1. Introduction (myself, topic, IMA + FP group)

2. Motivating Example: SIR model, how can we simulate this?

- Population Size N
- Contact Rate **β**
- Infectivity **γ**
- Illness Duration **δ**

3. System Dynamics SD approach

- top-down
- differential equations
- draw SD dynamics

4. ABMS approach

- bottom-up
- more realistic: heterogenous agents, network- & spatial effects

5. what is an agent

- Uniquely addressable entity with internal state
- Living in an environment
- Pro-actively initiate actions

  Change internal state
  Send Messages
  Create new agents
  Kill themselves
  Interact with environment
- Reacting to messages with actions

6. Develop with the FP Group an ABS model of the SIR SD approach

- state
- message protocoll
- occasionally
- after

7. How do we implement this in Haskell in a general way?

- state of the art: oop
- agent & environment represenation? no classes / objects in haskell
- agent-agent & agent-environment interaction? no method calls and references

- updating of agents & environment? no mutable data and no side-effects

8. FRP Yampa & Actor Model

- yampa allows to make them pro-active through time-sampling
=> hybrid approach with continuous time-flow and discrete events
- Ultimate Goal: stay pure and never run within IO

9. What is an Agent then in our implementation

- SF AgentIn s m → AgentOut s m

10. update-strategies

- sequential & parallel, collapsing environment in parallel case
- conversations in case of sequential

11. looking into code

- Agent.hs
- FrSIRSNetworkAgent.hs

12. run Examples

If TIME: Show SD emulation