SEN9110 Simulation Masterclass
Lecture 1: Introduction

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

Brightspace: SEN9110 (2024-2025 Q1)

Agenda of this lecture

- Overview of masterclass: topics, interests, etc.
- Overview of schedule, assignments, group activities
- Review of simulation (discrete event, agent-based)
- Foundation: Systems Theory

What will you learn during this course

- Foundation of Simulation in Systems Theory
- Formal Theory of Modeling and Simulation
- The DEVS Formalism
- Object-oriented Simulation
- Distributed Simulation and HLA
- Multi-Paradigm Simulation
- Multi-Resolution Simulation
- Real-Time Simulation and Emulation
- Interactive Simulation and Gaming
- Simulation Languages (including practice with a new language)

- In-depth study of a special topic in simulation

What kind of master thesis projects relate to the simulation masterclass

Problem owner	*Example MSc projects at the Policy Analysis section*
Allseas	Simulating options for offshore windfarm installation
SWECO	Optimizing traffic patterns in the City of Groningen
GGD	Simulating cold vaccine distribution in The Netherlands
Dutch Customs	Logistics preparation for extra scanning of 1 million packages/day
KLM Cargo	Dock and yard management simulation for Air Cargo operations
Port of Rotterdam	Port call efficiency optimization using discrete-event simulation
RHDHV	Supply Chain Visibility for the FMCG Industry
Erasmus MC	Reducing inter-regional transfers in neonatal care with simulation
Picnic	Simulating a new fully automated logistics distribution center
PostNL	Time slot planning for same-day parcel delivery
Heineken	Studying options for rail and barge for inland transportation
Dutch Police	Modeling the behavior of fugitives fleeing from a crime scene
Systems Navigator	Lock Systems Operations & Maintenance
Schiphol	Optimizing the baggage backbone for cross-platform traffic

Special topics for group work

- Cloud-based simulation (all)
- Crowd simulation (all)
- Cyber-physical Systems (all)
- Financial Markets Simulation (all)
- Input analysis and input modeling for simulation (all)
- Live, Virtual, Constructive simulation (all)
- Modeling human behavior (all)
- Output analysis for simulation (all)
- Simulation and forecasting (CoSEM, EPA)
- Simulation and the Digital Factory (TIL, TEL, PEL)
- Simulation and Optimization (all)
- Simulation and Virtual Reality (all)
- Simulation for ICT and Telecommunication (CoSEM)
- Simulation for Policy Analysis (CoSEM, EPA)
- Simulation for strategic decision making (CoSEM, EPA)
- Simulation in the Energy Sector (CoSEM)
- Simulation of Logistics Systems (TIL, TEL, PEL)
- Simulation of Transportation Systems (TIL, TEL, PEL)
- Supply Chain Simulation (TIL, TEL, PEL)
- Verification, Validation and Accreditation (all)
- Web-based simulation (all)

Week 2 to 6: find and study papers, write, and discussions with teacher
Week 7: finish term paper and preparation for brief summary lecture
Week 8: brief (10 minute) summary lecture by each group + questions

Groups of 4 students. Different groups have to take different topics. You are also allowed to provide your <u>own topic</u>.

Material

- Reading materials: Some core papers to get started are digitally available via Brightspace / website
- For each group: Specific papers to prepare for the presentation in week 8 and the term paper, available via Brightspace / website
- Each group will be coached for the special topic
- There will be at least 3 meetings of 30 minutes per group with the teacher; we can focus on the paper and on the simulation package

Studying a new simulation language

- **Plant Simulation** (full license available, GUI)
- **Enterprise Dynamics** (full license, GUI)
- **AnyLogic** (student license available, GUI)
- **JaamSim** (open source, Java-based + GUI)
- **Simio** (full license available, GUI)

- **Arena** (student license available, GUI)
- **FlexSim** (trial license available, GUI)
- **ProModel** (demo license available, GUI)
- **AutoMod** (student license available, GUI)
- **ExtendSim** (student license available, GUI)

- **DSOL** (open source, Java-based, code)
- **pydsol** (open source, Python – in development)
- **Salabim** (open source, Python-based, code)
- **SimPy** (open source, Python, code)
- **MESA** (open source, Python, ABM, code)
- **NetLogo** (open license; ABM, code)
- **Repast Symphony** (open, Java, ABM, code)
- **GAMA** (open source, Java, ABM, code)

- **DESMO-J** (open source, Java, code)
- **CiwPython** (open, Python, queuing, code)

Week 1: group formation; choosing the simulation package

Week 2: installation, learning about the package, running demo models

Week 3: implementing a small queueing model

Week 4: implementing a small model with animation

Week 5: carrying out a more extensive experiment with a model

Week 6: scoring the simulation package using a comparison sheet

Week 7: discussion of the simulation package i.r.t. others and comparing packages

Group work

- Make sure you set aside enough time together every week
  - the class does take **16-20 hours/week**, counting on the fact that you also divide the work properly within your group of 4 students!
- Divide tasks within the group
  - e.g., 4 sub-topics to be covered in the paper
  - coding simulation assignment, experimenting, making slides
  - co-referencing each others' work for quality
  - often working with pairs of two students is quite effective

Grade and Group Size

Grade is based on:

- Written Exam, open book (50%)
- Discussion Paper and Group Lecture (30%)
- Simulation package assignments and report (20%)

Group size:

- **Normal group size is 4**
- I can allow groups of 5 in special cases; ask me first, though
- Groups of 3 will also work fine, but it is harder work
- Groups of 1 or 2 are discouraged, but students have succeeded to complete the course in the past. Groups of 1 or 2 will get 0.5 point extra for their discussion paper and simulation assignments.

Time spending

It is a very **intensive course!**

• General papers to study in-depth	4 h/w = 28 h	(1 EC)
• Lectures	4 h/w = 28 h	(1 EC)
• Exam preparation and exam / essay	week 9: 13 h	(0.5 EC)
• Term paper preparation, writing	40 h	(1.5 EC)
• Term paper presentation preparation + class	week 8: 4 h	
• Studying simulation package + assignments	24 h	(1 EC)
• Small report about simulation package	week 6/7: 3 h	
TOTAL:	140 h = 5 ECTS	

Communication

- All course materials are on https://simulation.tudelft.nl/SEN9110
- All papers to study are there, and also in the SEN9110 Reader

- All announcements will be on Brightspace (or mail if it is urgent)

- Handing in of Assignments and your marks will be in Brightspace

- If there is anything important, **mail me**

- Most classes are Mo and Fr at TBM Room C (Jaffalaan 5)
  - a few classes are in ECHO and at CEG
  - no recordings of classes

Topic 0: What you already know
Revisiting previous simulation courses

Q1: Who do we have in class? Which program?
Q2: What is your expectation of the course?
Q3: Any special topics you want addressed?

Why simulate?

- Identify bottlenecks in systems
- Evaluate alternative systems
- Prioritise investment decisions
- Compare performance of different production plans
- Train operators in structured decision making

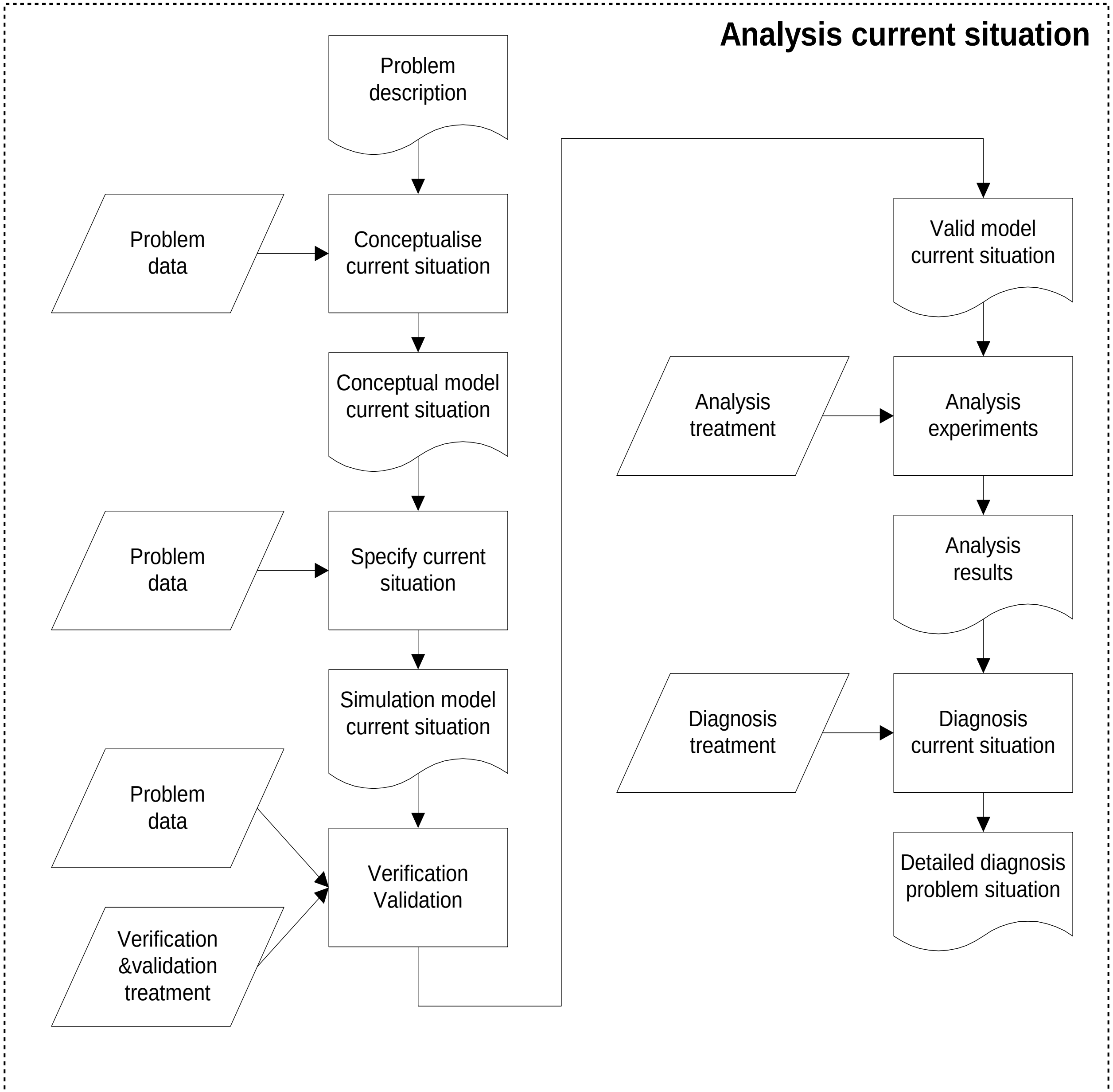
} quantitative analysis, what-if analysis

Definition of a simulation study

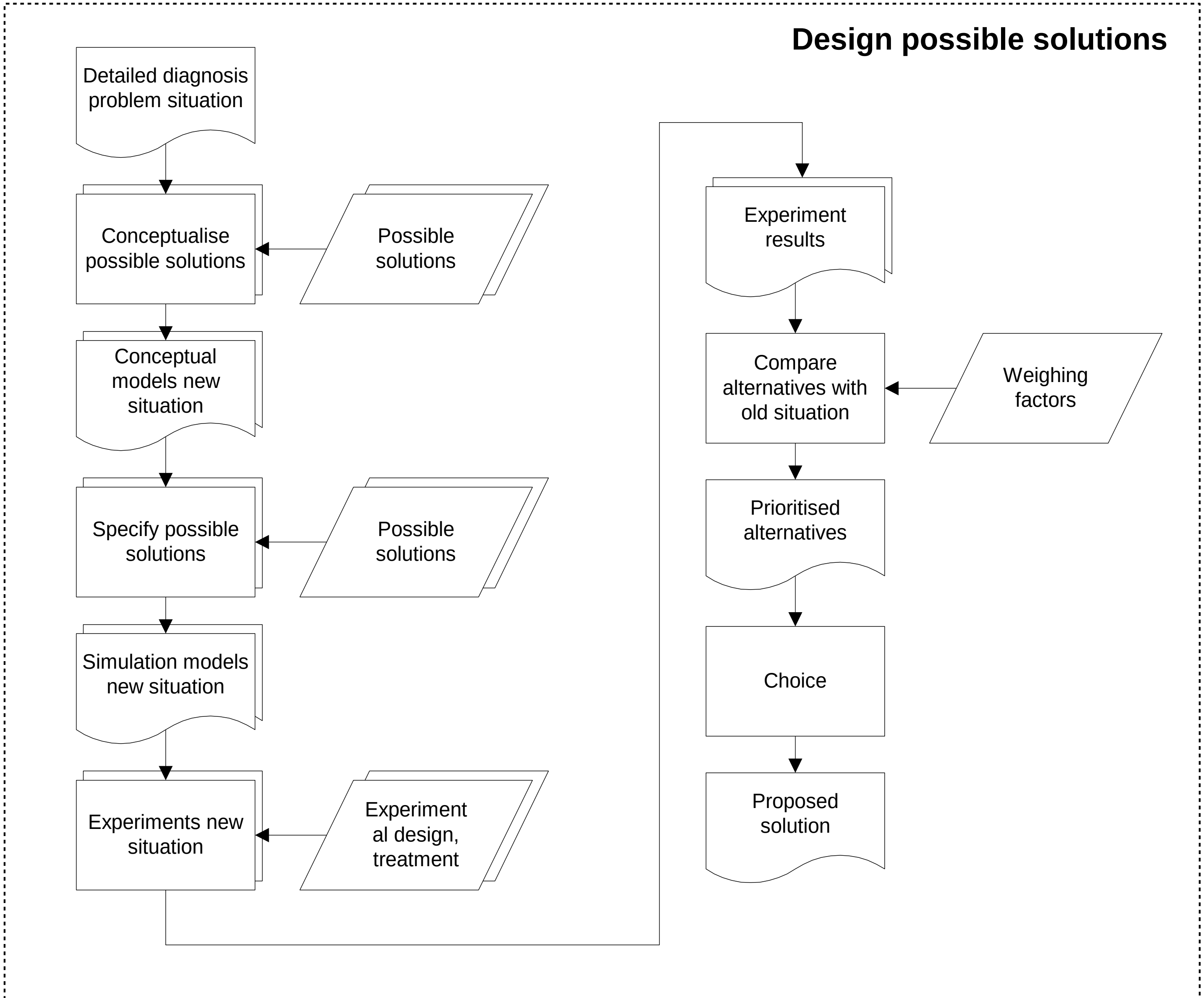
*The **process** of designing a **model** of a real **system** and conducting **experiments** with this model for the purpose either of **understanding** the **behaviour** of the system or of **evaluating** various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system.*

(Shannon, 1975)

Simulation project plan I
Analysis phase



Simulation project plan II
Design phase



Conceptualisation

Output:

- A number of conceptual models (data-void, instance independent) that can be used to describe the system

- Demarcation of the system
- Language by which the system can be described:
  - object based (object model)
  - process based (process model)
  - time based (event list)

Specification

Output:
- Working computer model that can be experimented with

- Specify objects and processes
- Reduction of the model
- Build simulation model

Verification / validation

Output:

- Simulation model that is correct and is a good representation of the real system

- Verification (correct representation of conceptual model)
- Validation (conform reality):
  - structural: testing of hypotheses on the model
  - replicative: compare values to real system values
  - expert: analysis of the model by experts

Treatment specification

Output:

- The treatment under which the system, or the model of it, is experimented with or observed

- Number of runs
- Run length
- Start-up time
- Values of parameters

Analysis and diagnosis

Output:

- Results of analysis and diagnosis of the experiments with the model of the current situation

- Current bottlenecks (long queues, idle resources, etc.)
- Sensitivity analysis for stability of results

Solution finding and experimentation

Output:

- A number of alternatives and evaluation of these alternatives on a number of performance indicators

- Sources: sensitivity analysis, bottlenecks, experts, design theories
- Experimental design
- Testing of hypotheses: changes significant?

Animation

- Animation gives graphical insight into the model

- Two kinds of animation:
  - insight into the execution of the model code (model representation)
  - insight into the meaning of the model and into the results (system representation)

Model data

- Data for:
  - generators of items
  - process durations in the model
  - resource availability
- How to gather data:
  - historical sources
  - expert opinions
  - measurements
  - analogous systems
- Data gathering is time consuming

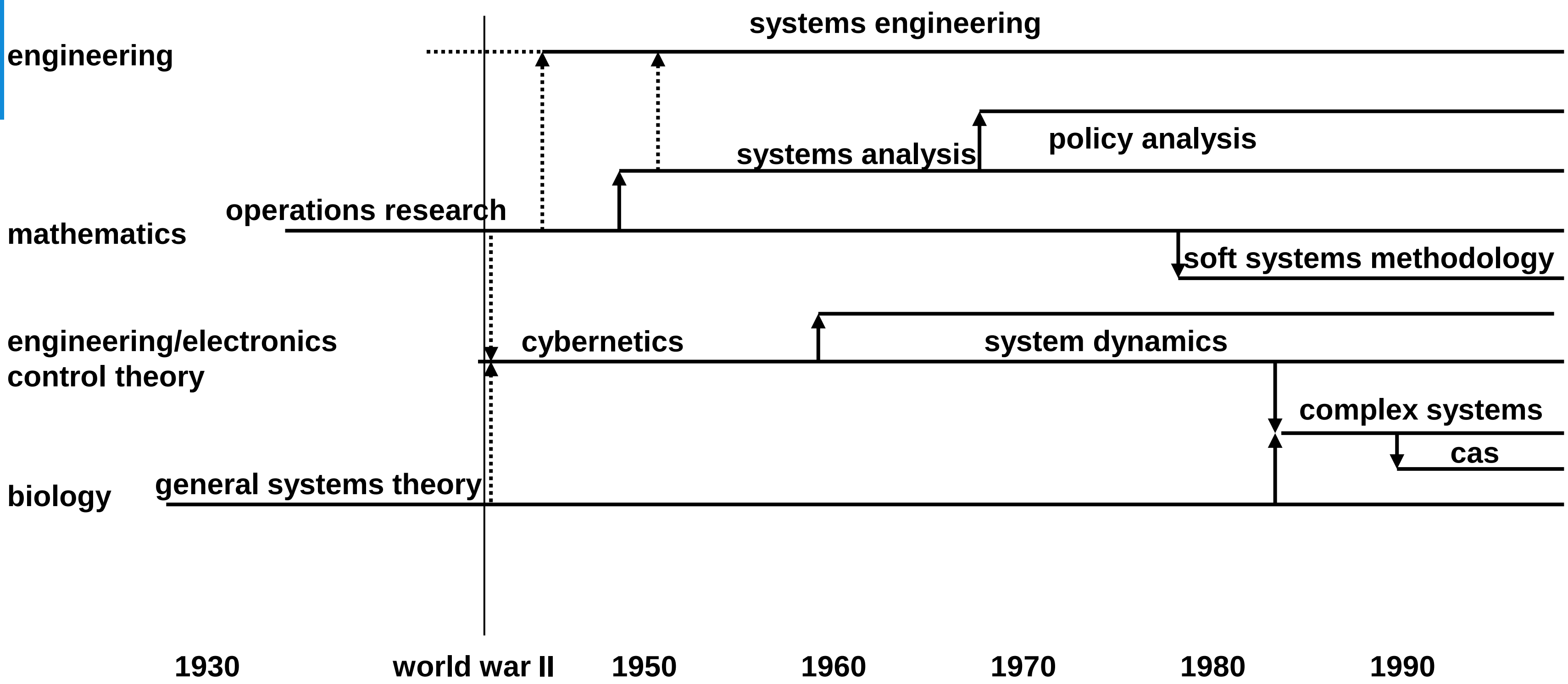
Limitations to current knowledge

- BSc simulation courses / EPA1352 / TIL courses
  - simulation environment Arena, Simio, Salabim, MESA, or Netlogo
  - develop a simulation model for one-time use
  - model a flow of activities of simple systems

- Missing :
  - (strict) terminology and concepts
  - understanding and applications of language to describe systems
  - ability to integrate system complexity through multi-formalism modeling and simulation
  - understanding of inner working of simulation languages
  - insight into (working of) other simulation languages

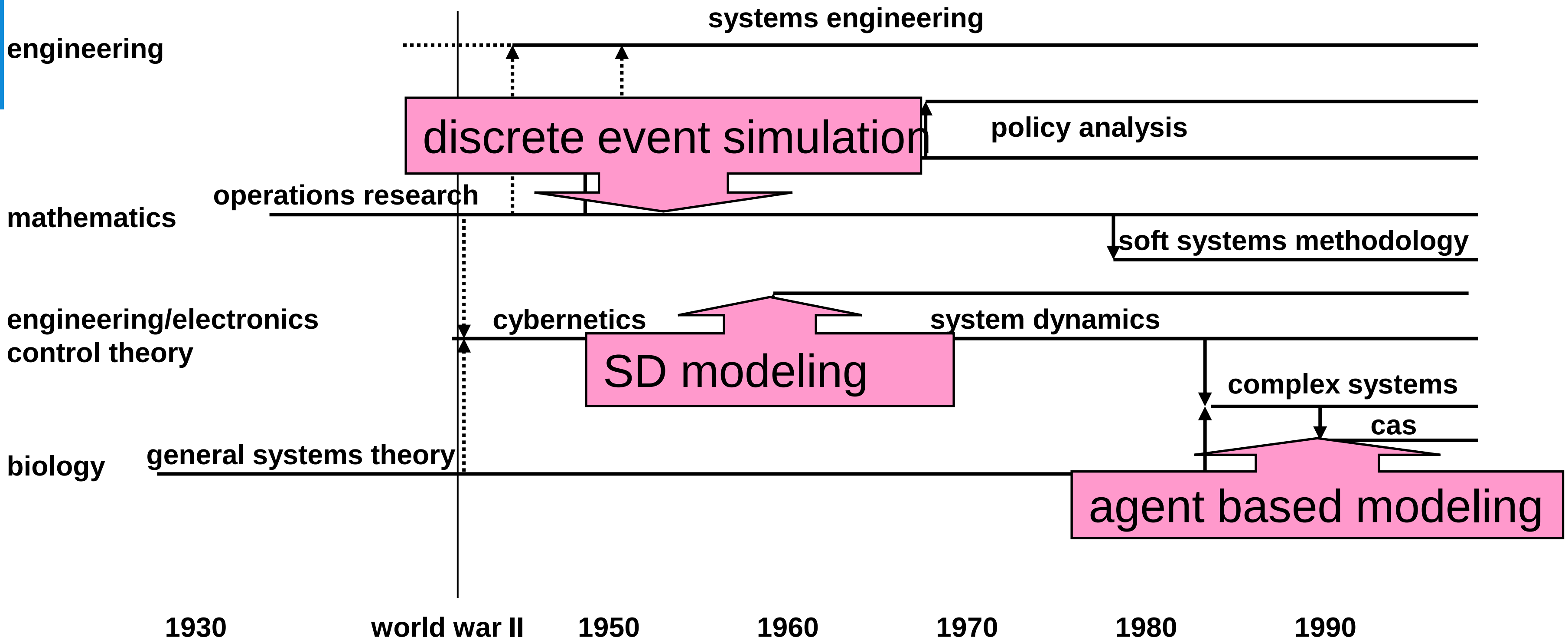
Topic 1: History and Terminology

Development of systems thinking



Slide courtesy Els van Daalen, TU Delft

Simulation modeling methods



Slide courtesy Els van Daalen, TU Delft

So what is the difference between discrete and continuous simulation?

What are the differences?

Where does agent-based simulation fit in?

DESS

The *Differential Equation System Specification (DESS)* assumes that the time base is continuous and that the trajectories in the system database are piecewise continuous functions of time. The models (system specifications) are expressed in terms of differential equations (*ordinary differential equations [ODESS]* and/or *partial differential equations [PDESS]*) that specify change rates for the state variables. The corresponding simulation concept is that of numerical solvers, numerical integrators and differential algebraic equation (DAE) solvers.

Ref: Hild, 2000

DTSS

The *Discrete Time System Specification* (DTSS) assumes that the time base is discrete so that the trajectories in the system database are sequences anchored in time. The models are expressed in terms of difference equations that specify how states transition from one step to the next. A forward marching timestepping algorithm constitutes the associated simulator.

Ref: Hild, 2000

DEVS

The *Discrete Event System Specification (DEVS)* assumes that the time base is continuous and that the trajectories in the system database are piecewise constant, i.e., the state variables remain constant for variable periods of time. The jump-like state changes are called *events*. The models specify how events are scheduled and what state transitions they cause. Associated simulators handle the processing of events as dictated by the models.

Ref: Hild, 2000

System Specification

System Specification	DESS	DTSS	DEVS
Time Base	continuous (reals)	discrete (integers)	continuous (reals)
Inputs, States, Outputs	real vector space	arbitrary	arbitrary
Input Segments	piecewise continuous	sequences	discrete events
State & Output Segments	continuous	sequences	arbitrary; piecewise constant

Ref: Prof. B. Zeigler (ACIMS)

Use precise terminology in modeling

Discrete event modeling

Regression analysis

Cost-benefit analysis

Simulation models

Decision-event trees

GIS-based models

Agent-based modeling

Physical systems modeling

Analytical models

Computer Models

System Dynamics

Spreadsheet modeling

Statistical models

Transport models

PowerSim models

Slide courtesy Els van Daalen, TU Delft

Use precise terminology in modeling

Discrete event modeling

Regression analysis

Cost-benefit analysis

Simulation models

Decision-event trees

GIS-based models

Agent-based modeling

Physical systems modeling

Analytical models

Computer Models

System Dynamics

Spreadsheet modeling

Statistical models

Transport models

PowerSim models

model families

Slide courtesy Els van Daalen, TU Delft

Use precise terminology in modeling

Discrete event modeling

Regression analysis

Cost-benefit analysis

Simulation models

Decision-event trees

GIS-based models

Agent-based modeling

Physical systems modeling

Analytical models

Computer Models

System Dynamics

Spreadsheet modeling

Statistical models

Transport models

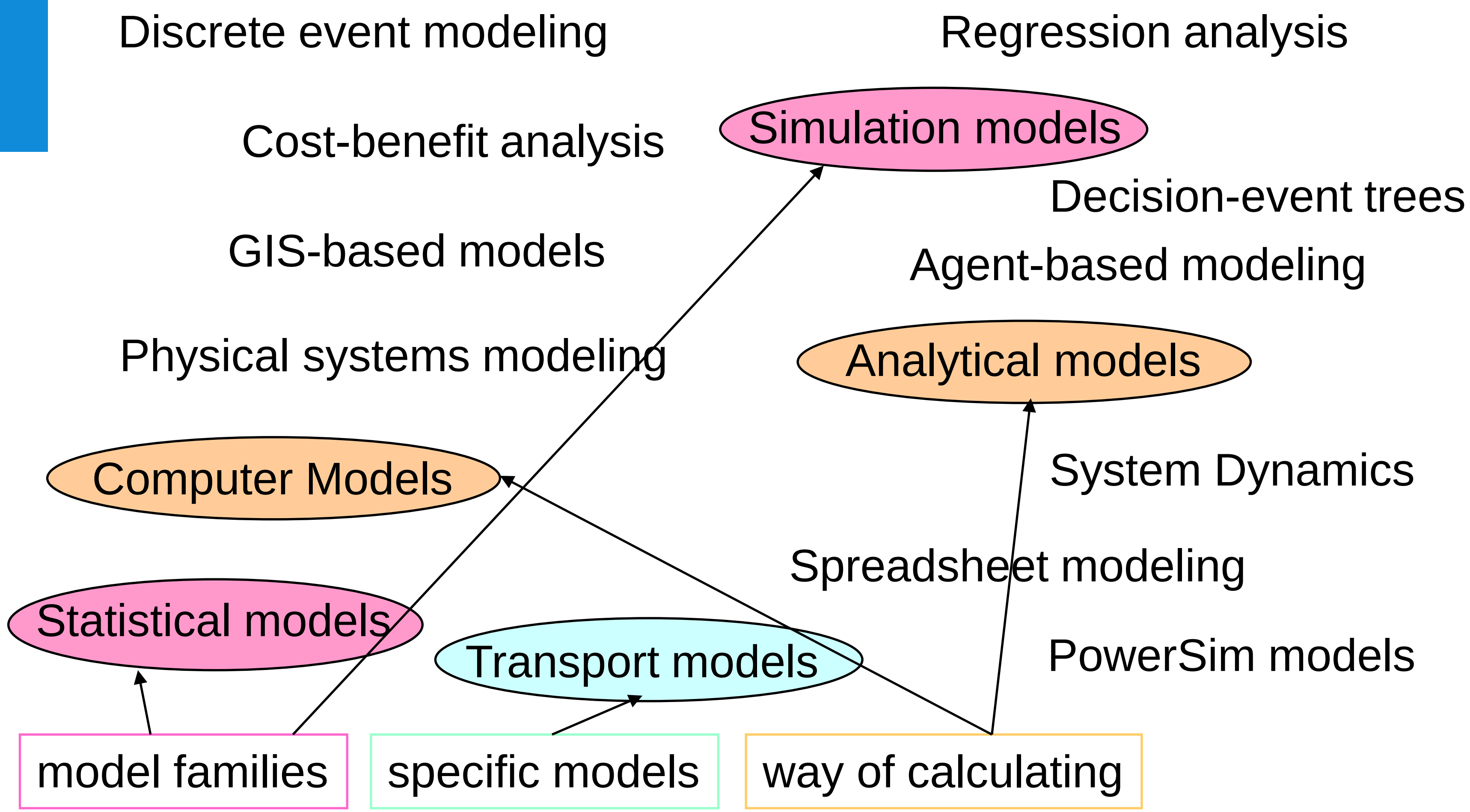
PowerSim models

model families

specific models

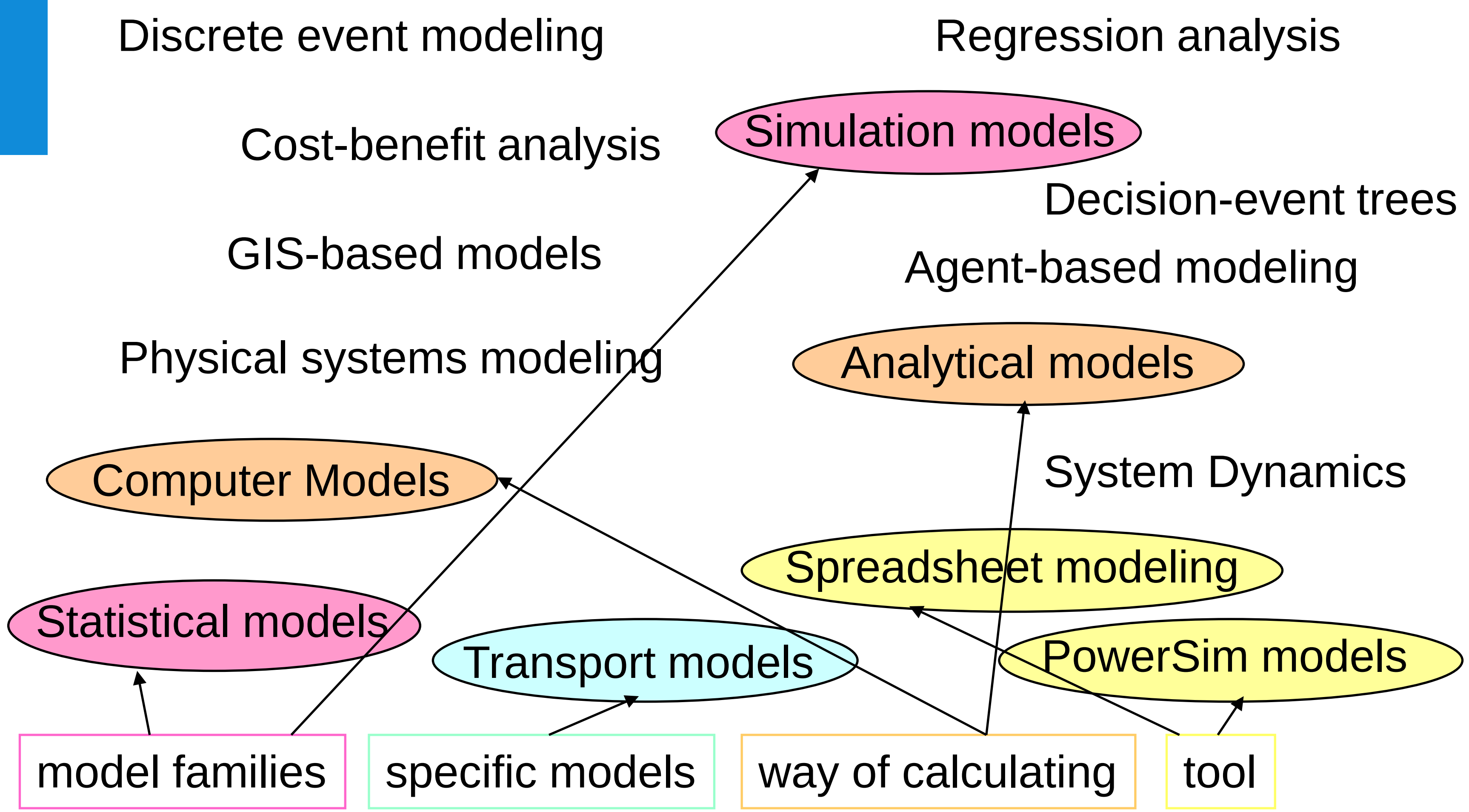
Slide courtesy Els van Daalen, TU Delft

Use precise terminology in modeling



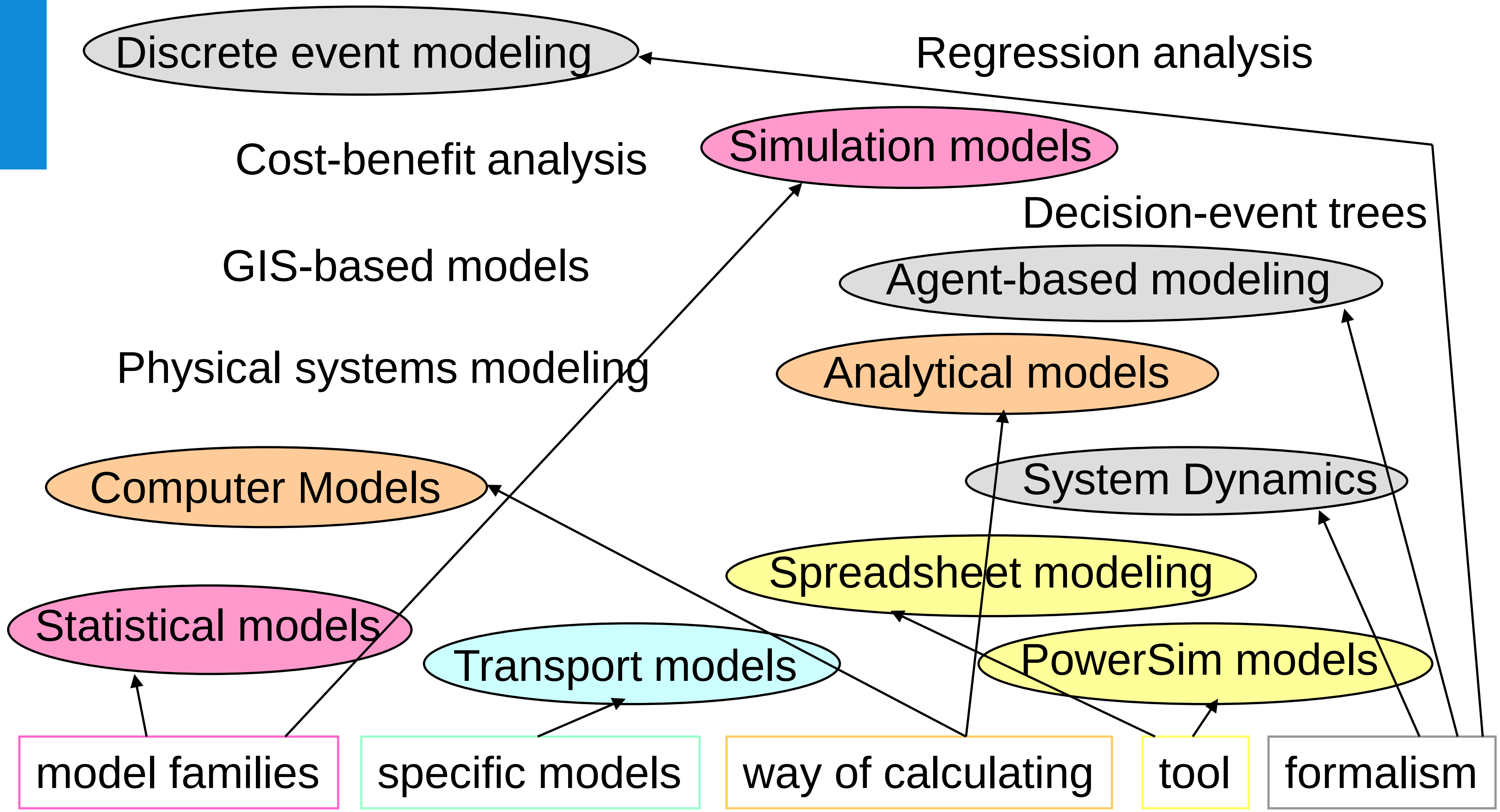
Slide courtesy Els van Daalen, TU Delft

Use precise terminology in modeling



Slide courtesy Els van Daalen, TU Delft

Use precise terminology in modeling



Slide courtesy Els van Daalen, TU Delft

Topic 2: Time and State Relationships

R.E. Nance: The time and state relationships in simulation modeling (1981).

Read the paper!

Terminology problem under consideration

- Every worldview uses its own terminology. What is an object, a process or an event?
- Confusing definitions of concepts have one advantage: from a marketing perspective the customer will never be able to use another language or tool!

What are the consequences?

- There is thus an isolation of users, and fewer and fewer people use more than one language.
- Languages start mixing structure, semantics and modeling requirements.

Objectives of Nance’s paper

1. To illustrate the imprecisions and subtle representational differences present in the different worldviews.

2. To demonstrate the counterproductive influences of the imprecisions and differences.

3. To present a set of definitions that clearly separate yet coordinate time-state relationships.

4. To consider the implications of this approach to simulation model development.

Confusing definitions of event

- An event is the instantaneous change in the value of one or more state variables.
- An event is a particular time when something happens or should have happened.
- An event is the change of the state of the system that indicates the start of an activity.

We'll get to another definition in a moment!

Time-state relationships

- In order to overcome the lock in for a language and inability to understand the concepts of a language, Nance proposes a small set of well defined concepts relating **time and state**.
- In this course we use all these concepts quite strict. We consider them as an essential body of knowledge of a systems engineer.

Objects and attributes

- A **model** of a system is comprised of objects and relationships among the objects.
- An **attribute** is an object descriptor. Attributes are either indicative or relational.
- An **object** is anything that can be characterized by one or more attributes to which values are assigned.
- Behaviour of objects is specified in its **methods**. Methods are either publicly or privately accessible.

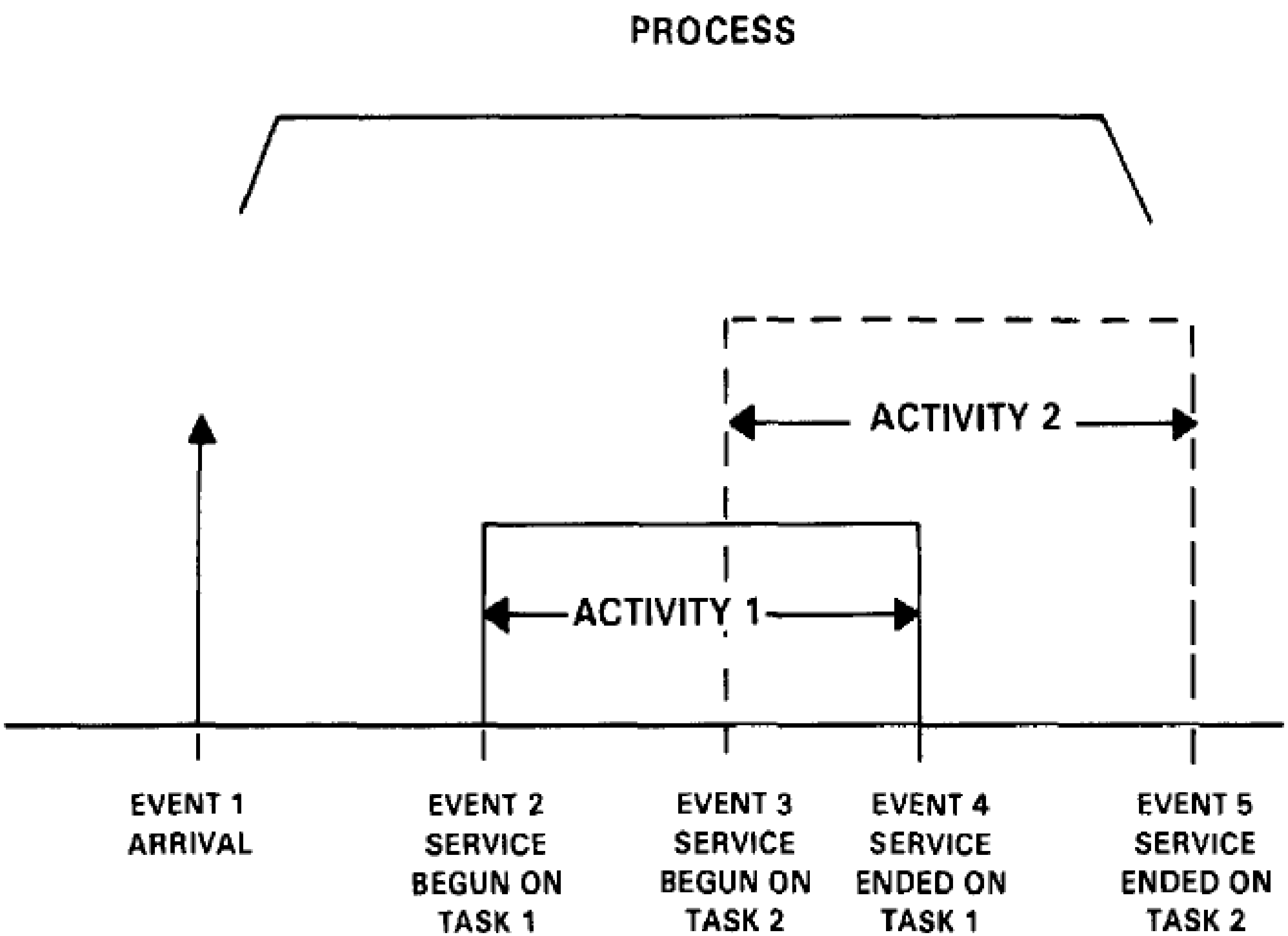
Time and state definitions I

- An **instant** is a value of system time at which the value of at least one attribute changes.
- An **interval** is the duration between two successive instants.
- A **span** is the contiguous succession of one or more intervals.
- The **state** of an object is the enumeration of all attribute values of that particular objects at a particular instant.

Time and state definitions II

- An **activity** is the state of an object over an interval.
- An **event** is a change in object state, occurring at an instant.
- A **process** is the succession of states of an object over a span.

Time and state definitions III



Homework

- **Read** the paper by Nance about time and state relationships in simulation modeling

- Prepare **questions** about the paper (if you have any) for the next class (on Friday)

- **Form groups** of 4 students, **register** your group in Brightspace, and **email** me your group number and your **top-3** of the **paper topics** and **top-3** of the **simulation languages** your group wants to study before Thursday 17:00

- On Thursday evening I will allocate the topics to the groups, before Friday's class; you will also find it on the course website