

Process Simulator Tips & Techniques

Instructor Info:

Rebecca Santos, Solutions Consultant
rdossantos@promodel.com

Bruce Gladwin, SVP Commercial Products
bgladwin@promodel.com



10/2019 Version 10.4.0.4145
Process Simulator Special Topics Webinar
For Software Version: 10.4.0.4145
Copyright © 2019 ProModel Corporation
705 E Timpanogos Pkwy
Orem, UT 84097
801-223-4600

Agenda

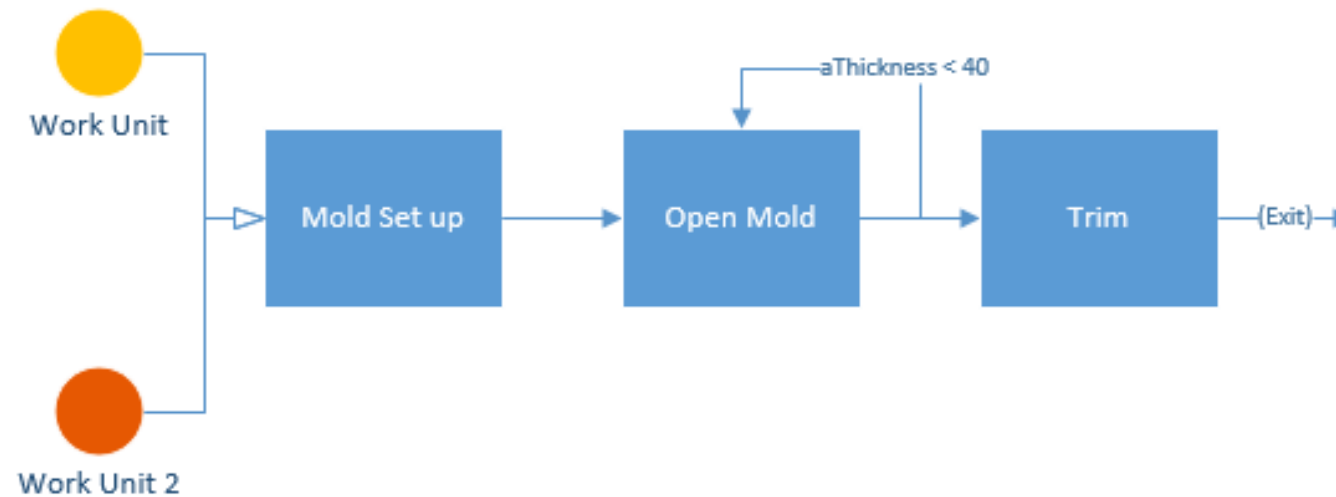
- Loops based on conditions
- Modeling Equipment & Facility as Resources
- Get vs. Jointly Get
- Using Priorities to Control Events
- Shift Carryover Functionality
- Default and Custom Batching Techniques

Loops

- Repetitive processes such those in composites manufacturing or complex rework loops may require the use of loops.
- Loops will repeat a block of logic over and over again until a condition is met.

Loops

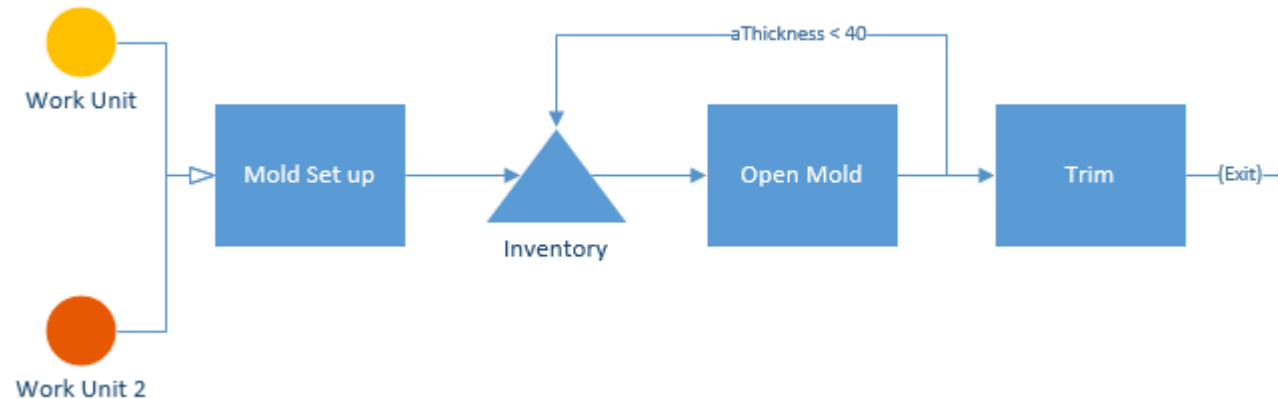
- Easy to lock up the model when a route comes out of an activity and goes back to the same activity.



Model File:
Loop 1.pspkg

Loops

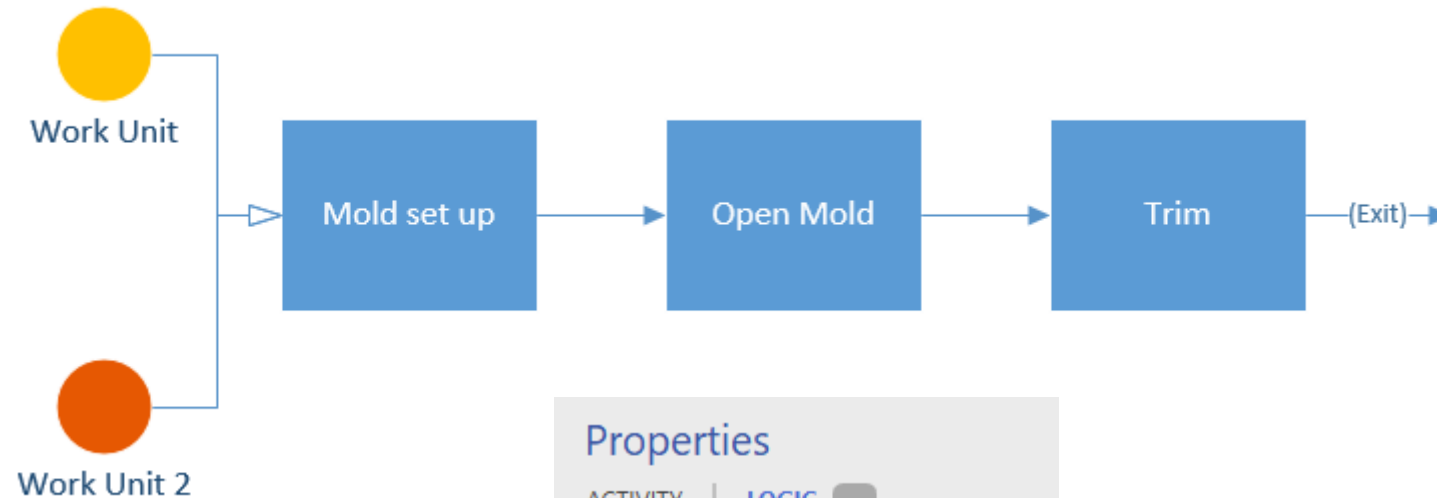
- A buffer can solve the model lock issue



Model File:
Loop 2.pspkg

Loops

- A While Do loop can simplify the model



Properties

ACTIVITY | LOGIC ...

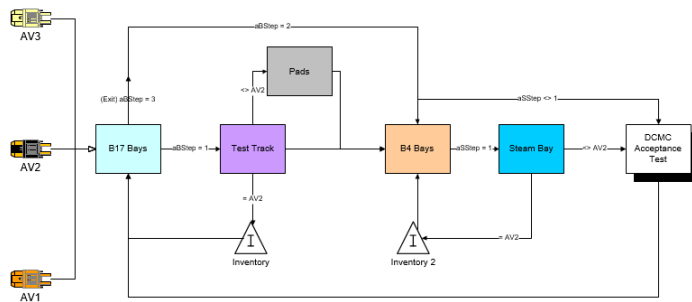
```
1 While aThickness < 40 Do
2 {
3     Wait 1 hr
4     inc aThickness, Rand(6)
5     Inc aLayer
6 }
```

Model File:
Loop 3.pspkg

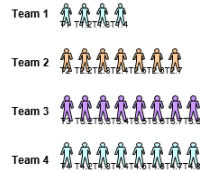
Activity Based vs Station Based

- Instead of modeling based on physical locations (Activity-based model) it can be beneficial in terms of complexity to create a model based on stations and model physical locations as resources

Armored Assault Vehicle
Garage & Test Track Operations

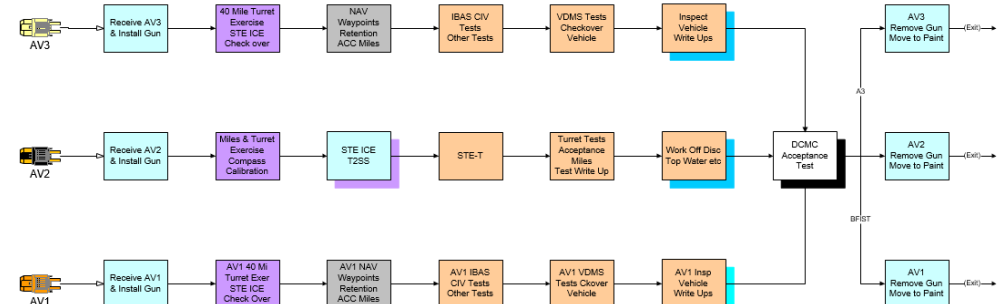


Labor Pool

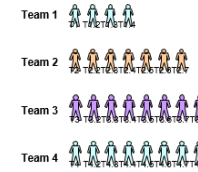


Process Performance Scoreboard			
	Complete	WIP	Cycle (Days)
AV3	0000	0000	0000
AV2	0000	0000	0000
AV1	0000	0000	0000
All	0000	0000	0000

Armored Assault Vehicle
Garage & Test Track Operations



Labor Pool



Facility Constraints



Process Performance Scoreboard			
	Complete	WIP	Cycle (Days)
AV3	0000	0000	0000
AV2	0000	0000	0000
AV1	0000	0000	0000
All	0000	0000	0000

Get vs. Jointly Get

Get Res1 and Res2

Vs.

Jointly Get Res1 and Res2

Resources are
captured
Immediately if
available

Resources are only
captured when all
resources are
available

Using Priorities to Control Events

- How and Where to Use Priorities
- Understanding Priority Levels for Resource Usage
- Entity & Downtime Control with Priorities

Where to Use Priorities

- Activity & Routing Dialogs
 - Priority for capturing a Resource to perform an Activity or make a Move
 - Dialogs allow Levels 1-10, plus Interrupt



Properties

ACTIVITY | LOGIC

Name: Prep_Entity

Capacity: 1

Buffers: In: 0 Out: 0

Time: 2 Min

Multi Entity: Define

Hourly Cost: 0

Resource

Name: Worker

Priority: 0

Keep:



Properties

ACTIVITY | LOGIC

Downtimes 1

+ Add - Delete

Unscheduled; Freq: 30 Min; Time: 15 Min;

Basis: Unscheduled

Frequency: 30 Min

First Time: Min

Interruptive:

Time: 15 Min

Resource:

Priority: 0

Disable:



Properties

ROUTE | LOGIC

Move Time: 1 Min

Resource

Name: Worker

Priority: 0

Keep:

Where to Use Priorities

- Activity & Routing Logic
 - Priority for capturing a Resource to perform an Activity or make a Move
 - Free Form Logic allows Levels 0-999



Properties ×

ACTIVITY | LOGIC ⋮

1 Get Worker, 299 Or Supervisor, 199



Properties ×

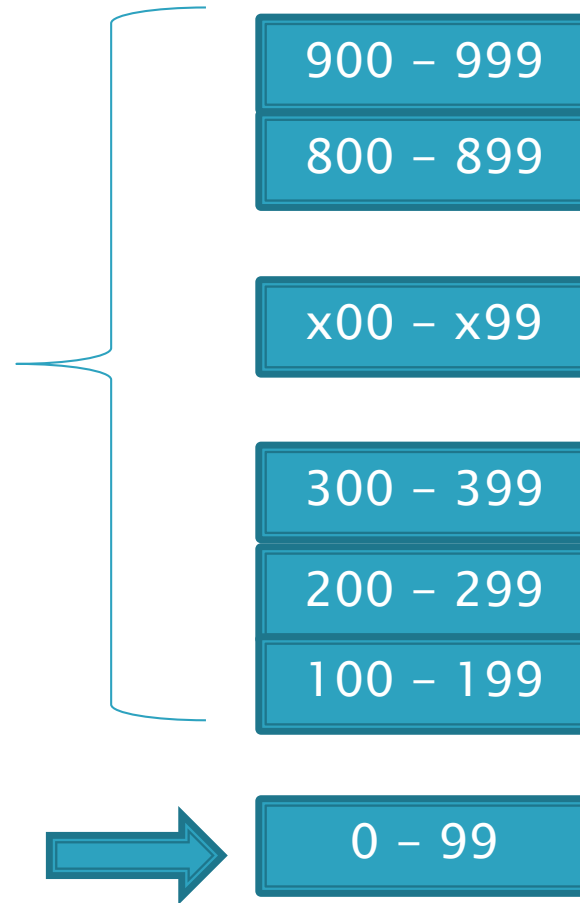
ROUTE | LOGIC ⋮

1 Use Worker, 50 For U(3,1) min

Understanding Priority Levels

Preemptive Priorities:
(i.e. can “takeover”
another event that
is currently in process)

Non-Preemptive Priorities:
Queue for processing
According to priority
Level, but cannot takeover
Another event already in
Process.



Queued entities with priorities that are at least 1 level higher than the current entity will preempt the entity that currently owns a resource.

In order to preempt a current downtime, a entity must have a priority level that is 2 levels higher than the DT priority.

Queued downtimes with priorities that are at least 1 level higher than the current entity will preempt the entity that currently owns a resource.

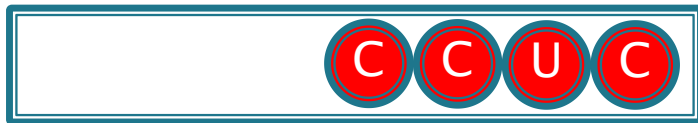
Events Queued for Resource Usage

Entity "Events" Queued for Inspector



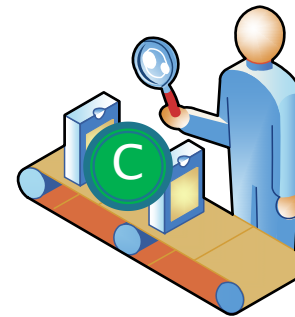
Default Entity Priority Level is 0

Downtime "Events" Queued to Occur



Default DT Priority Level is 99

Note: Interrupt Entity Priority = 100



Inspection

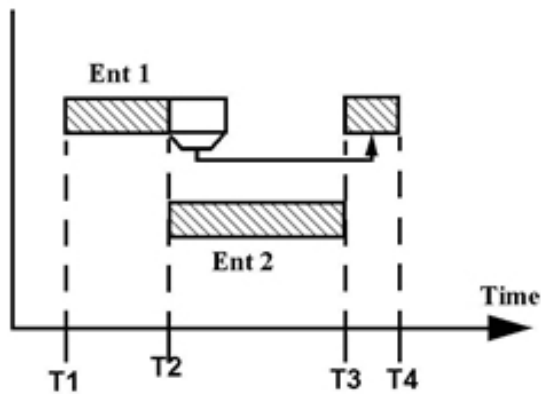
Departing Events



Once a departing event frees a resource, priority levels on queued entities and downtime events determine the next event to capture the resource.

Entities Preempting Other Events

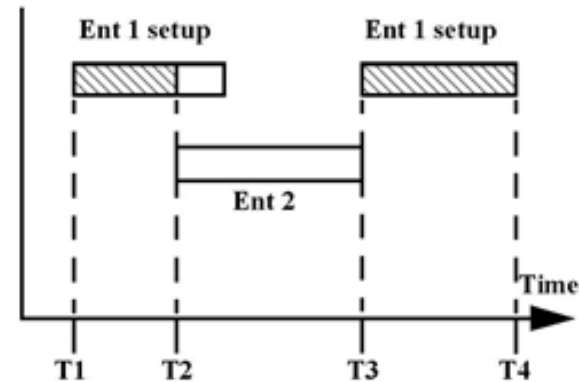
Entity Preempting an Entity



Ent 1 processing resumes upon completion of Ent 2 processing

Ent 1 Priority < 100
Ent 2 Priority \geq 100

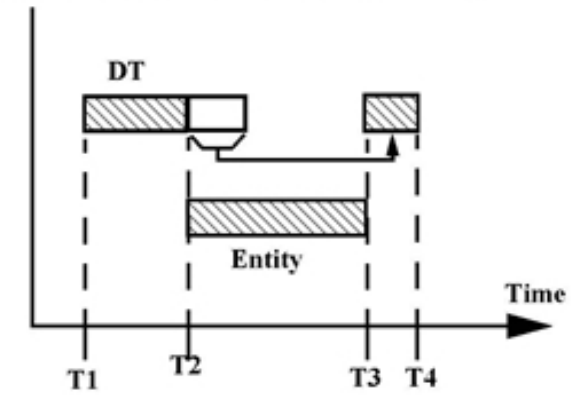
Entity Preempting Entity in Setup



Ent 1 setup must start over upon completion of Ent 2 processing

Ent 1 Priority < 100
Ent 2 Priority \geq 100

Entity Preempting a Downtime

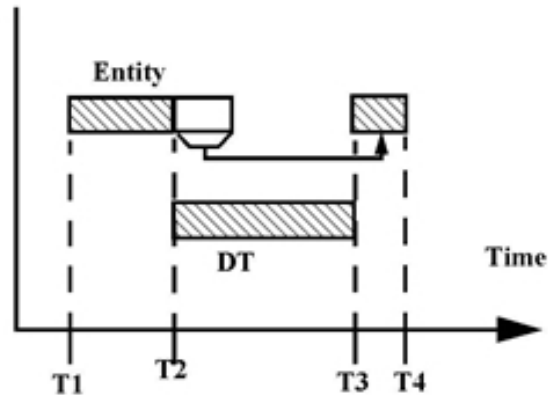


Downtime resumes immediately upon completion of entity processing

DT Priority < 100
Entity Priority \geq 200

Downtimes Preempting Other Events

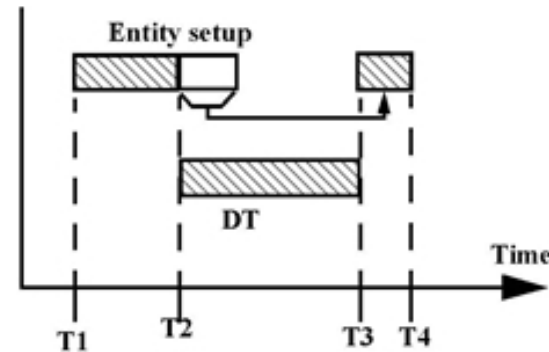
Downtime Preempting an Entity



Remaining entity processing time resumes upon completion of downtime

Entity Priority < 100
DT Priority >=100

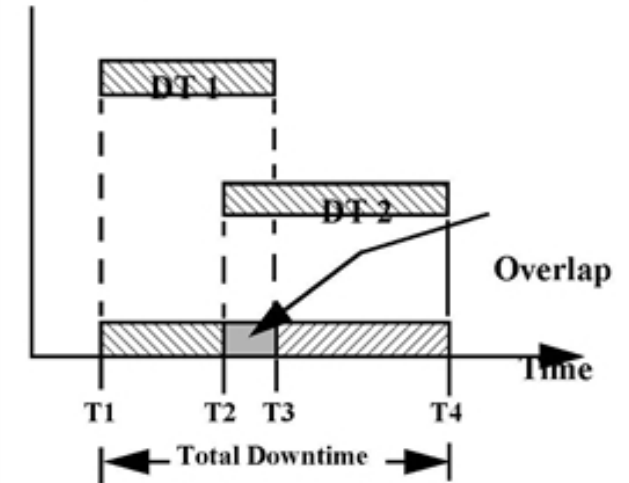
Downtime Preempting Entity in Setup



Remaining setup time resumes upon completion of downtime

Entity Priority < 100
DT Priority >=100

Overlapped/Concurrent Downtimes



DT1 Priority < 100
DT2 Priority >=100

Shift Carryover Functionality

- What is Shift Carryover?
 - Allows for a resource unit coming on shift to take over processing an entity for a resource unit going off shift.
 - Only works for Resources having multiple units.
 - There must be 1 or more units assigned to each shift.
 - Does not work between multiple, named resources.
- ProModel Demo Model
- Process Simulator Demo Model

ProModel demo allows us to see Resources move whenever a shift change occurs.

Simulation

Variables Arrays Locations One Location Zoom to Fit Zoom Views Panel Views List Plot Configurations Play Pause Stop Animation User Pause by Time User Pause by Date Step Events Trace Filter Trace to File Continuous Debug Logic Trace

Layout

2011.05.09 Mon 09:26

Work Carryover Between Shifts - Demo

Tool 1
Tool 2

Mill

Inspect

Que
1

Lathe

Tool 1
Tool 2

Model File:
Shift Carryover Demo 2018.pkg

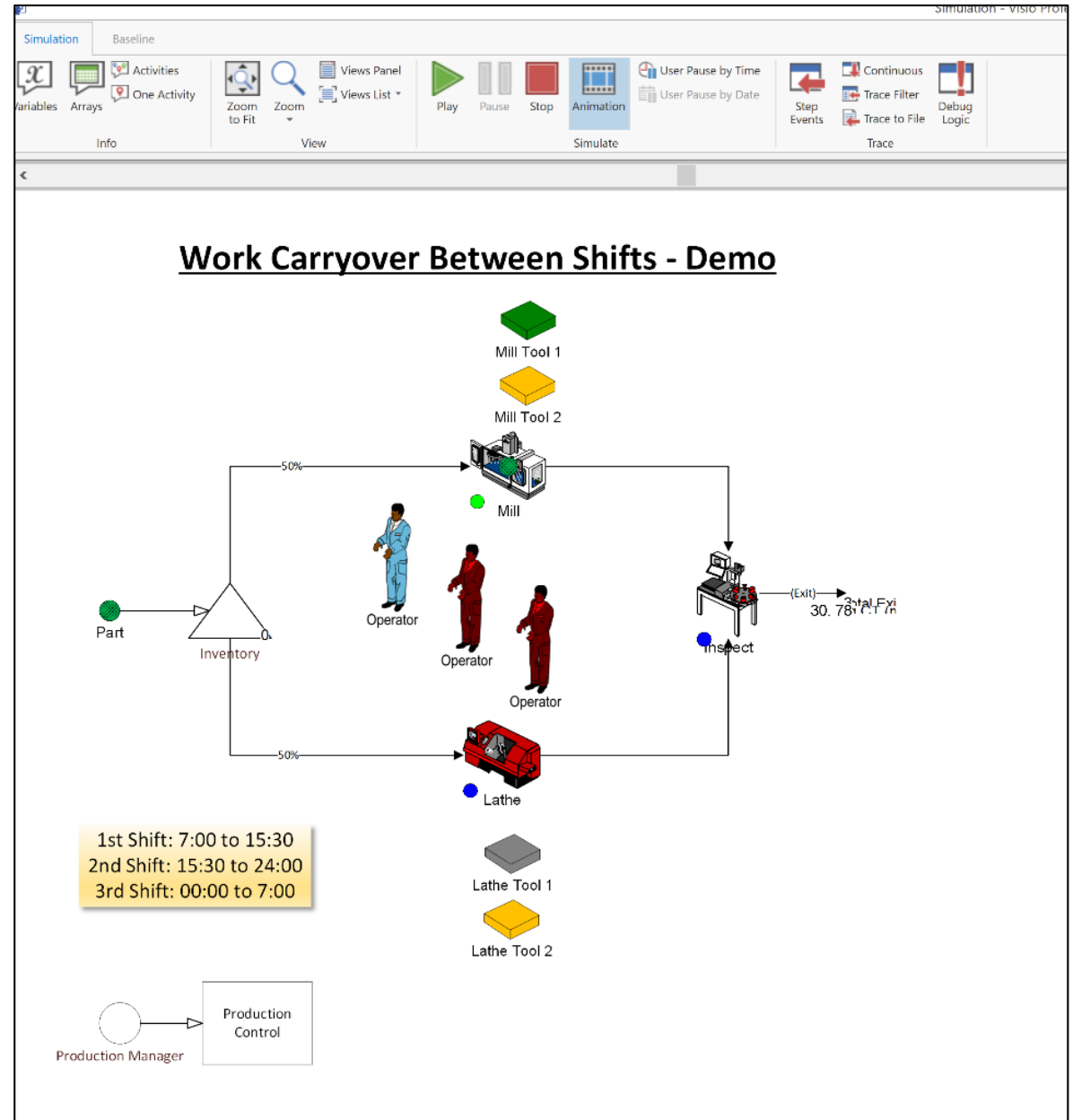
Break Node

Off-Shift Node

1st Shift: 7:00 to 15:30
2nd Shift: 15:30 to 24:00
3rd Shift: 00:00 to 7:00

Model File:
Shift Carryover Demo PCS 2019.pspkg

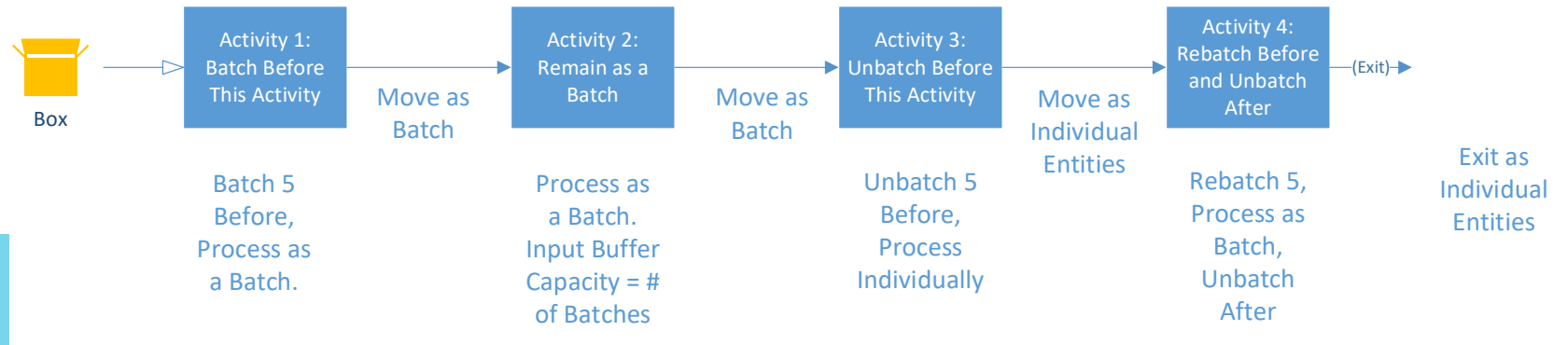
Process Simulator does not have mobile resources, but we can still see processing control change from one unit of a resource to another when a shift change occurs. Resource units that are red are off-shift.



Default Rules for Batch/Unbatch

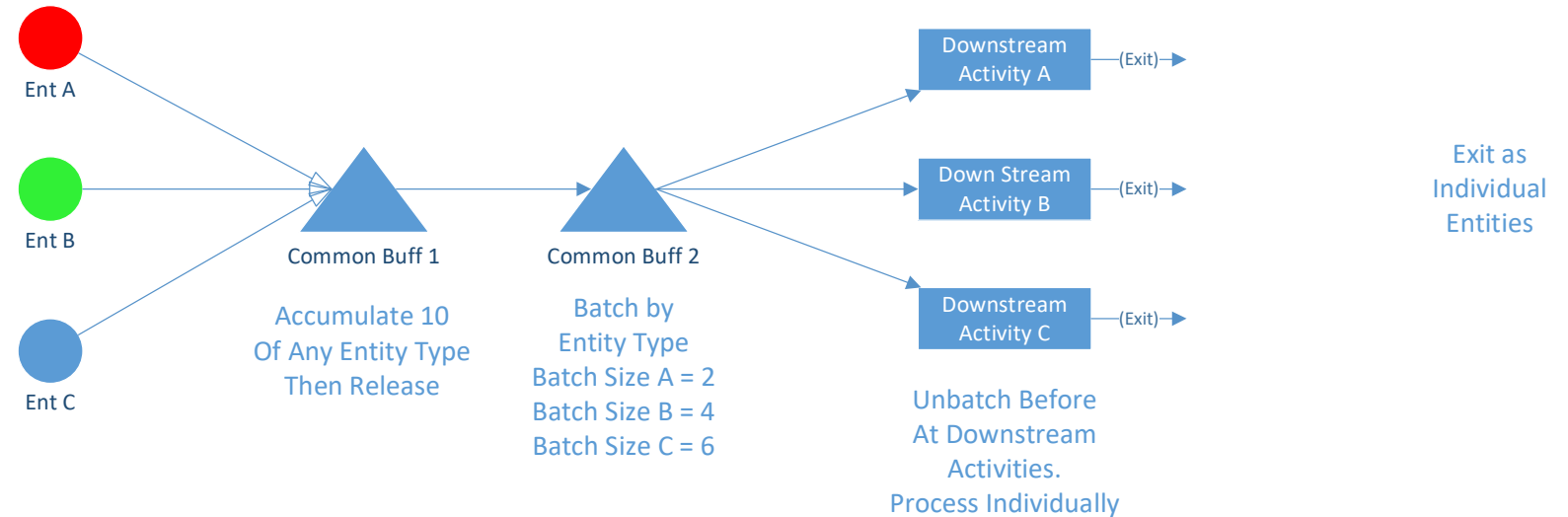
- Batching and Unbatching is normally done in the Input Buffers and/or Output Buffers of an Activity
 - Buffer capacity must be large enough for the Batch Size
 - Applies to both Batching and Unbatching, before or after an Activity.
- Batching can also be done in a Buffer element
 - More capability to control functionality over an Activity

Default Batching and Unbatching Using Activities



Model File:
Default Batching.pspkg

Batching and Unbatching Using Buffers provides more capability

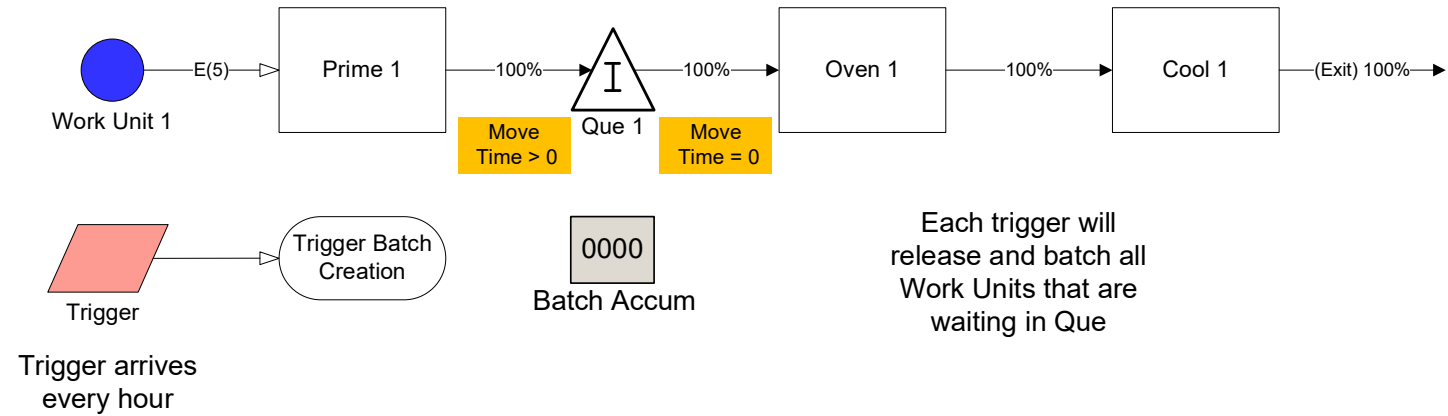


Custom Batching

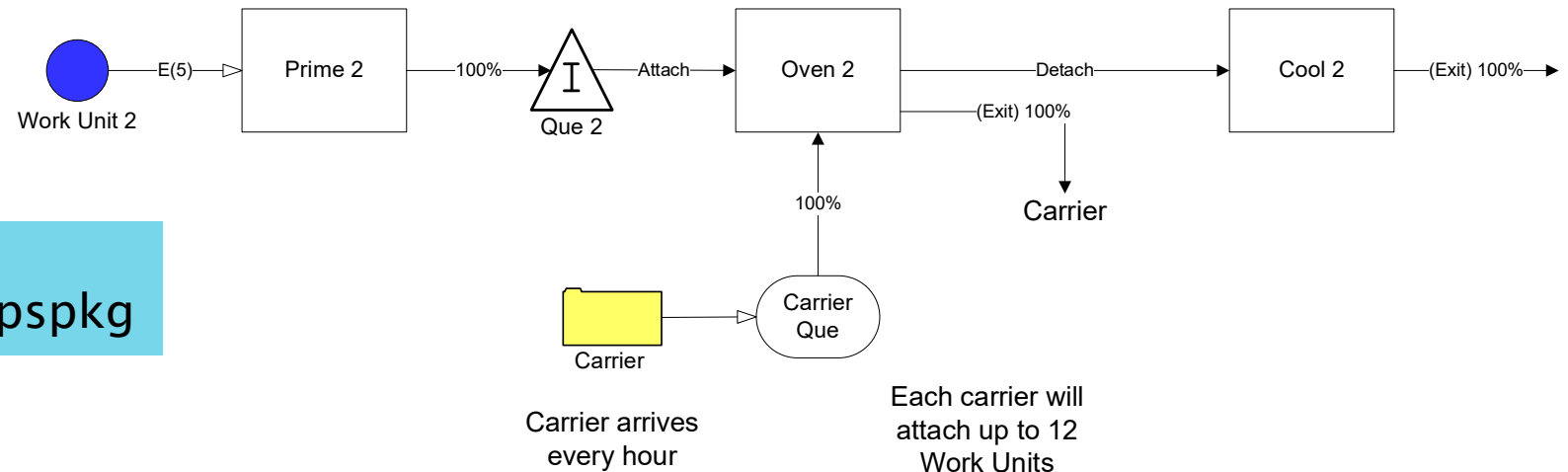
- 2 ways to Batch by Time instead of Quantity
 - Use a “carrier entity” to Attach & Detach the entities you want to be batched.
 - Use a Wait Until statement and Variables to control batching.

Custom Batching Examples: Batch by Time

Example 1: Using “Wait Until” Statement and Variables



Example 2: Using “Attach & Detach” Routings and Carrier Entity



Model File:
Custom Batching Demo v2019.pspkg

FINISHED

- Thanks for attending this training course! We hope it was helpful.
- Remember, help is only an email or phone call away.
- Good luck and happy modeling!

Technical Support
888-776-6633
support@promodel.com
6 am - 6 pm M-F, Mountain Time