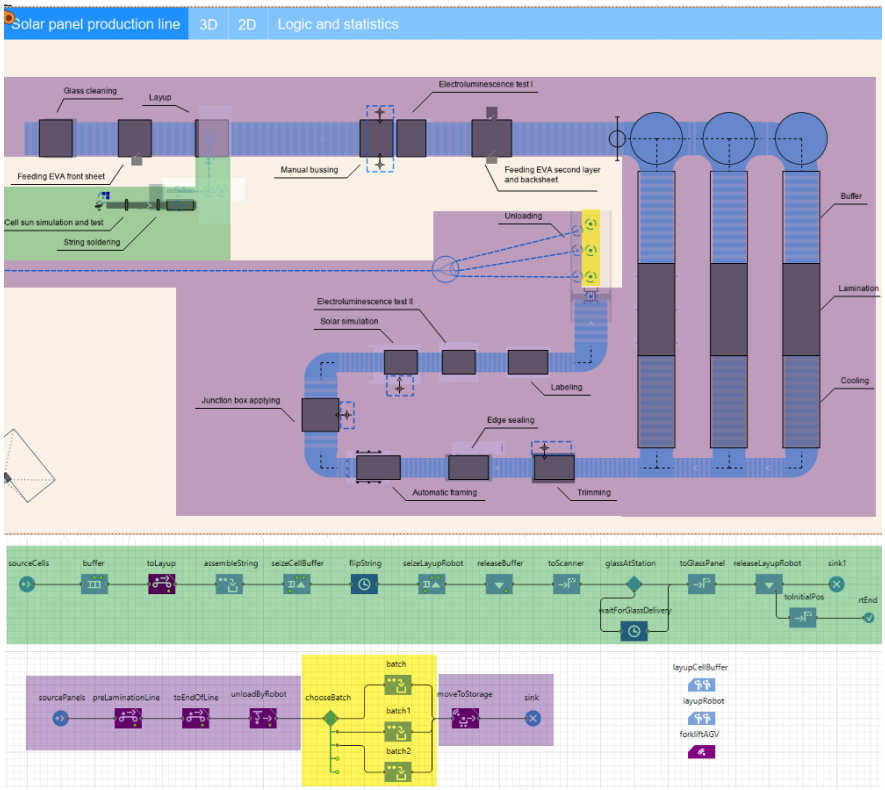
Q & A

Could you talk about a little on the advantages of using MHL [Material Handling Library] over PML [Process Modeling Library]?

- The MHL is specifically designed for being able to model factory layouts, workshop processes, and transportation routes – all aspects where the physical-level space is more important and more involved than what you would typically model using the PML.
- Initially, there were some physical-level elements added to PML, such as the Pallet Rack. However, as more use cases were found that required more complex behaviors specific to that area, these have become improved upon in the MHL's Storage object. These PML features are now considered "legacy" features and it's advised to use the MHL equivalent.
- It should be noted that these two libraries are ***not*** independent from one another.
    - One good example is the Solar Panel Production Line (runnable online here). The annotated screenshot below shows the correspondence between the logic blocks and the layout. PML is used to describe the process-related complexities in seizing/releasing, delays, batching, and simple movements in areas where the physical-level aspects are less important. MHL is used for describing the more complex movement in the conveyor system and involving the crane; it's able to do so in less blocks, as the process is much more simple and the machine times are defined within the physical level objects.

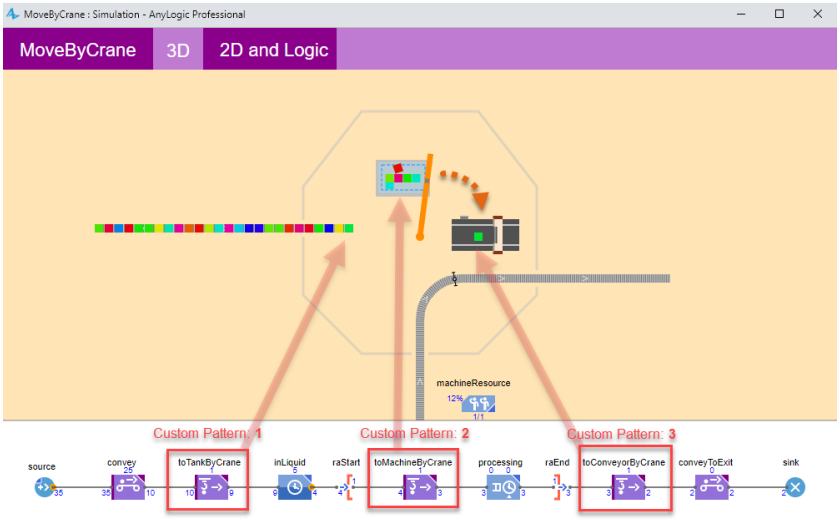


If we map an already existing factory layout or supply chain in AnyLogic, does the software suggest the ways to optimize the factory layout or supply chain?

- Due to all the complexities involved in any operation and all of the differing priorities of any given metric, AnyLogic does not make any out-of-the-box assumptions about what it means to "optimize" the existing operation.
- Instead, AnyLogic's focus is on giving you the tools to be able to create any operation without any hard limits and in a way that can be both elegant and flexible. Using the various data-gathering objects and ways to record model events, you can setup experiments to help you make justified decisions in how you can optimize your operation as you see fit.

In the first crane [MoveByCrane] example, how was the throughput at the end of the process prioritized?

- Within the MoveByCrane block – and also the SeizeCrane block – there is an option to specify whether the agent selection order is FIFO (First In First Out) or a custom pattern. Selecting the latter will expose a field, called "Custom pattern", allowing you to specify a numerical weight to the given agent, with higher values corresponding to higher priority. Whenever the crane is free, it will choose the currently waiting agent with the highest priority.
- Note: as this field is both dynamic and gives access to a local variable for the requesting agent, you can have the priority change over time or even be based on the agent itself!
- The image below shows an example where there are waiting agents in all 3 parts of the operation and crane is moving to the final movement operation, as it has the highest priority.



- Relevant help articles: [MoveByCrane block]

Can the crane speed be dependent on the agents being moved as well as the progress along the path?

- Yes! For being dependent on the agent, the speed for all cranes is a dynamic field that gives you access to the agent and whether the agent is loaded on the crane. These update at all major action moments for the crane.
- Specific to the progress along a path or passing over a certain area, you can use the built-in functions for querying the crane's position and updating its speed. For example, you may have a cyclic event that checks the crane's position relative to a polygonal node; this is made even easier by rectangular and polygonal node having a `contains` function – returning true or false whether a given XY point is within its bounds.
- Note: The "Speed and access restrictions" within the properties of a node are specific to transporters.
- Relevant help articles: Crane, Rectangular node and Polygonal node

Can transporters translate in two axes simultaneously (e.g., forward and up)? Or only axis at a time?

- For context: as seen in the ASRS model, or any model using the Store and Retrieve blocks in conjunction with Storage objects, transporters can raise and lower to meet the height of the cell.
- Transporters, however, do not currently have a toggleable option between for step-by-step or concurrent movements like Cranes objects. If this has a significant effect on your logic, one possible manual implementation is to adjust the lifting or raising time to account for this.