

# CS4632 Object Oriented Modeling and Simulation

Models with Resources

Dr. José M. Garrido C.

Department of Computer Science  
Kennesaw State University

Spring 2017

# Resources

- Processes are the **active** components of system
- Resources are **passive** components of a system
- In the process-interaction approach to simulation, resource pools need to be defined and created.
- Processes can acquire a number of resource items from a resource pool.

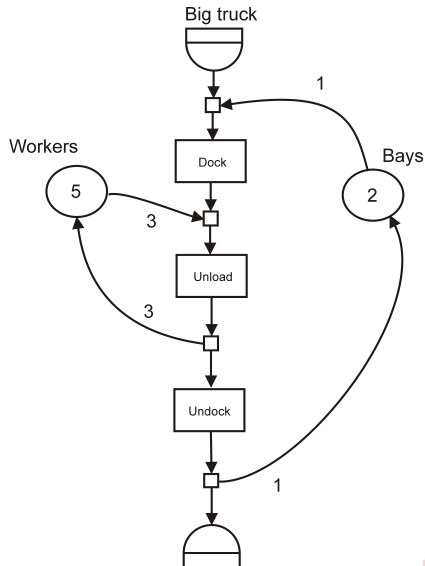
# Model of a Warehouse System

- A warehouse receives two types of trucks, **small** and **big** trucks, that arrive periodically with goods.
- Each truck needs an unloading **bay** to dock and to unload the goods.
- Small trucks need two **workers** to unload, big trucks need three workers.
- The warehouse has a limited number of unloading bays and a limited number of workers.
- Trucks may need to wait if there are no available unloading bays and/or not sufficient available workers.
- Each type of truck has a mean arrival rate and a mean unloading time interval.

The simulation model of the Warehouse system consists of the following components:

- Process definitions
  - Small trucks
  - Big trucks
  - Arrivals of small trucks
  - Arrivals of big trucks
  - The main process
- Standard Resource definitions, implemented as objects of class Res
  - Unloading bays
  - Workers

# Activity Diagram of Big Truck



# Big Trucks

The following sequence of activities is performed by big trucks:

- ➊ After arriving, acquire one unloading bay if available, otherwise wait.
- ➋ Dock into bay, this takes a constant time interval.
- ➌ Acquire 3 workers if available, otherwise wait.
- ➍ Unload, takes a random time interval (from a uniform distribution).
- ➎ Release workers.
- ➏ Undock, takes a constant time interval.
- ➐ Release unloading bay.
- ➑ Depart and terminate.

# Small Trucks

The following sequence of activities defines the behavior of small trucks.

- ➊ After arriving, acquire one unloading bay if available, otherwise wait
- ➋ Acquire 2 workers if available, otherwise wait
- ➌ Unload, takes a random time interval (from a uniform distribution)
- ➍ Release workers
- ➎ Release unloading bay
- ➏ Depart and terminate

# Trace of Simulation Run

The trace in the output shows the relevant events, such as:

- Processes (trucks) start to request resources
- Resources are allocated to the processes
- Other processes start to wait for available resources
- Processes terminate after they have completed all of their activities



# Model Implementation

- The complete model implementation of the Warehouse model is stored in the archive file, `wareh.jar`.
- This archive includes the following source files:  
`Wareh.osl`, `Sarrivals.osl`, `Barrivals.osl`,  
`Struck.osl`, **and** `Btruck.osl`.

# Summary Results

End Simulation of Busy Warehouse System

date: 04/17/2016 time: 09:15

Total small trucks serviced: 36

Average period small truck spends in warehouse: 24.759

Small truck average wait period: 17.113

Total number of big trucks serviced: 43

Big truck average wait period: 21.189

Avg per btruck in warehouse: 34.101

# A Model of Machine Parts-Replacement

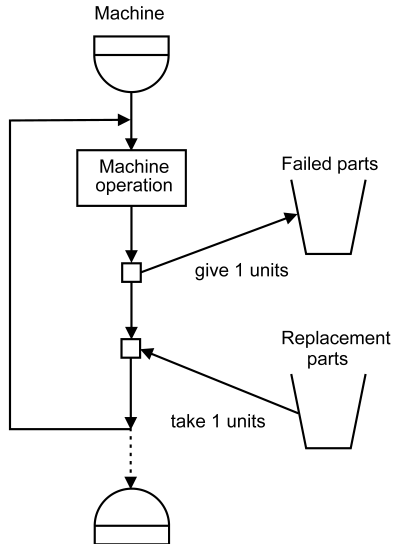
- There are several machines in a shop, each having a part that **fails** periodically.
- The part that fails needs to be removed from the machine, and a replacement part installed, if available.
- This removal and replacement of parts is carried out by an operator.
- The parts that fail are to be **repaired** by the repairman, who also has other jobs to carry out when not repairing parts.

# Detachable Resources in Model

The model includes two different resource **containers**, which are implemented as objects of class *Bin*. These resource containers are:

- The parts that fail, *fault\_parts*;
- The replacement parts, *rep\_parts*. These are the parts that have been repaired.

# Activity Diagram of Machine



# Activities in the Model

- The machine processes are **producers** of damaged parts; they give items to the *fault\_parts* resource container.
- The machine processes are also **consumers** of replacement parts; they take items from the *rep\_parts* resource container.
- The figure shows a simplified activity diagram of the machine process with the two resource containers.
- The repairman process is a **consumer** of damaged parts; it takes items from the *fault\_parts* resource container.
- The repairman process is also a **producer** of replacement parts; it gives items to the *rep\_parts* resource container.

# Model Implementation

- The source files with implementations for all classes are stored in archive file `preplace.jar`.
- This archive includes files: `Preplace.osl`, `Machine.osl`, and `Repairman.osl`.

# Machine Replacement Parts Summary Results

```
End Simulation of Machine Parts Replacement System  
date: 4/18/2016 time: 09:27  
Total machines serviced: 3  
Average down period: 15.299999999998931  
Average up period: 2782.15476170466  
Average machine utilization: 0.7273607220142901
```