BPM project report

Carlo Alberto Carrucciu
Francesco Salerno

Business Process Modelling 2019/2020

1 INTRODUCTION

The scenario presented by the problem deal with the planning of a museum trip for a school class. While modelling the problem solution we have to care about different case and diverse possibilities to carry out the whole process. Firstly, we need to clarify the presence of three different actors that must interact in order to complete in a correct way the procedure. These three actors are:

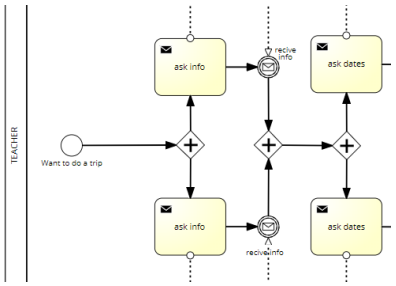
- Teacher
- Museum
- Bus Company

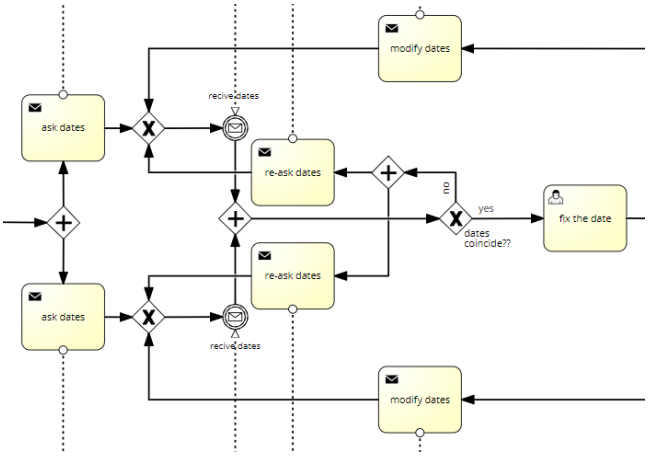
The process modelling will be presented using the BPMN language and the tools used to work over the project are Signavio useful to model the three connected pools and WoPed and Woflan for the workflow analysis . The first activity that lets start the process is put in action by the teacher, who sends an information request about the costs of the visit and availability of the dates to the museum booking office, and at the same time she contacts a bus company to obtain a bus rent preventive. Once teacher receives back the information by the two others actor she decides when to book the visit and consequently she rent the bus for the same dates. As the problem require we put effort in modelling such a way to structure the model in order to let the dates coincide before the booking is completed. However a variant is allowed due to the fact that once the date is confirmed until two days before the trip time the number of the reservation may be modified or the date can be changed or moreover is possible to cancel the whole trip. The number of the modifications has not a given limit until the process is completed or cancelled. The last request is to model the process in order to let the teacher books another time the trip only if the previous one was completed.


2 BPMN MODEL

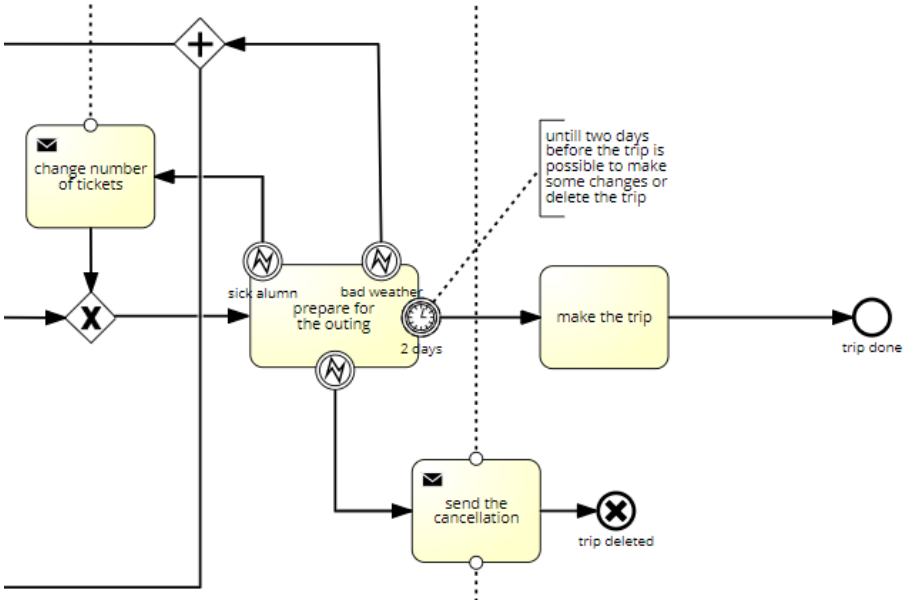
What we are going to present is the implementation of the model for the problem before described. The language used to perform it, is BPMn which has been built over the Signavio Academic Cloud Platform shared by the components of the group. Each actor is modelled as an individual pool.

The main actor is the teacher, which is linked with the other two actors by the flow messages Instead the bus pool and the museum pool never communicate directly together along all the process. The process begins in the teacher pool, with the teacher that want to organise an outing to a museum for one of her classes. As the firsts, in which the teacher ask information, most of the tasks, are communication task (message task) and for the teacher run in a parallel way, one flow to the museum and one to the bus company; obviously in order to respect the logic sequence. The two "ask info" generate the starting event in both the other two pools, that is as starting message event and so are immediately involved in the collaboration, all the three actors. The process continues in the teacher pool, more or less always in parallel, also asking dates and receiving dates for the availability of the resources . In this case it's necessary that the dates coincide, so we meet a condition that take us to a possible loop. We decided to model the teacher with a *Xor gateway*, and make the other two dependent by the teacher, so modeling that with the event based. In order to respect semantic and generate different event, more activities have been used for the same goal like : ask dates , re-ask dates, change dates ; while only one caption message event for each collaborator has been used in the teacher pool to get the dates.





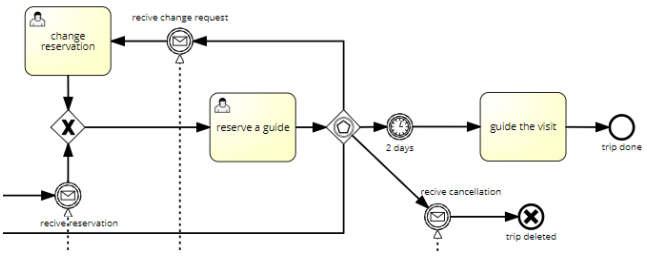
Modify dates comes from the end of the process; in fact after the date are fixed we spoke about the possibility to make some changes. We introduce the activity "preparing the outing" in order to model this scenario, adding different boundary interrupting events.



We have four different scenarios:

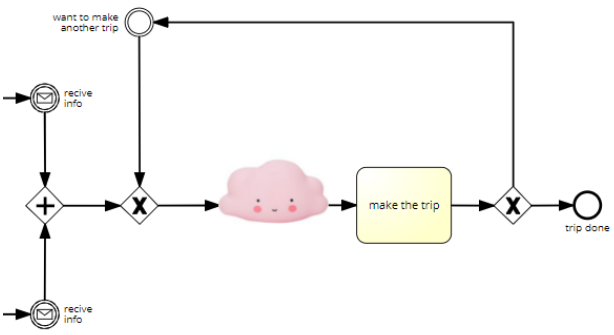
- **Sick alumns** : in this case the loop is short and easy, it's an example of a situation in wich is necessary to modify the number of tickets (take a few ticket less or a few ticket more) ; so assuming the bus it's always the same, teacher will advise in this case only the museum.. this is probably the only asymmetry between the bus pool (and Petri net module) and the museum pool.
- **Bad weather**: In this case instead, it's presented a situation in which both bus company and the museum should be contacted; with this unexpected event, the desire is to change the fixed date. The loop comes back to the "receive dates" event
- **Cancellation** : we don't specify a reason but is mentioned the possibility to cancel the trip and terminate the process with a cancellation end event.
- **2 days**: all this modifies are possible only until two days before the programmed date. After this deadline is not more possible to make changes and we assume the class is so going to complete the visit in a positive manner.

All this occurrences are represented in the secondary pools (as before for the date loop) like an event based gateway, cause all the next tasks depends by the teacher acts.

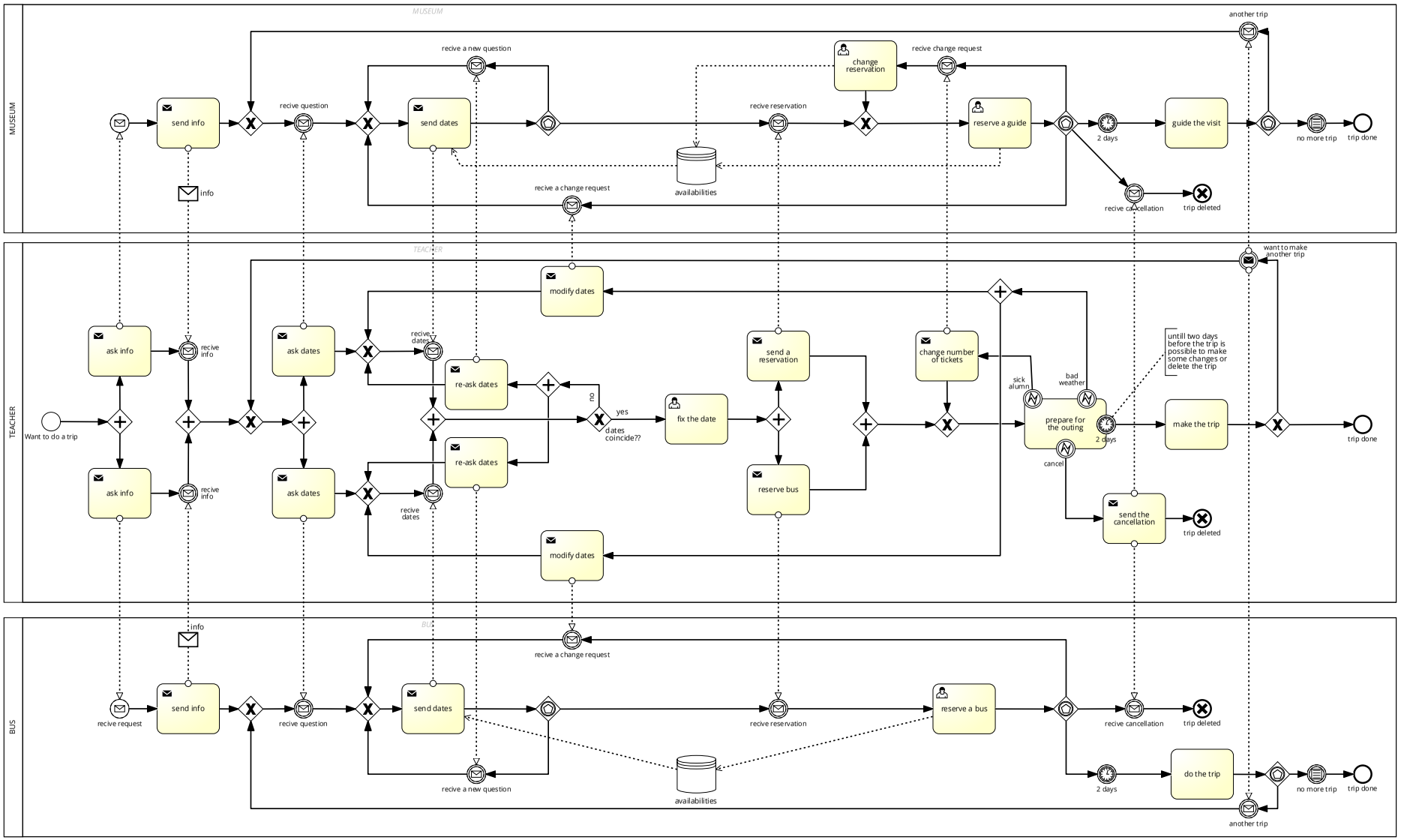


All this occurrences are represented in the secondaries pools (as before for the date loop) like an event based gateway, cause all the next tasks depends by the teacher acts.

At this point the process seems to be completed, but we are going to add an additive loop to allow the teacher, only in the positive case the trip has been done, to reorganize another trip(for this or for another class). So we just add a Xor split between the task "Make the trip" and the "Trip done" end event... that takes the process back to before to ask the dates. In fact the supposition is that now the teacher already knows the information she needs to organise the visit. In the other to pools it corresponds to another event based.

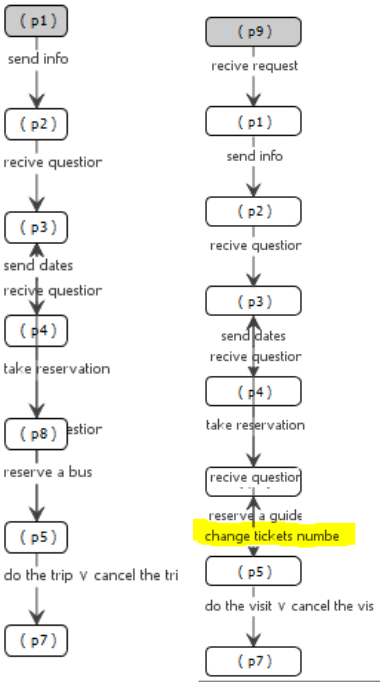


Now the process model can be considered complete. It's implementation using Signavio is showed in the next page.


MUSEUM
another trip
recive a new question
change reservation
recive change request
send info
recive question
send dates
recive reservation
reserve a guide
2 days
guide the visit
no more trip
trip done
info
availabilities
recive a change request
recive cancellation
trip deleted
TEACHER
want to make another trip
modify dates
ask info
recive info
ask dates
recive dates
re-ask dates
send a reservation
change number of tickets
untill two days before the trip is possible to make some changes or delete the trip
sick alumn
bad weather
prepare for the outing
2 days
make the trip
trip done
Want to do a trip
no
yes
dates coincide??
fix the date
cancel
reserve bus
send the cancellation
trip deleted
modify dates
BUS
info
recive a change request
recive request
send info
recive question
send dates
recive reservation
reserve a bus
recive cancellation
trip deleted
recive a new question
availabilities
2 days
do the trip
no more trip
trip done
another trip

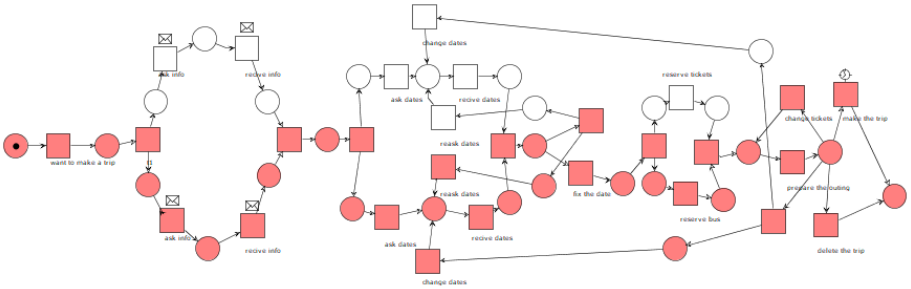
3 WORKFLOW ANALYSIS

After the implementation of the problem using the language of BPMn we moved to an other level of representation in order to be able to analyse some fundamental properties useful to describe our model construction. The language used to perform it is Petri-net and the platform where we worked over is Woped which let us study better the model and verify its correctness. We are going to present each of the three actors in a different paragraph and within each section will be presented the petri net workflow model and the properties analysis tables resulted by Woped . Before that we show the reachability graphs. We didn't add the teacher one because it presented many different possible solutions (precisely 18). Anyway here there are the Bus Company and Museum Reachability Graphs.



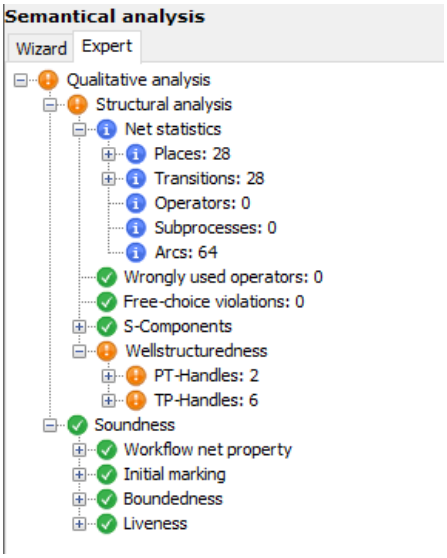
What the images beside shows it's evident. The unique logical difference between the reachability graphs of these two actors is the possibility, in the Museum model, to change the number of the reservation until 2 days before the trip starts. Anyway this change as has been specified in the chapter above can be done as many times as it is required. These have been resulted by the relative Woped function.

3.1 Teacher

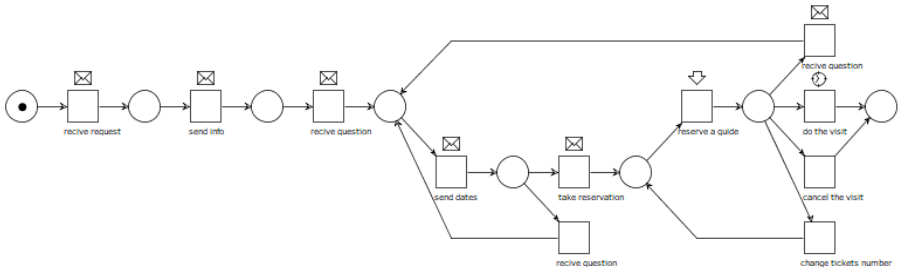


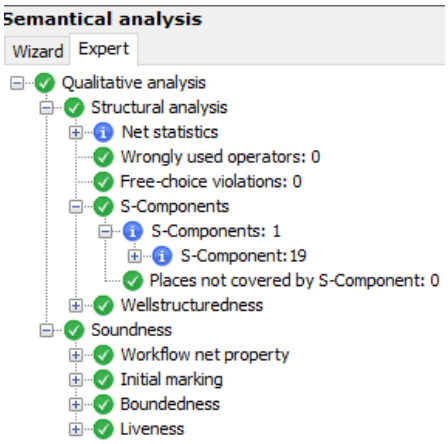
The net is:

- neither an S-system or a T-system
- but it's free choice
- deadlock free
- all soundness requirements are satisfied
- not well handled ans so not well structured.. it contains 2 PT-handles and 6 TP-handles
- S-covered by 8 different S-components (one is shown in the figure as a red coloured track)



3.2 Museum

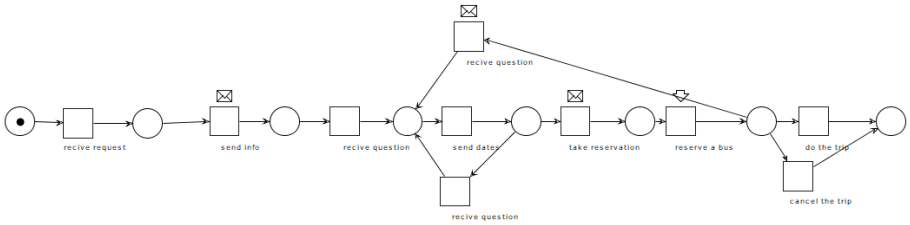


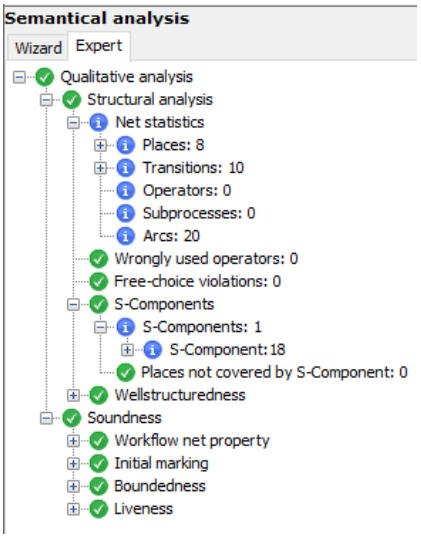


The net is:

- An S-system (but not a T-system)
- so it's free choice
- deadlock free
- all soundness requirements are satisfied
- Well handle and well structured (in fact there are no PT- or TP-handles)
- S-covered by one unique S-component to which all the 19 elements of the network belong

3.3 Bus Company



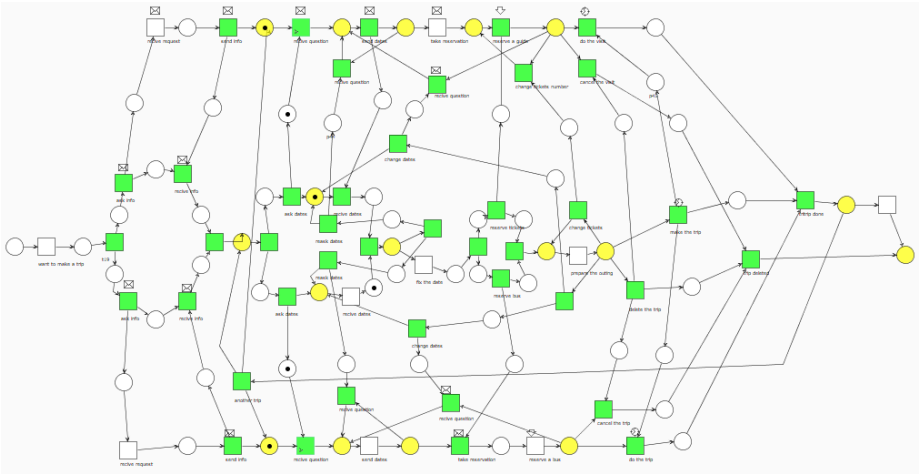


The net is:

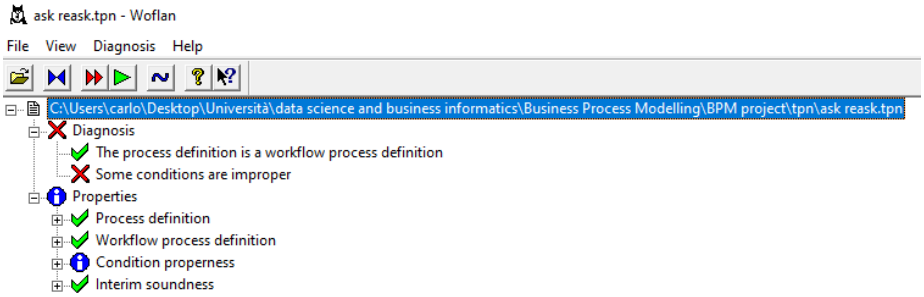
- An S-system (but not a T-system)
- so it's free choice
- deadlock free
- all soundness requirements are satisfied
- Well handle and well structured (in fact there are no PT- or TP-handles)
- S-covered by one unique S-component to which all the 18 elements of the network belong

4 CONCLUSION

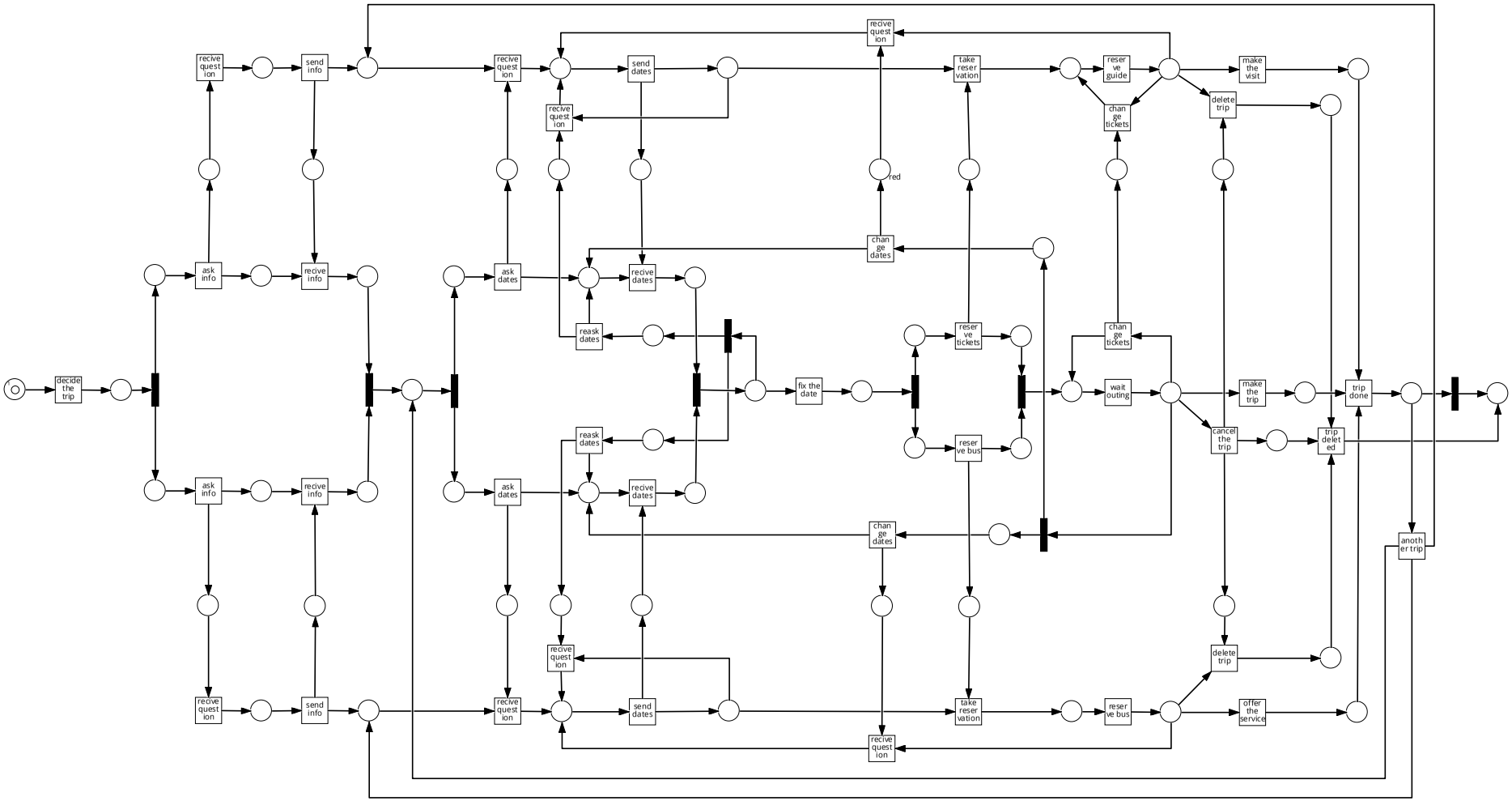
Adding places for the input and output documents, we are finally able to connect the precedent analyzed workflow modules, and create the entire net corresponding to the *bpmn colaboration diagram* first designed. Also here, in the good case the trip has been done, there's a possibility to repeat it, sending three tokens back to the correct position in the process.



Using Woped to analyzed this large net is still impossible; but some good results is obtainable importing the net on Woflan, a software for analyze Petri net.



In conclusion the whole seems to be sound and also playing the token game on Woped (as in the image) is possible to verify the soundness by simulation. In the last page is attached the whole workflow net designed by Signavio, not useful in the analysis but easier to understand by inspection.


decide the trip
ask info
ask info
recive quest ion
send info
recive info
recive info
recive quest ion
ask dates
ask dates
recive quest ion
send dates
recive dates
recive dates
recive quest ion
reask dates
reask dates
fix the date
chan ge dates
chan ge dates
recive quest ion
send dates
recive quest ion
take reser vation
reser ve tickets
reser ve bus
take reser vation
reser ve bus
reser ve guide
chan ge tickets
chan ge tickets
wait outing
make the visit
delete trip
make the trip
cancel the trip
delete trip
offer the service
trip done
trip delet ed
anoth er trip
red