Accelerator Digital Twin Challenges

Tia Miceli

Human Language

- **Agreeing on definitions**
  a. Digital twin/shadow (shadow is maybe not useful anymore)
  b. Simulation
  c. Virtual accelerator
  d. Surrogate model
  e. Framework
  f. + others?

Human Nature

- **Trying new things is hard**
  a. Sim codes: everyone builds their own
     - Coding languages
     - Electron vs. hadron
  b. Control system universes

- **Funding agencies make difficult to plan**
  a. NP/HEP vs. more collaborative models like in EU
  b. Need to “stake a claim” at the end of a couple years

“AI” For Accelerators Means a Lot of Different Things

- Anomaly detection
- Anomaly detection → precursors
- Precursors → mitigation
- Optimization
- Drift → optimization again
- Regulation vs. optimization (control loop speed)
- Determining “detailed lattice”
- More?

“AI” For Accelerators Means a Lot of Different Things

- Anomaly detection
- Anomaly detection → precursors
- Precursors → mitigation
- Optimization
- Drift → optimization again
- Regulation vs. optimization (control loop speed)
- Determining “detailed lattice”
- More?

Digital Twin use cases

Fun Stuff

- **Uncertainty quantification**
    a. When is it needed on models?
    b. Which methods are most compatible given loop time?

- **Modeling fidelity**
    a. What is good enough?
    b. Multi-fidelity?
    c. Hierarchical components

- **DT use cases**
    a. (Previous page)

Conceptual challenges

(Besides human language)

What to model?

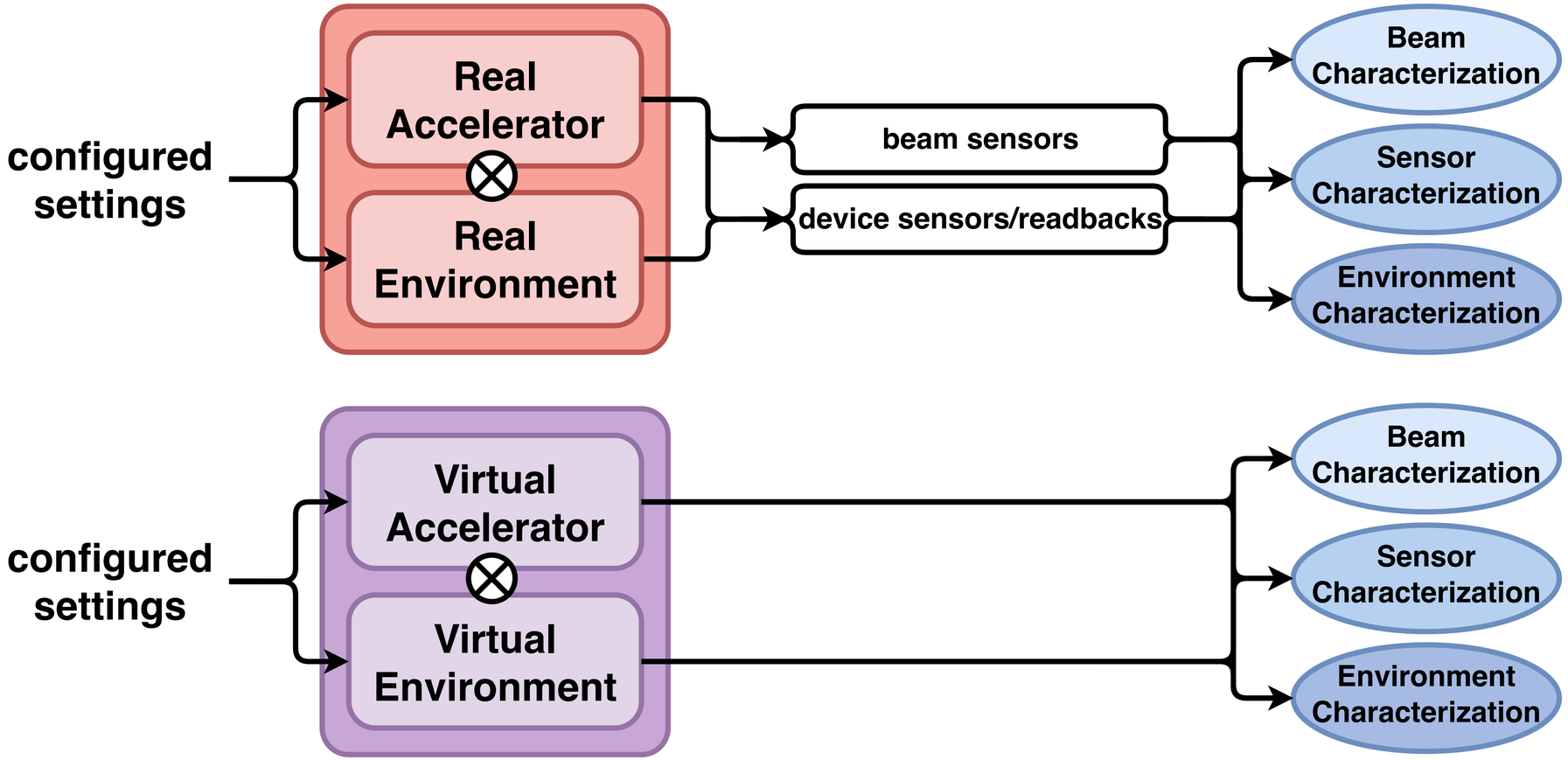
- Beam!
- …and stuff that affects beam
  - Cavity, magnets
  - Power supplies
    - Environment, temperature, humidity, nearby loads
  - “Breathing” in RF from power company
- Response of beam instruments
- 🐢🐢🐢…

Choosing where to match reality

- ADC/settings (current & voltage)
- Particle positions
- Beam characteristics
  - mean position, momentum, time/phase
  - Distribution, moments, more…

Conceptual challenges

(Besides human language)



Achievable DT

- Human language → we can define a glossary
- Acknowledge that DT is a conceptually higher thing than just it’s use cases
- Modern software design → can abstract pieces (optimizers, simulation, control system)