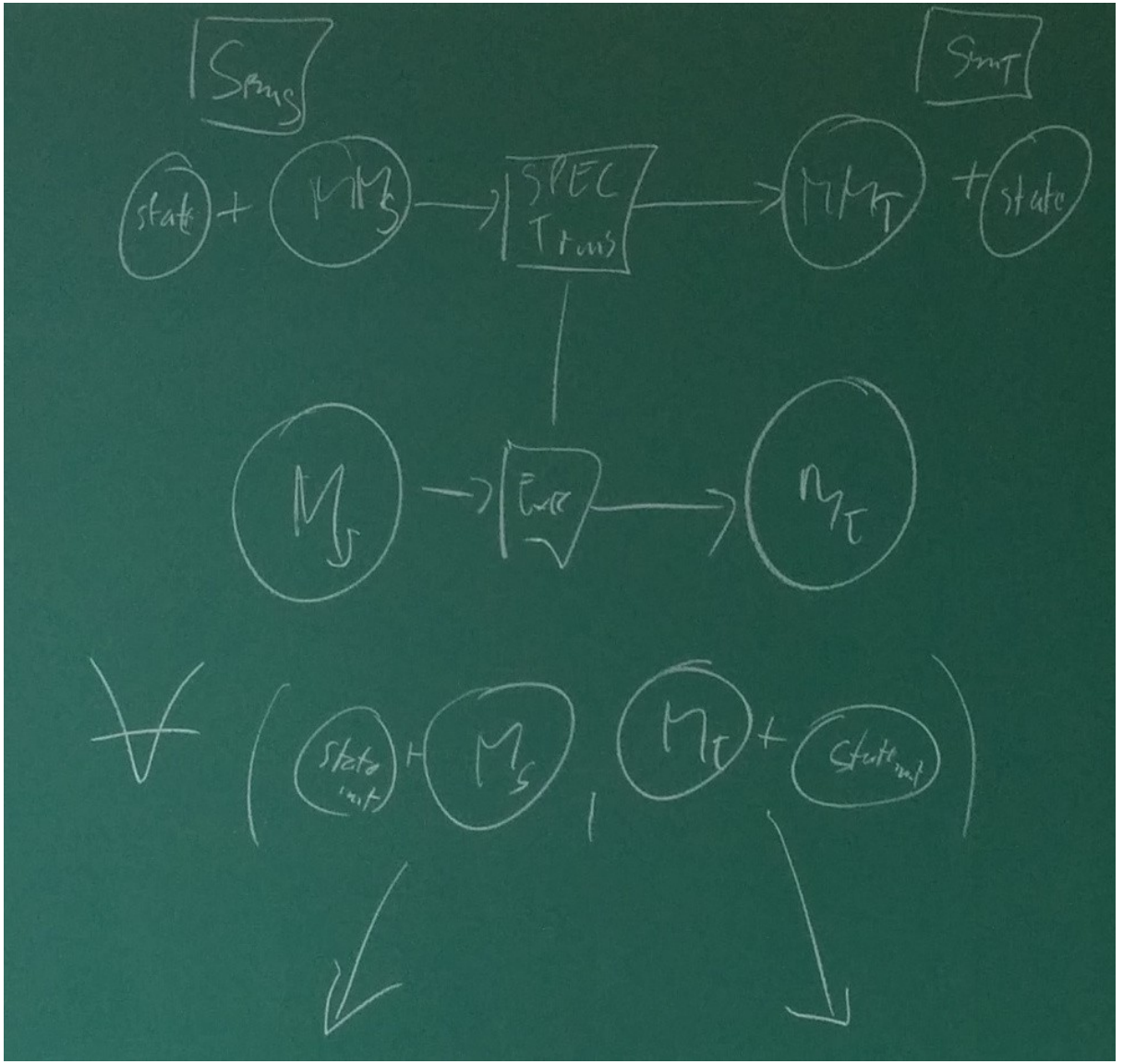
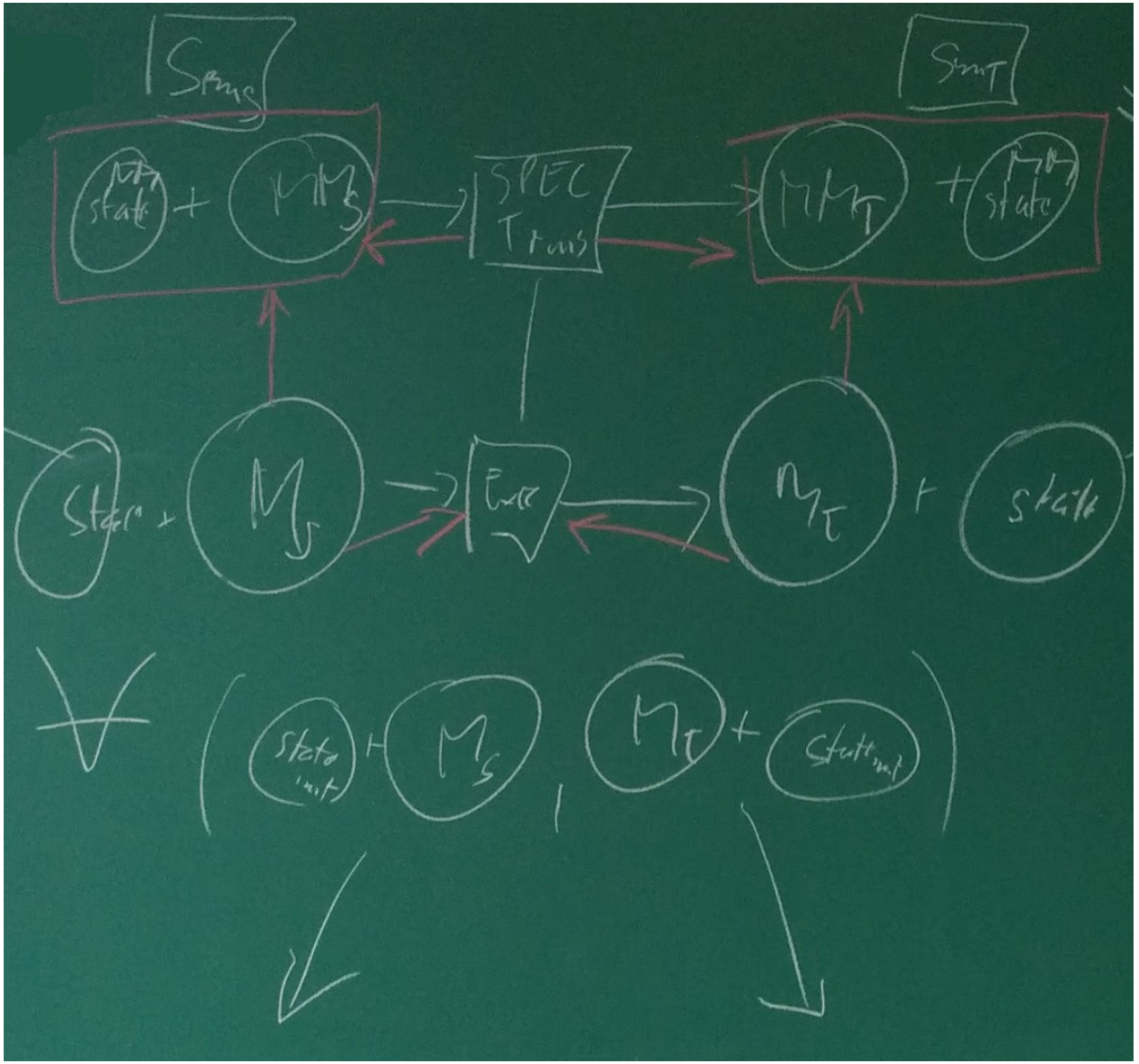
Semantics Preserving Transformations

Holger Giesse, Ekkart Kindler, Mark Lawford,
Tom Maibaum, Fernando Orejas & Jens Weber

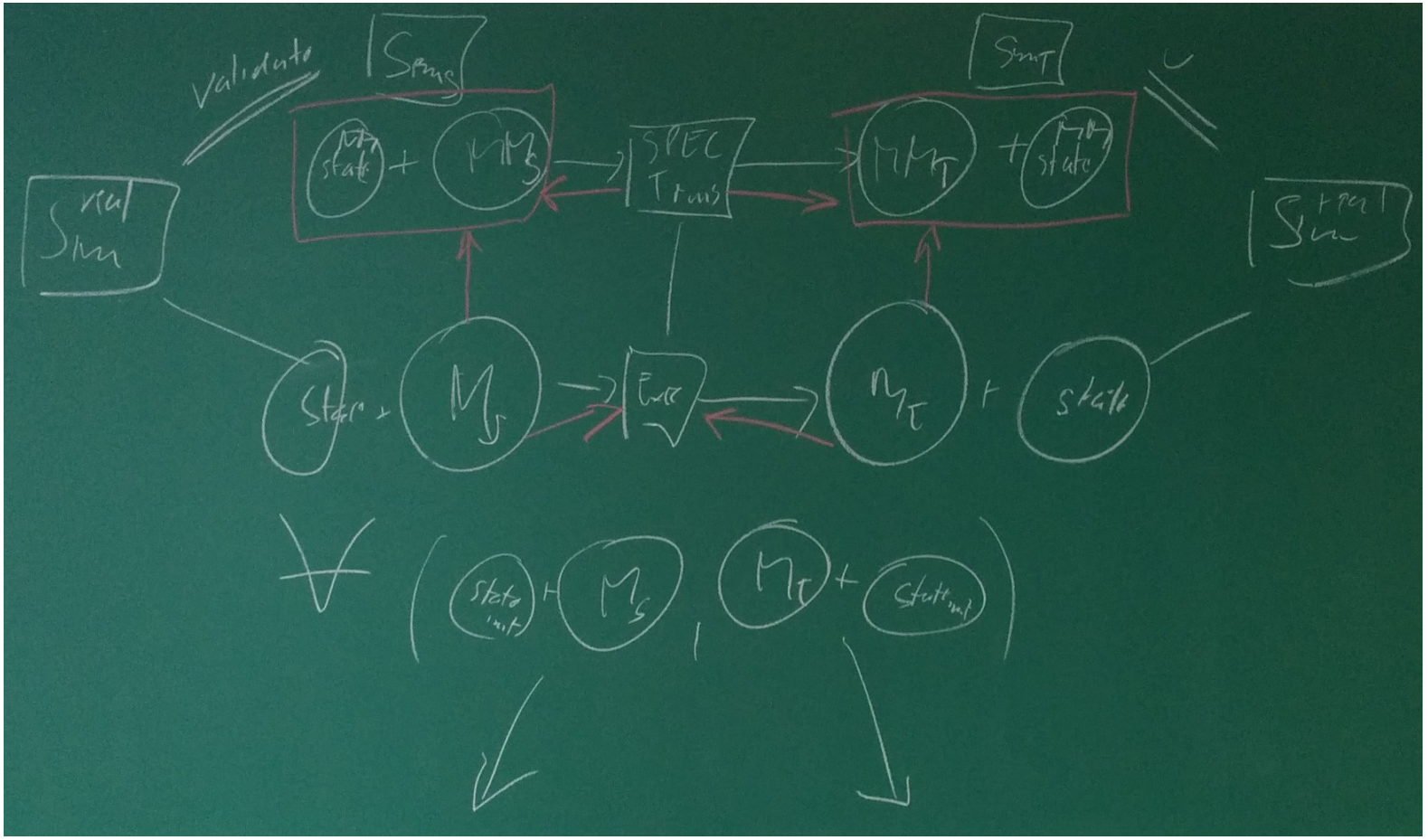
Holger's Framework



Tom's version of Holger's Framework



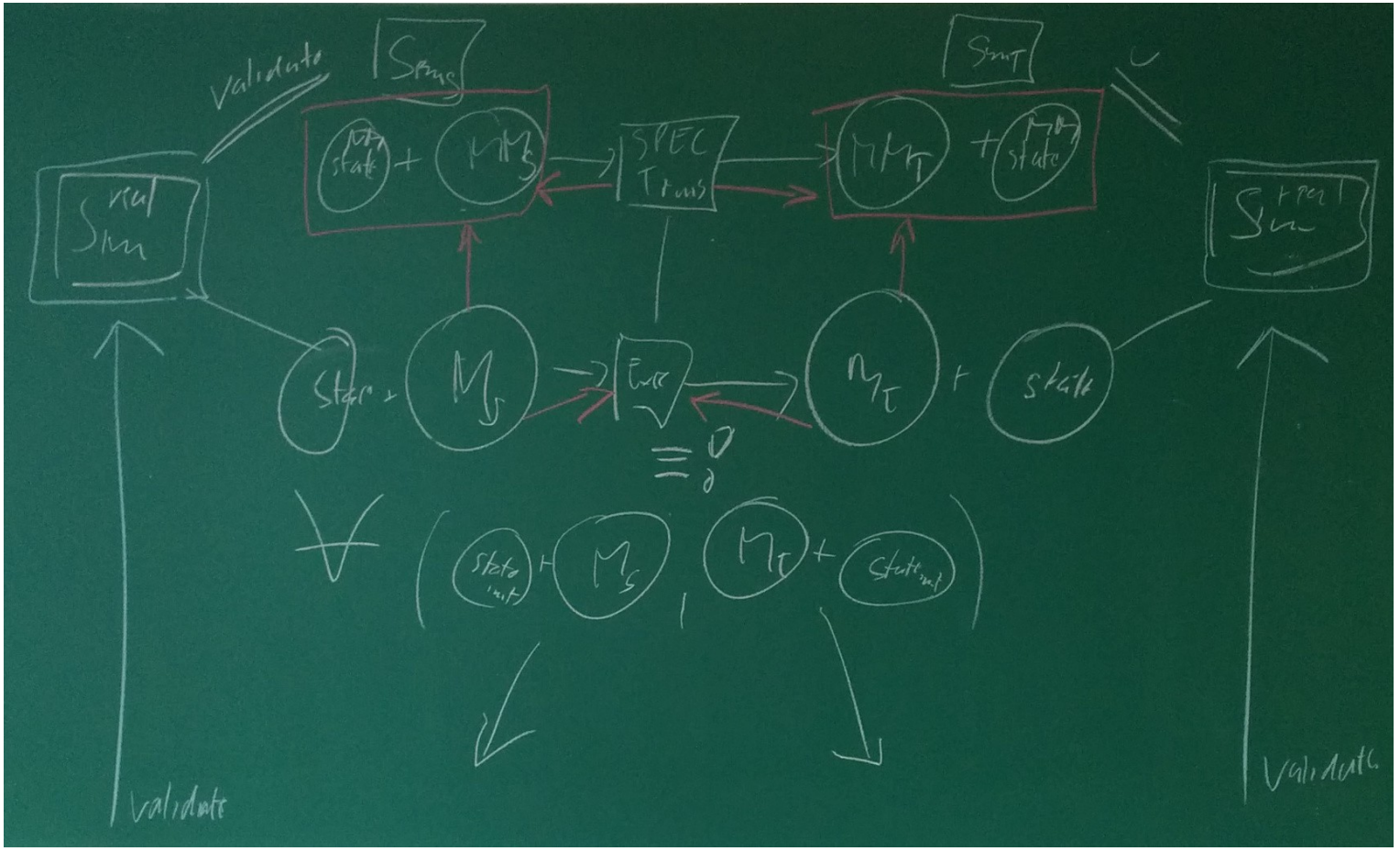
How do you do it in practice?



Semantics Preserving BX in Engineering Practice

- Typically do not have formal semantics so can't prove BX.
- You can validate BX rules via simulation
- Back to back testing of simulation and code reference may be helpful here
- You are effectively producing an assurance case that the transformation is valid in case when formal semantics is not used/available

How do you define “sematics preserving”?



How do you define semantic equivalence?

- Bisimulation? Maybe too strong, might want simulation
- For preservation of LTL vs. CTL would want a different definition of semantic equivalence
- Can have different notions of equivalence at different levels of abstraction
- More appropriate measure of “nearness”
  - How much does it change the risk?