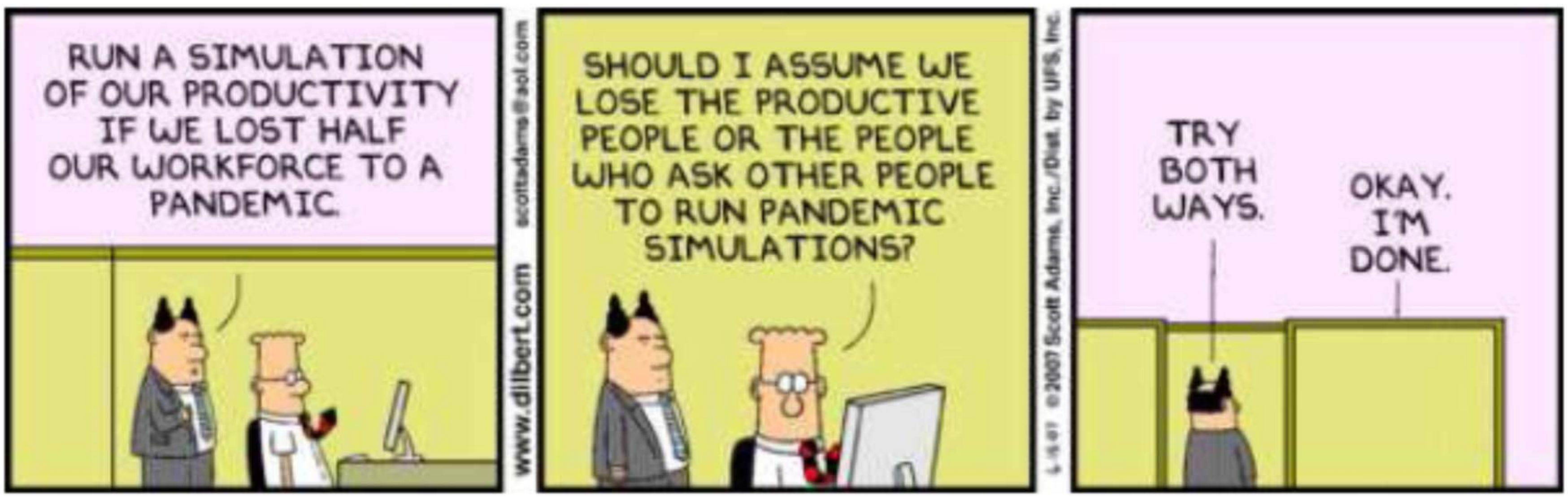


RUN A SIMULATION OF OUR PRODUCTIVITY IF WE LOST HALF OUR WORKFORCE TO A PANDEMIC.
SHOULD I ASSUME WE LOSE THE PRODUCTIVE PEOPLE OR THE PEOPLE WHO ASK OTHER PEOPLE TO RUN PANDEMIC SIMULATIONS?
TRY BOTH WAYS.
OKAY. I'M DONE.
www.dilbert.com

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Simulation

Lecture 3

Simulation study and more modelling

This lecture

- How to perform a simulation study?
- Validation
- Simulation assignment
- More modelling

After this lecture:

- You can make more complicated simulation models
- You know about steps in a simulation study and validation
- You can start working on the assignment

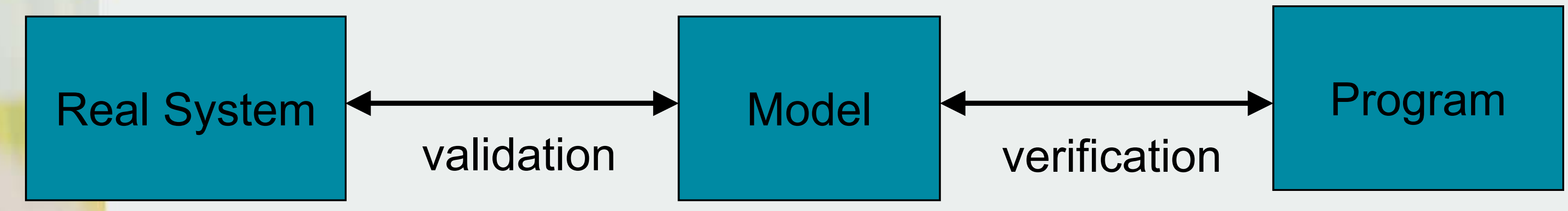
Steps in a sound simulation study

Simulation is more than a programming exercise

Looks so easy in theory, but in practice.........

Steps in a sound simulation study

1. Formulate the problem and plan the study
   - Question to answer
   - Scope
   - Performance measures
   - Scenarios
   - Software
   - Resources
2. Collect date and define a model
   - Conceptual model:
     - system description
     - assumptions
     - data
     - Input analysis
     - level of detail
3. Validate conceptual model
4. Computer program and verification


Real System
validation
Model
verification
Program

Steps in a sound simulation study (2)

5. Pilot runs
6. Validate programmed model
7. Design experiments
8. Production runs
9. Analyze output
10. Document, present, and use results

Validation

- You are expected to study reading material by yourself
- There will be a question at the exam about this.

Development of simulation model (from previous lecture)

In step 1:

- Performance measures

In step 2:

- System description
- Input data/distributions
- Assumptions
- Events (event graph)
- State
- Event handlers: in words or flow diagram/pseudo-code
  - Update state
  - Update performance measures
  - Generate new events

Validation:

process of determining whether a simulation model is an accurate representation of the system *for the particular objectives of the study.*

Simulation model is valid:

if it can be used to take decisions on the system *equal to* the decisions that would be taken on the basis of an experiment with the real system

Increasing validity

1. Collecting information
2. Communication
3. Maintain `assumptions document (conceptual model)
   - Overview: goals, issues, performance measures
   - description of subsystems and interaction
   - assumptions
   - summaries of data
   - sources of information

Increasing validity

4. Validate components of the model by quantitative techniques
   - Lab tests
   - Input distibution `goodness of fit’
   - Sensitivity analysis
5. Validate output
   - Existing situation
   - Reality check of expert opinion
   - Turing test
   - Calibration/fine-tuning
   - Statistical procedures
6. Animation

Validate output

$$\mu_s : \text{system}$$

$$\mu_m : \text{model}$$

$$\hat{\mu}_m : \text{result of simulation}$$

$$|\hat{\mu}_m - \mu_s| \leq |\hat{\mu}_m - \mu_m| + |\mu_m - \mu_s|$$

$$|\hat{\mu}_m - \mu_m| : \text{good experimentation}$$

$$|\mu_m - \mu_s| : \text{validation}$$

Level of detail

- guided by objective of the study
- entity in real system does not have to be the entity in simulation
- sensitivity
- start moderately, more detail possibly later
- only issues of interest
- data availability
- time, money constraints
- computer constraints

Credibility

‘ *effectivity = quality * acceptance* ‘

Quote Rob van der Hoorn, Vrumona from workshop Supply Chain Management

Credibility (2)

- Involvement, understanding, agreement of problem owners/managers
- Demonstration of validation and verification
- Your reputation
- Visualizations: `that's my system!'

Simulation assignment: the Uithoflijn

- Before you start: make sure that you understand discrete-event simulation models
- Practice!!!
  - Takes about 2 hours
  - Saves time during the assignment

Simulation assignment: the Uithoflijn

- “ *A new tramline between Utrecht CS and the Uithof will start operating*”
- Now: Test runs are in process
- The goal of the simulation assignment is to perform a simulation study of the operational performance of the Uithoflijn

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Simulation assignment: research questions

- What are feasible frequencies for the Uithoflijn?
- Which turn-around should be used at the end points?
- U-ov plans to use a frequency of 16 trams per hour. How well does this frequency handle the amount of passengers from the prognosis? How much passenger growth can be handled with this frequency?

- Detailed information on cs.uu.nl/docs/vakken/mads/sim_assignment_information.html
- Read, read, read this carefully and accurately

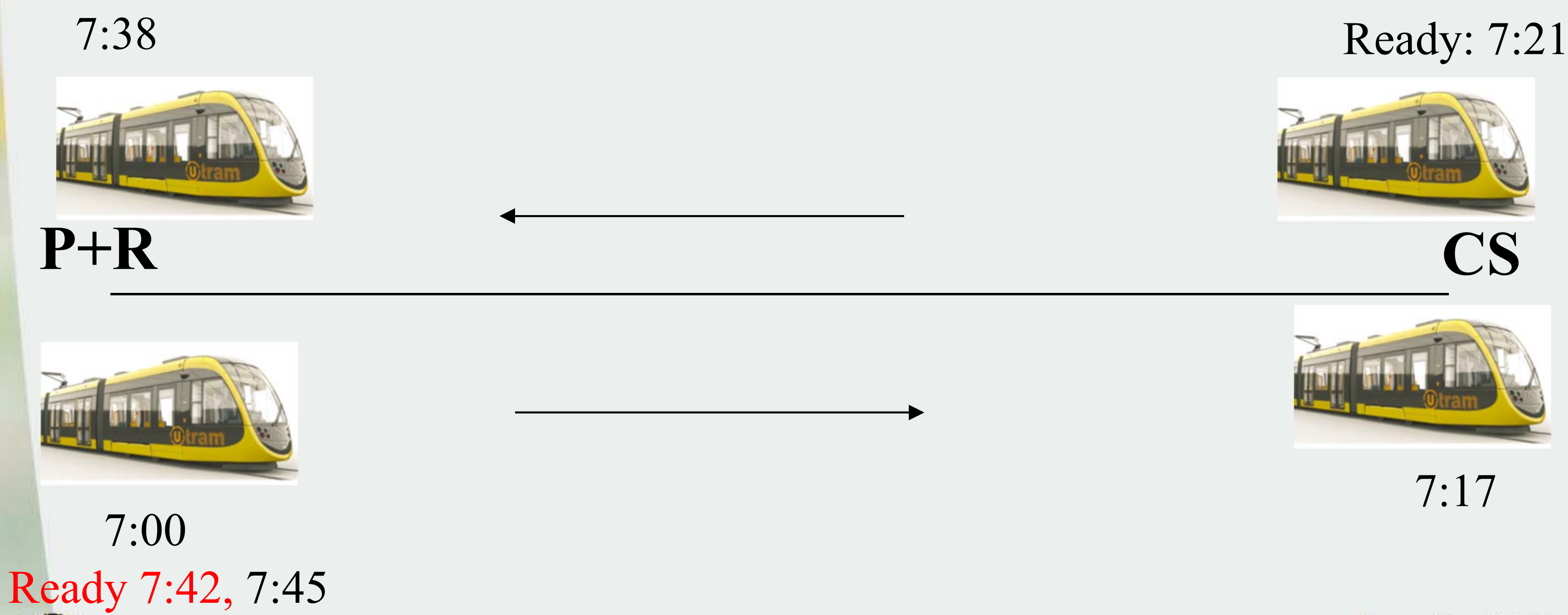
Simulation assignment: the Uithoflijn (2)

- **Perform a complete scientifically sound simulation study**
- Use imperative programming language; *pair programming is strongly recommended*
- You have to implement a **discrete-event** simulation
- You have to use probability distributions for driving and dwell times of the tram. The number of passengers entering and exiting the tram has to be stochastic as well.
- You are allowed to use libraries for probability distributions

Timetable and number of trams

Example

- Suppose we want a frequency of 12 per hour, every 5 minutes
- One way travel time 17 minutes
- Turnaround time 4 minutes



Timetable and number of trams

- Suppose we want a frequency of 12 per hour
- One way travel time 17 minutes
- Turnaround time 4 minutes
- First tram
  - starting at P+R at .00
  - arrives at CS at .17.
  - Then it leaves CS at .21 and
  - arrives again at P+R at .38
  - And can leave for the next trip at .42
- Timetable
  - Departures P+R: .00; .05; .10; .15 etc
  - Departures CS: .01; .06; .11 .16, etc
- Since the round trip times equals 42 minutes(17+4+17+4) we need $\left\lceil \frac{42}{5} \right\rceil = 9$ trams

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Data files

- Driving times Nieuwegein-lijn
- Passenger counts line 12
- Passenger number prognosis by Province of Utrecht
- Artificial files for validation

Available at surfdrive

Milestones

- Work in groups of 3 (recommended) or 2
  - If you do not have a group, meet in the break
- Intermediate milestones
  - Bring written material and/or laptop and explain your work to practicum supervisor
  - Practicum supervisor gives feedback
  - Pass is required for additional examination
- Content of milestones:
  1. Simulation model on paper including assumptions (no input analysis yet).
     - Next Monday, Tuesday, Thursday
  2. Implemented simulation model and input analysis

Final report

- The report has to be between 10 and 20 pages of 11 pt A4. This excludes pictures and tables.
- Statement of the contributions in the individual group members.
- Complete report of the assignment is required to pass the course.
- Discussion meeting at the end is mandatory

This is a real-life problem

- This is modelling, so motivation of choices and assumptions is important
- **There is no unique solution:**
  - Slightly different event-scheduling models are valid
  - Different assumptions on minor details
  - Different probability distributions may be sufficiently valid
  - You have to design your output analysis
- No output check by Domjudge
- **Common sense:**
  - always keep practical consequences in mind
- Nonsense is not accepted (costs points)

Tips from last year's students

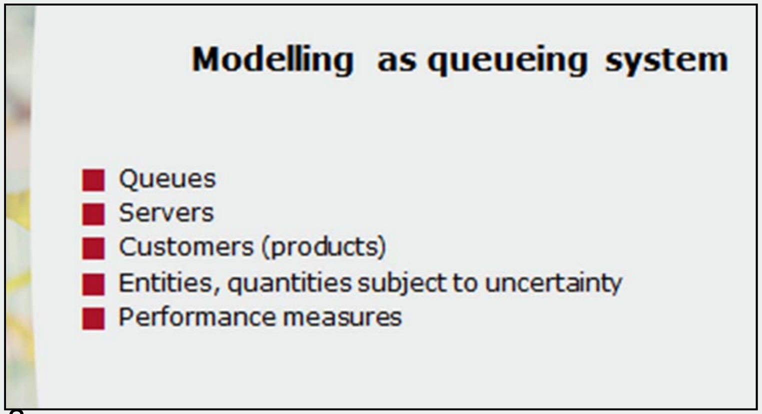
- Finish your model before your start implementing
- Programming is harder than expected
- Work in parallel on implementation and input analysis
- Start in time, input analysis is a lot of work
- Use *r* for statistics
- There is no unique solution, explain what you do
- If you are uncertain about the correctness of certain parts and running out of time, assume correctness and proceed with the work

Modelling as queueing system

- Queues
- Servers
- Customers (products)
- Entities, quantities subject to uncertainty
- Performance measures

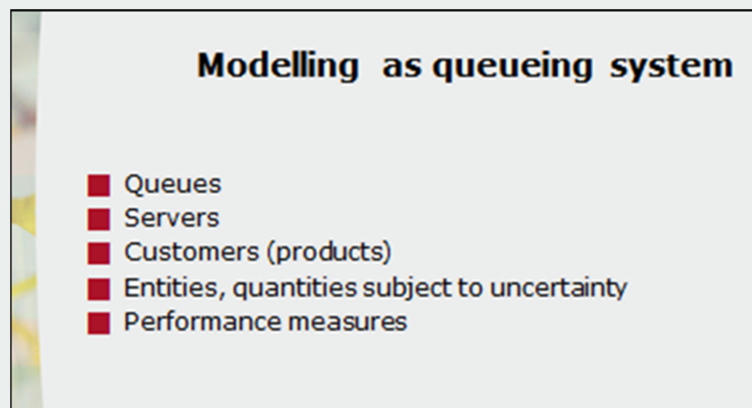
Queueing systems: examples

- Production line with buffers
- Luggage handling system at Schiphol
- Dynamic bus station
- Terminal layout at an airport
- Planning trucks at gates at HEMA DC
- Design of a the layout of a warehouse
- Planning of a distribution network for DHL
- Inventory management



Design of a the layout of a warehouse

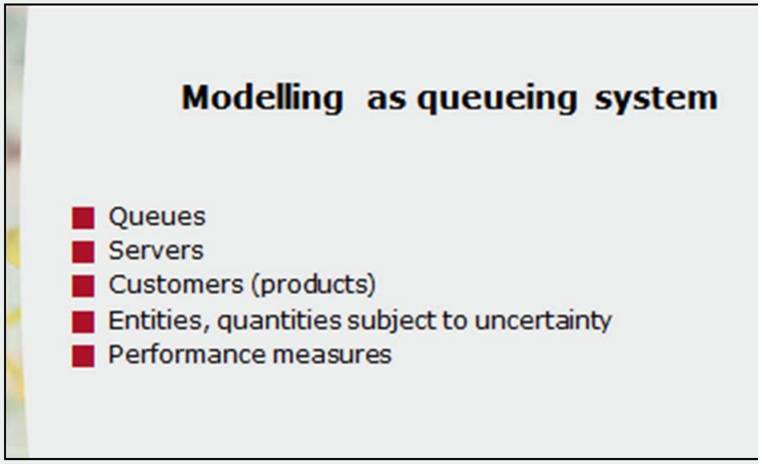
- Products: orders to be picked
- Servers: order pickers, transportation equipment, item in storage location
- Uncertainty: order arrival, availability of item
- Performance measures: throughput time of a set of orders
- Decision: where to locate what, how many order pickers, equipment
-



Production line with buffers

- Production line:
  - One product
  - Different stages/machines
  - Buffers of limited size between the machine
  - Machines may go down

- Model as queuing system
- Which trade-off is there when we want to determine the buffer size?



Machine with failures, e.g.computer server

- Jobs arrive according to a Poisson process with 60 per hour
- Processing times follow an exponential distribution with an average of 30 seconds
- Machine goes down after 3 hrs on average (exponential distribution)
- Repair takes 15 minutes on average (exponential distribution)

- Find discrete-event simulation model to measure production per day.

Events

- JobArrives
- JobFinished
- MachineGoesDown
- MachineRepaired

Event handlers

- JobArrives
  - Advance time to t_{now}
  - If machine is idle and up
    - Set machine state to busy
    - Generate processing time p
    - ScheduleJobFinished $t_{now}+p$
  - Otherwise put job in queue (update state of queue accordingly)

- *Problem: machine may go down during processing????*

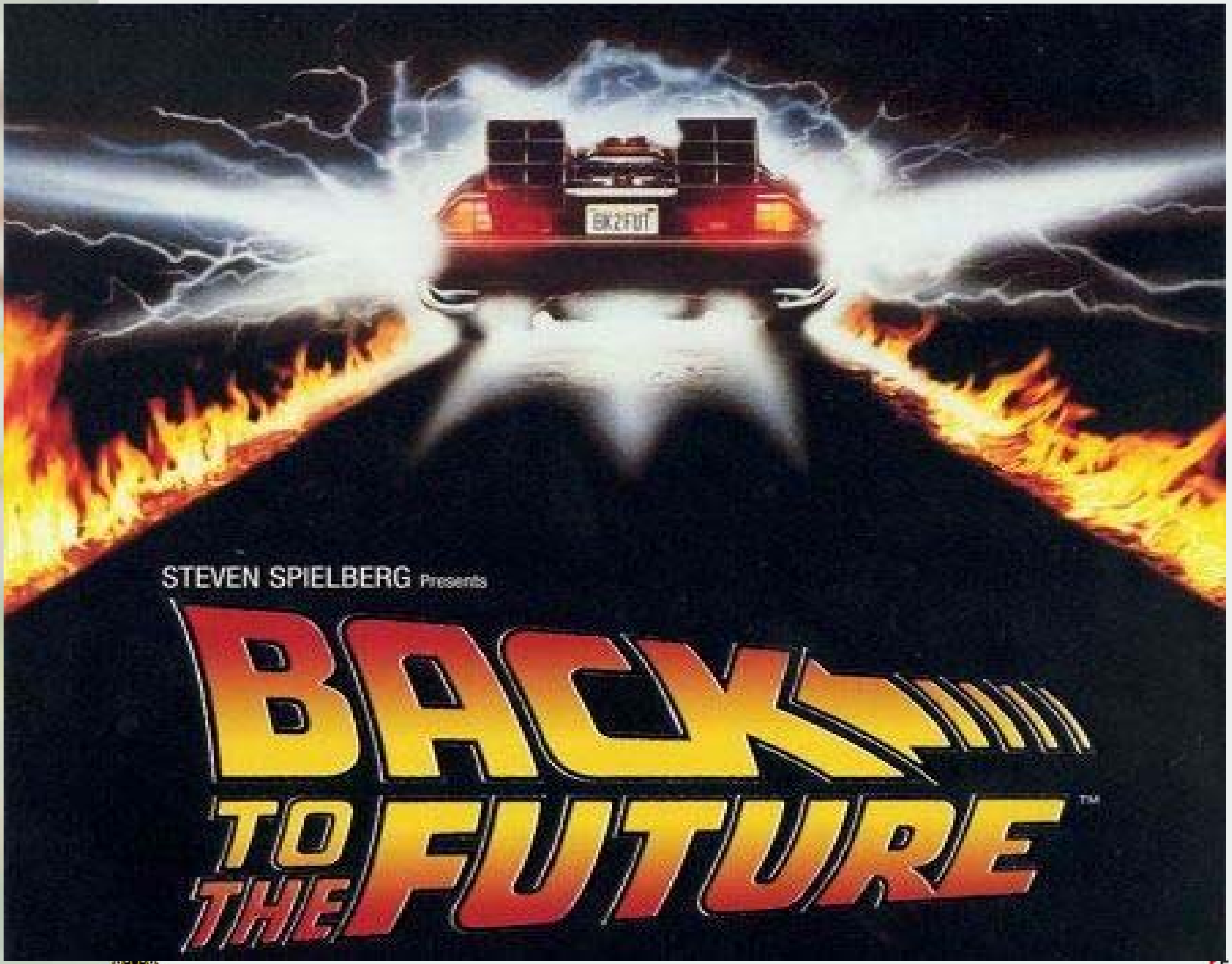
Solution for *machine may go down during processing*

1. MachineGoesDown
   - Schedule MachineRepaired
   - If Machine is busy
     - t_{down} =expected downtime
     - Remove event JobFinished from event list
     - Schedule event JobFinished t_{down} later than its old scheduled time

Wrong!!!!!

2. JobArrives
   - Advance time to t_{now}
   - If machine is idle and up
     - Set machine state to busy
     - Generate processing time p
     - *Check if machine will go down before* $t_{now}+p$
     - t_{down} =expected downtime
     - ScheduleJobFinished $t_{now}+p + t_{down}$
   - Otherwise put job in queue (update state of queue accordingly)

Wrong!!!!!



STEVEN SPIELBERG Presents
BACK TO THE FUTURE

Do not mess around with the future!

- The event list is the **future**!!!!!!
- State is the **present** **(you may include a little bit of past)**
- Performance measures logged in you program is the **past**
- Your simulation program must **not**:
  - Use information about future events
  - Reschedule events
  - Remove events from the event list
- *It acts as the real system, so does not use info on the future*

Do not mess around with the future!

- The only exceptions:
  - Going to a next moment in time by selecting next event from the event list (then this becomes your present)
  - Generating the future by scheduling a new event*:*
    - *Use random number from probability distribution*
    - *Do not combine with information from other events*

We use random numbers (from probability distributions) to reflect uncertainty about the future!!

It may invoke invalid modelling

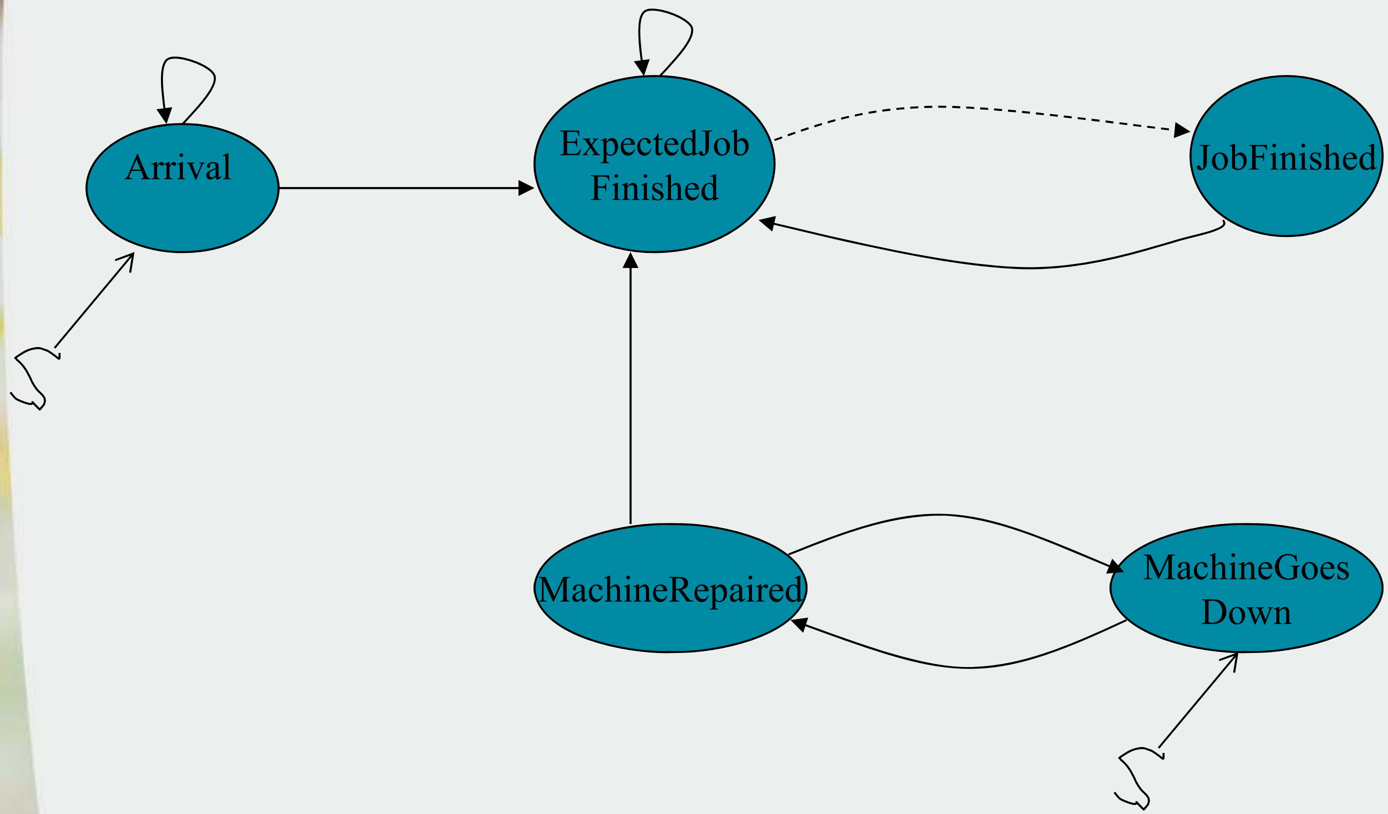
- Suppose in example,
  - job is removed and sent to competitor if it is not completed within 3 hours
  - there are additional repairmen
- Look again at:

  JobArrives:
  - Advance time to t_{now}
  - If machine is idle and up
    - Set machine state to busy
    - Generate processing time *p*
    - *Check if machine will go down before* $t_{now}+p$
    - t_{down} =expected downtime
    - ScheduleJobFinished $t_{now}+p + t_{down}$
  - Otherwise put job in queue (update state of queue accordingly)

It may invoke invalid modelling

- Job arrives at 10.00, has processing time of 5 mins, we `foresee' in 2 minutes the server will go down for 3.5 hours, so sent to competitor
- Bur after 1 hour downtime an additonal repairmen becomes available to help and the downtime is reduced to 2 hours
- Sent to competitor to early

A possible event graph



ExpectedJobFinish:

- If Machine down:
  - check how much work you have completed,
  - you know the remaining processing time (and you can use that in Machinerepaired to schedule ExpectedJobFinish)
- If Machine up:
  - check if you have been working on the job since its start,
    - if so schedule JobFinished,
    - otherwise scheduled ExpectedJobFinished at now + remaining processing time.

Wrap up

After this lecture:

- You can make more complicated simulation models
- You know about steps in a simulation study and validation
- You can start working on the assignment

Exam 2008 assignment 1: Maintenance of logistic equipment

- Service requests arrive according to Poisson process with an average of 28 per working day and are handled in FCFS order
- Service has two subsequent steps:
  1. Hardware specialist:
     - Travelling ~ N(1 hour, $(15 \text{ minutes})^2$)
     - Working ~ exp(30 minutes)
  2. Software specialist
     - Travelling ~ N(1 hour, $(15 \text{ minutes})^2$)
     - Working ~ exp(1 hour)

Draw event graph and indicate the time delays on the arcs for the following cases:

b) Request for Software Specialist is placed after Hardware Specialist is finished

c) Request for Software Specialist is placed 30 minutes after Hardware Specialist started travelling