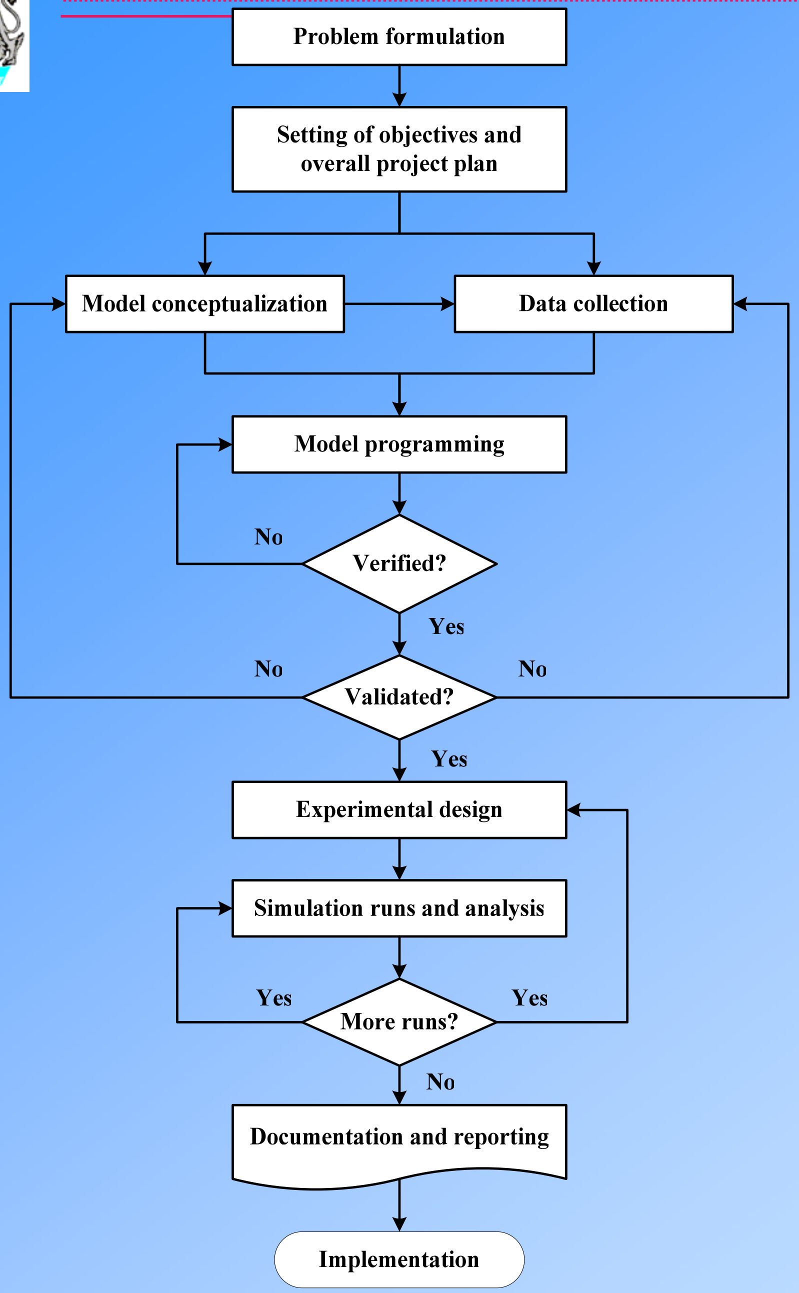
Discrete-Event Simulation – How it is performed

Structure of a simulation study



Development of a conceptual model

Development of a simulation model

Experimenting with a simulation model

Development of a Conceptual Model

Simulation approach:

1. Describe how the modelled system operates
   (*specify its operation algorithm – develop a conceptual model*)
2. Develop a simulation model: computer programme that realises this algorithm
   (*convert the conceptual model into a computer program*)
3. Experiment with the simulation model (*simulate*) and analyse simulation results

Development of a Simulation Model:

Real-World System

↓

Conceptual Model

↓

Simulation Model

Conceptual Model:

A non-software specific description of the simulation model that is to be developed, describing the objectives, inputs, outputs, content, assumptions and simplifications of the model.

Actually this is a functional specification of the computer software.

“The construction of a model of a system is probably as much art as science” *(Jerry Banks et. al.)*

Objectives: The purpose of the model and modelling project

Inputs:

- *Input data:* those elements of the model that reflect influence of its environment (either variables or constants)
- *Experimental factors:* those elements of the model that can be altered to provide a better understanding of, or improvement in, the real world (either quantitative (e.g., number of servers) or qualitative (e.g., queue discipline))

Outputs: Report the results from simulation runs

Content: The components that are represented in the model and their interconnections. It should be described in two dimensions:

- *the scope of the model:* the model boundary or the breadth of the real system that is to be included in the model
- *the level of detail:* the detail to be included for each component in the model's scope

Assumptions made either when there are uncertainties or beliefs about the real world being modelled

Simplifications incorporated in the model to enable more rapid model development and use

Four Requirements of a Conceptual Model

<u>Validity:</u> A perception, on behalf of the modeller, that the conceptual model will lead to a computer model that is satisfactory accurate for the purpose at hand

<u>Credibility:</u> A perception, on behalf of the clients, that the conceptual model will lead to a computer model that is satisfactory accurate for the purpose at hand

<u>Utility:</u> A perception, on behalf of the modeller and the clients, that the conceptual model will lead to a computer model that is useful as an aid to decision-making within the specified context

<u>Feasibility:</u> A perception, on behalf of the modeller and the clients, that the conceptual model can be developed into a computer model

Keep the Model Simple:

KISS = "Keep it Small and Simple"

Model simplicity:

- *constructive simplicity* – attribute of the model
- *transparency* – attribute of the client
  (how well he/she understands the model)

Communicating the Conceptual Model

The output of conceptual modelling should be described in a ***project specification***

Project specification:

Background to the problem situation
Objectives of the simulation study
Expected *benefits*
The conceptual model
Experimentation scenarios to be considered
Data requirements: data required, when required, responsibility for collection
Time-scale and milestones
Estimated *cost*

The iterative nature of simulation studies:

The project specification can change during a simulation study because of:

- Omissions of the original specification
- Changes of the real world
- An increased understanding of simulation on behalf of the clients
- The identification of new problems through the development and use of the simulation model, e.g., specification of simulation goals and ways for reaching them

To handle these changes, a **Specification Change Form** should be maintained

Four Examples of Representing the Content of the Conceptual Model:

1. Component list

A list of model components with some description of the detail included for each one

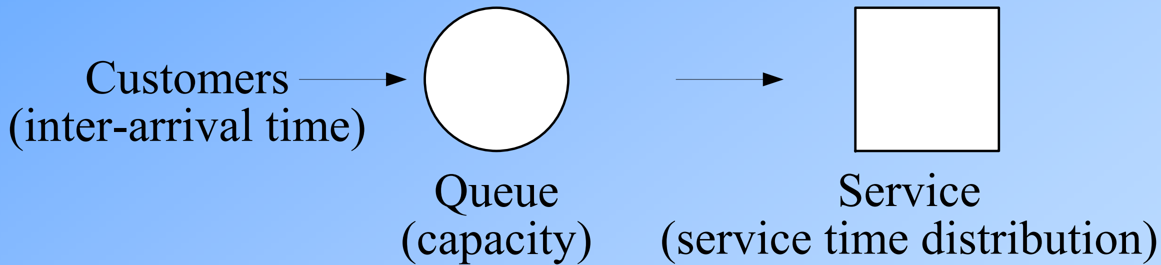
Example: Component list for a single server queuing system

Component	Detail
Customers	Time between arrivals (distribution)
Queue	Capacity
Service desk	Service time (distribution)

2. Process flow diagram (process map)

A process flow or process map, showing each component of the system in a sequence and including some description of the model detail

Example: Process flow diagram for a single server queuing system

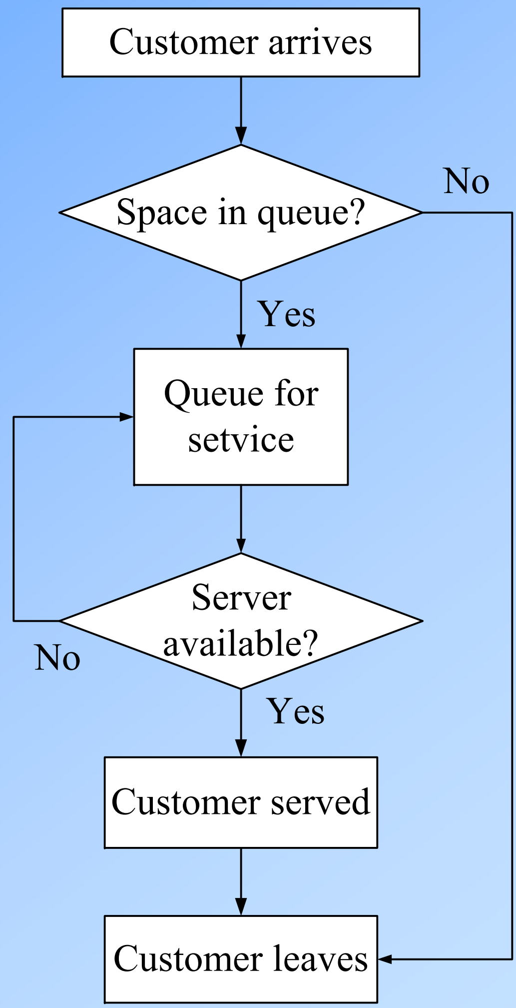


Sample software: ARIS

3. Logic flow diagram

Use standard flow diagram symbols to represent the model logic

Example:
Logic flow diagram for a single server queuing system



Sample software: Visio, Flowcharter

4. Activity cycle diagram

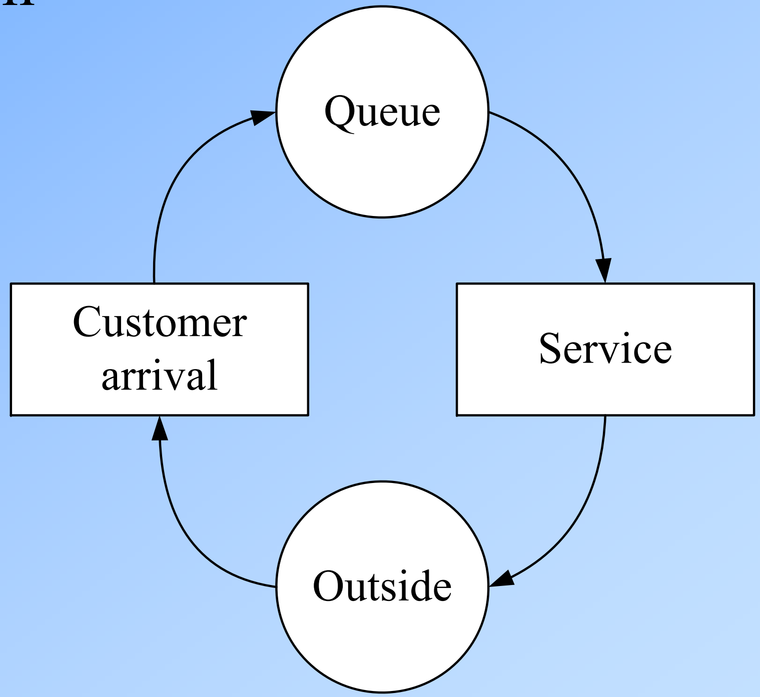
Is used for discrete-event simulation models

Circles - Dead states, when an item waits for something to happen;

Rectangles - Active states, when an item is acted upon;

In general, active and dead states alternate

Example: Activity cycle diagram for a single server queuing system

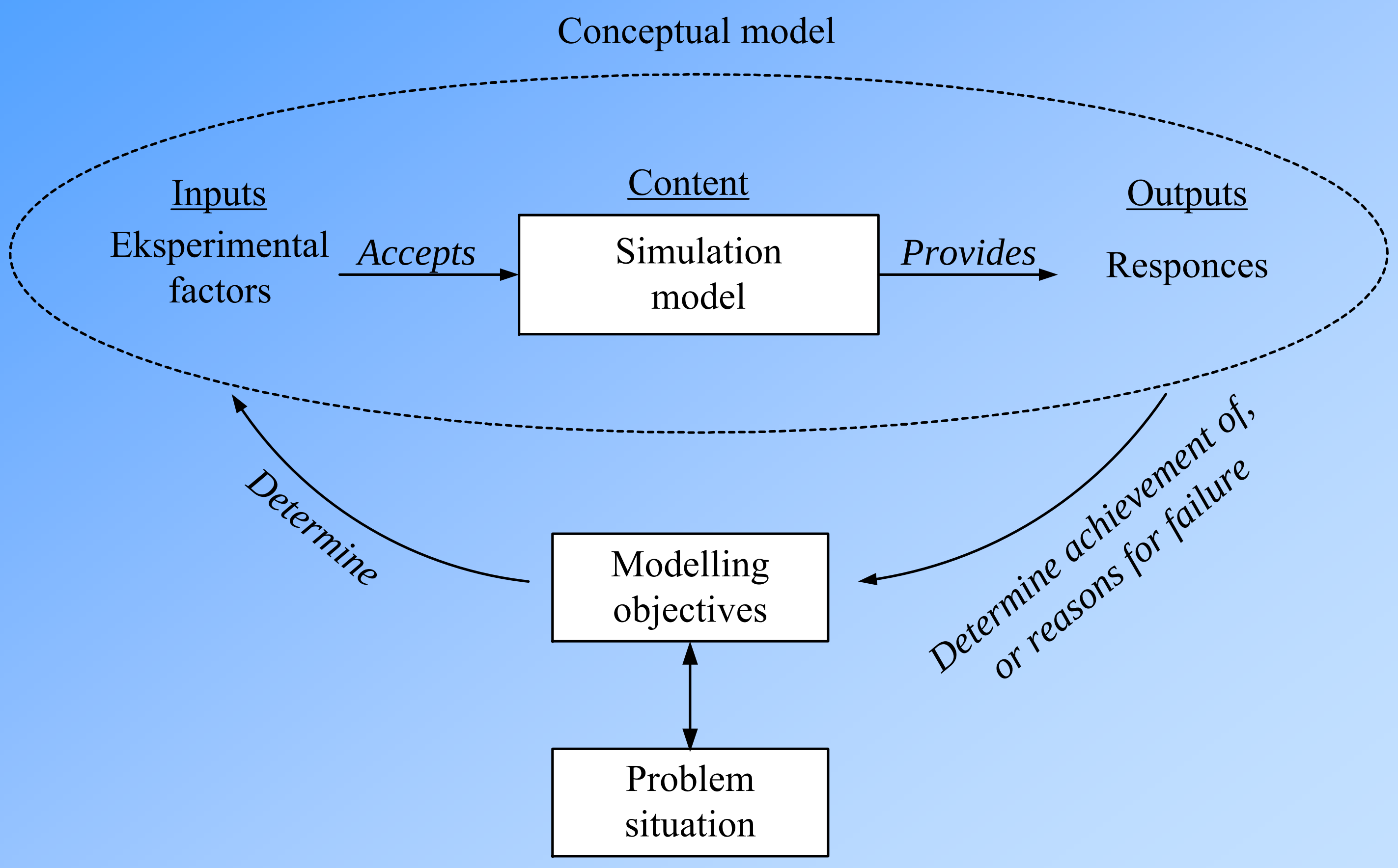


Some other approaches to representing the content of the conceptual model:

- Petri nets
- Event graphs
- Condition specification
- UML (the unified modelling language)
- Various diagramming techniques

A Framework for Conceptual Modelling

An outline of a framework for conceptual modelling



General recommendations:

- <u>Educate</u> the clients:
  - Explain how simulation works
  - Explain what simulation can and cannot
  - Discuss how simulation could help in the clients' particular situation
- Check <u>project constraints</u>:
- <u>Time-scale</u>
- Requirements for <u>visualization</u>
  (e.g., no visualization, or simple schematic visualization, or 2D visualization, or 3D visualization)
- Type of <u>model use</u>: Do the clients wish to use the model themselves? If so:
  - What data input and result viewing facilities do they require?
  - What level of interactive capability is necessary for experimentation?

Developing an understanding of the problem situation:

Problem Formulation

Useful advises for developing a good understanding of the problem situation:

- Be sure that the clients understand, and are able to explain, the problem situation
- Be active: provide the right prompts and speak with right people,
stimulate discussions
- Make assumptions about areas with a limited knowledge
- Make simplifications where possible
- Provide mutual understanding with the clients by preparing and
discussing descriptions of the problem situation

<u>Sample Problem Situation:</u> ***Fast-Food Restaurant***

A fast-food restaurant is experiencing problems with one of the branches in its network. Customers regularly complain about the length of time they have to queue at the service counters. It is apparent that this is not the result of shortages in food, but a shortage of service personnel

Determining the model objectives:

Setting of Objectives

The modelling objectives are central to the modelling process:

- means to determine the model nature
- reference point for model validation
- guide for experimentation
- metrics to judge the success of the project
- key to planning the simulation project

A question to be answered:

"By the end of this study, what do we hope to achieve? "

(e.g., increasing throughput by changes in production scheduling)

Determining the model objectives:

Setting of Objectives

Three aspects to be considered:

- What is that the clients wish to achieve?

(e.g., increasing throughput, reducing costs, improving customer service)

- What level of performance is required?

(e.g., increase throughput by 5%, or minimize costs)

- What constraints must the modeller work within?

(e.g., increasing throughput by changes in production scheduling)

Sample Modelling Objectives: ***Fast-Food Restaurant***

To determine the number of service staff required during each period of the day to ensure that 95% of customers queue for less than 3 minutes for service. Due to space constraints, a maximum of 6 service staff can be employed at any one time

Setting an Overall Project Plan:

Planning the simulation project in terms of:

- people involved
- hardware and software
- time-scheduling
- costing

Deciding on the <u>inputs and outputs:</u>

<u>Inputs:</u>

- *Input data*:
  - as defined by the model content
  - as necessary for model validation

- *Experimental factors*:
  - moving from the modelling objectives
  - discussing with the clients

- *Data entry* for experimental factors:
  - directly into the model code
  - through a set of menus
  - through a data file
  - via a spreadsheet (or other third party software)

<u>Outputs</u> (responses):

- As defined by modelling objectives:
  - measurement of achievements (e.g., throughput)
  - analysing system operation (e.g., why target behaviour was not achieved: utilization of resources and WIP levels, etc.)
- Deciding on a way of reporting:
  - as numerical data *(e.g., mean, max, min, standard deviation)*
  - as graphical data *(e.g., time-series, histograms, Gantt charts, pie charts)*

Sample Experimental Factors and Responses: ***Fast-Food Restaurant***

- *Experimental factors:*

  - Staff rosters (total number of staff at each hour of the day)

  - *Responses (to determine achievement of objectives):*

  - Percentage of customers queuing for less than 3 minutes

- *Responses (to identify reasons for failure to meet objectives):*

  - Waiting time for each customer in the queue: histogram, mean, standard deviation, min and max

  - Time-series of a mean queue size by hour

  - Staff utilization (cumulative percentage)

Designing the model <u>content</u>:

A. Identify the model <u>scope:</u>

- Decide on model *components*. - The experimental factors and outputs provide the basis of what the model needs to include
- Decide on key *interconnections* between the model components

B. Identify the necessary <u>level of detail:</u>

- The model should provide sufficient *accuracy*

<u>Useful technique</u>: **Prototyping:**
Start with a *simple model* and gradually increase the level of detail

<u>Sample Model Scope:</u> ***Fast-Food Restaurant***

Component	Include/ Exclude	Justification
Customers	I	Flow through the service process
Service staff	I	Experimental factors, required for staff utilization response
Food preparation staff	E	Food shortages are not significant
Cleaning staff	E	Do not interconnect with speed of service
Tables	E	Not related to waiting for food
Kitchen	E	Food shortages are not significant

Sample Model Level of Detail:

Fast-Food

– Part 1

Component	Detail	I/E	Comment
Customers	Inter-arrival times	I	Modelled as a distribution
	Order size	E	Represented in service time
Service staff	Service time	I	Modelled as a distribution, taking account of variability in performance and order size
	Staff rosters	I	Experimental factor
	Absenteeism	E	Could be represented by perturbations to the staff rosters

Sample Model Level of Detail: ***Fast-Food – Part 2***

Component	Detail	I/E	Comment
Service staff	Utilization	I	Required for staff utilization response
Queue	Queuing	I	Required for waiting time and queue size responses
	Capacity	E	Assume no effective limit
	Queue behaviour (jockey, balk, leave)	E	Behaviour not well understood

Methods for Model Simplification:

Simplifications are ways of reducing the model complexity by:

<u>Reducing the scope:</u>

Removing components and interconnections that have little effect on model accuracy

<u>Reducing the level of detail:</u>

Representing more simply components and interconnections while maintaining a satisfactory level of model accuracy

Methods for Model Simplification:

The main purpose of model simplification:

To increase the model *utility* while:

- not significantly affecting its *validity* and *credibility*

Theoretical basis: **Pareto law**

A good simplification:

brings the benefits of faster model development and run-speed (*utility*), while:

- maintaining a sufficient level of model accuracy (*validity*)
- and not compromising model *credibility*

Useful <u>simplification methods:</u>

A. To reduce <u>*the scope*</u>:

A1. Excluding components and details:

In case if their omission has little effect on the model accuracy

(e.g., if, while modelling machine repairs, data on machine down time are available, there is no need in modelling the repairing process explicitly, i.e. taking into account waiting for a repairman, actual repair times, etc.)

Useful simplification methods:

A. To reduce *the scope*:

A2. Excluding infrequent events:

Usually system operation is analysed under normal working conditions, i.e., without considering rare events

A3. Splitting models:

Dividing a model into several submodels (e.g., splitting a model of a logistics system into submodels for its transportation and warehousing parts), - preferable, if there is no feedback between submodels

Useful simplification methods:

B. To reduce the *level of detail*:

B1. Aggregation of model components:

- *Black-box modelling:* Representing an operation as a time delay (e.g., when servicing clients or processing parts)
- *Grouping entities:* Representing a group of items by a single entity (e.g., when moving produced parts in batches)

B2. Replacing components with random variables:

Used to represent complex components that are dependent on various random factors (e.g., a transportation process that is influenced by various factors: traffic congestions, weather conditions, technical condition of a car, etc.)

Useful **simplification methods:**

B. To reduce the *level of detail*:

B3. Reducing the rule set:

Excluding rules that have a small impact on model accuracy (e.g., when modelling human behaviour in queues: using a simplified set of rules, like “choosing the shortest queue” or “not joining a queue over a certain length”, but not considering human behaviour in details (jockeying, balking, leaving, etc.))

Data Collection

- GIGO: Garbage In – Garbage Out
- Preparing input data:

  - Observation of actual processes

    But: Be aware of a Hawthorne effect

  - Historical records
  - Expert data
- Supporting tool in the Arena simulation system:

  Input Analyzer

Data Collection

- Statistical analysis:

  - to check if the IDD (Independently and Identically Distributed) assumption is true

  - If not, - correlation analysis and control:

    - autocorrelation
    - cross-correlation
    - providing correlation in data models (e.g., on the basis of regression models)

Data Collection

- Deciding about probability distributions:

  - evaluation of theoretical distributions

  - development of empirical distributions

- Evaluation of theoretical distributions:

  - deciding about a shape (e.g., normal, exponential, etc.)

  - estimation of parameters

  - quality control; popular criteria:

    - Pearson

    - Kolmogorov-Smirnov

Data Collection

<u>Some popular distributions:</u>

• Continuous:

- uniform, triangular
- exponential
- normal
- lognormal, Erlang, Weibull, gamma, beta

• Discrete:

- Poisson

Model Programming

Developing a simulation model:

Converting the conceptual model into a computer program

A survey of simulation programming tools –
in the 2nd part of the course

Specific feature of simulation models – Visualization

Program Verification

<u>Program debugging:</u>

Checking if the developed programme indeed realizes the conceptual model:

"Does the developed model operate as we think the original system does ?"

Supporting tool in the Arena simulation system:

Run Controller

Program Verification

<u>Guidelines:</u>

- Incorporate outside doubters
- Conduct a walkthrough: review the program's logic (as suggested by software testing procedures, e.g., simulate manually a small number of entities)
- Consider extreme situations:
  - deterministic version, when random factors are substituted by particular values (e.g., mean values)
  - intensive/slow incoming flows and slow/quick servicing
  - increased rate of infrequent events (e.g., machine breakdowns)

Model Validation

<u>Checking model adequacy:</u>

Operation of the model is compared with that of the modelled system:

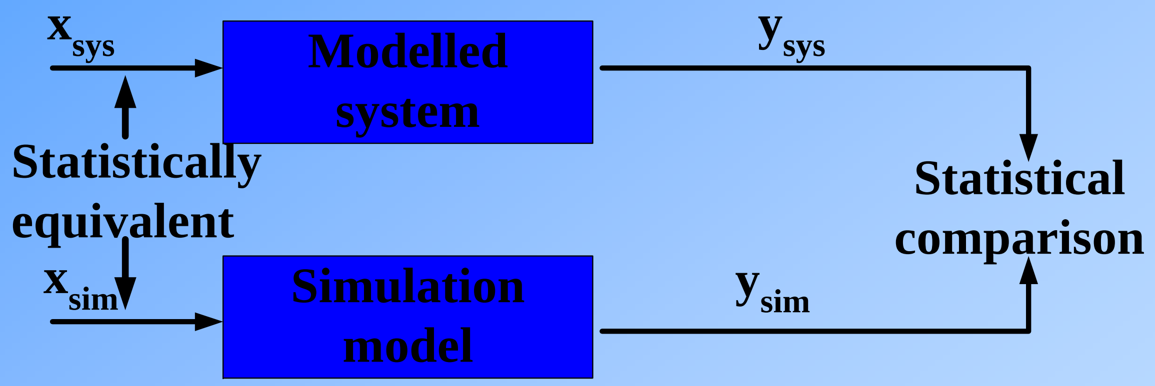
> "Does the developed model operate as the original (modelled) system does ?"

Usually incorporates ***model calibration***: iterative comparing the model to actual system behaviour and using the discrepancies between the two, and the insights gained, to improve the model.

Model Validation

Guidelines:

- Statistical comparison of model and real system behaviour:



- Expert evaluation of model behaviour
- Visualization

Model Validation

Statistical comparison of model and real system behaviour:

Following a statistical approach to comparing alternative system designs:

- Simulation results:

$$y_{sim}^{i},\ i = \overline{1,\ n}$$

- Results of experimenting with the real system:

$$y_{sys}^{i},\ i = \overline{1,\ n}$$

Model Validation

Statistical comparison of model and real system behaviour:

- Development of a confidential interval for the following value:

$$\xi = E\left[Y_{sim}\right] - E\left[Y_{sys}\right]$$

- Checking: Does it incorporate zero ?

Experimental Design

Planning simulation experiments:

- Tactical planning
  - Statistical planning of simulation experiments, e.g.:
    - How many runs to perform ?
    - Do we need to eliminate correlation between observations ?
    - If a transient period should be extracted ?
- Strategic planning
  - In the same way as planning experiments with real systems, e.g., performing optimisation or sensitivity analysis:
    - Factorial design
    - ANOVA

Analysis of Simulation Results

Two types of simulated systems:

- Terminating systems
  - More easy analysis, as observations are usually uncorrelated
- Non-terminating systems
  - Analysis of steady-state operation (vs transient one)
  - More complicated analysis, as observations are usually correlated

Supporting tool in the Arena simulation system:

Output Analyzer

Analysis of Simulation Results

<u>Terminating systems:</u>

- Initial conditions are known, e.g., queues are empty and servers are idle
- Terminating conditions are known, e.g.:
  - Limited operation time
  - Performing a certain operation plan
  - Out of order
- Individual observations $x_1, x_2, ..., x_n$:
  - Represent the whole simulated period, e.g., a number of served clients, an average time in queue, or a number of assembled cars <u>during a working day</u>
  - Statistically independent (and often – normally distributed)

Analysis of Simulation Results

Terminating systems:

- Usually evaluate a mathematical expectation of the observed random variable ***X***:

$$\mu_x = E[x]$$

- Point estimation:

$$\hat{\mu}_x = \bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

Analysis of Simulation Results

Terminating systems:

- Interval estimation:

$$\mu_x \in \bar{x} \pm \frac{S \cdot t_{n-1}^{\alpha/2}}{\sqrt{n}},$$

- where observations x_i are statistically independent and normally distributed;

- ***S*²** - estimation of a variance of observations:

$$S^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \bar{x})^2,$$

Analysis of Simulation Results

Terminating systems:

- $\alpha = 1 - P$ – confidence level, equal to a probability to develop a wrong interval that does not include the mathematical expectation;

- P – confidence probability;

- $t_{n-1}^{\alpha/2}$ - 100(1-α) percentage point of a Student *t*-distribution with *n-1* degrees of freedom

Analysis of Simulation Results

<u>Terminating systems - Example:</u>

x_i – number of clients served during day ***i*** in a simulated shop:

i	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
x_i	**93**	**113**	**107**	**103**	**112**	**103**	**112**	**100**	**98**	**105**

Then a confidence interval of [99.89, 109.31] was developed:

$$\bar{x} = 104{,}60, \quad S = 6{,}59, \quad t_9^{0{,}05/2} = 2{,}26, \quad P = 0{,}95,$$

$$\mu_x \in 104{,}60 \pm 4{,}71 = [99.89,\ 109.31]$$

Analysis of Simulation Results

Terminating systems – Comparison of alternative system designs:

Two alternative system designs, resulted in simulated observations of random variables $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$:

$$X_1 : x_i^1,\ i = \overline{1, n_1} \text{ and } X_2 : x_j^2,\ j = \overline{1, n_2}\,.$$

Their comparison – by developing a confidence interval for a difference of mathematical expectations for $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$:

$$\xi = \mu_1 - \mu_2\,,$$

where $\mu_1 = E[x_1],\quad \mu_2 = E[x_2]$

Analysis of Simulation Results

Terminating systems – Comparison of alternative system designs:

In case if the developed interval incorporates zero, these alternatives could not be distinguished; otherwise a corresponding design could be considered as preferable.

If $\boldsymbol{n_1 = n_2 = n}$, the confidence interval could be developed using the previous approach for a single system, with

$$d_i = x_i^1 - x_i^2$$

instead of $\boldsymbol{x_i}$

Analysis of Simulation Results

<u>Comparison of alternative system designs - Example:</u>

Simulation of a manufacturing cell, estimating the total time that the processed parts spend in the system, in case of 2 alternative queue processing disciplines:
FIFO *(First-In-First-Out)* versus
SPT *(Shortest Processing Time)*

In this case the alternative FIFO corresponds to $\boldsymbol{X_1}$, and SPT corresponds to $\boldsymbol{X_2}$.

Analysis of Simulation Results

Comparison of alternative system designs - Example:

Simulation results:

i	*1*	*2*	*3*	*4*	*5*
x_i^1	8,87	31,52	14,14	14,11	16,72
x_i^2	9,19	19,18	12,30	13,04	17,79
d_i	-0,32	12,34	1,84	1,07	-1,07
i	*6*	*7*	*8*	*9*	*10*
x_i^1	27,78	34,05	22,96	10,98	11,66
x_i^2	11,49	21,61	10,50	13,27	8,50
d_i	16,29	12,44	12,46	-2,31	3,16

Analysis of Simulation Results

<u>Comparison of alternative system designs - Example:</u>

As a result, the following confidence interval was developed:

$$\bar{d} = 5{,}60,\ S = 6{,}96,\ \alpha = 0{,}05,\ t_{n-1}^{\alpha/2} = 2{,}26,$$

$$h = 4{,}98 \text{ and } \xi \in [0{,}62,\ 10{,}58]$$

As it is located on the right side of zero, the alternative SPT is preferable (with a probability of 0,95) in the considered situation.