

# Syntax error

Assignment.

You will have to go through a large number of process models. For every model you will have to determine which syntax errors were made. This document describes the modeling language that was used and which syntax errors exist. The models are numbered and the errors are coded. You will get a form with the model numbers, which you will have to complete with the error codes.

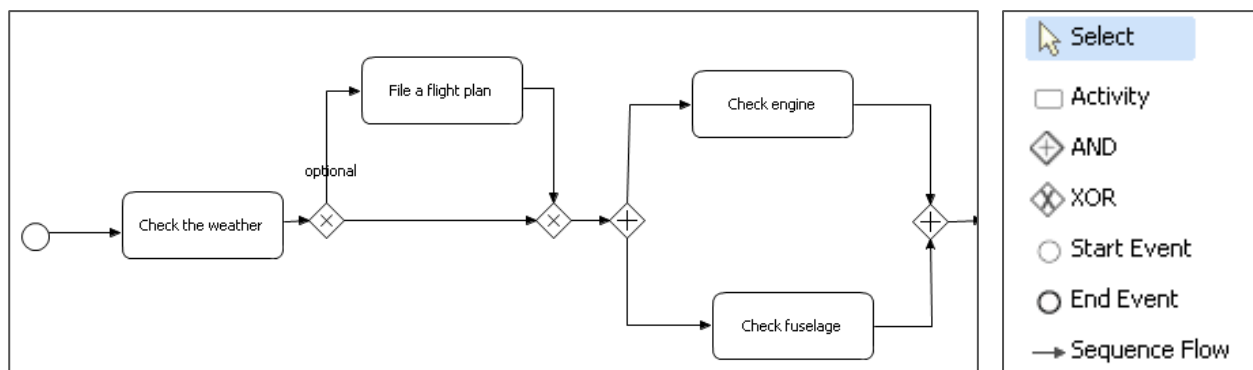
Thank you for your cooperation!

Jan Claes

## 1. Business Process Model Notation

The modeling sessions were performed using a simplified modeling language inspired by BPMN.

- The start of the process is represented by a **start event**
- The end event(s) is/are represented by (an) **end event(s)**
- Every step in the process is represented by an **activity**
- A decision is modeled by an **XOR split and XOR join** gateway
- Parallel paths are modeled by an **AND split and AND join** gateway
- Order of activities and gateways is represented by **sequence flow** edges



## 2. Syntax errors

Below is a list of syntax errors and a code. The last column displays a short explanation of the selected code. It might help you to remember the codes more easily.

- Note that one type of errors can occur multiple times. In that case, please specify the number of times that error was observed (e.g., 4W means that four time a wrong type of join gateway was used to join split paths).
- Note that one model can suffer from multiple syntactical errors. In that case, you have to mention all codes that apply on the model next to its number.
- However, it is not important in which order you put these codes on the form.
- If you observe any other syntactical errors in the process model, please use the code '>' somewhere and describe the error next to this symbol.

Syntactical error	Code	
Contains no end event (but does contain a start event)	0	(0 end events)
Contains no start event	0s	(0 start events)
Contains a start event in the middle (ingoing edge on start event)	B	(between)
Contains an end event in the middle (outgoing edge on end event)	B	(between)
There are multiple end events	E	(multiple ends)
One, but not all of the paths are not closed (missing end event?)	P	(path not closed)
Contains no split gateways at all	S	(no splits)
Forgot some, but not all split gateways	F	(forget some)
Contains no join gateways at all	J	(no joins)
Contains and, but no xor join gateways at all	Jxor	(no xor joins)
Contains xor, but no end gateways at all	Jand	(no and joins)
Forgot some, but not all join gateways	G	(forget some)
Forgot xor join gateway	Gxor	(forget some)
Forgot and join gateway	Gand	(forget some)
Forgot xor gateway at end event	Ge	(forget some)
One gateway combines a join and split feature	C	(combination)
Wrong type of join combined with a certain split	W	(wrong type)
Gateway with only one ingoing and one outgoing edge	1	(1 edge in/out)
Wrong nesting of gateways	N	(wrong nesting)
AND and XOR are joined together in one join gateway	T	(joined together)
Forgot join gateways in case of iterations (edges that go back in the model)	I	(wrong iterations)
Some edges are missing (space between two items, but no edge)	M	(missing edges)
Used a start event instead of a gateway	GWs	(gateway start event)
Used start event as a data object symbol	DB	(database)
Used start event where some input is expected (as a message event)	IN	(input)
Modeled Petri-net style using start events as places	PN	(Petri-Net)
Used empty activities	-	(empty)
Other	>	(describe)