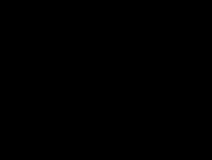


Building Adaptive Three-Dimensional Interfaces for Critical Complex Control Systems

# Perspectives on model-driven development



# Objectives

- How to **prototype** user interfaces
  - Multiple architectures and target platforms
- How to achieve **adaptable** user interfaces
  - That are still usefull for model checking
- How to automatically generate **correct** user interfaces from **fine-grained** UI modeling languages
- How to describe user interface guidelines to **validate** a UI model specification: “what we have is really what we want?”

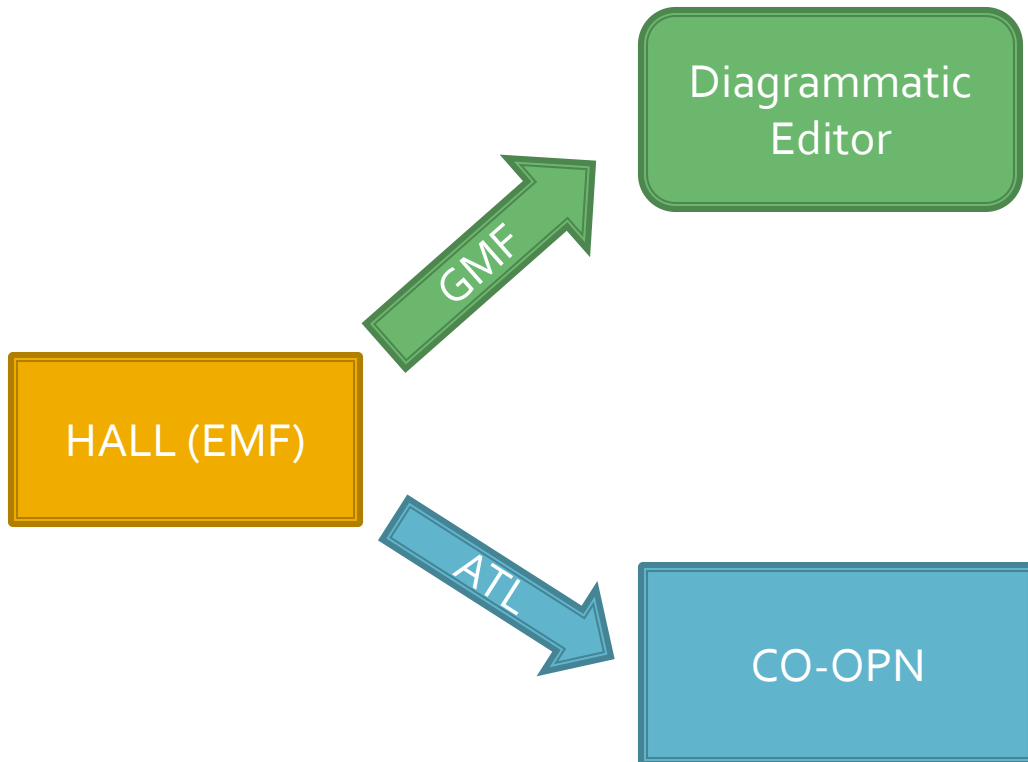
# Overview

HALL

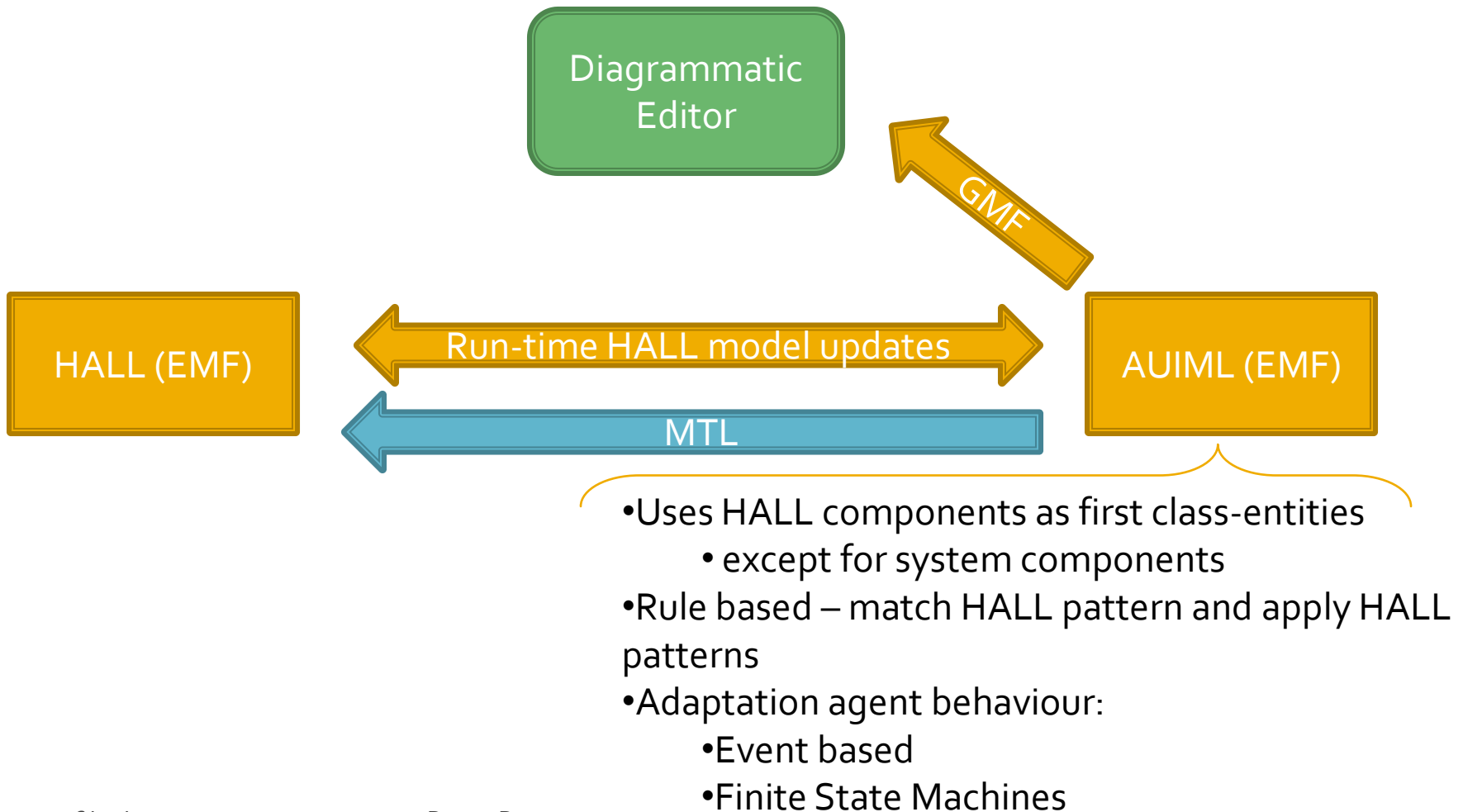
User-Interface modeling language

- 2D + 3D data structures
- Structure Description (hierarchical structures of components)
- Behaviour description
  - Event based
  - Finite state machines
  - WF Data Structures
- Describe user profiles
  - Describe visual components
  - Describe user task components
- Describe system components

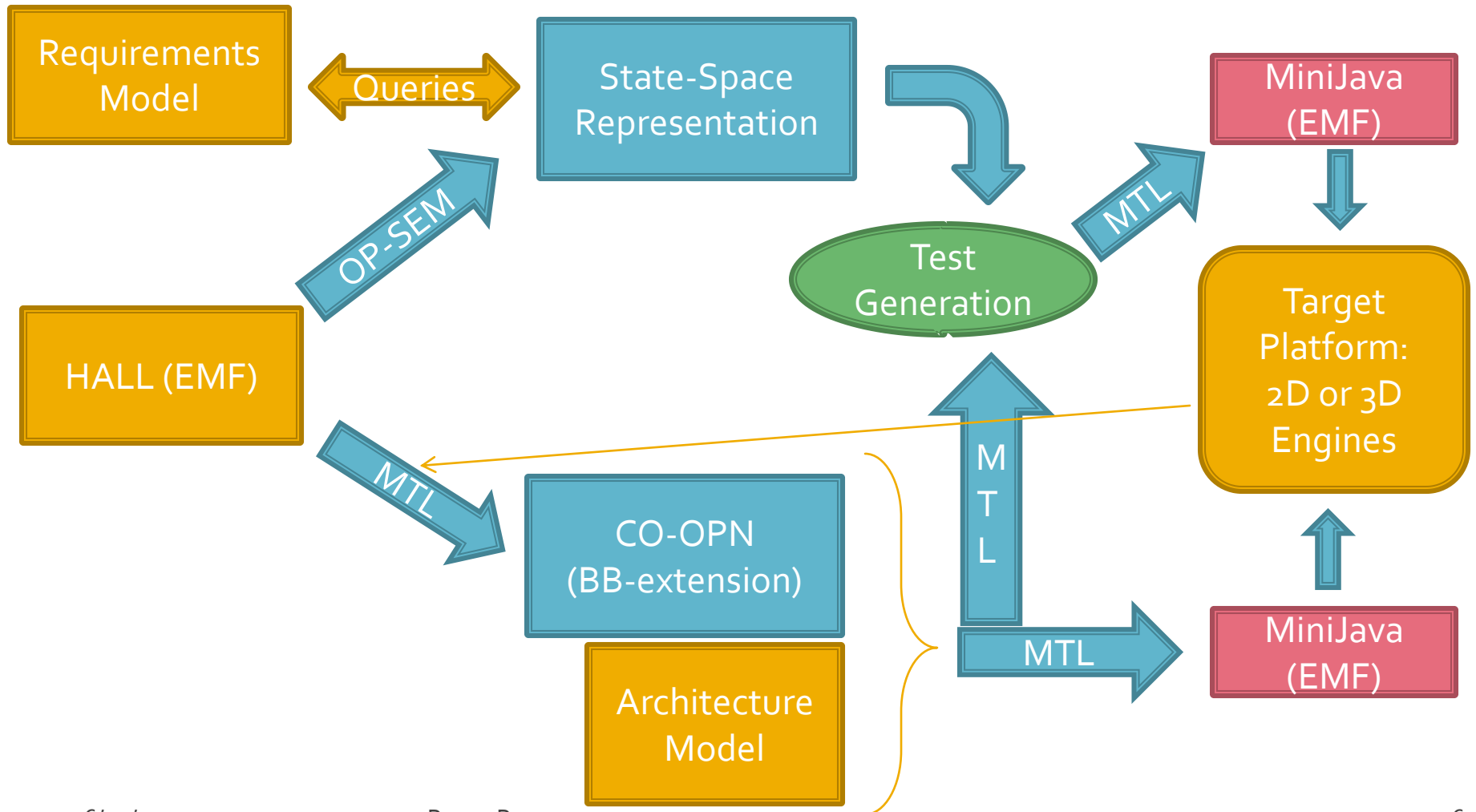
# Overview



# Overview



# Overview



# Still Open Questions

- What do you mean with MTL?
  - What makes a good model **translation** language?
  - Do we really need a DSL for model translations?
  - Correctness guarantees versus expressiveness
- Simulation/checking at runtime:
  - How state-space simulation in a “not-so-far” time-horizon can help us to **predict** and prevent faults in **critical** systems.