SEN9110 Simulation Masterclass
Lecture 3. System Specification

Prof.dr.ir Alexander Verbraeck
a.verbraeck@tudelft.nl

Brightspace: SEN9110

Agenda of this lecture

- Questions about Ackoff's "System Model Mismatch" paper
- Questions about transformations in models / systems (Ashby)
- Some deeper definitions around systems, time and state (Klir)
- The notion of locality (Overstreet) and simulation model execution
- Inside a simulator

Homework of last week

- Read Ackoff's paper and try to answer the following question:

  What are the consequences of the mismatch he sketches for simulations of complex, socio-technical systems

  *What **is** the mismatch?*

Topic 1: Klir's System Specification

From: G.J. Klir, *Facets of Systems Science*, Kluwer, New York, 2001

Definitions of a “system”

Common sense definition:
A system is a set of certain things and a relation defined on that set.

$$S = (T, R)$$

- R — a relation defined on T (systemhood)
- T — a set of certain things (thinghood)
- S — a system

Definitions of a "system"
Properties of Relations

- **Relation** has the same set theoretical meaning as in the previous lecture, it is a **pair of two elements** from one or two sets.

- For R, a relation on T x T, with x, y, z $\in$ T, the following holds:
  - R is **reflexive** iff (x, x) $\in$ R for each x $\in$ T
  - R is **antireflexive** iff (x, x) $\notin$ R for each x $\in$ T
  - R is **symmetric** iff, for every x and y in T, whenever (x, y) $\in$ R, then also (y, x) $\in$ R
  - R is **antisymmetric** if and only if, for every x and y in T, whenever (x, y) $\in$ R and (y, x) $\in$ R, then x = y
  - R is **transitive** iff, for any three elements x, y, z in T, whenever (x, y) $\in$ R and (y, z) $\in$ R, then also (x, z) $\in$ R

Properties of Relations: Examples

Suppose T is a set of persons

- R is **reflexive** iff $(x, x) \in R$ for each $x \in T$
  - Example: R = "person x is family member of person y"
- R is **antireflexive** iff $(x, x) \notin R$ for each $x \in T$
  - Example: R = "person x is parent of person y"
- R is **symmetric** iff, for every x and y in T, whenever $(x, y) \in R$, then also $(y, x) \in R$
  - Example: R = "person x is a co-worker of person y"
- R is **antisymmetric** if and only if, for every x and y in T, whenever $(x, y) \in R$ and $(y, x) \in R$, then $x = y$
  - Example: R = "x paid the restaurant bill of y"
- R is **transitive** iff, for any three elements x, y, z in T, whenever $(x, y) \in R$ and $(y, z) \in R$, then also $(x, z) \in R$
  - Example R = "person x is taller than person y"

Definitions of a "system"
Properties of Relations

One can distinguish the following classes of relations:

Equivalence relation : reflexive, symmetric, transitive
example: "x has the same birthday as y"

Compatibility relation : reflexive, symmetric
example: "countries x and y share a border"

Partial ordering relation : reflexive, antisymmetric, transitive
example "$x \geq y$"

Strict ordering relation : antireflexive, antisymmetric, transitive
example: "$x > y$"

Definitions of a "system"

In a **graduate school** *object* , one can distinguish such things as *graduate students*, *professors*, *research subjects*, and *master projects*. A research subject can be interesting for a graduate student or a professor. A professor can supervise a graduate student in a Master Project if they are both interested in the same research subject. A professor cannot supervise more than 3 graduate students.

Is the graduate school expressible in terms of the common sense definition of systems?

A set of graduate students G
A set of professors P
A set of research subjects O
A set of master projects M
A supervision function Sp such that $m = Sp(g,p,o)$ with $m \in M$, $g \in G$, $p \in P$, $o \in O$

$$\mathbf{S = (T=\{G,P,O,M\},\ R \subseteq (G \times M \times \{0,1\}) \times (M \times O) \times (O \times P) \times (O \times G) \times (P \times M \times [0,3]))}$$

System Conceptual Frameworks

The common sense definition is useful for merely identifying what is a system and what is not. But the nature of the possible relations within a system is not specified yet.

For further characterization, one needs a **system framework** that gives precise mathematical axioms for system concepts and categories. Some famous frameworks are:

- **Mesarovic and Takahara's** General Mathematical System Theory
- **Wymore's** Tricotyledon theory of system design
- **Klir's** General System Problem Solving framework
- **Zeigler's** Framework for System Specification

*A distinction can be made between frameworks with a deductive or **inductive** inclination.*

Epistemological categories of System (Levels of System Specification)

An *object*, as a part of the world, can be *investigated* by an *investigator* with different *instruments,* this investigation brings gradual levels of *knowledge.* As such, the systems constructed by the investigator to account for the object can be of various types. The epistemological category of a system (*in Klir's framework*) corresponds with the type of knowledge involved in the representation of a given object.

A system at each level subsumes all information contained at the lower levels.

\-

Source System
Data System
Generative System
Structure System
Meta System

\+

A very similar hierarchy can be found in Zeigler's framework under the term ***"Levels of System Specification"***

The Source System

The **Source System** (*primitive system, experimental frame*).

This category of system merely gives the variables, their state sets, and their support sets. The particular values of variables or relations between variables are known at this level.

Support is the indexing variable(s), in case of simulation that is **time**. A system whose variables are distinguished (input, output) is called a **directed system**, otherwise, the system is **neutral**.

Example: System of Corona tests at a certain test location
Input variables: number of people with complaints, available number of tests
Output variables: waiting time, capacity utilization
Support: time
(directed system)
For each variable, its unit and range set (state set) would be described

The Data System

The **Data System** supplements the source system with actual state variables. It does not inform on how the state variables are related with each other.

Time (day) is the **support** in the example below.

Example:

Time (day)	1	2	3	4	5	6	...
Nr of people	180	190	210	220	235	252	...
Available tests	200	200	200	200	250	250	...
Avg waiting time	0.1	0.1	0.15	0.2	0.25	0.18	...
Capacity utilization	90%	95%	97%	98%	96%	95%	...

The Generative System

The **Generative System** represents the object as a **support-invariant (*time invariant*) relation** between variables.
Example:

$$v(t+1) = 1 - v(t).$$

Examples: DEVS atomic models, Markov chains are generative systems.

If a given time-invariant relation between variables can be expressed by a ***function*** *(i.e. mapping, single-valued transformation)* the system is **deterministic**, aka a **determinate machine**. Otherwise it is non-deterministic.

Simulation is a process of **data generation** from a **time-invariant representation** of an object of interest, ***(a generative system).***

The Generative System (2)

The generative system can **reproduce** the behavior of the system (up to a certain extent). If we reproduce state over time, we are using a **simulation** as the generative system.

Example generative systems we could use for the testing example:

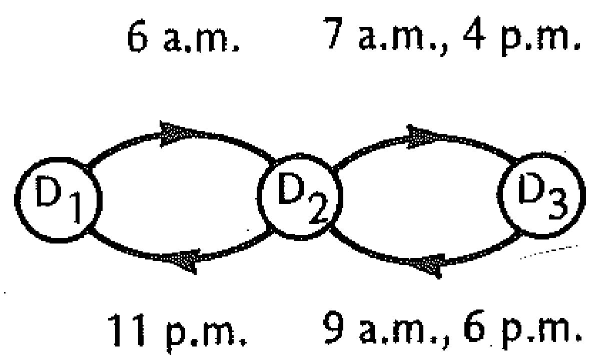
- *A number of regression models that link the variables to time*
- *A data analytics model that links the variables to time and each other*
- *An SD model that describes the system and (re)produces state over time*
- *An Agent-based simulation model that describes the system and (re)produces state over time*
- *A discrete-event simulation model that describes the system and (re)produces state over time*
- *Etc.*

Structure Systems and Meta Systems

The next two levels are integrative.

- **Structure systems** are the result of the coupling relation between well defined systems with input.

- **Meta Systems** have the capability of switching from one behavior to another, as specified in an invariant way on a chosen support.



REPLACEMENT PROCEDURE r

Structure Systems and Meta Systems

*Examples of **structure systems** for this case:*

- *For the DES or ABM model we could model the persons and the test facility as separate subsystems and link these.*
- *This could be expanded with a submodel for the appointment system or a subsystem for the lab.*
- *Going even further, the test facility could be broken down into smaller subsystems; note that you would have to create a source system, data system and generative system for each of the subsystems **and** guarantee that the output of one subsystem can function as the input for the next subsystem!*

*Example of a **meta system** for the testing case:*

- *A different behavior set for compulsory testing (e.g., for certain flights at Schiphol Airport) versus voluntary testing. Certain subsystems would behave differently after applying this "switch".*

Topic 2: How Discrete Event Simulation Works

Based on: T.J. Schriber, D.T. Brunner. "Ch. 24 How Discrete-Event Simulation works". In: J. Banks (Ed.), *Handbook of Simulation*. Wiley, New York, 1998

Reading for this class: T.J. Schriber, D.T. Brunner, & J.S. Smith. Inside Discrete-Event Simulation Software: how it works and why it matters. Proceedings WSC 2015, L. Ylmaz at al. (Eds.). IEEE, 2015, pp 1-15.

Discrete Event Simulation: Worldview

- A simulation language is based on a **worldview**. How does one see the world?
- Discrete simulation language worldviews interrelate with the concept of "**locality**" (Overstreet, 1986). How does one group behavioral logic?
- Discrete simulation language follow either the **process interaction** worldview, **event scheduling** worldview or **activity scanning** worldview

Discrete Event Simulation: Worldview

- **Event scheduling** provides locality of **time**: each event routine describes related actions that may all occur at one single instant
- **Activity scanning** provides locality of **state**: each activity routine describes all actions that occur because a particular state is reached
- **Process interaction** provides locality of **object**: each process routine describes the entire action of a particular model object

Event Scheduling

Clock:=0

Find next segment to execute

Change the clock

Execute the selected segment

Ready?

no

yes: print results

Process interaction (e.g. post office)

Customer process

Entry event for customer process

Customer process waits in queue till selected

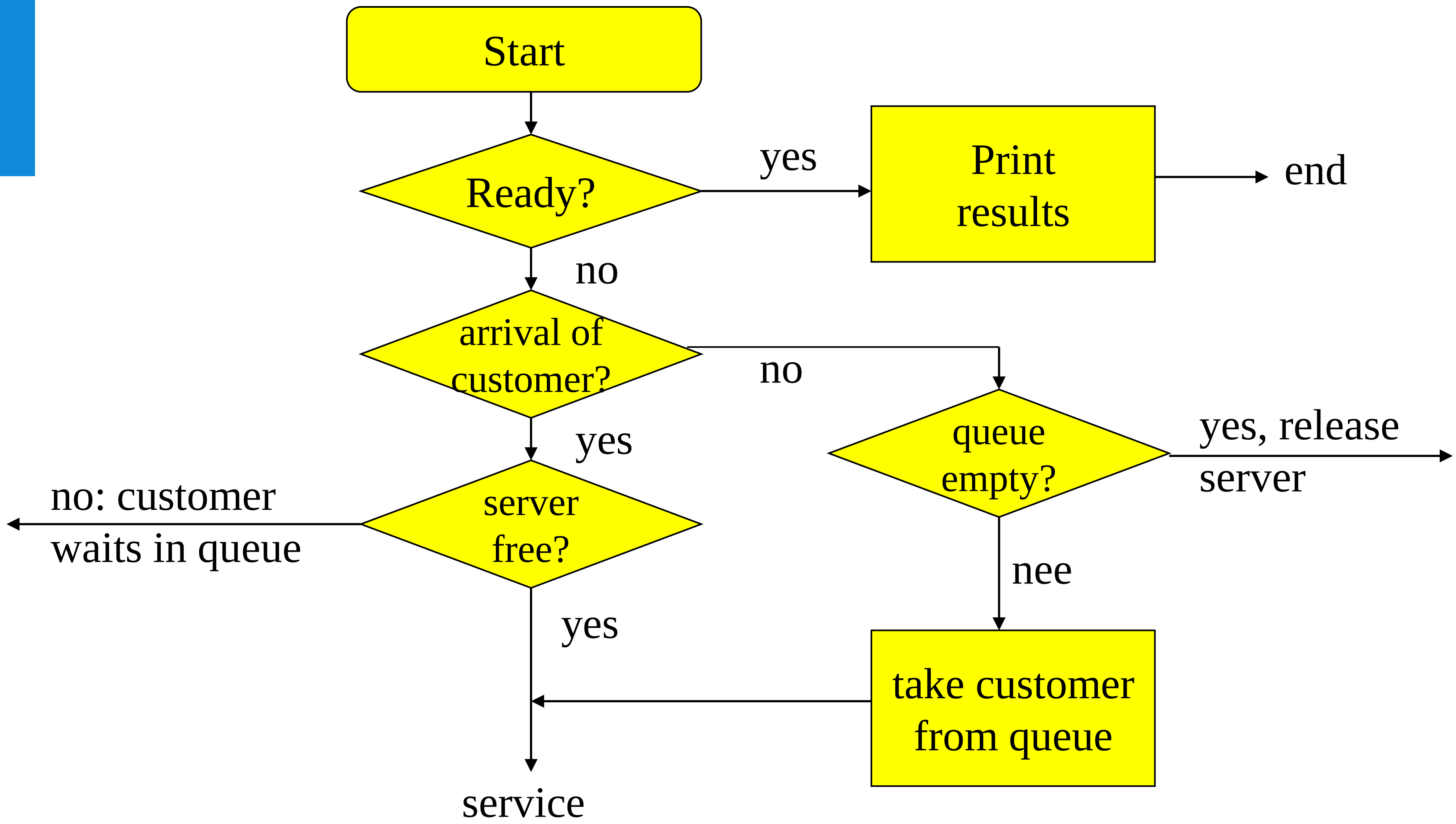
Put the clerk on “occupied”

Customer process waits during service time

Put the clerk on “free”

Leave event for customer process

Activity scanning (post office example)

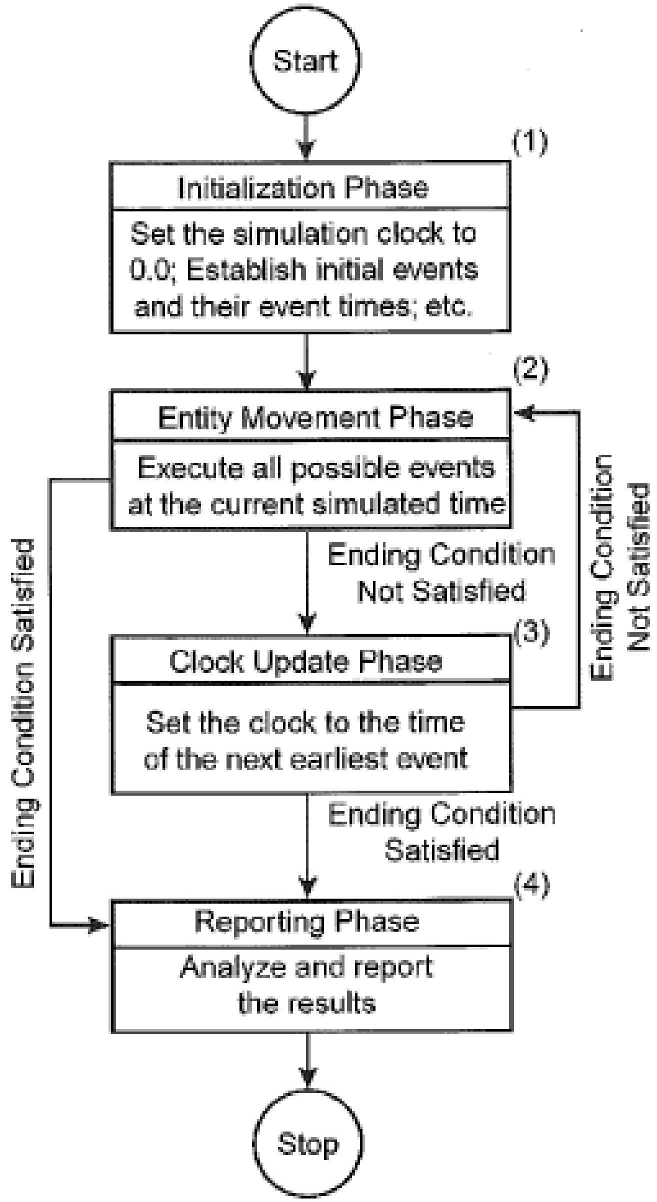


Inner working of DES Software

Execution of a replication:

4 phases

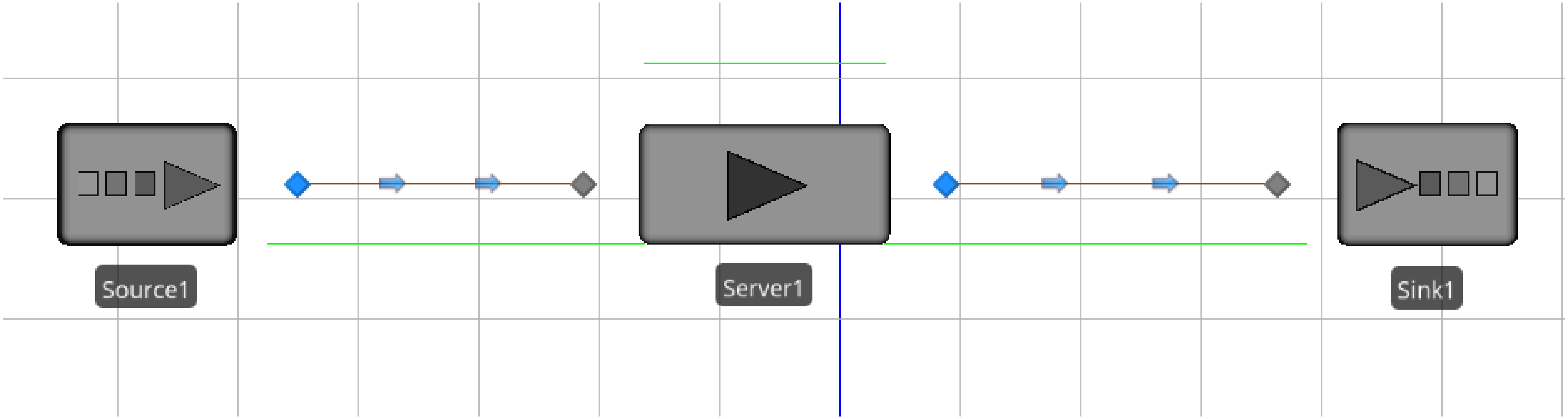
- Initialization
- Entity movement
- Clock update
- Reporting



Picture: Schriber & Brunner

Event Calendar in practice (Simio)

Simple model (queuing system with 1 server)

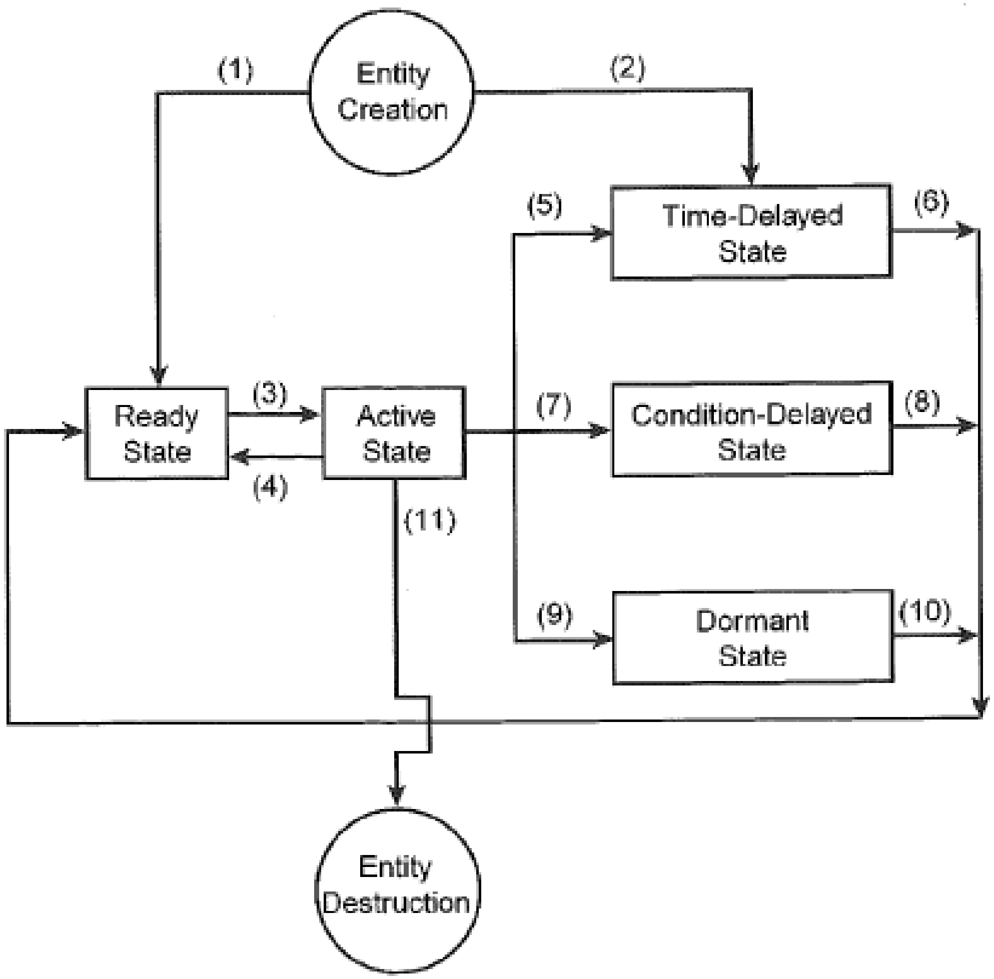


Inner working of DES Software

Entity states:

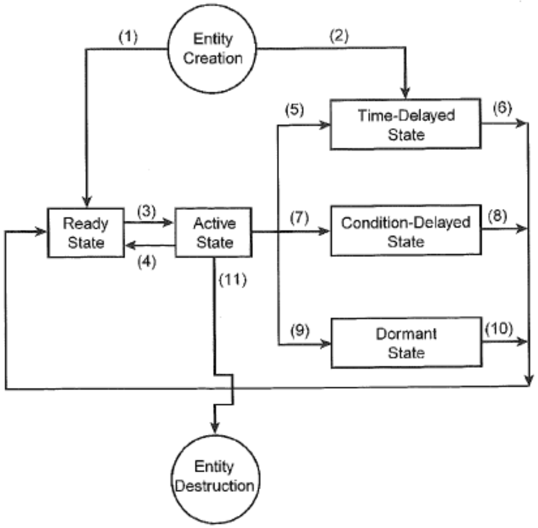
- time delayed
- condition delayed
- dormant
- active
- ready

Examples?



Picture: Schriber & Brunner

Inner working of DES Software



Entity states and *examples:*

- time delayed: waiting for <u>known</u> time

  activity takes 10 minutes
- condition delayed: waiting for <u>condition</u>

  in queue, waiting for the server to become free
- dormant: waiting for <u>external</u> event

  e.g., input from modeler
- active: only <u>one entity</u> at a time (event!)

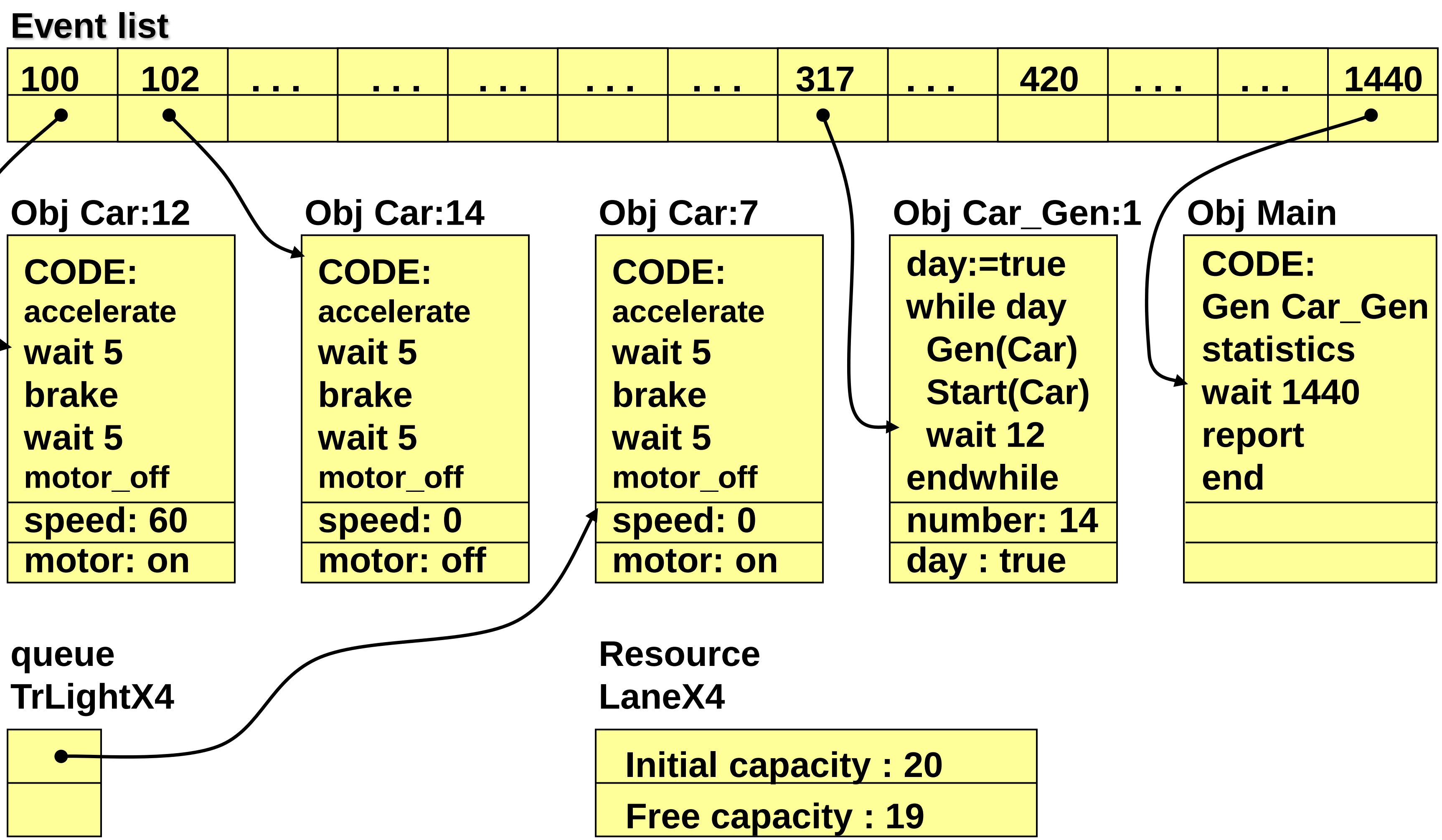
  entity has been generated and moves to server
- ready: <u>not yet active</u>

  other entity that can execute at this instant but has to wait

Picture: Schriber & Brunner

Discrete event simulation

The ‘future event list’ (traffic example)



Model Trace (Simio)

Trace - Tracing to C:\Users\alexandv\Documents\Model_trace.csv

Time (Hours)	Entity	Object	Process	Token	Step	Action
0	-	-	-	0	-	Object 'DefaultEntity.11' created.
	DefaultEntity.11	Source1	OnEntityArrival	1	[Transfer] ToProces...	Entity 'DefaultEntity.11' transferring from '[FreeSpace] Source1' into station 'Source1.Processing'.
	Source1	Source1	OnEntityArrival	0	[End]	Process 'Source1.OnEntityArrival' ended.
	DefaultEntity.11	Source1	OnEnteredProcessing	2	[Begin]	Process 'Source1.OnEnteredProcessing' started.
					[EndTransfer] IntoP...	Entity 'DefaultEntity.11' ending transfer into station 'Source1.Processing'.
					[Transfer] ToOutpu...	Entity 'DefaultEntity.11' transferring from '[Station] Source1.Processing' into station 'Source1.Outp...
			OnEntityArrival	1	[End]	Process 'Source1.OnEntityArrival' ended.
			OnEnteredOutputBuffer	0	[Begin]	Process 'Source1.OnEnteredOutputBuffer' started.
					[EndTransfer] Into...	Entity 'DefaultEntity.11' ending transfer into station 'Source1.OutputBuffer'.
					[Transfer] ToOutpu...	Entity 'DefaultEntity.11' transferring from '[Station] Source1.OutputBuffer' into node 'Output@Sou...
		Output@Source1	OnEnteredFromAssociatedObj...	1	[Begin]	Process 'Output@Source1.OnEnteredFromAssociatedObject' started.
					[Fire] EnteredEvent	Firing event 'Output@Source1.Entered'.
					[Transfer] ToOutbo...	Entity 'DefaultEntity.11' transferring from '[Node] Output@Source1' onto link 'TimePath1'.
		TimePath1	OnEntered	3	[Begin]	Process 'TimePath1.OnEntered' started.
					[Assign] EntityMove...	Assigning state variable 'TimePath1.Token.3.ReturnValue' the value '0.0166666666666667'.
						Assigning state variable 'DefaultEntity.11.Movement.Rate' the value '142.8' Meters per Hour.
					[EndTransfer] Onto...	Entity 'DefaultEntity.11' ending transfer onto link 'TimePath1'.
					[End]	Process 'TimePath1.OnEntered' ended.
		Source1	OnEnteredProcessing	2	[End]	Process 'Source1.OnEnteredProcessing' ended.
			OnEnteredOutputBuffer	0	[End]	Process 'Source1.OnEnteredOutputBuffer' ended.
		Output@Source1	OnEnteredFromAssociatedObj...	1	[End]	Process 'Output@Source1.OnEnteredFromAssociatedObject' ended.
0.00350140056022409			OnExited	1	[Begin]	Process 'Output@Source1.OnExited' started.
					[Fire] ExitedEvent	Firing event 'Output@Source1.Exited'.
					[End]	Process 'Output@Source1.OnExited' ended.
0.00655278052637456	-	-	-	0	-	Firing event 'Source1.EntityArrivals.Event'.
	Source1	Source1	OnEntityArrival	1	[Begin]	Process 'Source1.OnEntityArrival' started.
					[Create] Entities	Creating '1' object(s) of entity type 'DefaultEntity' using creation method 'NewObject'.
	-	-	-	0	-	Object 'DefaultEntity.12' created.
	DefaultEntity.12	Source1	OnEntityArrival	0	[Transfer] ToProces...	Entity 'DefaultEntity.12' transferring from '[FreeSpace] Source1' into station 'Source1.Processing'.
	Source1	Source1	OnEntityArrival	1	[End]	Process 'Source1.OnEntityArrival' ended.
	DefaultEntity.12	Source1	OnEnteredProcessing	2	[Begin]	Process 'Source1.OnEnteredProcessing' started.
					[EndTransfer] IntoP...	Entity 'DefaultEntity.12' ending transfer into station 'Source1.Processing'.

Lists in simulation language

- Several lists contain (pointers to) the types of entities
- Example of (many) condition-delayed lists?

Entity State	Generic Name of Entity List(s)	Comments
Active	None	There is a maximum of one active entity.
Ready	Current events list	There is only one such list.
Time-delayed	Future events list	There is only one such list.
Condition-delayed	Delay list	There are potentially many such lists.
Dormant	User-managed list	There are potentially many such lists.

Table: Schriber & Brunner

Placing an event [token in a process] on the future event list (Simio)

0.0166666666666667	DefaultEntity.11	Server1	OnEnteredInputBuffer	2	[Seize] Server
					[Transfer] ToProces...
		TimePath1	OnReachedEnd	1	[End]
		Input@Server1	OnEnteredToAssociatedObject	0	[End]
		Server1	OnEnteredProcessing	3	[Begin]
					[EndTransfer] IntoP...
					[Decide] IfProcessT...
					[Delay] ProcessingT...

Server1	OnEnteredInputBuffer	2	[Seize] Server	Object 'DefaultEntity.11' checking availability to seize '1' resource(s) of resource type '[ParentObje...
				Object 'DefaultEntity.11' found '1' currently available resource(s) of resource type '[ParentObject] ...
				Object 'DefaultEntity.11' seized '1' capacity unit(s) of resource 'Server1'.
				Object 'DefaultEntity.11' completed seize of '1' resource(s) of resource type '[ParentObject] Server1'.
			[Transfer] ToProces...	Entity 'DefaultEntity.11' transferring from '[Station] Server1.InputBuffer' into station 'Server1.Proc...
TimePath1	OnReachedEnd	1	[End]	Process 'TimePath1.OnReachedEnd' ended.
Input@Server1	OnEnteredToAssociatedObject	0	[End]	Process 'Input@Server1.OnEnteredToAssociatedObject' ended.
Server1	OnEnteredProcessing	3	[Begin]	Process 'Server1.OnEnteredProcessing' started.
			[EndTransfer] IntoP...	Entity 'DefaultEntity.11' ending transfer into station 'Server1.Processing'.
			[Decide] IfProcessT...	Token branching on condition 'ProcessType==Enum.TaskProcessType.SpecificTime'.
				Token sent to 'True' exit.
			[Delay] ProcessingT...	Delaying token for '0.00345282753372506' Hours until time '0.0201194942003917' Hours.

Future event calendar of M/M/1 model (Arena)

```
100.0 Hours>view cal

***  Entities on current events chain: 0 ***


***  Entities on future events heap  : 2 ***


***  Entity 92 to arrive at time 105.01777 at block

      1 2$              CREATE,1,HoursToBaseTime(0.0),Entity 1:
                            HoursToBaseTime(EXPO(1)):
                            NEXT(3$);

Entity.SerialNumber   = 0
Entity.Type           = 0
Entity.Picture        = 0
Entity.Station  (M)   = 0
Entity.Sequence (NS)  = 0
Entity.JobStep  (IS)  = 0
Entity.CurrentStation = 0
Entity.PlannedStation = 0
Entity.CreateTime     = 98.919317
Entity.VATime         = 0.0
Entity.NVATime        = 0.0
Entity.WaitTime       = 0.0
Entity.TranTime       = 0.0
Entity.OtherTime      = 0.0

***  Entity 1 to cause end of replication at time INFINITY
```

Homework

- Read DEVS papers (class 4)

- Start working on term papers
- Finalize installation of simulation software
- Run demos and experiments in simulation software
- Build first model for Assignment 1

- First meeting in week 3:
    - longlist of papers that have been read / you want to read
    - first observations based on the readings
    - narrow down topic
    - simulation software progress