Using Ptolemy II as a Framework for Virtual Entity Integration & Orchestration in Digital Twins

MODDIT 2021 WORKSHOP AT MODELS 2021, OCTOBER 12TH, ONLINE OCTOBER 2021

David A. Manrique Negrin
Loek Cleophas
Mark van den Brand

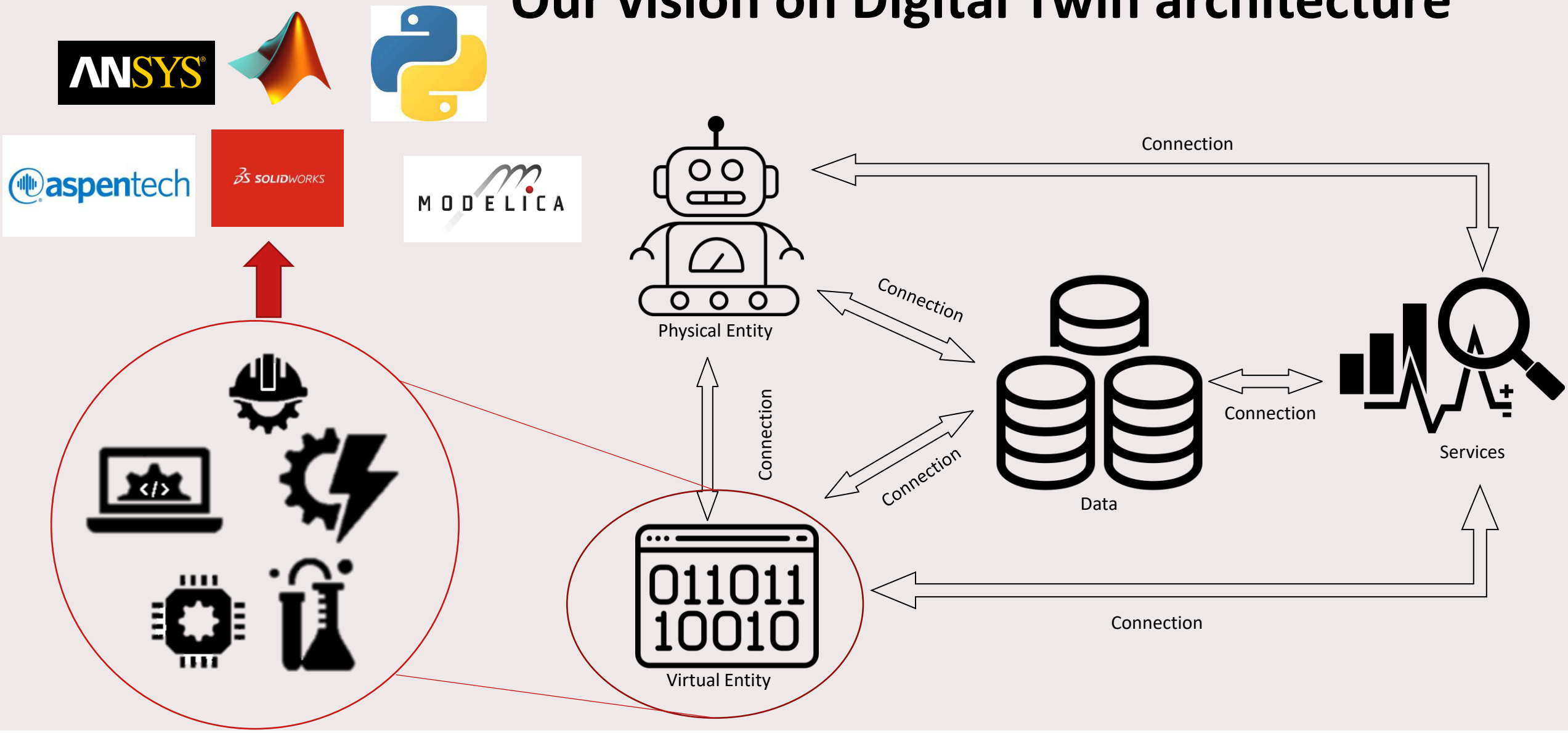
Dept. of Math and Computer Science — Software Engineering and Technology group

Digital Twin TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Agenda

- Problem definition
- Background
- Solution design – Ptolemy II implementation
- Results
- Conclusion & future work

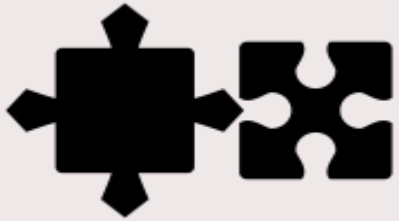
Our vision on Digital Twin architecture



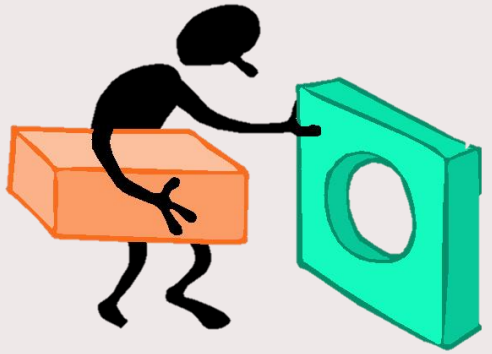
TU/e

Problem definition

- Different **tools** have **different semantics and syntax**.



- **Manual integration** becomes **unfeasible**, automation is needed.



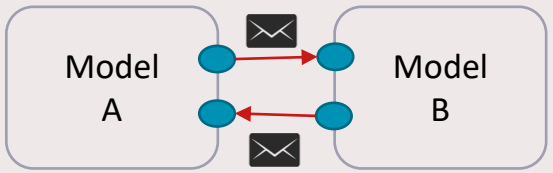
- A **framework** to **automate** the **integration** is required. Commercial options exist, but they only support their own tools for modeling.

Problem definition

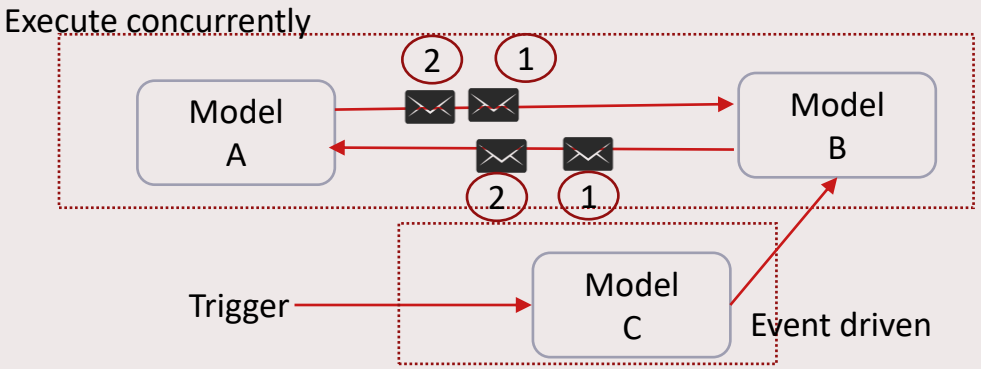
Our aim: Find a **framework** to **integrate heterogenous models** & **extend** it.

❖ **2 aspects** for integration:

- **Communication**: interface & encapsulation
- **Orchestration**: sequence of execution (control flow) and how data is exchanged (data flow)

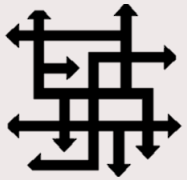


Communication

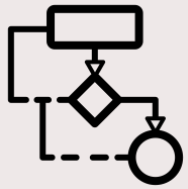


- Ptolemy II is a promising framework for integration.

Integration



Reduce Complexity



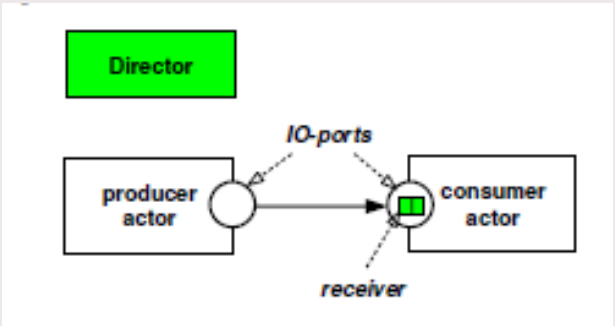
Reproduce behavior

Startup

TU/e

Background – Ptolemy II

- **Objective**: support analysis & experimentation with CPSes
- **Actor oriented**
- **Directors** are **orchestrators** of actors
- **Separated control flow and data flow**



Why Ptolemy II:

1. **Open-source**
2. Accessible and sufficient **documentation** & **support**
3. Artefacts are **modular & re-usable**

Background – AES-Lab [1]

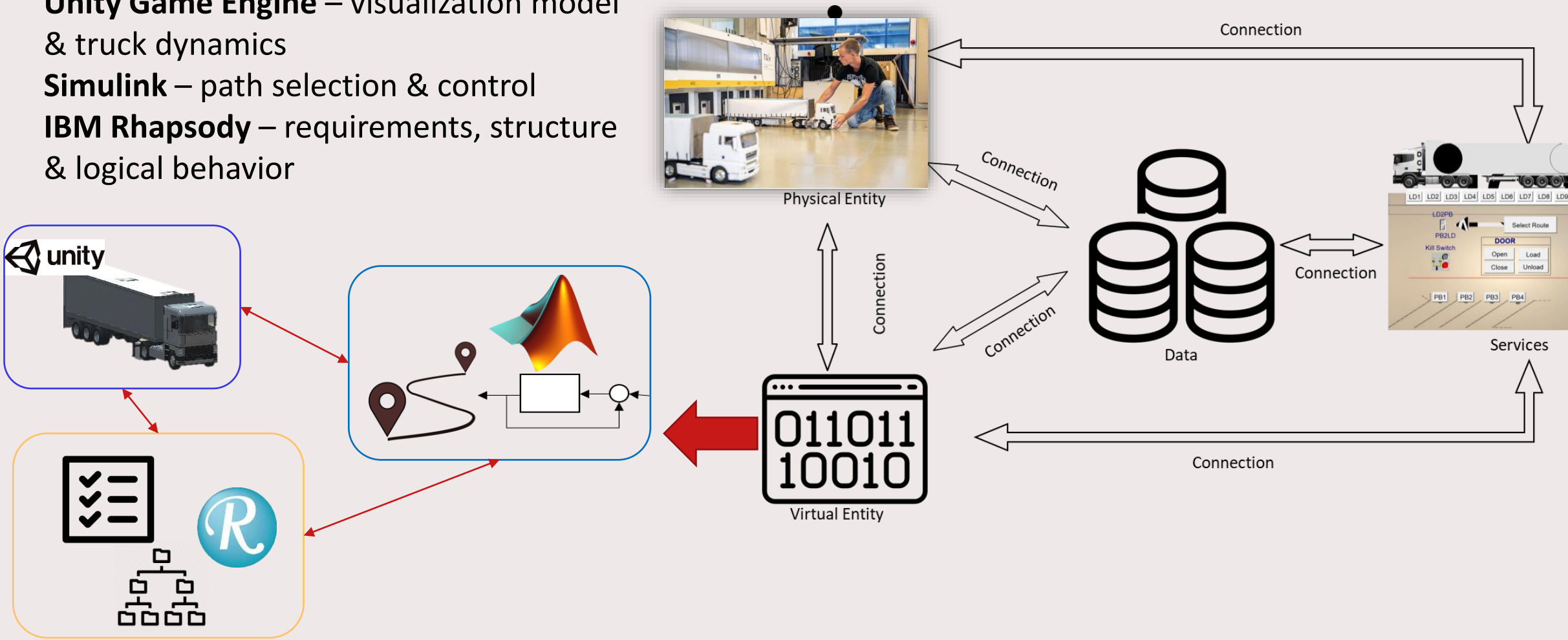
- **Scaled-down truck** for driving and docking in a DT.
- Used for **testing of autonomous driving**: path generation, avoidance of obstacles & driving control.

[1] Barosan, I., Basmenj, A. A., Chouhan, S. G., & Manrique, D. (2020, September). Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center. In *European Conference on Software Architecture* (pp. 542-557). Springer, Cham.

AES-Lab Digital Twin

Models:

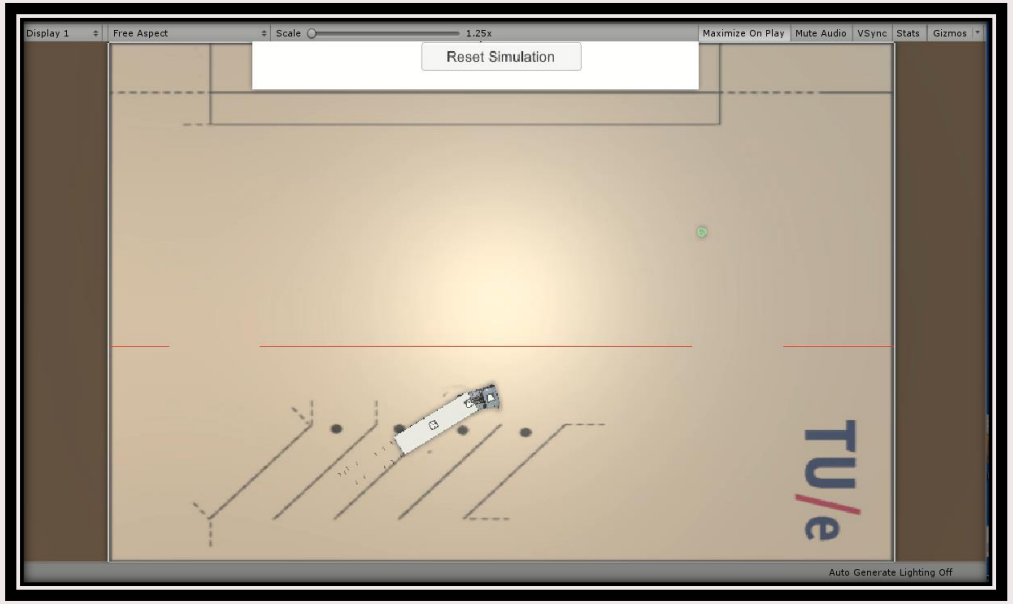
1. **Unity Game Engine** – visualization model & truck dynamics
2. **Simulink** – path selection & control
3. **IBM Rhapsody** – requirements, structure & logical behavior



Final behavior results

Use case for validation:

1. Truck at **initial position**.
2. Signal a **docking station** as **destination** for truck
3. Truck **generates a path.**
4. Truck **follows path**, **avoiding obstacles** & with **similar maneuvers** (speed & steering) as in [1].



[1] Barosan, I., Basmenj, A. A., Chouhan, S. G., & Manrique, D. (2020, September). Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center. In *European Conference on Software Architecture* (pp. 542-557). Springer, Cham.

Original implementation vs our objective

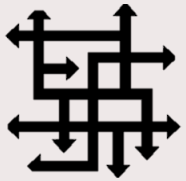
Implementation in [1]:

- Starting the execution **requires manual setup** & startup.
- **Complex Integration** between models.

Our implementation aims:

- **Decrease complexity** of integration, set-up & startup.
- **Reproduce** original **behavior.**

Integration



Reduce Complexity

Reproduce behavior

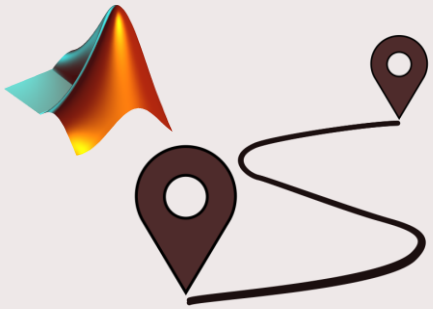
Startup

[1] Barosan, I., Basmenj, A. A., Chouhan, S. G., & Manrique, D. (2020, September). Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center. In *European Conference on Software Architecture* (pp. 542-557). Springer, Cham.

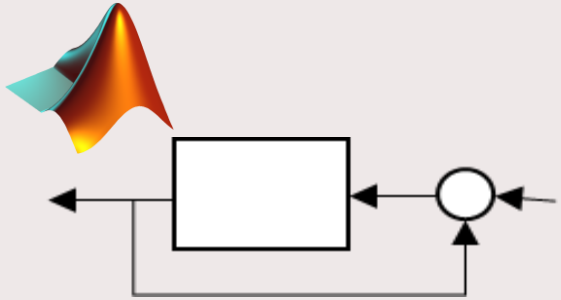
First steps for implementing of DT in Ptolemy II

Simulink models were split:

- To test director execution & model integration.



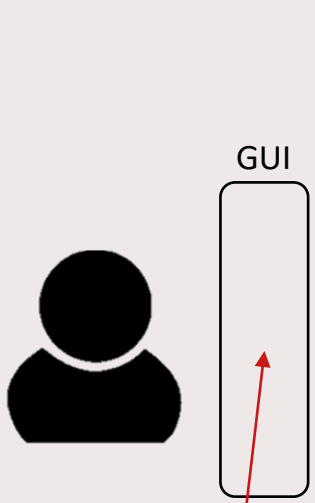
Path planner

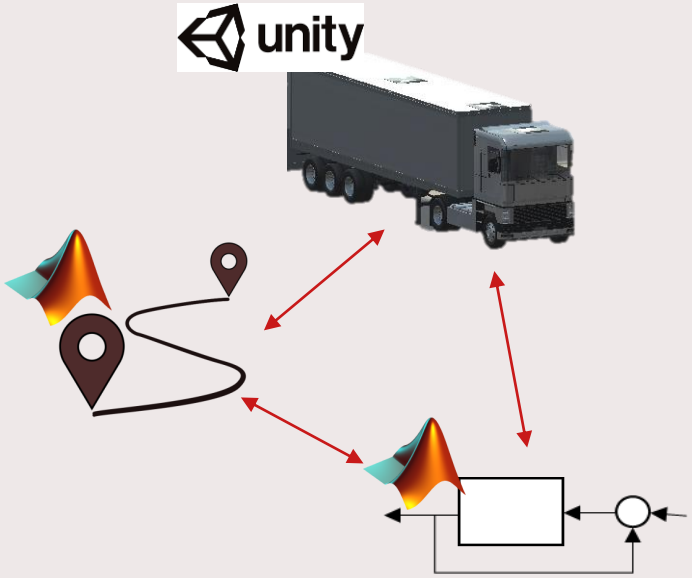


Control

- In a real application the separation would aid maintainability.

Integration of models:





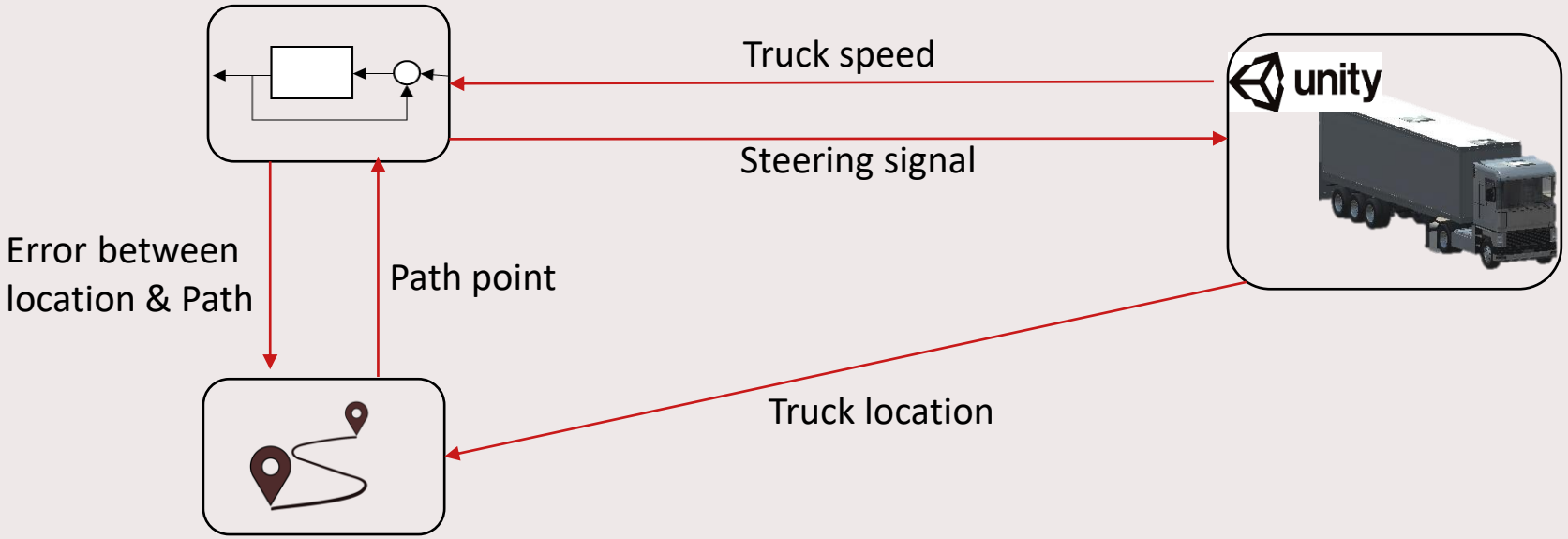
- Choose final destination
- Start truck movement

Ptolemy II Implementation - Actor

REQUIREMENTS:		Our Solution:
R1	Execution of all the models	Selected **Simulink actor** for Simulink models & **Exec actor** for Unity models.
R2	Simulink simulation state visible	New **Simulink actor**: • **Reduction of complexity** to call Simulink models. • **Enable visibility** on simulation states.
R5	Reduction of complexity in integration & setup	
R3	Model's communication on each time-step	**S-function** generation for Simulink models & **communication** over TCP-IP using **Python actor.**
R4	Enable configuration of Simulink solver (time-step)	Simulink actor **enables configuration** of Simulink **time-step.**

Ptolemy II Implementation - Orchestration

All models required **concurrent exchange & execution** of data

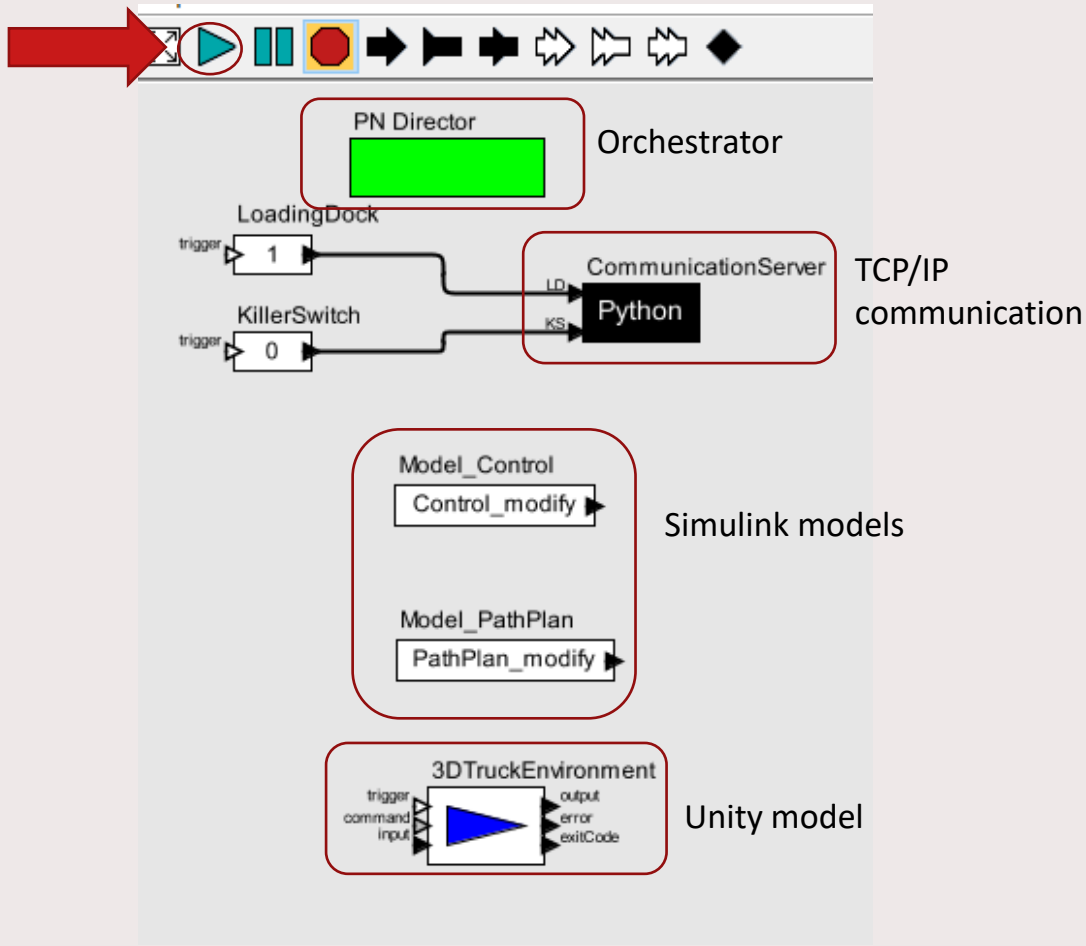


Our implementation → **PN director** for orchestration
Communication → **TCP/IP protocol** with python actor

Results - Analysis

Results: successful execution

- Decreased complexity of integration:
  - **Simulink** actors execute models with **only 2 parameters**
  - **Unity models** executed with **1 parameter**
- Reproduced original behavior:
  - Using **TCP/IP communication**
  - Orchestration of all models by **PN director**
- Setup and startup **complexity reduced**:
  - **One click** to start execution



1] Barosan, I., Basmenj, A. A., Chouhan, S. G., & Manrique, D. (2020, September). **Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center**. In *European Conference on Software Architecture* (pp. 542-557). Springer, Cham.

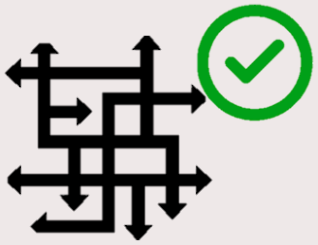
Conclusions

- Ptolemy II **facilitates models' integration**.
- **Actors reduce complexity** of such integration & aid **scalability**.
- **Director** can **orchestrate** control flow to **reproduce DT behavior.**
- **Simplified startup**.

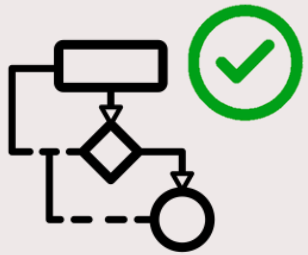
Future work

- **Implement data flow** control for orchestrator - Implement the data exchange within the **Simulink actor**.
- **Simplify further** the **integration** for Simulink models **- Automate** the generation of **S-function** for Simulink models.
- Experiment **on integration technologies** for other type of models, e.g., FMI.

Facilitate Integration



Reduce Complexity



Reproduce behavior

Simplified Startup

Thank you.