**VALIDATION AND VERIFICATION TECHNIQUES**

Validation and verification are crucial steps in the development and use of simulation languages and models. They ensure that the simulation accurately represents the real-world system or process it is intended to mimic (validation) and that it correctly implements the intended model (verification). Here's a brief overview of both concepts in the context of simulation languages:

**Verification**

Verification is the process of ensuring that the model was correctly implemented in the simulation language. It involves checking that the code or model behaves as intended and is free of errors. This step is about ensuring the internal correctness of the model within the simulation environment. It typically involves:

* **Code Inspection**: Reviewing the simulation code to ensure it correctly implements the model's equations, logic, and algorithms.
* **Testing**: Conducting tests to check the model's behavior under various conditions and comparing the outcomes with expected results. This can include unit testing for individual components and integration testing for the entire model.

**Validation**

Validation, on the other hand, is the process of ensuring that the simulation accurately reflects the real-world system or process it is meant to **represent**. This is about the external correctness of the model, ensuring that it produces outcomes that are consistent with real-world observations or expected theoretical results. Validation activities include:

* **Data Comparison**: Comparing simulation results with real-world data or outcomes from trusted models to check for accuracy.
* **Expert Review**: Having domain experts review the simulation outcomes and the model's assumptions to ensure they align with the real-world system's behavior.
* **Sensitivity Analysis**: Testing how changes in model inputs affect outputs, which can help identify if the model behaves realistically under a range of conditions.

**Process**

Both verification and validation are iterative processes. They start from the early stages of model development and continue throughout the lifecycle of the simulation project. It's common to go back and refine the model based on findings from verification and validation activities.

**Tools and Techniques**

Various tools and techniques are used for verification and validation, including:

* **Automated Testing Tools**: Software that automates the process of code testing and analysis.
* **Statistical Analysis Software**: Tools that help compare simulation results with real-world data statistically.
* **Model Checking**: A formal verification technique that systematically checks whether a model meets certain criteria.

The importance of validation and verification in simulation cannot be overstated. They are essential for building confidence in the simulation results, which is critical when simulations are used for decision-making, policy formulation, or system design in complex and critical applications.

Validation and verification of simulation models are essential steps to ensure that these models are both accurate representations of the real world (validation) and correctly implemented (verification). Let's delve into the specifics of each process and their significance in the context of simulation modeling.

**Why Are Validation and Verification Important?**

* **Reliability**: They ensure that the model is reliable and can be trusted for making decisions, predictions, or understanding complex systems.
* **Accuracy**: Through validation and verification, one can be confident that the model accurately represents what it is supposed to and that the simulation results are credible.
* **Usefulness**: Validated and verified models are more likely to be useful for their intended purpose, whether that's forecasting, policy analysis, system design, or educational purposes.

**Process and Iteration**

Both validation and verification are not one-time activities but rather ongoing processes that occur throughout the lifecycle of a simulation project. As new data becomes available or as the model is used in new ways, additional validation and verification may be necessary.

**Other Validation and verification Techniques**

This section describes validation techniques and tests commonly used in model verification and validation. Most of the techniques described here are found in the literature, although some may be described slightly differently. They can be used either subjectively or objectively. By “objectively,” we mean using some type of mathematical procedure or statistical test, e.g., hypothesis tests or confidence intervals. A combination of techniques is generally used. These techniques are used for verifying and validating the submodels and the overall model.

* Animation: The model’s operational behaviour is displayed graphically as the model moves through time, e.g., the movements of parts through a factory during a simulation run are shown graphically.
  + Purpose: Helps in visually verifying the operational behaviour of the simulation model over time.
  + Application: Used to identify and correct errors in model logic or behavior by observing the graphical representation of the model's processes, such as the movement of parts through a factory.
* Comparison to Other Models: Various results (e.g., outputs) of the simulation model being validated are compared to results of other (valid) models. For example, (1) simple cases of a simulation model are compared to known results of analytic models, and (2) the simulation model is compared to other simulation models that have been validated.
* Purpose: Validate the simulation model by comparing its outputs to those from other validated models.

Application: Ensures consistency and accuracy by benchmarking against known analytic models or other credible simulation models

* Degenerate Tests: The degeneracy of the model’s behaviour is tested by the appropriate selection of values of the input and internal parameters. For example, does the average number in the queue of a single server continue to increase over time when the arrival rate is larger than the service rate?
  + Purpose: Tests the model's behavior under extreme or degenerate conditions.
  + Application: Used to check for logical consistency and robustness, such as ensuring that queues behave as expected under **high load conditions**
* Event Validity: The “events” of occurrences of the simulation model are compared to those of the real system to determine if they are similar. For example, compare the number of fires in a fire department simulation to the actual number of fires.
  + Purpose: Ensures that the events simulated by the model closely match real-world occurrences.
  + Application: Comparing the frequency and characteristics of simulated events to actual events to verify realism and accuracy
* Extreme Condition Tests: The model structure and outputs should be plausible for any extreme and unlikely combination of levels of factors in the system. For example, if in-process inventories are zero, production output should usually be zero.
  + Purpose: Tests the model's outputs under extreme or unlikely scenarios.
  + Application: Used to validate the model's structural integrity and logical consistency under conditions such as zero inventories leading to zero production.
* Face Validity: Individuals knowledgeable about the system are asked whether the model and/or its behavior are reasonable. For example, is the logic in the conceptual model correct and are the model’s input-output relationships reasonable?
  + Purpose: Gather qualitative assessments of the model's realism and appropriateness from domain experts.
  + Application: Engages experts to review the model's logic, inputs, and outputs for reasonableness and accuracy.
* Historical Data Validation: If historical data exist (e.g., data collected on a system specifically for building and testing a model), part of the data is used to build the model and the remaining data are used to determine (test) whether the model behaves as the system does. (This testing is conducted by driving the simulation model with either samples from distributions or traces (Balci and Sargent 1984b).)
  + Purpose: Uses existing data to validate the model's ability to replicate real-world behavior.
  + Application: Part of the data is used to build the model, and the remainder is used to test the model, ensuring it behaves similarly to the real system.
* Historical Methods: The three historical methods of validation are rationalism, empiricism, and positive economics. Rationalism requires that the assumptions underlying a model be clearly stated and that they are readily accepted. Logic deductions are used from these assumptions to develop the correct (valid) model. Empiricism requires every assumption and outcome to be empirically validated. Positive economics requires only that the model’s outcome(s) be correct and is not concerned with a model’s assumptions or structure (causal relationships or mechanisms).
  + Purpose: Applies philosophical approaches to validation, focusing on assumptions, empirical evidence, and outcomes.
  + Application: Rationalism, empiricism, and positive economics are used to build and validate models based on clear assumptions, empirical validation, and correct outcomes, respectively.
* Internal Validity: Several replications (runs) of a **stochastic model** are made to determine the amount of (internal) stochastic variability in the model. A large amount of variability (lack of consistency) may cause the model’s results to be questionable and if typical of the problem entity, may question the appropriateness of the policy or system being investigated.
  + Purpose: Assesses the amount of stochastic variability in the model to ensure reliability.
  + Application: Conducting multiple runs of the model to examine internal variability and its impact on the model's conclusions.
* Multistage Validation: Naylor and Finger (1967) proposed combining the three historical methods of rationalism, empiricism, and positive economics into a multistage process of validation. This validation method consists of (1) developing the model’s assumptions on theory, observations, and general knowledge, (2) validating the model’s assumptions where possible by empirically testing them, and (3) comparing (testing) the input-output relationships of the model to the real system.
  + Purpose: Combines various historical methods into a comprehensive validation process.
  + Application: Involves developing assumptions, empirically testing them, and comparing the model's input-output relationships with the real system.
* Operational Graphics: Values of various performance measures, e.g., the number in queue and percentage of servers busy, are shown graphically as the model runs through time; i.e., the dynamical behaviors of performance indicators are visually displayed as the simulation model runs through time to ensure they behave correctly.
  + Purpose: Uses graphical displays of performance measures to verify correct model behavior over time.
  + Application: Dynamically visualizes key performance indicators during simulation runs to ensure expected behavior.
* Parameter Variability - Sensitivity Analysis: This technique consists of changing the values of the input and internal parameters of a model to determine the effect upon the model’s behavior or output. The same relationships should occur in the model as in the real system. This technique can be used qualitatively—directions only of outputs—and quantitatively—both directions and (precise) magnitudes of outputs. Those parameters that are sensitive, i.e., cause significant changes in the model’s behavior or output, should be made sufficiently accurate prior to using the model. (This may require iterations in model development.)
  + Purpose: Determines the impact of changes in model parameters on outputs.
  + Application: Identifies sensitive parameters and adjusts them to accurately reflect their real-world counterparts, ensuring the model's robustness and reliability.
* Predictive Validation: The model is used to predict (forecast) the system’s behavior, and then comparisons are made between the system’s behavior and the model’s forecast to determine if they are the same. The system data may come from an operational system or be obtained by conducting experiments on the system, e.g., field tests.
  + Purpose: Uses the model to forecast future system behavior and compares these forecasts to actual outcomes.
  + Application: Validates the model's predictive capability by comparing its forecasts against real system behavior or experimental data.
* Traces: The behaviors of different types of specific entities in the model are traced (followed) through the model to determine if the model’s logic is correct and if the necessary accuracy is obtained.
  + Purpose: Follows specific entities through the model to verify correct logic and accuracy.
  + Application: Used to ensure that the model accurately represents the logic and behavior of different entity types within the simulated environment
* Turing Tests: Individuals who are knowledgeable about the operations of the system being modeled are asked if they can discriminate between system and model outputs. (Schruben (1980) contains statistical tests for Turing tests.)
  + Purpose: Asks experts to differentiate between system outputs and model outputs to assess model realism.
  + Application: Employs statistical tests to determine if experts can tell the difference between the model and real system outputs, indicating the model's accuracy.

Each of these techniques contributes to a thorough process of validation and verification, ensuring that simulation models are both accurate representations of reality and correctly implemented. Combining these techniques provides a robust framework for establishing the credibility of simulation models.

**Validation: Errors**

ν As a general rule, if there are discrepancies between observed and simulated data, the between observed and simulated data, the technical structure of a model should be the last factor to suspect.

1. Model inadequate

2. Lack of calibration

3. Errors in the code Errors in the code

4. Errors in the inputs

5. Errors in the use

6. Errors in the experimental data

**Model Adequacy**

ν Are all the important processes for a given Are all the important processes for a given environment included?

ν Are the processes modelled correctly? Are the processes modelled correctly?

ν Was the range of data used to develop model components for process simulation wide components for process simulation wide enough to include our conditions?

**Errors in the Code**

ν Following steps can be undertaken to Following steps can be undertaken to check a code:

1. Do calculations using for instance a spreadsheet and compare with model results

2. Verify that simulation results are within the known physical and biological reality

3. Run simulations with highly contrasting inputs

**Errors in the inputs**

ν The effect of an error in the inputs used to run a simulation is proportional to the sensitivity that the model has for that input that the model has for that input

. ν A model is an interpretation of a system, i.e. elements interrelated in the real word.

ν If correctly structured, a model contains the submodels to simulate the most important processes in a given environment.

**Errors in the use**

ν All models have limitations in their use given by their structure; using the model in by their structure; using the model in conditions where non-simulated processes are important causes wrong estimates for most of the simulated processes.

**Errors in the Experimental Data**

ν The experimental data used to test the model and the experimental data used to test model predictive capabilities are affected by the experimental error, which can be large.

ν Only a large number of experimental data allows a meaningful evaluation of model allows a meaningful evaluation of model performance in statistical terms.