BPMN® Poster Series #4

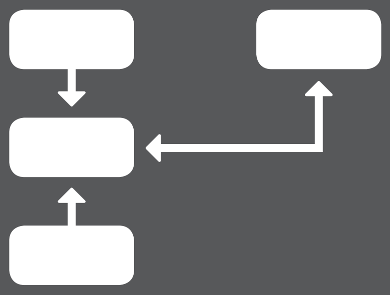
BPMN Patterns: cancellation and force completion patterns

by Gregor Jošt

The aim of Workflow Patterns is to provide a conceptual basis for process technology. They can be used for examining the suitability of a workflow system for a particular project, by assessing strengths and weaknesses of various approaches to process specification. The patterns can range from very simple to very complex and cover the behavior that can be captured within most business process models. Generally, the patterns are classified as Control Flow, Resource, Data, Exception Handling and Presentation patterns. Each of those is further divided into subcategories. This poster introduces the Cancellation and Force Completion Patterns, which belong to the Control Flow. Cancellation and Force Completion Patterns can be found in many exception handling mechanism in processes and they represent how the finishing of one activity may cause the cancellation of another.

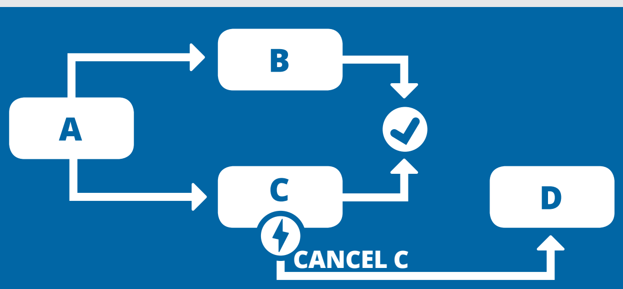
INTRODUCTION

Cancellation and Force Completion Patterns utilize the concept where the completion of one activity may cause the cancellation of an activity or group of activities. Additionally, many forms of exception handling are based on such concepts as well. In this poster we will represent all aspects of cancellation patterns, namely Cancel Task, Cancel Case, Cancel Region, Cancel Multiple Instance Activity and Complete Multiple Instance Activity.



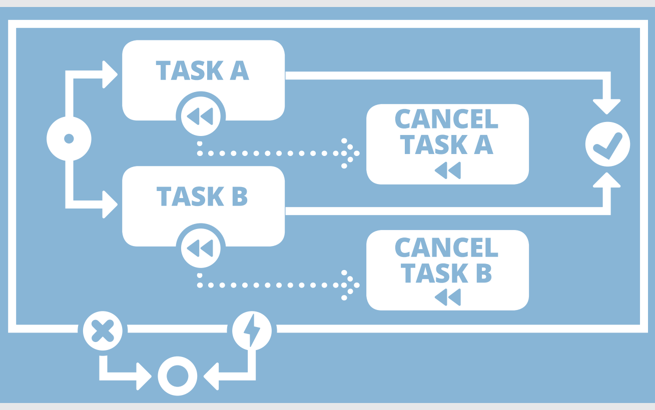
CANCEL TASK *(WCP19)*

The focus of Cancel Task pattern is to provide a mechanism that could withdraw a task, which has already been enabled or executing. The pattern must ensure that such an activity does not complete and if possible, the running instance is paused and removed. BPMN supports Cancel Task pattern by using Boundary Intermediate Events, attached to the Activity that is supposed to be cancelled.



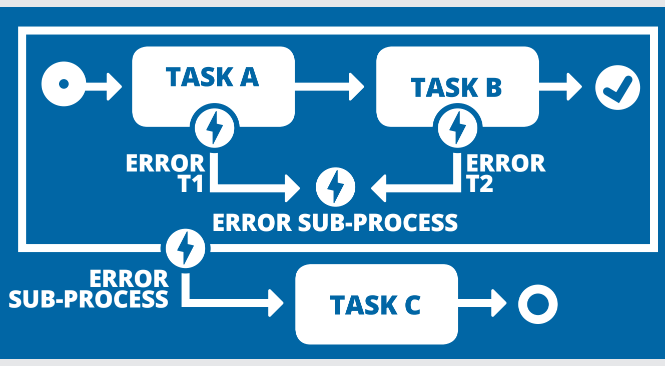
CANCEL CASE *(WCP20)*

Cancel Case pattern focuses mainly on halting a specific process instance and withdrawing all tasks that are associated with it. Process instance is removed, which includes currently executing Tasks, Tasks that might execute in the future and any Sub-Processes. Process instance in this case is recorded as having been unsuccessfully completed. In BPMN this is easily achieved by incorporating transactions. If a Cancel End Event associated with the transaction will be triggered, all activities associated with a process instance are terminated.



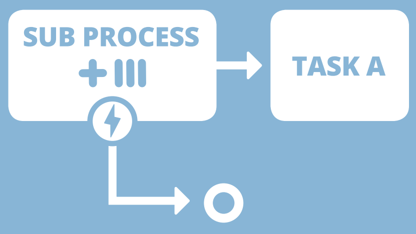
CANCEL REGION *(WCP25)*

If we want to disable just a subset of Tasks (unlike a complete process, as in Cancel Case pattern), we can implement the Cancel Region. The pattern has the ability to disable a subset of Tasks in a process instance. Such an approach enables us to handle unexpected errors and exceptions. The Cancel Region pattern is supported in BPMN by either using Sub-Processes with a corresponding Error Event to enable the cancellation.



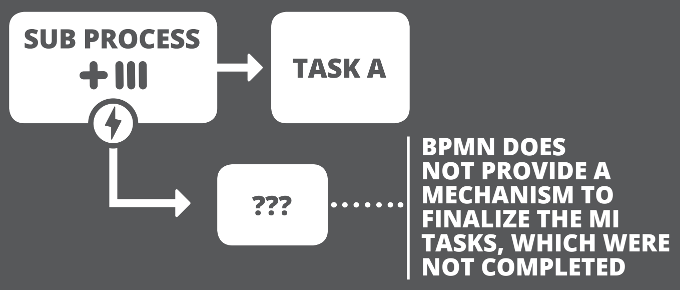
CANCEL MULTIPLE INSTANCE TASK *(WCP26)*

According to BPMN 2.0.2 specification, the multi-instance (MI) Task is a type of Tasks that acts as a wrapper for a Task which has multiple instances spawned in parallel or sequentially. In the Cancel Multiple Instance Tasks, the required number of instances is known at design time and are ran concurrently. If at any time the MI Task is cancelled, any instances which are not completed, are withdrawn, while the already executed instances are unaffected by the cancellation. This is easily achieved by MI Task which has an Error Intermediate Event attached at the boundary. When the MI Task is cancelled, the Boundary Event is triggered to terminate any remaining MI instances.



COMPLETE MULTIPLE INSTANCE TASK *(WCP27)*

Complete Multiple Instance Task pattern makes the required number of instances is known at design time and they are ran concurrently. The pattern provides a mechanism to finalize MI Tasks, which were not completed, at any time during their execution. However, this pattern is NOT supported by BPMN, since there is no mechanism available, which would cancel the remaining MI Tasks instances.


SUB PROCESS
TASK A
???
BPMN DOES NOT PROVIDE A MECHANISM TO FINALIZE THE MI TASKS, WHICH WERE NOT COMPLETED

@goodelearning /goodelearning /company/good-e-learning

Free Resource Library
www.goodelearning.com