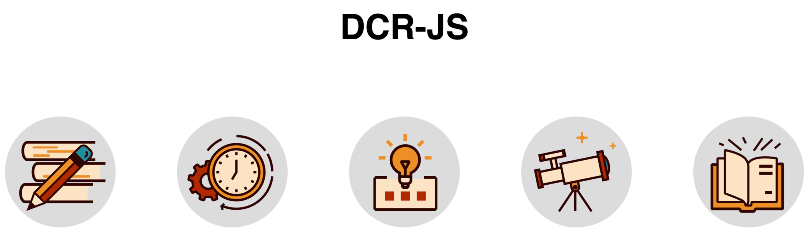
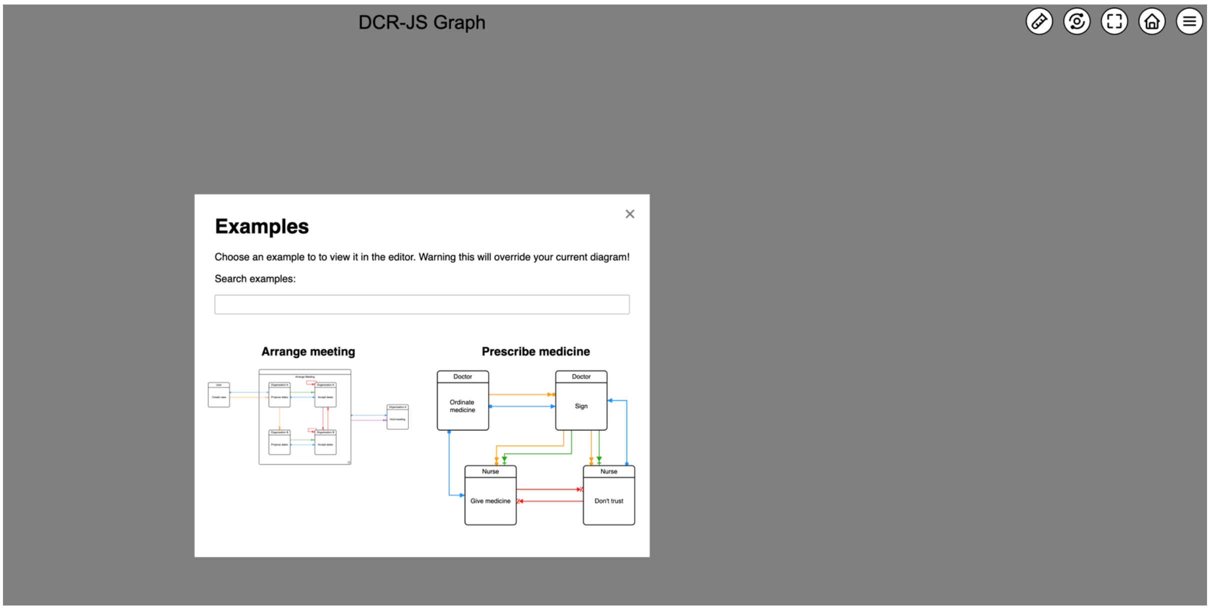


Figure 4: DCR-JS landing page



Choose an example to to view it in the editor. Warning this will override your current diagram!

Figure 5: Selection of Examples

A. Getting started

The tool can be found at https://hugoalopez-dtu.github.io/dcr-js/. From the home page (c.f. Figure 4), the different modes of the application can be accessed. From right to left: Modeling, Simulation, Conformance, Discovery, and Log Generation. For all features, a menu can be opened by clicking the three bars in the top right corner, allowing for import and export, etc. and you can return to the homepage by clicking on the house icon next to the menu.

Starting with the modeling page, one can start freely creating DCR graphs using the control palette on the left hand side. Alternatively, the user can start with a predefined set of templates. Simply click on the menu and choose the “Examples” option (c.f. Figure 5).

At any point in the interaction with a mode, a user can click the home button (c.f. Figure 6) to access the remaining features. If models (or logs) will be used in other modes, it is important to store them in memory via the corresponding “Save” functionality (c.f. Menu in Figure 6).

To verify the functionality of the model, the user can move to the simulation mode (c.f. Figure 7). This mode opens instantiating the initial marking of the DCR graph, where enabled events will be colored in green, and disabled events will have dashed borders. The user simply needs to click on the events to execute them, generating a trace (c.f. Trace Info in Figure 7). At any point, the trace can be finalized, and multiple traces can be generated to create an event log. This log can be download in XES file for further use. If one wishes to generate non-conforming traces, it is possible to toggle the “comet icon” in the top right, which disables the enabledness and acceptance checks. Each trace name can be edited while generating the trace, and the event log name can be edited at any time.

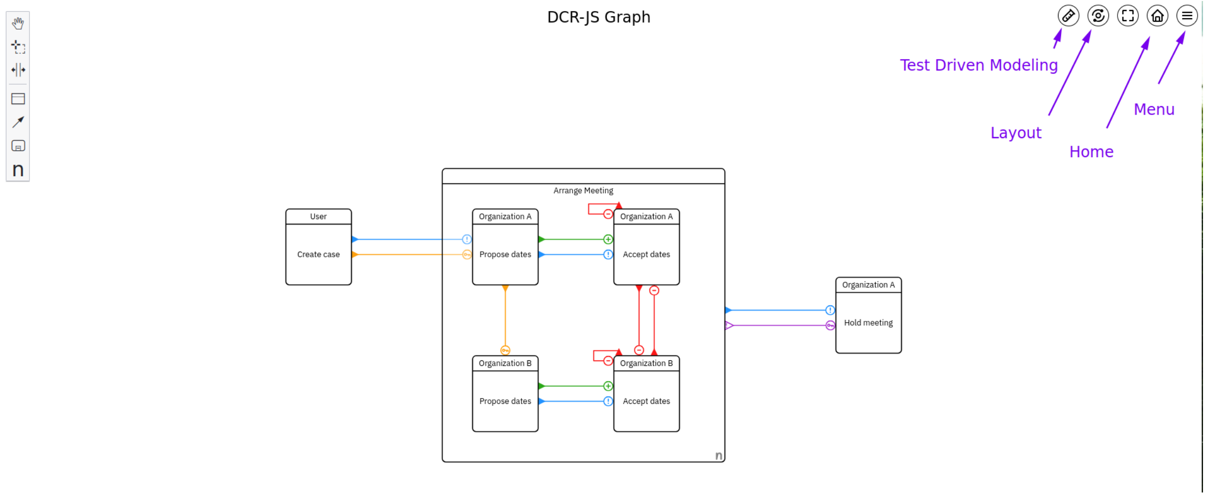


Figure 6: Modeler page with annotations

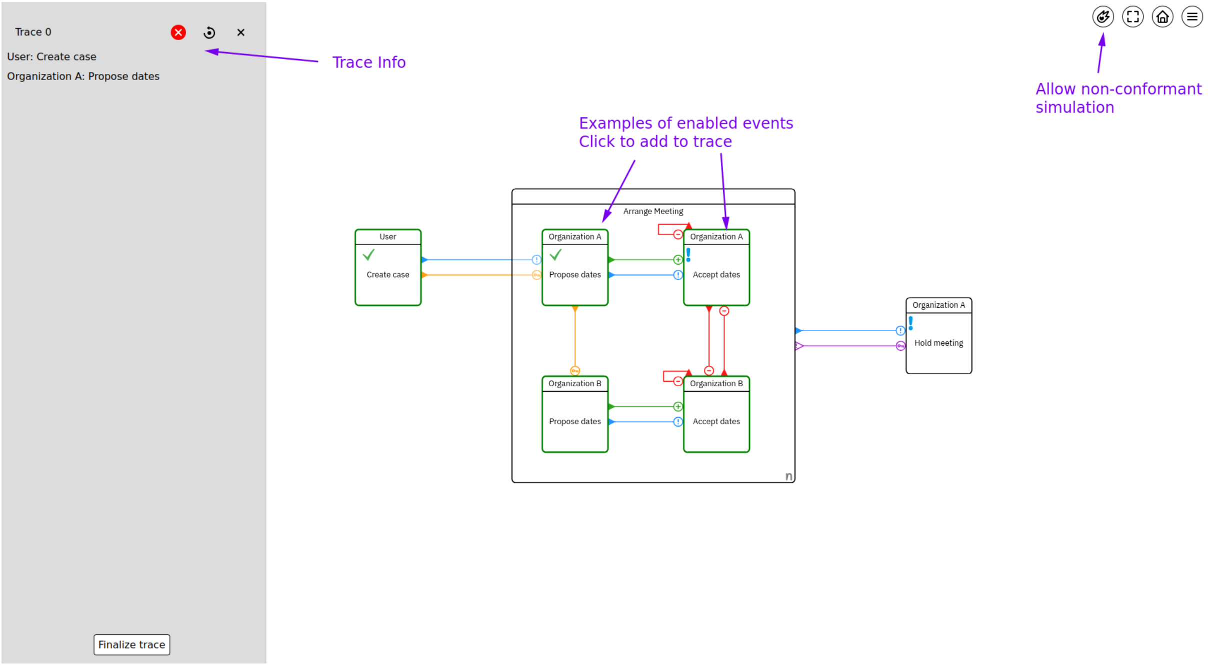
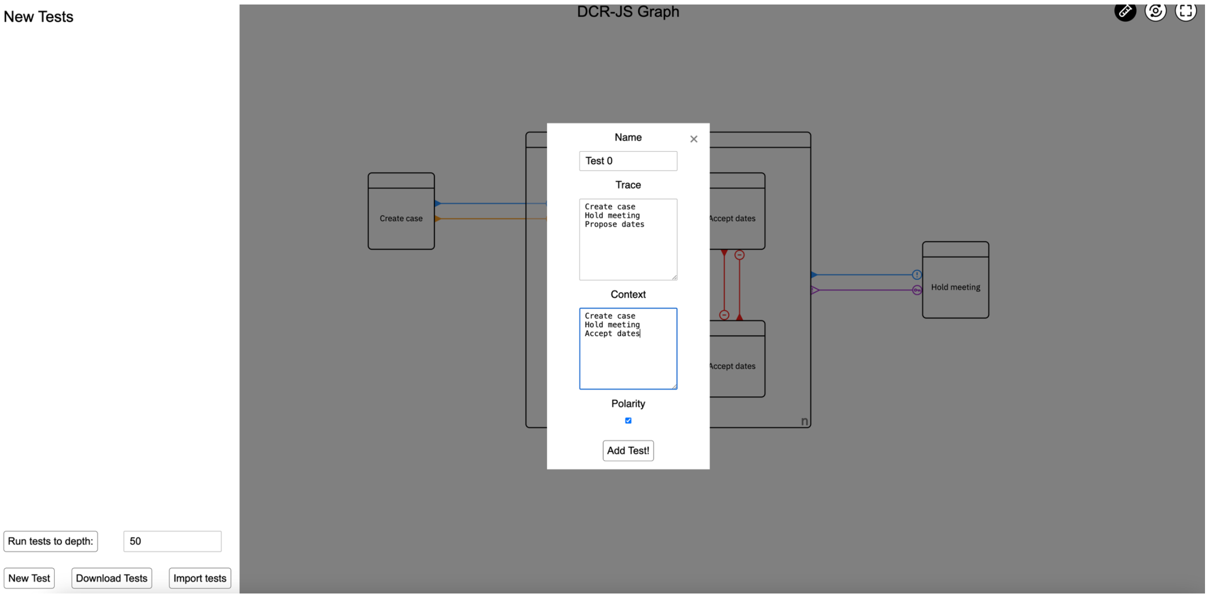


Figure 7: Simulation page with annotations

An alternative way of testing the model is to access the test-driven modelling page (c.f. Figure 6), opening the panel in Figure 8. This mode (still in development) shows alternative ways to define open tests. The user can then define the events that need to be present in the trace, as well as the context of events included and its polarity (clicked for positive tests, and disabled for negative tests).

Clicking on the process discovery mode will open the page in Figure 9. Here, different process discovery algorithms could be selected. Moreover, the user can select to apply noise filtering and encapsulation via nestings. Finally, the user can specify whether to save the log and the model for downstream tasks.

Accessing the conformance checking mode will open the page in Figure 10, which automatically will perform checking against saved logs or models. If one of them needs to be imported, this can be done via the corresponding menu. By default, the results for conformance apply a heatmap visualization, showing compliant rules in green, violated rules in red, and inactive rules in grey. Selecting one particular trace will change the visualization of constraints for that particular instance. To remove the



Name

Test 0

Trace

```
Create case
Hold meeting
Propose dates
```

Context

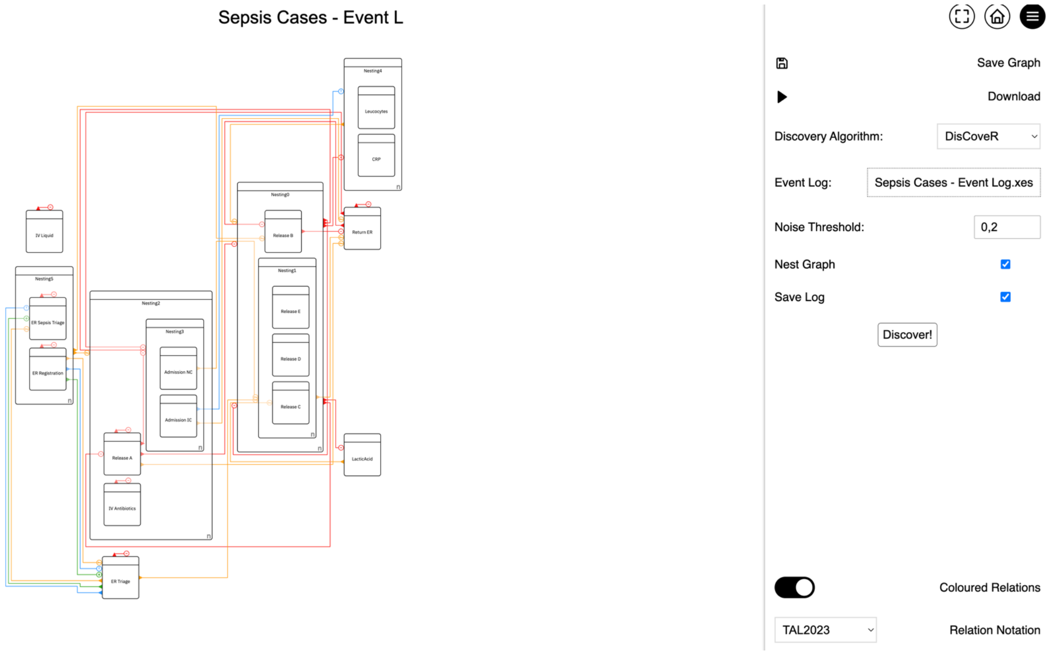
```
Create case
Hold meeting
Accept dates
```

Polarity ☑

Add Test!

Run tests to depth: 50

Figure 8: Test-Driven Modelling



Discovery Algorithm: DisCoveR

Event Log: Sepsis Cases - Event Log.xes

Noise Threshold: 0,2

Nest Graph ☑

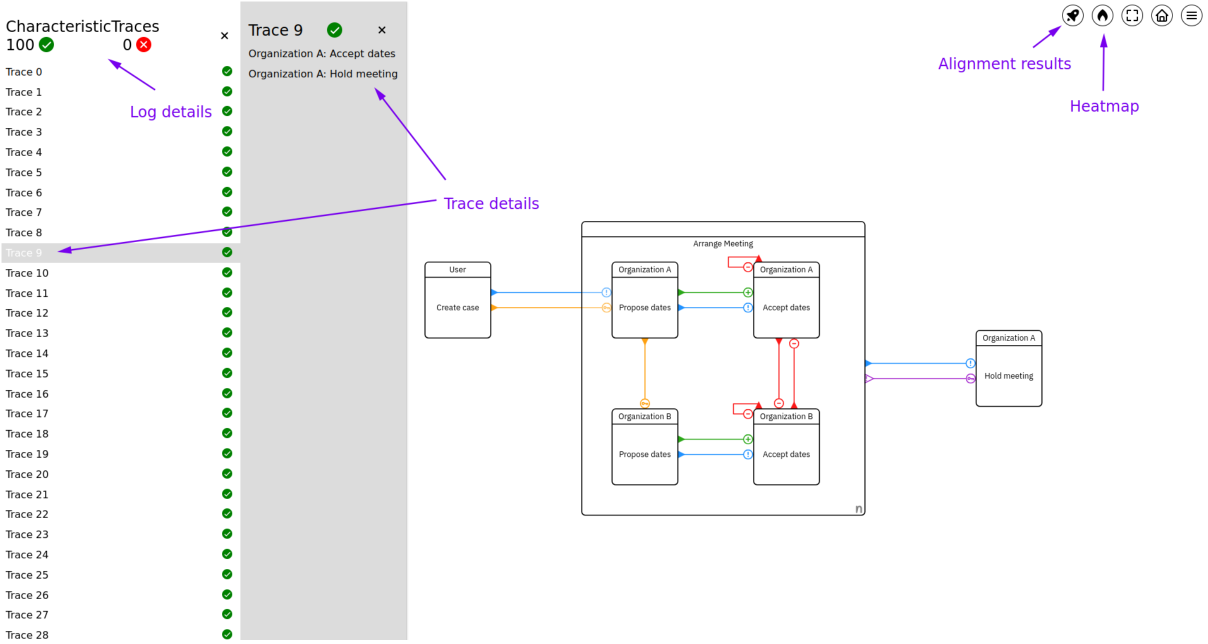
Save Log ☑

Relation Notation: TAL2023

Figure 9: Process Discovery Pane

heatmap visualization, the user can click on the flame icon. Selecting the rocket icon will present the user with alignment results. These two features are mutually exclusive. If none are selected, the results are simply shown as the rule-based violation count in the trace menu on the left.

Finally, the log generation mode will open the page in Figure 11. Here, the user will need to specify the number and length of the traces, as well as the noise percentage (creating potentially non-conforming traces). The results will be stored in an XES file and will be made available to the user for download.



CharacteristicTraces ×

100 ✓ 0 ✗

Trace 0 ✓
Trace 1 ✓
Trace 2 ✓
Trace 3 ✓
Trace 4 ✓
Trace 5 ✓
Trace 6 ✓
Trace 7 ✓
Trace 8 ✓
Trace 9 ✓
Trace 10 ✓
Trace 11 ✓
Trace 12 ✓
Trace 13 ✓
Trace 14 ✓
Trace 15 ✓
Trace 16 ✓
Trace 17 ✓
Trace 18 ✓
Trace 19 ✓
Trace 20 ✓
Trace 21 ✓
Trace 22 ✓
Trace 23 ✓
Trace 24 ✓
Trace 25 ✓
Trace 26 ✓
Trace 27 ✓
Trace 28 ✓

Trace 9 ✓ ×

Organization A: Accept dates

Organization A: Hold meeting

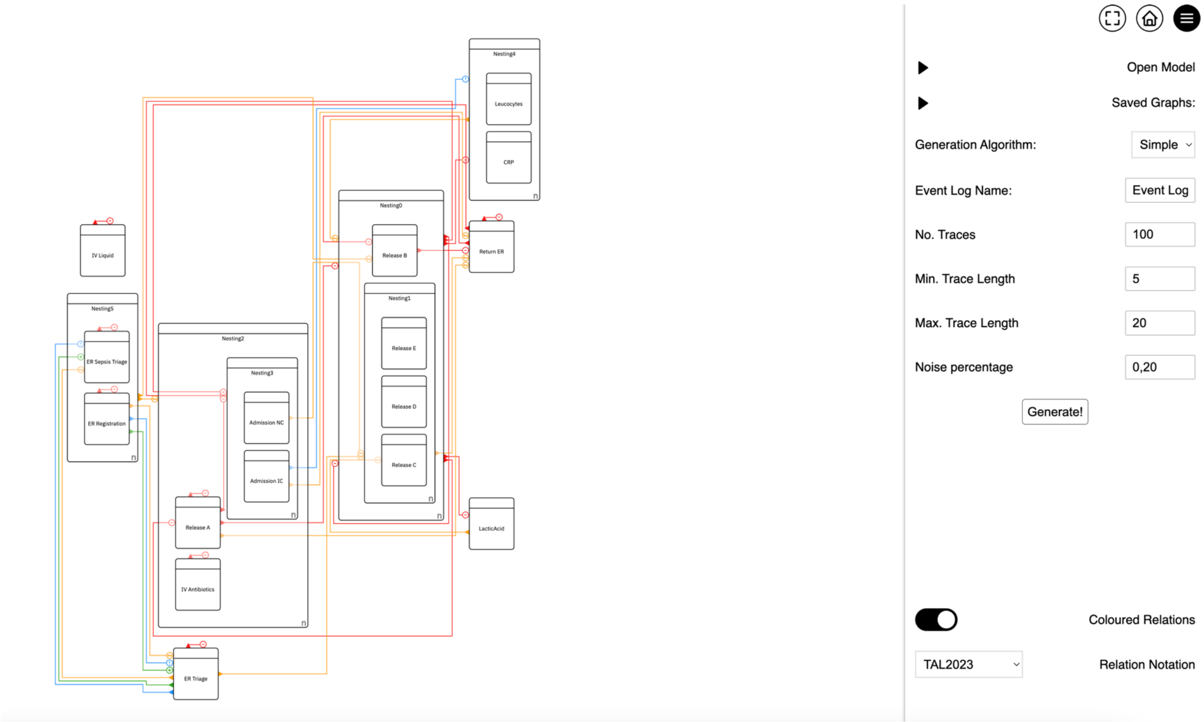
Log details

Trace details

Alignment results

Heatmap

Figure 10: Conformance page with annotations



▶ Open Model

▶ Saved Graphs:

Generation Algorithm: Simple

Event Log Name: Event Log

No. Traces 100

Min. Trace Length 5

Max. Trace Length 20

Noise percentage 0,20

Generate!

Coloured Relations

TAL2023 Relation Notation

Figure 11: Log Generation