REBS – Assignment 1

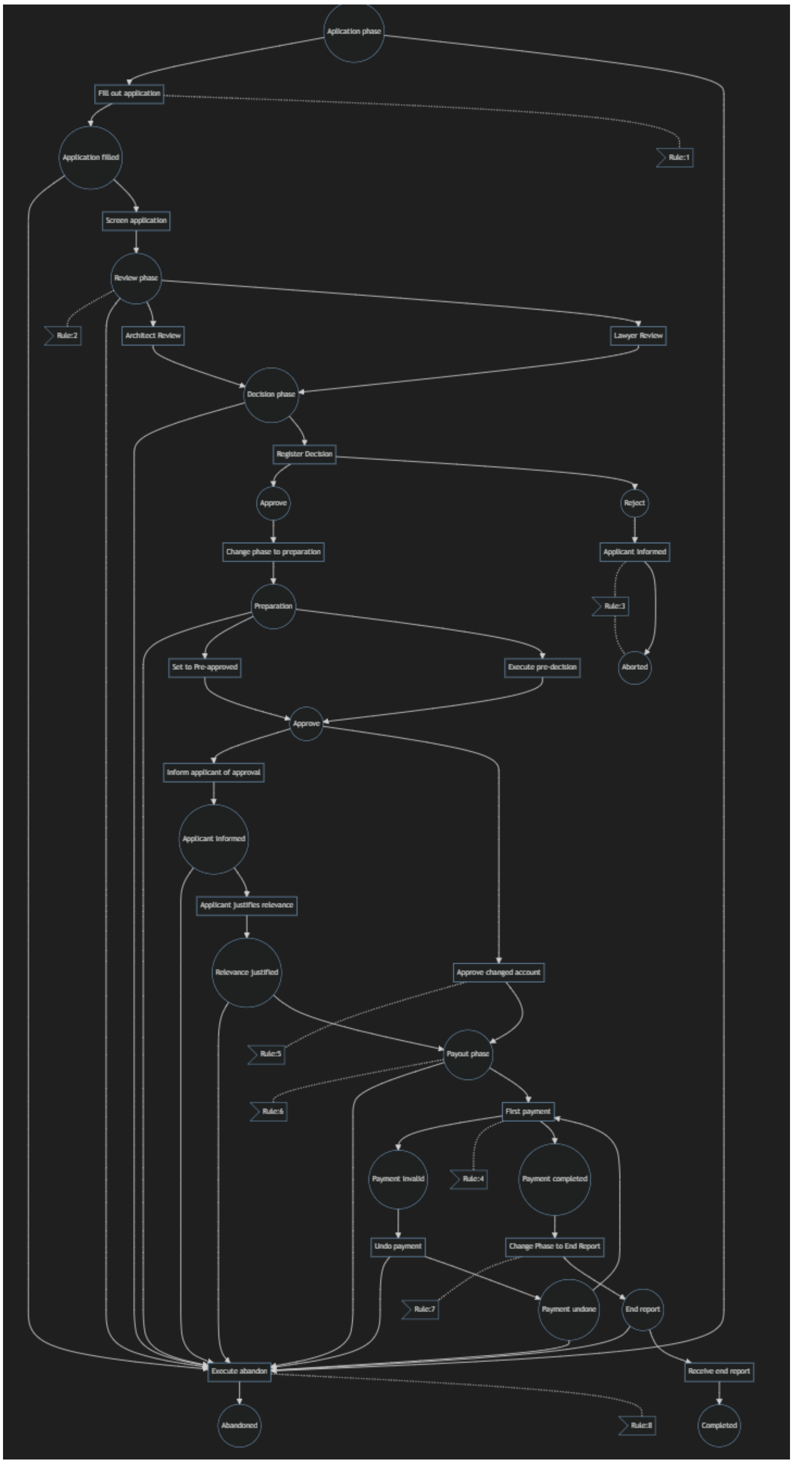
Albert Friis-Hansen (ghv657), Felix Fleckenstein (vcb456)
Group 3

22. januar 2024

1 Part 1

1.1 Task 1 and 2

Note that in our model places are circles, transitions are rectangular and directed arcs are arrows. Notes to show the rules, are indeted in the leftside and showed with striped lines.



Figur 1: Petrinet

1.2 Task 3

1.2.1 Is your Petri net live and/or quasi-live?

Live means for the possibility for every transition can be reachable from any state, which is not possible in our petrinet. TO given an example, as soon as you move through the application phase, you cannot come back.

Our petrinet is quasi live, because all transitions are possible to fire through, at some point in the model, but they may not be reachable from any marking.

1.2.2 Is your Petri net bounded and/or safe?

Our petri net is neither bounded nor safe. The issue is that ’First payment’ has an AND transition to both ’payment completed’ and ’payment invalid,’ leading to the possibility of placing unlimited tokens in ’payment complete’ following the second route through ’payment undone.’

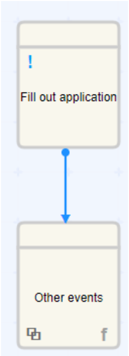
1.2.3 Is your Petri net a WorkFlow net?

Our answer:
Our model is a workflow net, where it represents a model showing the proccess that happens when a application is submitted. Also we have a endpoint at either "Completed"or "Abandoned".

Our model doesn’t strictly adhere to the definition of a workflow net, as it has multiple endpoints. A workflow net typically has a single endpoint.

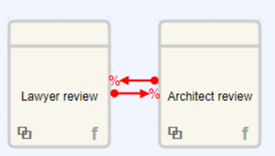
2 Part 2

2.1 Task 1



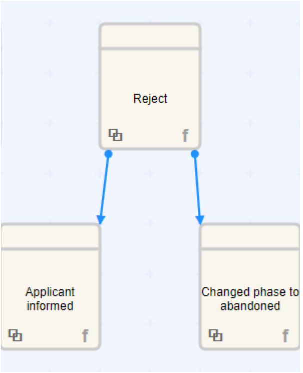
Figur 2: Graph 1

This should have modeled that "Fill out application"should always happen first. It does however, not. We have since assignment 1 changed it to do that, as can be seen in our Assignment 2.



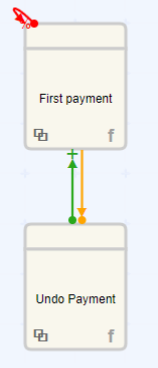
Figur 3: Graph 2

In this graph, we've modelled the relationship between the lawyer review and the architect review. These arrows means, that either excludes the other from occuring.



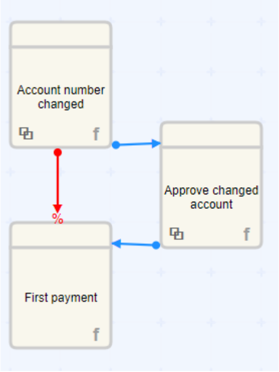
Figur 4: Graph 3

This figure describes that once we've rejected an applicant, we must eventualy both inform the applicant, and change the phase to abandoned. It doesn't matter which order it occurs in, as long as it happens after reject.



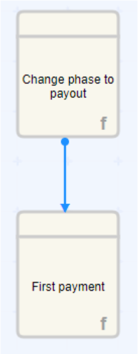
Figur 5: Graph 4

This model depicts how the payments work. We can see here that first payment exludes itself from happening once it has happened, to ensure we dont receive payment twice for the same application. The yellow (condition) arrow tells us that "Undo Payment"cannot happen before first payment has happened at least once. The green (include) arrow means that if "Undo Payment"happens, then first payment must be the next step.



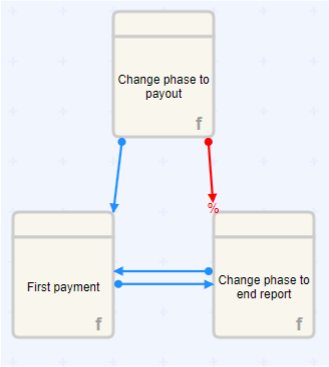
Figur 6: Graph 5

This model tells us that the first payment cannot happen before the account number is changed and approved. We should have included "first payment"again eventually so it can be run again.



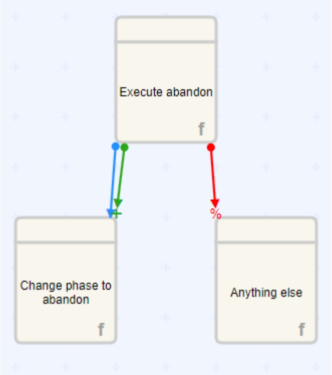
Figur 7: Graph 6

This figure tells us that if we change the phase to payout, then we must eventually receive first payment. We shouldn't have any pendings.



Figur 8: Graph 7

This graph describes that we cannot change the phase to end report before we've received the first payment.

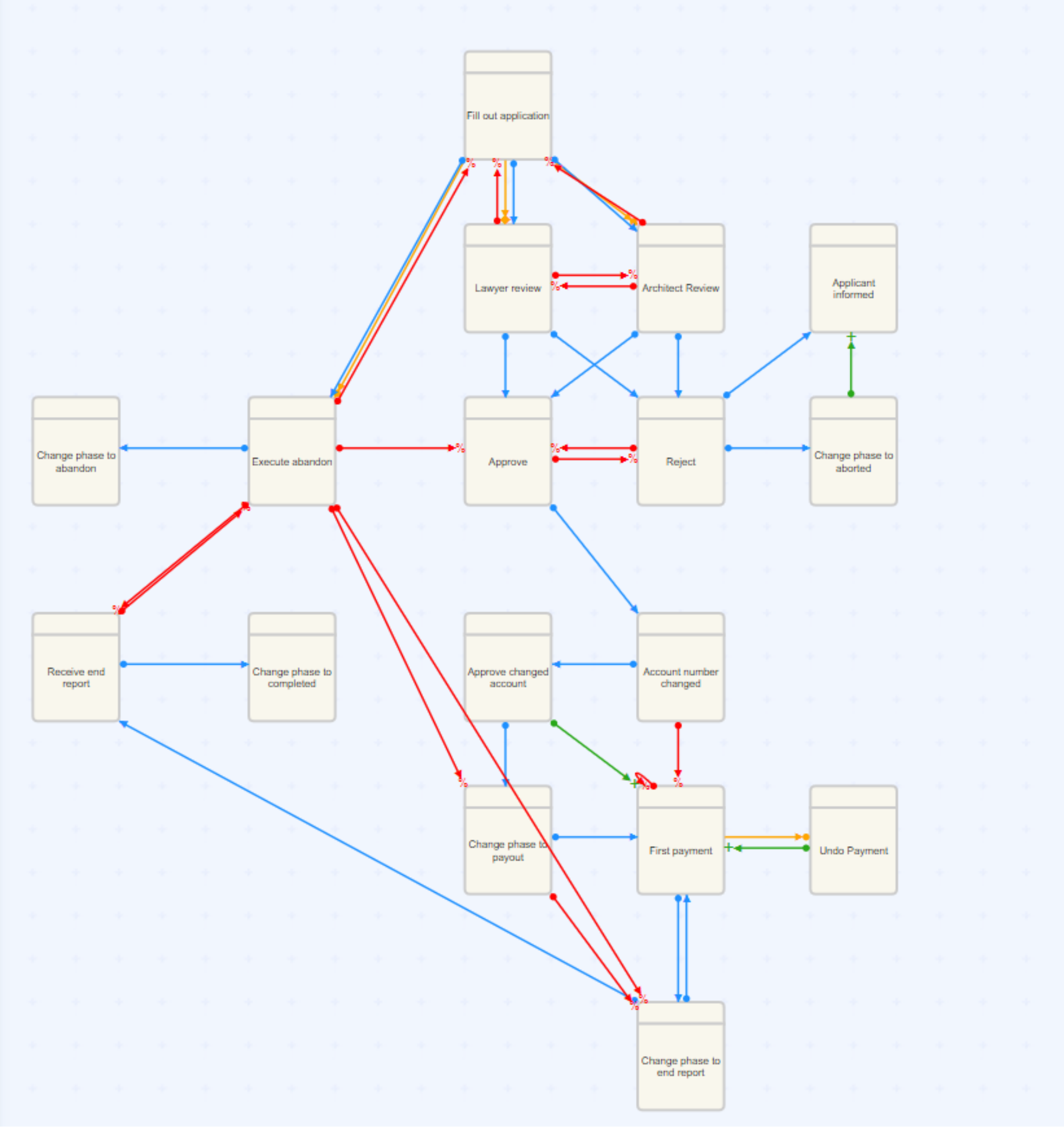


Figur 9: Graph 8

This graph tells us that if we execute abandon, the only thing we can do is to change the phase to abandon. The include isn't relevant.

2.2 Task 2

We've updated the model in assignment 2, and it can be seen underneath here.



Figur 10: Fully updated DCR

FEEDBACK ON "FIRST PAYMENT"TO "CHANGE PHASE TO END REPORT"

2.3 Task 3

2.3.1 Which relations did you not use in your models?

- No response
- Value
- Spawn
- Milestone

We didn’t use milestone, as we weren’t clear on what it meant. We should have considered using it for better representation.
What milestone means:
The milestone relation in the context of DCR graphs is a way to model a point in the process that needs to be reached before certain actions or events can take place. It is used to represent a significant achievement or state in the workflow that must be accomplished before proceeding to subsequent activities.

2.3.2 Could some of the rules have been modelled in more than one way? If so, give one or two examples.

One example of alternative modeling could have been using the Milestone relation at the beginning of the DCR graph. This could ensure that ’Fill out application’ is the only way to start the process.

2.3.3 How does your model differ from the Petri net? Do they exhibit the same language?

DCR graphs and Petri nets differ in language and notation. DCR graphs use a graphical notation with nodes representing events, conditions, responses, and milestones, while Petri nets use a more mathematical notation with places, transitions, arcs, and tokens. They are both used to model dynamic systems but employ different languages and notations. In a petri net the order will be the same every time, where in the DCR graph the order can vary.