Entity Transfer

What We'll Do ...

- **Types of Entity Transfers**
- **Model 8-1: Resource-Constrained Transfers**
- **Models 8-2, 8-3: Transporters**
- **Conveyors**
  - Model 8-4: Non-accumulating
  - Model 8-5: Accumulating

Types of Entity Transfers So Far

- **Connect**
  - Zero-delay
- **Route**
  - Non-zero-delay —
  - Stations, animated Routes
  - Fixed routes vs. entity-dependent Sequences
- **Connect and Route both assume:**
  - No limit on number in transit at a time
  - Entities have their own feet

New Types of Entity Transfers

- ***Resource-constrained*** **transfers**
  - Limit total number of entities in transit at a time
  - Entities still have their own feet
  - Telecommunications (number of packets), logistics (number of vehicles)
- **Material-handling devices**
  - *Transporters* – fork lifts, trucks, carts, wheelchairs
    - Usually place limits on numbers, capabilities of transporters
    - Like a Resource, except moveable
  - *Conveyors*
    - Belts, hook lines, escalators
    - Usually limit space on conveyor, speed
    - Non-accumulating vs. accumulating

Model 8-1: Small Manufacturing System with Resource-Constrained Transfers

- **Original system (Model 7-1)**
  - Assumed all transfer times = 2 minutes … keep (for now)
  - Parts have their own feet … keep (for now)
  - No limit on number of parts in transit at a time … dump
    - Now – no more than 2 parts can be in motion at a given time
    - If other parts are ready to go, they must wait until there's room to go
- **Model via existing constructs — think creatively**
  - Model "space" on the "road" as a Resource
  - Limit the number of Units of this Resource
  - Entity must Seize unit of "space" resource before beginning trip, Release it at end of trip

Two Ways to Model Resource-Constrained Transfers

- **Both use a new `Transfer` Resource representing space on the transitways**
  - Capacity set to **2** in Resource data module
- **Maybe the most obvious way (but won't do) …**
  - Before each Route module insert a Seize module to Seize one unit of **`Transfer`**
  - After each Station module insert a Release module to free up one unit of **`Transfer`**

Two Ways to Model Resource-Constrained Transfers (cont'd.)

- **Different way (will do, to illustrate new modules, set up for transporters and conveyors) …**
  - Replace Route modules with *Leave* modules (Advanced Transfer panel)
    - Transfer Out: Seize unit of **`Transfer`** resource before leaving station
      Resource, Resource Set
    - Also contains the Route operation
    - Get individual queues, with animation, for parts waiting to go
  - Replace Station modules with *Enter* modules (Advanced Transfer panel)
    - Defines the Station
    - Option of an unload Delay time (**`0`** for this model)
    - Transfer In: Release **`Transfer`** resource
- **Effect – slight increase in cycle times in system**

Transporter Concepts

- **Carts, fork lifts, trucks, wheelchairs, people, …**
- **When entity is ready to go somewhere, it needs to be "picked up" and moved**
- **Use *Transporters* — "moveable" resources**
- **Activities: *Request*, *Transport*, *Free***
  - Transporter Selection Rule: If > 1 transporter is available when Requesting
  - When freed and > 1 entity is waiting: Priorities, closest one
- **Two types of Transporters**
  - *Free-Path* (we'll do)
    - Travel time depends only on velocity, distance
    - Ignore "traffic jams" and their resulting delays
  - *Guided* (won't do)
    - AGVs, intersections, etc.

The Small Manufacturing System with Transporters

- **Have two carts to transport parts**
  - A cart can carry one part at a time
  - Carts move at 50 feet/minute
    - Will need to specify accurate distances between Stations
  - It takes 0.25 minute to load part on a cart, 0.25 minute to unload it from a cart
- **Modify Model 8-1 to Model 8-2**

The Small Manufacturing System with Transporters (cont'd.)

- **Create Transporter in Transporter data module (Advanced transfer panel)**
  - Name = **`Cart`**, Capacity = **`2`**, Velocity = **`50`**
  - Default the Distance Set (later), Units = Per Minute, Initial Positions
    - *Mind the units* – consistency here, in Distance Set (later)
- **Animation picture for `Cart` Transporter**
  - Transporter button  , Animate Transfer toolbar
  - Identifier = **`Cart`**, pictures for Idle, Busy, Inactive states
  - Ride point (details in book)
  - Drop it anywhere in flowchart view (hidden during run)

The Small Manufacturing System with Transporters (cont'd.)

- **Request a `Cart` – modify existing Leave modules**
  - Delay = **`0.25 Minute`** for load time
  - Transfer Out = **`Request Transporter`**
  - Transporter Name = **`Cart`**
  - Selection Rule = **`Smallest Distance`**
    - Applies when > 1 transporter is available
    - Others: Cyclic, Random, Preferred Order, Largest Distance (???)
  - Save Attribute = **`Cart #`** (remember which cart … for later)
- **Instead of Leave: Request-Delay-Transport**
  - More complex, more flexible – book has details, examples

The Small Manufacturing System with Transporters (cont'd.)

- **Free the Cart – modify existing Enter modules**
  - Delay = **`0.25 Minute`** for unload time
  - Transfer In = **`Free Transporter`**
  - Transporter Name = **`Cart`**
  - Unit Number = **`Cart #`** attribute of part entity
- **Instead of Enter: Station-Delay-Free**
  - More complex, more flexible – book has details, examples

Distances for Transporters

- **Define contents of Distance Set `Cart.Distance`**
- **Distances (in feet) moved by parts:**

	To				
	Cell 1	Cell 2	Cell 3	Cell 4	Exit System
Order Release	37	74			
Cell 1		45	92		
Cell 2	139		55	147	
Cell 3				45	155
Cell 4		92			118

Units!!

  - Blank cells: part movements that don't occur
- **Enter these data in Distance data module (Advanced Transfer panel)**
  - Name = **`Cart.Distance`**
  - Stations button, Distance for above data
  - Direction is implied

Units!!

Animating Transporter Movement

- **Add distances to animation**
- **Delete all the old Route Path animation objects**
  - But leave the Station animations
- **Add animated transporter distances with Distance button  , Animate Transfer toolbar**
  - Dialog, placement similar to Route Paths
  - Identifier = **`Cart.Distance`**
  - Click in Beginning Station marker, intermediate clicks, Ending Station marker
  - Options for Rotate, Flip
  - Grid, Snap to help place animated transporter distances

Parking Areas for Transporters

- **Animate transporters when they're free**
- **Parking button  , Animate Transfer toolbar**
  - Like a Queue animation –
  - Cursor becomes cross hairs, click near lower left of Station marker to start, click for first Point or head of Line
  - More clicks for more Points (double-click to end), or second click to end Line
  - Want enough points/space for all transporters (2 here)
  - Repeat for all Stations where Transporters could be freed

More Distances — Empty Transporters

- **Above Distances incomplete — only for part movements along their sequences**
- **Transporters must also move when empty (*deadheading*)**
  - In general, $n(n - 1)$ distances need definition for network with n nodes
  - Some not possible — Order Release to Exit System
- **14 more distances to define in Distances data module (not grayed):**

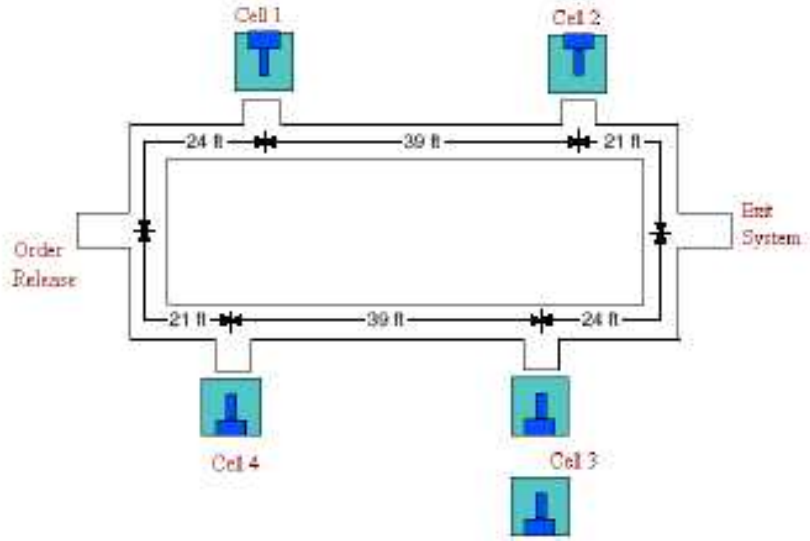
From \ To	Order Release	Cell 1	Cell 2	Cell 3	Cell 4	Exit System
Order Release		37	74			
Cell 1	155		45	92	129	
Cell 2	118	139		55	147	
Cell 3	71	92	129		45	155
Cell 4	34	55	92	139		118
Exit System	100	121	158	37	74	

Model 8-3: Refining the Animation for Transporters

- **As it stands, `Part` Entities disappear from animation when waiting to be picked up by a `Cart` Transporter**
  - Model logic OK … get right answers … animation is flawed
- **Solution – *Storage* for entity to reside in, be animated, while it waits for something (here, a `Cart` Transporter)**
  - Can get statistics on numbers in Storages

Conveyors

- **Replace Transporters with a conveyor**
- **Loop conveyor to follow main path, clockwise**
- **Six entrance/exit points**
  - Load, Unload takes 0.25 minute
  - Each part is 4 feet per side, but want 6 feet of conveyor space for clearance on corners
- **Speed = 20 feet/minute** *Units!!*



Conveyor Concepts

- **Entity to be conveyed must wait for space**
- **Conveyor consists of *cells***
  - Equal size, constantly moving – think of a narrow escalator
- **Entities might require multiple contiguous cells**
- **Must define *cell size*; tradeoff involved:**
  - Small cells: accurate model but slow execution
  - Large cells: just the opposite!
- **Entities *Access* space, *Convey*, and *Exit***
- **Conveyor = series of linear *Segments***
  - Link to form loops, diverge points, converge points

Types of Conveyors

- **Both travel in a single, irreversible direction**
- ***Nonaccumulating*: belt, bucket line, escalator**
  - Spacing between entities on it doesn't change
  - Entire conveyor stops for entity Access/Exit if Load/Unload time is > 0
- ***Accumulating*: rollers**
  - Conveyor never stops moving
  - If entity on it stops to Exit, other entities behind it are blocked and bunch up (entities ahead of it keep moving)
  - When blockage ends, blocked entities go on but maybe not all at once (spacing requirements)

Model 8-4: Small Manufacturing System with Nonaccumulating Conveyors

- **Modify Model 8-1 (resource-constrained transfer)**
- **Define new Variables `Load Time` and `Unload Time`, each with initial value 0.25**
- **Delete all the Route Paths**
- **Define Conveyor via Conveyor data module, Advanced Transfer panel**
  - Conveyer = **`Loop Conveyor`**
  - Segment Name = **`Loop Conveyor.Segment`**
  - Type = Non-Accumulating
  - Velocity = **`20`** (feet), Units = Per Minute *Units!!*
  - Cell Size = **`3`** (feet) *Units!!*
  - Max Cells Occupied = **`2`** (cells per entity)

Leave, Enter Modules for Conveyor

- **Change each Leave module**
  - Delay = **`Load Time`**, Units = Minutes
  - Transfer Out = Access Conveyor
  - Conveyor Name = **`Loop Conveyor`**
  - # of Cells = **`2`**
  - Connect Type = Convey
- **Change each Enter module**
  - Delay = **`Unload Time`**, Units = Minutes
  - Transfer In = Exit Conveyor
  - Conveyor Name = **`Loop Conveyor`**

Conveyor Segments

- **Define one-way lengths (in feet) of segments**
- **Segment data module, Advanced Transfer panel**
  - Name = **`Loop Conveyor.Segment`**
  - Beginning Station = **`Order Release`**
  - Next Stations button
    - Name Next Station in correct sequence
    - Give distance (in feet) to this next station
- **Segment animation**
  - Put Station markers in front of each Resource picture
  - Segment button  , Animate Transfer toolbar
  - Dialog, crosshairs, clicking just like Distances for Transporters

Conveyor Statistics

- ***Run* > *Setup* > *Project Parameters*** **to check Conveyor Statistics**
- **Get percent of time blocked (stopped)**
- **Utilization statistic is average percent of space occupied on conveyor (not percent of time that a part was on the conveyor)**
- **To see conveyor stop (it's nonaccumulating) more clearly, change `Load Time` and `Unload Time` to much greater values than `0.25`**

Model 8-5: Change Conveyors to Accumulating

- **Conveyor module**
  - Change Conveyor Type to Accumulating
  - Accumulation Length = 4 (in feet), amount of space the accumulated parts need on the conveyor
- **Running, see very little accumulation in animation**
  - To see more, increase **`Load Time`** and **`Unload Time`**