Resumo MS

Simulation

- The imitation of some real thing, state of affairs, or process, over time, representing certain key characteristics or behaviors of the selected physical or abstract system.
- Applied to complex systems that are impossible to solve mathematically.

Advantages

- When mathematical analysis is not available, simulation may be the only investigation tool
- When mathematical analysis is available, but is too complext, simulation may be the simpler solution
- Time compression and expansion
- Higher control
- Lower costs
- Decision-making support
- Sensitivity analysis
- Training tool
- Doesn't disturb the real system

Disadvantages/not appropriate if

- Problem can be solved by: common sense, simple calculations, analytical methods, direct experiments
- Simulation costs exceed savings
- Resources and time are not available
- Data is not available
- Verification and Validation are not practical due to limited resources
- System behavior is too complex (essential model is not easy to capture)

Life-cycle of simulation project

1. Problem formulation - statement of the problem
2. Set objectives and project plan - questions to be answered; methods/alternatives
3. Model conceptualization - requires experience; begins simple and adds complexity; captures essence of the system
4. Data collection - time consuming; determine what is to be collected
5. Model translation - write the code
6. Verification - does the code represent the model and run properly
7. Validation and calibration - compare model to actual system; Does model replicate system? How to calibrate the model?
8. Experimental design - determine alternatives to simulate (**calibration** is necessary to guarantee the scenario is accurate)
9. Production and analysis - actual runs + analysis of results; determine performance measures
10. More runs?

Approaches

- **Constructivism** - pegar em algo que não existe e aplicar conceitos em cima (**start from nothing**)
- **Realism** - consider everything that is real
- **Analogy** - take one concept and try to represent something else using that concept, for example, traffic flow as an ant colony
- **Abstraction** - focus on essential parts (abstract all other parts) (modelling)

Metrics and Performance Indicators

- **Metric** - Measurable quantities that precisely capture what we want to measure

  - **Performance metric** - Measure performance/quality
  - E.g., response time, throughput, delay, etc.
- **Indicator** - calculated measures of performance
  - Measure how good (or bad) the system is performing
  - Generally a combination of a few metrics
  - Key Performance Indicators (KPIs)
  - E.g. BMI, estimated road traffic death rate, etc.

Model classification

Dynamic vs. Static

- **Dynamic model** - represents a system as it evolves over time
  - E.g. a conveyor system in a factory
- **Static model** - time plays no role. Represents a system at a particular point in time
  - E.g. Monte-Carlo methods

Deterministic vs. Stochastic

- **Deterministic model** - no probabilistic components
  - E.g. worst-case analysis of a system
- **Stochastic model** - involves random variables/probabilities
  - E.g. most queuing and inventory systems

Continuous vs. Discrete

- **Continuous model** - the state of the system changes continuously
  - E.g. chemical processes
  - **Metaphors: fluid dynamics**
- **Discrete model** - the state of the system changes only at discrete points in time (**episodic**)
  - E.g. number of cars in a parking lot
  - **Metaphors**:
    * Event-oriented - used to represent systems where events change the state of a system (e.g., elevator)
    * Process-oriented - processes in timeline with communication between processes (e.g., multi-processor communication). The difference to the activity-oriented is the communication.
    * Activity-oriented - uses gantt-charts (e.g., scheduling of activities, management of a project)
    * Object-oriented - mix of the first three (e.g., UML)
    * Agent-based

Types

- **Descriptive** - describe how the system **is** given a system as input. Describe behavior of the system (doesn't consider anything else)
- **Normative** (optimization) - describe the system in its optimal state. Normalize the system. We want the system to be optimal
- **Predictive** - predict how the system will evolve. Running the simulation to see how the simulation evolves over time. See how the system will be in the future
- **Prescriptive** - Describes changes/transformations in the system to bring it to a desired/optimal state. For example, define the transformations that are necessary that: given a description of the system what transformations should I do to the system optimal
- **Speculative** (scenarization) - **constructivist** approach to model the system. Take a starting point and see how the system will behave if I add something new. Not just duplicating/replication stuff already in the system; add a completely new concept. Useful for *what-if* analysis

System

- Set of interacting components or entities operating together to achieve common goals or objectives
- **Workload** - amount of input for the system. **Affects the performance** of the system
- **Threshold** - parameter to compare systems. **Doesn't affect performance** (unless changing). Similar to benchmark

```
                  --------------------
                 | Real World System  |
                 |  ----------------  |
                 | | Assumed system | | ----
                 |  ----------------  |     |
                  --------------------      |
                            |               |
                            v               |
                  --------------------      |
                 |  Conceptual model  |     |
                  --------------------      |
                            |               | Validation
                            v               |
                  --------------------      |
              -- |    Logical model   |     |
             |    --------------------      |
             |              |               |
Verification |              v               |
             |    --------------------      |
              -> |  Simulation model  | <---
                  --------------------
```

Conceptual model

- Abstract essential features
- Select correct level of details (assumptions)
- Low levels of detail may result in loosing information and goals cannot be accomplished
- High levels of detail require: more time/effort; longer simulation runs; more likely to contain errors; more data for validation and verification
  - Accuracy of the model increases with level of details until platoon. Afterwards the increase of detail, decreases model accuracy
- Cost of the model increases exponentially with the level of detail

Logical model

- Shows the logical relationships among the elements of the model (flowchart)

Verification and Validation

- **Validation** - model represents accurately the real system. **Can validate descriptive and predictive models**
  - Needs data collection
  - May require help of subjects
- **Verification** - model implements correctly the logical model (**code is correct**)

Components of a system

- **Entity** - an object of interest in the system
  - Usually has multiple instantiation in the system

  - A system can have different types of entities concurrently
  - E.g., patients, visitors
- **Attribute** - characteristic of all entities
  - Each entity has its own specific value for each attribute
  - type of illness, age, gender, temperature, blood pressure, etc.
- **Resources** - what entities compete for
  - Entity **seizes** a resource, **uses** it, **releases** it
  - Can have **units of capacity** that change during the simulation (expending the resource)
  - E.g., doctors, nurses, x-ray equipment, etc.
- **Variables** - a piece of information that reflects some characteristic of the whole system, not of specific entities
  - Entities podem aceder e mudar variables
  - Outras variables mudam como resultado das system dynamics
  - E.g., number of patients in the system, number of idle doctors, current time, etc.
- **State** - a collection of variables that contains all the information necessary to describe the system at any time
  - E.g., {number of patients in the system, status of doctors (busy/idle), number of idle doctors}
- **Event** - an instantaneous occurrence that changes the state of the system
  - E.g., arrival of a new patient, completion of service (i.e., examination), failure of medical equipment, etc.
- **Activity** - represents a time period of specified length
  - E.g., surgery, checking temperature, x-ray, etc.

Parameters of a system

- **Exogeneous** - inputs of the system:
  - **Controllable variable** - manipulate the scenario. Can give exact value.
  - **Uncontrollable variable** - for example, historical data of customers given as input to the system.
- **Endogenous** - output of the system

Policies

- **Operation policies** - set of scenarios/different configurations that dictate how the system operates
- **Implementation policies** - não demos
- **Validation policies** - processes related to code. Guarantee that the system is working correctly
- **Calibration policies** - processes related to code. Guarantee that the system is working correctly
- **Implementing simulation results** - depois de fazer **operation policies**, escolher a que dá melhores resultados, e implementar uma simulação com a **policy** escolhida

Agent-based

Approaches

- **Agent-based** modelling
  - Agents as metaphor for system modelling
  - Simulation methodology for Multi-Agent-Simulation (MAS)
- **Agent-directed** simulation
  - Agents steer and manage the whole simulation process
  - ML and Meta-modelling for intelligent calibration and scenario management
- **Agent-oriented** simulation SW
  - Software architectures for simulation IDEs based on MAS

Motivations

- Systems are getting more complex

- Complex systems are different to model as a whole (aggregate)
- Higher level of tools available
- Human behavior is often neglected in other metaphors
- The environment becomes close to being an agent

Cooperative vs. Competitive MAS

- Cooperation - all agents have the same goal (global utility and performance)
- Collaboration - different goals but can help each other
- Competition - oposing goals (agents interested in their own utility)

Environment

- **Aspatial environment** (soup model) - no space representation
- **Cellular-automata** - cell behavior is affected by neighboring cells. Usado em agent-based simulations
  - Moore - interage com os 8 à volta
  - Von Neumann - interage na vertical e horizontal
  - Rotated Von Neumann - interage nas diagonais
- **Euclidean environment** - space is continuous (2D)
- **Geographic Information System (GIS)** - tipo mapa
- **Network** - relationships as explicit links (e.g., redes sociais)