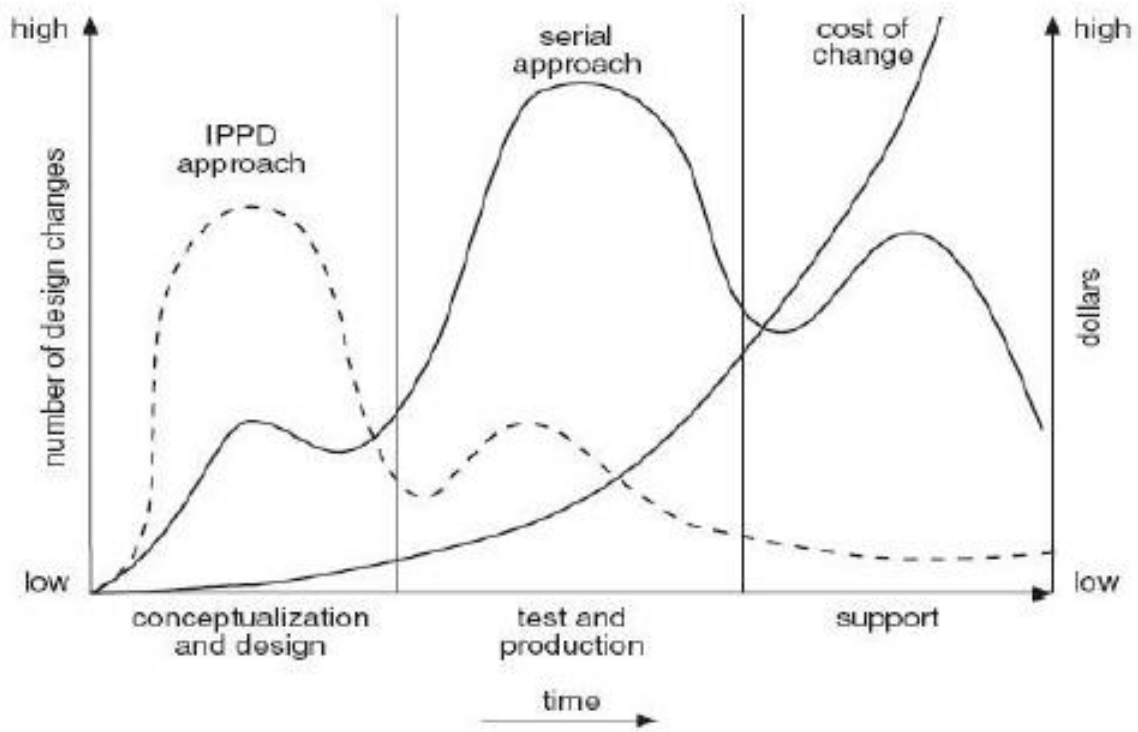
Evaluating the Production Consequences of Design Decisions using MATLAB and Simulink

Prof. Leon McGinnis (Georgia Tech ISyE)

Dr. Tim Sprock (now at NIST)

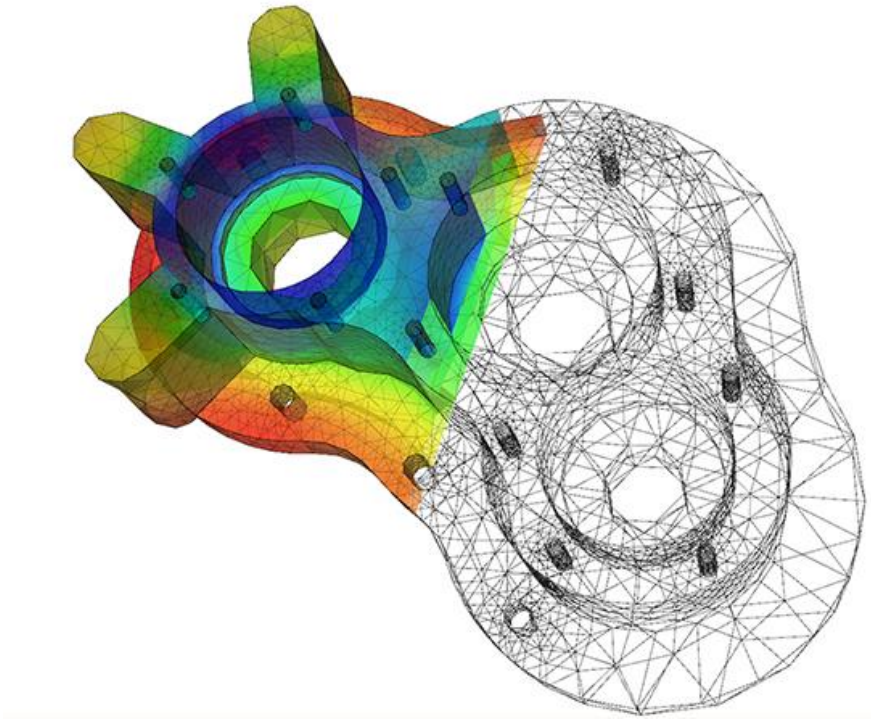
Dr. George Thiers (now at ModGeno)

Integrated Product-Process Development



A Fundamental Obstacle to IPPD

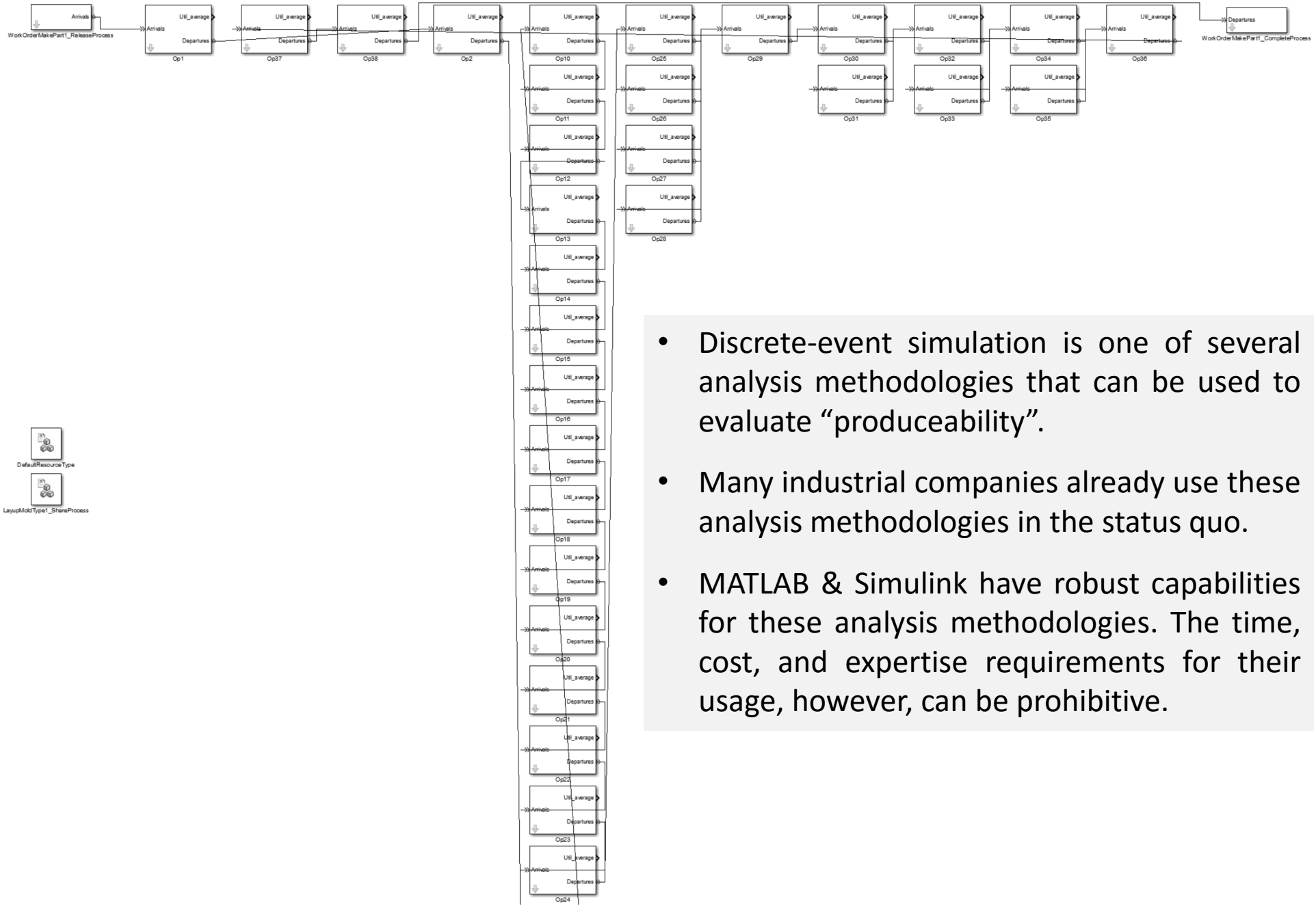
Tool support for the design and operation of industrial engineering systems (manufacturing systems, supply chains, sustainment systems, warehouses, distribution centers, …) **is far less sophisticated than for products themselves.**



An analogy: When designing a part in a CAD environment, finite-element analysis is push-button accessible – a mesh and the mathematical analysis model can be automatically generated.

For operations research analysis of industrial engineering systems (discrete-event simulation, statistics in support, optimization on top), we effectively create the mesh and write the analysis code by hand, each and every time, even to answer routine and well-understood questions which we have seen before and know how to answer.

A Fundamental Obstacle to IPPD

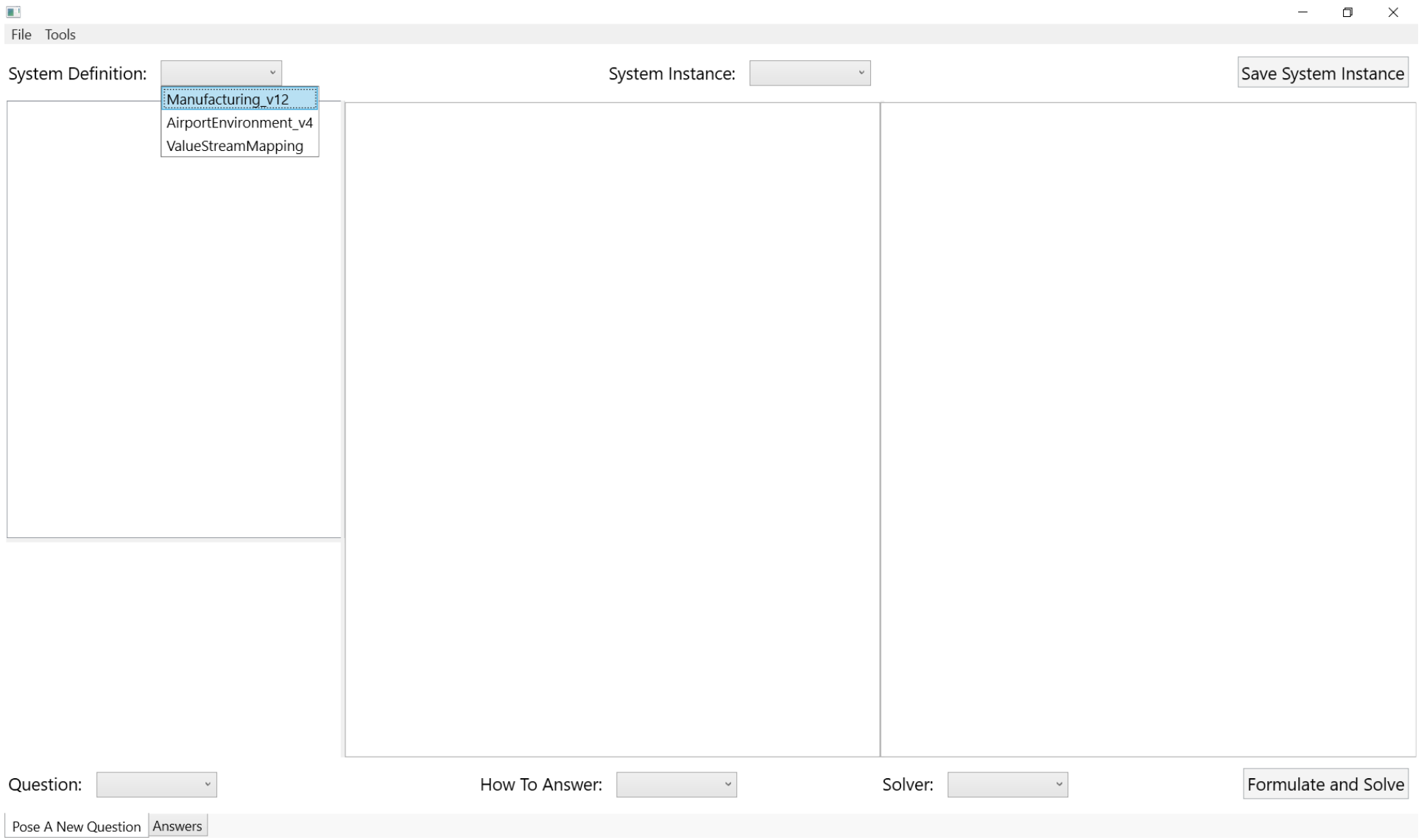


- Discrete-event simulation is one of several analysis methodologies that can be used to evaluate “produceability”.
- Many industrial companies already use these analysis methodologies in the status quo.
- MATLAB & Simulink have robust capabilities for these analysis methodologies. The time, cost, and expertise requirements for their usage, however, can be prohibitive.

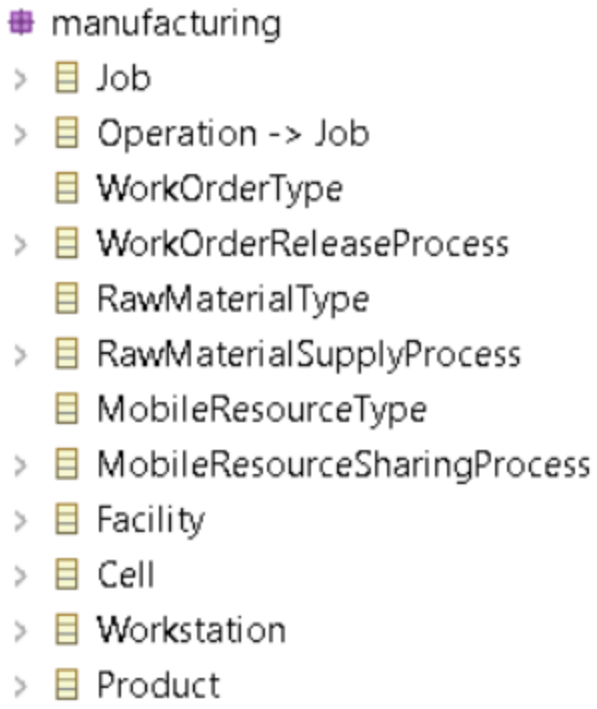
What Would Better Tool Support Enable?

- Predicting the behavior and performance of manufacturing process and facility designs, quickly and at very low cost.
- Receiving fast and frequent feedback about the production consequences of design decisions.
- Extending Value Stream Maps into variability exploration tools, such that standard hours, inventory buffers, and supplier delivery schedules can be chosen for robustness.
- Considering more improvement ideas and alternatives, evaluating more production scenarios and their impacts, and exploring more of a production system’s design space.

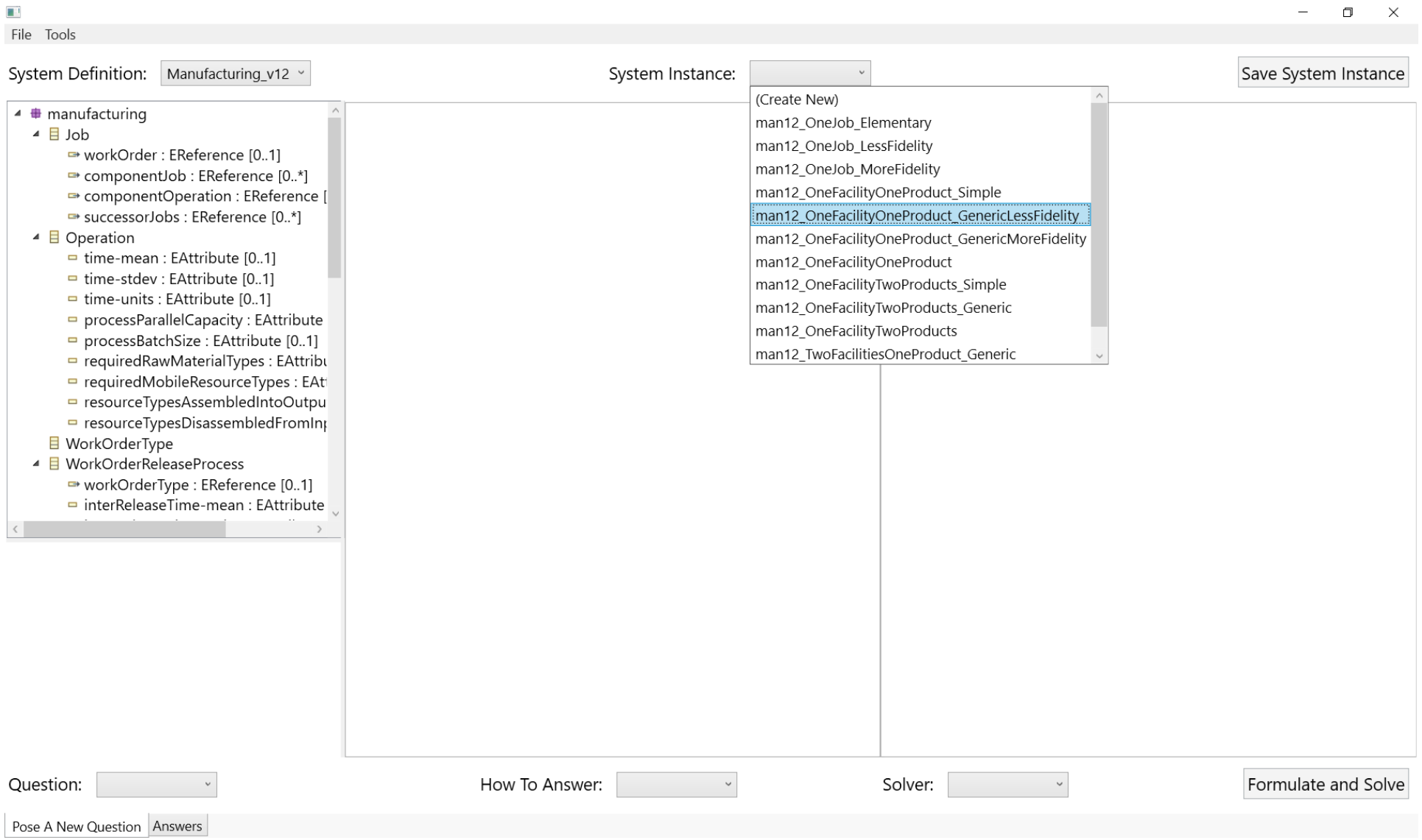
Results: Manufacturing Demo



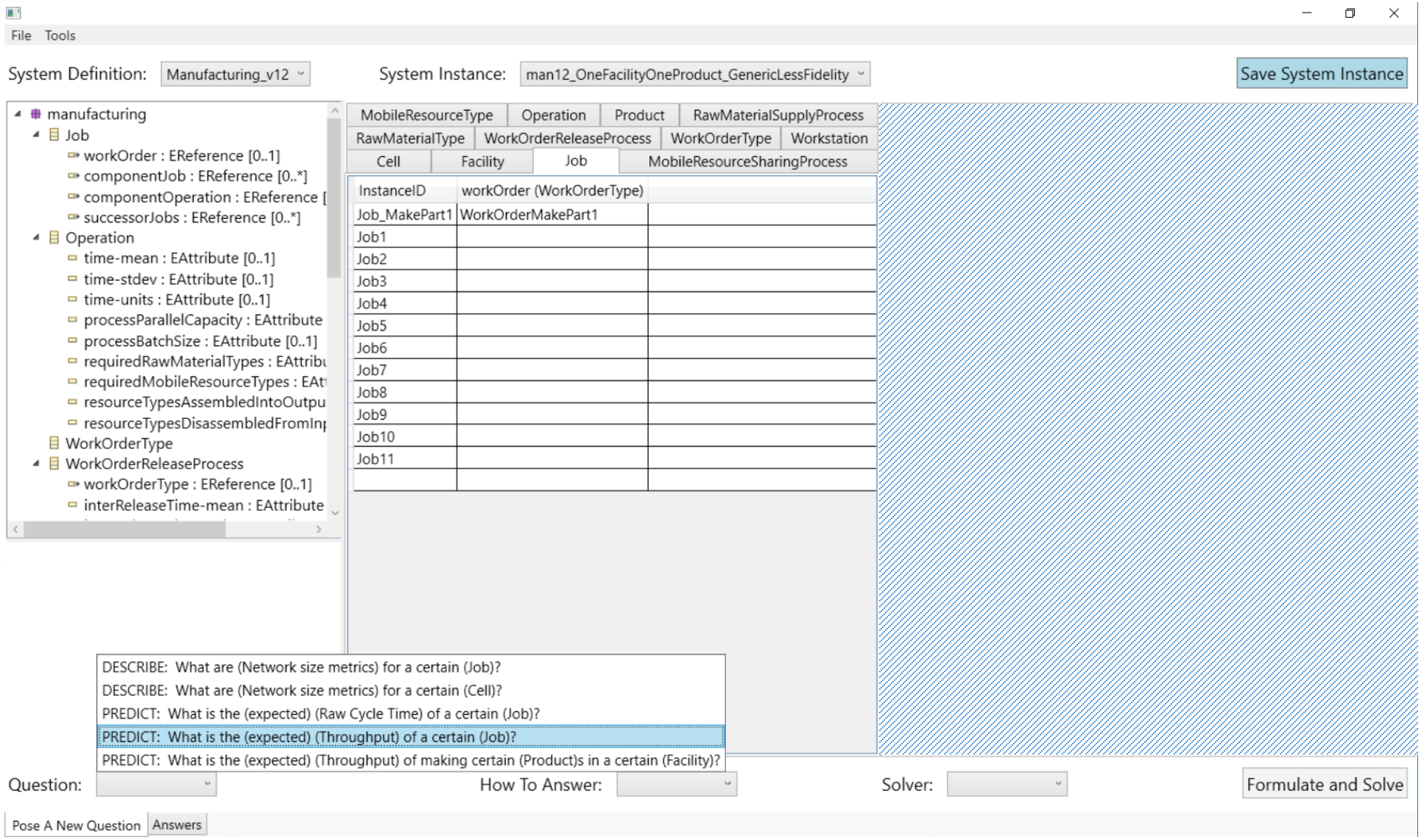
Results: Manufacturing Demo



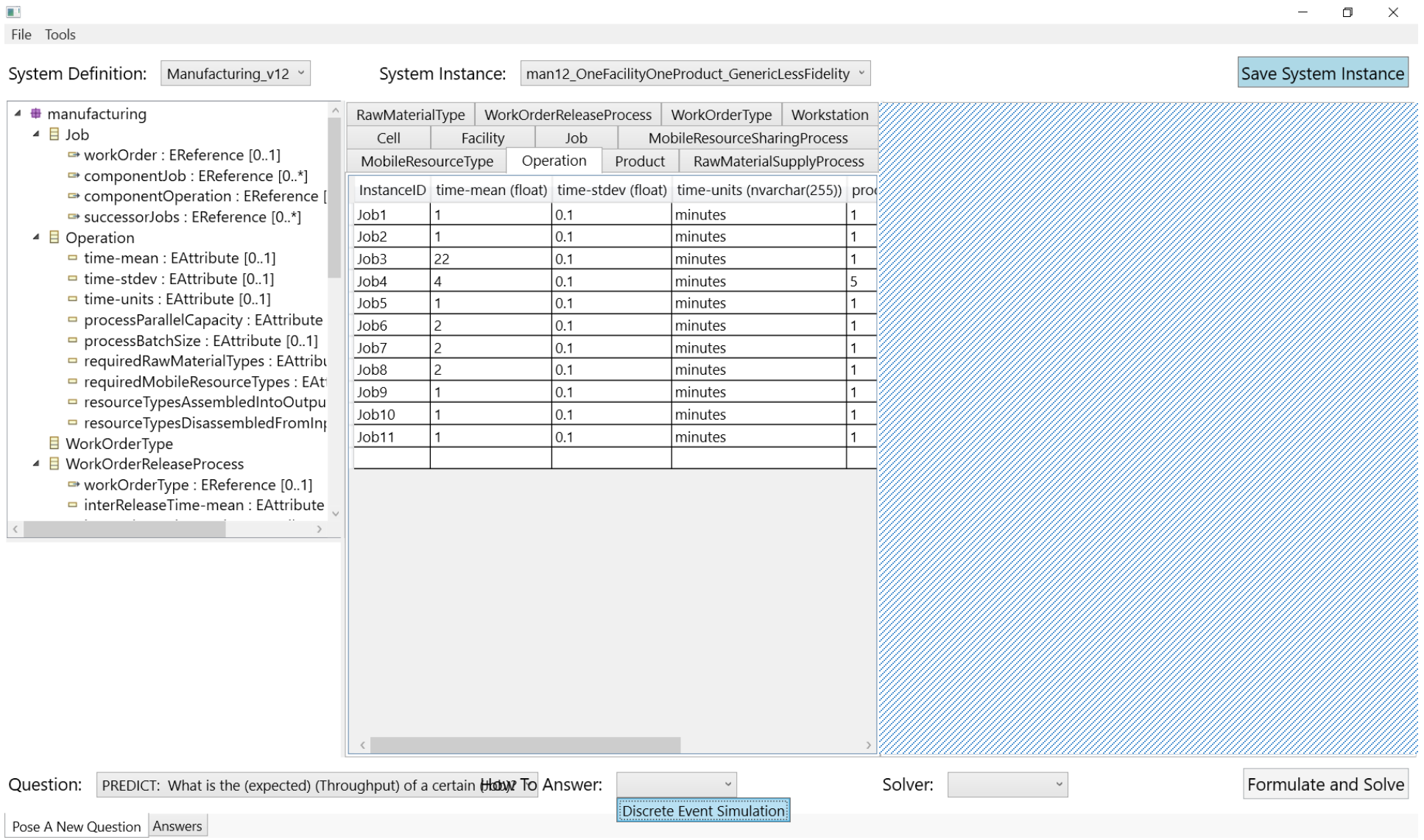
Results: Manufacturing Demo



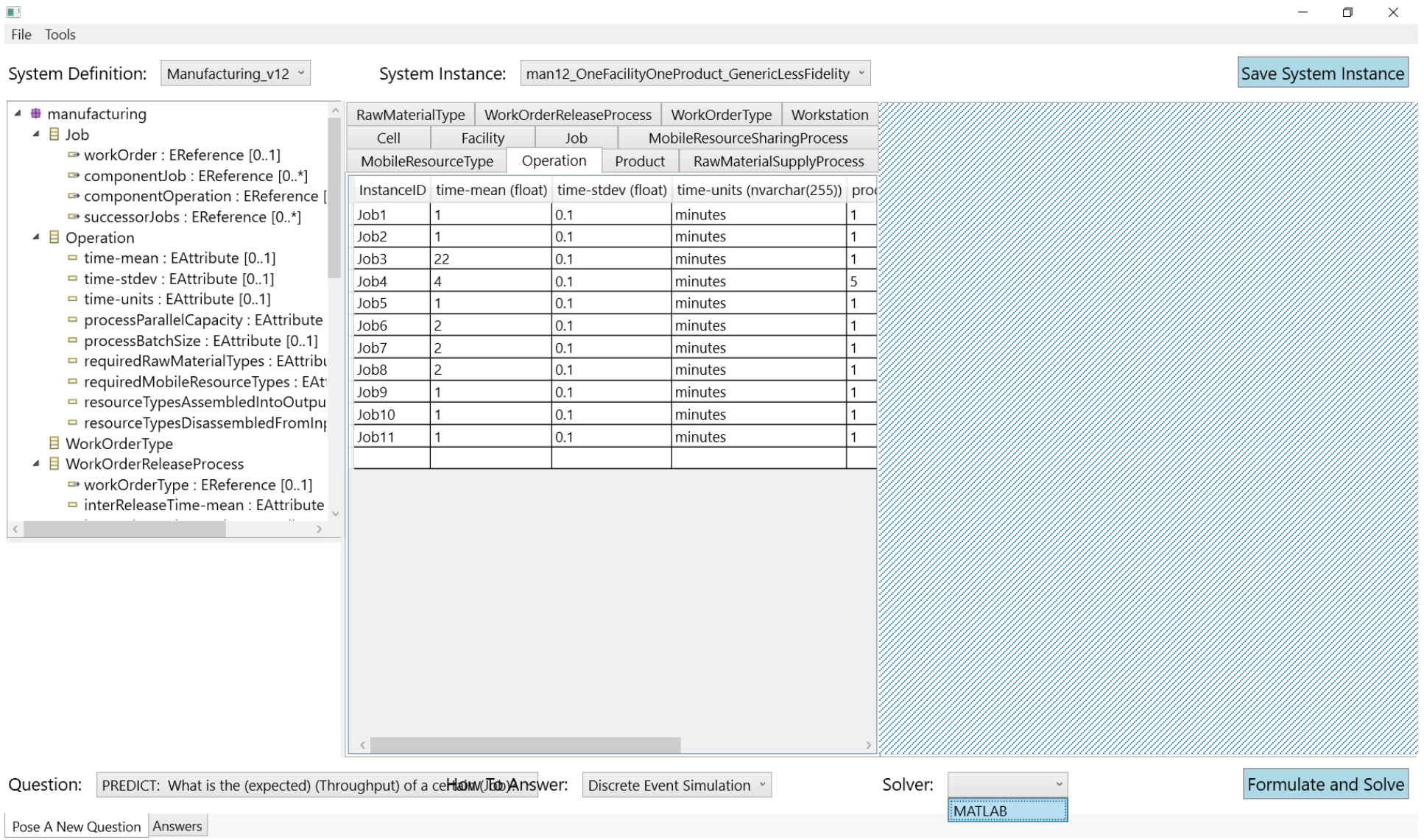
Results: Manufacturing Demo



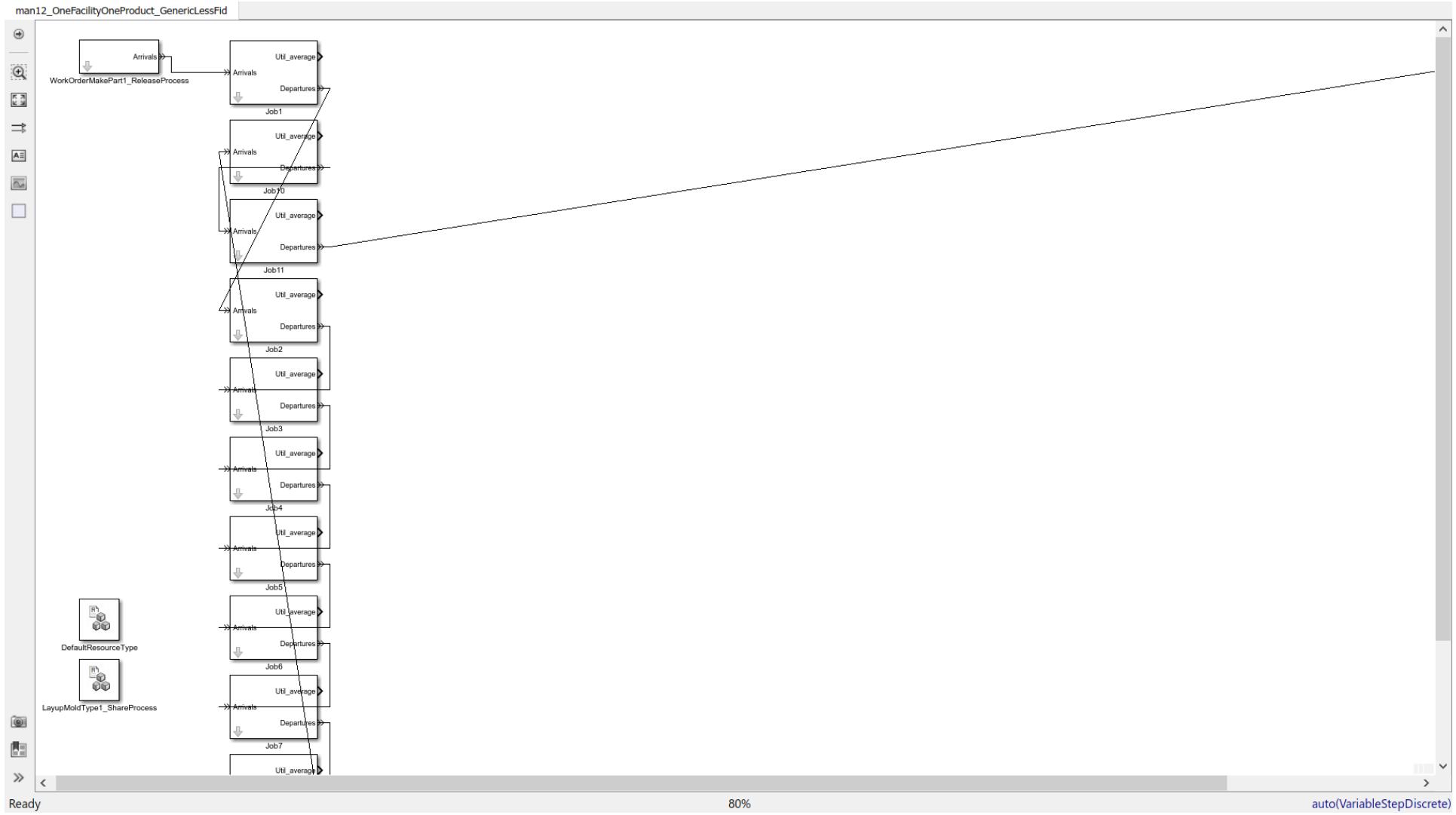
Results: Manufacturing Demo



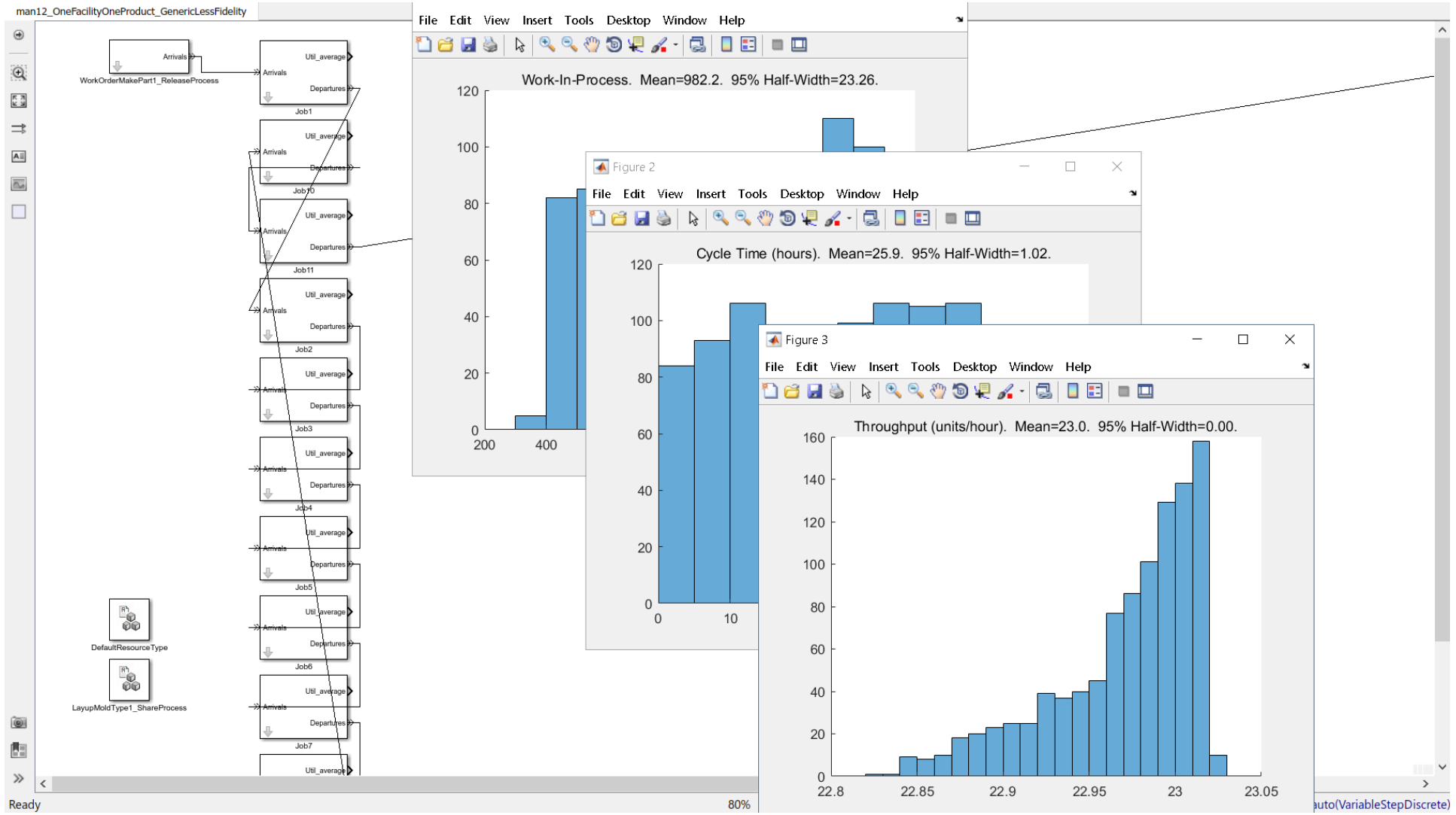
Results: Manufacturing Demo



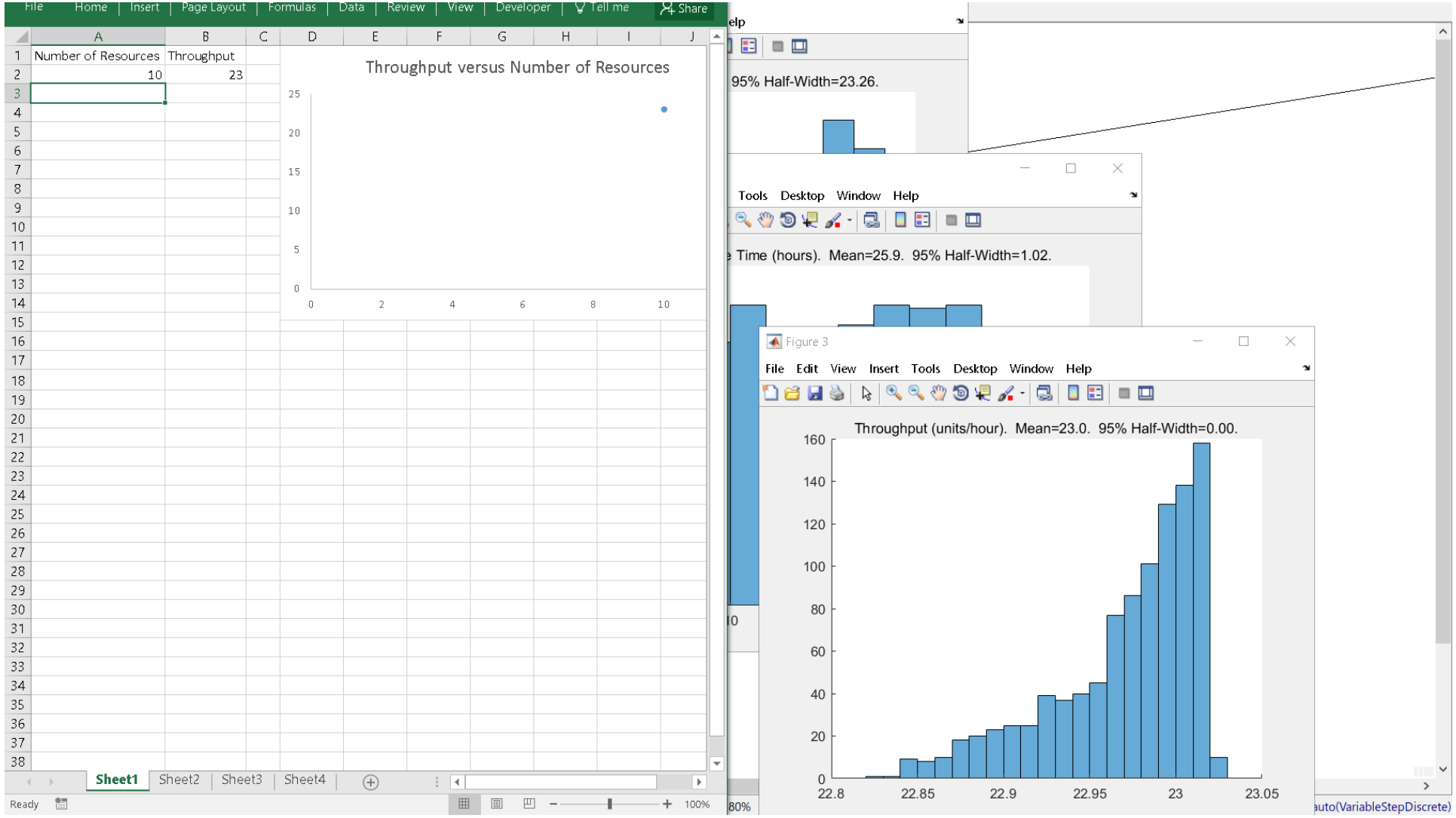
Results: Manufacturing Demo



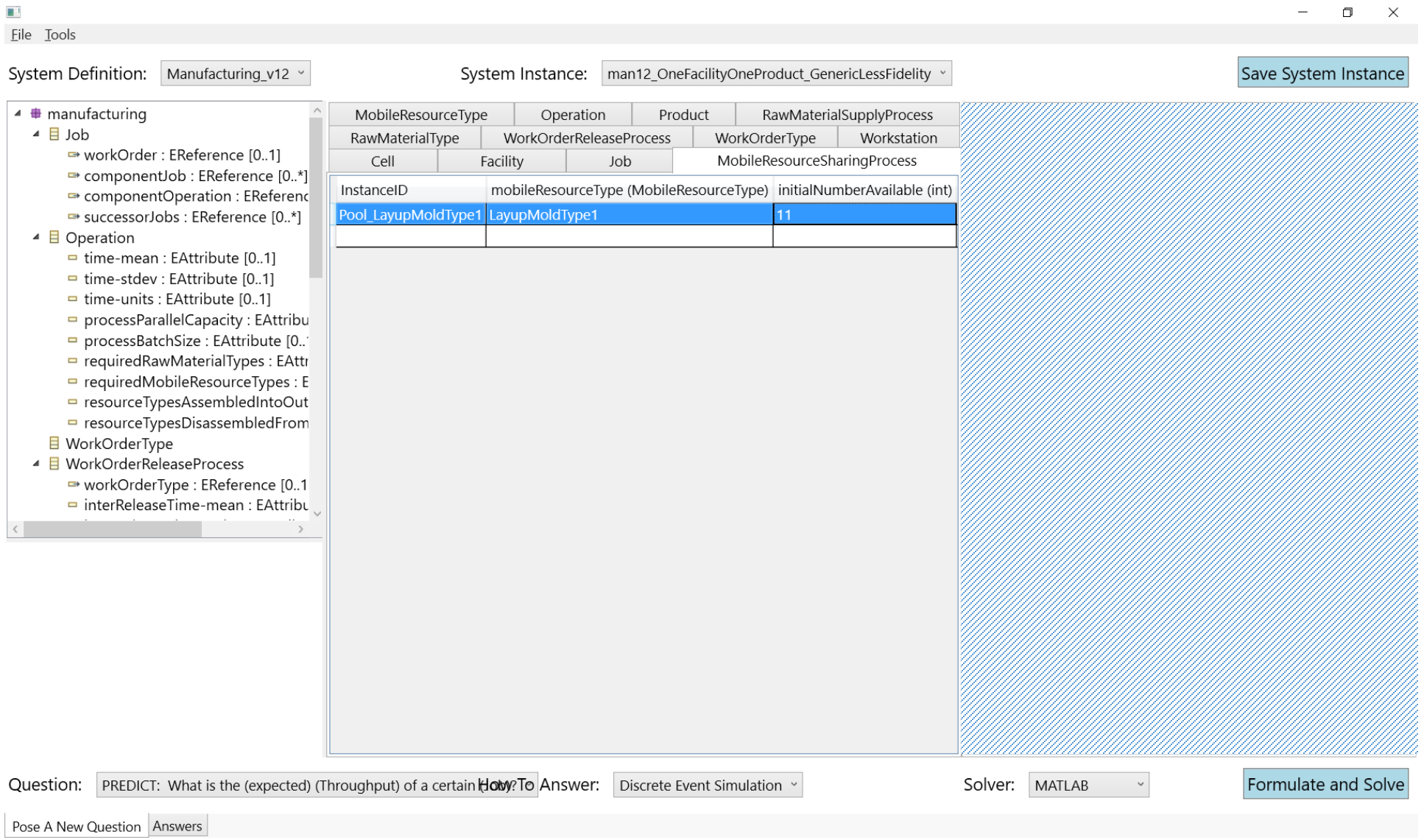
Results: Manufacturing Demo



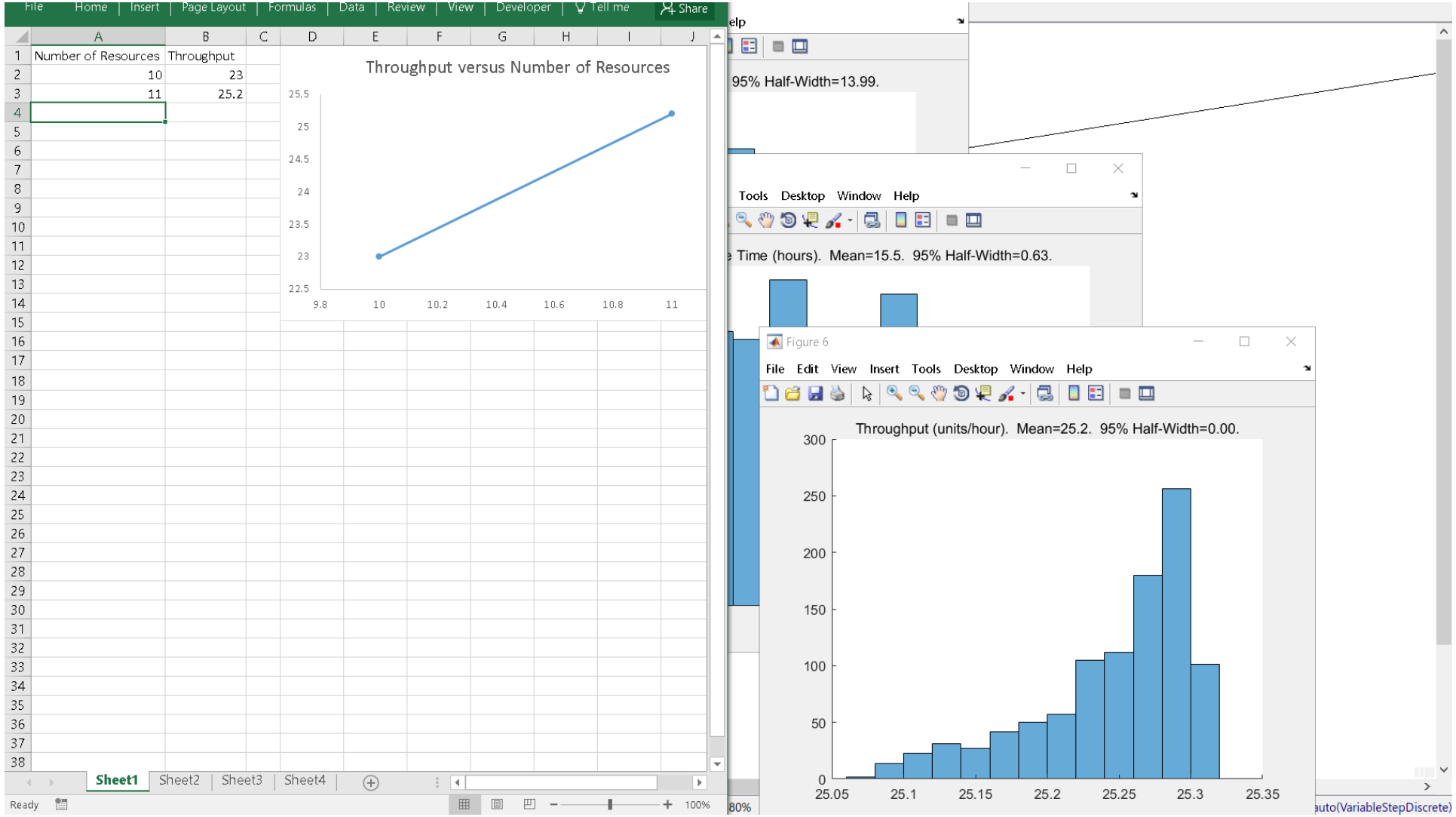
Results: Manufacturing Demo



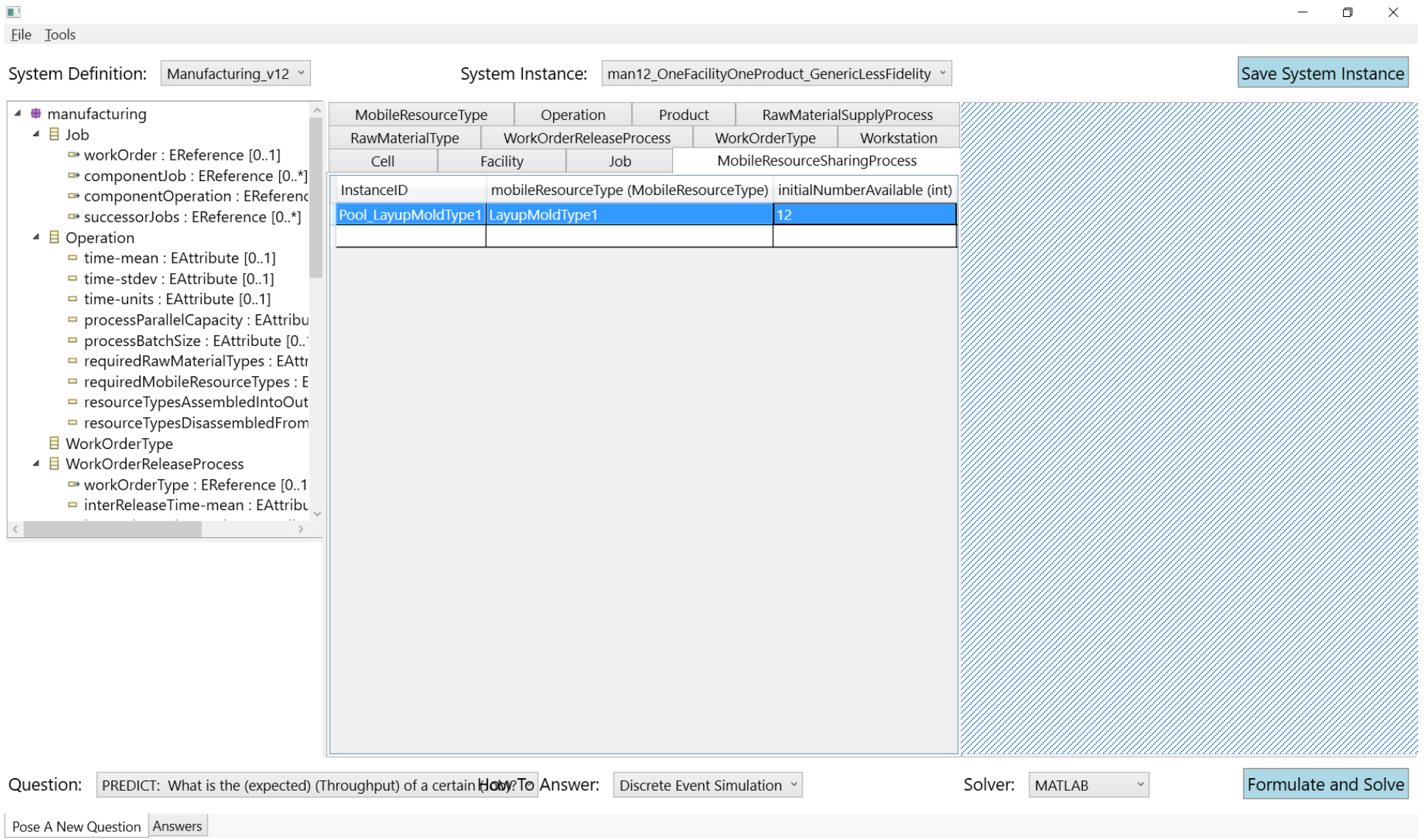
Results: Manufacturing Demo



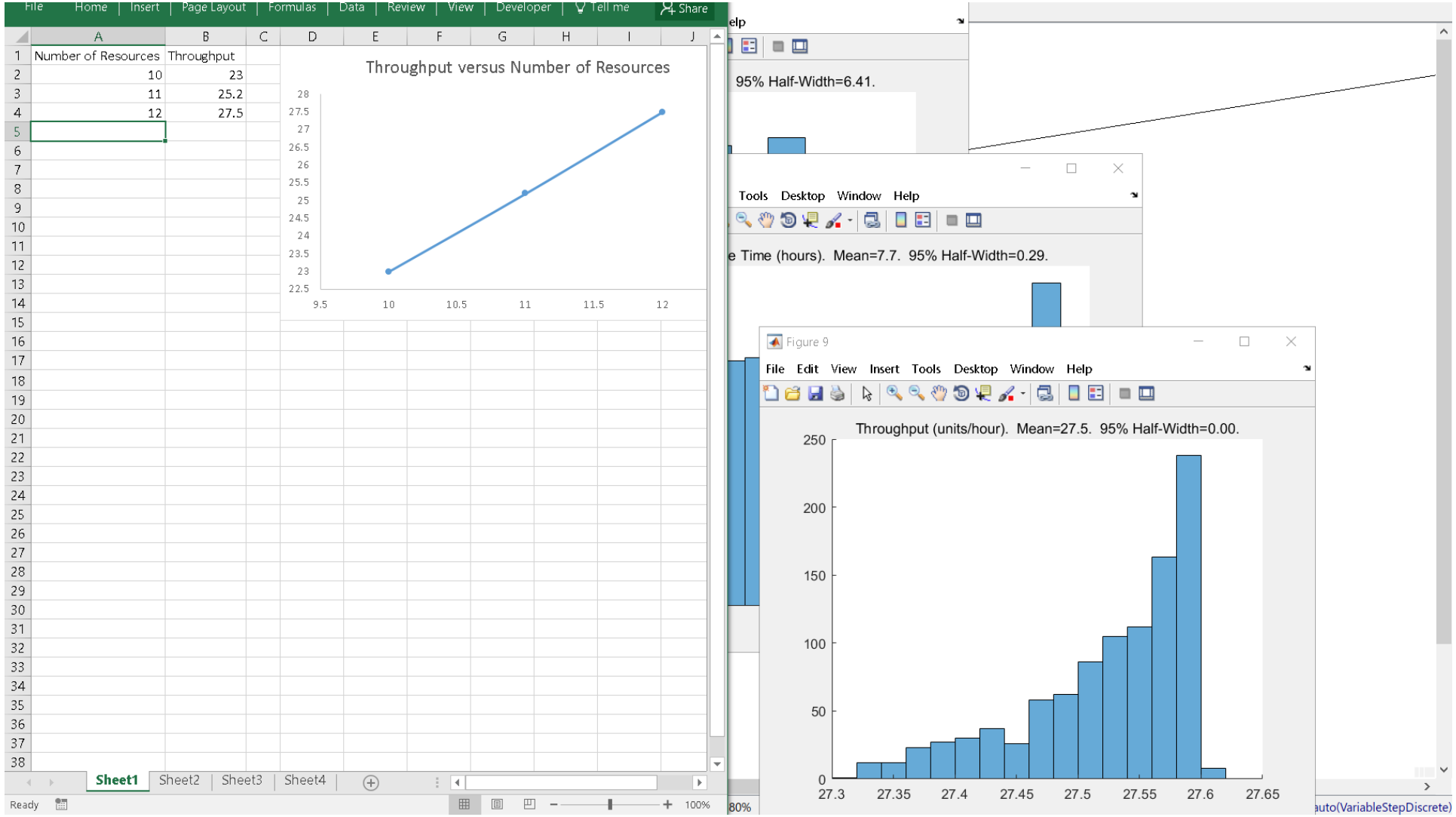
Results: Manufacturing Demo



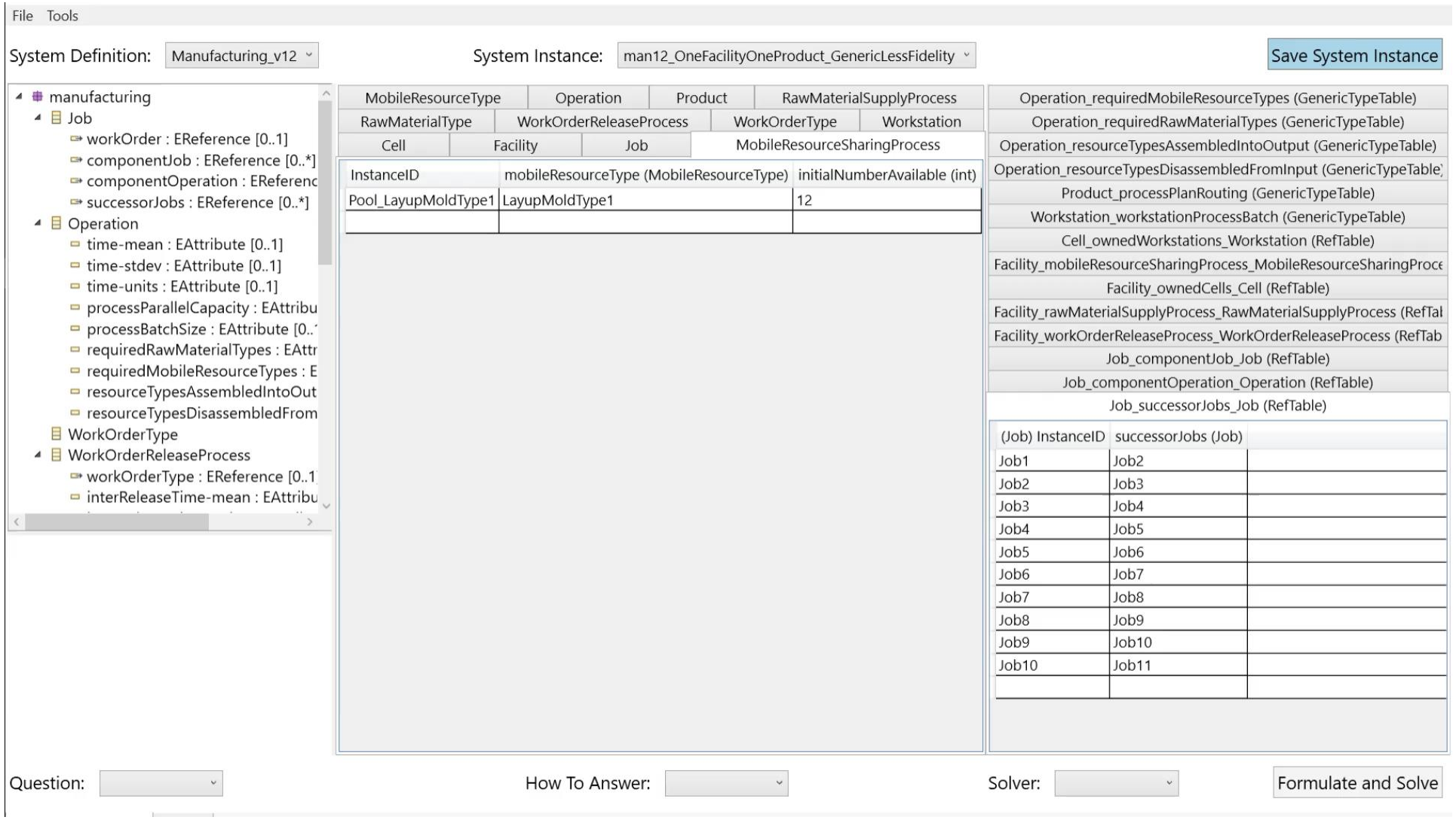
Results: Manufacturing Demo



Results: Manufacturing Demo



Results: Manufacturing Demo



Demonstration of Efficacy

To demonstrate an improvement in the accessibility and affordability of discrete-event simulation analysis for production systems, in one hour of work, I evaluated 100 different alternatives for a production system and answered several questions about each:

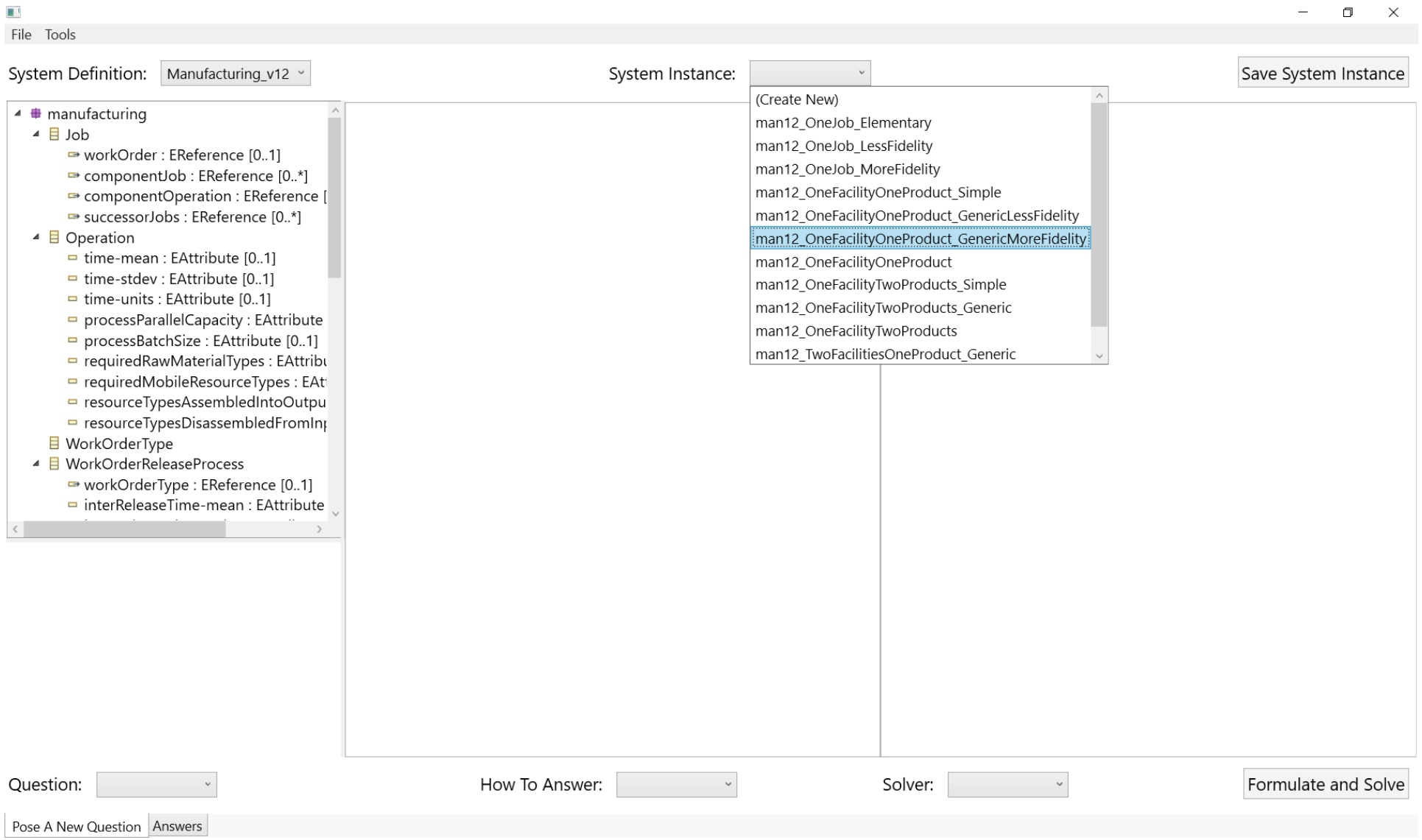
- Change resource numbers
- Change order release schedule
- Change material resupply variability

Parametric Changes

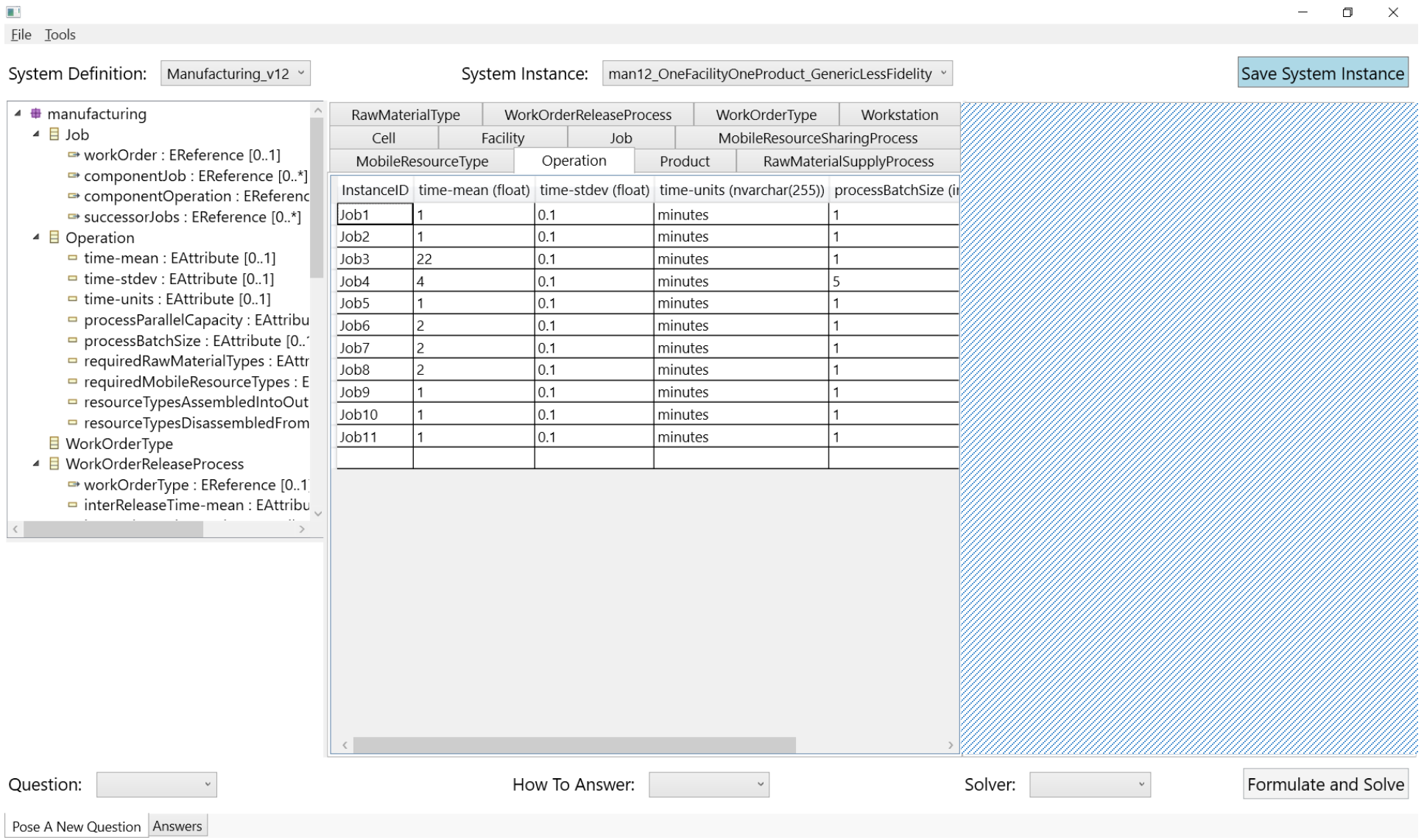
- Change process plan fidelity/ level of abstraction
- Change workstation batching rules
- Change a process plan's routing through a facility
- Change the facility executing a process plan
- Change other process plans executing concurrently in a facility

Structural Changes

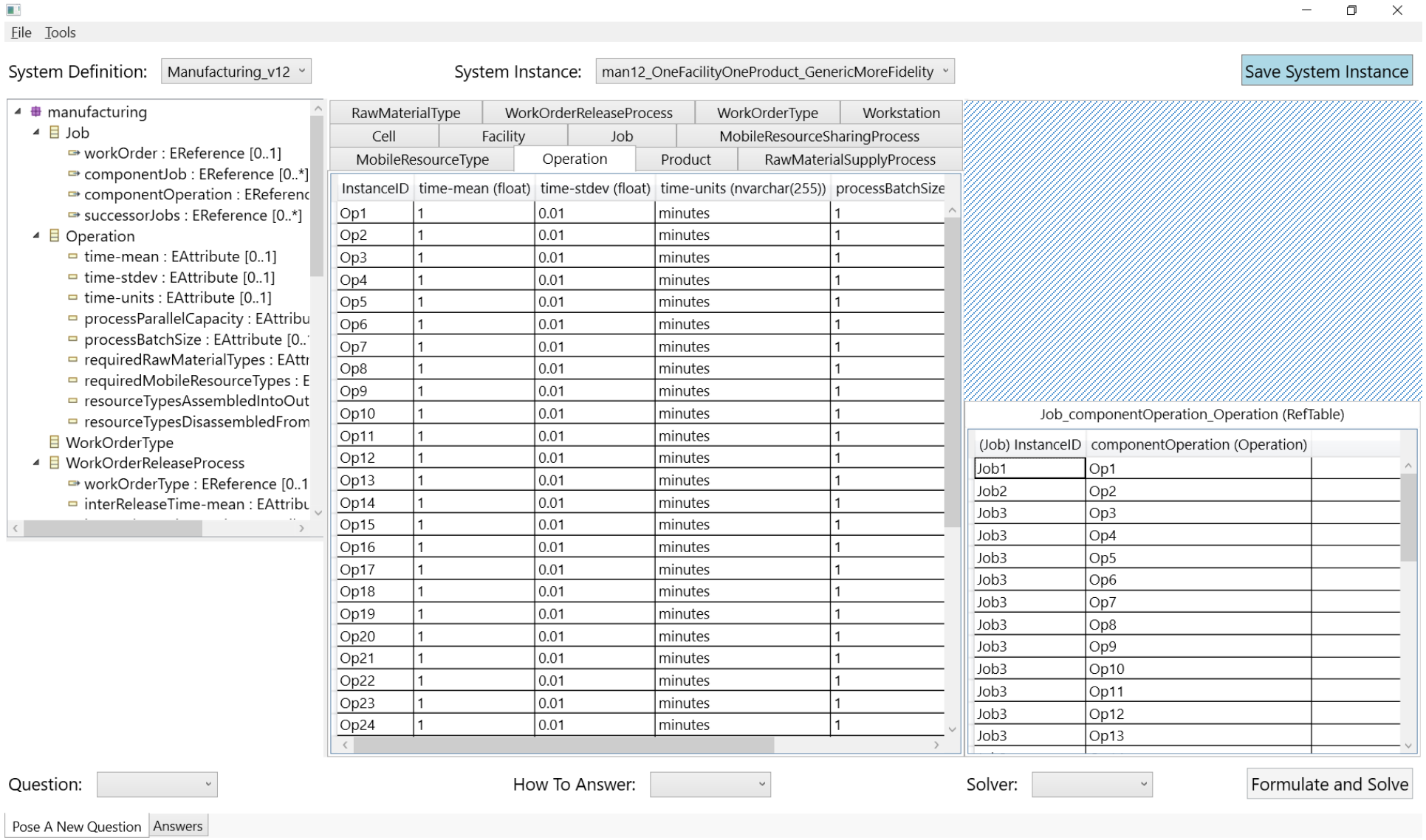
Results: Manufacturing Demo



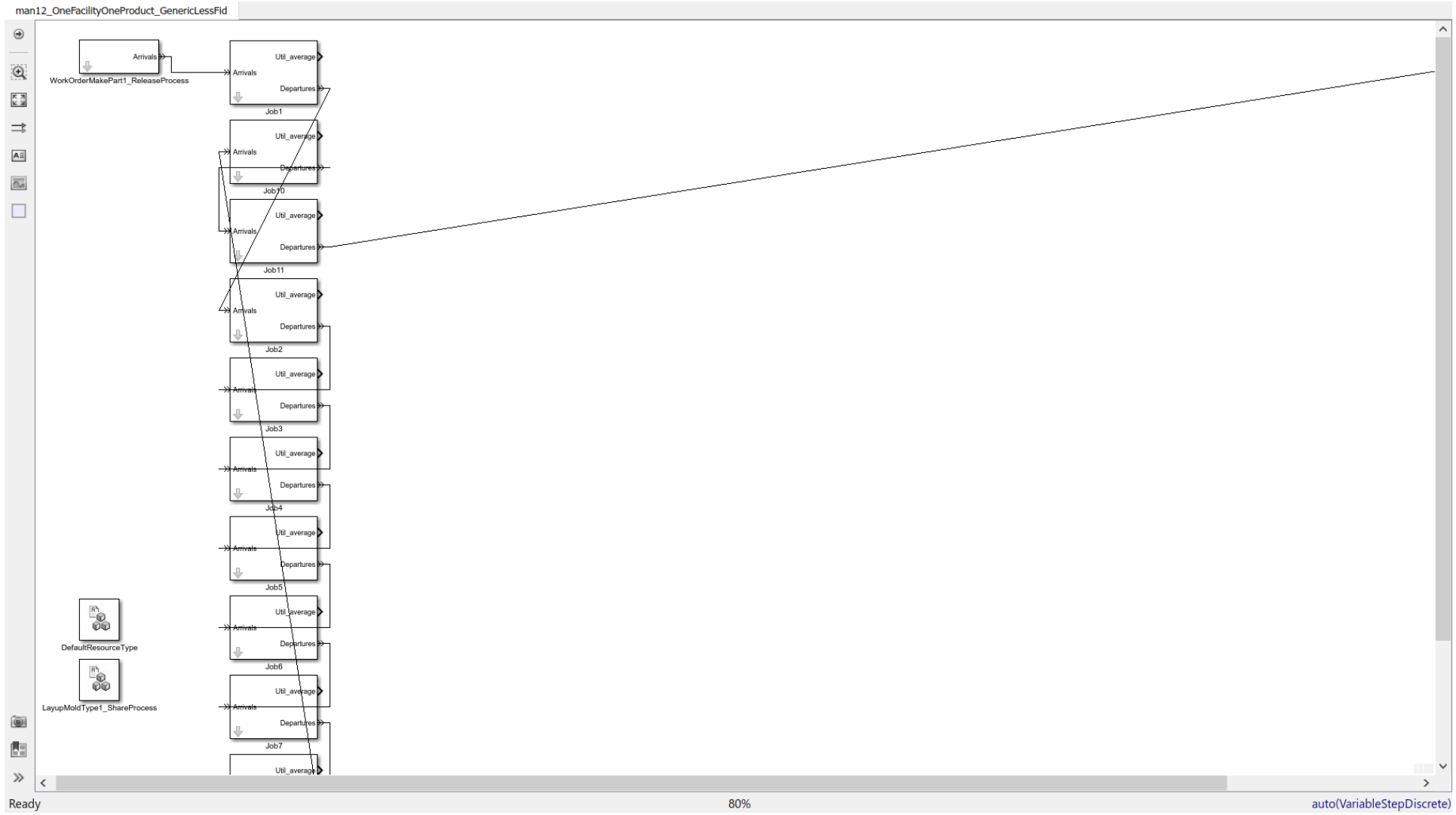
Results: Manufacturing Demo



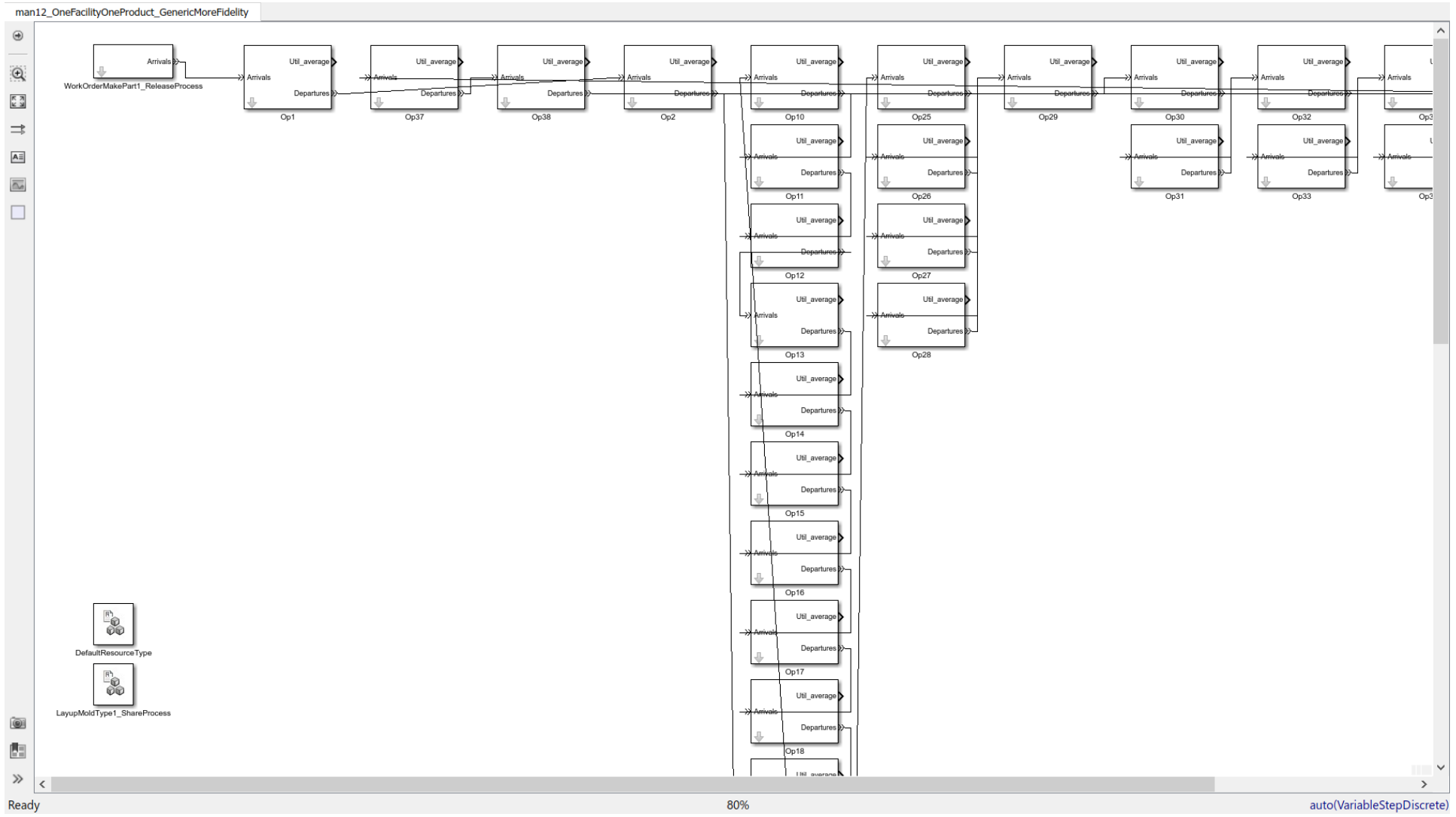
Results: Manufacturing Demo



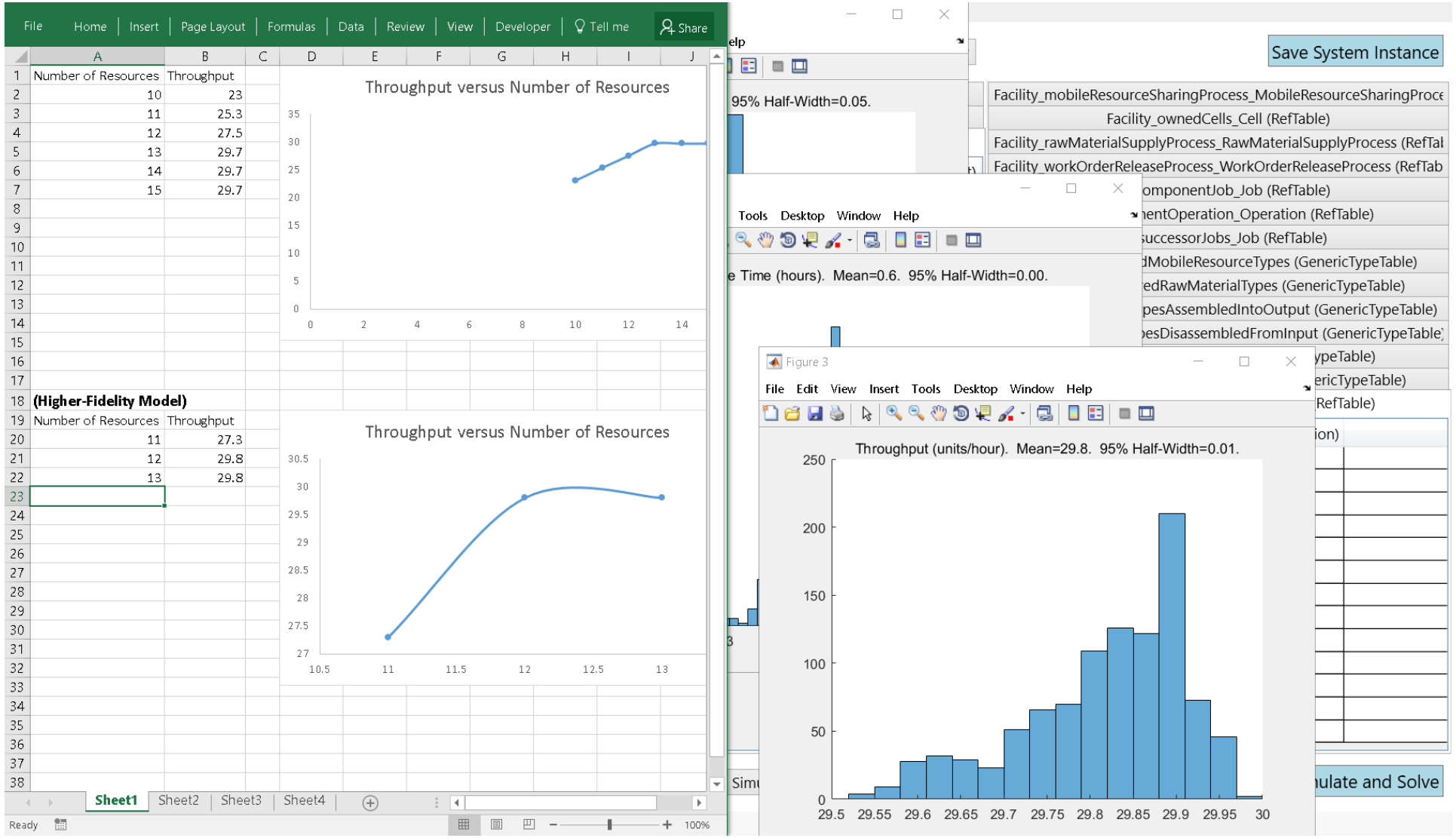
Results: Manufacturing Demo



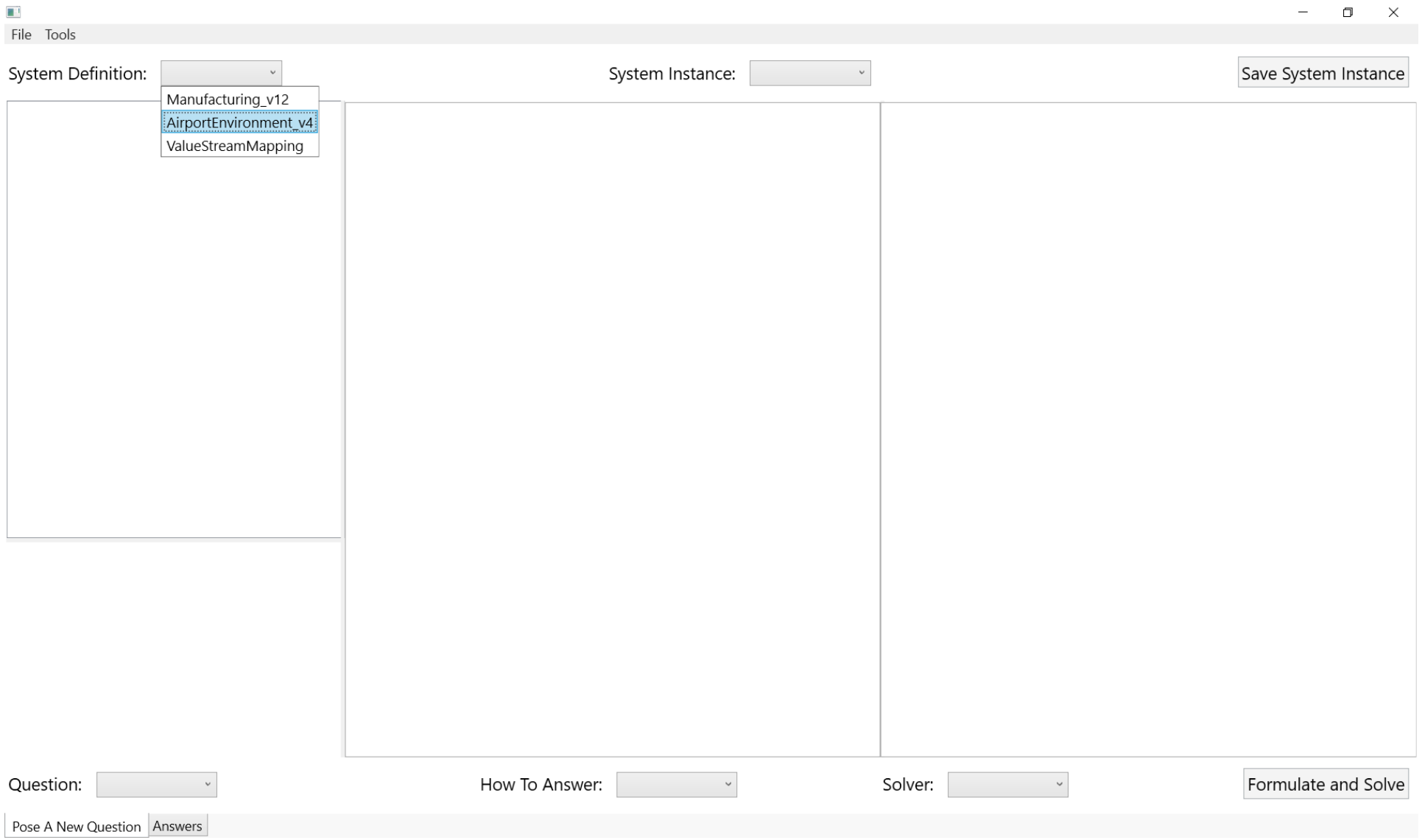
Results: Manufacturing Demo



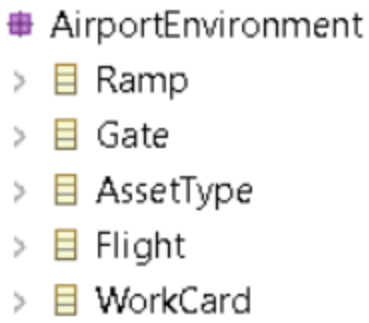
Results: Manufacturing Demo



Results: Air Cargo Sort Hub Demo

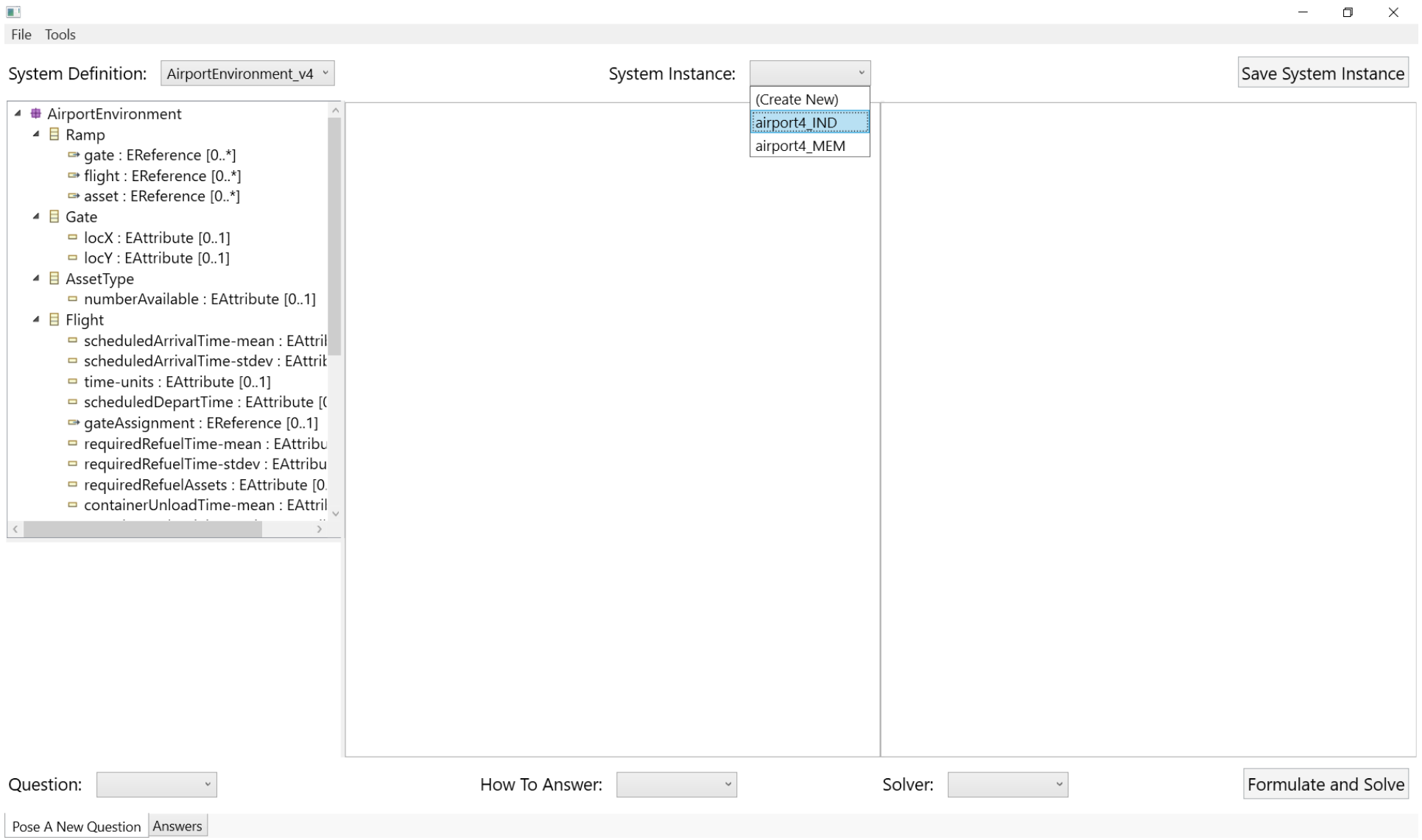


Results: Air Cargo Sort Hub Demo

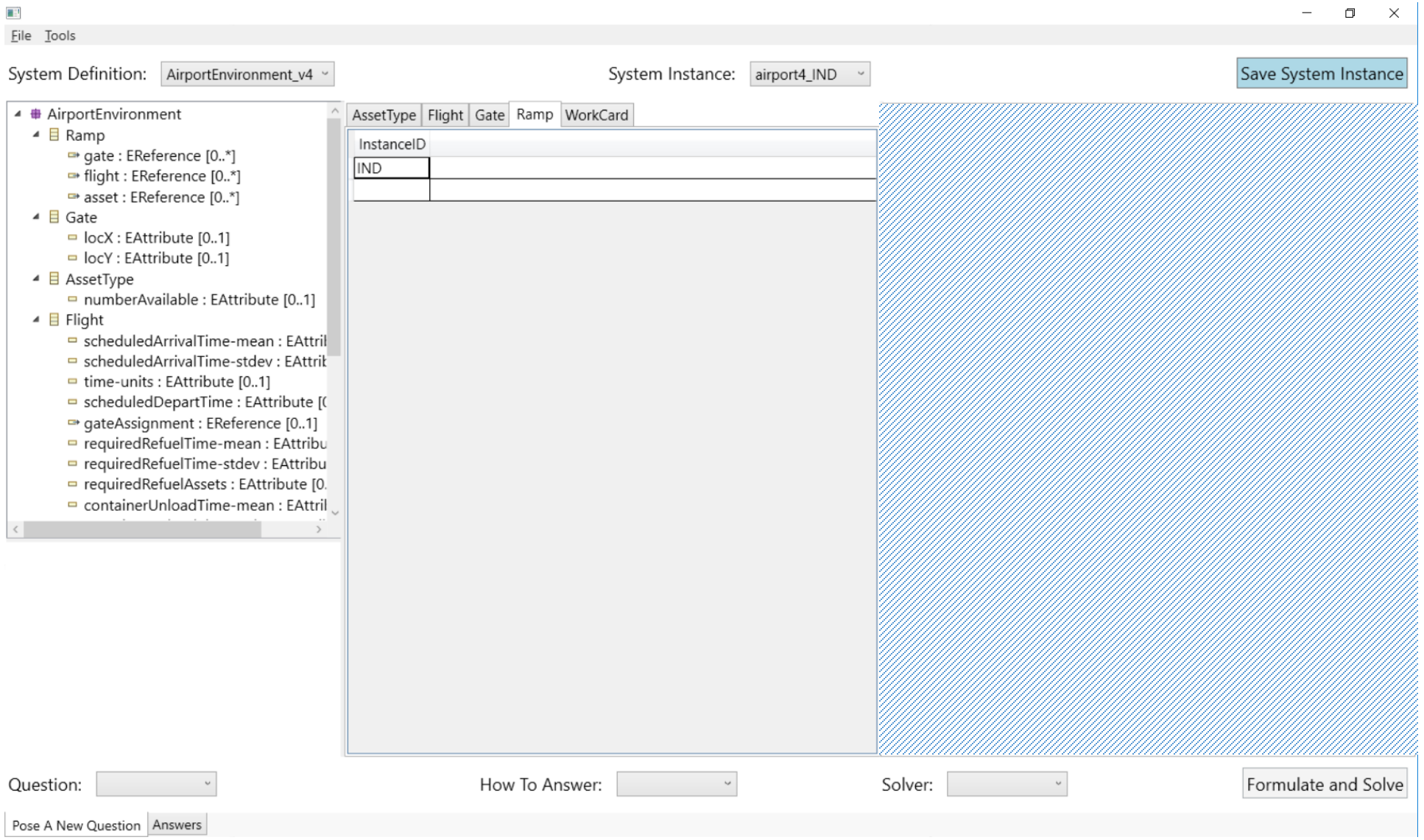


Complete Domain Switch, Same Analysis Generators

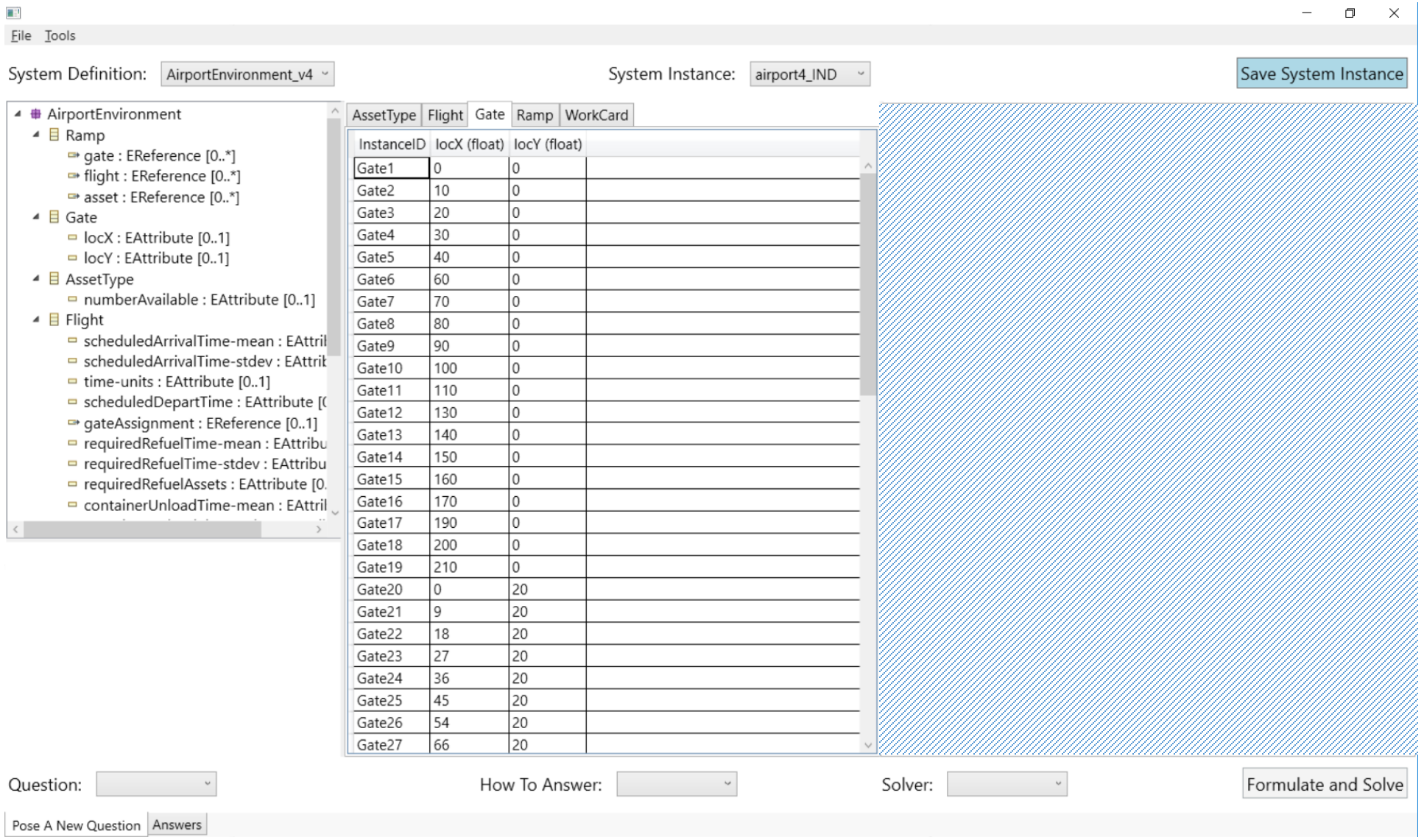
Results: Air Cargo Sort Hub Demo



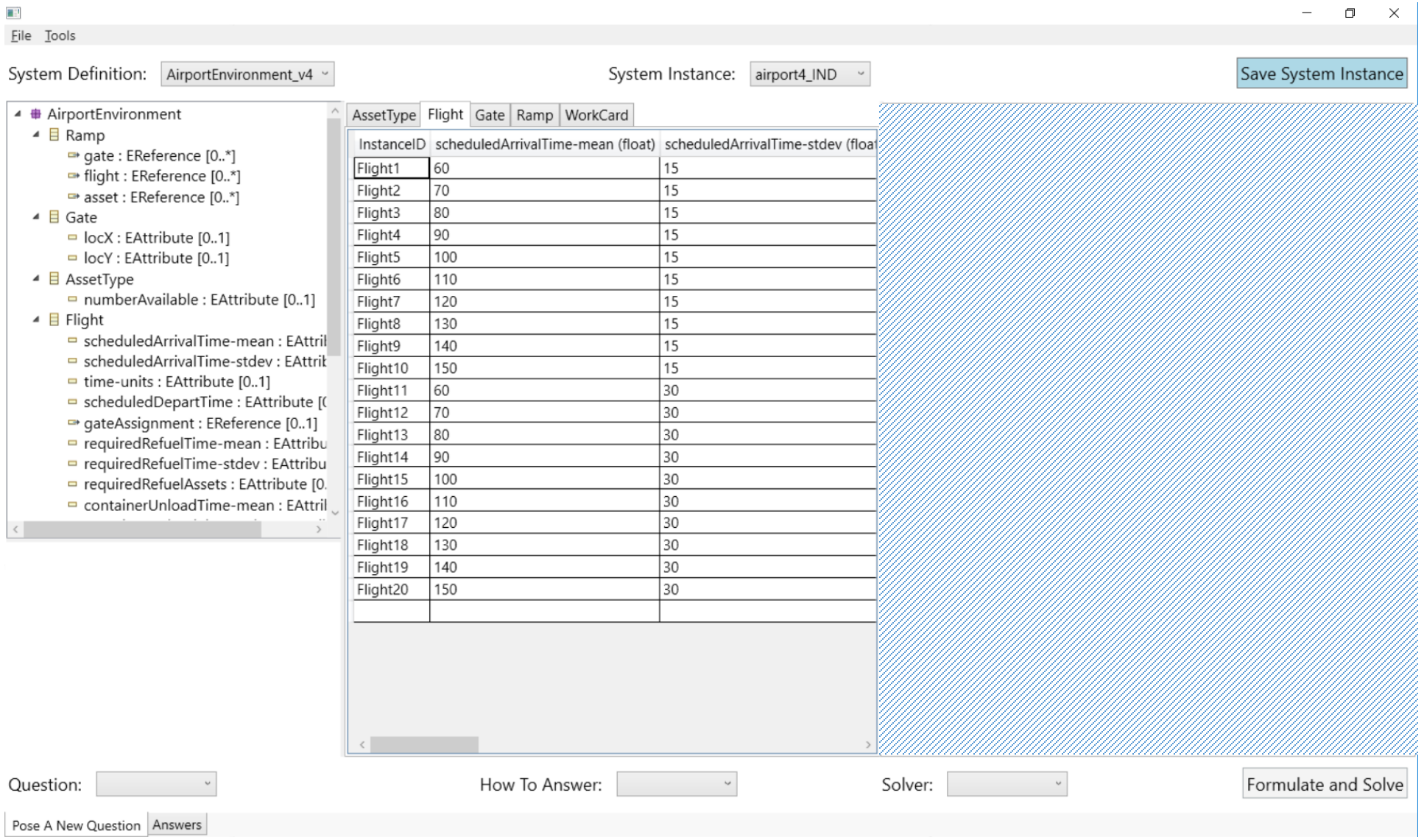
Results: Air Cargo Sort Hub Demo



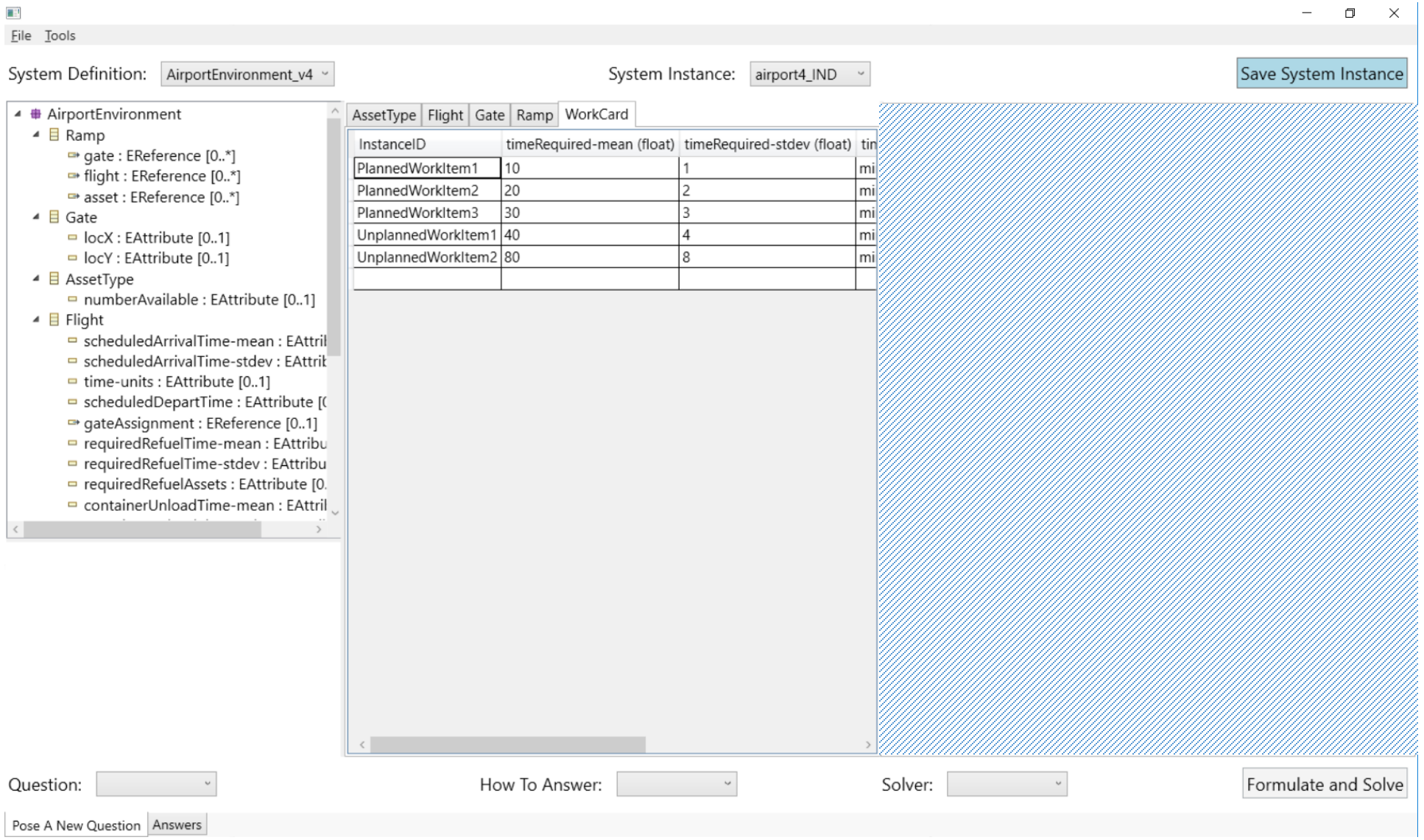
Results: Air Cargo Sort Hub Demo



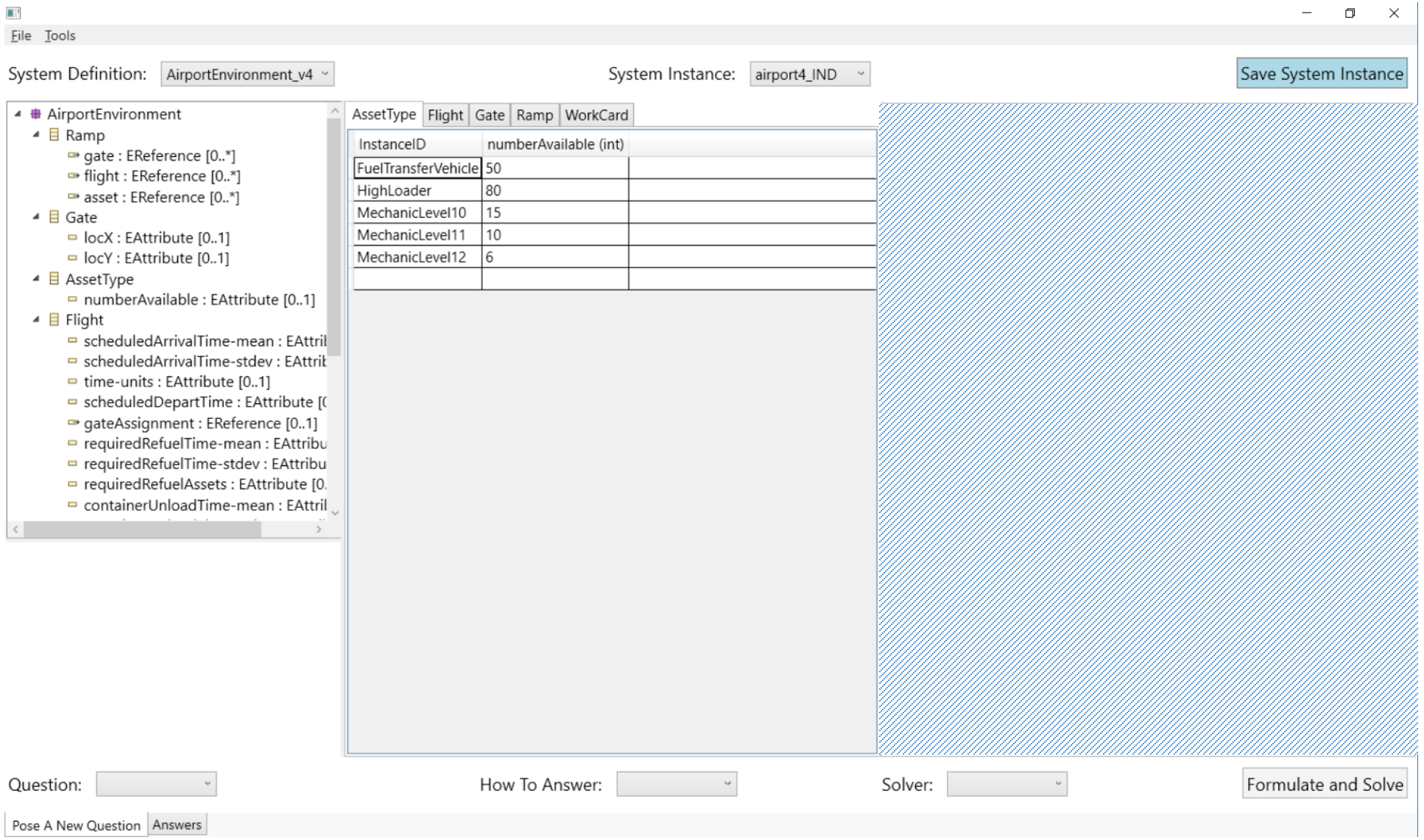
Results: Air Cargo Sort Hub Demo



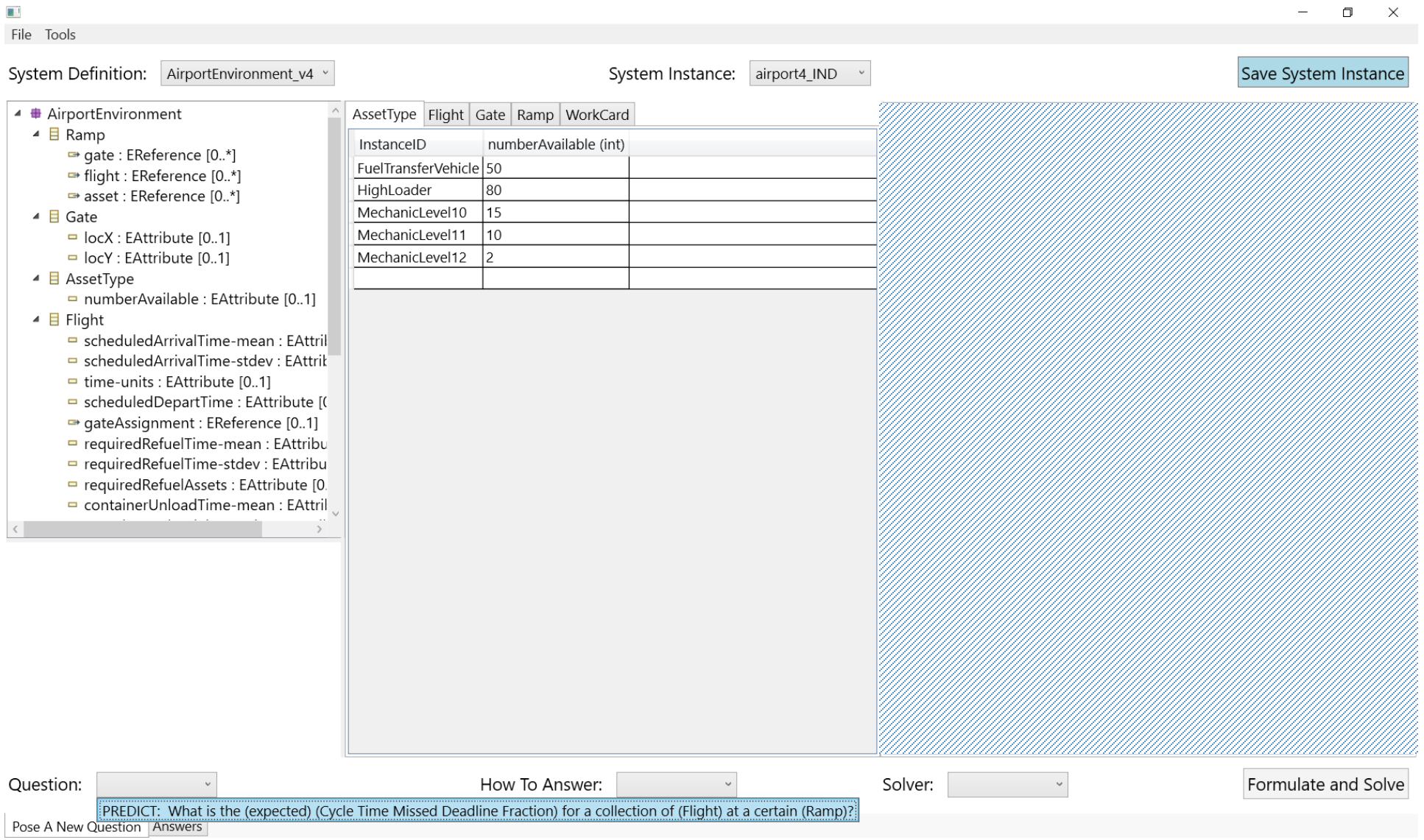
Results: Air Cargo Sort Hub Demo



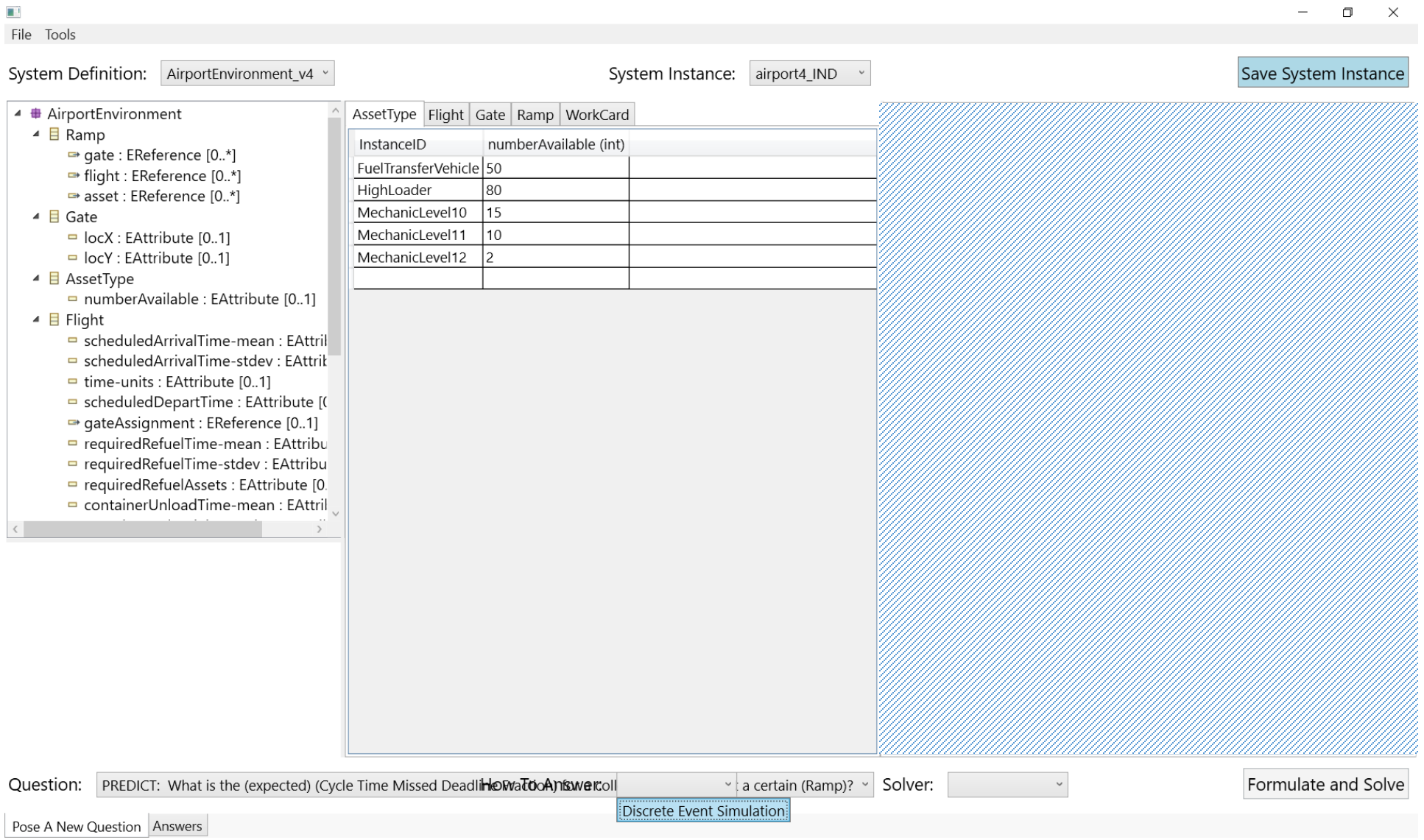
Results: Air Cargo Sort Hub Demo



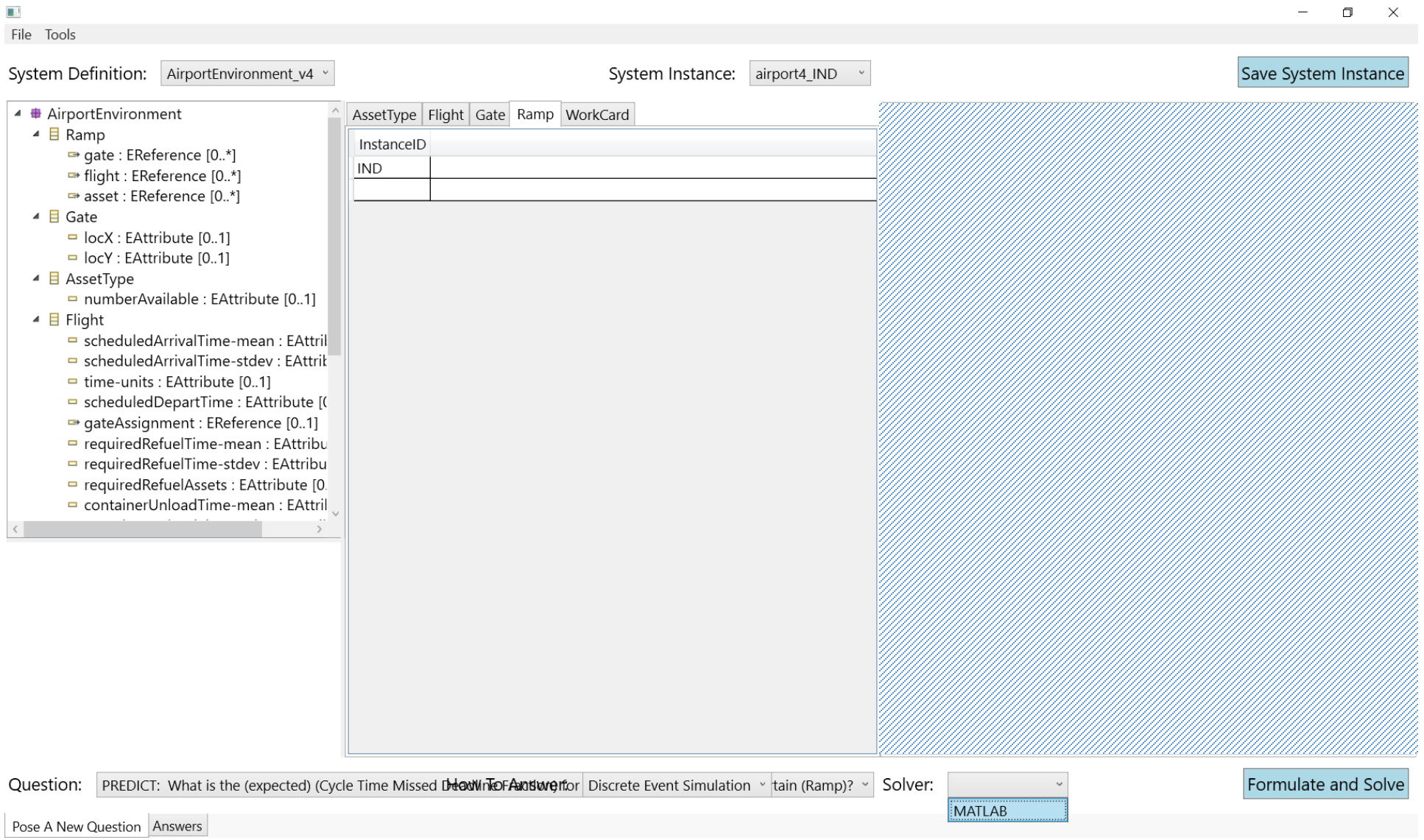
Results: Air Cargo Sort Hub Demo



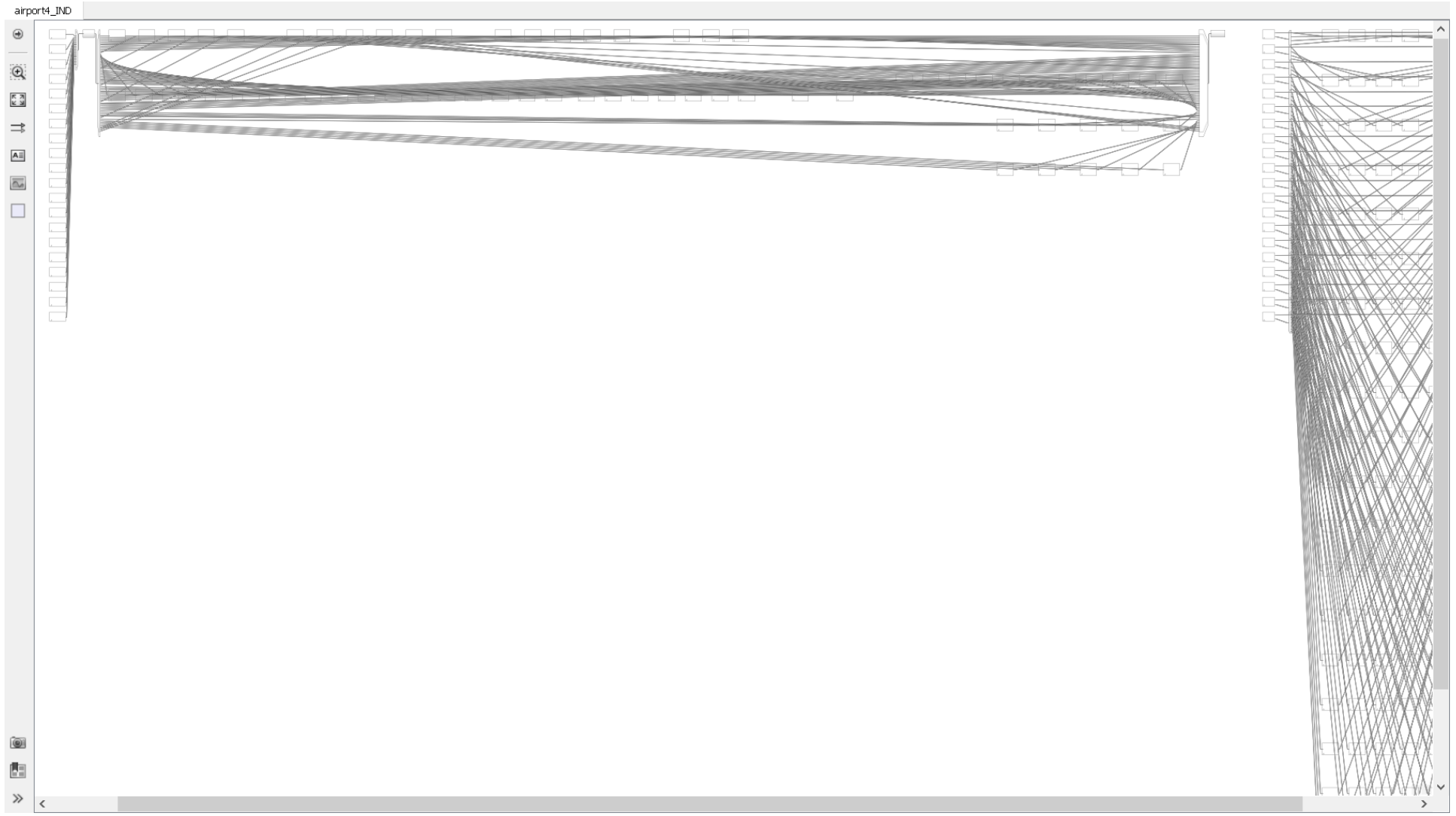
Results: Air Cargo Sort Hub Demo



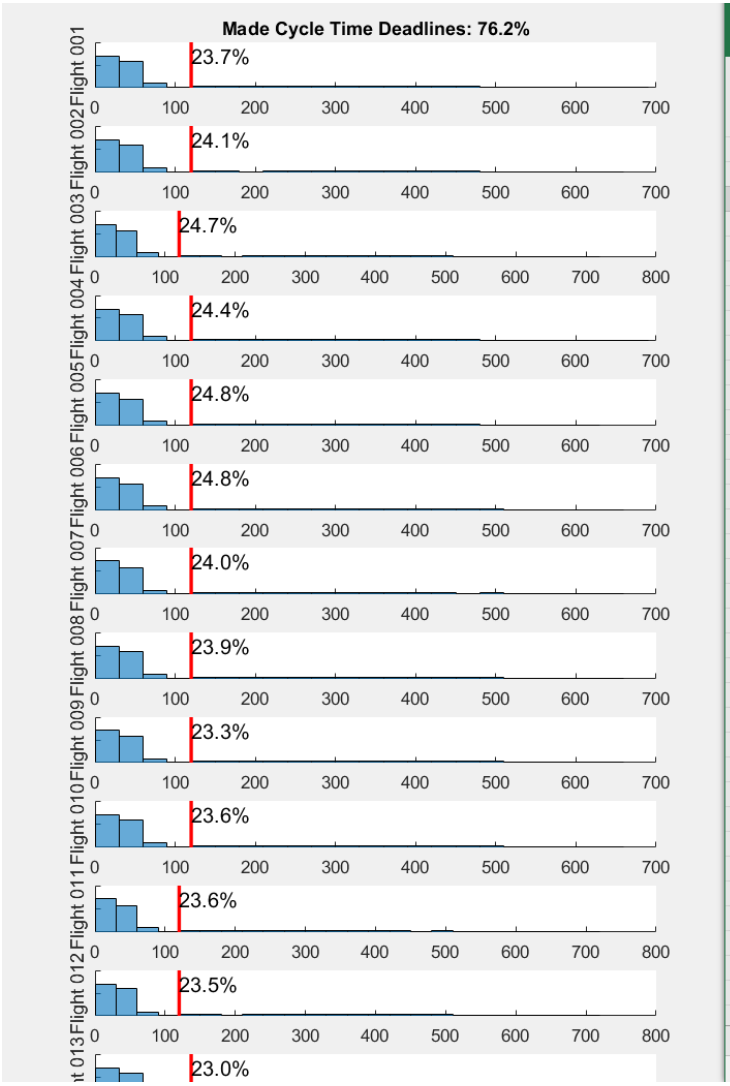
Results: Air Cargo Sort Hub Demo

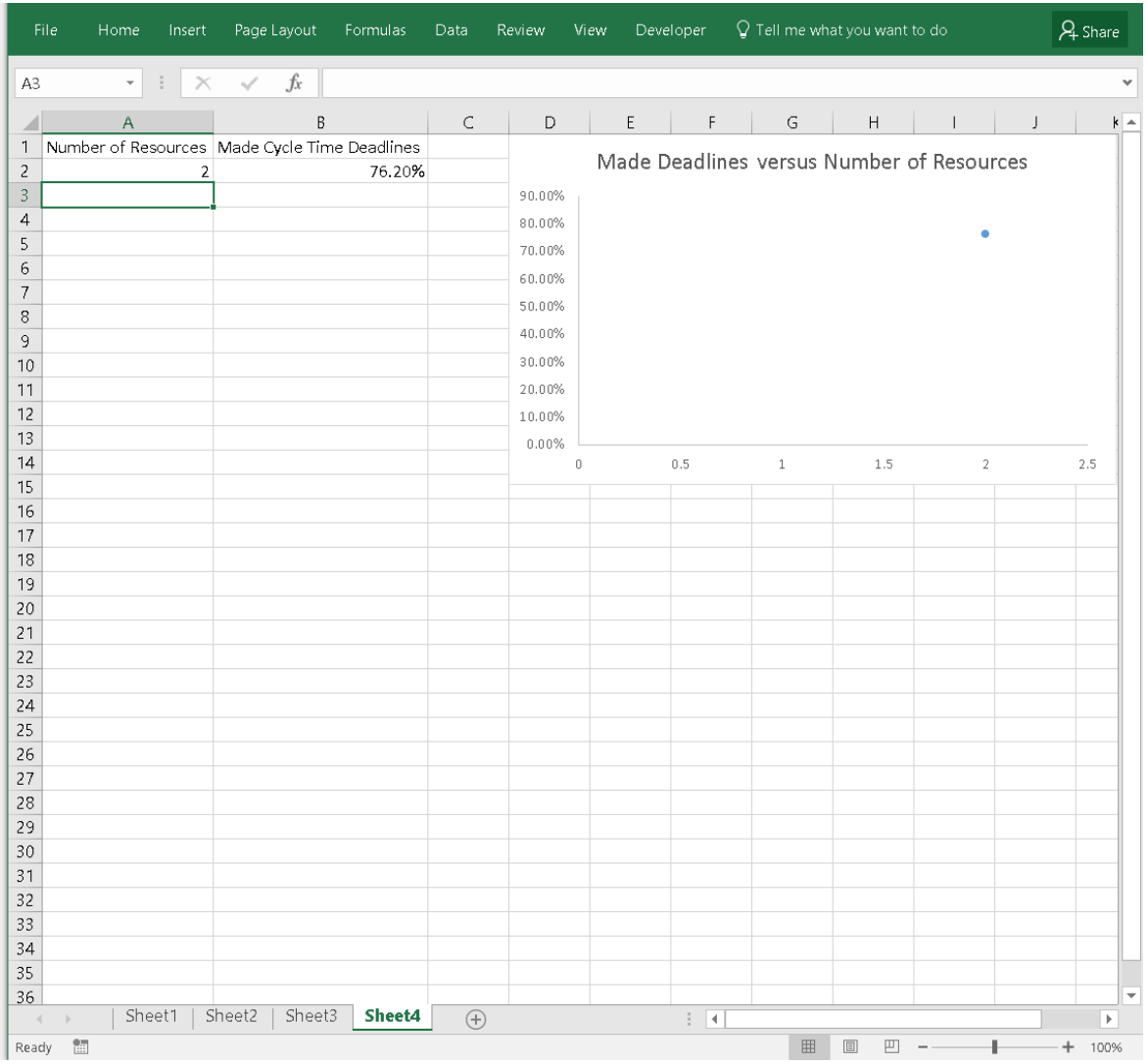


Results: Air Cargo Sort Hub Demo



Results: Air Cargo Sort Hub Demo





Demonstration of Efficacy

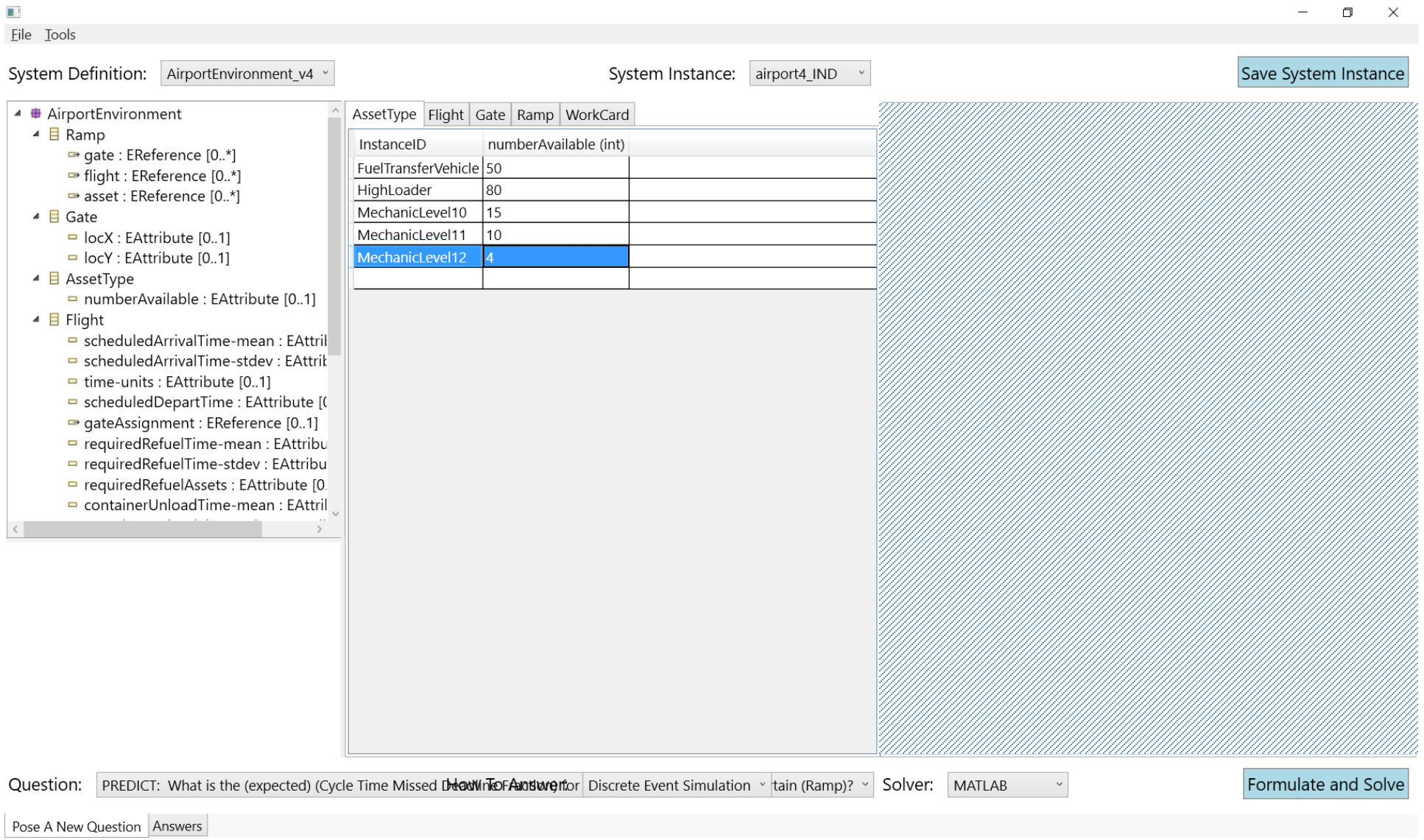
To demonstrate that the methodology is *more general than just the manufacturing domain* and also *more general than any particular analysis solver*, use the same tool to quickly evaluate 50 different alternatives for an air cargo sort hub, and answer several questions about each:

- Change resource numbers
- Change flight schedule
- Change parking plan
- Change airport
- Change maintenance profile (more preventative = less unplanned)
- Change maintenance rules

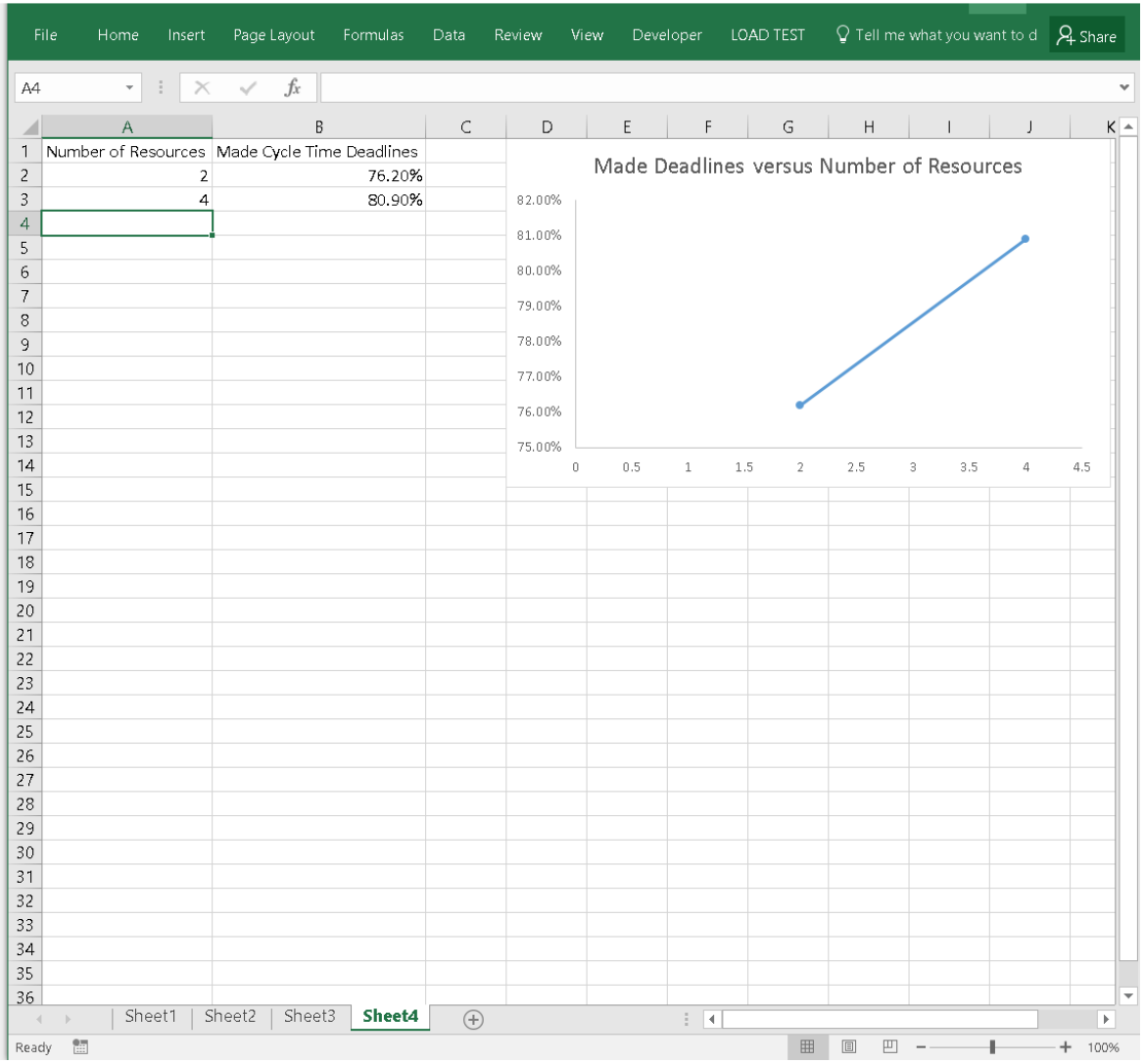
Parametric Changes

Structural Changes

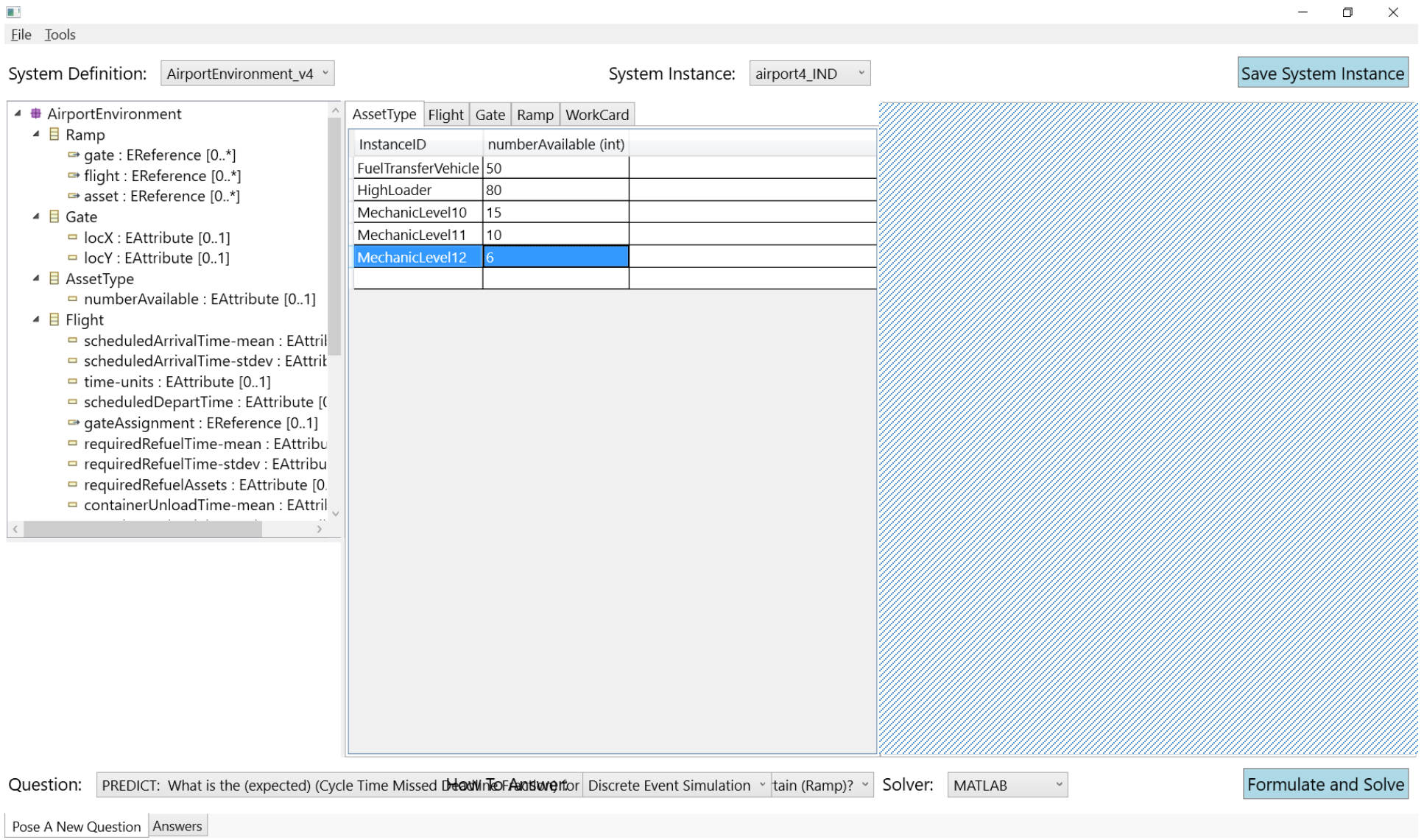
Results: Air Cargo Sort Hub Demo



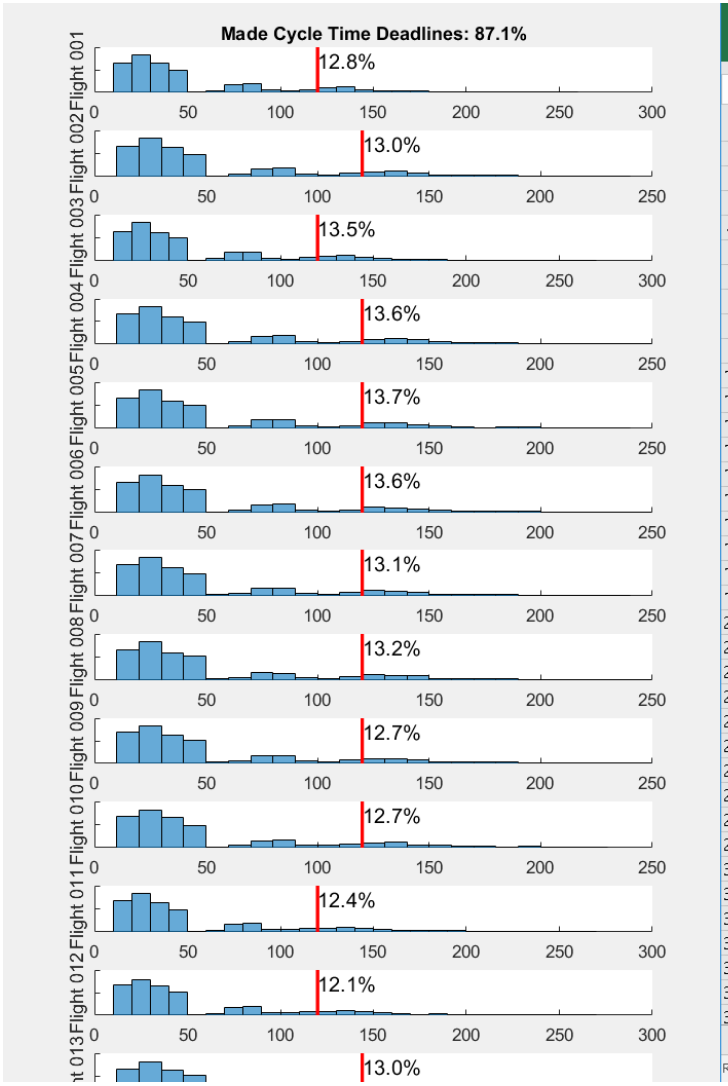
Results: Air Cargo Sort Hub Demo

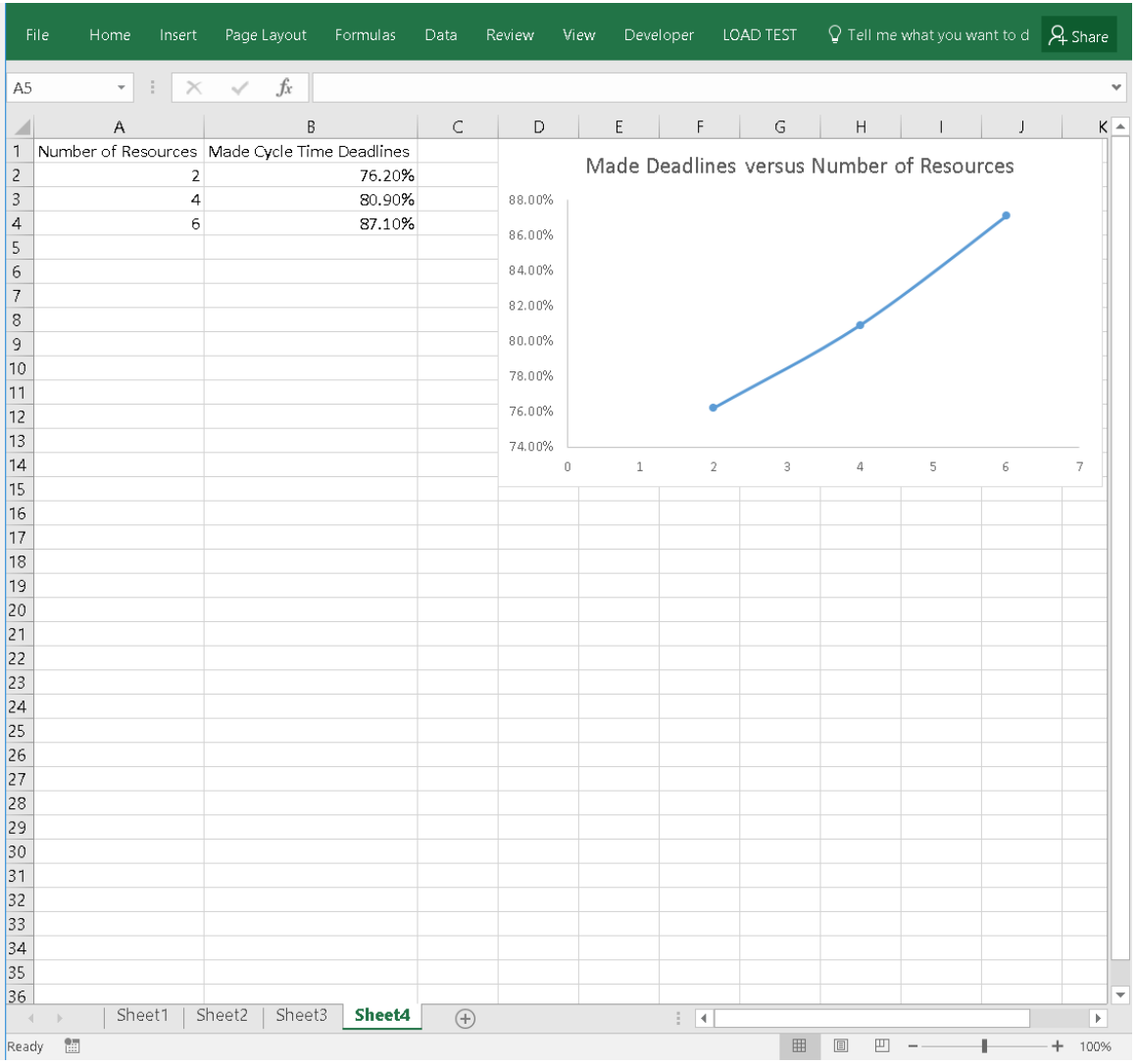


Results: Air Cargo Sort Hub Demo

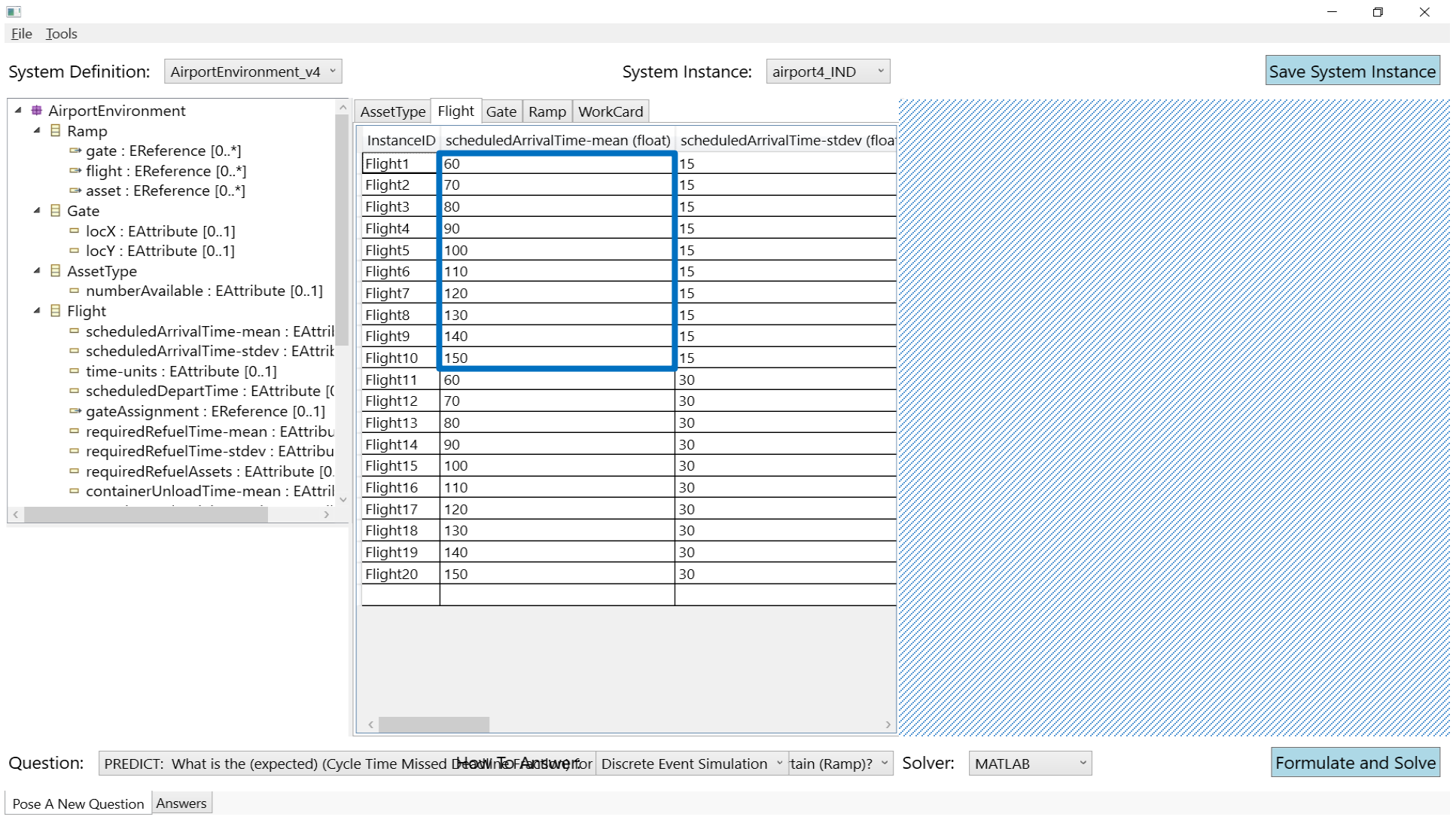


Results: Air Cargo Sort Hub Demo

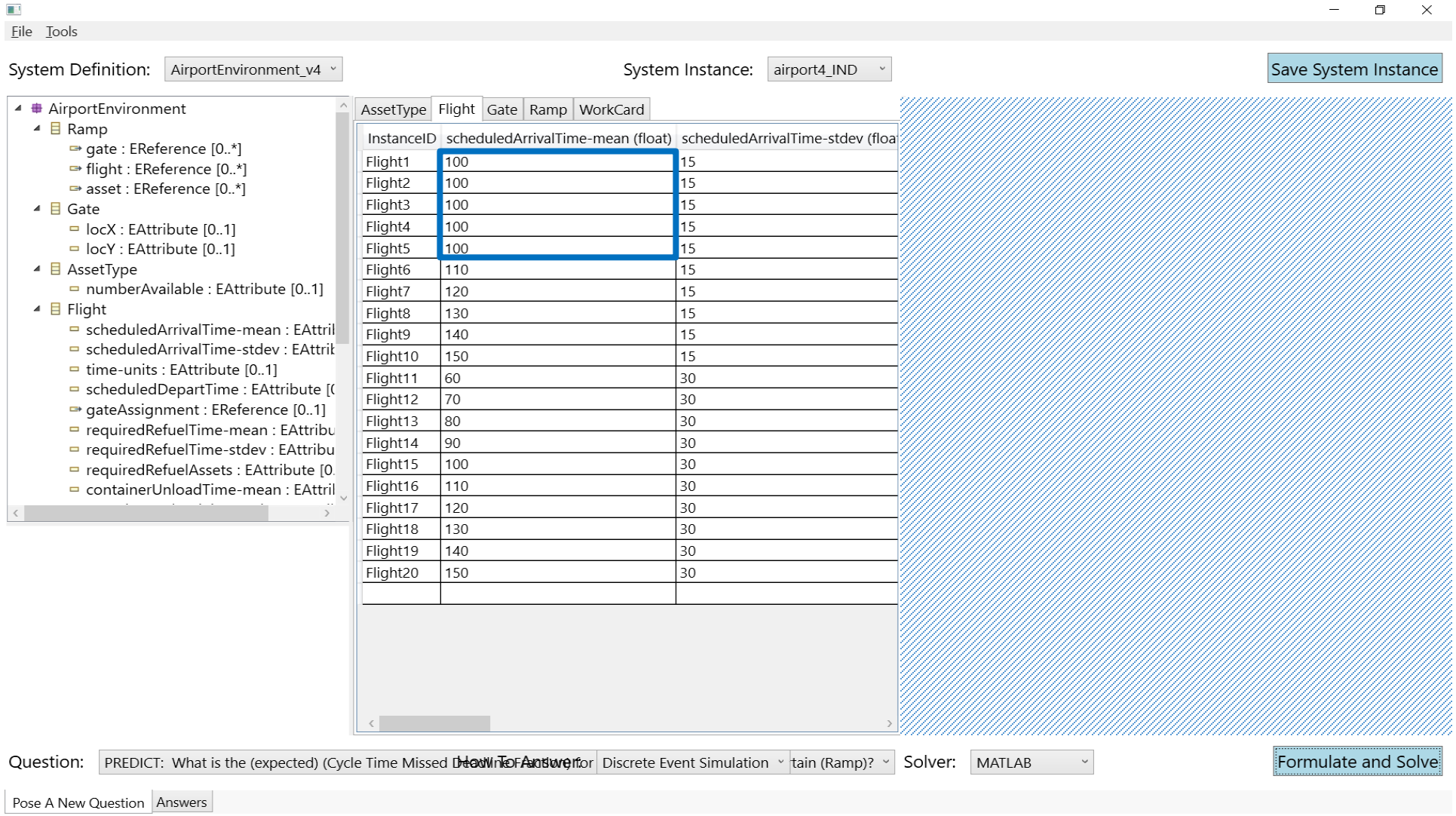




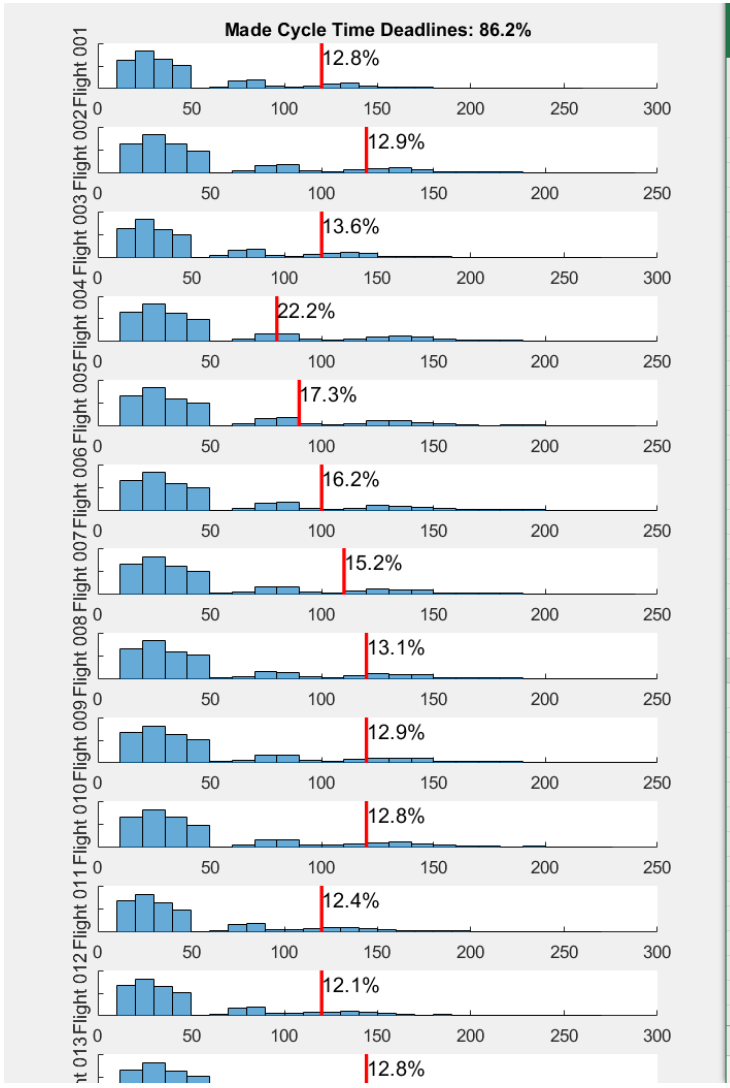
Results: Air Cargo Sort Hub Demo

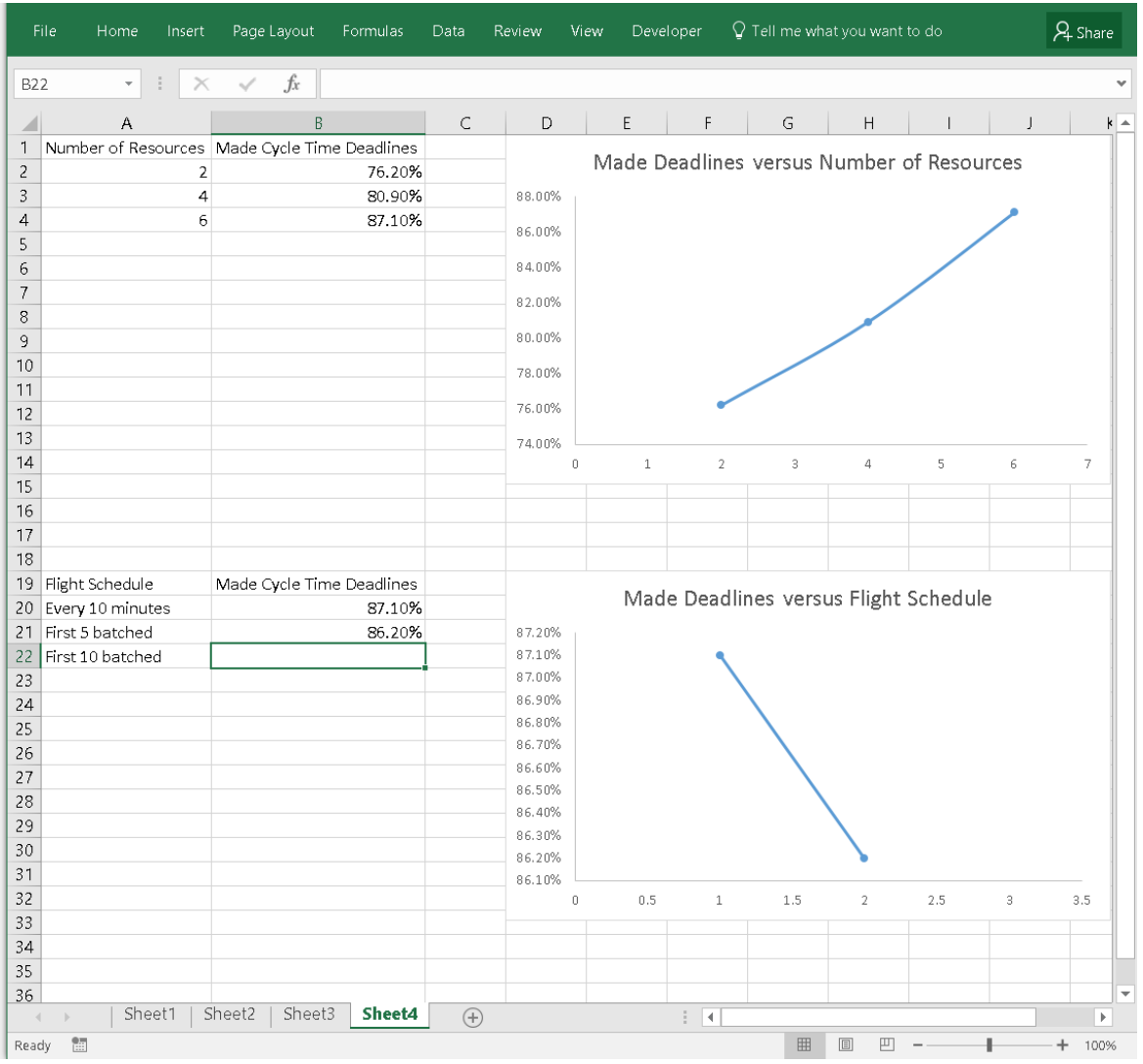


Results: Air Cargo Sort Hub Demo

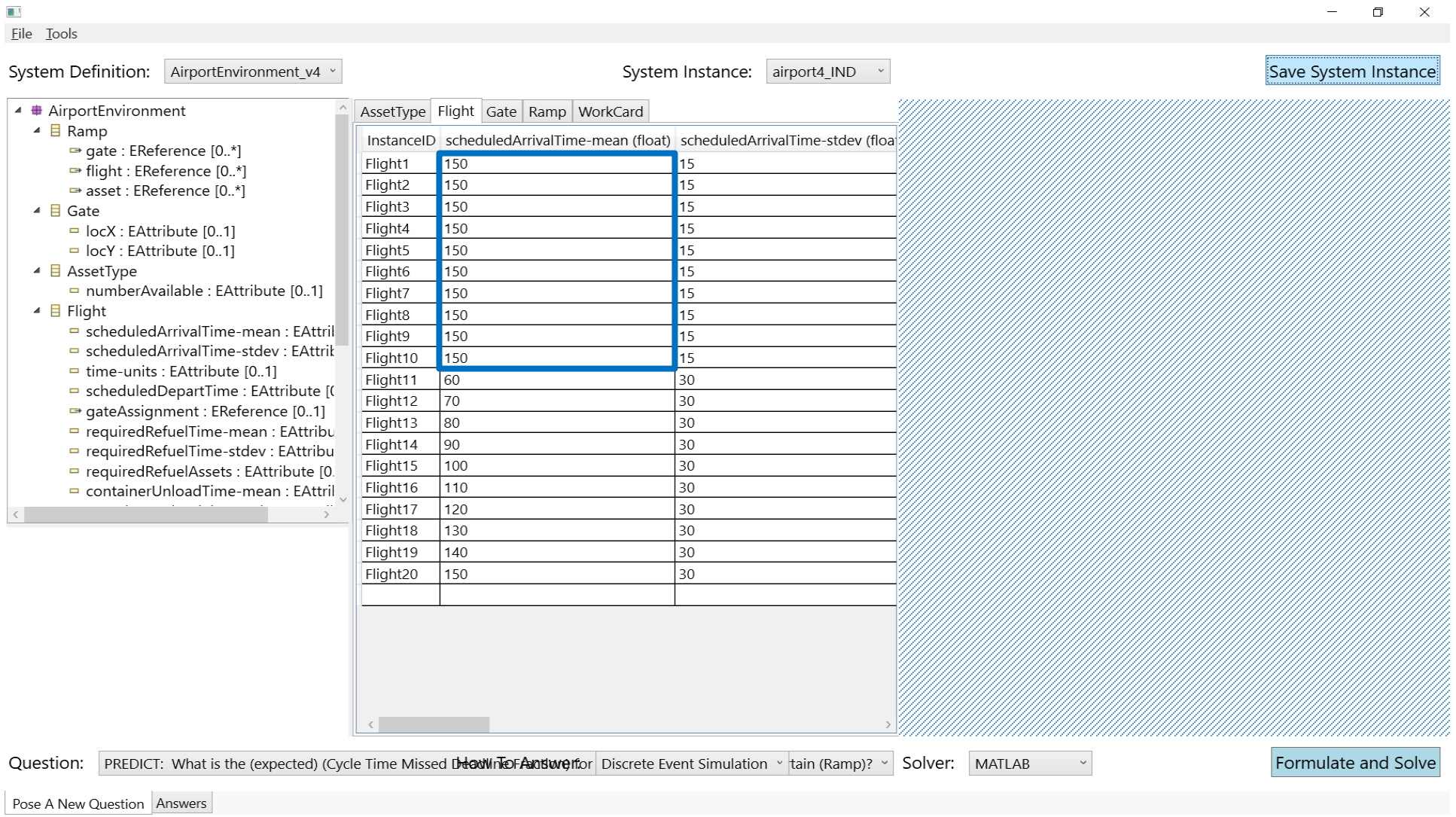


Results: Air Cargo Sort Hub Demo

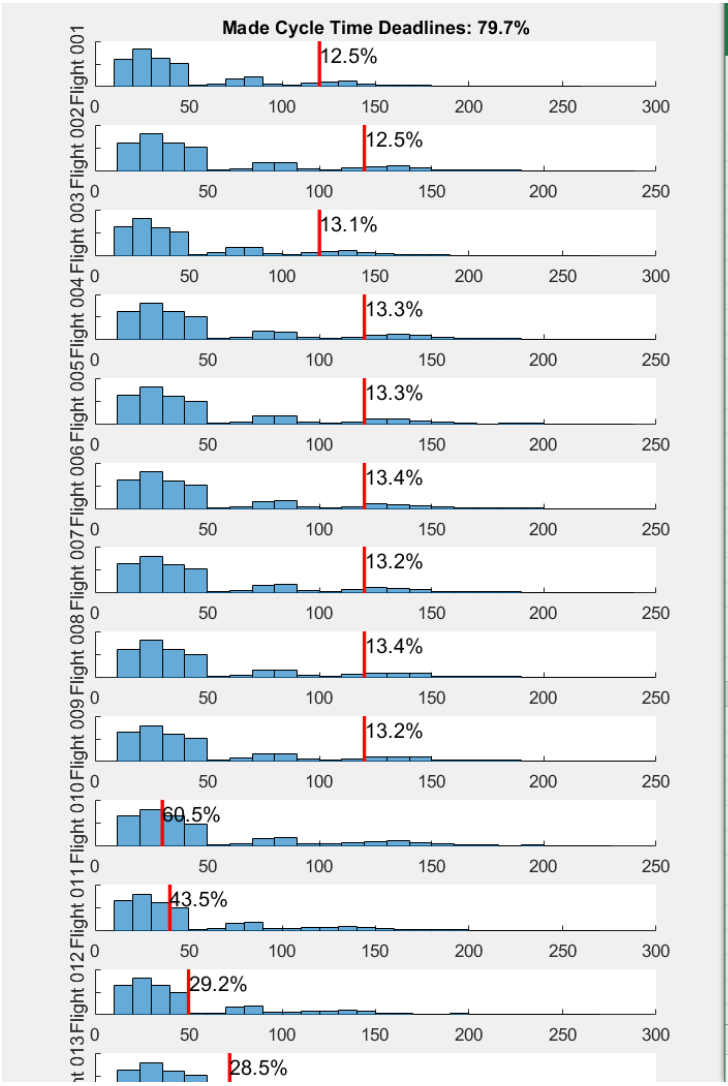


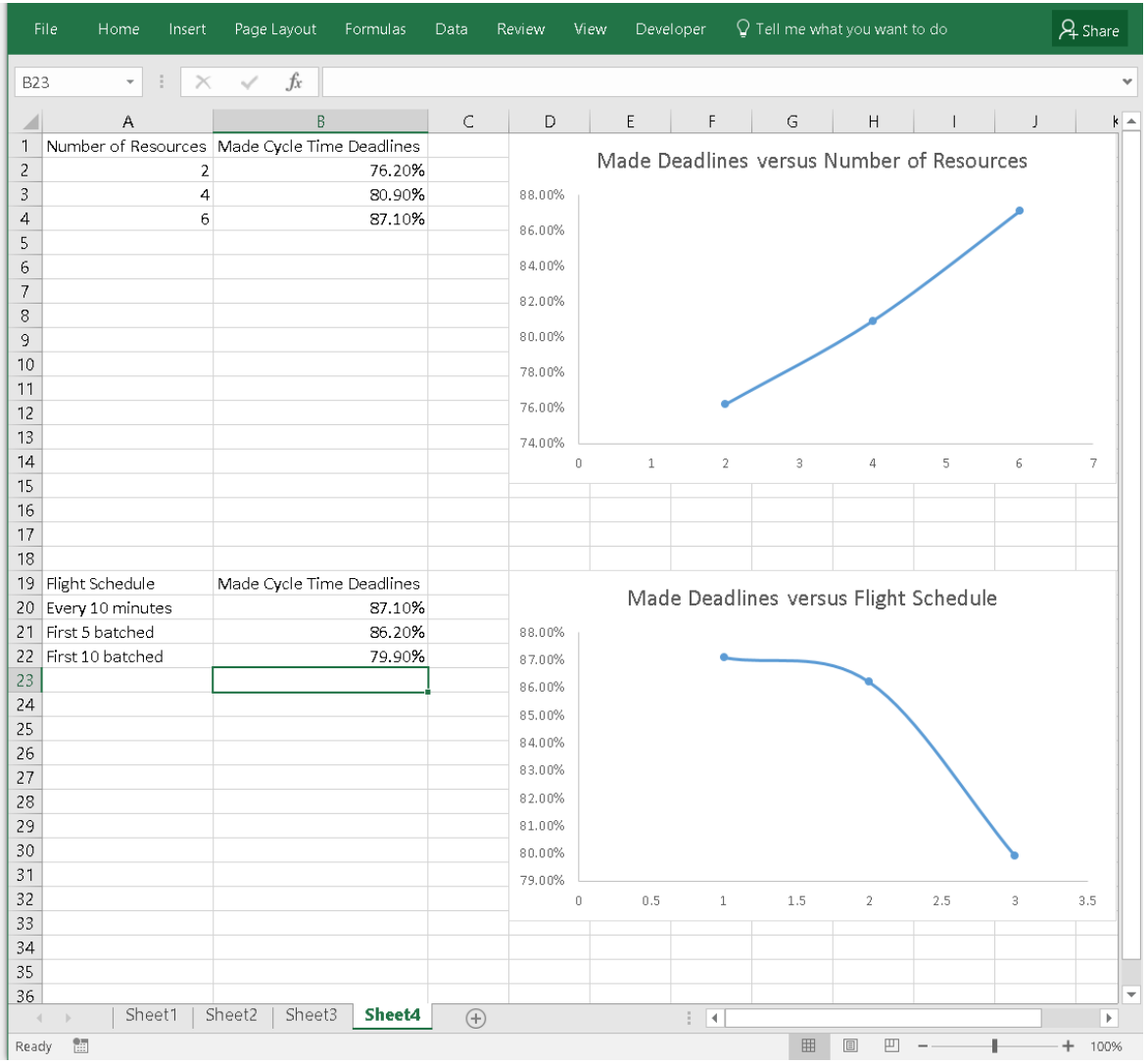


Results: Air Cargo Sort Hub Demo

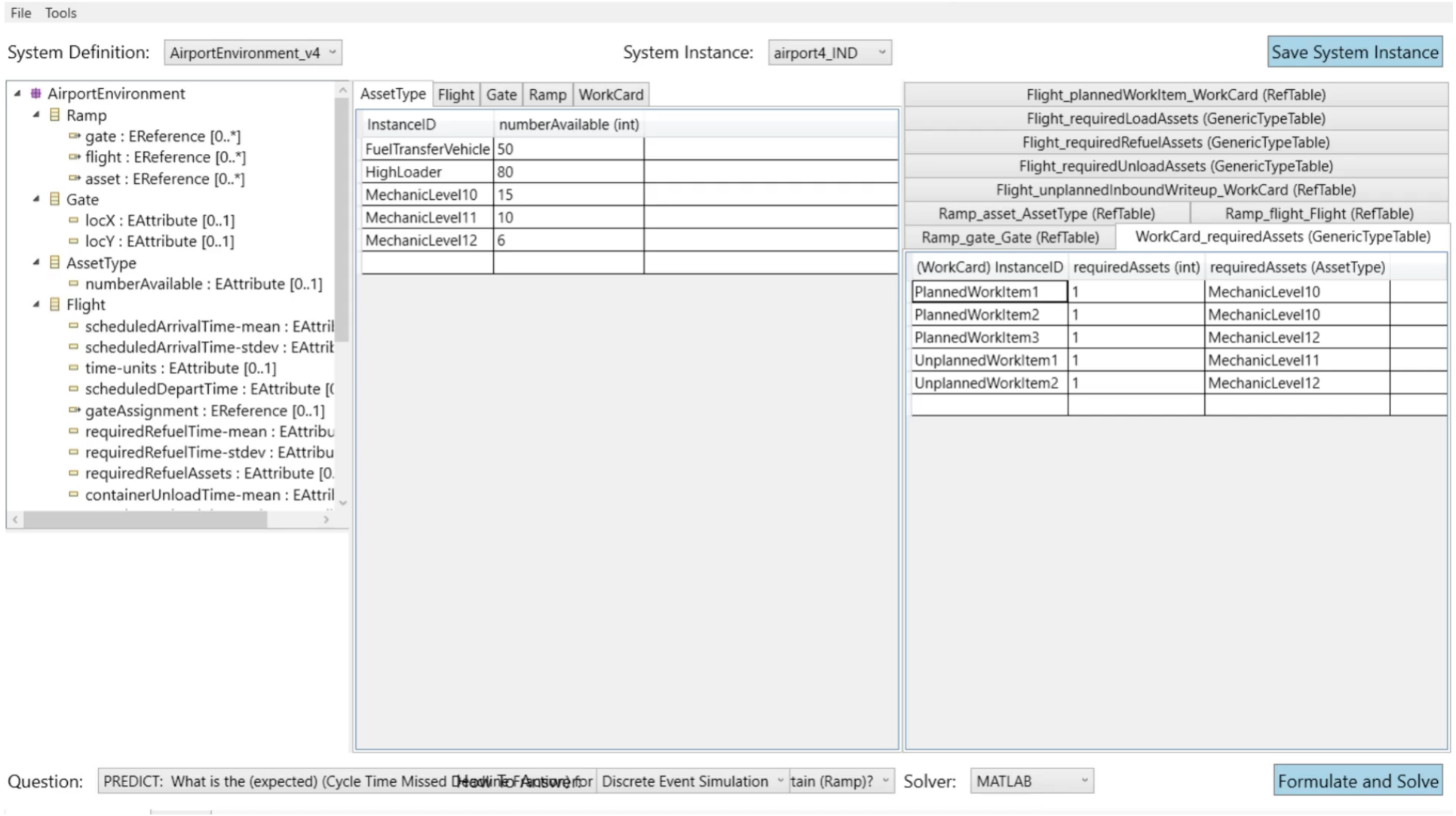


Results: Air Cargo Sort Hub Demo

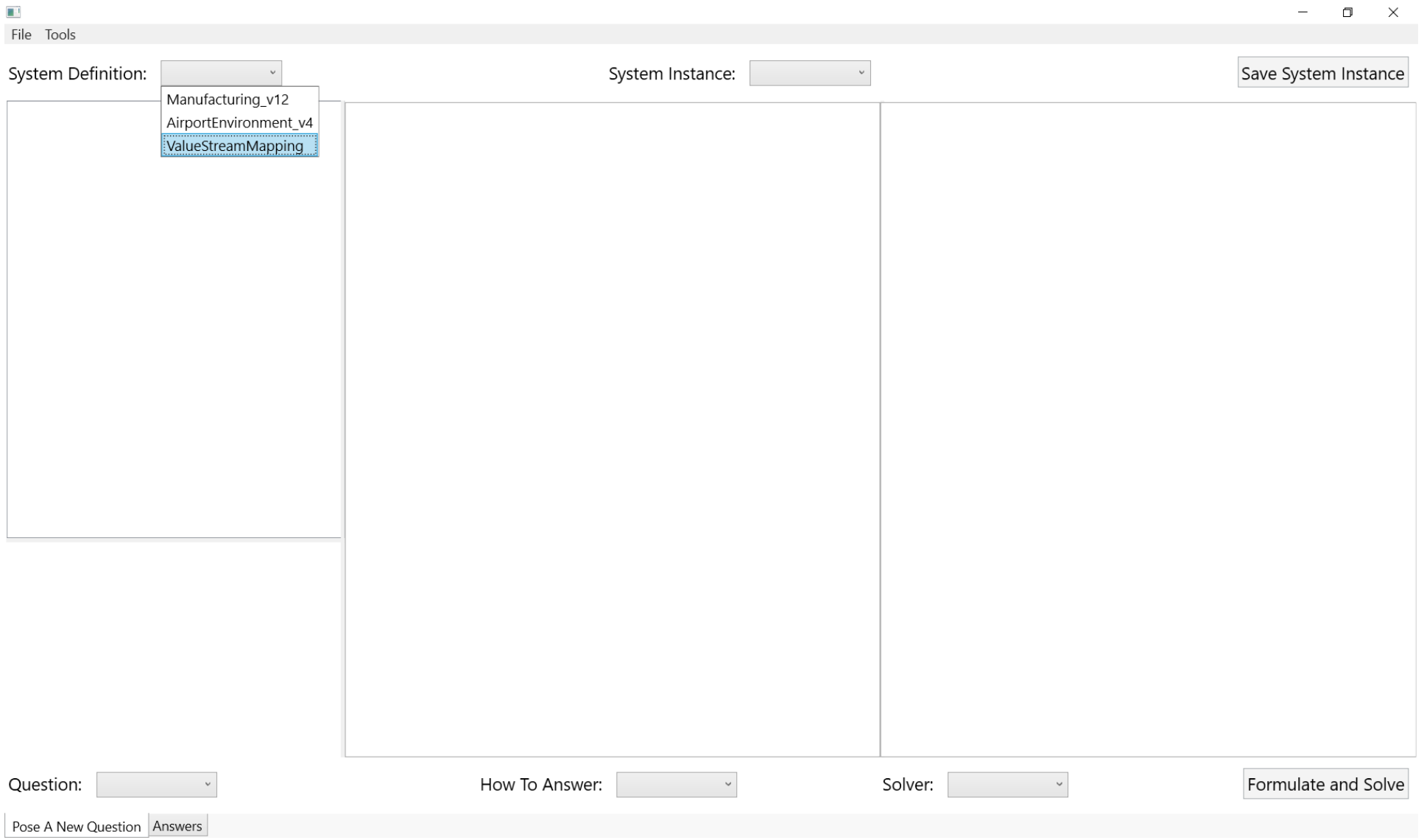




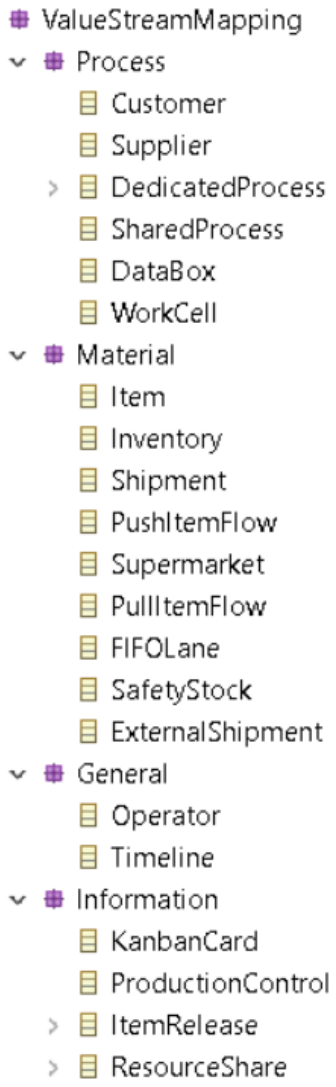
Results: Air Cargo Sort Hub Demo



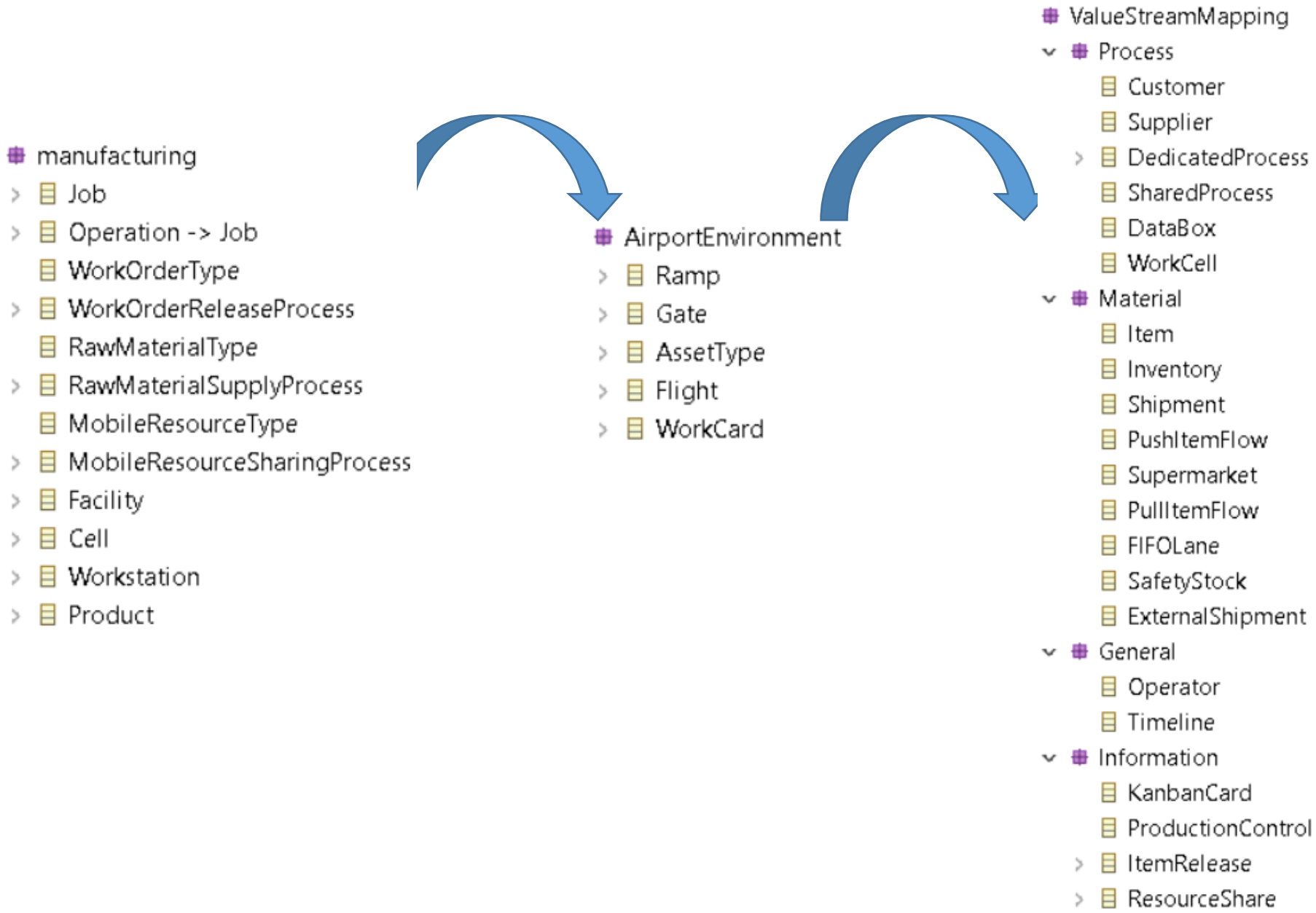
Results: Value Stream Mapping Demo



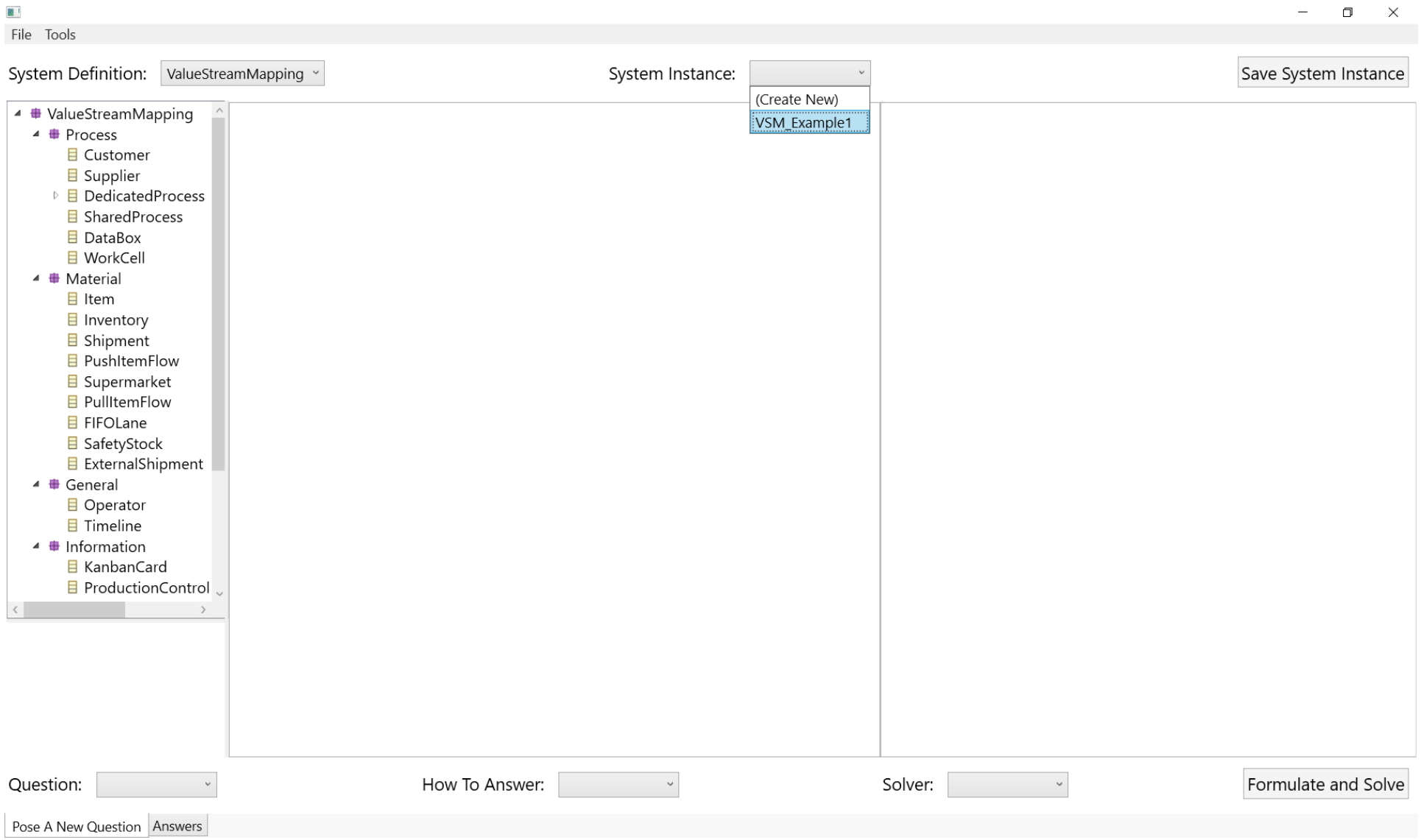
Results: Value Stream Mapping Demo



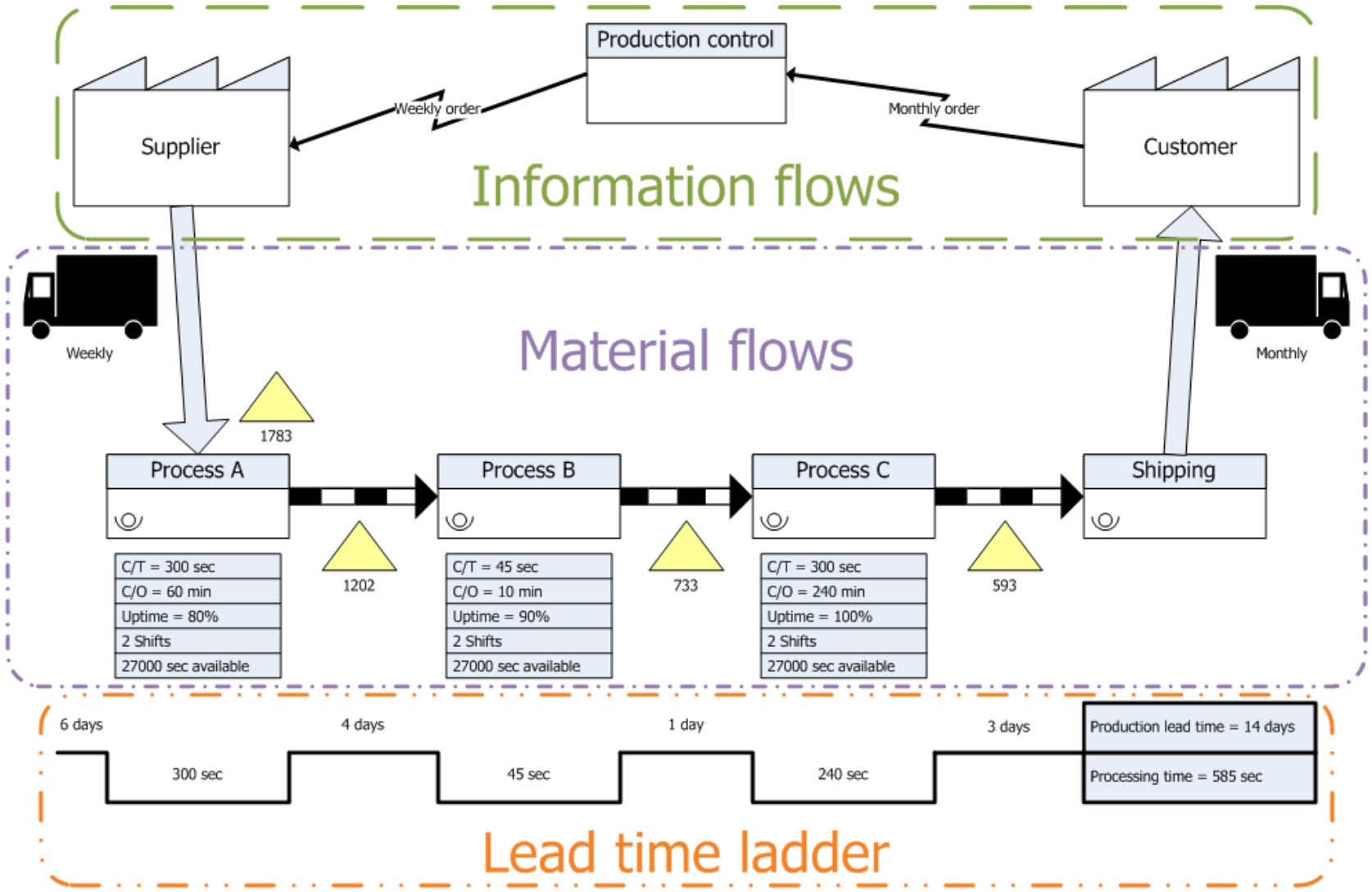
Complete Domain Switch, Same Analysis Generators



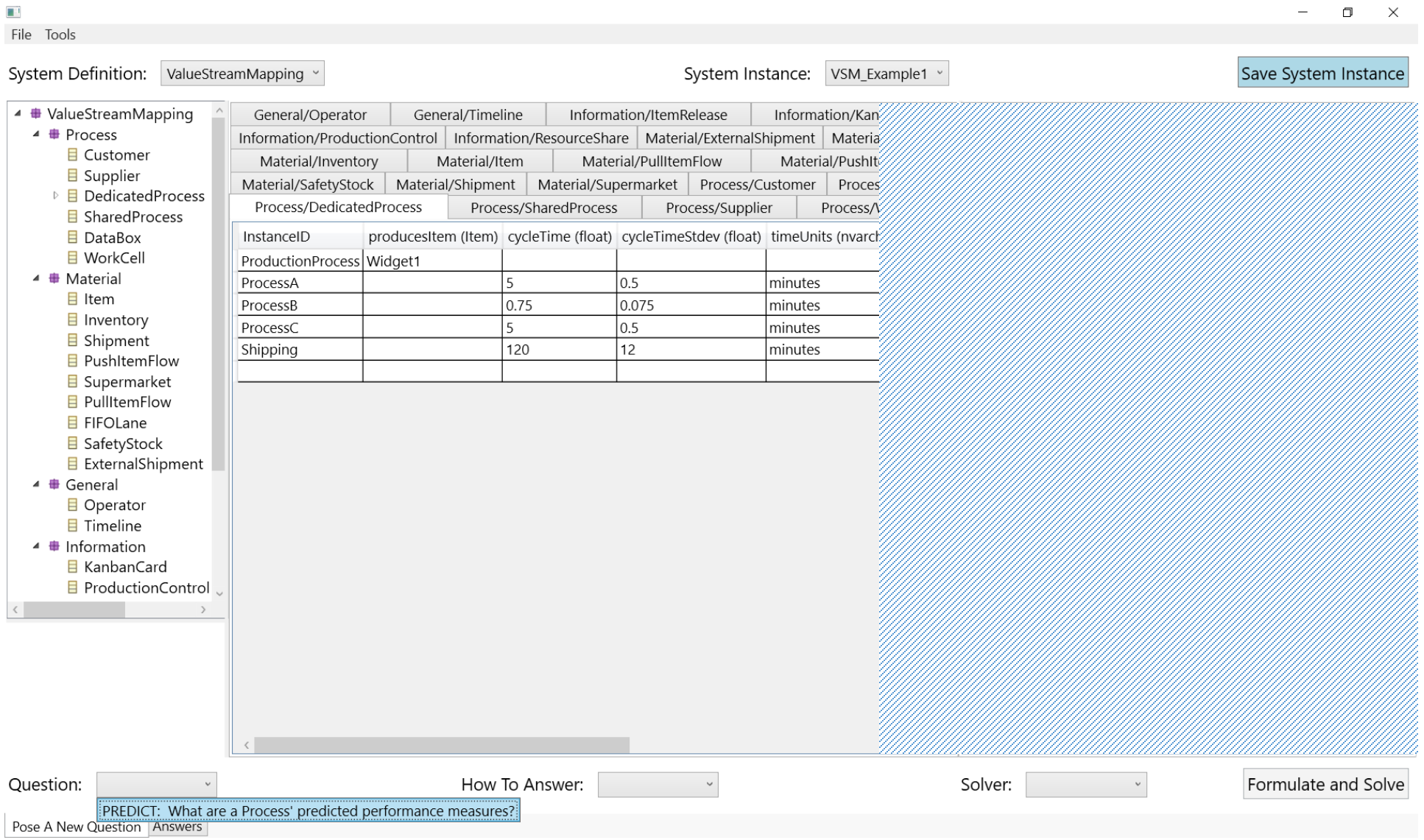
Results: Value Stream Mapping Demo



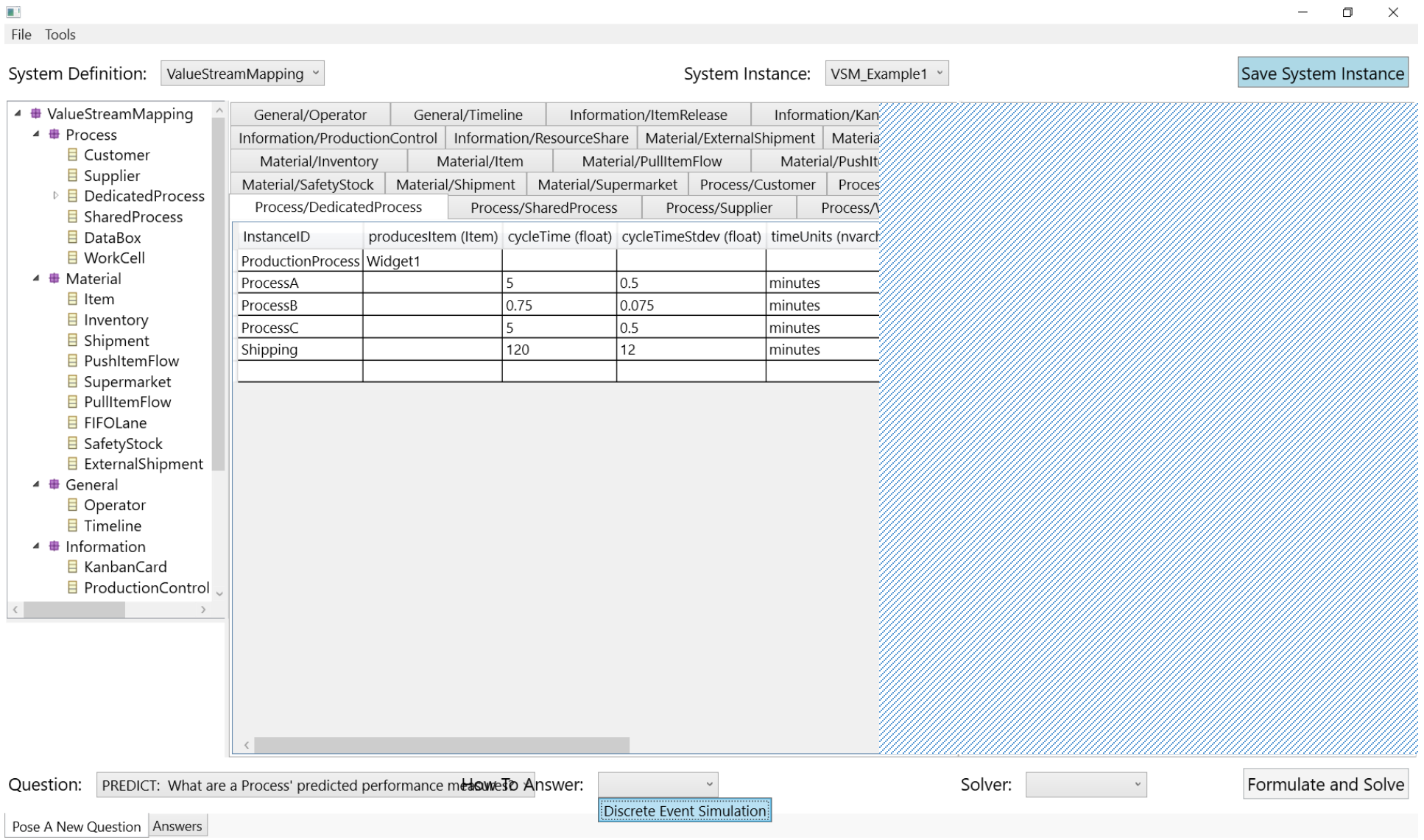
Results: Value Stream Mapping Demo



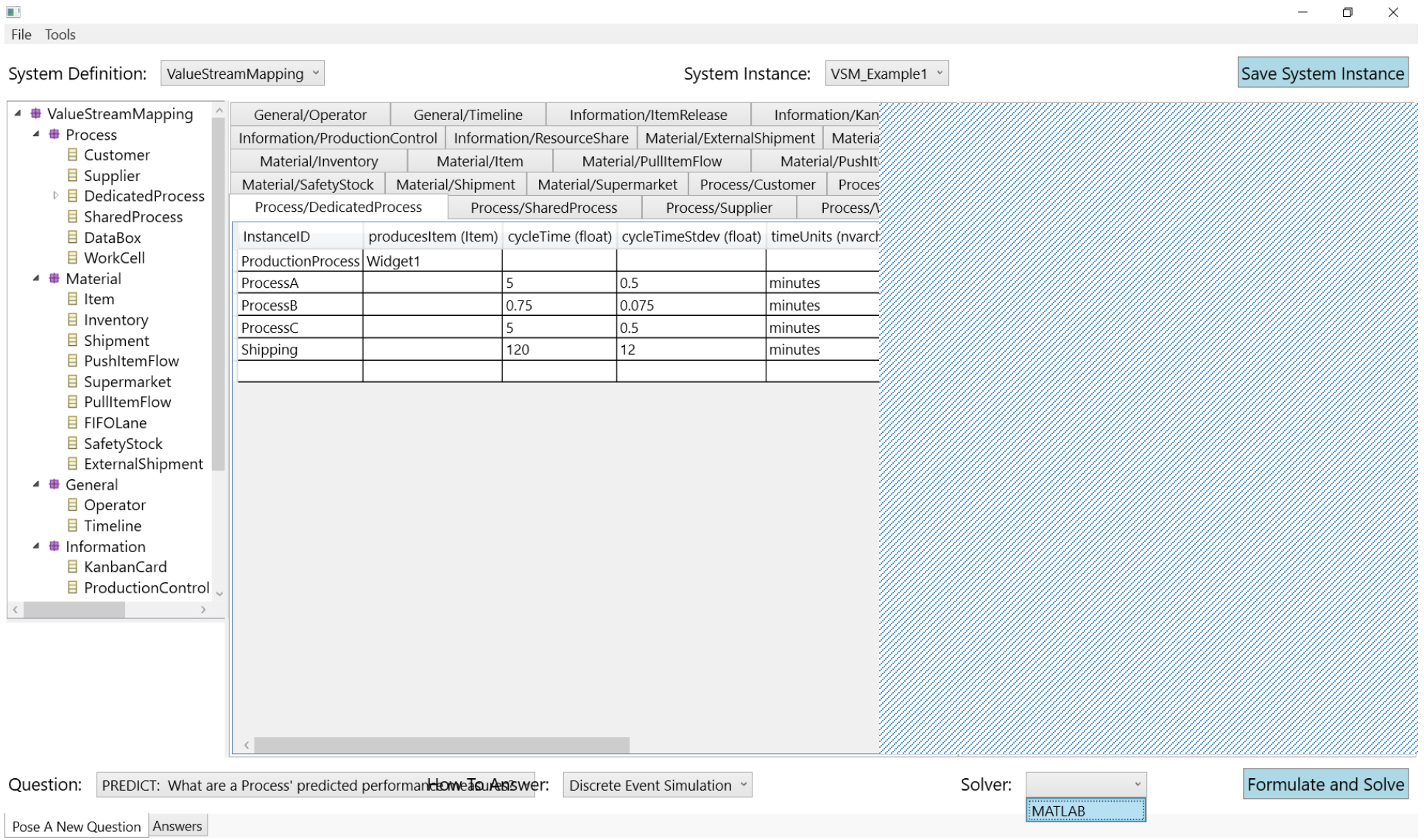
Results: Value Stream Mapping Demo



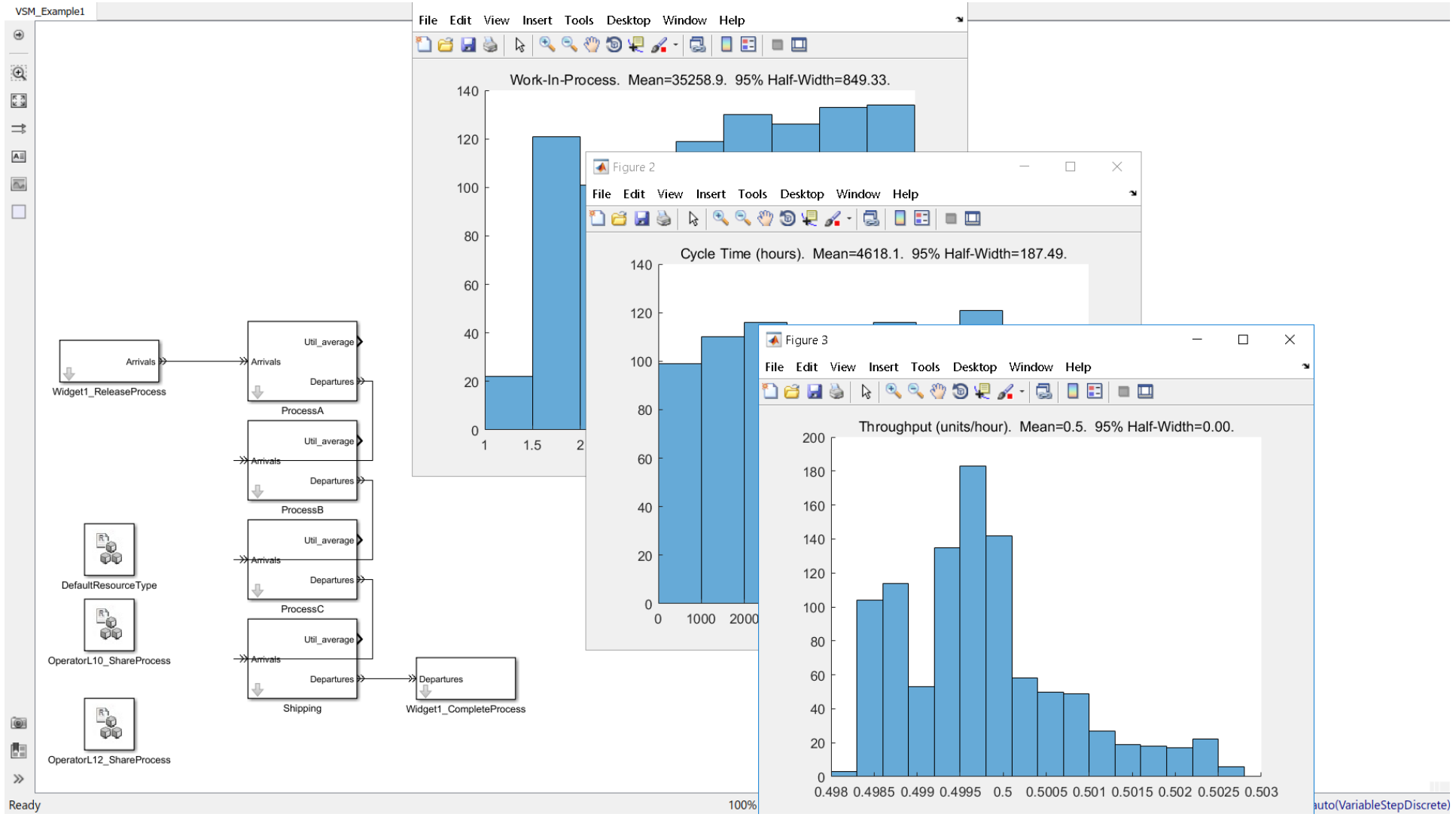
Results: Value Stream Mapping Demo



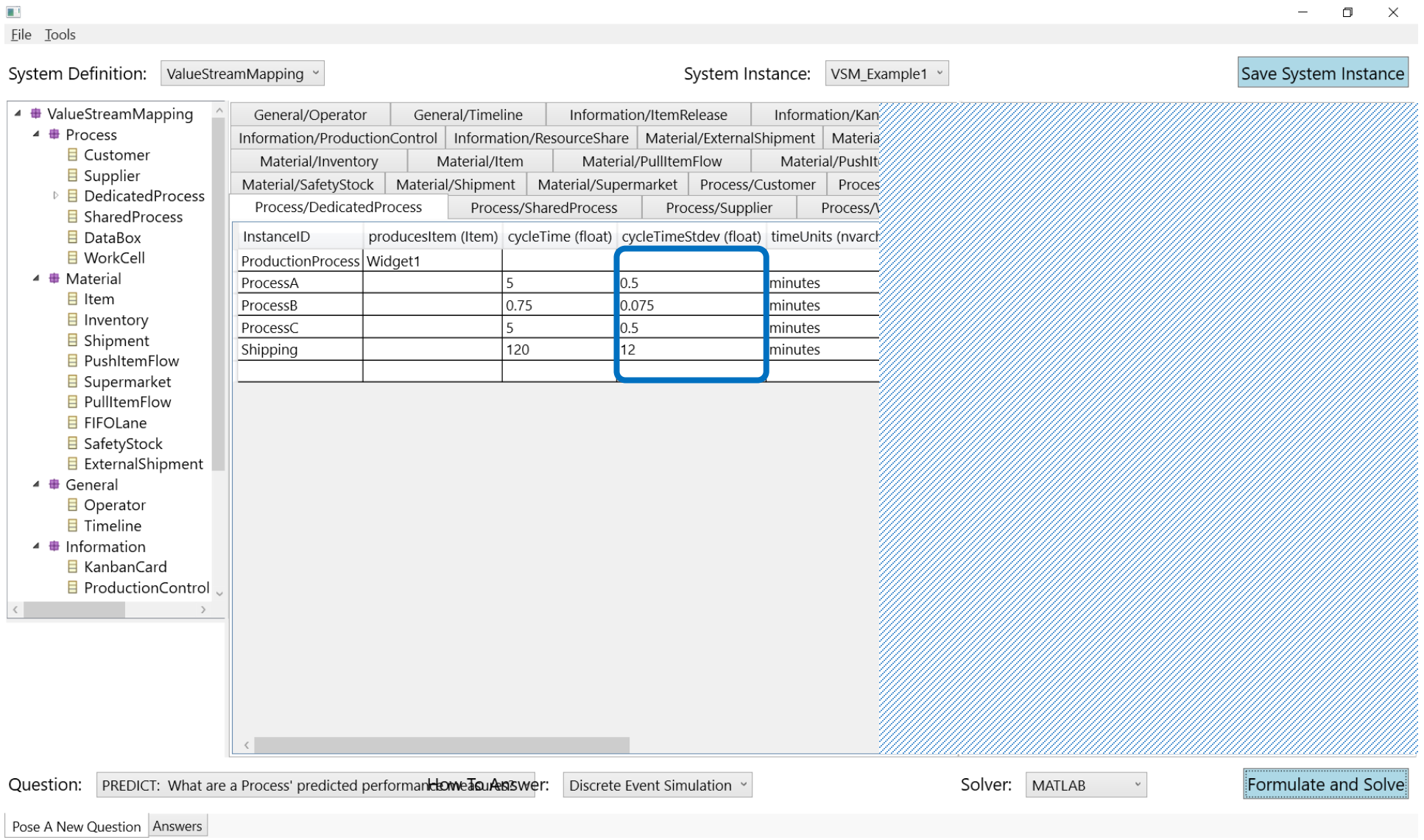
Results: Value Stream Mapping Demo



Results: Value Stream Mapping Demo



Results: Value Stream Mapping Demo



Summary

- Industrial Engineering is a domain that can benefit from improved tool support, we're creating it, and MATLAB and Simulink are an integral part of the solution.
  - ***Minimum***: MATLAB, Simulink, SimEvents.
  - ***Recommended***: Statistics Toolbox, Stateflow, Parallel Computing Toolbox.
  - ***Advanced***: Global Optimization Toolbox.

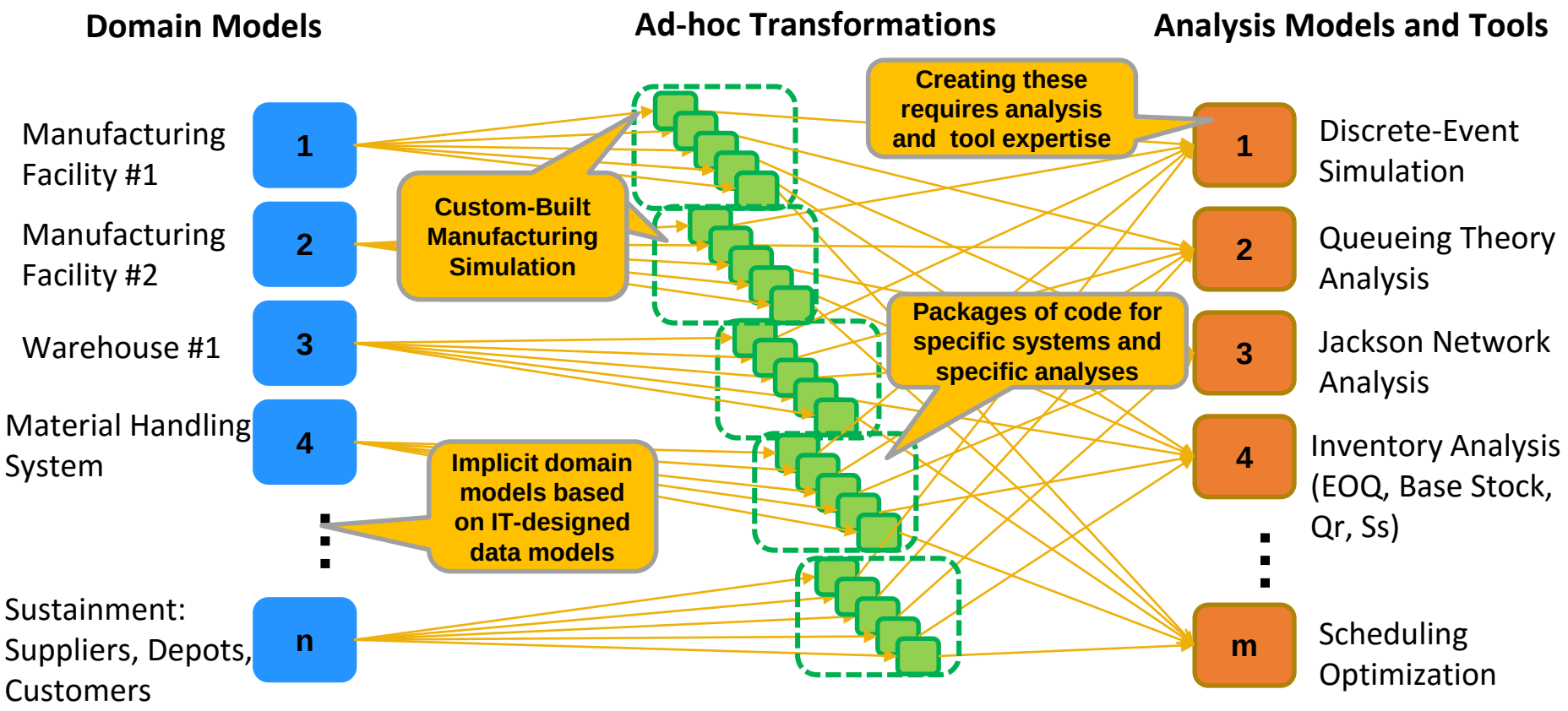
- Our contribution is Industrial Engineering domain knowledge, and expertise with modeling and model transformations. The MathWorks' contribution is analysis languages, solvers, and interpretation and visualization of results.

- Should you expect an "Evaluate Produceability" button to appear in your CAD tools in the next few months? This tool is a big step in that direction, although other puzzle pieces are needed to make it truly push-button.

Q & A

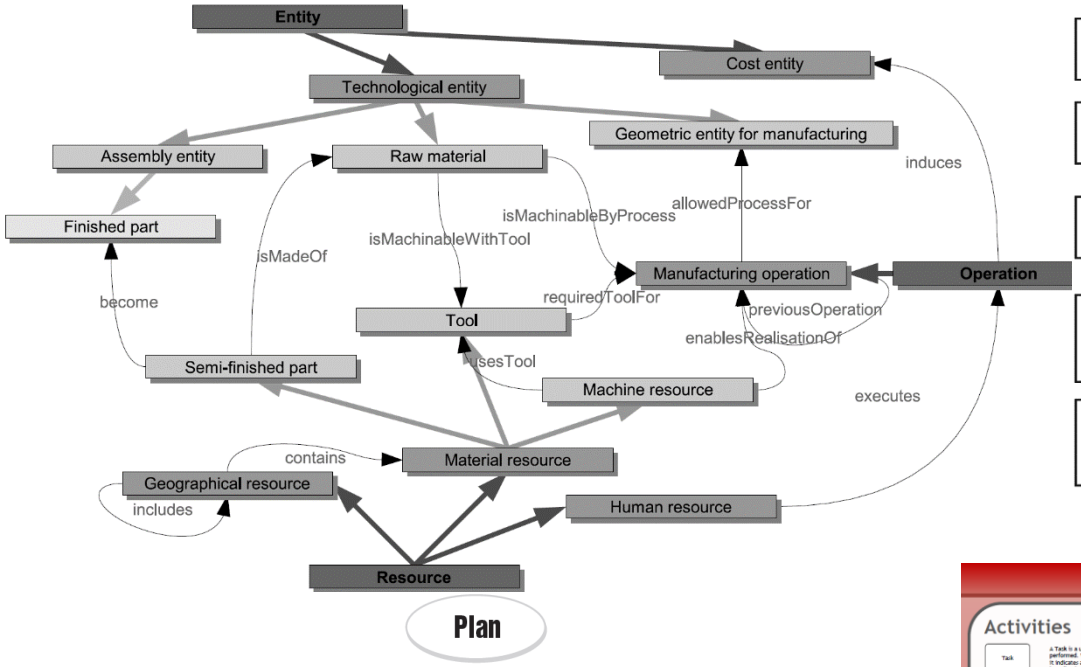
Backup

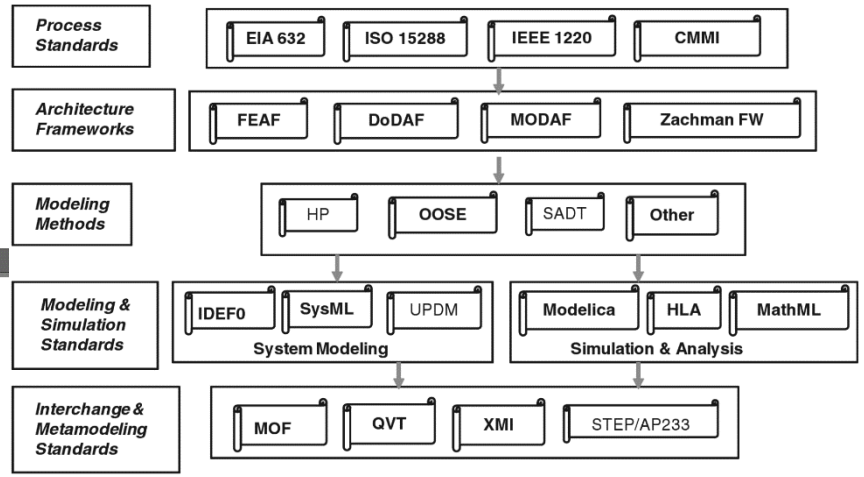
Status Quo of Automation: Manual and Ad-Hoc

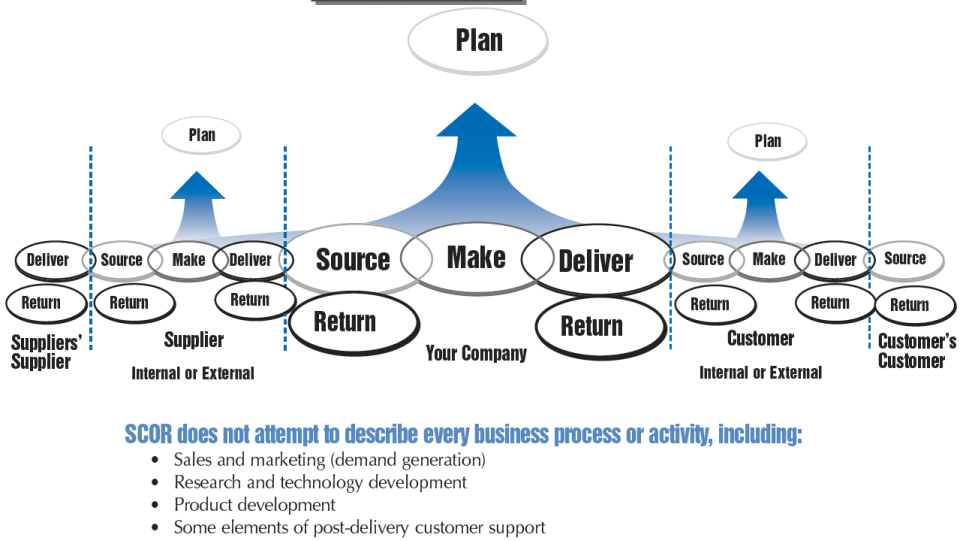


In the status quo, analysts commonly **hand-build custom analysis** to answer specific questions about specific systems. Automation can be added to make the formulations repeatable, but the issue remains that there is a unique transformation for every (domain, analysis) pair, severely limiting ROI of writing and maintaining each one.

We spent years searching for a perfect Industrial Engineering system model:

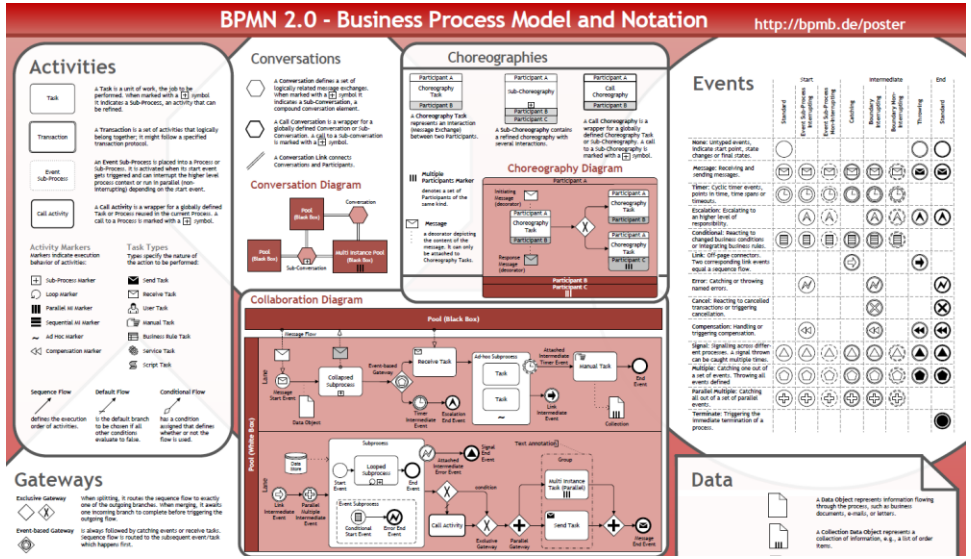






SCOR does not attempt to describe every business process or activity, including:

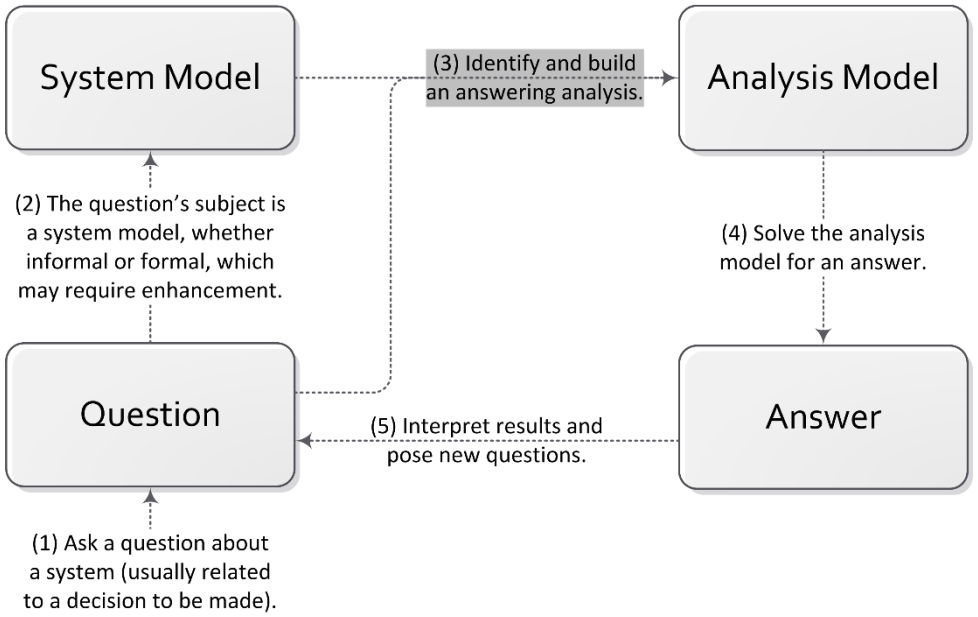
- Sales and marketing (demand generation)
- Research and technology development
- Product development
- Some elements of post-delivery customer support



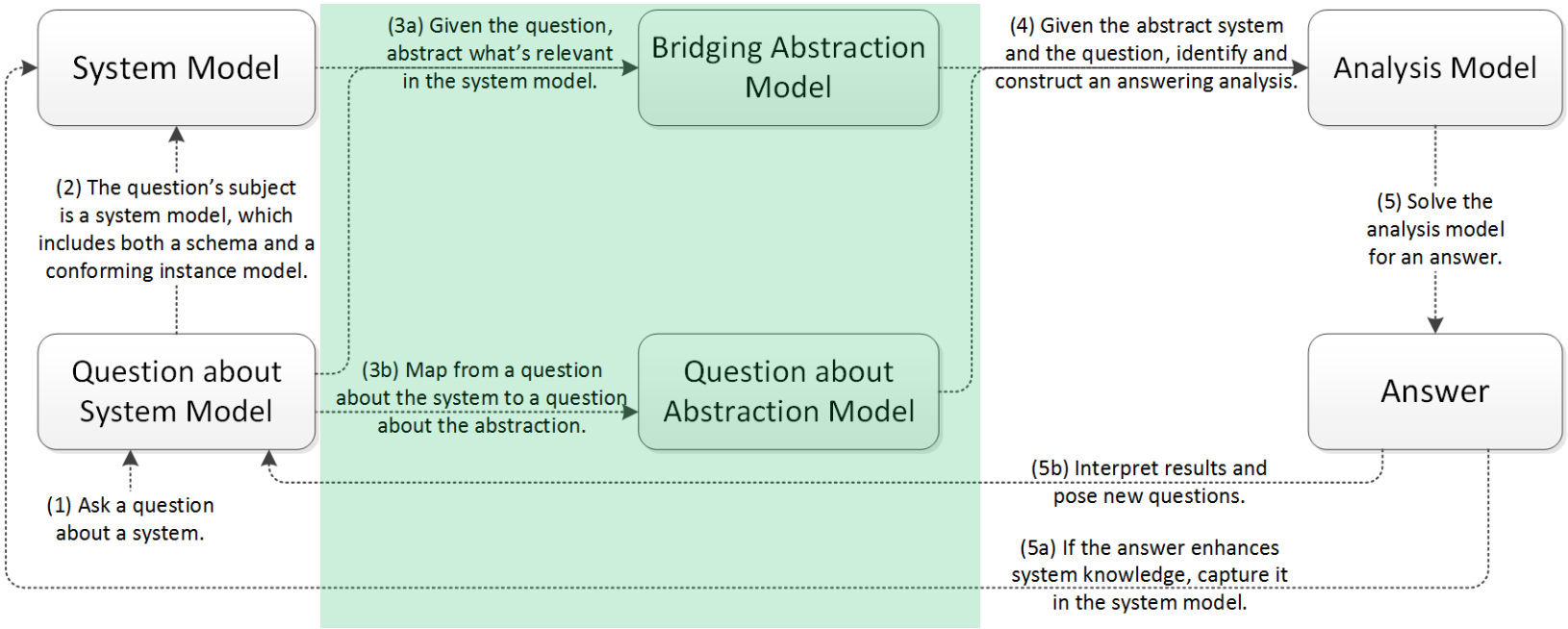
How to make a model robust and reusable? **Make it abstract.**

How to make a model user-friendly and accessible? **Make it concrete.**

The Solution



The Solution



MBSE for Industrial Engineering (versus Mechanical & Aerospace)

- The behavioral model is fundamentally different.
- We care a lot more about instance data. An aircraft or satellite has a controlled number of subsystems (which can each be quite complex; it's the number we care about) - one avionics system, one guidance and control system, one power system, ... A manufacturing system may have dozens of facility instances, hundreds of process plan instances, thousands of workstations, and tens of thousands of resource instances.

Issues with Process Modeling in SysML

- The subset of the language for activity modeling is very abstract, similar to the *Process Network* definition in our back-end bridging abstraction model. However, we never intended end-users to author directly in our bridging abstraction model, just map to it.
- There’s no facility for user customization, e.g. define a language within the language, as SysML structural modeling permits.
- It stores information at the wrong levels of abstraction. A specific process structure is stored at schema-level, and also elements' parameterization are stored at the schema-level, for example actions' token types and quantities input & output.
- SysML has limited capabilities for modeling structure-behavior integration. In version 1.3, a *Block*'s links to behavior are *Classifier Behavior*, *Owned Behavior*, and *Owned Operation*. How to express capability, assignment (both static and time-varying, the latter sequencing and scheduling)?