**PickBranch (along with Pick)**

My inference mentioned below.

**Study and observation:**

*“At execution time, the triggers for all branches are executed in parallel. When one trigger completes, then its corresponding action is executed, and all other triggers are canceled.”* – source [link](https://docs.microsoft.com/en-us/dotnet/framework/windows-workflow-foundation/pick-activity).

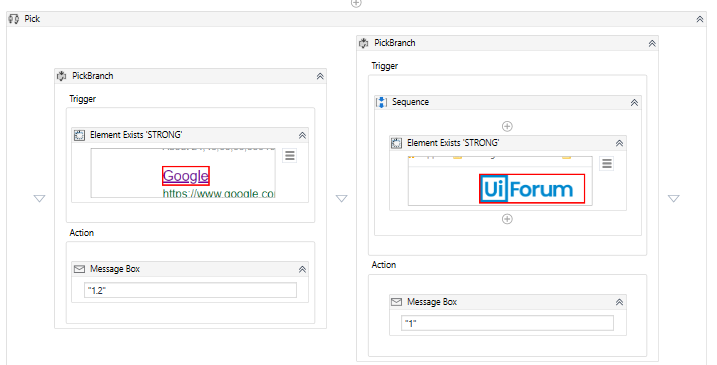
If you have multiple ‘PickBranch’ activities (inside a ‘Pick’ activity), all of them are executed in parallel. Whenever a sequence (or single activity) inside the ‘Trigger’ part of any ‘PickBranch’, the corresponding ‘Action’ is executed and all the other ‘PickBranch’ activities are ignored.

**Usage:**

From what I can imagine is that this might be useful in cases in both attended and unattended modes. When there can be multiple outcomes resulted by any event (e.g. button click etc.), instead of placing activities like ‘Element Exists’/’Image Exists’ etc. in a series, we can use ‘PickBranch’ which can result in a faster execution due to parallelism.

**Trials:**

I tried something like the one mentioned below screenshots. Both the ‘Element Exists’ activities in the above example have the same delays in the properties tab. Whichever element is identified first, the corresponding ‘Message Box’ will be executed. So, if ‘Google’ is identified first by UiPath, then the message ‘1.2’ is displayed first or if ‘UiForum’ is detected first, then ‘1’ will be displayed.



If you try the below screenshot, you can see I have inserted a ‘Delay’ before the 2nd ‘Element Exists’ activity. In this case, since there is a hardcoded delay before the 2nd ‘Element Exists’, the 1st ‘Element Exists’ activity will always execute first (given the element actually exists). Hence, the message ‘1.2’ is displayed always.

