Arena Basics

ISyE 6644

Seong-Hee Kim (Georgia Tech)

Barry Nelson (Northwestern)

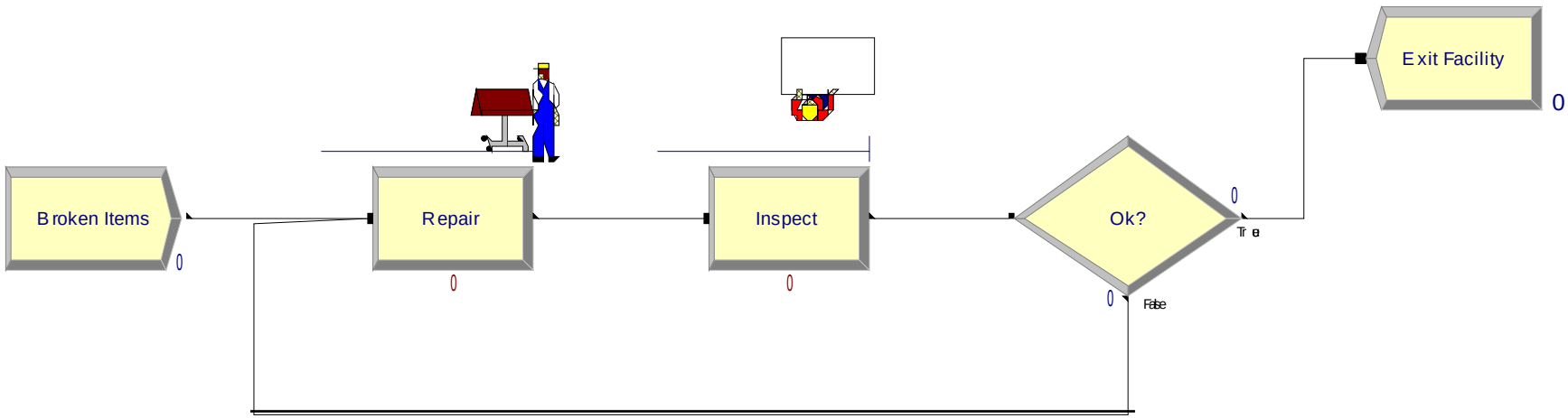
Overview

- We now move to the design and analysis of dynamic systems that evolve through time.
- We will use **Arena**, from Rockwell Software, which is one of several popular “discrete-event” simulation software packages.

Arena World View

- Arena takes the *process interaction* world view.
- Entities flow through a network of modules that describe their logical behavior.
- We describe the network by developing a process flowchart.

Flowchart Approach

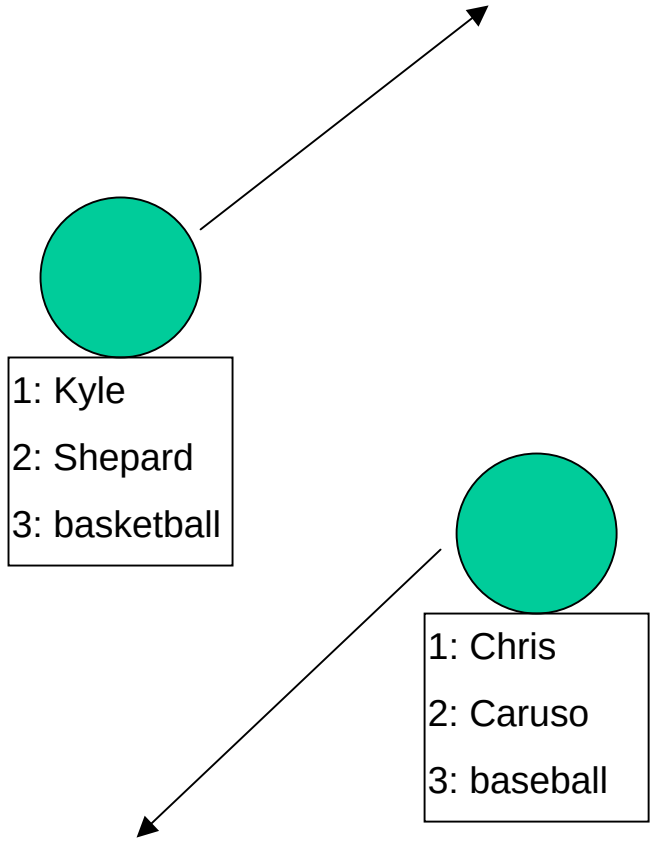


About modules…

- Arena contains a very large number of modules that are organized into *panels*.
- The panels are structured from high level to low level concepts:
  - Basic Process
  - Advanced Process & Advanced Transfer
  - Blocks & Elements (a programming language)
- Our goal is not to learn lots of modules, but rather to understand concepts that allow us to learn new modules as needed.

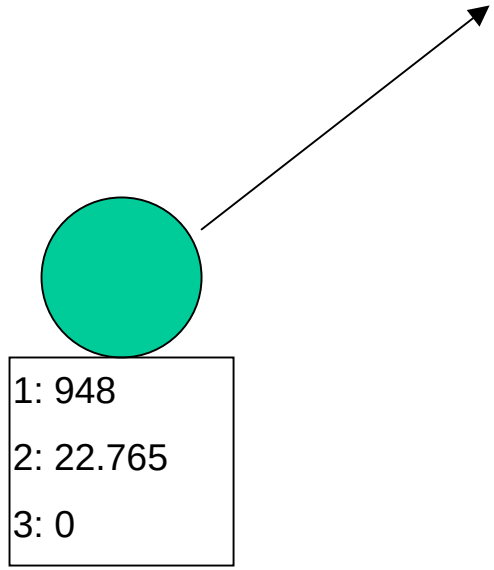
Entities

- *Entities* are dynamic elements that pass through the system.
- Entities are distinguished by their *attributes*.
- Ex: people, parts, information, paperwork, etc.



More on Entities

- Entities must be *Created* to get them into the model, and are *Disposed* when they leave.
- Unfortunately, attributes must be *numerical values.*



Queueing

- Entities queue when they need processing.
- In Arena…
  - An entity tries to *Seize* a *Resource*.
  - The time the entity uses the resource is the *Delay.*
  - If the resource is not available, the entity waits in a *Queue.*
  - The entity *Releases* the resource when processing is complete.

Resources

- Resources have…
  - A Name (up to you)
  - A Capacity (number of identical units of this resource; think # of servers).
  - And can have a Schedule (how many available when).
- And Resources can be animated.

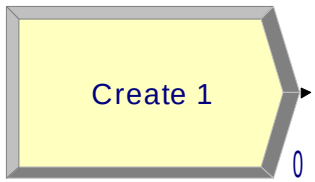
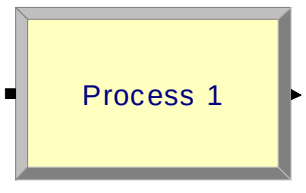
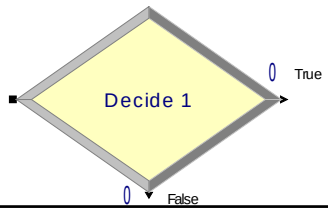
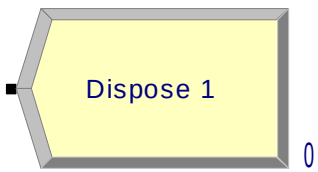
More on Resources

- Resources are automatically defined by some modules (e.g., Process)
- Resources can be defined manually, and the properties of all resources are changed, via the Resources spreadsheet on the Basic Process panel.
- There is also a Schedule spreadsheet for specifying Resource schedules.

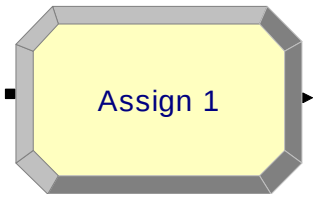
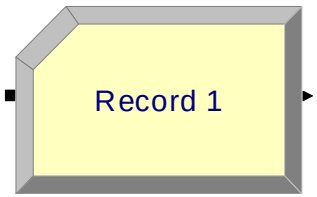
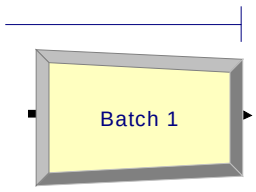
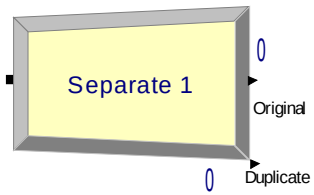
More on Queues

- Queues are created automatically by some modules (e.g., Process), and can be defined manually.
- Properties of a queue, including the ranking rule, are defined via the Queue spreadsheet.
  - First-in or Last-in first out
  - Lowest or Highest attribute value first

Basic Process Modules

	Push (possibly) batches of entities into the model with a (possibly) random time between.
	Models Queue-Seize-Delay-Release of Resource, or any part of this (like pure Delay).
	Make decisions about where to go next based on conditions or chance.
	Take entities out of the model and (perhaps) record statistics.

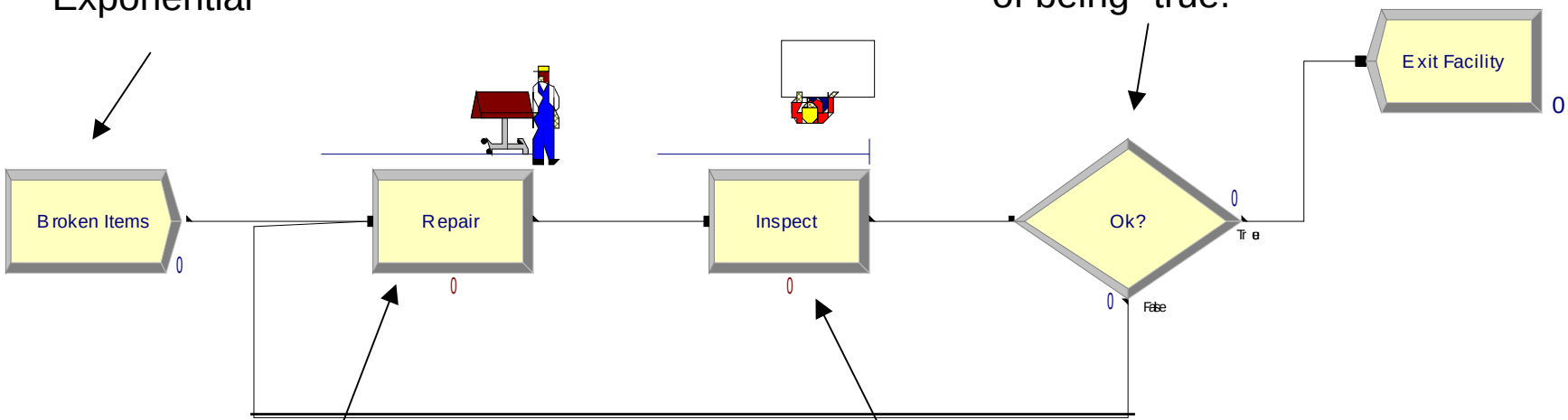
Basic Process Modules

	Assign values (especially Attributes) when an entity passes through.
	Record information when entities pass through, typically statistics on entities.
	Combine multiple entities into a single entity.
	Split multiple entities that were combined, or duplicate a single entity.

Example

Create item entities; note that "Random" time between means Exponential

The decision is "2-way by chance" with 90% chance of being "true."



Action is "Seize-Delay-Release" to represent a queue.

The delay can be given by an expression, in this case Expo(0.125), exponential with mean 0.125.

Basic Animation

- Entity movement (via module connections) and queues are automatically animated.
- The entity movement does **not** correspond to the passage of simulated time.
- Later we will learn how to animate transportation delays.

Entity Animation

- The Entity spreadsheet allows you to change the entity picture for each entity type.
- The Entity Type is a name, usually given when the entity is created.
  - Create: Entity Type: Items
- An Assign module can be used to change the entity Type or Picture as it moves through the model.

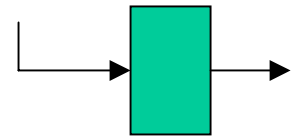
Queue Animation

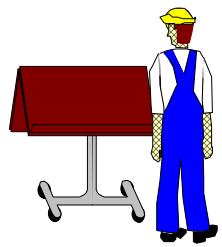
- The default queue is the sideways T.
- The queue symbol can be dragged anywhere, or reoriented.
- Often need to make the queue picture longer (which has no effect on queue capacity).

To lengthen the queue symbol, select it, grab the end, and pull.

Resource Animation

- Clicking the resource button lets you add a resource picture.
- You select pictures for the Busy, Idle, Inactive and Failed states.
- The Identifier must be the name of a resource already in the model (e.g., defined by a Process)





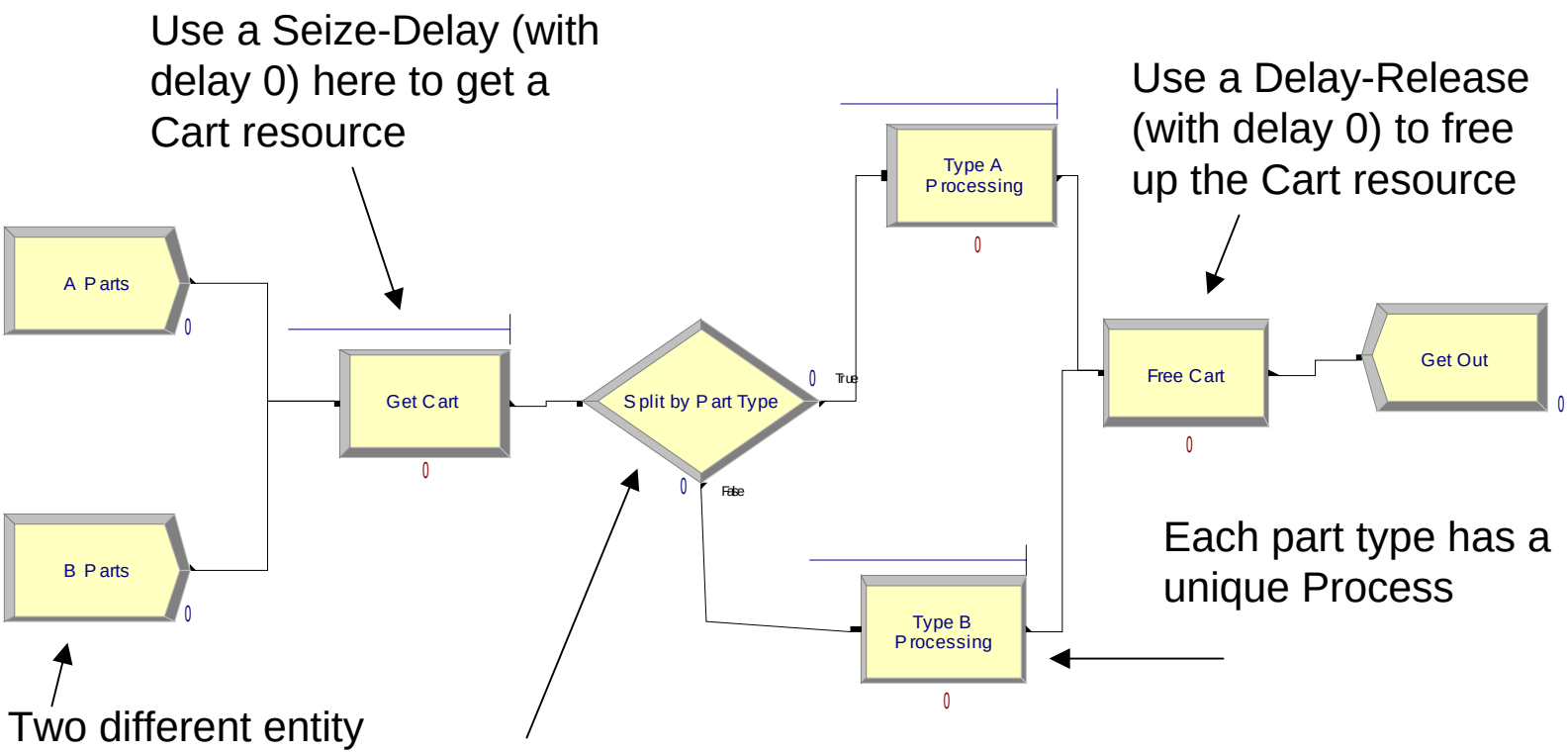
Delays

- Arena gives a default distribution for time between creations (“Random” = Expo) and delay (“Triangular”).
- If we want to put in a different distribution, we select “Expression” and enter the appropriate Arena function, such as WEIB, POIS, etc.
- We often get the expressions from the Input Analyzer.

Seize-Delay-Release

- Seize-Delay-Release need not be done in a single Process.
- One Process may be used to Queue and Seize the resource, a number of other modules may represent the processing, and yet another Process may finally Release the resource.

Example


Use a Seize-Delay (with delay 0) here to get a Cart resource
Use a Delay-Release (with delay 0) to free up the Cart resource
A Parts
B Parts
Get Cart
Split by Part Type
True
False
Type A Processing
Type B Processing
Free Cart
Get Out
Each part type has a unique Process
Two different entity types are created
Use the entity type in 2-way by Condition decision.

Internal Variables

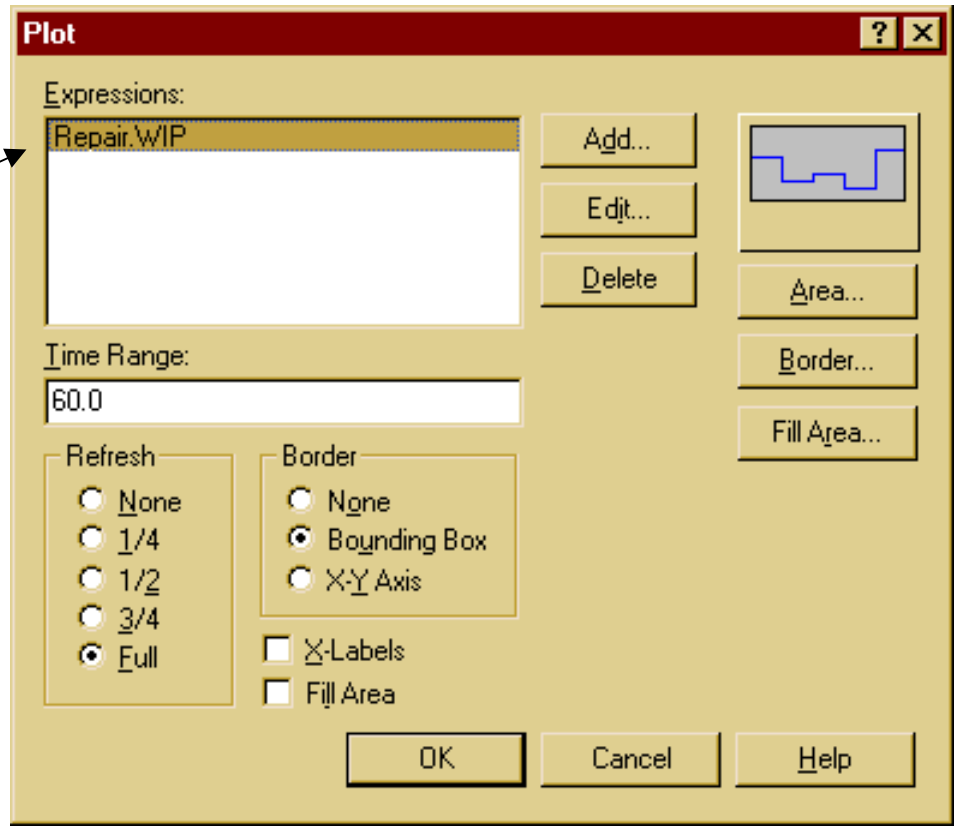
- Arena keeps a number of internal variables continually updated.
- These variables are useful for making choices in a Decide module, displaying in animated plots, or for recording statistics.
- The basic syntax is Name.Quantity

Basic Process Variables

- Create
  Name.NumberOut
- Process
  Name.NumberIn
  Name.NumberOut
  Name.WIP
  Name.WaitTime
- Decide
  Name.NumberOut True
  Name.NumberOut False
- Assign
  Name.NumberOut
- Batch
  Name.NumberOut
- Separate
  Name.NumberOut Orig
  Name.NumberOut Dup
- Record
  Name.NumberOut
- Dispose
  Name.NumberOut

Example

We can use the internal variable **Repair.WIP** to create a dynamic plot of the number of parts at the Repair Process

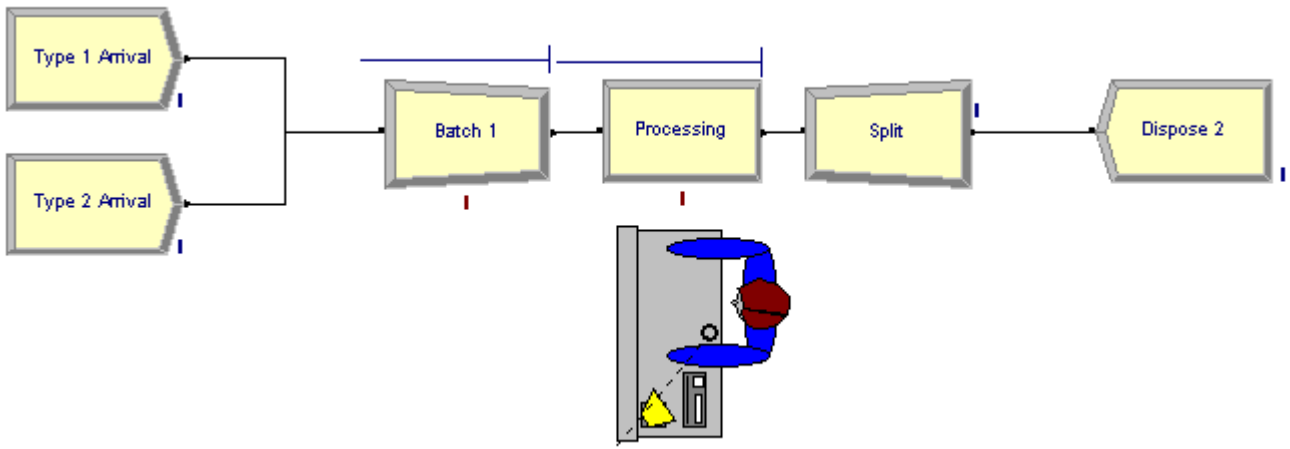


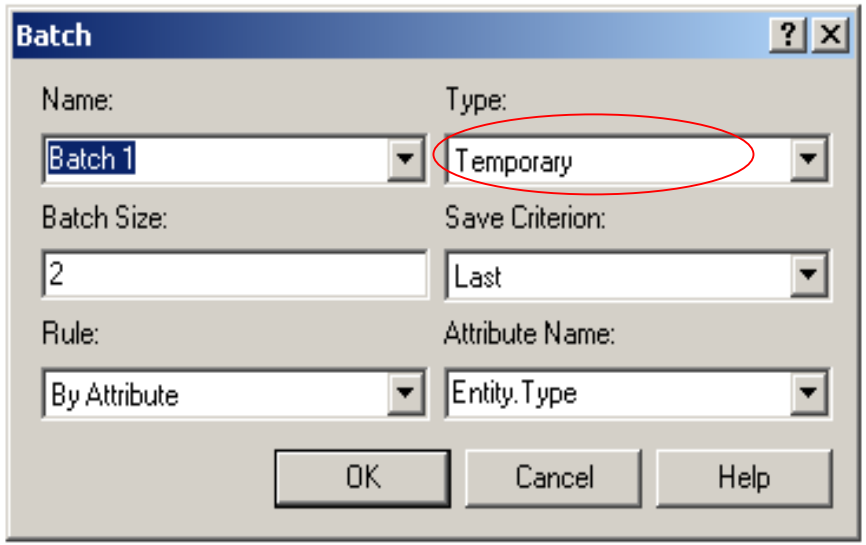
Simulated Time

- The simulation keeps its own internal clock that jumps forward from event time to event time.
- The time on the simulation clock is accessible through the Arena variable TNOW.
- TNOW is useful for marking entities or making time-based decisions.

Batch and Split

- Entities are processed in a batch of size two and then the system split entities that were combined.





If one chooses “Permanent” as Type, batched entities will never be split.

Separate module can be used to generate a duplicate of an entity. An example will be shown when we discuss Advanced Input Modeling.