# What Is the Use Case for ABS?

## Reiner Hähnle

### Department of Computer Science, TU Darmstadt

Joint Work with Crystal C. Din, Einar B. Johnsen, Ka I Pun, S. Lizeth T. Tarifa

Software Engineering Group

International ABS Workshop 2018, TU Darmstadt

# Strong Points of ABS



- Simulation and Analysis
    - Still a unique selling point!
- Natural models in differing domains
- Modeling cost, deployment, variability
- Formal semantics
- Good language documentation
  (Thanks to crystal & Rudi!)
- Deployment on distributed architecture

# Weak Points of ABS

Despite Rudi's & others' heroic efforts:

- ▶ So far, no generally usable IDE
- ▶ All analysis tools have serious limitations

- ▶ There is no model checker for ABS
- ▶ No white paper or up-to-date tutorial
  - ▶ Lack of teaching materials
- ▶ Only Erlang backend supports all features
  - ▶ Scaling? Efficiency?

# Weak Points of ABS

Despite Rudi's & others' heroic efforts:

▶ So far, no generally usable IDE

▶ All analysis tools have serious limitations

▶ There is no model checker for ABS

▶ No white paper or up-to-date tutorial
  ▶ Lack of teaching materials

▶ Only Erlang backend supports all features
  ▶ Scaling? Efficiency?

It doesn't matter if the glass is half empty or half full.

There is clearly room for more wine.

## Existing Use Cases

1. ABS as model of existing system (CompuGene, Fredhopper, SF 4.0, Yarn)
   - ▶ Simulation and analysis of ABS model that are impossible for modeled system
   - ▶ Create conjectures about modeled system
   - ▶ Create test cases / experiments for modeled system
   - ▶ Use analysis of ABS model to optimize modeled system

   Challenge: model extraction, scalability of simulation & analysis

# Attempting to Classify Use Cases for ABS

## Existing Use Cases

1. ABS as model of existing system (CompuGene, Fredhopper, SF 4.0, Yarn)
   - ► Simulation and analysis of ABS model that are impossible for modeled system
   - ► Create conjectures about modeled system
   - ► Create test cases / experiments for modeled system
   - ► Use analysis of ABS model to optimize modeled system

   Challenge: model extraction, scalability of simulation & analysis

2. ABS as formalization of informally given system (Form*, Memory model, SLA)
   - ► Modeled system does not exist in executable / experimentable form
   - ► Investigate consequences of different modeling decisions
   - ► ABS model opens up new usage scenarios (training, prediction, deployment, ...)
   - ► ABS model replaces informal artifact as reference (documentation, certification)

   Challenge: model validation, scalability of analysis

The definitive simulation language, following Dahl & Nygaard

# A Different Vision

The definitive simulation language, following Dahl & Nygaard

## A New Use Case

System development from scratch with ABS

- A variant of Model-driven Development
- Early prototyping, visualization
- Analysis of incomplete model possible

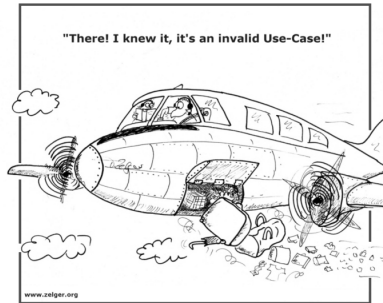Challenge: Generation of correct and efficient code, IDE, libraries

# General Remarks About Use Cases

- ABS was developed to model software, but is more generally applicable
- Stay away from application areas that have firmly entrenched methods:
  Automotive, embedded systems, . . .
- Working with domain experts from the start is crucial

# General Remarks About Use Cases

▶ ABS was developed to model software, but is more generally applicable
▶ Stay away from application areas that have firmly entrenched methods: Automotive, embedded systems, . . .
▶ Working with domain experts from the start is crucial



"There! I knew it, it's an invalid Use-Case!"

www.zelger.org

**Inevitable Consequences**