

**SIMUtools 2023**

Seville, Spain

# Generative AI with Modeling and Simulation of Activity and Flow-based Diagrams

**Abdurrahman Alshareef,**

Assistant Professor at *King Saud University* and Consultant at *RTSync*

**Nicholas Keller, Priscilla Carbo, and Bernard P. Zeigler**



جامعة  
الملك سعود  
King Saud University



# Generative AI with Modeling and Simulation of Activity and Flow-based Diagrams

Abdurrahman Alshareef<sup>1</sup>, Nicholas Keller<sup>2</sup>, Priscilla Carbo<sup>2</sup>, and Bernard P. Zeigler<sup>2</sup>

<sup>1</sup> Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia  
ashareef@ksu.edu.sa

<sup>2</sup> RTSync Corp., Chandler AZ 85226, USA  
{nicholas.keller,priscilla.carbo,zeigler}@rtsync.com

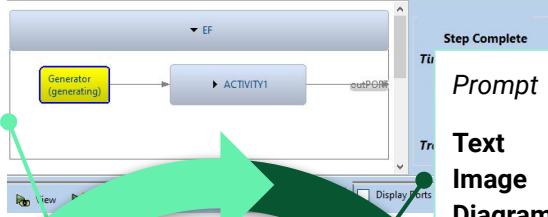
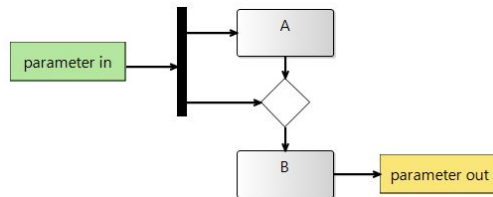
**Abstract.** In systems engineering and model-based design, the complexity and interrelationships across different system elements always demand continuous elaboration and expansion in various overlapping domains. We examine how such a phenomenon can be assisted with generative AI and benefit from large language models (LLMs), such as GPT-4. We demonstrate ways of incorporating generated text and outputs from LLMs into the modeling process. The approach can customize the GPT-4 model with an activity metamodel specified in Eclipse Ecore or predefined activity diagrams encoded in a textual format for learning from instances. Alternatively, the descriptive text from the LLM can be provided as input to a parser, resulting in an activity that can be readily transformed into a discrete event system specification (DEVS) model with simulation capability. We will discuss how the process can be enhanced in a simulation environment, thus offering the opportunity to examine a variety of scenarios and arguments for incorporating generative AI or general AI as a collaborative agent in the domain of interest. One scenario could begin with a simplified text describing a generic process, yielding an approximate representation as a starting point for further elaboration by modelers to a complex specification through a systematic, guided, and well-defined framework. We demonstrate the approach with activity and flow-based diagrams in a manner applicable to SysML, UML, and systems engineering at large.

- Suitable for systems engineering and model-based design
- Examining the level of assistance needed to enhance the modeling and simulation in such systems
- Experiment with GPT
- Explore different ways
- Finding the right formula/mix/combination
- Training, fine tuning, reinforcement learning
- Transformation and translation
- DEVS formalizations
- DEVS Markov simulation
- Simulation and results
- Lesson learned

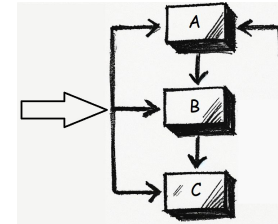
**Simulate**  
 In a discrete-event environment within a well-defined experimental frame and support for parallel, multithreaded, real-time, deterministic, and stochastic specification

$(p_i, v_i) \in \mathcal{O} \cup \mathcal{I} \cup \mathcal{C}$ ,  $v_i \in X^b$ ,  
 $OPorts = \{out_1, \dots, out_m\}$ ,  $Y_p = V$  (an arbitrary set)  
 is the set of output port and value pairs  
 $S = phase \times \sigma \times task \times C$ , where  
 $phase = \{passive, waiting, active\}$ ,  
 $\sigma = \mathbb{R}_{0, \infty}^+$ ,  $task \subseteq X^b$ ,  
 $C = \{(p, c) : p \in IPorts, c \in \{true, false\}\}$   
 is the set of input ports and conditions  
 $\delta_{ext}((phase, \sigma, C, task), e, X^b) =$   
 $(waiting, \infty, (p_i, true), (p_i, v_i))$   
 if  $p_i = in_i \wedge \exists (p_j, c) \wedge c = false \wedge i \neq j$   
 $(active, 0, C, (p_i, v_i))$   
 if  $p_i = in_i \wedge \forall (p_j, c) \wedge i \neq j \implies c = true$   
 $\delta_{int}(phase, \sigma, C, task) =$   
 $(passive, \infty, (p_i, false), x) \in X^b$   
 if the queue is empty  
 $(passive, \infty, (p_i, true), x) \in X^b$   
 if the queue is not empty

**Transform**  
 Activities and flow-based diagrams into fully-fledged DEVS models



**Prompt**  
 Text Code Sketch  
 Image Audio Note  
 Diagram



**Extract**  
 Requirements Logic  
 Documentation Steps  
 Equations

```

{
  "name": "A",
  "class": "Action",
  "hasQueue": false,
  "distribution": {
    "type": "exponential",
    "mean": "10.0"
  }
}
  
```

**Refine**  
 Generative outcomes and models into an interpretable results with rules and correctness checks

**Simulation with Generative AI**

# GPT for simulation modeling and optimization

It is proving to be quite challenging to get GPT models to perform well with mathematical models and reasoning

And if they do at all, it's at far more computationally demanding processes

So, can DEVS, as a mathematical formalism, coupled with activity modeling, help with that?

Either with custom GPT, refining the inputs and outputs, or with reinforcement learning, let see how it works.

# Things to consider..

Cost of GPUs and training

Simulation demands substantial computational power

The new economic of tokens

Prompt engineering

Time complexity

The trip back and forth

How to utilize the dedicated compute services to deliver the best results that can push computational boundaries and perform the desired tasks while delivering quality

## Possible future questions about:

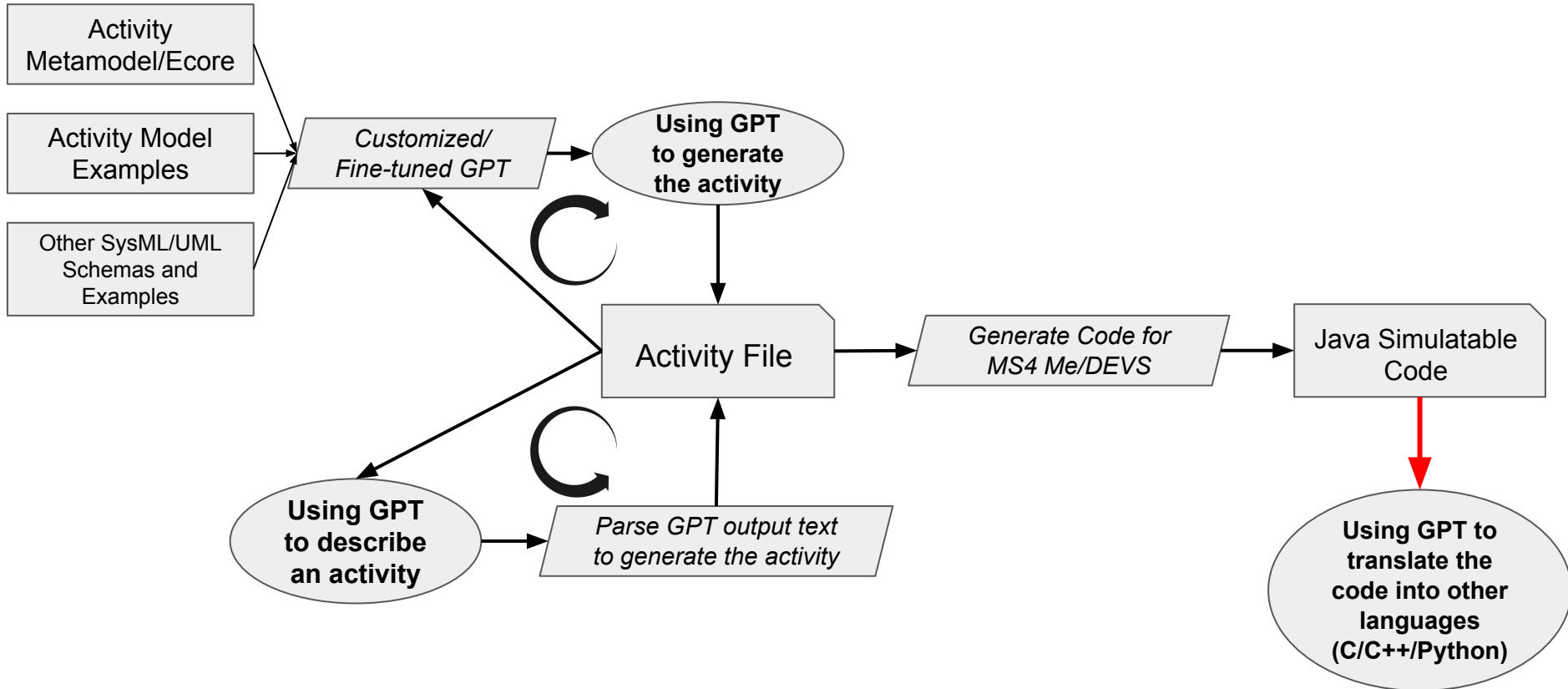
Delivering the desired outcomes and complying with requirement and regulation  
Detect safety issue and potential harms (think, hiring and education)  
Conducting risk assessment  
Train, pre-train, transfer learn, and fine tune  
Involving human oversight  
Deployment and production  
Responsible use for security and quality

**“balance the fostering of innovation with the need to safeguard against possible harm”**

*Excerpt from The New York Times on Dec. 8, 2023*

Let see how it would  
look like

# Prompt engineering, customization, and scenario/steps generation



# Simple start..

Sure! Here's a simplified workflow for processing online orders:

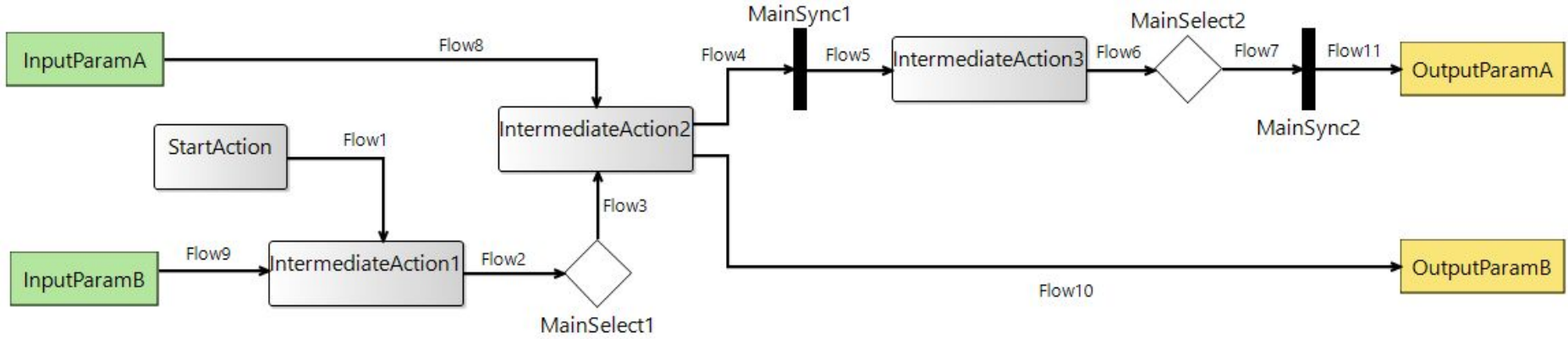
1. Receive order details.
2. Verify payment information.
3. Check product availability.
4. If product is available, proceed to step 5. If not, inform customer and cancel order.
5. Prepare order for shipment.
6. Pack items securely.
7. Generate shipping label.
8. Hand over package to shipping carrier.
9. Provide customer with tracking information.
10. Order is shipped.
11. Customer receives order.
12. Follow up for feedback or support.

Hope this helps! Let me know if you have any other questions.

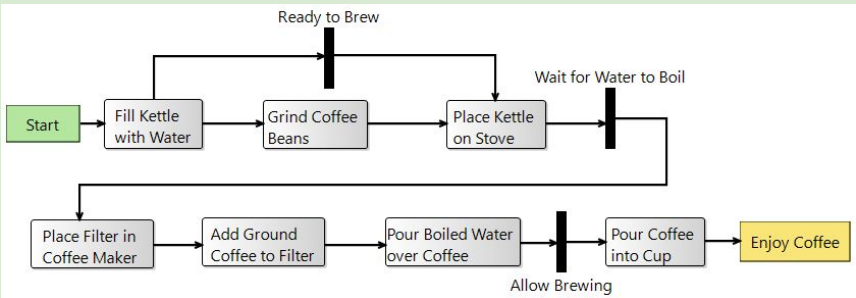
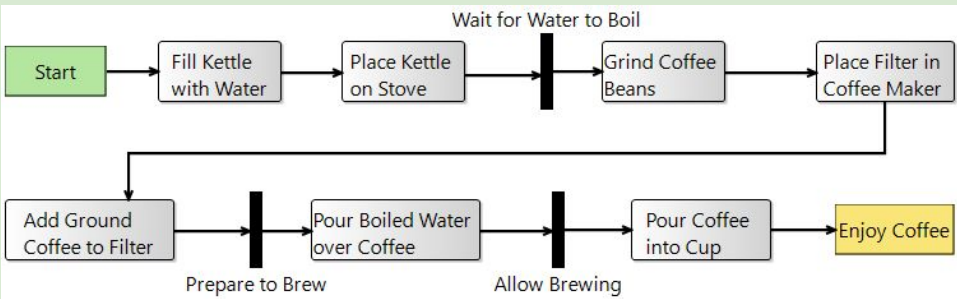
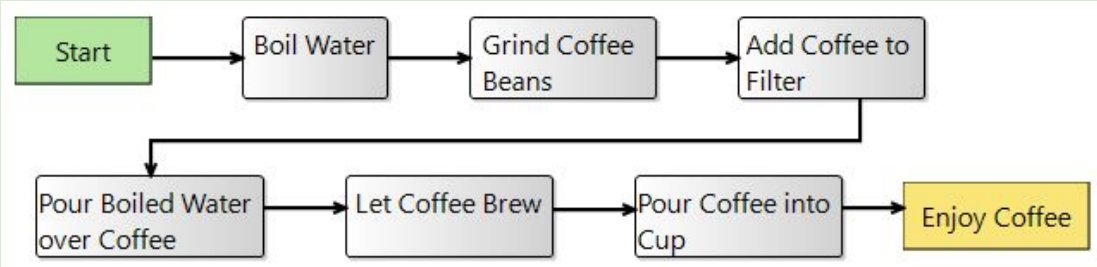
*Other examples: coffee making, cooking, starting business, or launching a rocket into space*



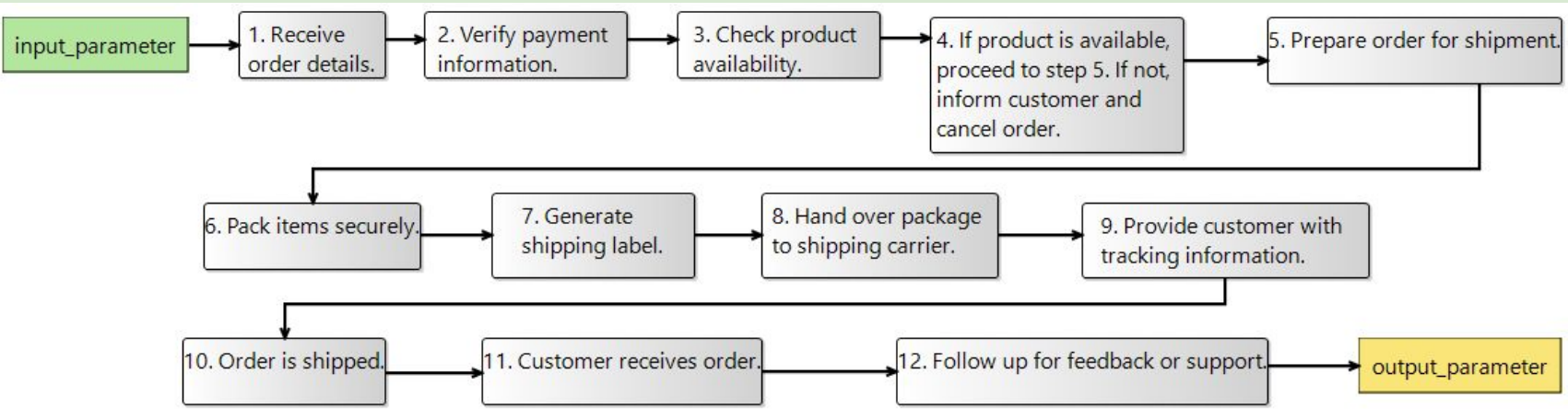
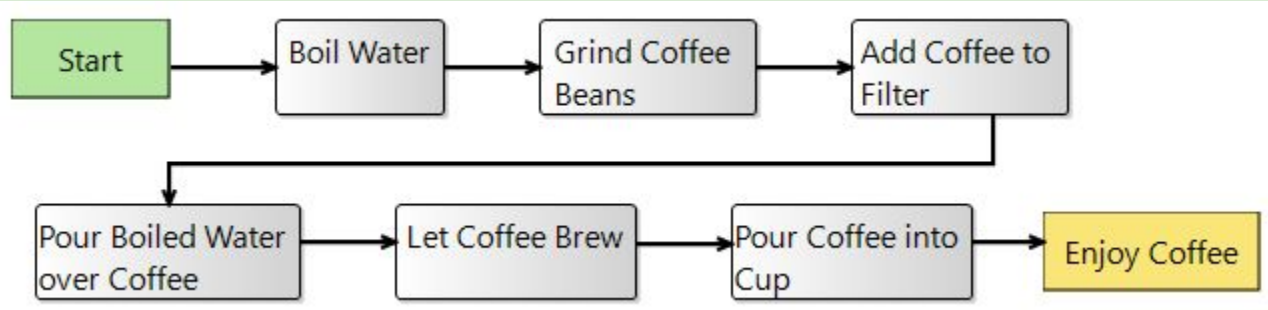
# Some results from the first steps (1)



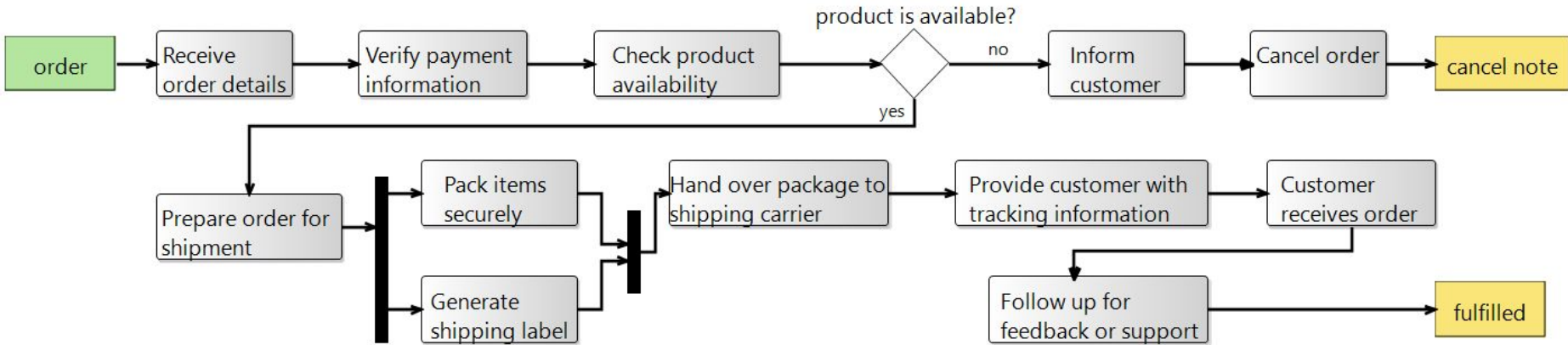
# Some results from the first steps (2)



# Some results from the first steps (3)



# Tool augmentation with full support (enhancement and full specification)



**Flow**

**Main**

**Semantic**

**Style**

**Appearance**

**Properties**

Label:

Source:  Action Navigate to Accepted Papers Section:

Target:  Action

Propability:

Distribution Type:  None  Uniform  Normal  Exponential  LogNormal  Pareto

Constant:

Lower:

Upper:

Mean:

Sigma:

Alpha:

**Due Select**

**Main**

**Semantic**

**Style**

**Appearance**

**Properties**

Name:

activity : Activity:

Early Flow Label:

Ontime Flow Label:

Late Flow Label:

Due Notify Flow Label:

Expiry Notify Flow Label:

Min T:

Max T:

**Palette**

- Action
- Sync
- Select
- DueSelect
- Input Parameter
- Output Parameter
- Activity
- Flow

Hierarchical Activities

**And simulate**

## *Current*

**Simulation is already used for testing and benchmark**

## *Future*

**Simulation to explore knowledge/safety gaps in systems with disciplined approaches from M&S**

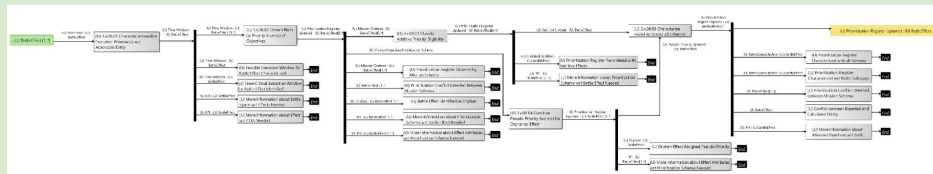
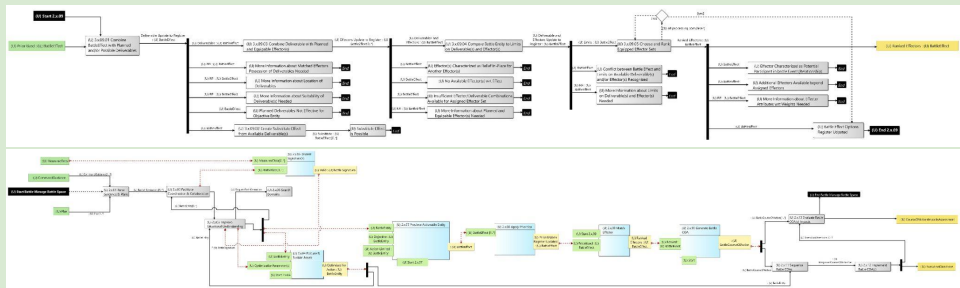
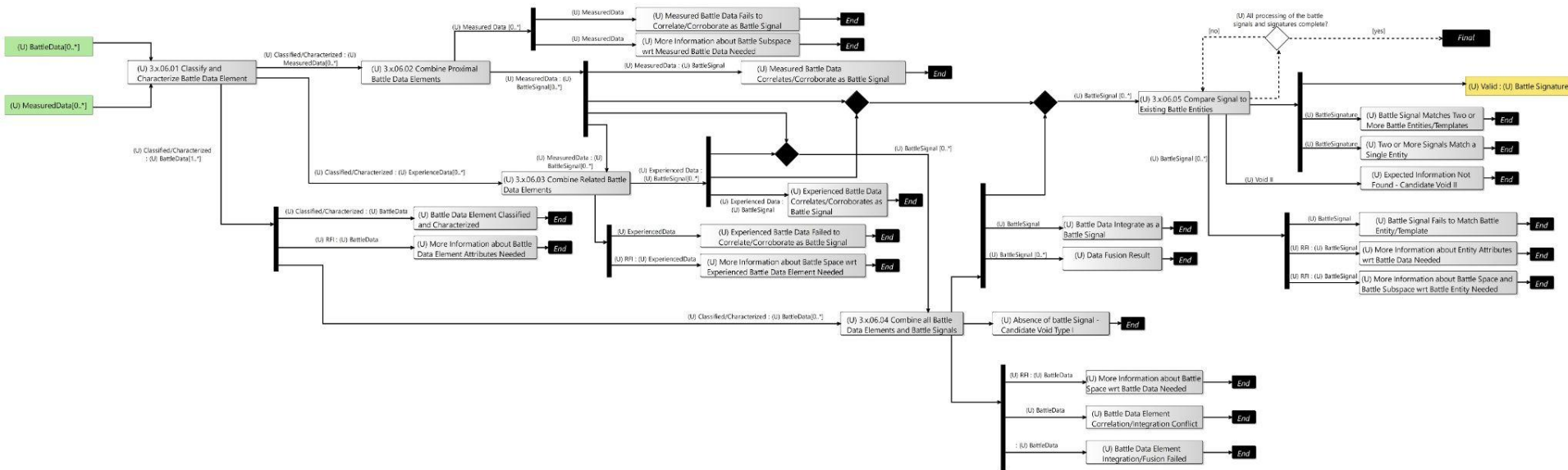
# Demo with DEVS Markov Activity models

*Current, **future!***

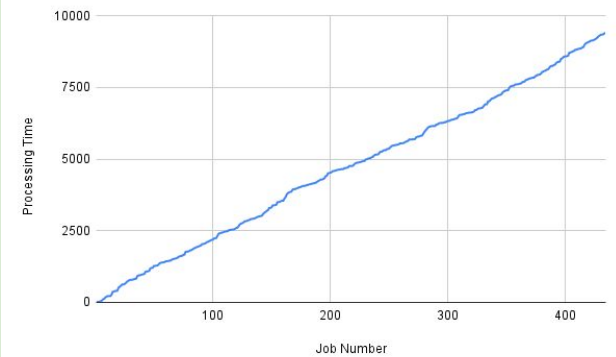
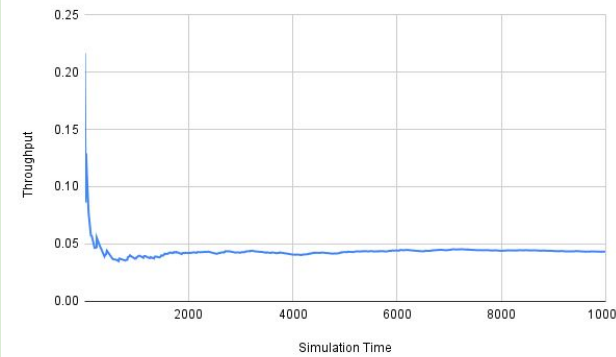
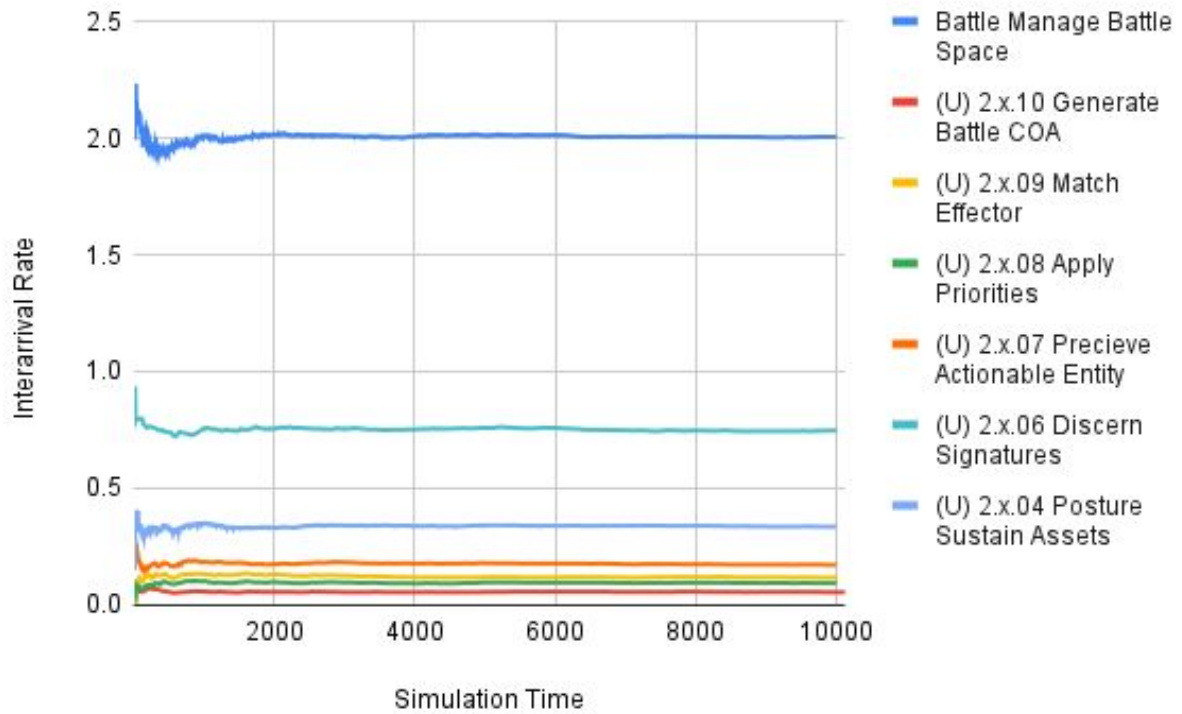
**Activity DEVS Markov modeling,  
Accounting for parallel flows in  
generated activities,  
Combinatorial design,  
And large activities**



# Glimpse at large activities



# Possible interpretations of the results



# Potentially

Towards a full-fledged system  
with services and applications

*Through*

Real-time Modeling

Digital engineering

Interactive software design

Data interchange for modeling,  
monitoring, and optimized  
execution

---

# Lesson learned

AI assistance and proper employment and use will be a key factor in accelerating system and software development but its is also necessary to ensure correctness and quality

*Hence*

Optimization

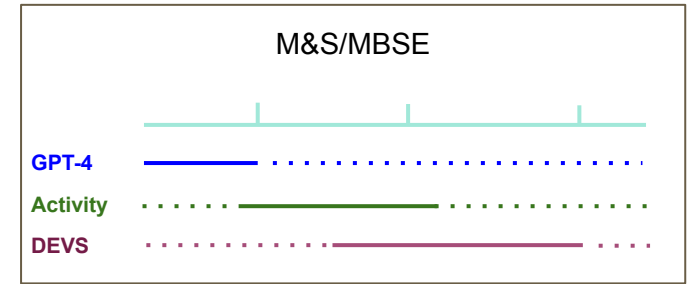
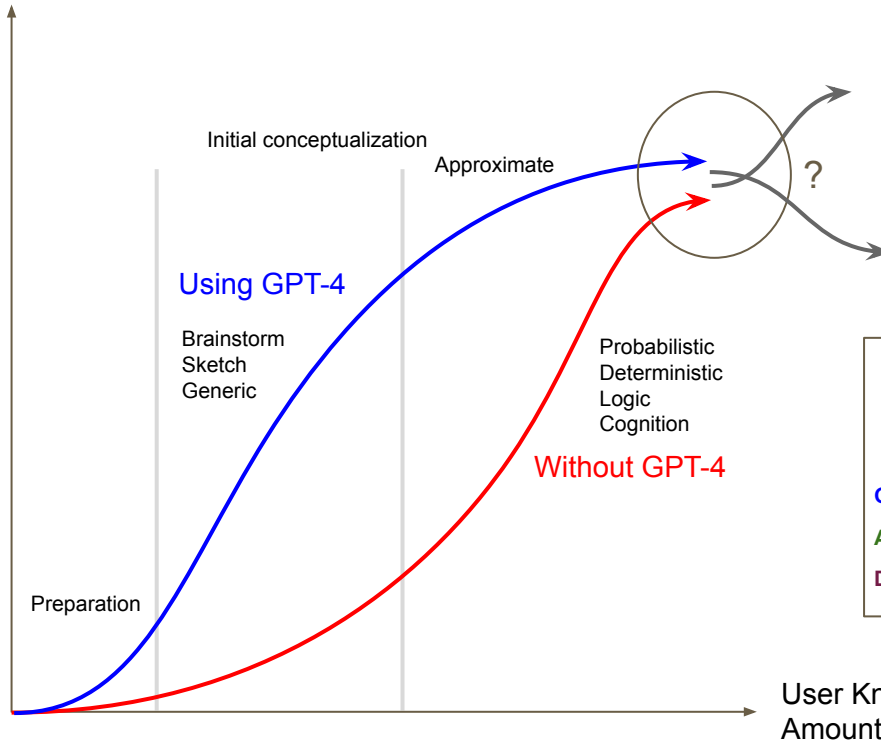
Computational performance

Parallelization

High Performance Computing

---

Achieved Complexity and  
Size of System/Model



**Thank You!**

**Q&A**

---