# Process assistant

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

### 0.1 Context

All jobs contain a large amount of repetitive work.
This repetitive work may be described at a high level using what is called a process.
It is quite common for most workers to have an internal procedure for the different things they do, but it is very rare that you will find explicit procedures.
The idea of the procedure assistant is to help workers define their implicit procedures in an attempt to make them explicit and allow them to share these procedures and improve them.

### 0.2 Learned in this study

### 0.3 Things to explore

- Design and improve your processes
- Work through tasks using a process, where you record what you do, so that you may leave the task and come back to it later and have information about its last state
  - What prevents people from methodically recording what they do? Is it mostly because it is time consuming and of little value, or because it's too difficult to do properly?
    - A part of the problem seems to be that interlacing recording with doing the actual steps may create a mental burden on the person, creating the effect of working on two tasks in parallel instead of only on the task at hand
- Are there any graph to programming language converter?
- How do you determine the beginning/end of a workflow (subroutine boundaries)?
- What is the easiest format to share this sort of information with others?
  - It has to be easy to compare such that others with their own processes may compare them against yours
    - This would potentially require the development of a restricted language so that nodes may be matched between graphs
- What are the processes/decisions that I go through the most during the day?
  - How easy are they to automate?
- Processes are non-linear by nature, in that we try to make them into a sequence but there are always numerous elements that will interrupt a process in progress

## 1 Overview

One of the biggest challenges of making procedures explicit is taking the time to do so.
Doing so is a meta exercise that very few people find of interest.
Furthermore, this exercise is complicated by the fact that many activities are guided by some form of preemption (interrupting a task being carried out), which may make them harder to write down.

The most common tool used to make a process explicit is the directed graph.
In it, each task/action is a node and each directed arrow indicates from which task to which task one may proceed.

It is possible for the graph to have cycles, which indicates a set of steps that may require to be executed multiple times before the condition to exit the loop is triggered.

# 2 Purposes/Benefits

- Can be (visually) inspected
- Can be shared and compared with others to gather feedback
- Can be stored and retrieved later (allowing you to forget about it and rapidly get back to it)
  - Can be followed by an individual that knows how to do the steps but not necessarily in what order
- Can be automated
  - Steps can be automatically executed
  - Decisions can be executed without any user intervention (increases the perceived productivity of the user)
  - Notifications to users for manual steps

# 3 Steps

- Create a new process (i.e., do the thing that needs to be done)
- Write down as many steps you do implicitly as possible
- Link steps to one another
- Simulate the process in your head to verify if you forgot any steps
- Follow the process
- Revise as needed
- Extract reusable processes into their own process graph
- Indicate process triggers (what starts a process? an event? a scheduled event? the completion of another process?)
- Define pre-conditions (things necessary for the process to start)

# 4 Tools

- Workflow/Flow chart
  - Track the most popular paths
  - Track the duration of each step
  - Track the frequency of each step
- Checklist
  - Track the most used items

## 4.1 Languages

- BPMN - Business Process Model and Notation
- UML Activity diagram (actors/roles, artifacts, actions)

# 5 Notes

- Processes should be archived as they change. This allows you to observe which processes change often, and which parts change the most.

# 6 See also

# 7 References

## 7.1 Process/Workflow

### 7.1.1 Open source

- http://www.bpmn.org/
- https://bpmn.io/

### 7.1.2 Paid offers

- https://camunda.com/
- https://www.process.st/
- https://www.pipefy.com/
- https://kissflow.com/
- https://www.flokzu.com/
- https://missions.ai/

## 7.2 Issue tracker

- https://www.atlassian.com/software/jira
- https://www.redmine.org/