Modelling and Verification of a Distributed Interlocking System

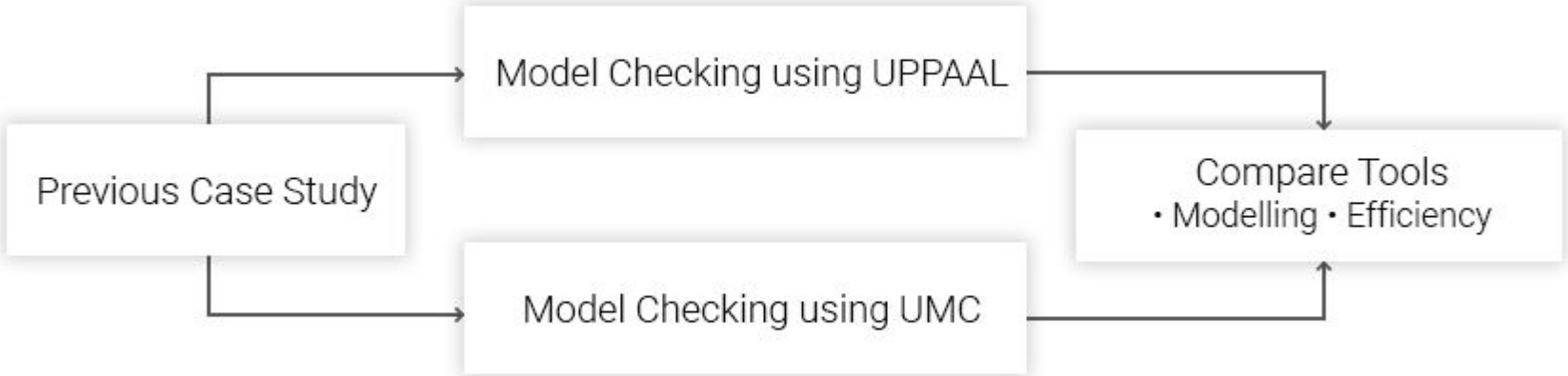
using **UPPAAL** and **UMC**

Van Anh Thi Trinh & Per Lange Laursen
DTU Compute, Technical University of Denmark

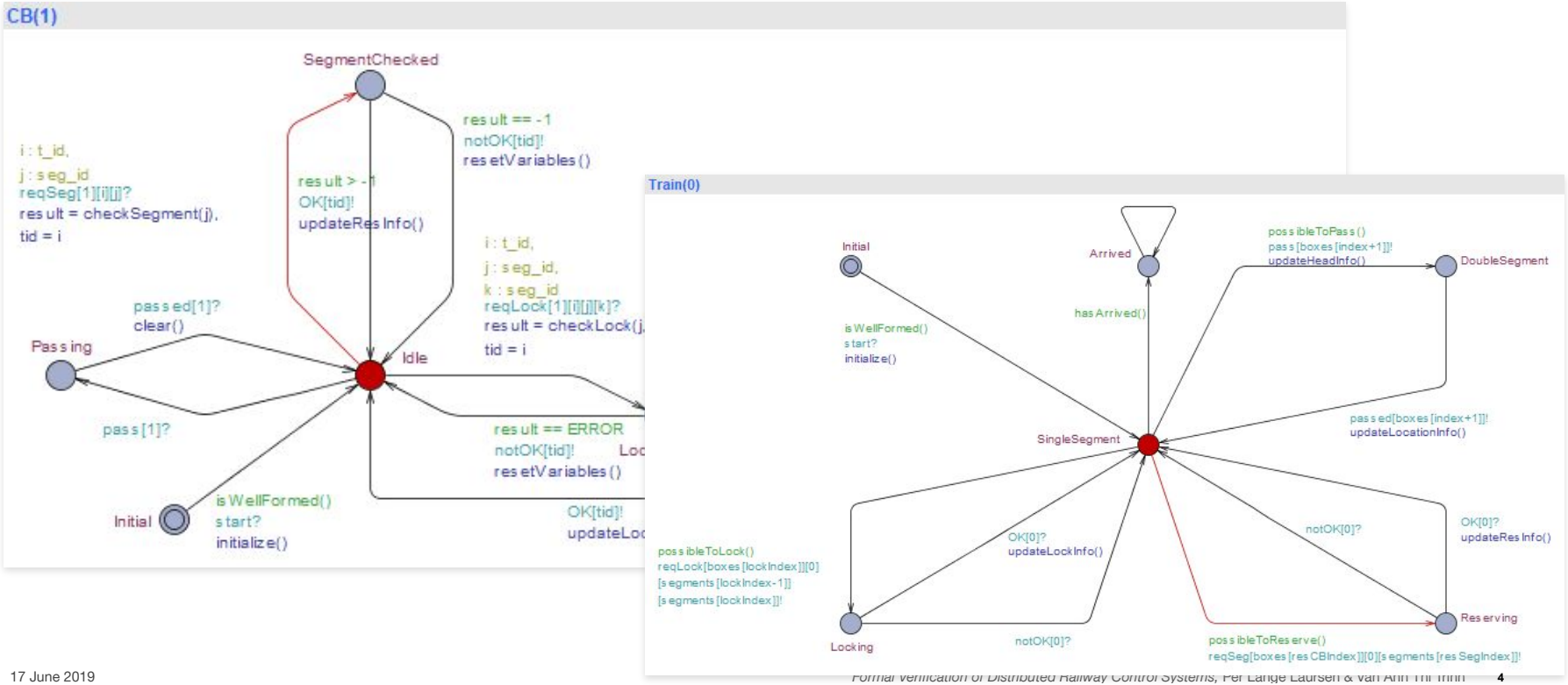
Outline

- Objectives
- Modelling with UPPAAL
- Modelling with UMC
- Testing
- Experiments
- Conclusion and Future Work

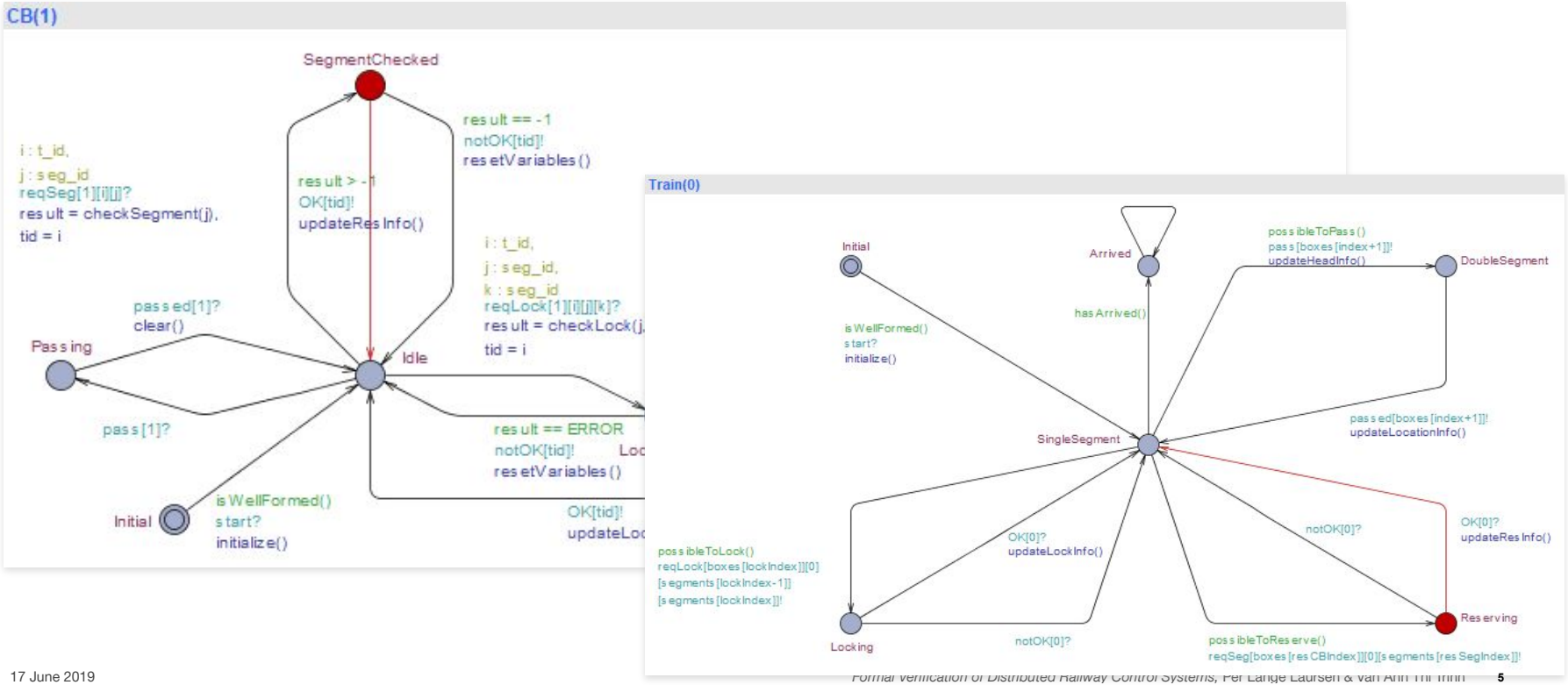
Objectives


Model Checking using UPPAAL
Previous Case Study
Compare Tools
• Modelling • Efficiency
Model Checking using UMC

Modelling with UPPAAL



Modelling with UPPAAL



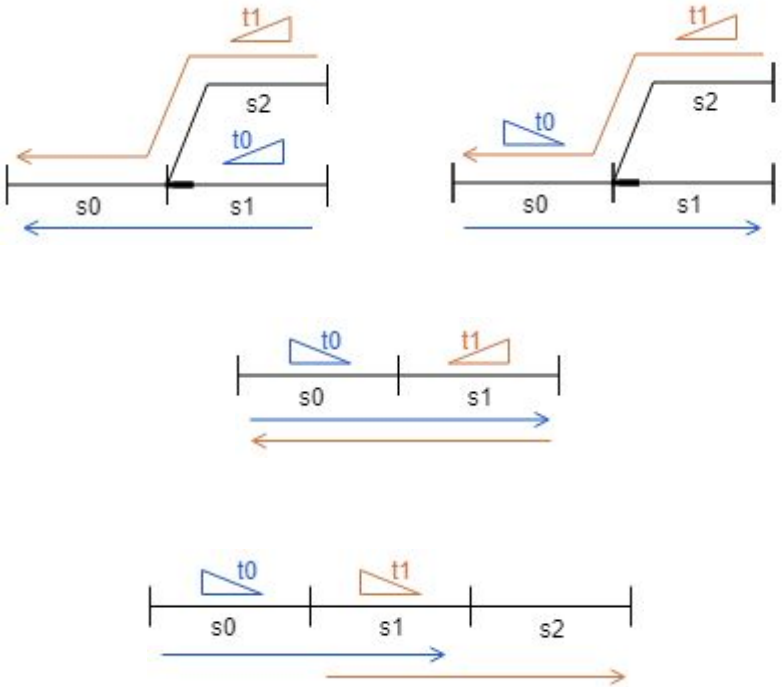
Modelling with UMC

```
16      resCBIndex:int = 1;
17      resSegIndex:int = 1;
18
19      headSeg:int = -1;
20      locks:int = 0;
21
22      resLimit = 1;
23      lockLimit = 1;
24  Transitions
25      SingleSegment -> Arrived {[index == segments.length-1]}
26      Arrived -> Arrived
27
28      SingleSegment -> DoubleSegment {
29          [resSegIndex > index + 1 && lockIndex > index + 1 && index + 1 < segments.length] /
30
31          //updateHeadInfo
32          boxes[index+1].pass;
33          headSeg = segments[index+1];
34      }
35
36      DoubleSegment -> SingleSegment {/
37          //updateLocationInfo
38          curSeg = headSeg;
39          headSeg = -1;
40          if(requiresLock[index +1]){
41              locks--;
42          };
43          index++;
44
```

Testing by Model Checking

Four different properties have been model checked for a collection of testing networks:

- No collision
- No derailment: When a train enters a point from a branch, this must be connected to the stem
- No derailment: A point is not switching when a train is passing it
- Will arrive

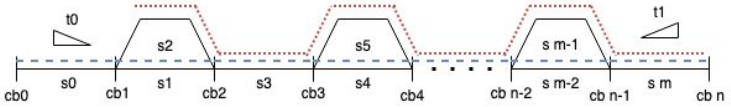


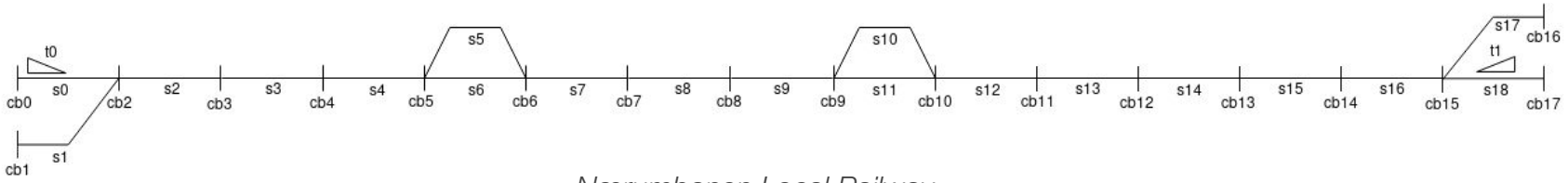
Experiments

The different properties have then been checked with different configurations.

Purpose:

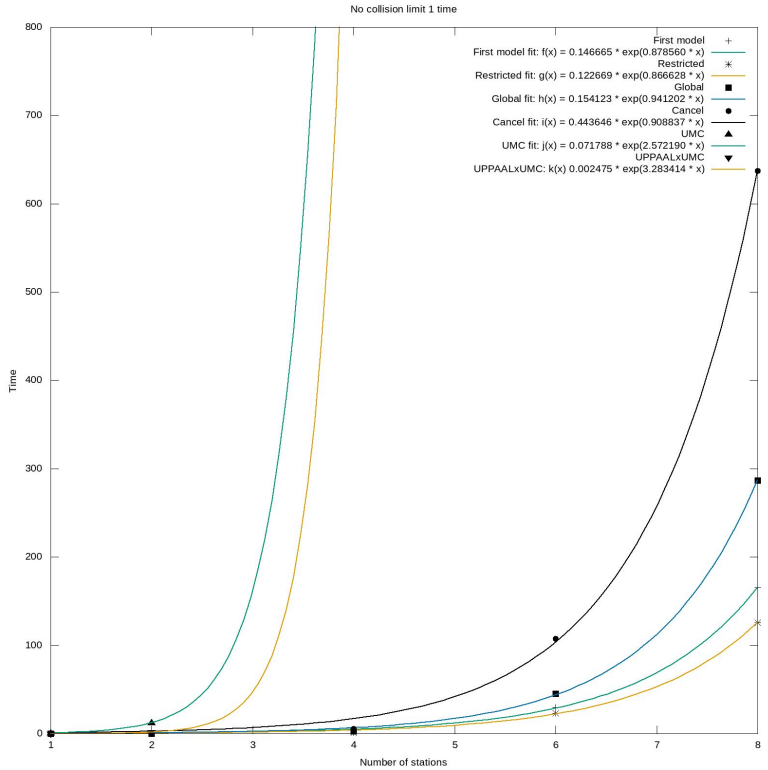
- Scalability
- Real-world networks

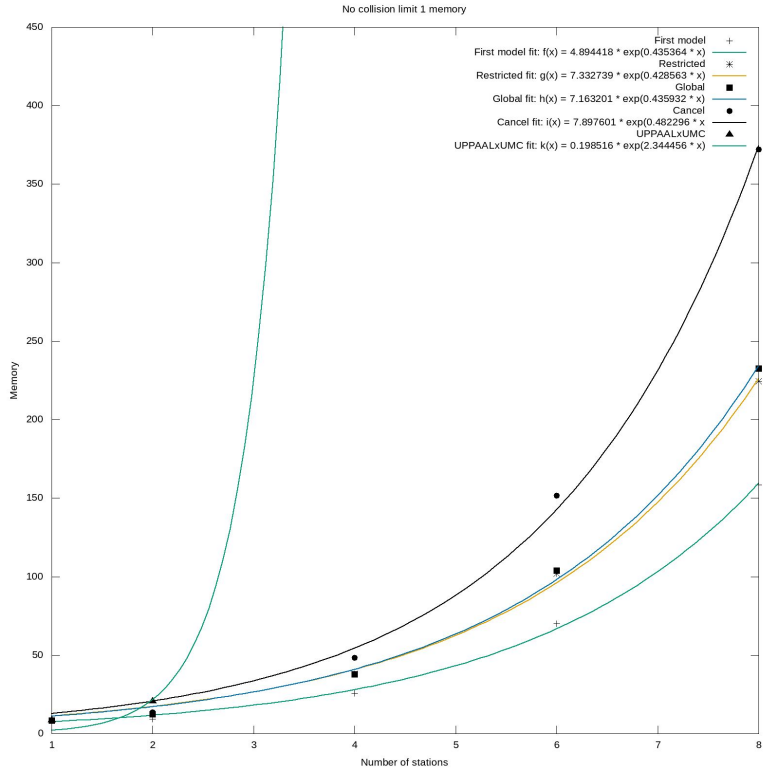




Nærumbanen Local Railway

Experiment Results





Conclusion and Future Work

Conclusion

- Modelling in UPPAAL and UMC is straightforward
- UPPAAL was successfully used to verify a real-world railway network (not yet tried with UMC)

Ideas for future work:

- UMC model optimization
- Alternative/additional strategies to improve verification time and reduce livelocks
- Examining real-time constraints related to network components

Thank you