1. Maude specification

We are using the predefined Maude module for object-based programming to encode the snapshot metamodels in Maude. Thus, we can represent state machine snapshots and Business Process Modeling Notation (BPMN) process snapshots as objects in Maude. In addition, we add behavioral relationships from the system relationship model (SRM) such that we can instantiate them to create the system configuration described in the use case.

Our Maude rules always rewrite multisets of objects, where individual objects are denoted in angle brackets ($< \cdots >$). Interactions lead to rule merges in Maude, where the left and right sides of each rule are added to a global rule. The global rule is a valid Maude rule since the union of two multisets always results in a multiset. Furthermore, behavioral relationships, as defined in the interaction, are added to the rule as additional objects on both sides. Similarly to the process for graph transformation (GT) rules, individual rules are deleted.

We will now explain how rewriting rules are generated for the use case. To apply our approach to the use case, we need to define local higher-order model transformation (HOT)s from state machines and BPMN processes to rewriting rules.

1.1. State machine semantics

The generation of rewriting rules is similar to the generation of GT rules. Each state machine transition leads to one rewriting rule, which rewrites state machines conforming to the state machine snapshot metamodel. Listing 1 shows the rewriting rule for the transition turn red-amber of the traffic light model.

Using Maude's search command and search graph visualization, we can verify that the generated state space conforms to the traffic light state machine. The generated Maude specifications and example executions can be found in (Kräuter 2023).

1.2. BPMN semantics

The rewriting rule generation is similar to the GT rule generation and supports the same subset of BPMN semantics. Similarly, one or more rules are constructed for each flow node in a BPMN process, where objects conform to the BPMN snapshot metamodel. Listing 2 shows the Maude rewriting rules for the task Switch to P1 of the TJunction controller.

Like the GT rules, the rewriting rules move tokens by changing their position according to the given BPMN model. Thus, we can generate Maude rewriting rules that implement the behavioral semantics of the BPMN models in the use case (see (Kräuter 2023)).

1.3. Check behavioral consistency

The defined interactions change the rules for switching to phases 1 and 2. Listing 3 shows the resulting rule for switching to phase 1, where synchronization with the end of the task was chosen. The rule changes all traffic lights simultaneously and finishes the task. Similarly, a global rule for switching to phase 2 is created and all individual rules are deleted.

Finally, we can use the Maude model checker module to check the requirements formalized by the Linear Temporal Logic (LTL) properties 1-4. Atomic propositions can be represented in Maude using the encoded SRM and the snapshot metamodels.

References

Kräuter, T. (2023, February). *Artifacts - Towards behavioral consistency in multimodeling*. https://github.com/timKraeuter/Towards-behavioral-consistency-in-multi-modeling.

```
var X : String . --- Object id
rl [turn_red_amber] : < X : FSM | name : "trafficLight", state : "red" >
                   => < X : FSM | name : "trafficLight", state : "red-amber" > .
```

Listing 1 Maude rule for *turn red-amber*

```
vars o0 : Oid . --- Object ids
vars SIG M T : MSet . --- Signals, messages and tokens
vars P S : Configuration . --- Processes and subprocesses
--- Switch to P1 start rule
rl [Switch_to_P1_start] :
< "use-case-execution" : BPMNSystem | messages : (M), processes : (
    < o0 : ProcessSnapshot | name : "T-Junction controller", tokens : ("Phase_2_Switch_to_P1" T),
    signals : (SIG), subprocesses : (S), state : Running > P) >
    =>
< "use-case-execution" : BPMNSystem | messages : (none), processes : (
    < o0 : ProcessSnapshot | name : "T-Junction controller", tokens : ("Switch_to_P1" T), signals : (
    none), subprocesses : (S), state : Running > P) > .
--- Switch to P1 end rule
rl [Switch_to_P1_end] :
< "use-case-execution" : BPMNSystem | messages : (M), processes : (
    < o0 : ProcessSnapshot | name : "T-Junction controller", tokens : ("Switch_to_P1" T), signals : (
    SIG), subprocesses : (S), state : Running > P) >
    =>
< "use-case-execution" : BPMNSystem | messages : (none), processes : (
    < o0 : ProcessSnapshot | name : "T-Junction controller", tokens : ("Switch_to_P1_A_&
    _C_are_green_Phase_1" T), signals : (none), subprocesses : (S), state : Running > P) > .
```

Listing 2 Example rewriting rules for the TJunction controller

```
vars X Tl1 Tl2 Tl3 : String . --- Object ids
vars P S : Configuration . --- Processes and subprocesses
vars SIG M T : MSet . --- Signals, messages and tokens
rl [Switch_to_P1] :  < "use-case-execution" : BPMNSystem | messages : (M), processes : (< X :
    ProcessSnapshot | name : "T-Junction controller", tokens : ("Switch_to_P1" T), signals : (SIG),
    subprocesses : (S), state : Running > P) >
                        < "A" : BehavioralRelationship | from : X, to : Tl1 >
                        < "B" : BehavioralRelationship | from : X, to : Tl2 >
                        < "C" : BehavioralRelationship | from : X, to : Tl3 >
                        < Tl1 : FSM | name : "trafficLight", state : "red-amber" >
                        < Tl2 : FSM | name : "trafficLight", state : "amber" >
                        < Tl3 : FSM | name : "trafficLight", state : "red-amber" >
                         =>
                     < "use-case-execution" : BPMNSystem | messages : (none), processes : (< X :
    ProcessSnapshot | name : "T-Junction controller", tokens : ("Switch_to_P1_A_&_C_are_green_Phase_1"
     T), signals : (none), subprocesses : (S), state : Running > P) >
                        < "A" : BehavioralRelationship | from : X, to : Tl1 >
                        < "B" : BehavioralRelationship | from : X, to : Tl2 >
                        < "C" : BehavioralRelationship | from : X, to : Tl3 >
                        < Tl1 : FSM | name : "trafficLight", state : "green" >
                        < Tl2 : FSM | name : "trafficLight", state : "red" >
                        < Tl3 : FSM | name : "trafficLight", state : "green" > .
```

Listing 3 Global Maude rule to switch to phase 1