

Automatic Deployment Space Exploration Using Refinement Transformations

Joachim Denil (Karel de Grote, University of Antwerp)

Antonio Cicchetti (Mälardalen University)

Matthias Biehl (Royal Institute of Technology)

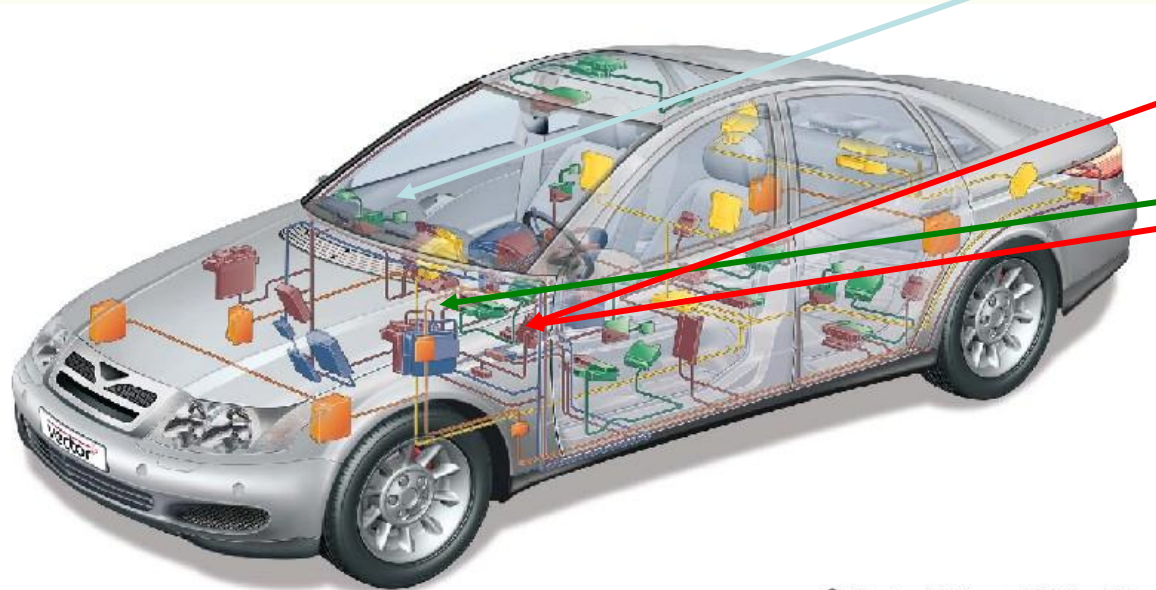
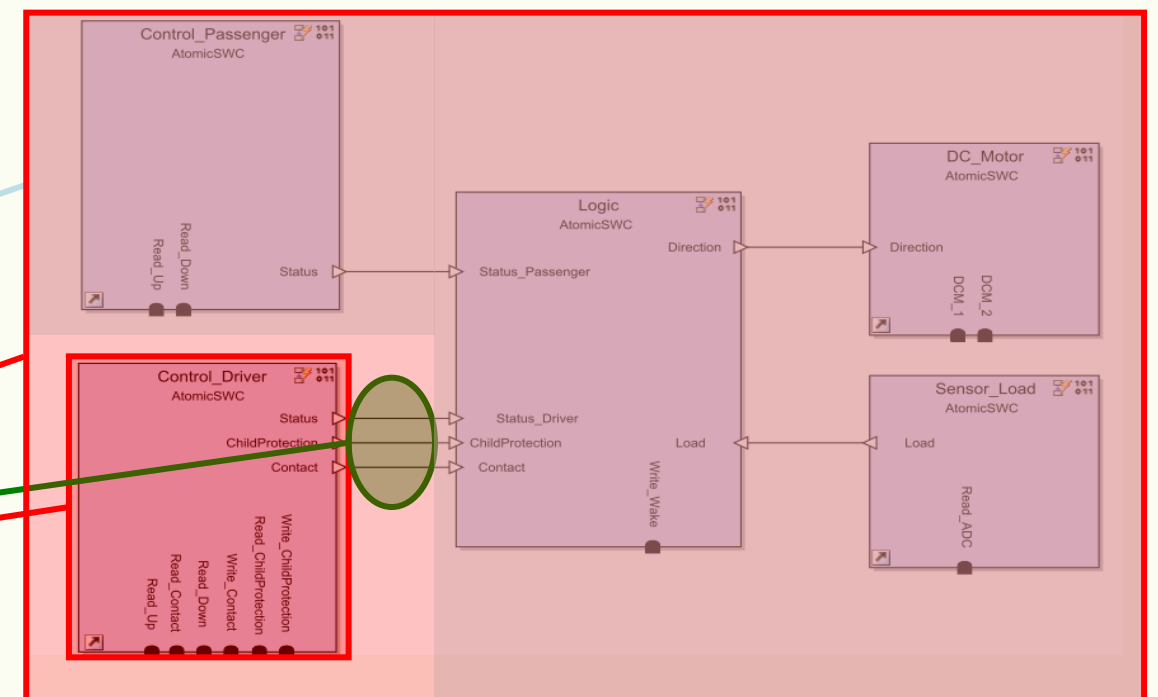
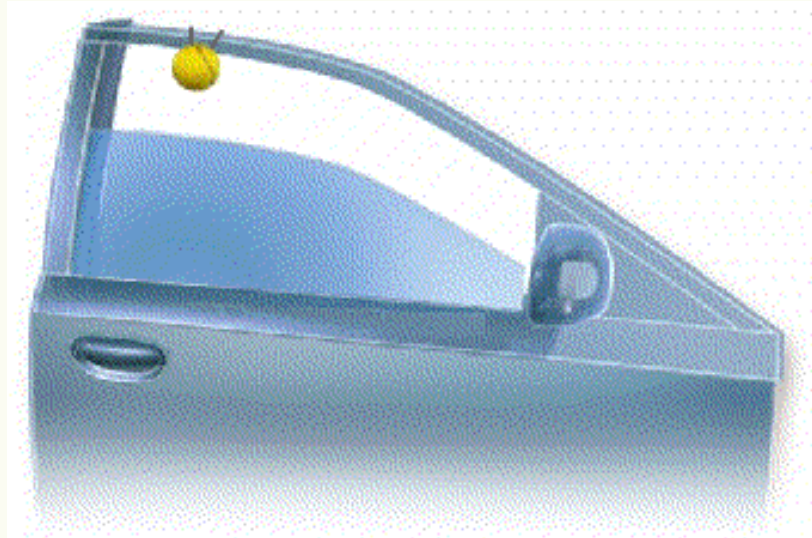
Paul De Meulenaere (Karel de Grote)

Romina Eramo (University of L'Aquila)

Serge Demeyer (University of Antwerp)

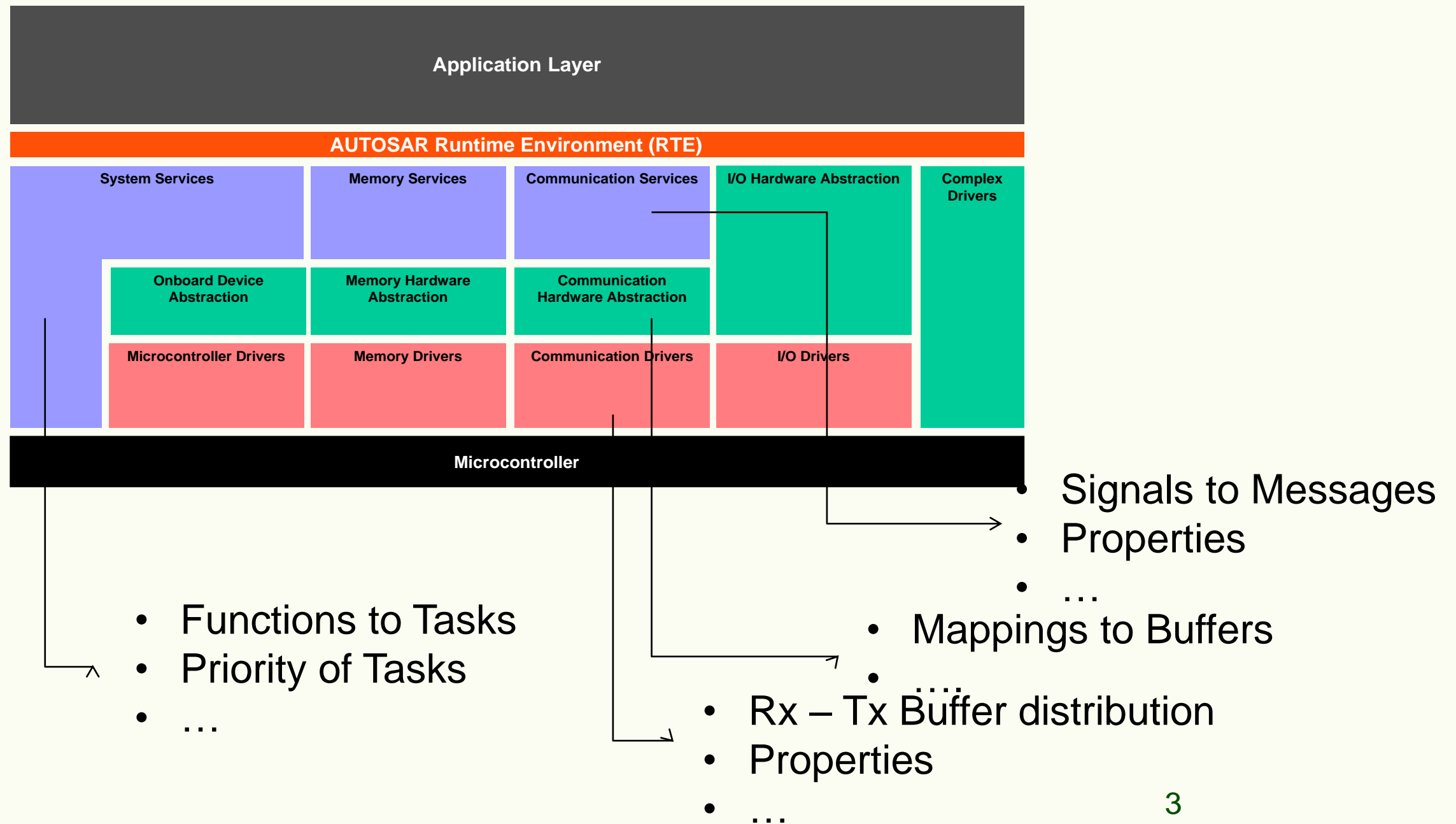
Hans Vangheluwe (University of Antwerp, McGill University)

Introduction



© Vector Informatik GmbH

Introduction to Deployment

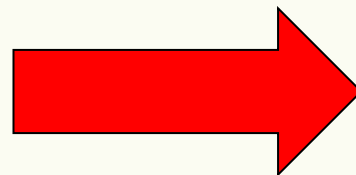
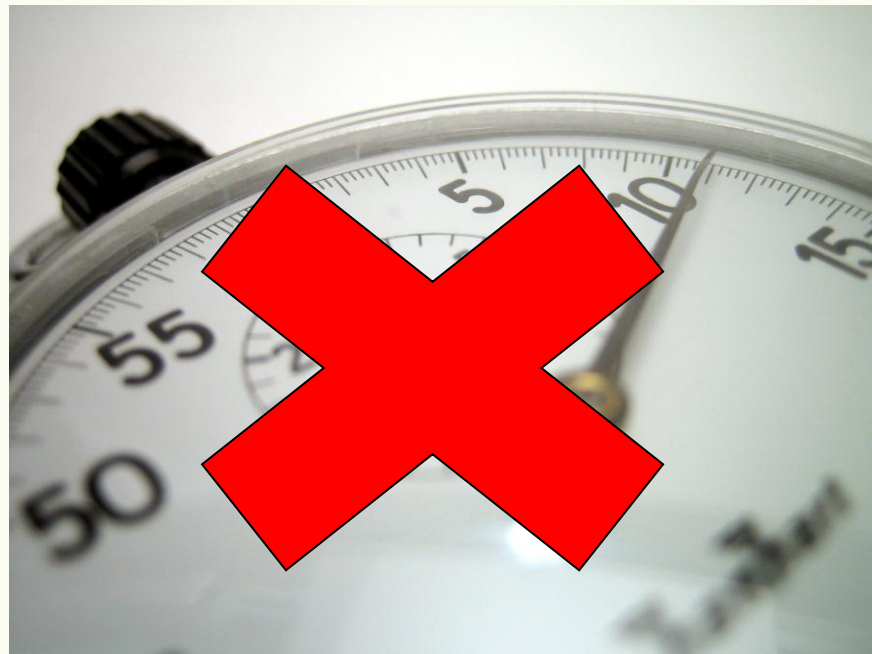


Introduction to Deployment

- AUTOSAR:
 - Framework:
 - Architectural Language based on a Meta-Model

Introduction

Deployment choices impact the real-time behavior

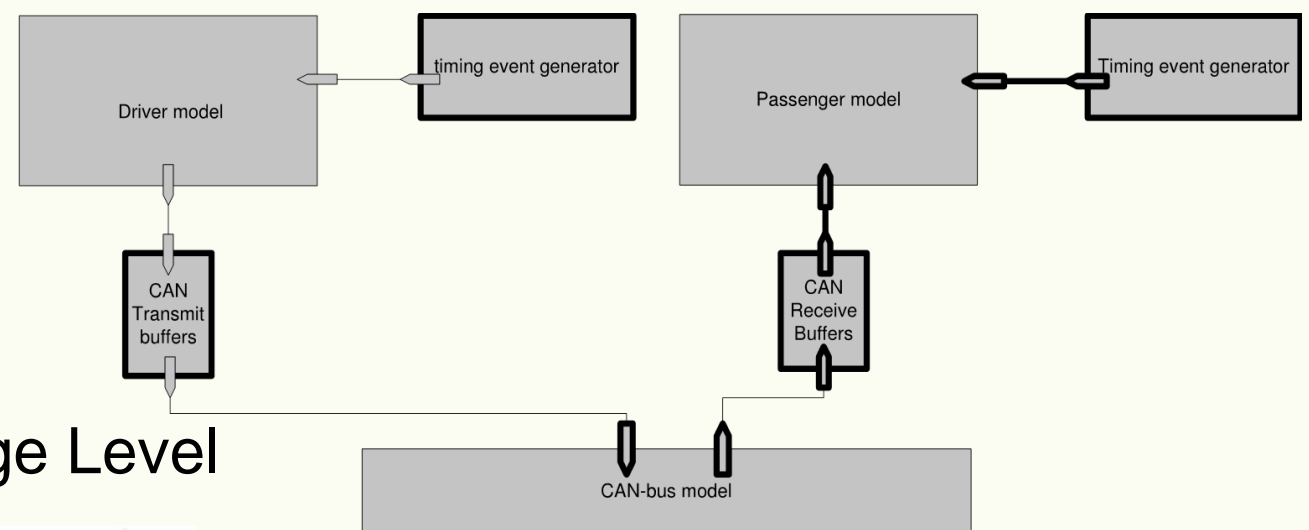


Assessing Real-Time Behavior

System Level



Implementation Level



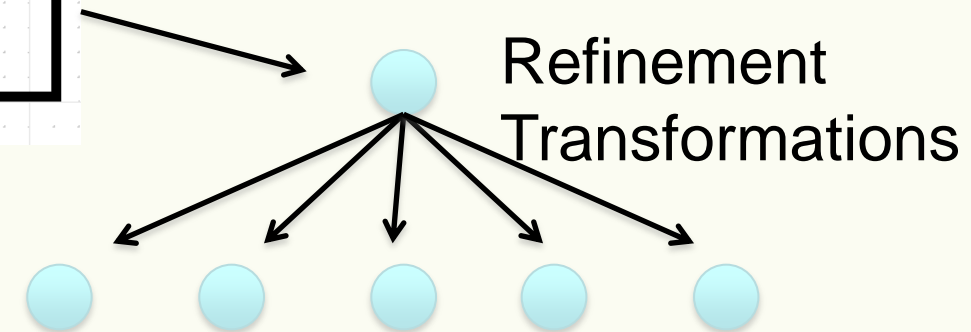
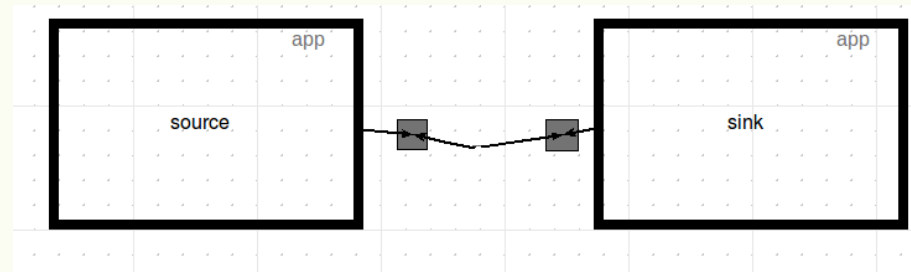
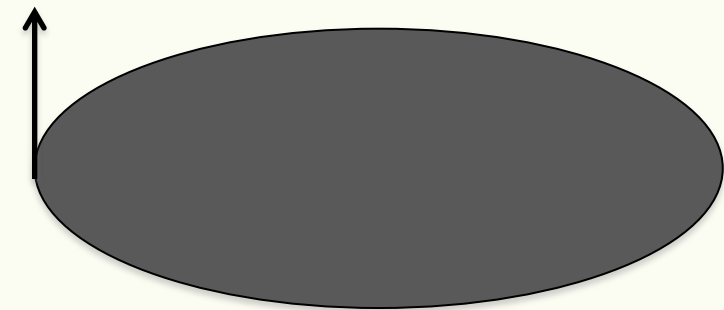
Task/Message Level

L	$f(L)$	$\sum_{j=1}^2 \left\lceil \frac{L}{T_j} \right\rceil C_j$
0	0	0
1	1	3
2	2	3
3	3	3
4	4	4
5	4	4
6	4	4

Evaluation cost

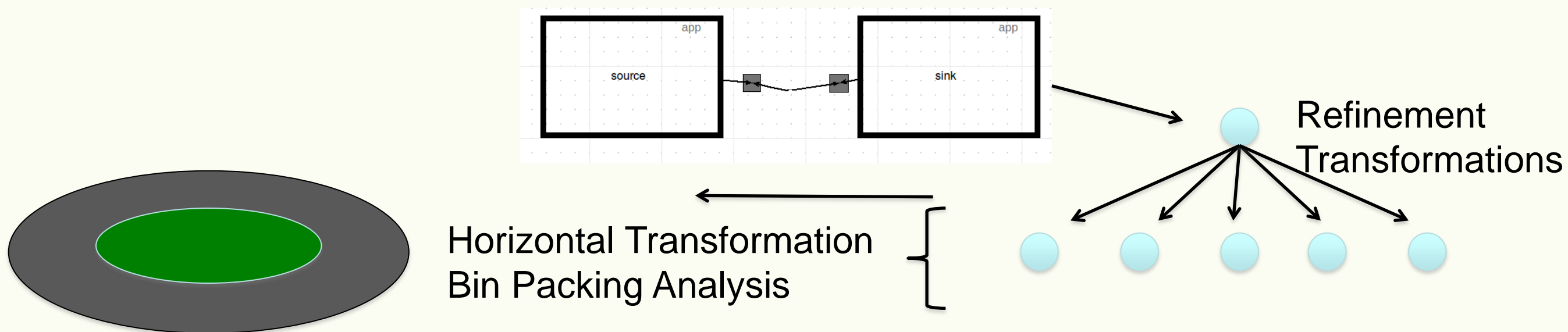
Approach

Full deployment space



- Janus Transformation Language
- Answer Set Programming
- Based on backtracking

Approach

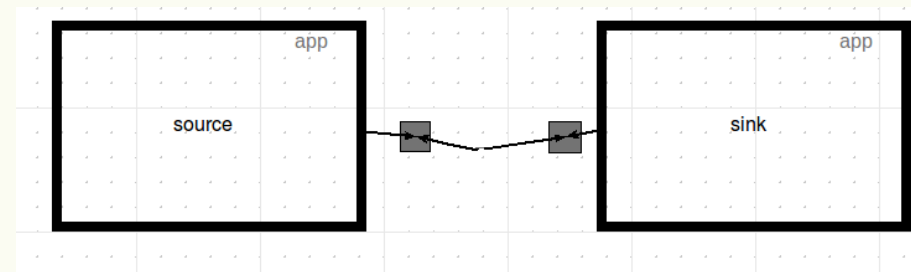


- Transformation:
 - To most appropriate formalism for checking behavior
 - Same level of abstraction

Good Solution:

- Enough temporal space on processor
- Meets Deadlines
- No lost messages

Approach

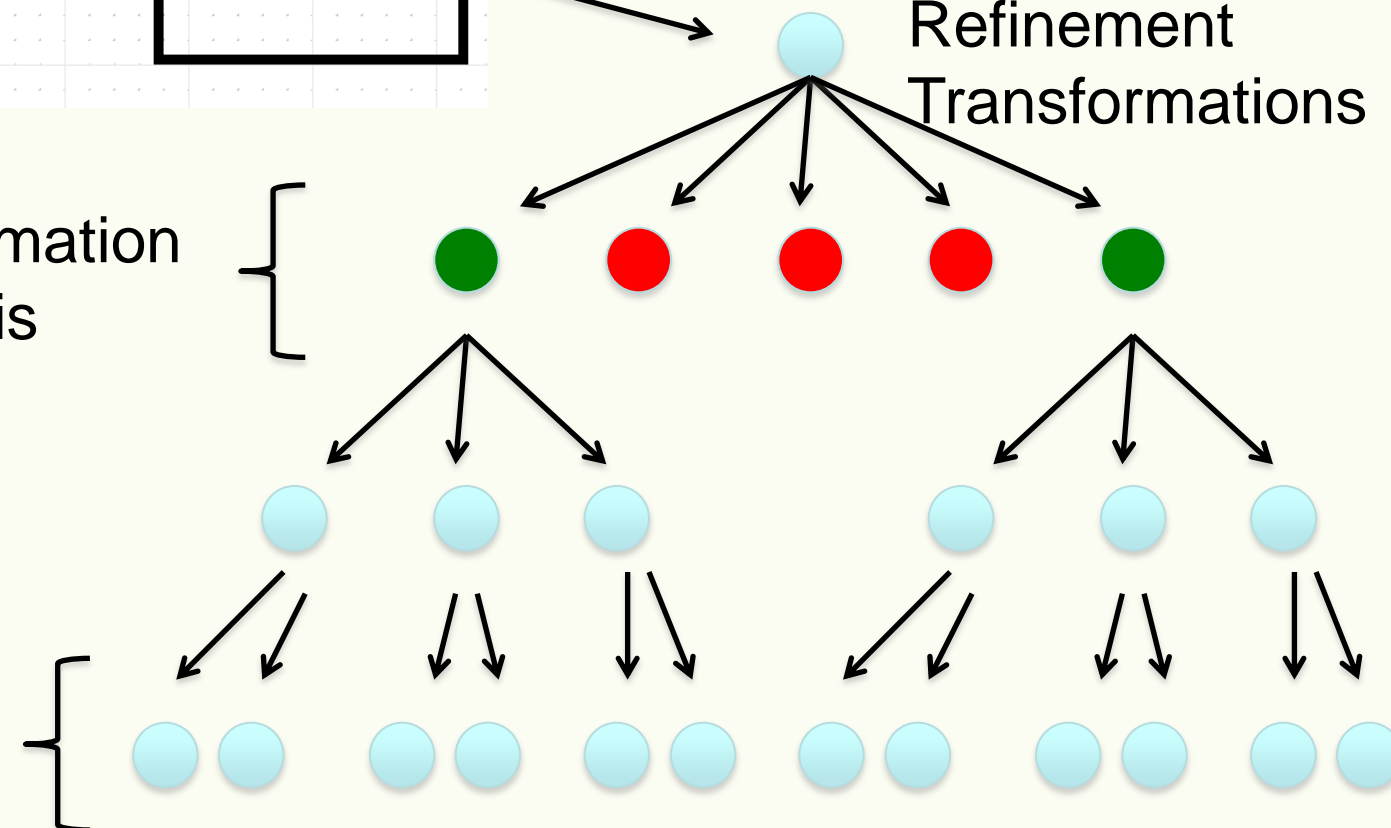


Refinement Transformations

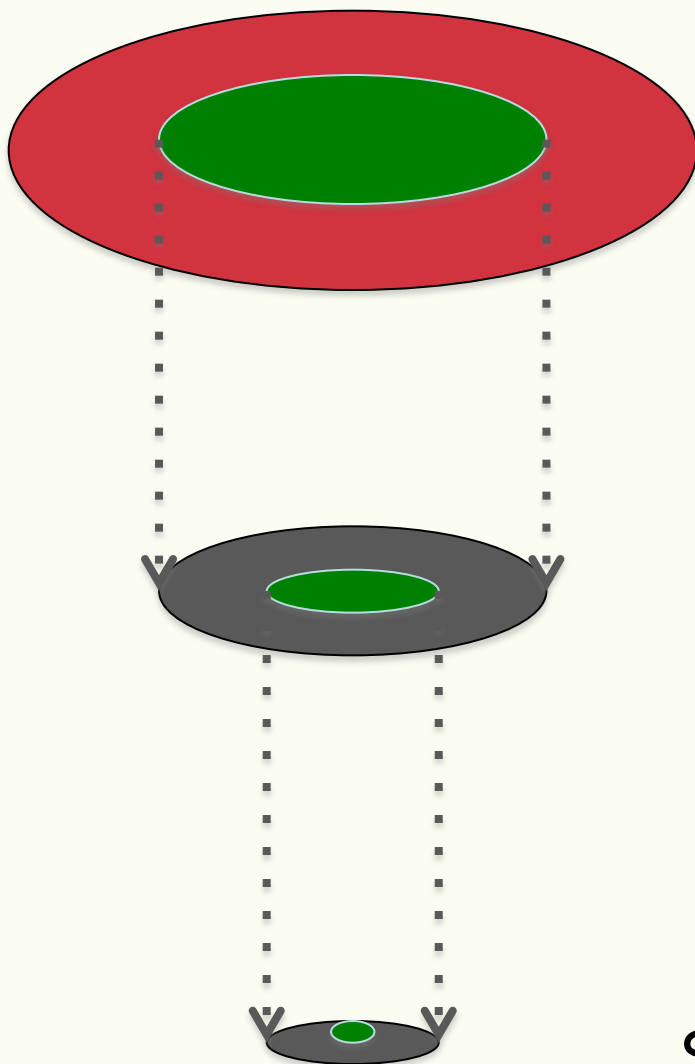
Horizontal Transformation
Bin Packing Analysis

Schedulability
Analysis

Simulation Model {



...



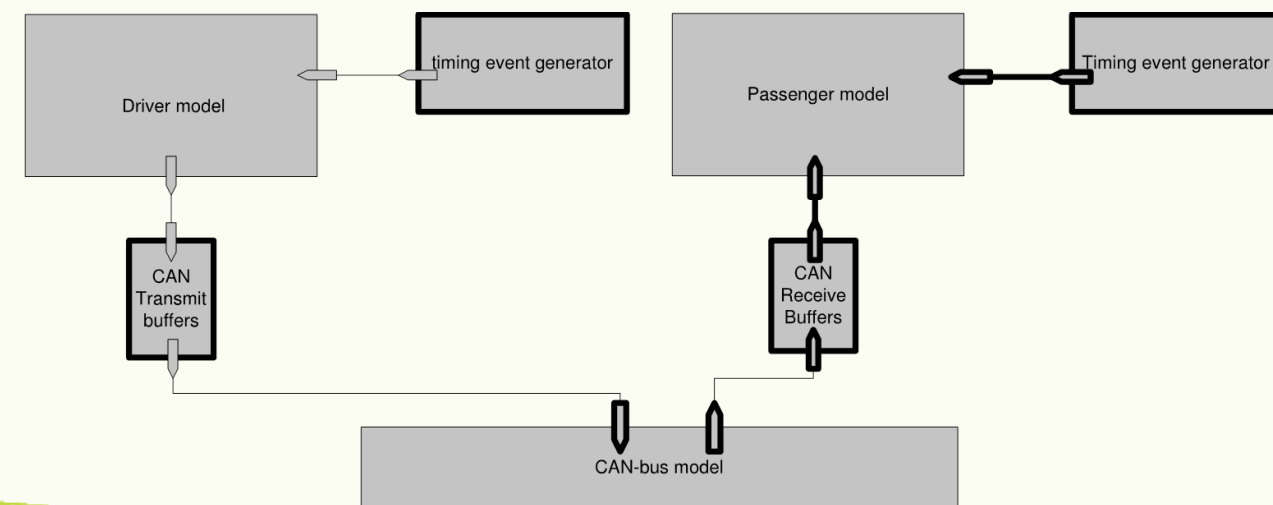
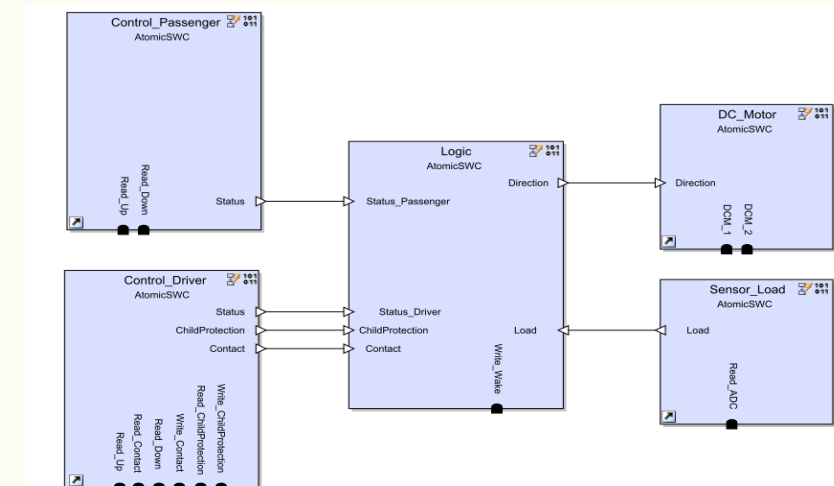
Contribution

- Combinatorial Explosion due to choices
- Evaluation cost rises with every abstraction level

=> Prune solutions as early as possible on different levels of abstraction using an appropriate formalism

Case Study

- Power Window Controller:
 - 1 refinement transformation:
 - Distribute Tx & Rx Buffers of CAN
 - Mapping of Messages to Buffers
 - ...
 - 1 Horizontal transformation:
 - DEVS Deployment model
 - Acceleo (M2T)



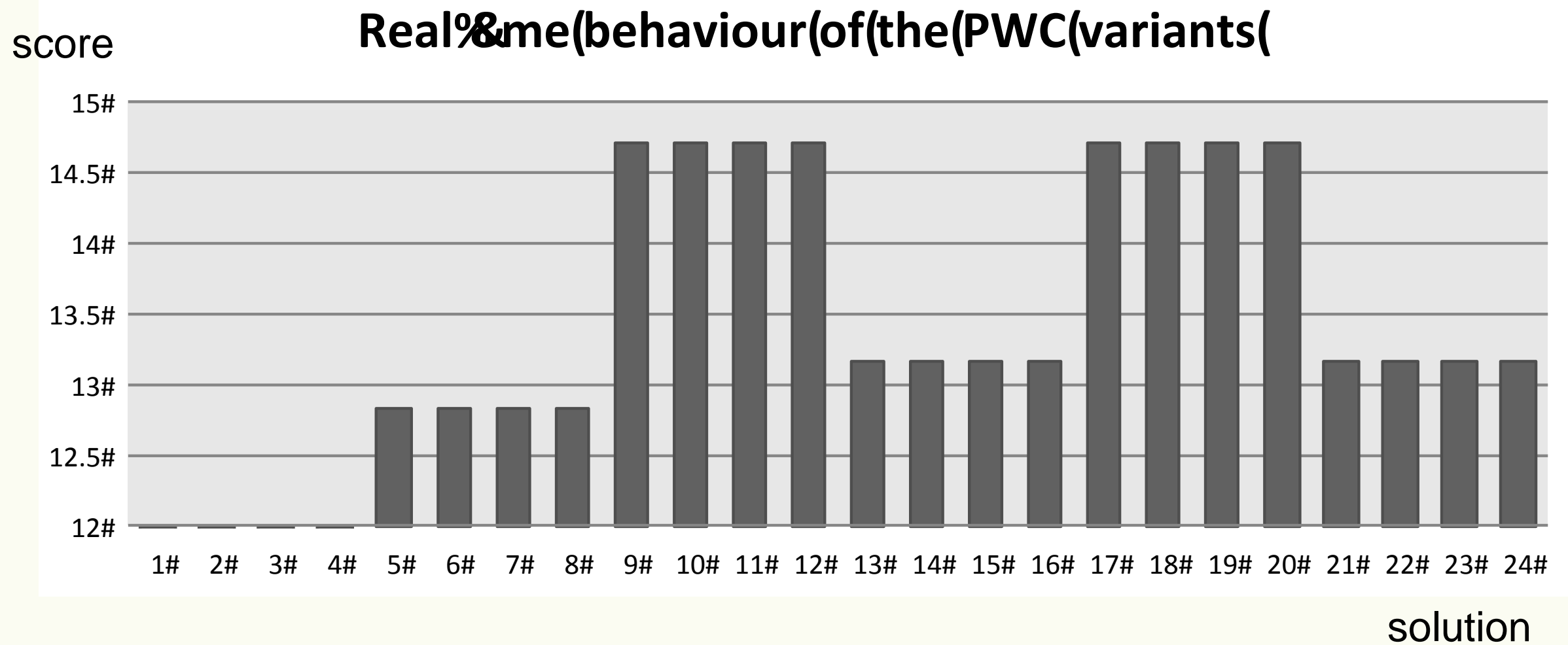
Results

```
##### Target Model 1
model(m1, x_smallAROut).
node(x_smallAROut,"x_//@theSystem/@ecu.0", x_Ecu).
node(x_smallAROut,"x_//@theSystem/@ecu.1", x_Ecu).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@canifConfig",x_CanifConfig).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@canifConfig",x_CanifConfig).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@canifConfig",x_CanifConfig).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@canConfig",x_CanConfig).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@canConfig",x_CanConfig).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@canifConfig/@ipduToHohMap.0", x_IpduToHohMap).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@canifConfig/@ipduToHohMap.1", x_IpduToHohMap).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@canifConfig/@ipduToHohMap.0", x_IpduToHohMap).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@canifConfig/@ipduToHohMap.1", x_IpduToHohMap).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@comConfig/@ipdus.0", x_TxIPDU).
node(x_smallAROut,"x_//@theSystem/@ecu.0/@comConfig/@ipdus.1", x_TxIPDU).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@comConfig/@ipdus.0", x_RxIPDU).
node(x_smallAROut,"x_//@theSystem/@ecu.1/@comConfig/@ipdus.1", x_RxIPDU).
...
prop(x_smallAROut,"x_//@theSystem/@ecu.01", "x_//@theSystem/@ecu.0", x_name, x_Driver).
prop(x_smallAROut,"x_//@theSystem/@ecu.11", "x_//@theSystem/@ecu.1", x_name, x_Passenger).
prop(x_smallAROut,"x_//@theSystem/@ecu.0/@canConfig.0", "x_//@theSystem/@ecu.0/@canConfig",x_cancellation,x_false).
prop(x_smallAROut,"x_//@theSystem/@ecu.0/@canConfig.1", "x_//@theSystem/@ecu.0/@canConfig",x_multiplexing,x_false).
prop(x_smallAROut,"x_//@theSystem/@ecu.1/@canConfig.0", "x_//@theSystem/@ecu.1/@canConfig",x_cancellation,x_false).
prop(x_smallAROut,"x_//@theSystem/@ecu.1/@canConfig.1", "x_//@theSystem/@ecu.1/@canConfig",x_multiplexing,x_false).
prop(x_smallAROut,"x_//@theSystem/@ecu.0/@canifConfig.0", "x_//@theSystem/@ecu.0/@canifConfig",x_enableTxBuffering,x_false).
prop(x_smallAROut,"x_//@theSystem/@ecu.1/@canifConfig.0", "x_//@theSystem/@ecu.1/@canifConfig",x_enableTxBuffering,x_false).
...
```

Results

- Deployment space:
 - 192 possible solutions
 - ... Most are the same because of combination of parameters
- Constrained deployment space, using domain knowledge (Sensitivity Analysis):
 - in JTL refinement transformation
 - 24 solutions left
- Result score based on:
 - OS Idle Time / Bus Idle Time
 - End-to-End latency
 - Response times of tasks
 - Missed deadlines / losing messages = penalty

Results



Discussion

- Large Search Space:
 - Parallelization
 - branches are independent
- Only performance:
 - Others could be added as well
 - Pareto-optimality
- Domain knowledge
 - Can be added in the refinement transformations
 - Evaluate sensitivity of the goal function to the deployment choices

Conclusion

- Automatic Deployment Space Exploration:

- Techniques are feasible
- Optimal solution can be found

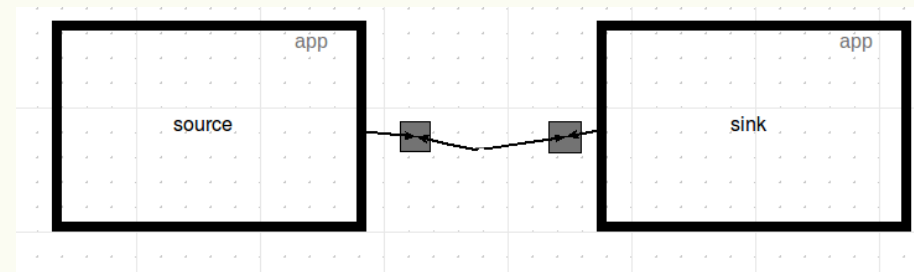
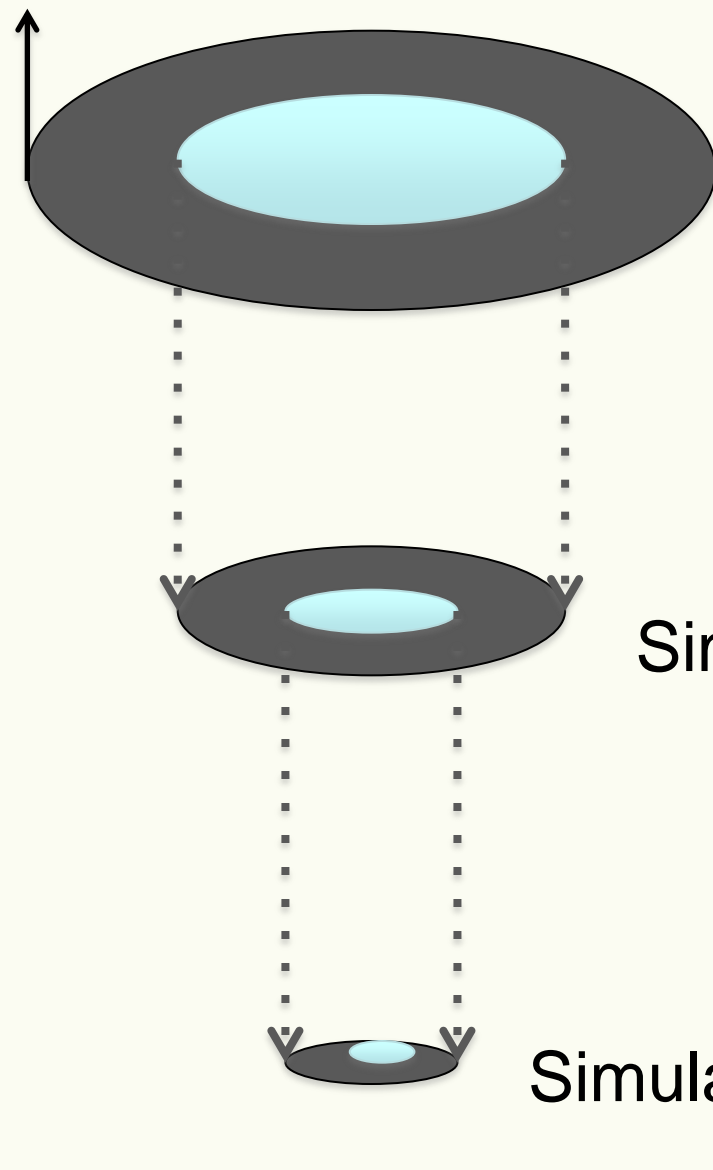


- Need for Transformation Language

- Match/Rewrite separated
- Backtracking
- Black-Box components

Q&A

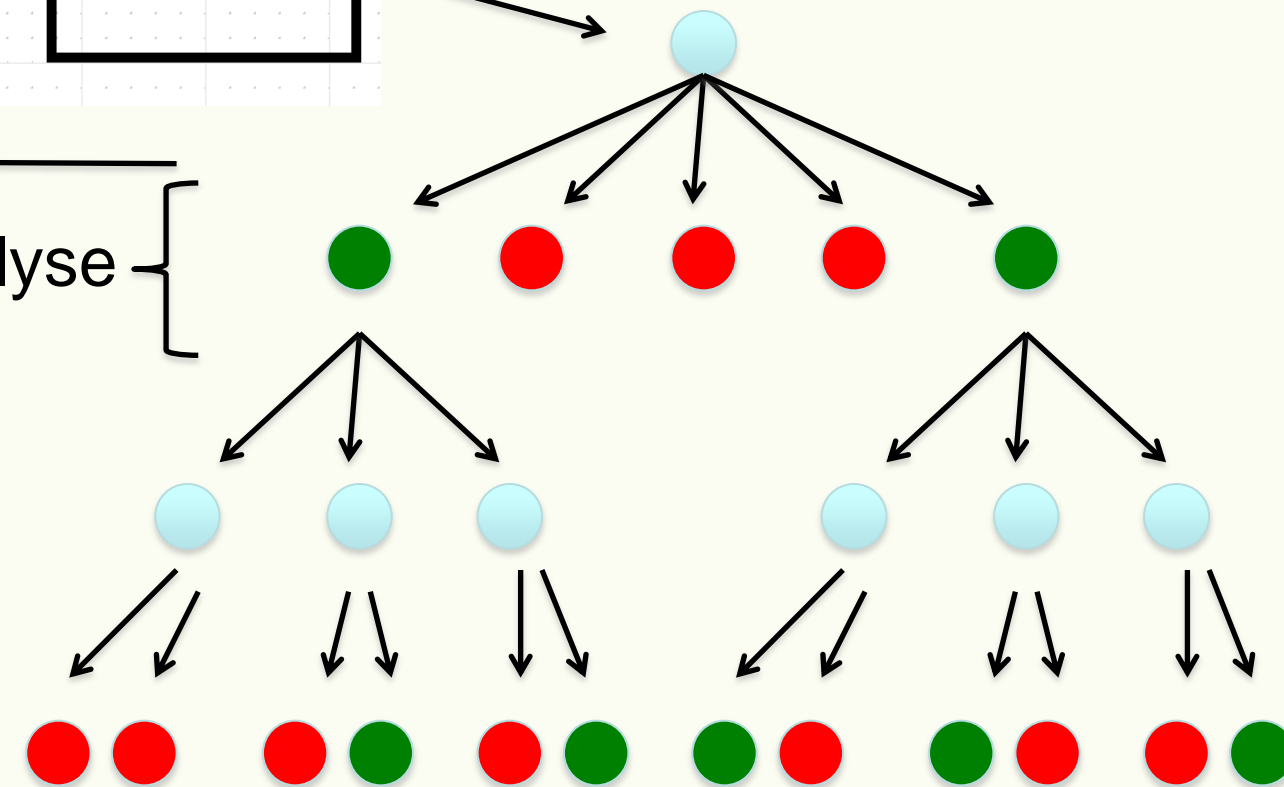
Full deployment space



Simulate/Analyse

Simulate/Analyse

Simulate/Analyse



...