Project Report

Business Process Modelling

Data Science and Business Informatics

Gaetano Antonicchio
ID: 616685
Academic Year 2020/2021

Abstract

The scenario presented by the problem deals with the design of a process involving a patient that wants to start a rehabilitation therapy. Before starting the analysis, it was important to identify all actors involved in the process. I identified three main actors that must interact with each other in other to carry out their respective activities. The actors are:

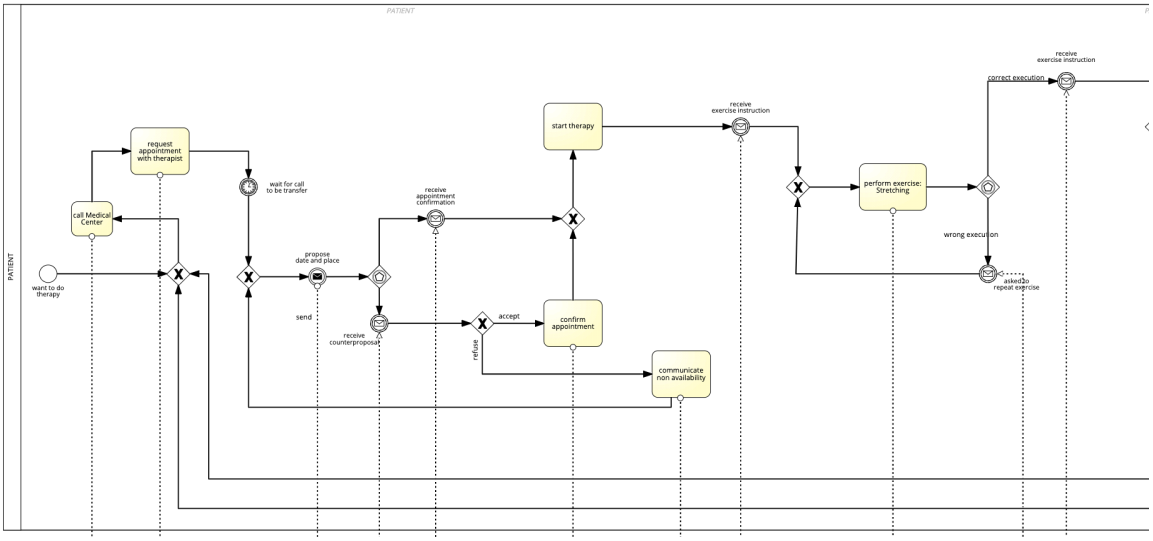
- Patient
- Medical Centre
- Therapist

The process modelling will be presented using the BPMN language for providing a simplified dynamic of the whole process, and Petri Nets for the soundness analysis. The BPMN diagrams were designed in **Signavio**, whereas the Petri nets analysis was conducted on both **WoPed** and **Woflan**.
Regarding the general design choices, since the interaction between the actors is a key point in the whole process, I've decided to construct a **collaboration diagram**. In the representation of the process in BPMN notation, I created three pools (one for each actor) in which the various tasks are performed. Each actor interacts directly with the others, therefore I decided to employ **message flow arrows** to represent the flow of information between the various pools. **Artefacts** were instead used for visually representing objects outside the main process (i.e. database artefacts were used in the therapist pool). The processes represented in the pools, are strictly depended on one another, therefore in order to control the flow of activities and impose dependency, I've decided to use **decision based gateways**. The other gateways used are **XOR split** and **XOR join**. Furthermore, the problem asked to design a process in which the patient has to perform a series of exercises which were not specified. I firstly considered the possibility to design a singular loop task representing a generic rehab exercise, but then I decided to add three cycles representing three different exercises that are usually performed during a rehabilitation session ("stretching", "knee extension" and "step walk").
In the next section, I describe (with the aid of a bpmn diagram) the sequence of activities that are performed in each pool.

BPMN MODEL

1. Patient Pool

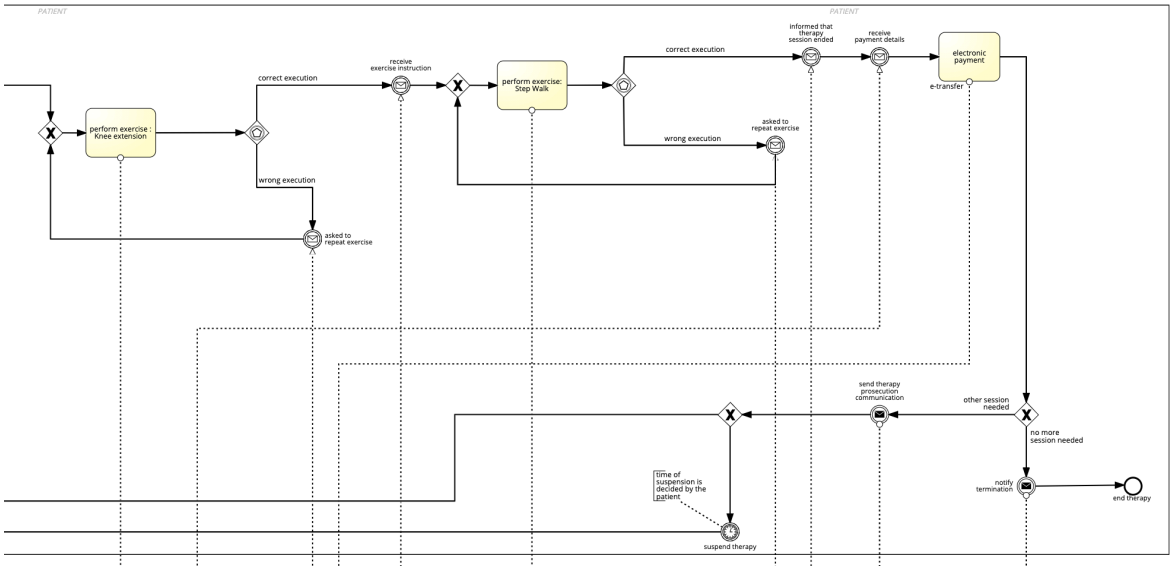
The main pool is represented by the Patient pool, which is the one that starts the whole process. The process starts with the task "*call Medical Centre*" through which the patient gets in contact with the clinic. After this, the task "*request appointment with therapist*" is enabled and the Medical Centre, once it receives the communication through an **arrow flow message** it transfers it to the therapist. This task is fundamental because it activates the therapist pool, which plays an important role in the continuation of the whole process. In order to represent the waiting time due to the call transfer, I've included an **intermediate time event**. After the patient gets in contact with the therapist, he sends a data and place proposal. At this point the best way to represent the control held by the therapist, (who can accept the appointment or send a counterproposal) is to add an **event-based gateway**. If the patient receives a counter proposal, the **XOR split** allows the patient to decide to confirm the appointment with the new date and place provided by the therapist, or to communicate the non-availability and repeat the process again, until both resources agree on a common date and place. The **XOR join** placed before the task "*propose date and place*" is essential for the repetition of this sequence of activities.



Once the therapy session starts, I've added three cycles that represents the three exercises proposed by the therapist.

The **catching intermediate message events** inserted in each cycle of exercises, are used to represent the instruction received by the patient and the feedback provided

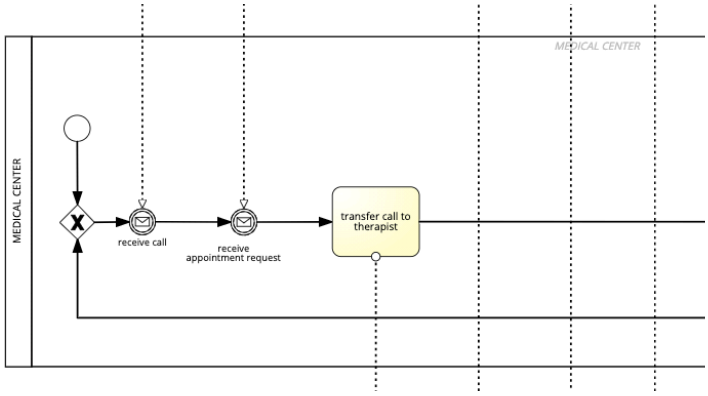
by the therapist. Each cycle can be repeated based on the result of the evaluation made by the therapist. Even in this case, **event based gateways** were needed to represent this scenario, as well as a **XOR join** before the task "*perform exercise*" for the repetition of the process. If the execution is correct the patient waits until the therapist provides new instruction on the next exercise. Once all the series of exercises are executed correctly, the patient is informed that the session ended and soon after he receives the payment details by the Medical Centre. The payment is done by an electronic payment, and the money transfer is represented by a message flow arrow labeled "*e-transfer*".



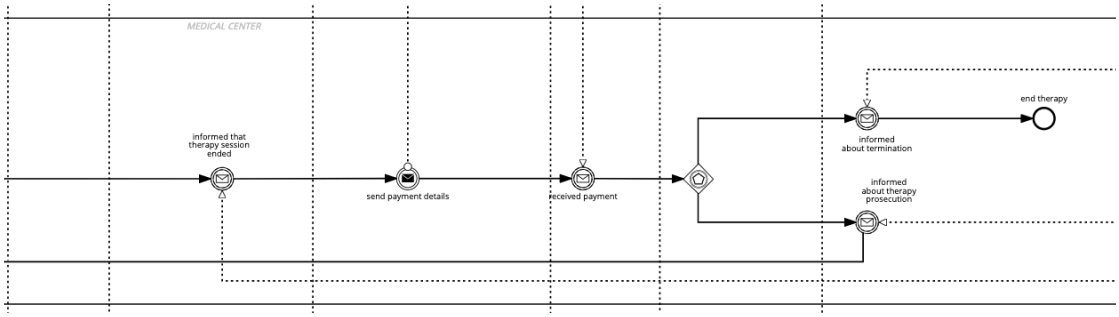
Once the payment is done, the first **XOR split** allows the patient to decide if he needs more sessions or not. If he doesn't, he notifies the Medial Centre and the therapy ends. Otherwise, he informs the Medical Centre that he needs more sessions. At this point, I've added a **XOR split**, to allow the patient to decide between booking another appointment right away (which is represented by a directed arc connected to the **XOR join** before task '*want to do therapy*'), or suspend temporary the therapy and contact the therapist on a later stage. The suspension time, was represented by an **intermediate time event**, whose time is decided exclusively by the patient. Once this time expires, the process starts again, through the connection to the **XOR join** placed before task '*want to do therapy*'. The connection between **XOR join** at the beginning and **XOR split** at the end, are necessary to guarantee the possibility for the process to be repeated. In fact in case of continuation, the patient can perform again the task "*call Medical Centre*".

2. Medical Centre Pool

In the Medical Centre pool are executed all administrative tasks. The process was designed two times. In the first design, the process started with a **starting message event** "*receive call*" and it ended with the task "*end therapy*", without the possibility of repeating the process in case the patient wanted to continue the therapy. This initial design, although sound, it generated unbounded places once it was encoded (along with the others) in a workflow system. Therefore, after being pointed out by Woflan the points that endangered the soundness, I modified the *bpmn diagram* (and the related *workflow net*) in order to guarantee soundness to the whole system, without altering the activity flows.

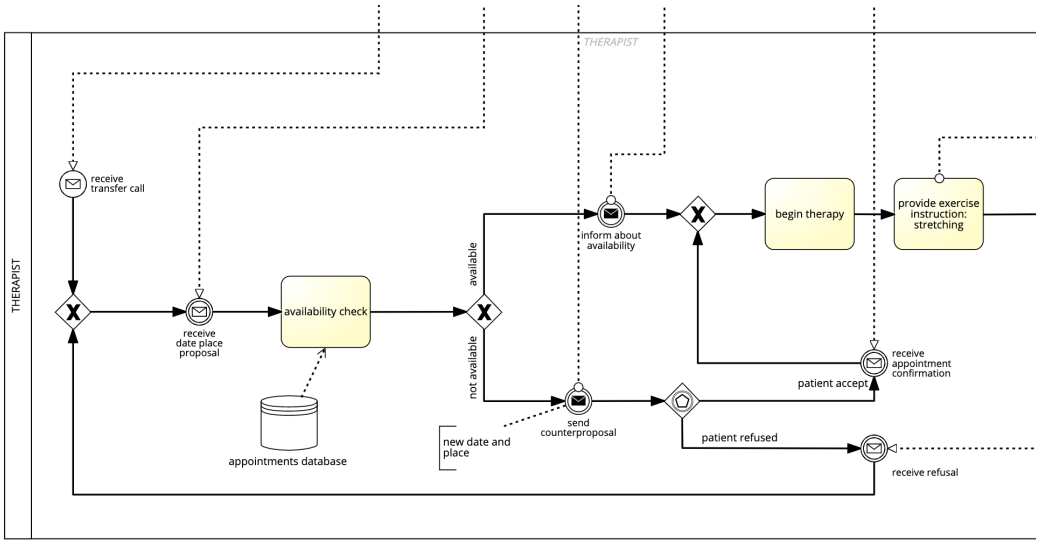


The final (revised) model I propose, starts with an initial event that activates the task "*receive call*", which is preceded by a **XOR join** for allowing the process to be repeated until the patient decides to terminate the therapy. The Medical Centre then receives an appointment request by the patient, represented by a **catching intermediate message event**. The Centre then transfers the call to the therapist who can set an appointment with the patient. Under a global perspective, from this point on, the tasks performed by the Medical Centre are "paused" until the therapist communicates the Centre that the session with the patient has ended. At this point, the Medical Centre sends the payment details to the patient, and waits for the payment to be correctly received. Once the payment is collected, the **event based gateway** represents the decisions made by the patient in the last step of this process. If the patient communicates the termination of the therapy, the process of the Medical Centre ends. Otherwise, the patient communicates his intention to continue the therapy and the process starts again with the Medical Centre that waits to receive a call from the patient in order to book another appointment.



3. Therapist Pool

In the therapist pool, the process starts after an incoming message from the Medical Centre is received. The **XOR join** allows the process of defining the date and place, to be executed infinitely many number of times until both actors (therapist and patient) agree on the same date and place. After the therapist receives the proposal, he checks his availability through the database (represented by the database **artefact**), and then takes a decision through a **XOR split,** which allows him to either accept the appointment or propose new date and place.



If the therapist sends a counterproposal, the process becomes dependent on the decisions made by the patient.

This was represented by an **event based gateway**. When both the participants come to an agreement, the therapy session starts.

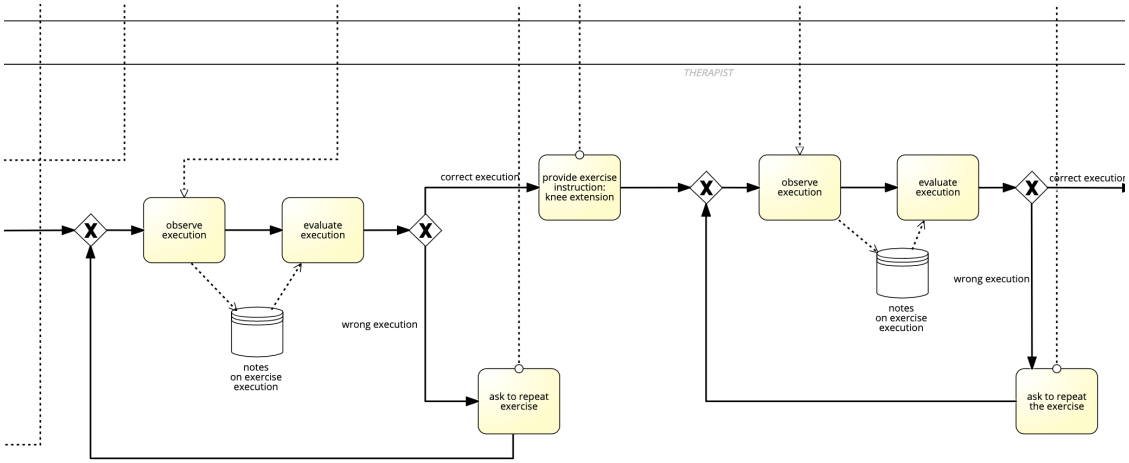
I provided **3 cycles** of exercises during which the therapist first sends the instructions about the execution, and then performs the task "*observe execution*" through which notes about the correctness of the exercise are taken and then used for performing the next task "*evaluate execution*".

After this task, the therapist can make a choice which has influences on the patient process.

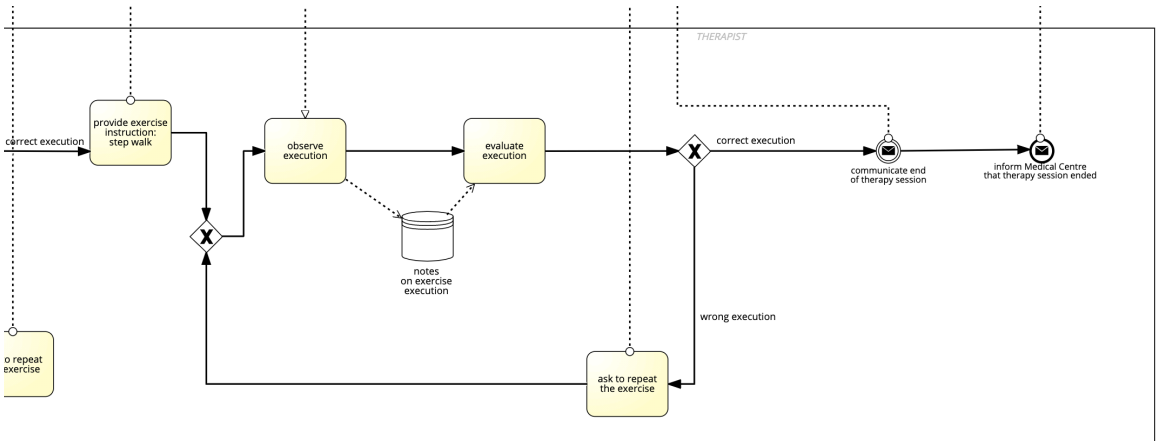
In fact the two are strictly dependent on one another.

If the exercise's execution is not correct, the therapist asks the patient to repeat the exercise once more, and the process goes back to "*observe execution*".

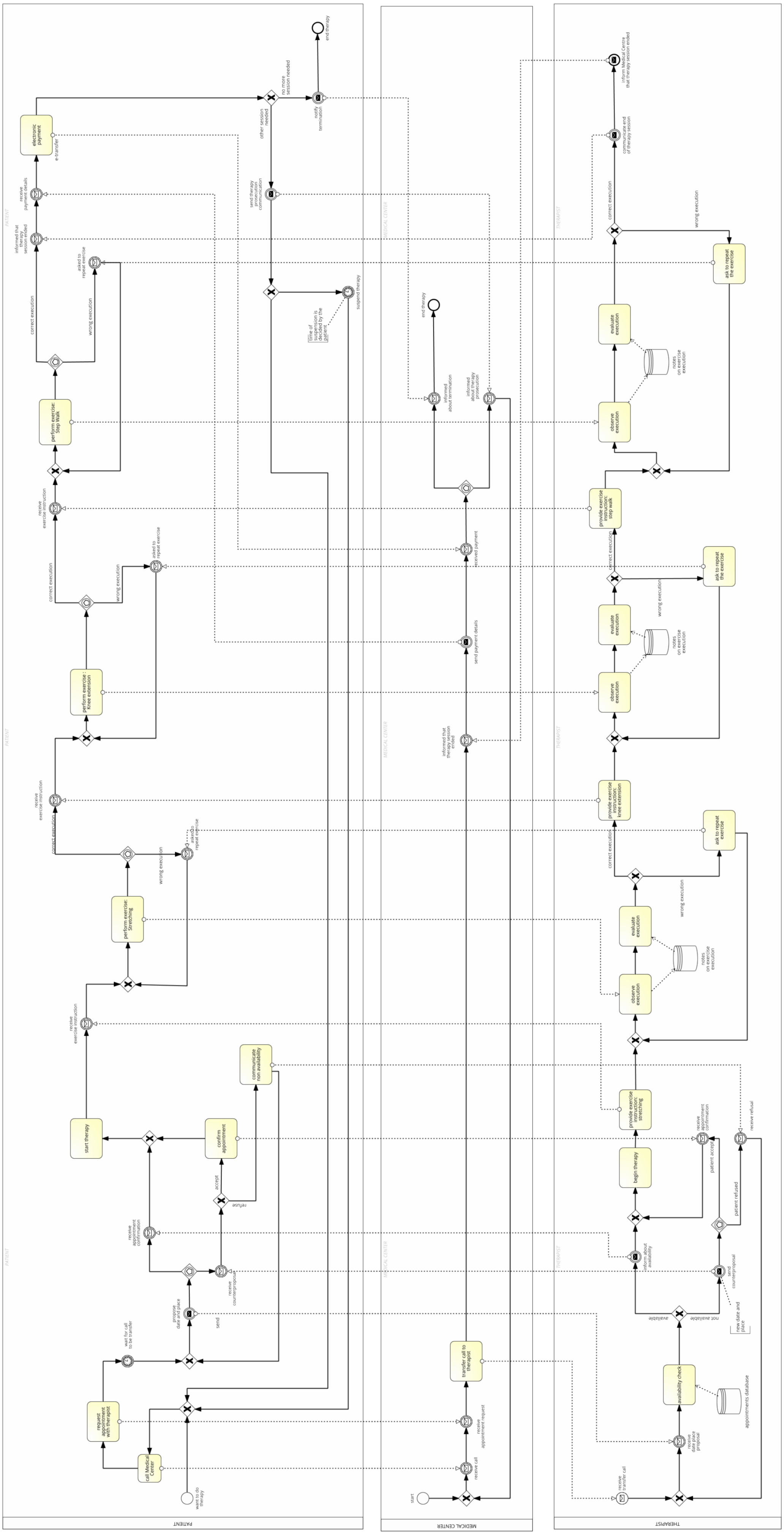
This dynamic was designed for all three exercises provided.



After the all exercises are correctly performed, a **throwing intermediate message event**, represents the therapist that informs the patient that the session ended. Right after this task, in order for the Medical Centre to bill the patient, a communication has to be sent by the therapist for informing about the termination of the session (I decided to don't design a parallel communication since usually the patient is the first actor to receive the information).
This was represented by a **throwing end message event** which ends the process in the therapist pool, and allows the Medical Centre to continue continue the process in its respective pool.



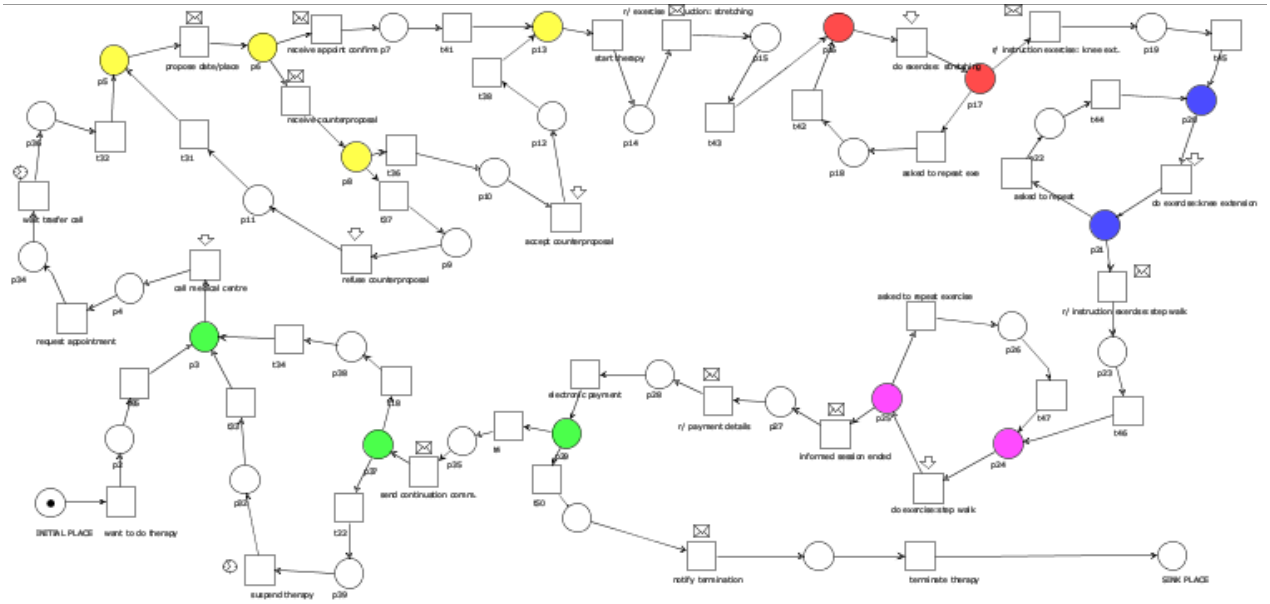
In the next page, I show the whole process in BPMN notation.



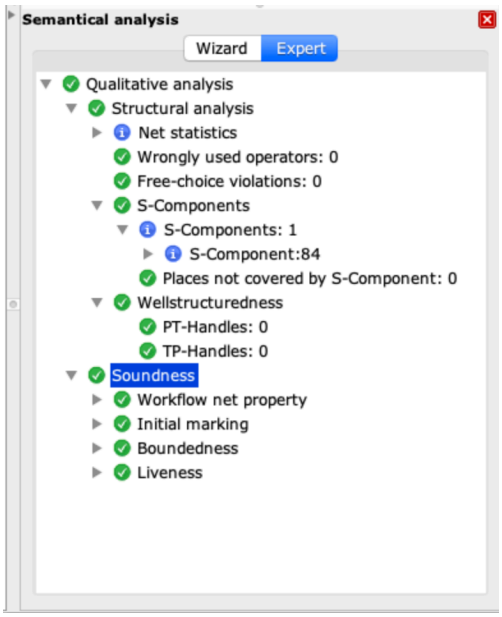
PETRI NETS - ANALYSIS

The **BPMN diagrams** were translated in **workflow nets** in order to carry out the **semantic analysis**. The translation process was done in three steps: first I added places in place of the arcs, then I encoded the gateways. Lastly, I de-sugared the net and I've added an initial and final place for evaluating it in terms of workflow nets. For all the nets analysed, is possible observe that since the finiteness of the process in guaranteed, the *coverability graphs* will coincide with the *reachability graphs*.

1. Patient



Structural and Soundness analysis:



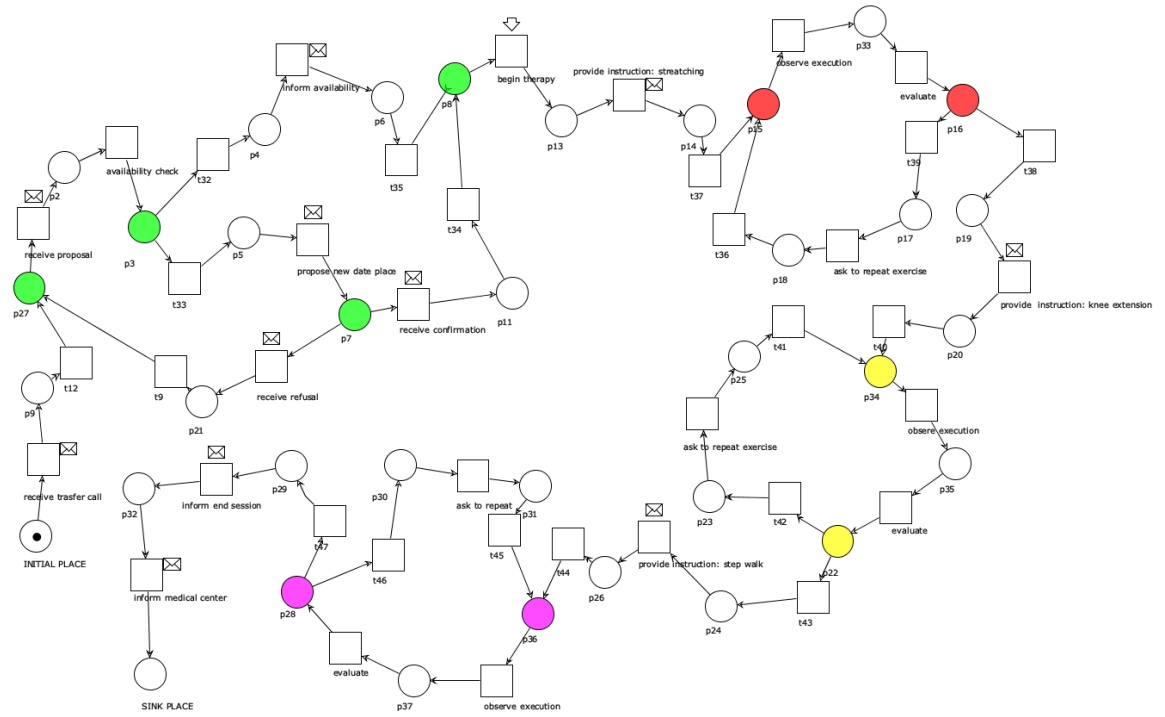
- A Workflow Net because:
  - ● INITIAL PLACE = ∅
  - SINK PLACE ● = ∅
  - ∃ path from INITIAL PLACE to SINK PLACE
- N is an **S-net** therefore it is **sound**
- Deadlock free, bounded and free-choice
- Well-Structured (there are **no PT- or TP handles**)
- S-coverable by 1 S-component which covers all 84 elements of the net

The net has:

- **39 places**, **45 transitions** and **90 arcs**

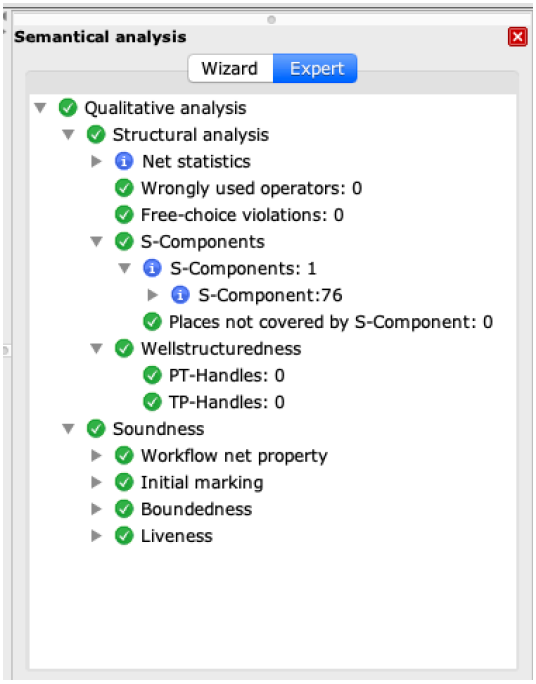
The *coverability graph* has **36 vertices** and **42 edges**

2. Therapist



Structural and Soundness analysis:

- WorkFlow Net
- Free choice
- N* is **strongly connected** and marks only the initial place, therefore is live
- **Deadlock free**, **Bounded** and **Safe**
- N is **well structured** because there are no PT/ TP handles
- N is **safe and sound** (this can be immediately checked by the fact that N and N* are S-nets)
- **S-coverable** by 1 S-component that includes 76 elements

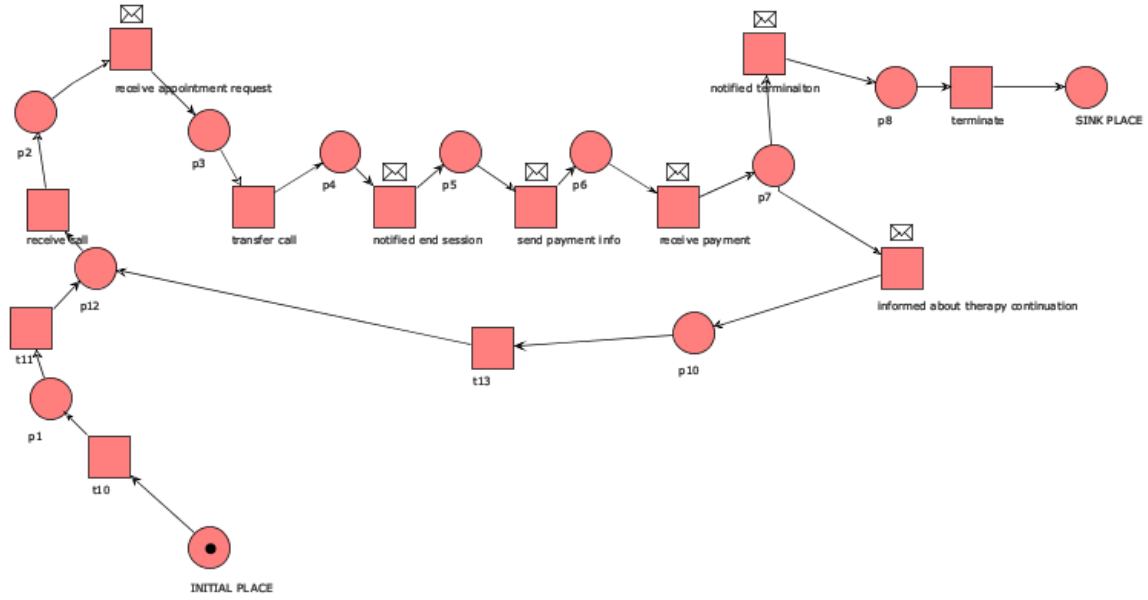


The net has:

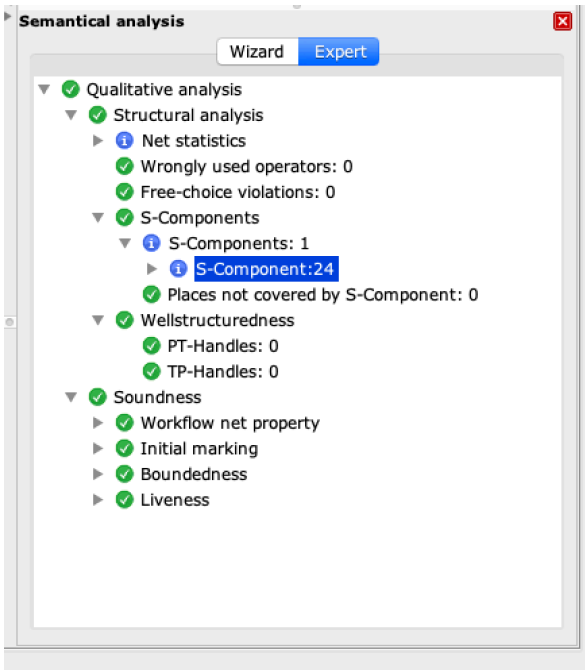
- **36 places**, **40 transitions** and **80 arcs**

The c*overability graph* has **34 vertices** and **38 edges.**

3. Medical Centre



Structural and Soundness analysis:



- Workflow net
- Free choice
- **Deadlock free** and **Bounded** and **Safe**
- All **soundness** requirements are satisfied, furthermore we can see that the N is an **S-net**
- If the sound blocks are iteratively substituted with single transitions is possible to conclude that the workflow net is **sound by construction**. This can be immediately checked by the fact that the net is composed by sound blocks and one single iteration.
- **S-Coverable** by 1 S-component with 24 elements (shown in red)
- **Well-structured** (No PT-TP handles)

The net has:

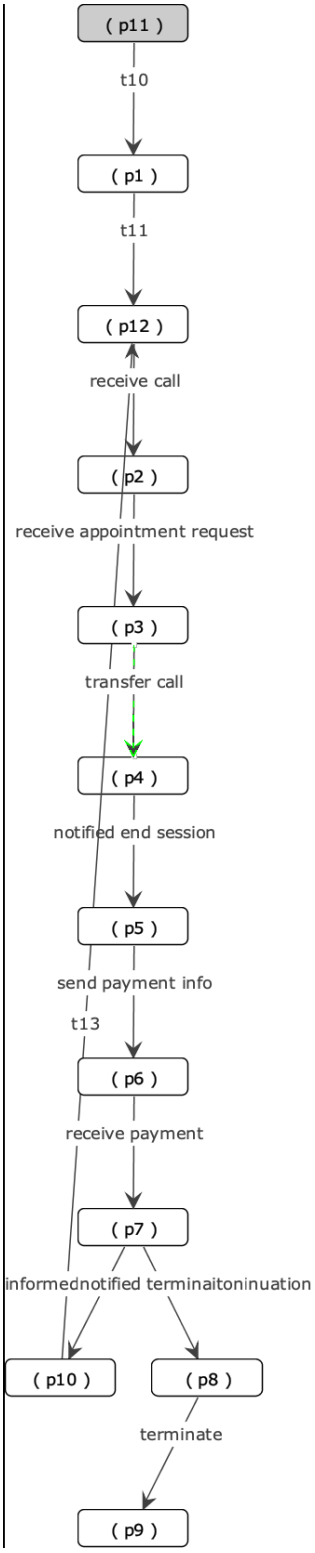
- **12 places**, **12 transitions** and **24 arcs**
- **One positive S-Invariant** of the form [kk..k] and **Semi-positive T-Invariants**

The c*overability graph* has **12 vertices** and **12 edges**.

I show below only the coverability graph for Medical Centre due to space limitation.

The *coverability graph* coincides with the *reachability graph*. We can see that since it is finite, it is bounded. The **behavioural property** of liveness cannot be satisfied for N, as after firing t10, p11 becomes dead. However if analyse the graph of N* all the behavioural properties are satisfied. There is one **cycle** that starts at place p12 and moves top down along the reachable markings, until the state where we have only one token in p7 is reached. From this point on we have the choice of repeating the cycle by putting the token back in p12 after firing transition "*informed about therapy continuation*" , or to terminate the process by firing transition "*informed about termination*" which puts a token in p8 which can fire only one transition "*terminate*".
After this step, a token is put in place p9 and the process is completed.

This graph shows also that **option to complete**, **proper completion** are satisfied. Once a token reaches place p9 there are no tokens left in other places. Furthermore, the only way to put the token back in p11 is through the "*reset*" transition. There are also **no dead tasks** since there is at least one arc labeled with the transition of the net.
Hence, N is sound.

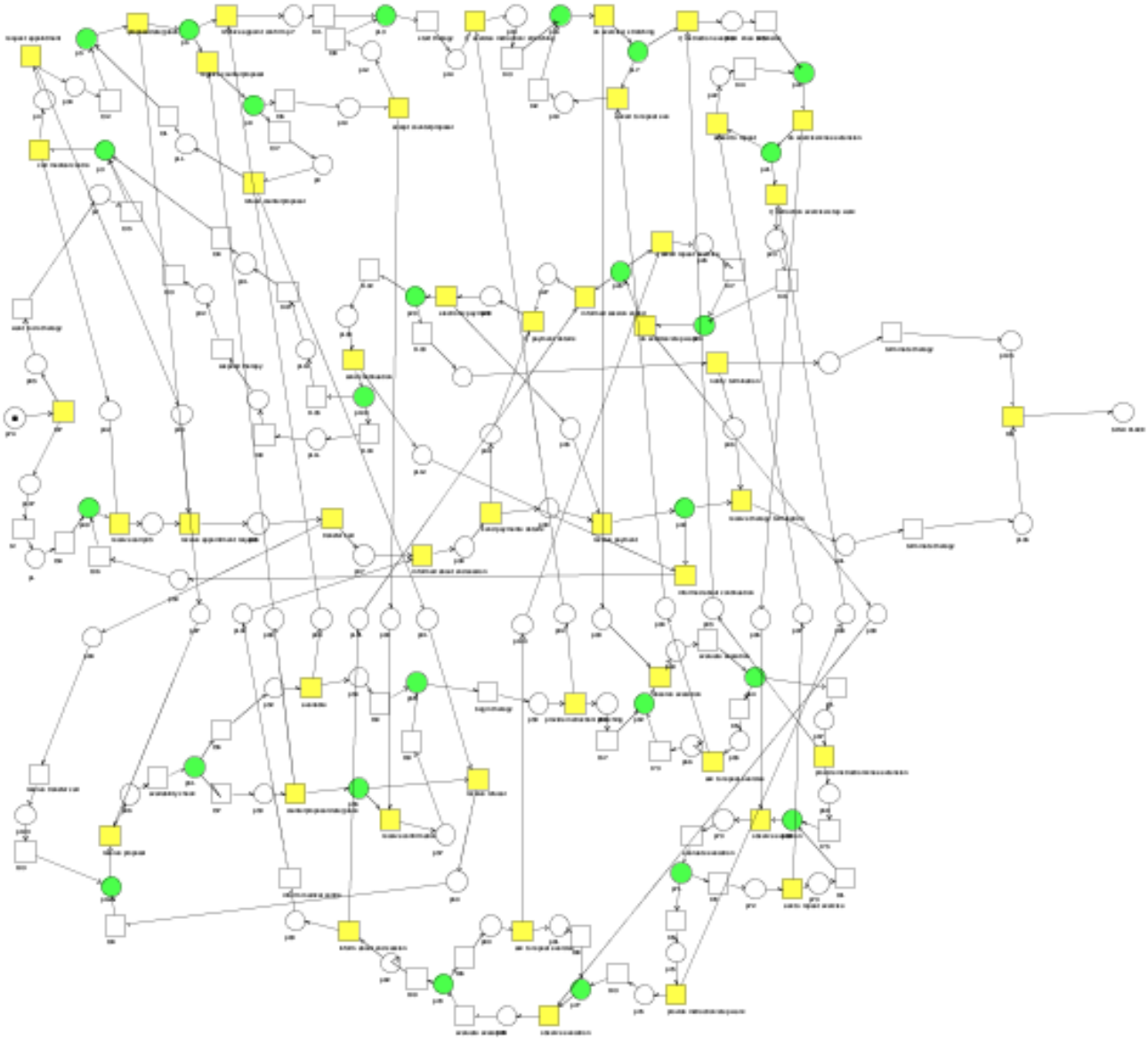


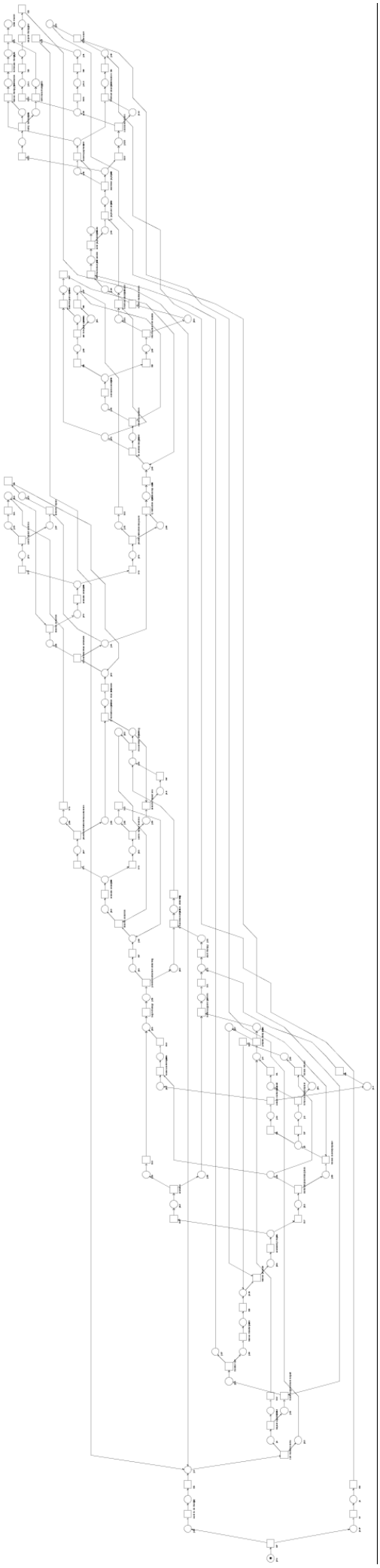
Conclusion

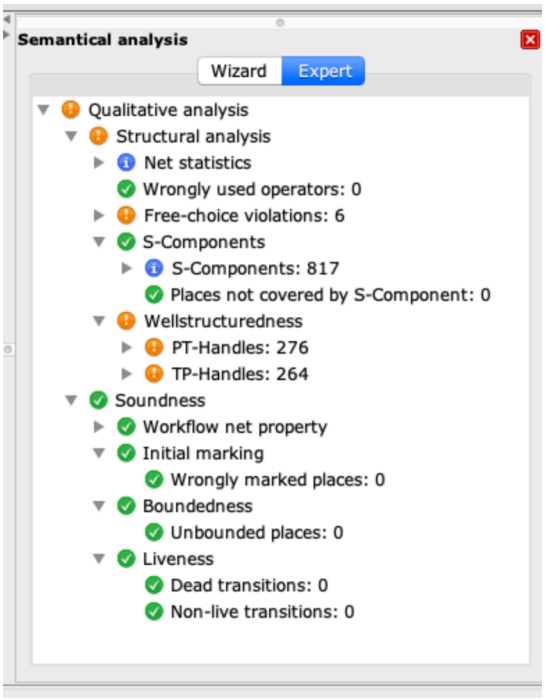
The workflow nets were transformed in **workflow modules** by adding places in their respective interfaces. The obtained modules were **structurally compatible**, therefore I joined them in order to inspect the soundness of the final **workflow system**. In my first attempt, the workflow system resulted unbounded at place p40, p39 and p45 (as described in the Medical Centre pool).

Therefore since it didn't respect proper completion, the system was not sound. However, thanks to the error sequence provided by Woflan, I was able to locate the problem and fix it. By adding a **XOR join** before the task "*receive call*" in the Medical Centre pool, and connecting it to the task "*receive continuation communication*", I was able to bound the system which is now **safe and sound**.

Below I show the *workflow system* designed in WoPed, and on the next page, I show the same with a more compact and readable design, along with the semantical analysis results provided by WoPed.







statistics

places	111
arcs	246
transitions	100