions belong to strongly connected components.

.

4.4 Coverability and Reachability Graph

Here we will present the coverability graphs for our scenario.

The *Coverability Graph* of a Petri Net is a directed graph that represents all possible states (markings) that can be reached from the initial marking. It allows us to explore the behavior of the workflow net, verifying properties like *boundedness*, *liveness*, and the possibility of *deadlocks*. Each node in the coverability graph represents a reachable marking, and each edge corresponds to the firing of a transition that moves the net from one state to another. The main aim of this analysis is to estimate the size of the state space and understand the complexity of its execution paths. To evaluate the complexity of the Workflow Net, we perform an estimation of the Reachability Graph size for both the complete model with and without the variant. The estimation is based on the formula:

The estimation of the reachability graph is calculated using the formula:

$$\text{Estimated Number of Nodes} = 2^P \tag{1}$$

$$\text{Estimated Number of Edges} \approx T \times 2^P \tag{2}$$

Where P is the number of places and T number of transitions.
Based on the Petri Net models:

- **Original Version**: 100 places, 69 transitions, and a maximum of 1 token per place.
- **Variant Version**: 106 places, 73 transitions, and a maximum of 1 token per place.

Reachability Graph Estimation Results

Table 1 summarizes the estimated size of the reachability graph for both models:

Model	**Number of Places (P)**	**Number of Transitions (T)**	**Estimated Nodes**	**Estimated Transitions**
Original Version	100	69	2^{100}	69×2^{100}
Variant Version	106	73	2^{106}	73×2^{106}

Table 1: Reachability Graph Estimation for Both Versions

The estimations illustrate the exponential growth in the state space with the addition of places and transitions in the variant. This significant increase implies greater complexity during verification and simulation.

The growth in the Coverability Graph's size has a direct impact on:

- **State-Space Exploration**: The variant version demands greater computational power for full exploration.
- **Liveness and Boundedness Checks**: Although both models are live and bounded, the complexity increases in the variant due to additional decision paths.
- **Deadlock Detection**: The additional synchronization paths introduced in the variant increase the risk of deadlocks during re-enrollment cycles.

5 Conclusions

The analysis of the Complete Workflow Net, both with and without its variant, reveals several structural and behavioral propertiessuch as: The models are both **live** and **bounded**, ensuring reliable execution. The variant introduces more decision points and synchronization paths, adding complexity. Despite not being **Free-Choice** or **Well-Structured**, the models maintain soundness. The reachability estimation indicates significant growth in state space with the variant, which requires efficient simulation techniques.

Future improvements could focus on restructuring the model to enhance concurrency and reduce redundant loops in negotiation paths, improving overall efficiency.

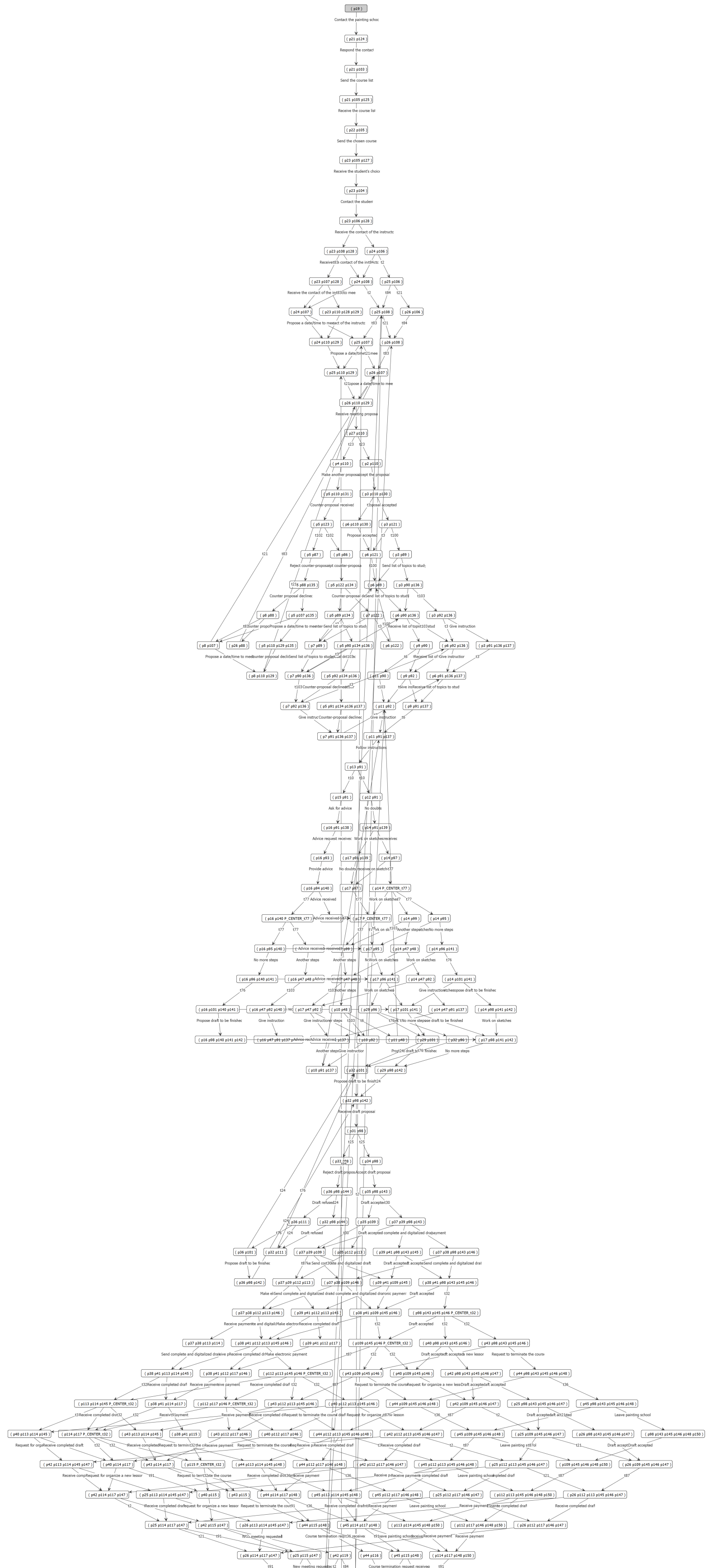


Figure 18: Complete Coverability Graph of the Workflow Net Without Variant

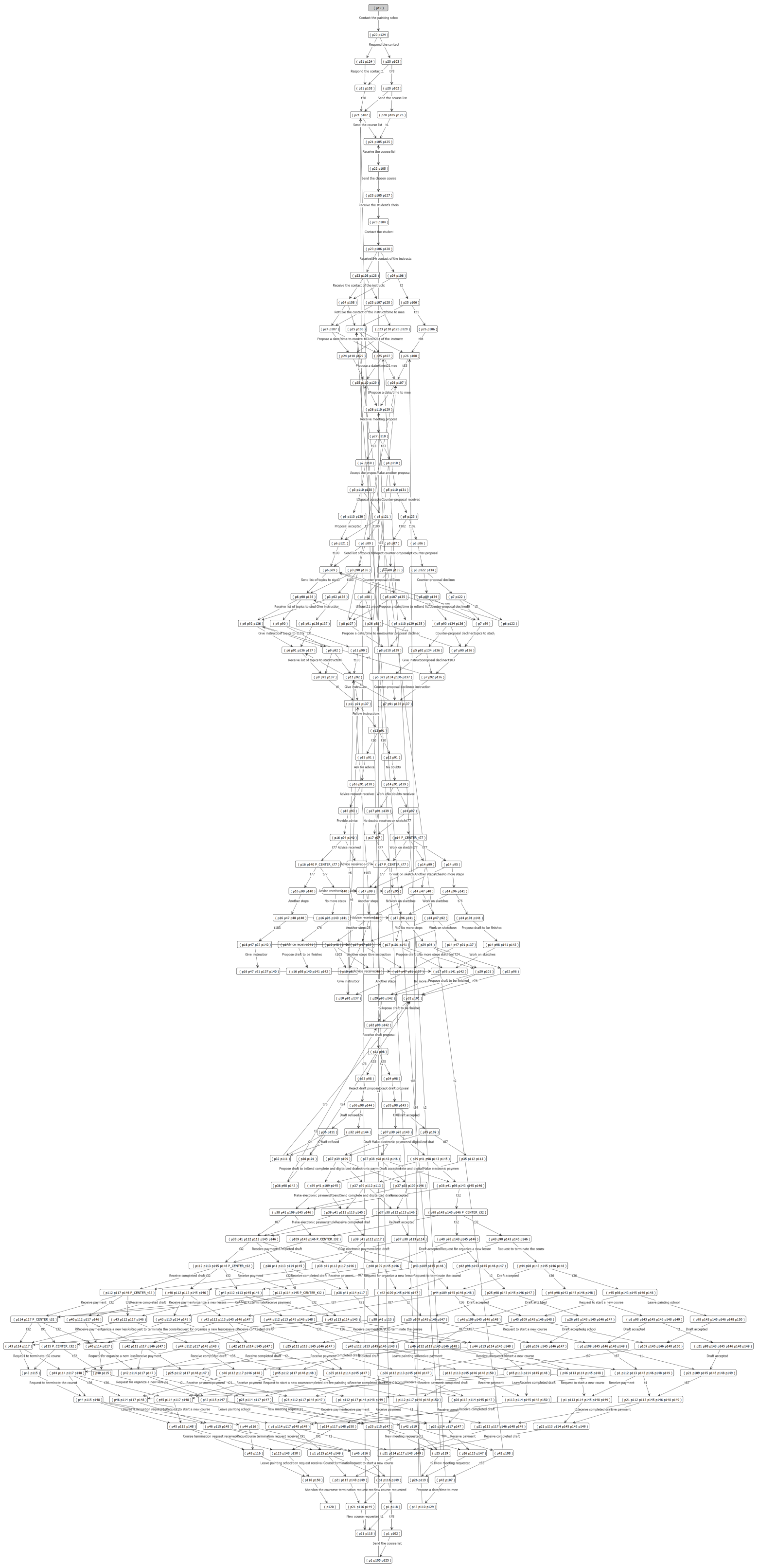
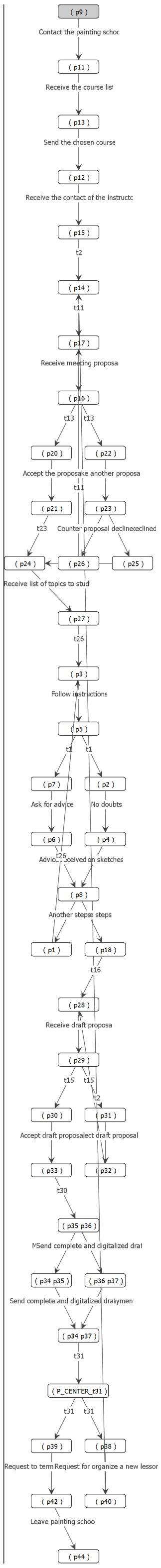
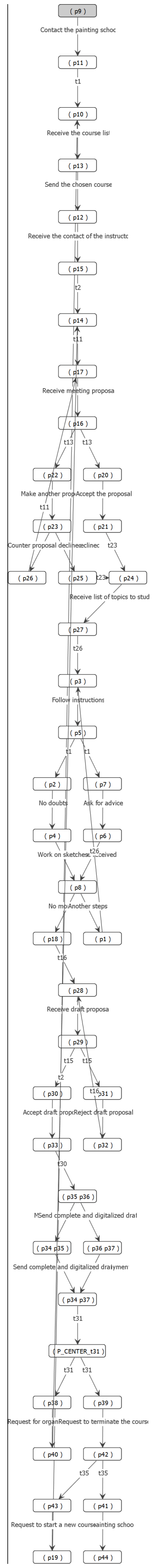


Figure 19: Complete Coverability Graph of the Workflow Net With Variant

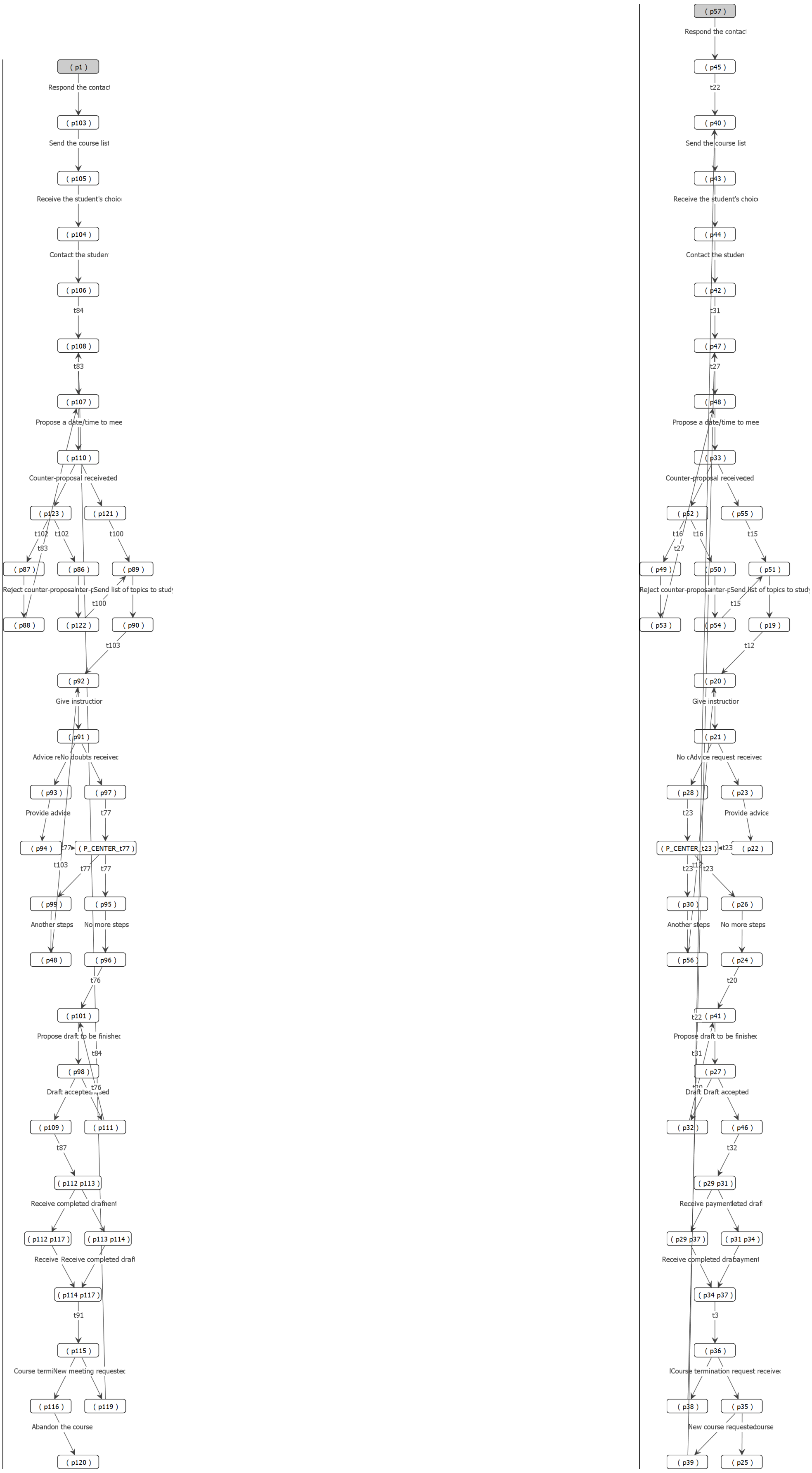


(a) User Net Without Variant



(b) User Net With Variant

Figure 20: Comparison of User Net with and without Variant



(a) School Net Without Variant

(b) School Net With Variant

Figure 21: Comparison of School Net with and without Variant