



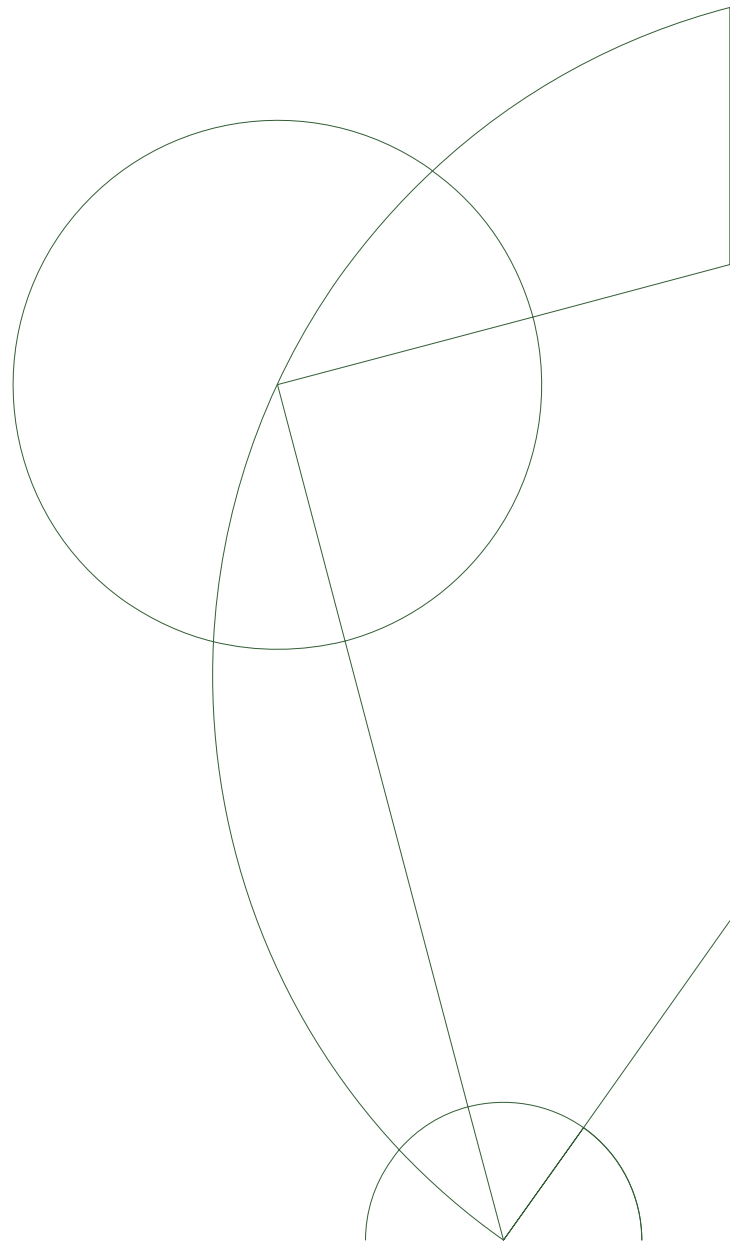
Reactive & Event-Based Systems

Assignment 1: Process Models & Event-Based Systems

Frederik Ingwersen

Rasmus Winther

November 30, 2019



Model descriptions

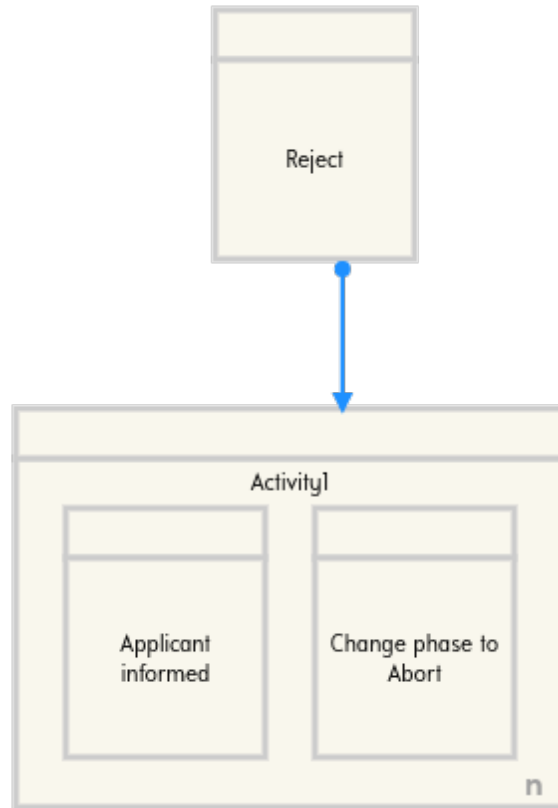
Pattern 1

With this model, we had to find all the activities in the Dreyers log, which we simply did by making a small Python script. We added an additional activity called "Case" to avoid having a condition arrow pointing to each and every one of the activities.



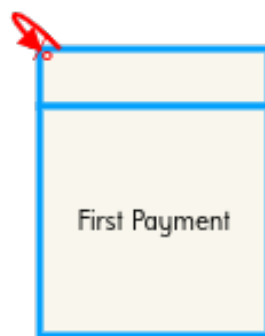
Pattern 2

Again, we nested the two within an outer activity to only draw one pending arrow. In the assignment text, however, it is stated that both Applicant informed and Change phase to Abort should happen after Reject, eventually. We were a bit in doubt whether this should mean that Applicant informed should happen after Reject and then Change phase to Abort, or if the order is irrelevant. We chose the latter.



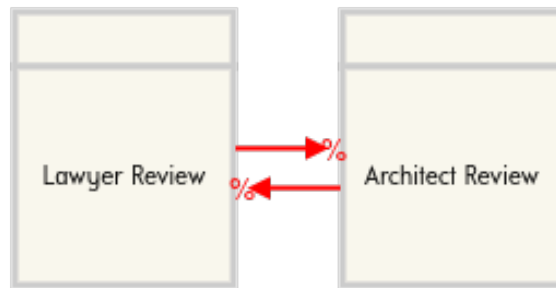
Pattern 3

This simply excludes itself after it has been executed.



Pattern 4

These simply exclude each other after execution, making sure only one of them is ever executed.



Engine

Our own Engine

We decided to write our own DCR engine with Go.

Engine overview

XML Format

Conformance Checker

To check how a graph satisfies a trace we compile a set of all legal events with the current marking of the graph. If the event is included in the set of legal events the event is executed and the marking is updated. This process is repeated every event in the trace. If any event is not in legal events is the trace not satisfied. If every event is executed the set of pending events is checked and if no events are pending the trace the graph is concluded to satisfy the trace.

Tests