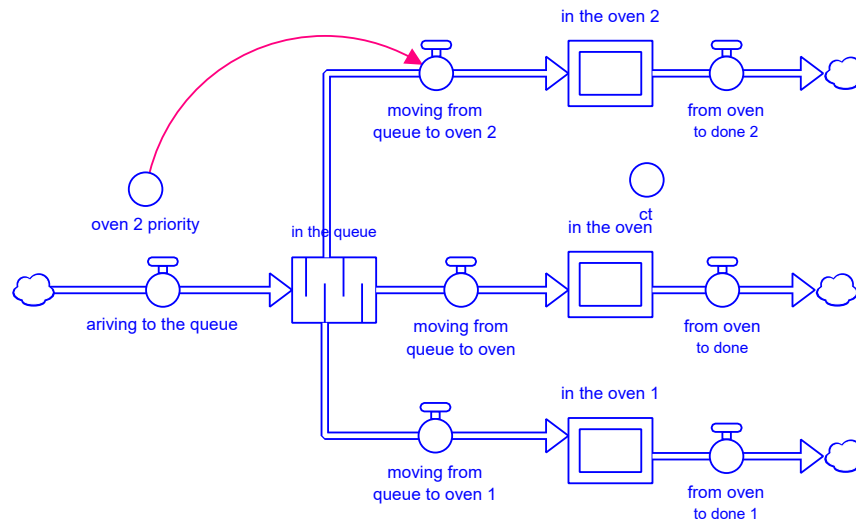


# Discrete Batch Processing



with Stella<sup>®</sup>

Presented by Bob Eberlein --- Jun 20, 2019

# Today's Topics

- Buckets and batches (continuous/discrete representations)
  - Cycle time and average residence
  - Mixing metaphors
- Attributes and their tracking
  - Co-flows and their discrete counterpart
- Queueing and dispatch
  - Determining where work will be done
- Jumping the queue
  - Using the attribute to prioritize
- Dumping the queue
  - Removing older material
- Mixing process times in assembly lines
- Summary observations
- Q & A

# Old and New

- Will be covering existing functionality...
  - Queues, Ovens, Conveyors
  - Been around for a long time
- ... and new additions to 1.9
  - Attribute tracking
  - Round robin dispatching
  - Processing prioritization
  - Purging old material
- I will endeavor to remark on what is new
  - But to summarize...

## What's New in 1.9

- An attribute can be attached to flows
  - Very much like time stamping material for Cycle Time
  - Tracks through chains of stocks and flows
  - Can be measured downstream
  - Cycle Time measurements can be used
- Attributes and Time Stamps Always Track
  - On Conveyors, Queues and Ovens
  - Also non-negative stocks if Cycle-Time Integration
- Queue dispatch control
  - Using priorities for destinations
  - Round robin
- Enqueueing control
  - Allows queue jumping using attributes
- Queue purging based on time in process
- Conveyor updates
  - FIFO when transit times change
  - New inflow options
    - Immigration and emigration in demographic models
    - Disease progression and transitions in aging chains

## Continuous v. Discrete (batch.stmx)

- The stock and flow representation is a liquid metaphor
  - Stocks are tubs of liquid and flows are taps and drains
  - Each flow is made up of many bits that are not distinguished
    - H<sub>2</sub>O molecules
  - The stock gets those bits and they mix
    - Completely and perfectly (color, temperature...)
  - The outflow is a bit of the stock
    - All the same characteristics
- We often think of things in batches
  - Explicit in queues, conveyors and ovens
  - Implicit in our own view of delay processes
- Cycle time integration technique allows for this interpretation
  - No change to diagram
  - Just a different way of thinking about computation

Note – in this model we don't really need the bottom stock

## Attribute Tracking (attribute.stmx)

- Very similar to the batch model
- Top is a standard coflow structure
  - Total attribute based on:
    - Incoming attribute
    - Stock changes
  - Perfect mixing
    - Average of outflow is just average
- Bottom stock uses new attribute feature
  - Underlying stock is treated as FIFO
    - Cycle-time integraton method
- Continuous version slower to change
- Neither is really right
  - A leaking conveyor would have more fidelity
  - Could still track attributes the same way

## Queueing and Dispatch (OvensNormal.stmx)

- Material arrives for processing
  - Constant in this model, but typically random
    - Often POISSON
- Material enters a queue
  - A batch each DT
  - Queues are (usually) FIFO
- Available batches are dispatched to processors
  - Using the Oven element which is a simple batch processor
    - Capacity, cook time, fill time
  - Next available processors takes the batch
- By default the first connected target gets the top priority
  - Arrays act like a sequence of elements
  - Early processors get more, later processors less
    - Round robin changes this

# Dispatch Priority (OvensPrioritize.stmx)

- New feature in version 1.9
  - As is the round robin option
- Specifies which processor is considered next
  - Smaller number higher priority
    - 1<sup>st</sup>, 2<sup>nd</sup>, ...
    - Ordering is ordinal, but values need not be integers
  - Ovens most common
    - Capacity constrained conveyors can also be used
  - Filling ovens are given precedence
    - Priority used if more than one oven filling
      - Rare
- Round robin within a priority
  - If all priorities the same just round robin

# Processing Priority (PrioritizedProcessing.stmx)

- Allows some batches to skip to the front of the line
  - Or the middle
    - Ahead of those with bigger attribute value
  - New in 1.9
  - Queue is no longer FIFO
- Uses the attribute value to manage the skipping
  - Attribute takes on a slightly different meaning
    - The attribute is not really an attribute but a queue jumping priority
  - The attribute carries through
    - Prepping and baking in this case both do queue jumping
- Model comment
  - We ignore shelving
    - Could be treated as a stage transition time (stock, oven or conveyor)
    - More importantly might limit capacity – orders would pile up

## Purging Older Material (PrioritizedProcessing2.stmx)

- When material gets too old it may need to be removed
  - Spoilage if the material is perishable
  - Expedited processing if things are simply taking too long
- This is done by purging the queue based on cycle time
  - Time stamp upstream from the que
  - Material can be discarded
    - Purge flows do not process only 1 batch
- Purging is quite flexible
  - A variant of a conveyor can be constructed this way
    - PurgingConveyor.stmx
  - Conveyors and the FIFO variant will be covered in another Webinar

# Mixing Process Times

- Easy to manage different characteristics using arrays
  - But the array elements never mix
    - Hard to do fixed resource availability
  - Once merged, they can't be recovered
    - Lose track of the different elements
- Can use the array element to set an attribute
  - Mix the material into a consolidated processing chain
  - Break the material back out using ATTRCOUNT (or a Cycle-Time function)
- Can array a queue by type then mix during dispatch
  - MixedProcessArrayedQueue.stmx
  - The last array element suffers the most when capacity constrained
- Can consolidate into a single queue
  - MixedProcessSingleQueue.stmx
  - Note the two cascaded queues
    - Flows get their array dimensions from stocks
  - No preference by type when capacity constrained

## Discrete Stocks- Observations

- Discrete elements can add fidelity
  - When perfect mixing is not appropriate
- With 1.9 it is easier to mix and match discrete and continuous
  - Cycle time and attribute tracking can be anywhere
    - Little performance penalty
- Controlling material paths
  - Mapping gives the physical layout
  - Prioritizing queue inflows
  - Managing queue dispatch
  - Purging queues based on age
- Batches are still monolithic
  - Not composed of a number of elements
  - Can use arrays to break that down

# Questions?

## **More information:**

<https://iseesystems.com/help>

## **Upcoming Webinar Topics:**

Conveyors, Aging and Disease Progression  
Using Stella to Trace Causality

# Contact

## Support

[support@iseesystems.com](mailto:support@iseesystems.com)

Bob Eberlein

[beberlein@iseesystems.com](mailto:beberlein@iseesystems.com)