

# The ProB Validation Tool

Michael Leuschel



# ProB in 1 slide

- **Flexible & Extensible Validation Tool for High-level Specification Formalisms**
- Multiple Languages:
  - B, Z, CSP, Event-B, Promela, dSL, ...
- Multiple Validation Technologies derived from operational semantics:
  - Animation, Model Checking, Refinement Checking, ...

# But we have proofs !?

- ▼ ? simplification rewrites
  - ▼ ?  $\wedge$  goal
    - ▶ ✓ remove  $\in$  in right $\in P \rightarrow (N \leftrightarrow S)$ 
      - ✓ functional goal
    - ▼ ?  $\forall$  hyp (inst p)
      - ✓  $\tau$  goal
      - ▼ ? remove  $\in$  in right(p) $\in 1 \dots (\text{size}(p))$ 
        - ▼ ? eh (dom(right(p))=1  $\dots$  (size(p)))
          - ▼ ? sl/ds
            - ▼ ? ah (size(p) $\in \mathbb{N}$ )
              - ▼ ?  $\wedge$  goal
                - ?  $p \in \text{dom}(\text{size})$
                - ?  $\text{size} \in P \leftrightarrow \mathbb{Z}$
                - ✓ PP
                - ▶ ✓ ah ( $k \in \mathbb{N}$ )
  - ▶ ✓ sl/ds
  - ▶ ✓ remove  $\in$  in input $\in 1 \dots s \rightarrow S \setminus N$ 
    - ✓ functional goal



Axioms may be  
inconsistent



Deadlock



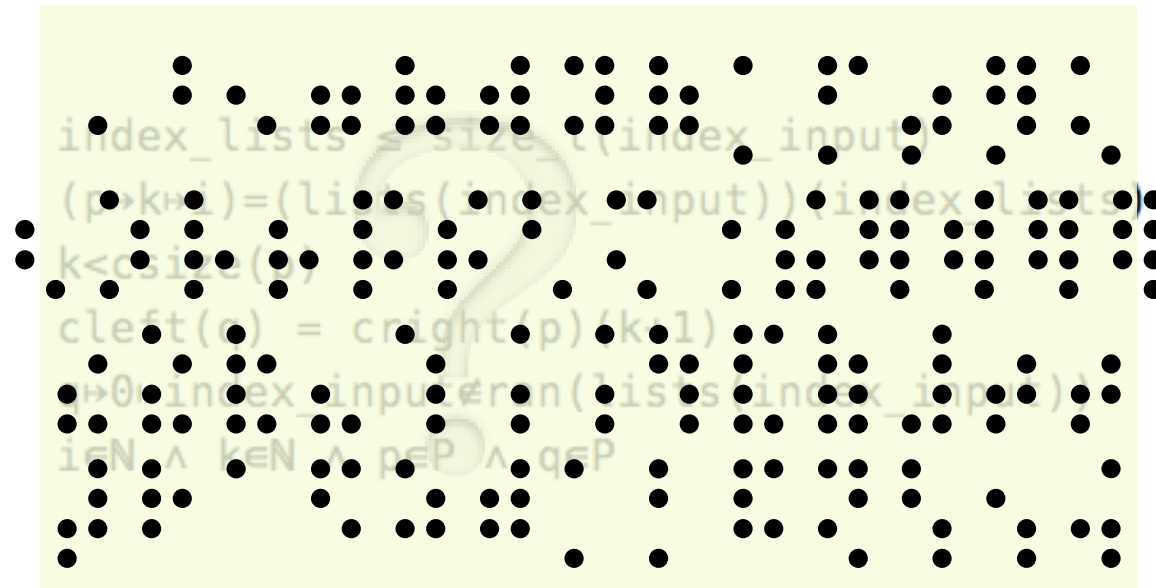


# Domain Experts and Managers need to understand the models



# Formal Models are hard to understand

What others see:  
see:



# Summary: Why additional validation

- It is easy to overlook missing/wrong functionality
- Even proven models contain mistakes
- Even FM experts can make mistakes
  - Specifications get more and more complex
- Only Domain Experts can spot certain errors



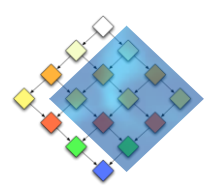
# Additional Validation offered by ProB

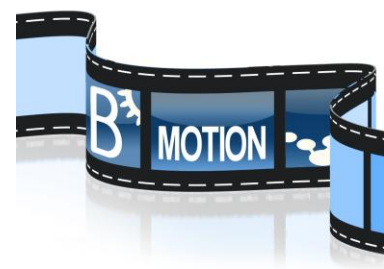
- Animation: show behaviour of model in clear terms
- Graphical Domain Specific Visualization
- Visualization of State Space
- Model Checking
- Refinement Checking



# Some Distinguishing

## Aspects of ProB

- Symmetry Reduction
- Constraint solving 
- Dealing with large Data
- Supports directly high-level formalism
  - Formalisms can be combined (CSP||B)
- Graphical Animation with BMotionStudio

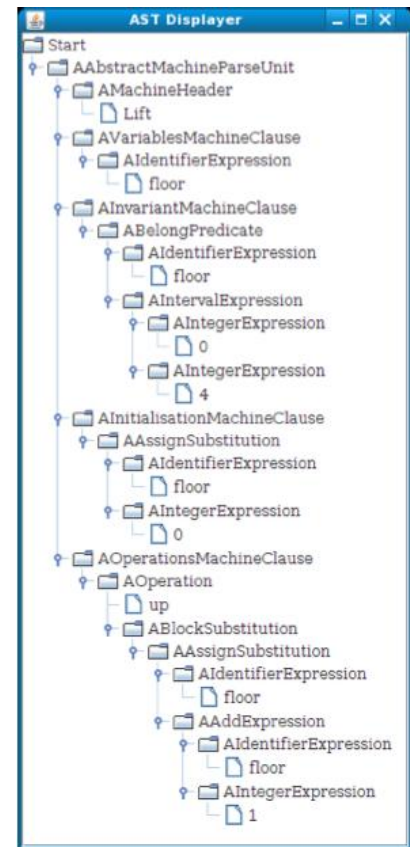


# How ?



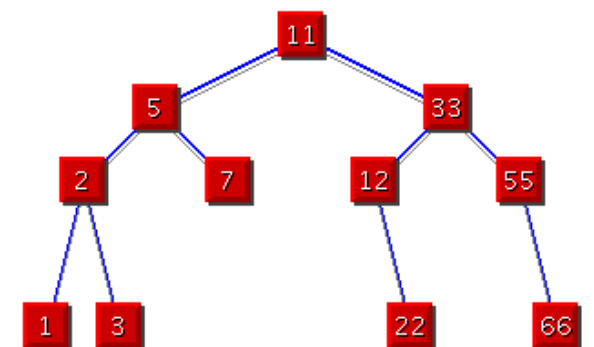
# Scaling up ProB

- Uses Constraint Logic Programming
- Many optimisations
- Handle industrial specifications in ProB:
  - New Parser, Atelier-B compliant
  - New Typechecker (unification-based)
  - Extended Interpreter: almost 100 % support for B



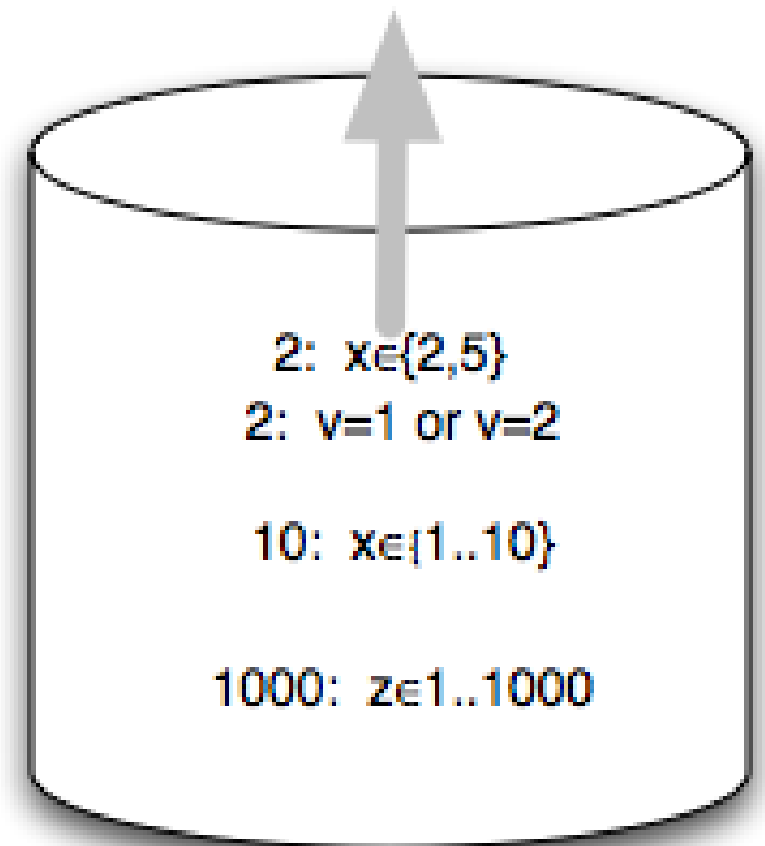
## 2. New Datastructure for Large Sets & Relations

- Before:
  - Sets represented as Prolog lists  
[int(1),int(2)] for {1,2}
- Now:
  - self-balancing AVL-Trees used by ProB kernel if possible



# 3. Improved Constraint Solving

- Heap of Choice Points for enumeration:
- priority: estimated # of solutions
- priority 0: gives (1) AVL tree



Priority Queue of Enumerations/Choice Points

$$dr = \text{ran}(ri) \ \& \ ri = r \sim \ \& \ 3:\text{dom}(ri) \ \& \\ r = \{(1| \rightarrow 1), (6| \rightarrow 2), (7| \rightarrow 3), (8| \rightarrow 4), (9| \rightarrow 5)\}$$



A scenic view of a large lake with mountains in the background and a rocky shore in the foreground. The water is a deep blue-green color, and the mountains are hazy in the distance. The foreground shows a rocky shore with several wooden posts sticking out of the water.

Could we have used other  
technologies ?

Proof

SAT

SMT

# Sieve Experiment

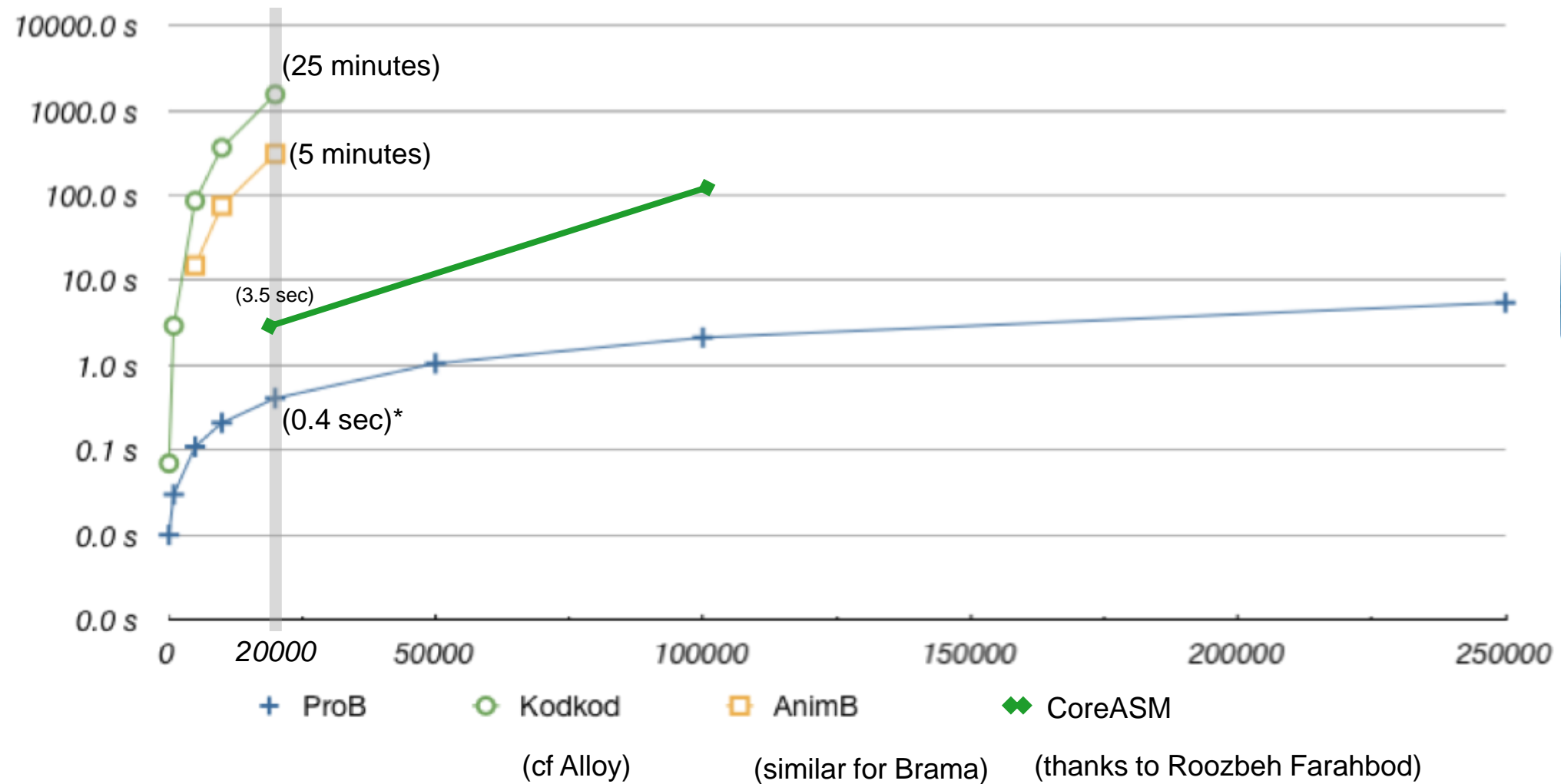
```
numbers := numbers - ran(%n.(n:cur..limit/cur|cur*n))
```

- For cur=2 & limit =
  - 10,000: 0.2 secs *older version:  
out of memory after 2 minutes*
  - 100,000: 2.1 secs
  - 1,000,000: 21.9 secs
- Could be further optimised

# Sieve Experiment



First Step of Sieve Prime Number Algorithm; using a set representation



\*(0.2 sec with ProB 1.3.2-beta5)

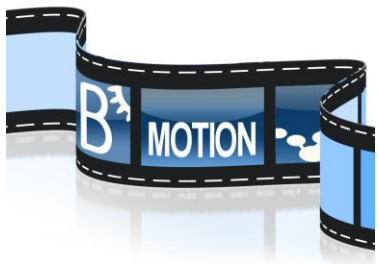
Disclaimer:  
Alloy/Kodkod can be much better than  
ProB at other tasks !



A scenic view of a large lake with mountains in the background and a rocky shore with wooden posts in the foreground. The water is a deep blue-green color, and the mountains are a hazy blue. The sky is a pale blue with some light clouds. The foreground shows a rocky shore with several wooden posts sticking out of the water.

# BMotionStudio

## Problem-specific animation



# BMotionStudio

Tool bar      WYSIWYG editor      Outline View

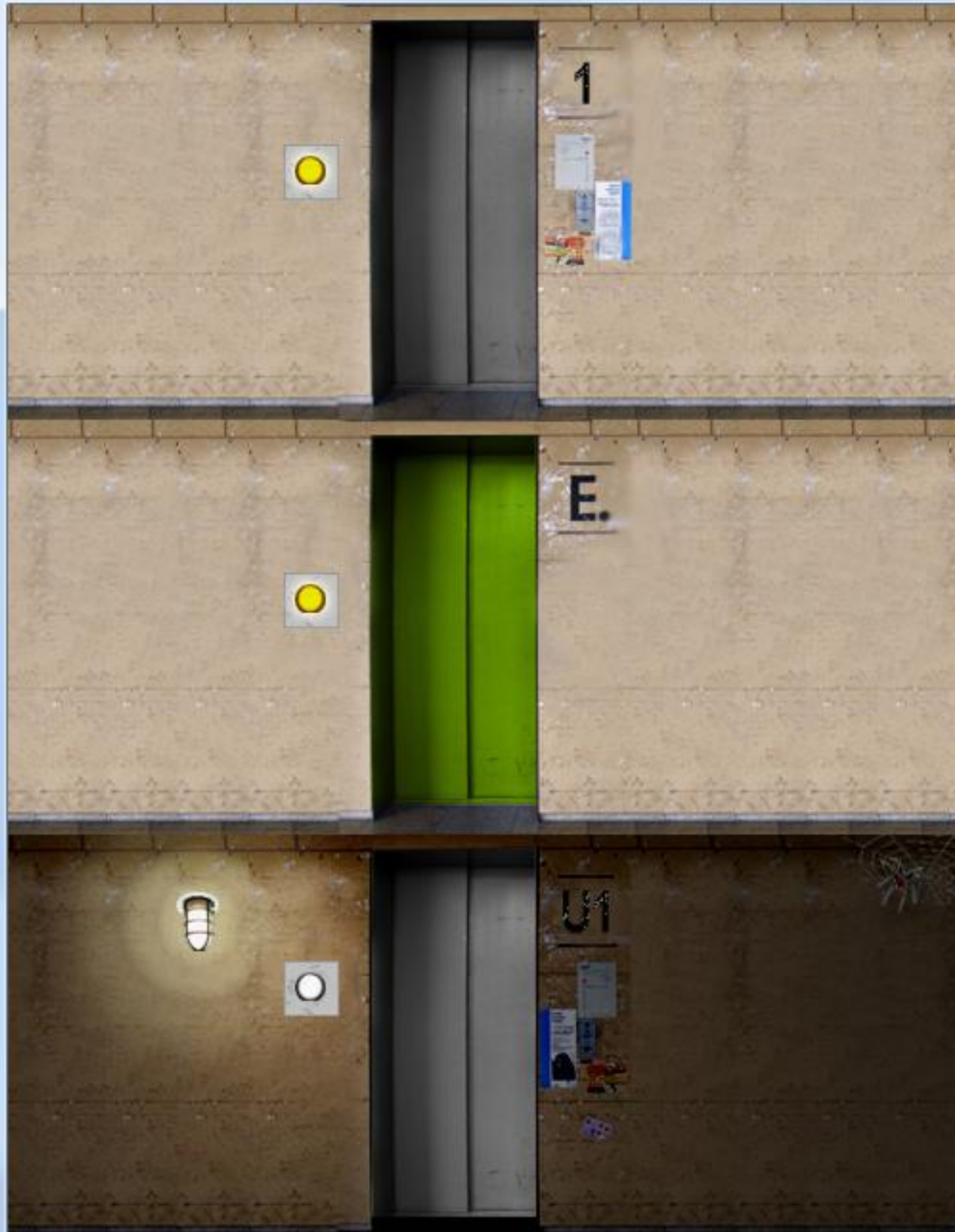
The screenshot displays the BMotionStudio application window. The interface is divided into several sections:

- Tool bar:** Located at the top left, containing various icons for editing and navigation.
- WYSIWYG editor:** The central workspace showing a 3D rendered scene of a control panel with buttons labeled '1', 'E', and 'U1'. A toolbar above it includes 'Locate' and 'Add B-Control'.
- Outline View:** Located on the right side, showing a hierarchical tree structure of the scene's objects, including 'surface', 'bt\_moveleft', 'bt\_movedown', 'user\_interface', 'lbd\_1', 'bt\_in\_1', 'bt\_in\_2', 'bt', 'door\_active', 'bt\_online', 'bt\_offline', and 'bt\_off\_1'.
- Properties View:** Located at the bottom, displaying a table of properties for the selected object.

Property	Value	Event	Mode
name		onclick	execute_operation
id	bt_in_1		
bounds			
x	111		
y	25		
width	55		
height	55		
image			

Properties View





Move Lift UP

Move Lift DOWN

Initialize machine



# Thanks !

- Jens Bendisposto
- Carl Friedrich Bolz
- Nadine Elbeshausen
- Fabian Fritz
- Marc Fontaine
- Michael Jastram
- Li Luo
- Daniel Plagge
- Mireille Samia
- Corinna Spermann
- Dennis Winter



- Michael Butler
- Thierry Massart
- Edd Turner

