

## Reading: Make your process flows easy to read

Ensure your process flows are easy to understand for a variety of stakeholders

*Note: this article is not directly derived from the BABOK Guide and is a supplementary material. Information contained in the article serves for professional development purposes but is not mandatory to pass IIBA exams.*



Photo by [Campaign Creators](#) on [Unsplash](#)

### Problem statement

In this article, we are looking at a popular problem: “**How might we make our process flows easy to read?**”. A question that many BAs may ask themselves when in need to present a business process to a variety of stakeholders.

We will use a UML activity diagram as a starting point and will review a few tricks to make them awesome.

### Basics — why activity diagram?

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

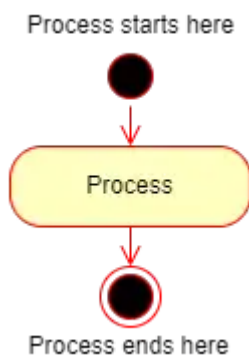
In the UML, activity diagrams are intended to model both business process and computational (automated) processes.

When dealing with people not familiar with business process modelling, a simple notation would always be a preference. A specialised tool like BPMN although being more sophisticated is often harder to grasp especially when featuring the full icon set (a common criticism for the notation).

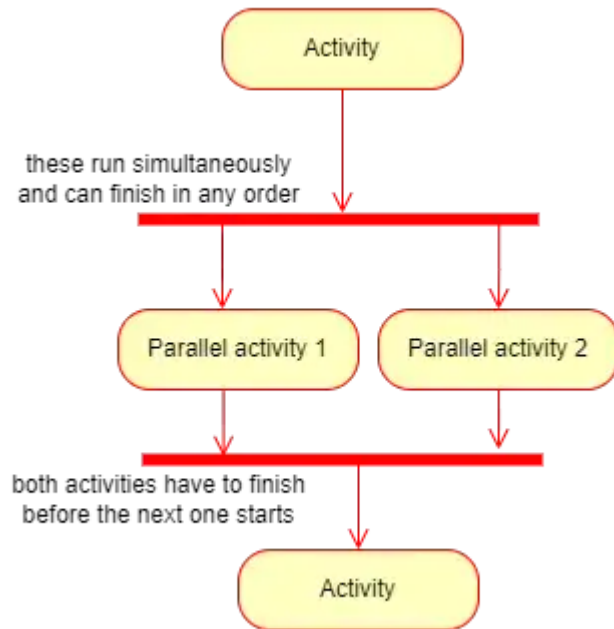
However, advice from this article is not specific to UML and can be applied to other notations, e.g. flowcharts, simplified BPMN, and even Universal Process Notation (UPN).

### Good practices

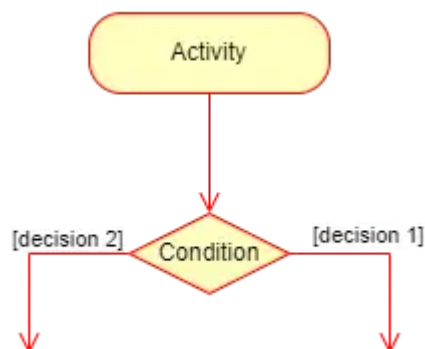
**1. Clearly** indicate start and finish of a process. In UML — use a node. A process always starts with a trigger(s) and it flows till it gets a result(s). Make sure they are visible and clearly defined in your models.



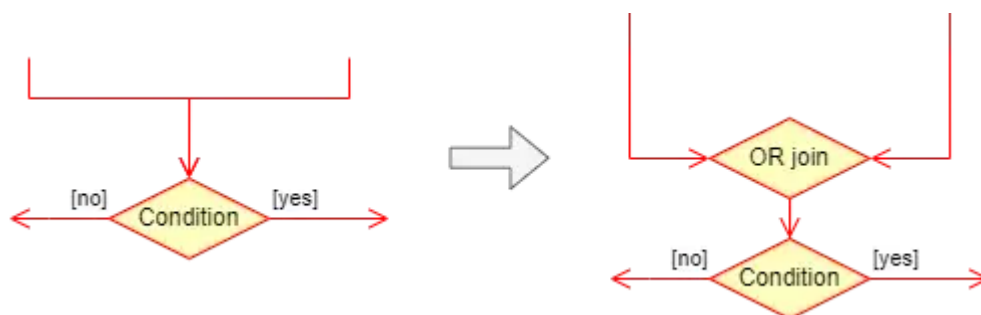
**2. When** the process requires actions performed in parallel, make sure to use a proper fork icon and always join your flows back. Those joins are the orchestration points in your process where multiple streams need to converge together for a process to continue. Not joining the forks in a logical manner will leave gaps in the flow and may cause confusion.



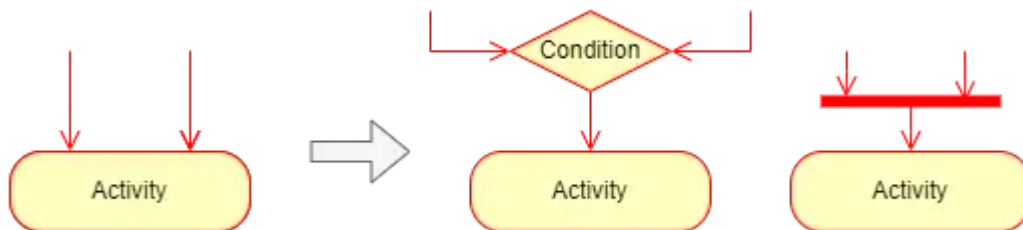
**3. When** your process includes a decision point (only one of multiple potential future actions can happen), those decisions should always be marked with labels to determine which path to chose.



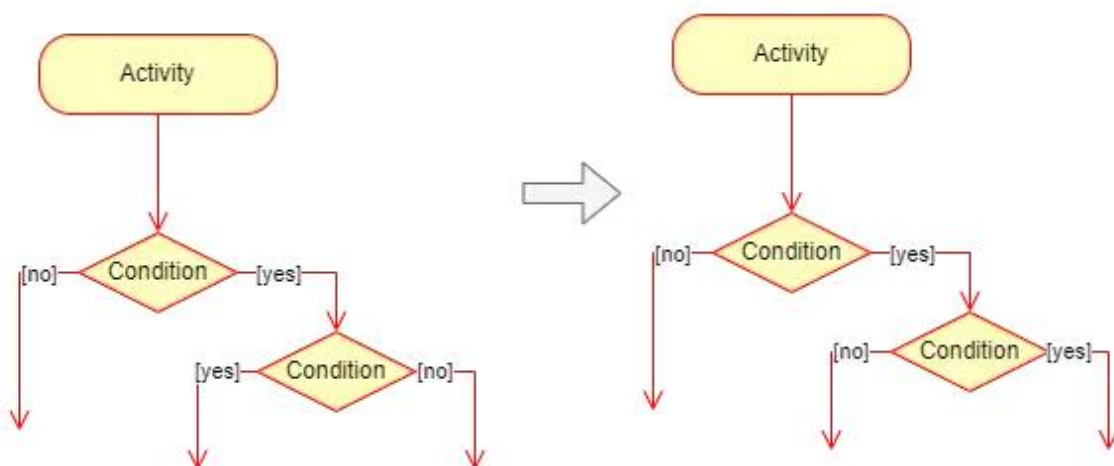
**4. Sometimes,** you have a decision happening straight after a previous decision or a fork in the process. Even though it sounds compelling to combine both actions into one element, it is better not to. This will allow you to show exactly what is happening in the process and make it easier to comprehend and maintain.



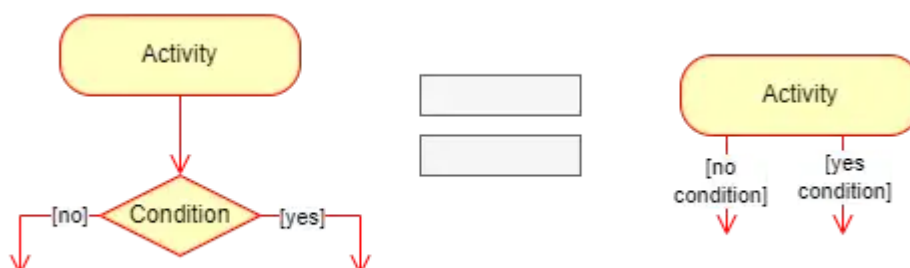
**5. Same applies** to a situation when you have multiple inputs into one activity. Make sure it is understood what is the logic behind entering that activity: either via using joins or via labelling the inputs.



**6. When** having several yes/no decisions, try to keep all “yes” and all “no” flows on the same side.



**7. Although** you may have several outputs for one activity, try to avoid it as well — instead, use a decision node. If having several outputs seems like a better option, always label them to avoid confusion.

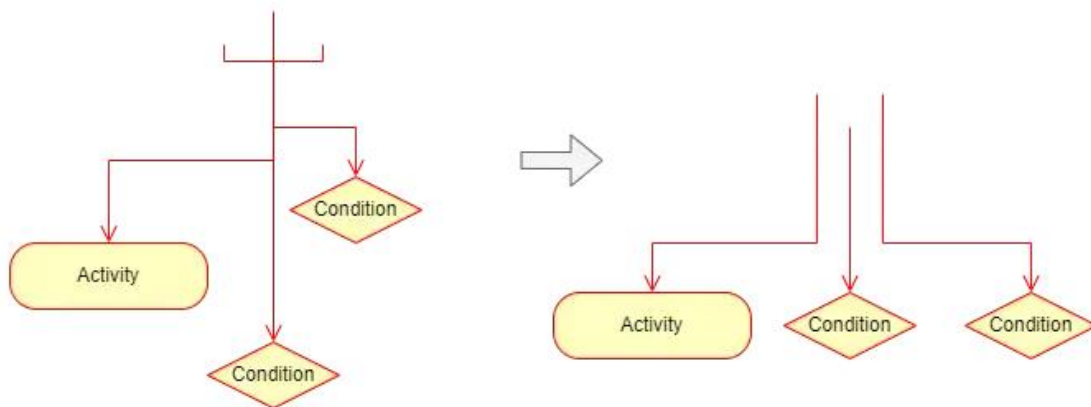


**8. Sometimes** you need to label a condition on a line (arrow), sometimes you need to label an event or data associated with the transition. Make sure it is clear what you model. For

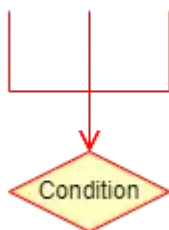
example, indicate [conditions] with square brackets. In this case, events/data won't need brackets.



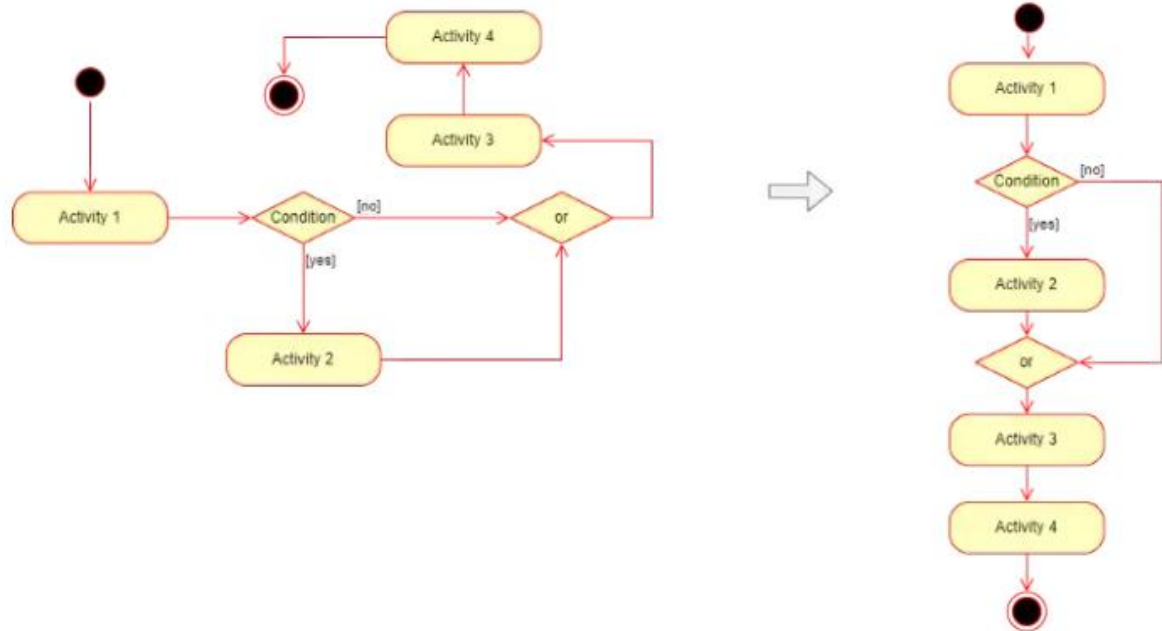
**9. When** you've got multiple control flows or messages floating around on your diagram, as much as possible try to avoid overlapping the lines. When the lines sit one on top of the other, it is impossible to track which one may decide to branch out later.



The only exception is to converge them all together straight before going into the next node. This is acceptable and easily understood.



**10. Whenever possible,** make your diagrams one directional. Top-to-bottom left-to-right is usually the easiest direction in English speaking world. Try to have one direction that will indicate the progression of the process (e.g along the vertical axis) and another to indicate branching (e.g. horizontal expansion of the model). Once selected, try not to mix them.



#### References:

1. A nice intro to basic activity diagrams: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
2. This article was originally published in [my medium blog](#).