Assignment 1: Process Models and Event-Based Systems

Reactive and Event Based Systems

Part 1: Modelling Workflows as Petri nets

Consider the following activities extracted from the Dreyers log introduced and examined in the paper *The Analysis of a Real Life Declarative Process*:

Fill out application
Approved - to board
Change phase to Review
Architect Review
Review
Lawyer Review
Register Decision
Change phase to Board meeting
Round ends
Round approved
Inform application of board review
Reject
Applicant informed
Change phase to Abort
Screening reject
Screen application
Execute pre-decision
Approve
Change phase to Preparation
Set to Pre-approved
Inform applicant of approval
Applicant justifies relevance
Change Phase to Payout
First payment
Payment completed
Change Phase to End Report
Account number changed
Receive end report
Change phase to Complete
Execute abandon
Change phase to Abandon

Task 1: Using these activities, model a workflow of an application process as a Petri net. You may imagine the details yourself, but you should ensure that your process captures at least the following rules:

1. Fill out application should always be the first event of the case.
2. Lawyer Review and Architect Review should never occur together.
3. Reject should always eventually be followed by Applicant informed and Change phase to Abort.
4. First payment should only occur once, unless Undo payment is executed afterwards, in which case it may be repeated once.

5. If Account number changed happens, then afterwards Approve changed account needs to be executed before one can execute First payment.
6. Change Phase to Payout should always eventually be followed by First payment.
7. After Change Phase to Payout has happened, Change Phase to End Report should not happen before we do First Payment. If Change Phase to Payout did not happen then Change Phase to End Report is not restricted by First payment.
8. Execute Abandon may happen at any time, after it is executed only Change phase to Abandon may happen.

You should at minimum use the activities mentioned in these rules, and may also use others if you think it makes sense.

Task 2: For each rule, show where it is captured in the model, either by annotating the model, or describing how the rule is implemented in text (as a short paragraph for each rule) .

Task 3: Answer the following questions and add a short (1 paragraph) argument for your answer:

1. Is your Petri net live and/or quasi-live?
2. Is your Petri net bounded and/or safe?
3. Is your Petri net a WorkFlow net?

Part 2: Modelling Event Patterns as DCR Graphs

Task 1: Model the rules from part 1 as individual DCR Graphs. (Yielding 8 graphs in total.) Write a short description for each explaining the semantics. For convenience you may want to use an *“other”* event in some of your models that represents all events not explicitly mentioned.

Task 2: Combine the separate patterns into a single model.

Task 3: Answer the following questions:

1. Which relations did you not use in your models?
2. Could some of the rules have been modelled in more than one way? If so, give one or two examples.
3. How does your model differ from the Petri net? Do they exhibit the same language?

Hand-in

For this assignment you should deliver a PDF report with sections for each task. You should include your models as figures in the report. (I.e. your report will at minimum contain a Petri net, 8 small DCR Graphs, one complete DCR model and 1 more final DCR model for task 2.4.)
The report should contain a front page stating your group name and the names of the group members that contributed to the submission.