

Assignment 1: Process Models and Event-Based Systems

Reactive and Event Based Systems

Setup

1) Install **DCR4Py**: https://github.com/paul-cvp/pm4py-dcr/tree/feature/dcr_in_pm4py_revised

Note that if you use github codespaces you may need to run these two additional steps:

- pip install click
- conda install python-graphviz

2) Download the following Jupyter Notebook and use it to solve the assignments below:

<https://drive.google.com/file/d/1VUqPcSBVZb3Hld0W6r7zsyDfVWwXf2lj/view?usp=sharing>

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
Architect Review
Lawyer Review
Reject
Applicant informed
Change phase to Abort
Change Phase to Payout
First payment
Undo payment
Change Phase to End Report
Account number changed
Approve changed account
Execute abandon
Change phase to Abandon

Task 1: Using these activities, model a simple workflow of an application process as a Petri net. It does not need to follow exactly the process as described in the paper and you may add additional 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** can not occur unless **Fill out application** happened before.
3. **Lawyer Review** and **Architect Review** should never occur together.
4. **Reject** should always eventually be followed by **Applicant informed** and **Change phase to Abort**.
5. **First payment** should only occur once, unless **Undo payment** is executed afterwards, in which case it may be repeated once more.
6. If **Account number changed** happens, then afterwards **Approve changed account** needs to be executed and we can no longer execute **First payment** until the approval has happened.
7. Change **Phase to Payout** should always eventually be followed by **First payment**.

8. 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**.

Create a copy of your Petrinet with one additional rule:

9. **Execute Abandon** may happen at any time, after it is executed only **Change phase to Abandon** may happen.

Task 2: For each rule describe how the rule is implemented (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 9 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:

- 1) Your Jupyter Notebook
- 2) An export of your Jupyter Notebook where all cells have been successfully executed as a PDF file.