Construction Operation Simulation

Lecture #3

DES engine and hand simulation

Amin Alvanchi, PhD
Construction Engineering and Management
Outline

- Introduction
- Updating system state
- Simulation engine
- Model initialization
- Event occurrence procedure
- Hand simulation
Main steps in simulation studies

1. Define the problem of the study
2. Choose another method for solving the problem
3. Is simulation a proper tool?
   - Y: Develop the logical simulation model
   - N: Collect data required for model development
4. Collect data required for model development
5. Develop the simulation model
6. Is model verified?
   - Y: Is model valid?
   - N: Develop the simulation model
7. Is model valid?
   - Y: Design simulation experiments
   - N: Analyze output data and adjust operation parameters to come up with the proper design
8. Design simulation experiments
9. Run construction operation simulation
10. Analyze output data and adjust operation parameters to come up with the proper design
11. Execute the construction operation according to the design!
So far we have identified DES model elements (entities, resources, activities, events, state variables)!

In DES model we are going to imitate system behaviour and updates! Which element can be used for this purpose?
So far we have identified DES model elements (entities, resources, activities, events, state variables)!

In DES model we are going to imitate system behaviour and updates! Which element can be used for this purpose?

Event element is our main tool for tracing system behaviour and all updates to the system state over time!

*Reminder:* System event is an occurrence that changes the state of the system.

To be able to trace all system updates we need to pick *mutually exclusive* set of events (so called main events) which:

1) Can create all possible system state updates (update all state variables)!
2) Do not have overlaps and duplications in creation of system updates.
Example: 1) In our earthmoving example we should pick *set of events which update all state variables:*
Example: 1) In our earthmoving example we should pick set of events which update all state variables:

- Number of trucks in hauling trip
- Number of trucks in dumping line
- Volume of soil dumped
- Dumping site status (Available/occupied)
- Number of trucks in loading line
- Number of trucks to the loading site
- Loader status (Idle/busy)
- Number of trucks in dumping line
Example: 1) In our earthmoving example we should pick *set of events which update all state variables*:
Example: 1) In our earthmoving example we should pick *set of events which update all state variables*:
Example: 1) In our earthmoving example we should pick \textit{set of events which update all state variables:}

- Loader status (Idle/ busy)
- Number of trucks in loading line
- Start loading
- Loading completion
- Number of trucks in hauling trip
- Start dumping
- Dumping site status (Available/ occupied)
- Dumping arrival
- Dumping completion
- Number of trucks in dumping line
- Volume of soil dumped
- Number of trucks to the loading site
- Loading arrival
Example: 1) In our earthmoving example we should pick **set of events which update all state variables**: Are all related events captured?
Example: 2) Avoid overlaps and duplications in creation of system updates.

- Loader status (Idle/ busy)
- Start loading
- Number of trucks in loading line
- Number of trucks to the loading site
- Loading completion
- Loading arrival

- Dumping site status (Available/ occupied)
- Start dumping
- Dumping arrival
- Number of trucks in dumping line
- Volume of soil dumped
- Dumping completion
- Number of trucks in hauling trip

+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
- 
+ 
-
Example: 2) Avoid overlaps and duplications in creation of system updates.

- Loader status (Idle/busy)
- Number of trucks in loading line
- Number of trucks in dumping line
- Number of trucks to the loading site
- Start loading
- Loading arrival
- Loading completion
- Volume of soil dumped

- Dumping site status (Available/occupied)
- Number of trucks in hauling trip
- Loading completion

- Avoid overlaps and duplications in creation of system updates.
Example: 2) Avoid overlaps and duplications in creation of system updates.

Do these events overlap in creating busy status of loader?
Example: 2) Avoid overlaps and duplications in creation of system updates.

- **Loading** arrival makes loader busy in one ways:
  1) After a *truck arrival* and when loader is idle, *loader starts loading*

- **Start loading** makes loader busy in two ways:
  1) After a *truck arrival* and when loader is idle, *loader starts loading* OR 2) after completion of loading a truck from queue starts loading!
Example: 2) Avoid overlaps and duplications in creation of system updates.
Example: 2) Avoid overlaps and duplications in creation of system updates. 

Do these events overlap in decreasing number of trucks in the line?

- Loader status (Idle/busy)
- Number of trucks in loading line
- Start loading
- Loading completion
- Loading arrival
Updating system state

- **Example**: 2) Avoid **overlaps** and **duplications** in creation of system updates.

- **Loading completion** decreases number of trucks in line in one way:
  1) After **loading completion** a truck from queue starts loading!

- **Start loading** decreases number of trucks in line in one way:
  1) After **loading completion** a truck from queue starts loading!
Example: 2) Avoid overlaps and duplications in creation of system updates.

All system state updates created by Start loading event are created by other events, so we can remove Start loading event from set of selected system activities!
What events are selected in general:

**Activity setup**
(station or job-site)

- Arrival event for an activity
- Activity
  - Queue
  - Resource
- Activity completion event

Updates created by middle events (e.g., start activity) are usually addressed by arrival and completion events and can be eliminated!
Consecutive activities share events:

We can usually remove events of the middle activity setups!
Example: *Mutually exclusive* set of events creating all system state updates with no overlap!
In class practice 1: In our paving operation example, determine mutually exclusive set of events and the way that they update state variables!
Simulation engine

- Tracking time in simulation systems:
  - Three different clocks:
    - **Wall clock**: Represents our current time!
    - **Computer calculation clock**: Represents time spent for doing calculations of the simulation model, it has the pace as wall clock, but starts from zero when a model calculation begins!
    - **System clock**: Represents time of being simulated system. It has completely different pace compared to the wall clock!
Tracking time in simulation systems:

Continuous simulation:

- Initial system state setting
- Update
- System clock

The smaller the time step, the more accurate the calculation!

For example, depending on the system update pace and accuracy required, time step can be set at 1 second, 20 seconds, 1 minute, 50 minutes, 10 hours, 1 day, 1 month, 3 months, 1 year, etc.

Discrete event simulation:
Simulation engine

- Tracking time in simulation systems:
  - Continuous simulation:
    - Even time steps
  - Discrete event simulation:
    - Un-even time steps based on event occurrence

For tracing time in continuous simulation engines for each move we just need to advance time by one step and update system state base on equations relating different system components!

In what basis we can advance time in DES models?
Future event list (FEL): For tracking time in DES engine we always maintain minimum number of future events identified for the system in a list called Future event list (FEL).

For every event on the list we maintain: type of event and time of occurrence. Depending on the type of information we need to collect in our simulation study we may also maintain information of entity (e.g., entity id) involved in the event.

Identified events in FEL are maintained according their time of occurrence, we also call these events scheduled events. For advancing time we just need to withdraw first scheduled event from the list and move the time to its time of occurrence!
Example: Time advancement in future event list:

Event (4): Dumping complete/610/ truck 4
Event (3): Loading complete/300/ truck 2
Event (2): Dumping arrival/140/ truck 3
Event (1): Loading arrival/100/ truck 1

Future Event List (FEL)

Current Time: 610

According to the specification you saw for FEL what computer programming feature you know for capturing FEL?
Future event list (FEL):

- FEL can properly be coded in computer programming as an *Array* variable!
- Time advancing in simulation engine is our way for withdrawing (removing) scheduled events from FEL.
- To be able to track the system updates during simulation study period we need to schedule new events (or add new upcoming events) to FEL over time!

How can we identify all upcoming events and properly advance the time?
Simulation engine

Simulation engine procedure:

- **Set FELtime to 0**
- **Sim Start**

- Withdraw current event from FEL

  - AnyScheduledEvent
    - Move to the earliest scheduled event
      - Return its time: t
        - Return event type and entity
          - Execute *event occurrence procedure* at time t

- **t < FinalSimTIme**
  - **Y**
  - **N**

Initial events are created during model initialization to be able to start simulation runs at the beginning.

Event occurrence procedures add new/upcoming events to FEL in to be able to continue simulation runs.
Model initialization

- **Model initialization**: To be able to start simulating our simulation model we need to set model initialization through which time of the model is set to 0, initial system state is set up and initial system events are scheduled and added to FEL.

- **Example**: We are going to model our simple earthmoving example one working day from 7 am to 7 pm. There are two trucks, one loader and unlimited spots for dumping, working in project. Since we start from start of the working shift: there will be no trucks in the system; loader is idle. Truck1 arrives at loading site at time 7:00:30 and truck2 arrives at loading site at 7:01:05. We need to add these two arrival events to FEL to be able to start a simulation run!

- **Future Event List (FEL)**
  - Event (1): Loading arrival/ 7:01:05/ truck 2
  - Event (1): Loading arrival/ 7:00:30/ truck 1

- **Initial condition**: After simulation begins, usually it takes some time for the model to pass initial condition, affected by *initial system state*. So, for simulation result analysis we usually need to let a simulation run continues until initial system state effect disappears!
Event occurrence procedure

- **Event occurrence procedure** is our main tool for scheduling new/upcoming events in our simulation model runs!

*Remember:* System event is an *occurrence* that changes the state of the system.

1) For scheduling new events we need first to determine system state changes expected after occurrence of an event:

**Example:** *Truck arrives to the loading site*; changes made to the state variables by this event:

1) number of trucks on their way to the loading site is decreased by one
2) if there is an idle loader, number of busy loaders is increased by one (number of idle loaders is decreased by one)
3) if all loaders are busy, number of waiting trucks for loading is increased by one

*Remember:* An activity is a period of time limited between two events where by occurrence of the first event, the occurrence of the second event will be known.
2) If by occurrence of an event a new activity(s) is commenced, according to the definition of activity element, we can determine the finish event(s) of the activity(s). These events are new events which are scheduled and added to FEL!

**Example 1:** *Truck arrives to the loading site:*

1) number of trucks on their way to the loading site is decreased by one
2) if there is an idle loader, number of busy loaders is increased by one (number of idle loaders is decreased by one); Schedule a new loading complete event and add it to FEL!
3) if all loaders are busy, number of waiting trucks for loading is increased by one

**Example 2:** *Loading completion event:*

1) number of trucks on their way to dumping site is increased by one; Schedule a new dumping arrival event and add it to FEL!
2) if there is no truck waiting to be loaded, number of idle loaders is increased by one
3) if there are trucks waiting to be loaded, number of trucks waiting to be loaded is decreased by one; Schedule a new loading completion event and add it to FEL!
In class practice 2: Two events among set of mutually exclusive events identified in our earth moving example are *Truck arrives to the dumping site* and *Dumping completion event*, what are expected system updates by these events occurrences? What are new events scheduled (added to FEL) as results of these events occurrences?
In class practice 3: Two events among set of mutually exclusive events identified in our road paving example are *Spreading completion* and *Compacting completion*, what are expected system updates by these events occurrences? What are new events scheduled (added to FEL) as results of these events occurrences?
Event occurrence procedure

- Event occurrence procedure in flowchart: Event occurrence procedures are important parts of simulation programs. To be able to implement them, by programming and through coding, more properly (specially for more complicated event occurrence procedures), it is suggested that we develop flowcharts for them!

Example: For “Truck arrives to the loading site” and “Loading completion” events we will have written procedure and procedure in flowchart as:
1) number of trucks on their way to the loading site is decreased by one
2) if there is an idle loader, number of busy loaders is increased by one (number of idle loaders is decreased by one); Schedule a new loading complete event and add it to FEL!
3) if all loaders are busy, number of waiting trucks for loading is increased by one
Event occurrence procedure

1) number of trucks on their way to dumping site is increased by one; Schedule a new dumping arrival event and add it to FEL!
2) if there is no truck waiting to be loaded, number of idle loaders is increased by one
3) if there are trucks waiting to be loaded, number of trucks waiting to be loaded is decreased by one; Schedule a new loading completion event and add it to FEL!
In class practice 4: Draw flowcharts for two events identified for our road paving example: *Spreading completion* and *Compacting completion*;
Hand simulation

- To better understand steps taken in computer simulation, in hand simulation we are going to follow similar steps a computer simulation program follows, though with hand and on paper!
- For doing hand simulation we need:
  1) Recognize model elements
  2) Determine event procedures
  3) Initialize the model
  4) Start simulation engine procedure; start advancing time and executing related event occurrence procedure

*Remember* to update system state after each event occurrence
Hand simulation

- **Example:** Suppose our earthmoving example with 3 trucks and 1 loader in the system. There is no limitation in number of dumping sites. At this stage we are going to hand-simulate a *deterministic* model where all activity durations are determined:
  - Loading: 10 minutes
  - Trip to dumping site: 5 minutes
  - Dumping: 2 minutes
  - Trip from dumping: 5 minutes

**Hand simulation:**

1- **Model elements:**

- **Entity:** Truck
- **Resource:** Loader
Example (cont’d):

Activity:
Since there is no limitations in dumping area, we are not going to see any line or waiting time in the dumping area, we can easily reduce number of activities to 2 (why?):

- Loading: 10 minutes
- Trip: 12 minutes

Event:
Loading site arrival; Loading completion

System state:
Num of trucks in loading Queue; Loader status; Num of trucks on trip(can be calculated from other two state variable after initial condition passed so we do not track its changes!)
Hand simulation

- Example (cont’d):

2- Determine event procedures

*Truck arrives to the loading site:*

1) number of trucks on trip is decreased by one
2) If loader is idle, make loader busy; *Schedule a new loading complete event!*
3) If loader is busy, number of waiting trucks for loading is increased by one

*Loading completion event:*

1) number of trucks on trip is increased by one; *Schedule a new loading arrival event!*
2) if there is no truck waiting to be loaded, make loader idle
3) if there are trucks waiting to be loaded, number of trucks waiting to be loaded is decreased by one; *Schedule a new loading completion event and add it to FEL!*
Example (cont’d):

3- Initialize the model

State variables: Num of trucks in loading Queue = 0; Loader status=idle

Events: Truck1 arrives at 6:59:01; Truck2 arrives at 6:59:52; Truck3 arrives at 7:19:30

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Arrival</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>1</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>2</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>3</td>
</tr>
</tbody>
</table>
Example (cont’d):

4- Start simulation engine procedure

Event 1: 6:59:01/ Arrival/ Truck1

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>1</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>2</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>3</td>
</tr>
</tbody>
</table>

It is recommended to have FEL array indexed / sorted by event time and placing new events in FEL in order!
Example (cont’d):

Event 2: 6:59:52/ Arrival/ Truck2

<table>
<thead>
<tr>
<th>System state</th>
<th>FEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Event</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
</tr>
</tbody>
</table>

Event 3: 7:09:01/ Completion/ Truck1

<table>
<thead>
<tr>
<th>System state</th>
<th>FEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Event</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
</tr>
</tbody>
</table>
Example (cont’d):

Event 4: 7:19:01/ Completion/ Truck2

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

System state FEL

Event 5: 7:19:30/ arrival/ Truck3

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

System state FEL
Example (cont’d):

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:29:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:31:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:39:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7:41:30</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7:49:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>3</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
<td>4</td>
</tr>
<tr>
<td>7:29:30</td>
<td>Completion</td>
<td>3</td>
</tr>
<tr>
<td>7:31:01</td>
<td>Arrival</td>
<td>2</td>
</tr>
<tr>
<td>7:39:30</td>
<td>Completion</td>
<td>1</td>
</tr>
<tr>
<td>7:41:30</td>
<td>Arrival</td>
<td>3</td>
</tr>
<tr>
<td>7:49:30</td>
<td>Completion</td>
<td>2</td>
</tr>
<tr>
<td>7:51:30</td>
<td>Arrival</td>
<td>1</td>
</tr>
<tr>
<td>7:59:30</td>
<td>Completion</td>
<td>3</td>
</tr>
<tr>
<td>8:01:30</td>
<td>Arrival</td>
<td>2</td>
</tr>
</tbody>
</table>
Example (cont’d):

Questions to be answered by gathering simulation data?

- What is average productivity (dumping per hour)?
- What is the maximum queue size?
- What is the average waiting time?
- What is idle-time percentage of the loader?
Example (cont’d):

Questions to be answered by gathering simulation data?

- What is average productivity (dumping per hour)?

### System state

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
<th>System state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td>First arrival</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td>First arrival</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td>First arrival</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:29:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:31:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:39:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:41:30</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:49:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

### Average productivity

Total 3 dumping trips completion in (7:49:30 - 6:59:01 =) 50' 29" or 3 in 0.843h or rate of **3.56/h** dumping trip completion.

Is this accurate?
Example (cont’d):
Questions to be answered by gathering simulation data?

- What is the maximum queue size? 1
Example (cont’d):

Questions to be answered by gathering simulation data?

- What is the average waiting time?

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
<th>Waiting time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td>Truck 1 in/out</td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td>Truck 2 in</td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td>Truck 2 out</td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td>Truck 3 in/out</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td>Truck 1 in</td>
</tr>
<tr>
<td>7:29:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td>Truck 1 out</td>
</tr>
<tr>
<td>7:31:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td>Truck 2 in</td>
</tr>
<tr>
<td>7:39:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td>Truck 2 out</td>
</tr>
<tr>
<td>7:41:30</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td>Truck 3 in</td>
</tr>
<tr>
<td>7:49:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td>Truck 3 out</td>
</tr>
</tbody>
</table>

Average: 0:05:41

Is this accurate?
Example (cont’d):

Questions to be answered by gathering simulation data?

- What is idle-time percentage of the loader?

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Queue</th>
<th>Loader status</th>
<th>Loader idle time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6:59:01</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6:59:52</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:09:01</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:19:01</td>
<td>Completion</td>
<td>0</td>
<td>0</td>
<td>Idle start</td>
</tr>
<tr>
<td>7:19:30</td>
<td>Arrival</td>
<td>0</td>
<td>1</td>
<td>Idle finish</td>
</tr>
<tr>
<td>7:21:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:29:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:31:01</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:39:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:41:30</td>
<td>Arrival</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7:49:30</td>
<td>Completion</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

| Total time | 0:50:29 | Total idle time | 0:00:29 |
| Idle percentage | 0.96%  | Is this accurate? |        |
Consider a mason-laborer problem in which the laborer stockpiles brick pallets on a scaffold. The scaffold is configured to allow for the stacking of two 30-brick pallets. There is one laborer who serves three masons. A schematic diagram showing the sequence of the operation is presented below.
A time study was carried out to obtain the durations required for the supply of packets and laying of bricks (each pallet-30 bricks).

Duration for the laborer to stockpile one brick pallets on a scaffold: 5 minutes
Duration for a mason to place one brick pallet: 13 minutes

Assume that there is an infinite supply of brick pallets and there are no bricks stacked on the scaffold at the beginning of the operation. Work starts at 7 am and finishes 7 pm.

*Hand simulate the model for 1 hours!*

*How long labor is idle? How long masons (in total) are idle?*

*Due in 2 weeks*
Thank you!