A Framework for Optimizing Simulation Model Validation & Verification

Bill Roungas
Alexander Verbraeck

Department of Multi Actor Systems
Delft University of Technology
Delft, The Netherlands
Email: v.roungas@tudelft.nl,
a.verbraeck@tudelft.nl

Sebastiaan Meijer

Department of Health Systems Engineering
KTH Royal Institute of Technology
Huddinge, Sweden
Email: sebastiaan.meijer@sth.kth.se

***Abstract*—Thirty years of research on validation and verification have returned a plethora of methods and statistical techniques through methodological and case studies. It is, however, this abundance of methods and techniques that poses a major challenge. Due to time and budget constraints, it is impossible to apply all the available methods and techniques in a single study, and as such a careful selection has to be made. This paper builds on two assumptions: a) simulations, real-world systems, methods, and techniques can be defined on the basis of different characteristics and b) certain methods and techniques are more suitable than others for different kinds of simulation studies. The present study aims at identifying the specific characteristics that make certain methods and techniques more effective and more efficient than others, when juxtaposing these with the simulations' and systems' different characteristics. The conclusion will advance a methodology for choosing the most appropriate methods and techniques for validating and/or verifying a simulation.**

Keywords–simulation; validation; verification; method selection.

I. Introduction

Back in 1972, based on Forrester's work [1][2], Meadows et al. [3][4] introduced World 3, a simulation of the world for the years 1900-2100. The purpose of the simulation model was to project the dynamic behavior of population, capital, food, non-renewable resources, and pollution. The model's forecast was that during the contemplated two centuries the world will experience a major industrial collapse, which will be followed by a significant decrease in human population. The model became very popular especially because of the increasing interest in environmental degradation encountered because of human activities [5]. Even though the model gained support for being "of some use to decision makers" [4] and generated the spark for many later global models, it had several shortcomings, for which it received a lot of criticism [6]. In turn, this criticism raised the question of whether, and to what extent, such simulation models are validated and verified. This is just one example of the notion that validation and verification (V&V) is a fundamental part of a simulation study [7].

The term V&V is used to characterize two relatively different approaches that almost always go hand by hand, namely validation and verification. Validation is this phase of a study that ensures that the simulation imitates the underline system, to a greater or lesser extent, and in any case satisfactorily [8], or in layman terms validation addresses the question of whether the built model is the also the "right" one [9]. On the other hand, verification is the phase of the study that ensures that the model and its implementation are correct [10], or in layman terms verification addresses the question of whether the model was built in the "right" way [9]. V&V has become a well-researched field with a significant amount of produced literature and commercial case studies. The large number of V&V methods and statistical techniques created or adopted by this wide range of research, is the greatest impediment to designing a V&V study.

The predetermined budget of a simulation study usually limits the amount of time and resources that can be spent on V&V. Additionally, the nature and the diverse characteristics of simulations limit the number of V&V methods and statistical techniques that are applicable to each simulation. In other words, not all V&V methods (hereinafter referred to as methods) and V&V-applicable statistical techniques (hereinafter referred to as techniques) are suitable for every simulation. To the best of our knowledge, a taxonomy for characterizing methods and techniques and, subsequently, matching them with different simulations does not exist. Therefore, the research question that this study will address is:

> How can the selection of V&V methods and V&V-applicable statistical techniques given the simulation and the real-world system at hand be optimized as to be more time efficient and rigorous?

This paper aims at identifying the majority of the available methods and techniques in order to classify them on the basis of their different characteristics and on whether they can be used to validate or verify a simulation, and eventually match them with characteristics of simulation models.

In Section II, a literature analysis on methods and techniques, simulation properties, and simulation study phases is conducted. In Section III, a methodology towards developing a framework for simulation V&V method and statistical techniques selection is proposed. In Section IV, a case study is presented to illustrate how the proposed framework can actually be implemented. Finally, in Section V, the future potential extensions of the framework are presented and final remarks are made.

II. Literature Analysis

In this section, a 3-step literature analysis is presented. The initial hypothesis of this study is that simulations exhibit certain properties that influence the effectiveness and applicability of methods and techniques. Therefore, the 3 steps of the literature analysis are the following:

Step 1: Identification of methods and techniques.

Step 2: Identification of simulations' properties potentially influencing the selection of methods and identification of simulations' and systems' characteristics potentially influencing the selection of techniques.

Step 3: Identification of the phases of a simulation study.

A. Step 1: V&V methods and statistical techniques

Methods are different in many aspects; some methods are strictly mathematical whereas others accommodate the more qualitative aspects of simulations, etc. Balci [11] identified more than 70 methods, which in turn categorized into four categories: informal, static, dynamic, and formal. Balci's [11] list is the most accurate representation of the body of work on methods and, even to date, is considered as the most extensive one. This paper adopts the list in reference - but not the categorization - and goes as far as to propose a new classification of methods.

On the other hand, numerous techniques have been proposed throughout the years, a subset of which are applicable in V&V studies. Moreover, techniques can be characterized in various ways, e.g., depending on the input they require (numerical, categorical etc.), or the purpose they are used for (goodness-of-fit, time series etc.).

In Section II-A1 and Section II-A2, the identified methods and techniques are listed, respectively, along with a brief definition for each one of them.

1) V&V Methods:

Acceptance Testing: Acceptance Testing is testing the model using the actual hardware and data to determine whether all the specified requirements are satisfied [12].

Alpha Testing: Alpha Testing is the operational testing of the alpha version of the model in a department within the company, yet not the one involved with the model development [13].

Assertion Checking: Assertion Checking checks what is happening as opposed to what the modeler assumes is happening thus detecting potential errors in the model [11].

Audit: An Audit is undertaken to assess how adequately the simulation study is conducted with respect to established plans, policies, procedures, standards and guidelines. The audit also seeks to establish traceability within the simulation study [11].

Beta Testing: Beta Testing is the operational testing of the beta version of the model under realistic field conditions [14].

Bottom-Up Testing: Bottom-Up Testing is testing each submodel, when the model is developed with a bottom-up development strategy, and once every submodel belonging to the same parent is finished and tested, then these submodels are integrated and tested again [11].

Cause-Effect Graphing: Cause-Effect Graphing aids in selecting, in a systematic way, a high-yield set of test cases and it is effective in pointing out incompleteness and ambiguities in the specification [15].

Comparison Testing: Comparison Testing is testing the different versions of the same simulation model [16].

Compliance Testing: Compliance Testing tests how accurately different levels of access authorization are provided, how closely and accurately dictated performance requirements are satisfied, how well the security requirements are met, and how properly the standards are followed [17]. It consists of the following techniques:

1) Authorization Testing, which tests how accurately and properly different levels of access authorization are implemented in the model and how properly they comply with the rules and regulations [12].
2) Performance Testing, which tests whether (a) all performance characteristics are measured and evaluated with sufficient accuracy, and (b) all established performance requirements are satisfied [12].
3) Security Testing, which tests whether all security procedures are correctly and properly implemented in conducting a simulation study [12].
4) Standards Testing, which substantiates that the simulation model is developed with respect to the required standards, procedures, and guidelines [11].

Control Analysis: Control Analysis analyzes the control characteristics of the model. It consists of the following techniques:

1) Calling Structure Analysis, which is used to assess model accuracy by identifying who calls whom and who is called by whom [14].
2) Concurrent Process Analysis, in which model accuracy is assessed by analyzing the overlap or concurrency of model components executed in parallel or as distributed [18].
3) Control Flow Analysis, which requires the development of a graph of the model where conditional branches and model junctions are represented by nodes and the model segments between such nodes are represented by links [13].
4) State Transition Analysis, which requires the identification of a finite number of states the model execution goes through and shows how the model transitions from one state to another [19].

Data Analysis: Data Analysis ensures that (1) proper operations are applied to data objects (e.g., data structures, event lists, linked lists), (2) the data used by the model are properly defined, and (3) the defined data are properly used [12]. It consists of the following techniques:

1) Data Dependency Analysis, which involves the determination of what variables depend on what other variables [20].
2) Data Flow Analysis, which is used to assess model accuracy with respect to the use of model variables [21].

Debugging: Debugging identifies errors causing the model to fail and changes the model accordingly in order to correct these errors [20].

Desk Checking: Desk Checking is when a person other than the modeler thoroughly examines the model to ensure correctness, completeness, consistency and unambiguity [13].

Documentation Checking: Documentation Checking ensures accuracy and up-to-date description of the model logic and its results [11].

Execution Testing: Execution Testing collects and analyzes execution behavior data in order to reveal model representation errors. It consists of the following techniques:

1) Execution Monitoring, which examines low-level information about activities and events taking place during model execution [11].
2) Execution Profiling, which examines high-level information about activities and events taking place during model execution [11].
3) Execution Tracing, which tracks line-by-line the execution of a model [11].

Face Validation: In Face Validation, people knowledgeable about the system under study subjectively compare model and system behaviors and judge whether the model and its results are reasonable [22].

Fault/Failure Analysis: Fault/Failure Analysis determines if any faults or failures can logically occur and in what context and under what conditions [14].

Fault/Failure Insertion Testing: Fault/Failure Insertion testing inserts an fault or failure into the model and observes whether the model will behave in the expected invalid manner [11].

Field Testing: Field Testing executes the model in an operational situation for the purpose of collecting information regarding the model validation [23].

Functional (Black-Box) Testing: Functional Testing ignores the internal mechanism(s) of the model and focuses on the generated outputs based on specific input and execution conditions [24].

Graphical Comparisons: In Graphical Comparison, graphs produced from the model are compared to graphs produced by the real-world system under study, in order to detect similarities and differences between the two [14].

Induction: Induction asserts that if every step a model follows is valid and the model terminates, then the model is valid [11]. Induction as a term can be found in many fields, like mathematics in which case it is a tool for directly proving theorems. In simulation model validation, where absolute validity does not exist [25], induction should more correctly be referred to as inductive reasoning, which is based on one or more inductive arguments, and the conclusions are not considered as the absolute truth but rather a strong evidence [26].

Inference: Inference is similar to Induction; it is a mental process by which one proposition is arrived at and affirmed on the basis on one or more other propositions assumed as the starting point of the process [26].

Inspections: Inspection is a five phase procedure conducted by four to six people. The phases include not only a validation phase but also suggestions for improvements and a follow-up [27].

Interface Analysis: Interface Analysis consists of the following techniques:

1) Model Interface Analysis, which is conducted to examine the (sub)model-to-(sub)model interface and determine if the interface structure and behavior are sufficiently accurate [11].
2) User Interface Analysis, which is conducted to examine the user-model interface and determine if it is human engineered so as to prevent occurrences of errors during the user's interactions with the model[11].

Interface Testing: Interface Testing consists of the following techniques:

1) Data Interface Testing, which assesses the accuracy of data inputted into the model or outputted from the model during its execution [14].
2) Model Interface Testing, which detects model representation errors caused due to interface errors [11].
3) User Interface Testing, which deals with the assessment of the interactions between the user and the model, and detects errors associated with those [27].

Lambda (λ) Calculus: λ-calculus is a mathematical tool for formally defining systems [28]. λ-calculus offers function that can be translated into validation rules.

Logical Deduction: Logical Deduction, also known as Deductive Reasoning, is similar to Induction but the conclusions are considered as logically true, or valid, if every step of the model is valid and the model terminates [26].

Object-Flow Testing: Object-Flow Testing assesses model accuracy by exploring the life cycle of an object during the model execution [11].

Partition Testing: Partition Testing, also known as equivalent class partitioning, partitions the model into functional representatives (partitions), assuming that all elements within each partition bear the same properties, and then, by selecting a representative element from each partition, each partition and subsequently the model is validated, thus eliminating the need for exhaustive validation [29].

Predicate Calculus: Predicate Calculus quantifies simple relationships (predicates) using boolean variables. Since, the model can be defined based on predicates, then its validation can be performed by manipulating these predicates [11]. Similarly to Deduction, Predicate Calculus' conclusions are logically true or valid.

Predicate Transformations: Predicate Transformations, or more formally known as Predicate Transformer Semantics, show that systems (in this case a simulation model) can achieve their goals, i.e., they are valid. Predicate Transformations associate a pre-condition to any post-condition, or in other words transform model output states into all model input states, thus providing the basis for proving model correctness [30].

Predictive Validation: In Predictive Validation, the model executes with past input data and the results are then compared with data from the real system [31].

Product Testing: Product Testing is a preparatory step for the Acceptance Testing, in which all requirements specification are tested in the same way as in the Acceptance Testing, with the only difference being that the test takes place within the development team whereas Acceptance Testing takes place at the client's premises [27].

Proof of Correctness: A Proof of Correctness expresses the model in a precise notation and then proves that the model terminates and thus satisfies the requirements specification with sufficient accuracy [32].

Regression Testing: Regression Testing ensures that correcting errors in the model during the validation process do not create new errors or adverse side-effects [11].

Reviews: Reviews are similar to an inspection but the review team also involves managers. Reviews are intended to give management and study sponsors evidence that the model development is being conducted according to the study objectives [12].

Semantic Analysis: Semantic Analysis attempts to determine the modeler's intent in writing the code [33].

Sensitivity Analysis: In Sensitivity Analysis, selected variables in the model are given different values (within a predetermined range) in order to observe the behavior of the model with regards to these changes [23].

Special Input Testing: Special Input Testing assesses model accuracy by subjecting the model in a variety of inputs and consists of the following techniques:

1) Boundary Value Testing, which tests the boundary values of the input and output equivalence classes (a set of values that bear similar characteristics and one value can act as a representative for the whole set)

[15].

2) Equivalence Partitioning Testing, which tests the model by partitioning input data into equivalence classes [12].
3) Extreme Input Testing, which tests the model based on extreme input values (minimum, maximum, or a mixture of those) [11].
4) Invalid Input Testing, which tests the model using incorrect input data [11].
5) Real-Time Input Testing, which tests the model using real-time input data from the real system [11].
6) Self-Driven Input Testing, which test the model by executing it under input data randomly sampled from probabilistic models representing random phenomena of a real system [11].
7) Stress Testing, which tests the model by subjecting it into heavy loads, like large volumes of data, intense activity over a short time span etc [15].
8) Trace-Driven Input Testing, which tests the model by executing it under input trace data collected from a real system [11].

Structural (White-box) Testing: Structural Testing is used to evaluate the internal structure of the model and consists of the following techniques:

1) Branch Testing, which tests the model under test data in order to execute as many branch alternatives as possible [13].
2) Condition Testing, which tests the model under test data in order to execute as many logical conditions as possible [11].
3) Data Flow Testing, which tests the model by using the control flow graph as to explore sequences of events related to the status of data structures and to examine data-flow anomalies [13].
4) Loop Testing, which tests the model under test data in order to execute as many loop structures as possible [16].
5) Path Testing, which tests the model under test data in order to execute as many control flow paths as possible [13].
6) Statement Testing, which which tests the model under test data in order to execute as many statements as possible [13].

Structural Analysis: Structural Analysis is used to examine the model structure and to determine if it adheres to structured principles [11].

Submodel/Module Testing: Submodel/Module Testing is a top-down form of testing in which is submodel is tested against its corresponding subsystem [11].

Symbolic Debugging: Symbolic Debugging is a verification method in which the use of "breakpoints" allows for a direct manipulation of the model execution while viewing the model at the source code level [11].

Symbolic Evaluation: Symbolic Evaluation assesses model accuracy by executing the model using as an input symbolic values and not the actual data values [34].

Syntax Analysis: Syntax Analysis assures that the mechanics of the programming language are applied correctly [13].

Top-Down Testing: In Top-Down Testing, the model testing starts from the submodels at the hishest level and moves downwards into the base submodels [35].

Traceability Assessment: Traceability Assessment matches, one to one, the elements of one form of the model to another [14].

Turing Test: In a Turing Test, experts are presented with two sets of output data, i.e., the model and reality, and without knowing which one is which, they are asked to differentiate the two [36].

Visualization/Animation: In Visualization/Animation, the model is tested by observing different graphs of the internal or external behavior of the model [11].

Walkthroughs: Walkthroughs are used to detect and document faults. WhilstThey are similar to an Inspection but less time consuming, they have fewer phases [15].

2) Statistical Techniques: In the statistical formulas shown in this section, wherever *M* and *R* are used as subscripts, they denote that the particular variable refers to the model or reality respectively. Moreover, unless explicitly stated, n with the appropriate subscript denotes the respective sample size.

t-Test: The t-Test, also known as Student's t-test, is a statistical hypothesis test, which determines whether the mean of a variable is significantly different from a constant value (one-sample test) or whether the mean of two variables is significantly different (two-sample test) [37]. The most common usage of t-test in simulation model V&V is the two-sample test (Model and Reality) with unequal sample sizes and variances. The latter is also known as Welch t-test [38] and its formula is:

$$t = \frac{\overline{X}_M - \overline{X}_R}{\sqrt{\frac{s_M^2}{n_M} + \frac{s_R^2}{n_R}}} \tag{1}$$

where $\overline{X}$ and s are the mean and variance respectively. The t-test is one of the most commonly used tests for the equality of means between model and reality.

Hotelling's T^2 Test: Hotelling's T^2 test is a generalization of the t-test for multivariate hypothesis testing [39]. As it is the case with t-test, Hotelling's T^2 test can also be used for one- or two-sample testing. Its formula for the two-sample test is:

$$T^2 = (\overline{X}_M - \overline{X}_R)'\left\{S_p\left(\frac{1}{n_M} + \frac{1}{n_R}\right)\right\}^{-1}(\overline{X}_M - \overline{X}_R) \tag{2}$$

where

$$X_i = \frac{1}{n_i}\sum_{j=1}^{n_i} X_{ij}, i = \{M, R\} \tag{3}$$

$$S_i = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(X_{ij} - \overline{x}_i)(X_{ij} - \overline{x}_i)' \tag{4}$$

$$S_p = \frac{(n_M - 1)S_M + (n_R - 1)S_R}{n_M + n_R - 2} \tag{5}$$

Analysis of Variance (ANOVA): ANOVA is a collection of statistical techniques for testing mean equality between three or more datasets [40]. It is similar to multiple two-sample t-tests but less prone to a Type I error. The most popular ANOVA test is the F-Test. In a nutshell, the F-Test is the ratio of the variability between the datasets to the variability within each dataset [41]. The formula is:

$$F = \frac{\sum_{i=1}^{K} n_i(\overline{Y}_i - \overline{Y})^2/(K-1)}{\sum_{i=1}^{K}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_i)^2/(N-K)} \tag{6}$$

where Y_i is the average of the i^{th} dataset, $\overline{Y}$ the overall average of the data, K the number of datasets, Y_{ij} the j^{th} observation of the i^{th} dataset, and N the total sample size.

Multivariate Analysis of Variance (MANOVA): MANOVA is similar to ANOVA but for cases where the dependent variables are more than one [42]. One of the most popular MANOVA tests is the Samuel Stanley Wilks' statistic, which is a summary based on the eigenvalues λ_p of the A matrix ($A = \sum_M * \sum_{res}^{-1}$), where $\sum_M$ is the model variance matrix and $\sum_{res}$ the error variance matrix. Wilks' formula is:

$$\Lambda_{Wilks} = \prod_{1...p}(1/(1+\lambda_p)) = det\Big(\sum_{res}\Big)/det\Big(\sum_{res}+\sum_{M}\Big) \quad (7)$$

and is distributed as Λ.

Simultaneous Confidence Intervals: Balci and Sargent [43] proposed the validation method of simultaneous confidence intervals (sci) for simulation models with multiple outputs. The sci are formed by the confidence intervals of each model output. They described three approaches for calculating the sci and choosing one approach over the others depends on whether the model is self- or trace-driven. In other words, the choice of the approach depends on whether the model's input data are coming from the same population as the system's input data but they are different or whether the model's input data are exactly the same as the system's.

Factor Analysis: Using factor analysis, *p* observed random variables can be expressed as linear functions of *m* (*m* < *p*) random variables, also called common factors, along with an error [44]. If $x = \{x_1, x_2, \ldots, x_p\}$ are the observed variables, $f = \{f_1, f_2, \ldots, f_m\}$ the common factors, and $e = \{e_1, e_2, \ldots, e_p\}$ the error, then there exists a

$$K = \begin{bmatrix} \kappa_{11} & \kappa_{12} & \ldots & \kappa_{1m} \\ \kappa_{21} & \kappa_{22} & \ldots & \kappa_{2m} \\ & \cdots\cdots\cdots\cdot & & \\ \kappa_{p1} & \kappa_{p2} & \ldots & \kappa_{pm} \end{bmatrix} \quad (8)$$

so $x = Kf + e$.

Principal Component Analysis (PCA): The idea behind PCA is that if there is a large number (*p*) of random correlated variables, orthogonal transformation can be used to convert these variables into a significantly smaller number (*m*) of uncorrelated variables, called principal components [44]. PCA is similar to factor analysis, and is often considered to be a method of factor analysis. Despite their similarities, PCA and factor analysis are different in the sense that PCA concentrates on the diagonal elements of the covariance matrix, i.e., the variances, whereas the factor analysis focuses on the non-diagonal elements. In mathematical terms, PCA can be defined as follows:

$$f_1 = a_1'x = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p = \sum_{j=1}^{p} a_{1j}x_j \quad (9)$$

$$f_2 = a_2'x = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p = \sum_{j=1}^{p} a_{2j}x_j \quad (10)$$

$$\vdots$$

$$f_m = a_m'x = a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mp}x_p = \sum_{j=1}^{p} a_{mj}x_j \quad (11)$$

where f is the m principal components, a' is a transposed vector of constants, and x is the p independent variables. It should be noted that PCA is particularly useful when $m \ll p$.

Kolmogorov-Smirnov Test: The Kolmogorov-Smirnov test (K-S test) is a non-parametric goodness-of-fit test that it can be one-sample, i.e., test whether a sample is distributed according to a known theoretical distribution (e.g., normal, binomial etc.), or two-sample, i.e., test whether two different samples are drawn from the same empirical distribution [45]. In simulation model V&V, the two-sample K-S test is the most common, i.e., comparing whether the data from the model and from reality are derived from the same distribution. The two-sample K-S test is calculated as follows:

$$D_{n_M,n_R} = sup_x|F_{M,n_M}(x) - F_{R,n_R}(x)| \quad (12)$$

where F denotes the empirical distribution of each dataset, which is calculated as follows:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{[-\infty,x]}(X_i) \quad (13)$$

where

$$I_{[-\infty,x]}(X_i) = \begin{cases} 1, & \text{if } X_i \leq x \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

Finally, the null hypothesis is rejected for a given α level if:

$$D_{n_M,n_R} > C(\alpha)\sqrt{\frac{n_M + n_R}{n_M * n_R}} \quad (15)$$

where $c(\alpha)$ is given in the Kolmogorov-Smirnov table.

Chi-square Test: The chi-square (χ^2) test is also a goodness-of-fit test which, similarly to the K-S test, it can also be a one- or two-sample test. The idea behind a two-sample chi-square test, which is more commonly used in model V&V, is that the simulation and operational data are partitioned in i bins, and then the number of points in each bin is observed on whether it is similar on both datasets [46]. Accepting the null hypothesis (H_0) means that the samples are drawn from the same distribution. The chi-square test can be calculated as follows:

$$\chi^2 = \sum_{i=1}^{k} \frac{(K_M x_{Mi} - K_R x_{Ri})}{x_{Mi} + x_{Ri}} \quad (16)$$

which follows the chi-squared distribution, and where i is the number of bins, x_{Mi} and x_{Ri} the observed values from the model and reality respectively, and K_M and K_R constants adjusting the inequality of the observations of the two datasets, which are calculated as follows:

$$K_M = \sqrt{\frac{\sum_{i=1}^{k} x_{Ri}}{\sum_{i=1}^{k} x_{Mi}}} \quad (17)$$

$$K_R = \sqrt{\frac{\sum_{i=1}^{k} x_{Mi}}{\sum_{i=1}^{k} x_{Ri}}} \quad (18)$$

Anderson–Darling Test: The Anderson–Darling test belongs to the class of quadratic empirical distribution function (EDF) statistics, which determine whether a sample is drawn from a specific distribution (one-sample) or whether two samples are drawn from the same distribution (two-sample) [47]. The two-sample formula of the test is calculated as follows [48]:

$$AD = \frac{1}{n_M n_R} \sum_{i=1}^{n_M+n_R} (N_i Z_{(n_M+n_R-n_m i)})^2 \frac{1}{i Z_{n_M+n_R-i}} \tag{19}$$

where $Z_{n_M+n_R}$ is the combined and ordered samples of the model and reality and N_i the number of observations in the model that are equal to or smaller than the i^{th} observation in $Z_{n_M+n_R}$.

Cramér–von Mises Criterion: The Cramér–von Mises criterion also belongs to the class of quadratic EDF statistics and is quite similar to the Anderson–Darling test [49]. Compared to the Cramér–von Mises criterion, the Anderson–Darling test places more weight on observations in the tails of the distribution. The two-sample Cramér–von Mises criterion is calculated as follows:

$$T = \frac{U}{n_M n_R (n_M + n_R)} - \frac{4 n_M n_R - 1}{6(n_M + n_R)} \tag{20}$$

where

$$U = n_M \sum_{i=1}^{n_M} (r_i - i)^2 + n_R \sum_{j=1}^{n_R} (s_j - j)^2 \tag{21}$$

and $(r_1, r_2, ..., r_{n_M})$ and $(s_1, s_2, ..., s_{n_R})$ the ranks of the sorted samples of the model and reality respectively.

Kuiper's Test: Kuiper's test is a goodness-of-fit test similar to the Kolmogorov-Smirnov test (K-S test) in the sense that it compares two cumulative distribution functions. Compared to the K-S test, Kuiper's test is sensitive not only to the median but also to the tail. Compared to the The Anderson–Darling test, which also provides equal sensitivity at the tails and at the median, Kuiper's test is invariant under cyclic transformations of the independent variable [50]. Kuiper's test is calculated as follows:

$$V = D_+ + D_- \tag{22}$$

where

$$D_+ = max_{-\infty<x<\infty}[S_M(x) - S_R(x)] \tag{23}$$

$$D_- = max_{-\infty<x<\infty}[S_R(x) - S_M(x)] \tag{24}$$

$$S_M(x_i) = \frac{i - n_M}{n_M} \tag{25}$$

$$S_R(x_i) = \frac{i - n_R}{n_R} \tag{26}$$

Coefficient of Determination (R^2): R^2 is yet another goodness-of-fit test that indicates the proportion of the variance of the dependent variable that is predicted from the independent variable or variables. The most commonly used extension of R^2 is the adjusted R^2 ($\overline{R}^2$), which adjusts for the number of explanatory terms in a model relative to the number of data points [51]. $\overline{R}^2$ is calculated as follows:

$$\overline{R}^2 = 1 - (1 - R^2)\frac{n_M - 1}{n_M - k - 1} \tag{27}$$

where

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} \tag{28}$$

$$SS_{residual} = \sum_{i=1}^{n_M} e_i^2 \tag{29}$$

$$SS_{total} = \sum_{i=1}^{n_M} (y_i - \overline{y})^2 \tag{30}$$

and *k* is the number of independent variables. The closer $\overline{R}^2$ is to one, the better the model is considered, since the results are explained in a large degree from the variation of the dependent variables and not from the residuals.

Mann-Whitney-Wilcoxon Test: The Mann-Whitney-Wilcoxon (MWW) test, also known as Mann–Whitney U test, is a non-parametric test that tests whether two samples derive from populations having the same distribution [52]. The MWW test can be calculated by first sorting all values from both datasets in an ascending order and assigning numeric ranks starting with 1 from the end of this sorted list. Then, the MWW values for both datasets are computed as follows:

$$U_M = R_M - \frac{n_M(n_M + 1)}{2} \tag{31}$$

$$U_R = R_R - \frac{n_R(n_R + 1)}{2} \tag{32}$$

where R indicates the sum of the ranks for each dataset. Finally, in order to determine whether the two samples derive from the same population, the minimum value between U_M and U_R is compared with the value from the tables.

White Test: The White test is a test for determining whether the variance of a model is constant, i.e., whether the model is homoscedastic (H_0) [53]. The White test is calculated as follows:

$$\hat{e}_i^2 = \delta_0 + \delta_1 \hat{Y}_i + \delta_2 \hat{Y}_i^2 \tag{33}$$

where Y_i are the predicted dependent variables of the model. Upon calculating δ_0, δ_1, and δ_2, the $R^2_{\bar{e}^2}$ can be computed and then the $\chi^2 = n_M R^2_{\bar{e}^2}$, which can then be tested with 2 degrees of freedom against the null hypothesis.

Glejser Test: The Glejser test also tests for Heteroscedasticity but instead of using the square of the residuals, it uses their absolute values [54]. The Glejser test is calculated as follows:

$$|e_i| = \gamma_0 + \gamma_1 f(x_i) + u_i \tag{34}$$

in which case the most common values for the $f(x_i)$ are: $f(x_i) = x_i$, $f(x_i) = \sqrt{x_i}$, and $f(x_i) = \frac{1}{x_i}$. The γ_1 of the equation with the highest R^2 is then tested and if it is found statistically significant, the null hypothesis of homoscedasticity is rejected.

Spectral Analysis: Spectral analysis tests whether two time series are equivalent [55]. Spectral analysis is a relatively complex statistical test, especially compared to the tests presented so far, and it is calculated as follows:

$$g_i(f) = \frac{1}{\pi}\left[2\sum_{p=1}^{L} k_L(p) C_i(p) cos(f_i(p)) + C_i(0)\right] \tag{35}$$

where $i = \{M, R\}$. $C_i(p)$ is the autocovariance function

$$C_i(p) = \frac{1}{T-p}\sum_{t=1}^{T-p}(x_t - m)(x_{t+p} - m) \tag{36}$$

$k_L(p)$ is a Bartlett weighting function for which several possibilities exists [56], and

m = mean of $X(t)$
T = total time period
X_t = observation at time t
f = frequency in cycles per unit of time
L = number of lags
p = number of time periods separating correlated observations (1,2,...,L-1)

Finally, in order to determine whether the two time series are equivalent, i.e., not rejecting the null hypothesis, the ratio $g_M(f)/g_R(f)$ should satisfy the inequality:

$$e^{-\phi} \leq \frac{g_M(f)}{g_R(f)} \leq e^{\phi} \tag{37}$$

where

$$\phi = Z_{\alpha/2}(4L/3T)^{1/2} \tag{38}$$

and $Z_{\alpha/2}$ = the two tail critical value for the standard normal distribution at a significance level of α.

Durbin–Watson Statistic: The Durbin–Watson statistic tests for the existence of autocorrelation in the residuals from a regression analysis [57]. The statistic is calculated as follows:

$$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=2}^{T} e_t^2} \tag{39}$$

where *T* is the number of observations. The value *d* is compared to the lower and upper critical values ($d_{L,a}$ and $d_{U,a}$) to test for positive or negative autocorrelation.

The statistical techniques described above as just a sample of the available techniques for simulation model V&V. Nevertheless, it is a representative sample that can be used in the majority of the cases. The aim of this section is to illustrate the various statistical techniques, which facilitates the categorization of these techniques and thus the selection of the most suitable ones given the problem at hand.

B. Step 2: Simulations' and systems' properties and characteristics

This step aims at identifying the properties and characteristics of simulations and the real-world system (hereinafter referred to as system) under study that can potentially influence the selection of methods and techniques.

1) Simulations' properties: Since simulations differ from one another in various ways, distinctions are made on whether they represent an existing system, or whether they simulate a system at a microscopic or macroscopic level, or whether they are intended for learning or decision making, and so forth. This is an indication that simulations can be characterized by various properties. Based on literature, this study has identified 10 properties of simulations. The rationale behind selecting those properties was to describe simulations with as much detail as possible. Hence, the properties span on multiple levels. Not all identified properties necessarily influence the selection of V&V methods, therefore, this step is not only about identifying the properties themselves but also determining which are the ones that really influence the effectiveness of a method; in other words, this step serves as the rationale for choosing those properties of simulations that are applicable to specific V&V methods, and provides for the reasons behind this selection.

The 10 identified properties of simulations are the following:

1) Access to the source code of the simulation. Accessibility, or lack of it, influences the selection of a V&V method [58], since several methods require some sort of a check on the code level. Hence, this property is included in the analysis.
2) The simulation represents an existing real-system for which real data exist [59]. The existence of, or more importantly the lack of, real data heavily influences the selection process since several methods require real data and thus cannot be used when no real data is available. Hence, this property is also included in the analysis.
3) The formalism the simulation is based on, like Discrete Event System Specification (DEVS), Differential Equation Specified System (DESS), System Dynamics, etc. [60]. Several frameworks and methods have been proposed on how to verify and validate DEVS [61][62], DESS [63][64], or system dynamics models [65][66], but they are either application specific or the same method can be used in more than one formalisms, making it independent of the actual formalism. Therefore, while formalisms are an important aspect of simulation modeling, their influence on the V&V method selection is minimal, ergo excluded from the analysis.
4) The simulation's worldviews: i) Process Interaction/Locality of Object, ii) Event Scheduling/Locality of Time, iii) Activity Scanning/Locality of State [67]. While worldviews allow for more concise model descriptions by allowing a model specifier to take advantage of contextual information, there is not any evidence from a literature point of view that they have an influence on the V&V method selection, hence, they are excluded from the analysis.
5) The fidelity level of the simulation (Low, Medium, High) [68]. While from a literature point of view there is no evidence to support the influence of the level of fidelity on the V&V method selection, common sense dictates that there must be some. Indeed, in order to characterize a simulation as of high fidelity, it must imitate an existing system and real-world data must exist, thus making the comparison and the final characterization possible. Therefore, as discussed in the second property and shown in Table I, the existence of data of the real system influences the V&V method selection, as does the level of fidelity. Yet, since the correlation between real data and high fidelity is almost 1-to-1, the fidelity level is excluded from the analysis for reasons of simplification.
6) The type of the simulation (Constructive, Virtual, Live) [69]. This classification, which is adopted by the U.S. Department of Defense [17], should be seen more as a continuum rather than as a discrete characterization. Once a simulation moves towards the Virtual or the Live side of the continuum, it can

also be referred to as 'game'. A game has the distinct characteristic that the game session is succeeded by debriefing, whereby the participants reflect upon the game session to link the content presented during the session with reality [70]. It has been demonstrated that debriefing can in general facilitate validation [71][72]. Moreover, while all methods identified in this paper are suitable for pure simulations (constructive), not all of them are appropriate for games. It would be interesting to examine which of the methods can also be used for validating games. Hence, this property is included in the analysis.

7) The purpose the simulation was built for (learning, decision making, etc.). Several case studies on V&V of simulations for different purposes have been reported; in training [73][74], in decision making [75], in concept testing [76], etc., but there are no reports of specific V&V methods being more effective for a certain purpose. Hence, this property is excluded from the analysis.
8) The simulation imitates a strictly technical, a socio technical system (STS), or a complex adaptive system (CAS) with multiple agents. There are several studies on modeling and validating simulations for STS [77] and CAS with multiple agents [78][79] but there are no indications that certain V&V methods are more effective for an STS or a CAS. Therefore, this property is excluded from the analysis.
9) The application domain of the simulation (logistics, business, physics, etc.). Although the application domain of the simulation plays a significant role in the modeling process, since different approaches are required (Newtonian physics for object movement, Navier–Stokes equations for fluid behavior, etc.) for modeling different systems [80], literature, or more precisely the lack of it, suggests that the V&V process and thus the V&V method selection is not affected by the application domain. Hence, this property is excluded from the analysis.
10) The functional (hard goals) and non-functional (soft goals) requirements of the simulation [81]. Validating the simulation's requirements is indeed an important part of the V&V process [82], since validation is always relative to the intended use [83], in other words the use defined in the requirements. Hence, making a distinction between the hard and soft goals is paramount and as such this property is included in the analysis.

2) Simulations' and systems' characteristics: Simulations and the systems they imitate can produce a variety of data, which can be characterized in various ways. Moreover, depending on the type of data and on the purpose of the V&V study, different statistical tests are usually necessary, which in turn depend on the produced output. Based on the literature review on the techniques presented in Section II-A2, the characteristics of simulations and systems that influence the selection of techniques are the following:

1) Number of datasets. The most usual case in simulation model validation is to have two datasets (model and reality). Nevertheless, there are cases where the number of datasets can be either one, e.g., when testing whether the model derives from a known distribution like the normal or gamma distributions, or more than two, e.g., when testing the results of more than one models against the operational data.
2) Number of variables. The most usual case in simulation model validation is to test one variable, e.g. in railway simulations, this variable is usually the amount of delay. Nevertheless, there are cases where the number of testing variables is more than one, e.g. simultaneously testing longitude and latitude values between model and reality.
3) Purpose of the statistical technique. A statistical technique can test for equality of means, the extent to which the data from the model and reality are similarly distributed, the extent to which two time series are equivalent, or it can be used to reduce the model's complexity.
4) Known parameters. Statistical techniques are divided in two major categories: parametric and non-parametric. Parametric techniques are the ones that require the mean and variance (μ, σ^2) to be known, whereas non-parametric techniques can deal with cases where these parameters are not known.
5) Type of data. The type of data simulations and systems produce range from strictly quantitative to purely qualitative. Usually, statistical techniques suitable for a V&V study should be able to deal with data that are either numerical or categorical (binary).
6) Size of samples. Simulation and system data are almost impossible to be normally distributed. Nevertheless, due to the Central Limit Theorem [84], when the size of a sample exceeds 30 (or 40 depending how close to be normally distributed the data are), it is assumed that it follows the normal distribution thus the techniques that work for the normal distribution are applicable.

C. Step 3: Phases of a simulation study

According to Sargent [85], there are 4 distinct phases of V&V: *Data Validation*, *Conceptual Model Validation*, *Model Verification*, and *Operational Validation*. *Data Validation* is concerned with the accuracy of the raw data, as well as the accuracy of any transformation performed on this data. *Conceptual Model Validation* determines whether the theories and assumptions underlying the conceptual model are correct, and whether the model's structure, logic, and mathematical and causal relationships are "reasonable" for the intended purpose of the model. *Model Verification* ensures that the implementation of the conceptual model is correct. Finally, *Operational Validation* is concerned with determining that the model behaves accurately based on its intended purpose. This study adopts Sargent's [85] characterization and aims at using it to classify the methods, in addition to the simulations' properties.

D. Conclusion of the Literature Review

It is evident that selecting one method or technique over another for a V&V study depends on several characteristics of the simulation, the system, the methods, and the techniques, as well as the phase of the simulation study. In Section III, a methodology that combines all three steps aiming at the development of a framework for method and technique selection is proposed.

III. Methodology

In this section, a methodology for selecting the most appropriate V&V methods (Section III-A) and statistical techniques (Section III-B) for a V&V study is proposed.

A. V&V method selection methodology

As discussed in Section II-B1, dimensions 3, 4, 5, 7, 8, and 9 are perceived to have little influence on the method selection, hence, there are excluded from the analysis. On the other hand, the purpose of the method selection, discussed in Section II-C, seems to be crucial; in other words, it is important to differentiate on whether the selected method will be used for data validation, conceptual model validation, model verification, or operational validation. Therefore, the list of the dimensions is refined, and is expressed in questions, as follows:

1) Does the V&V method require access to the simulation model's source code?
Possible answers: Yes or No. A positive answer to this question means that this method can only be used when the person or persons performing the V&V have access to the simulation's source code, whereas a negative answer means that it can be used in any occasion regardless of the accessibility to the simulation model's source code. It should be noted that the current study - and consequently this dimension - is not concerned with the specific programming language the simulation is built on (Assembly, C++, NetLogo, etc.), but solely with whether the application of a V&V method depends upon having access to the source code.
2) Does the V&V method require data from the real system?
Possible answers: Yes or No. A positive answer to this question means that this method can only be used when data from the real system are available, whereas a negative answer means that it can be used in any occasion regardless of the availability of data from the real system. It should be noted that the current study - and consequently this dimension - is not concerned with the nature of the data in general (qualitative or quantitative), but solely with their existence and availability.
3) Is the V&V method suitable for a game V&V study?
Possible answers: Yes or No. While all methods are suitable for pure simulations, some of them will be also suitable for games in particular. Although games often have a simulation model running on the background, in which case all methods would be applicable, in this study the term *game* is used to describe the layer that is on top of the simulation model and refers to the players' interaction.
4) For what type of requirements is the V&V method more suitable?
Possible answers: Hard (Functional), or Soft (Non-Functional), or Both. A method might be focused on either the functional part or the non-functional part of the model or on both.
5) For which type of study is the V&V method more suitable?
Possible answers: Data Validation (D. Val.), Conceptual Model Validation (C.M. Val.), Model Verification (M. Ver.), or Operational Validation (O. Val.). A method might be suitable for one or more of the available categories.

Table I summarizes the results of the analysis. The intended use of Table I is to act as a filtering mechanism. Whenever an individual or a team wants to verify and/or validate a simulation model, they can utilize this table to narrow down the applicable methods according to the different properties of the simulation at hand. The selection process is shown in Figure 1.

With regards to the first property, i.e., the accessibility to the source code, and in contrary to the second property, access to the source code does not imply that the methods categorized under "Yes" are stronger. Usually, access to the source code is associated with verification and in some cases conceptual model validation.

With regards to the second property, i.e., the availability of data from the real system, the methods categorized under "No" can be used irrespective of whether real data exist. Nevertheless, the methods categorized under "Yes" are more powerful in the sense that, if used appropriately, they provide evidence or a data trace of how the simulation should work. Hence, whenever real data are available, the methods categorized under "Yes" should be preferred, unless an alternative method is definitely more suitable.

With regards to the third property, i.e., the suitability of certain methods for the V&V of games, informal methods [11] seem to be the ones suitable for games. This is a preliminary conclusion that is expected to an extent. In games representing Complex Adaptive Systems (CAS), experts' opinion plays an important, and perhaps the most important, role [86], regardless of the game's level of fidelity [87] or use of technology [88]. It should be noted that although the term *Games* assumes both high-tech (computer-based) and low-tech (e.g., tabletop) games, the selection in Table I was made with a bias towards the high-tech games.

With regards to the fourth and fifth property, i.e., the type of requirements being tested and the purpose of the V&V study respectively, the answers are more or less self-explanatory. Some methods are more suitable for testing one type of requirement. As an example, regression testing is more appropriate for functional requirements (hard goals). Other V&V methods are better suited for one purpose, such as Structural (White-box) testing, which is more appropriate for conceptual model validation, while others are more suitable for testing both types of requirements (e.g., Graphical comparisons), or for more than one purpose (e.g., Trace-Driven Input Testing).

The novelty of the proposed framework does not lie in the content of Table I per se, but on the idea that the list of methods can be narrowed down to a manageable level, thus making the V&V of a simulation better grounded, faster, more accurate, and more cost effective.

There is a threat towards the validity of the content on Table I. The line between whether data from the real system are needed, or whether access to the source code is needed, or whether a specific requirement is definitely functional or non-functional, or whether the purpose is to validate the data, the conceptual model, the operational ability of the model, or to just verify the model, is not always clear and well defined. In Section V, future steps are proposed aiming at addressing and mitigating the above mentioned threat.

TABLE I. LIST OF V&V METHODS & PROPERTIES OF SIMULATIONS.

Method	Source Code	Real Data	Game	Requirements	Purpose
Acceptance Testing	No	No	Yes	Both	O. Val.
Alpha Testing	No	No	Yes	Both	O. Val.
Assertion Checking	Yes	No	No	Hard	M. Ver.
Audit	Yes	No	Yes	Soft	M. Ver.
Beta Testing	No	No	Yes	Both	O. Val.
Bottom-Up Testing	Yes	No	No	Both	M. Ver.
Cause-Effect Graphing	Yes	No	No	Hard	M. Ver.
Comparison Testing	No	No	No	Both	C.M. Val.
Compliance Testing: → Authorization Testing	No	No	No	Soft	M. Ver.
→ Performance Testing	No	No	No	Soft	M. Ver.
→ Security Testing	No	No	No	Soft	M. Ver.
→ Standards Testing	No	No	No	Soft	M. Ver.
Control Analysis: → Calling Structure Analysis	Yes	No	No	Hard	C.M. Val.
→ Concurrent Process Analysis	Yes	No	No	Hard	M. Ver.
→ Control Flow Analysis	Yes	No	No	Hard	C.M. Val.
→ State Transition Analysis	Yes	No	No	Hard	D. Val. & M. Ver.
Data Analysis: → Data Dependency Analysis	Yes	No	No	Hard	D. Val. & M. Ver.
→ Data Flow Analysis	Yes	No	No	Hard	D. Val. & M. Ver.
Debugging	Yes	No	No	Both	M. Ver.
Desk Checking	Yes	No	Yes	Both	M. Ver.
Documentation Checking	Yes	No	Yes	Both	C.M. Val.
Execution Testing: → Execution Monitoring	No	No	No	Hard	C.M. Val.
→ Execution Profiling	No	No	No	Hard	C.M. Val.
→ Execution Tracing	Yes	No	No	Hard	C.M. Val.
Face Validation	No	Yes	Yes	Both	O. Val.
Fault/Failure Analysis	No	No	No	Hard	C.M. Val.
Fault/Failure Insertion Testing	No	No	No	Hard	C.M. Val.
Field Testing	No	Yes	No	Both	O. Val.
Functional (Black-Box) Testing	No	Yes	No	Hard	C.M. Val.
Graphical Comparisons	No	Yes	Yes	Both	O. Val.
Induction	No	No	No	Both	C.M. Val.
Inference	No	No	No	Both	C.M. Val.
Inspections	No	No	No	Both	C.M. Val.
Interface Analysis: → Model Interface Analysis	No	No	No	Soft	C.M. Val.
→ User Interface Analysis	No	No	Yes	Soft	O. Val.
Interface Testing: → Data Interface Testing	No	No	No	Soft	D. Val.
→ Model Interface Testing	No	No	No	Soft	C.M. Val.
→ User Interface Testing	No	No	Yes	Soft	O. Val.
Lambda Calculus	Yes	No	No	Hard	M. Ver.
Logical Deduction	No	No	No	Both	All
Object-Flow Testing	No	No	No	Hard	O. Val.
Partition Testing	Yes	No	No	Hard	C.M. Val.
Predicate Calculus	Yes	No	No	Hard	M. Ver.
Predicate Transformations	No	Yes	No	Hard	M. Ver.
Predictive Validation	No	Yes	No	Hard	O. Val.
Product Testing	No	No	Yes	Both	O. Val.
Proof of Correctness	Yes	No	No	Hard	C.M. Val. & M. Ver.
Regression Testing	Yes	No	No	Hard	M. Ver.
Reviews	No	No	Yes	Both	C.M. Val.
Semantic Analysis	Yes	No	No	Both	M. Ver.
Sensitivity Analysis	No	No	No	Hard	O. Val.
Special input testing: → Boundary Value Testing	Yes	No	No	Both	M. Ver.
→ Equivalence Partitioning Testing	No	No	No	Hard	O. Val.
→ Extreme Input Testing	No	No	No	Hard	O. Val.
→ Invalid Input Testing	No	No	No	Hard	O. Val.
→ Real-Time Input Testing	No	Yes	No	Hard	O. Val.
→ Self-Driven Input Testing	No	No	No	Hard	O. Val.
→ Stress Testing	No	No	No	Hard	O. Val.
→ Trace-Driven Input Testing	Yes	Yes	No	Both	D. Val. & C.M. Val.
Structural (White-box) Testing: → Branch Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
→ Condition Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
→ Data Flow Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
→ Loop Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
→ Path Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
→ Statement Testing	Yes	No	No	Both	C.M. Val. & M. Ver.
Structural Analysis	No	No	No	Hard	C.M. Val.
Submodel/Module Testing	No	No	No	Both	C.M. Val.
Symbolic Debugging	Yes	No	No	Hard	M. Ver.
Symbolic Evaluation	Yes	No	No	Hard	C.M. Val.
Syntax Analysis	Yes	No	No	Hard	M. Ver.
Top-Down Testing	Yes	No	No	Both	C.M. Val.
Traceability Assessment	Yes	Yes	No	Both	C.M. Val.
Turing Test	No	Yes	No	Both	O. Val.
Visualization/Animation	No	Yes	Yes	Both	O. Val.
Walkthroughs	No	No	Yes	Both	C.M. Val.

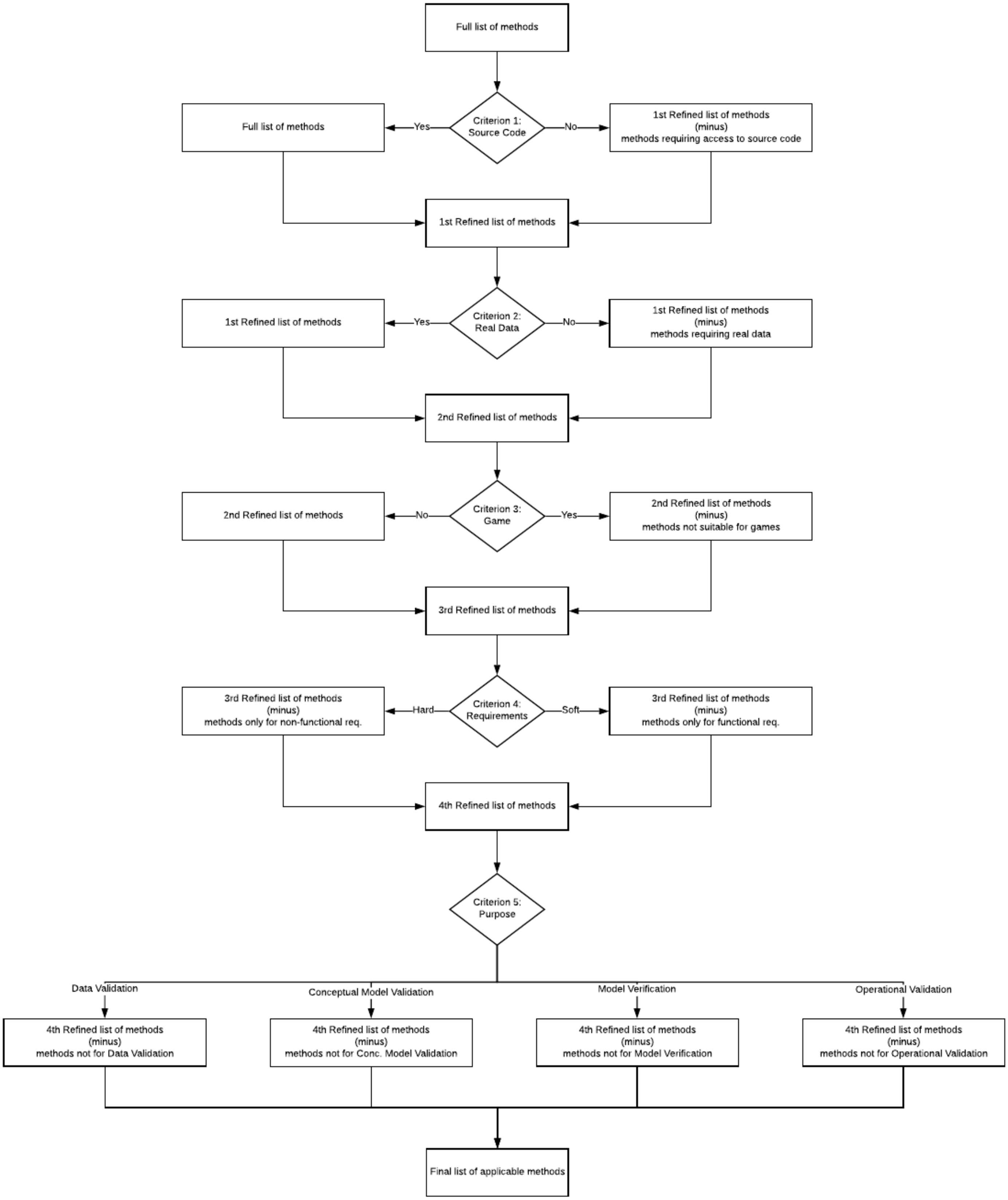
Full list of methods
Criterion 1: Source Code
Yes
No
Full list of methods
1st Refined list of methods (minus) methods requiring access to source code
1st Refined list of methods
Criterion 2: Real Data
Yes
No
1st Refined list of methods
1st Refined list of methods (minus) methods requiring real data
2nd Refined list of methods
Criterion 3: Game
No
Yes
2nd Refined list of methods
2nd Refined list of methods (minus) methods not suitable for games
3rd Refined list of methods
Criterion 4: Requirements
Hard
Soft
3rd Refined list of methods (minus) methods only for non-functional req.
3rd Refined list of methods (minus) methods only for functional req.
4th Refined list of methods
Criterion 5: Purpose
Data Validation
Conceptual Model Validation
Model Verification
Operational Validation
4th Refined list of methods (minus) methods not for Data Validation
4th Refined list of methods (minus) methods not for Conc. Model Validation
4th Refined list of methods (minus) methods not for Model Verification
4th Refined list of methods (minus) methods not for Operational Validation
Final list of applicable methods

Figure 1. The flow diagram of the selection process of methods.

TABLE II. LIST OF STATISTICAL TECHNIQUES.

Statistical Techniques	# of datasets	# of variables	Purpose	known parameters	Type of data	Sample size
t-Test	1 or 2	1	Mean equality	Yes	Numerical	Any
Hotteling's T^2 Test	1 or 2	>1	Mean equality	Yes	Numerical	Any
Analysis of Variance	>2	1	Mean equality	Yes	Numerical	Any
Multivariate Analysis of Variance	>2	>1	Mean equality	Yes	Numerical	Any
Simultaneous Confidence Intervals	1 or 2	>1	Mean equality	Yes	Numerical	Any
Factor Analysis	1	>1	Complexity reduction	Yes	Numerical	Any
Principal Component Analysis	1	>1	Complexity reduction	Yes	Numerical	Any
Kolmogorov-Smirnov Test	1 or 2	1	Goodness-of-fit	No	Numerical	Any
Chi-squared Test	1 or 2	1	Goodness-of-fit	No	Numerical & Categorical	Any
Anderson-Darling Test	1 or 2	1	Goodness-of-fit	No	Numerical	Any
Cramér–von Mises Criterion	1 or 2	1	Goodness-of-fit	No	Numerical	Any
Kuiper's Test	1 or 2	1	Goodness-of-fit	No	Numerical	Any
Coefficient of Determination	2	1	Goodness-of-fit	Yes	Numerical	Any
Mann-Whitney-Wilcoxon Test	2	1	Mean equality	No	Numerical & Categorical	Small
White Test	2	1	Heteroscedasticity	Yes	Numerical	Any
Glejser Test	2	1	Heteroscedasticity	Yes	Numerical	Any
Spectral Analysis	2	1	Time Series analysis	Yes	Numerical	Any
Durbin–Watson Statistic	2	1	Time Series analysis	Yes	Numerical	Any

B. *Statistical technique selection methodology*

The list of simulations' and systems' characteristics that influence the selection of techniques, which are explained in more detail in Section II-B2, are expressed in questions, as follows:

1) How many different datasets are going to be examined?
Possible answers: 1, 2, and/or >2.
2) How many different variables are going to be examined?
Possible answers: 1 or >1.
3) What is the purpose of the statistical test?
Possible answers: Mean equality, Complexity reduction, Goodness-of-fit, Heteroscedasticity, or Time Series analysis.
4) Are the sample parameters (μ, σ^2) known?
Possible answers: Yes or No.
5) What kind of data are going to be examined?
Possible answers: Numerical or Categorical.
6) What is the sample size?
Possible answers: Large, Small, or Any.

Table II summarizes the results of the analysis. Similarly to Table I, the intended use of Table II is to act as a filtering mechanism. Whenever an individual or a team wants to verify and/or validate a simulation model, they can utilize this table to narrow down the applicable techniques according to the different characteristics of the simulation at hand and the system. It should be noted that another significant factor in selecting a technique is the statistical power of the technique, i.e., the probability that the null hypothesis (H_0) is correctly rejected for the alternative hypothesis (H_1). The statistical power of a technique is not predetermined, which is the reason it is not included in this analysis. Nevertheless, the Neyman–Pearson lemma [89] is a test that determines which technique is the one with the greatest statistical power given several attributes, like the sample size and the statistical significance.

IV. A CASE STUDY

In this section, a case study illustrates how the framework, through the use of Table I and Table II, can be used. The case study is a computer simulation of a particular instantiation of the Dutch railways. The authors were assigned to validate the simulation model with regards to punctuality, in other words the precision of the delays of trains in the model.

In more detail, the simulation model was built on the Friso simulation package [90]. FRISO is ProRail's, the Dutch infrastructure manager, in-house simulation environment. Being a microscopic simulation environment, FRISO has the potential to - and depending on the model it usually does so- simulate the railway network in a detailed manner; it has the ability to depict the network down to a switch level, which is the case with this model. The model was built in 2014 and it simulates the train operations in one of the most heavily utilized sections (Amsterdam Central station - Utrecht Central station) of one of the largest corridors in the Netherlands (A2), during the whole month of June 2013. The intended use of the model was to examine the punctuality of the timetable with the particular focus being the Amsterdam and Utrecht central stations. A more in depth description of the model, including its input, output, and the final results can be found in [91].

With regards to the methods, the initial list, as it is shown in Table I, consists of 75 methods. Then with every step, the list is narrowed down. For this particular study, the selection process for each property, as shown in Figure 2, was as follows:

1) Access to the source code was not available; *Answer: No*. Using this criterion the available methods were reduced to 42.
2) There were available data from the real system; *Answer: Yes*. Using this criterion eliminated 33 more methods returning a total of 9 available methods. Nonetheless, all 42 methods could have be used in this particular case.
3) The main focus was on the punctuality, ergo functional (hard) requirements, but comments were also expected on the non-functional (soft) requirements; *Answer: Both (but main focus on hard)*. If on the previous criterion *Yes* was chosen as an option, choosing either *Both* or *Hard* on this criterion would leave the list intact (Total 9 methods).
4) The study was mainly concerned with the operational validity of the simulation, but to a degree also with the conceptual model validity; *Answer: C.M. Val & O. Val.*. Using this criterion and based on the selections on the previous criteria, the final number of available methods was reduced to 1 for the conceptual model validation and 7 for the operational validation.

For the operational validation, which was the primary

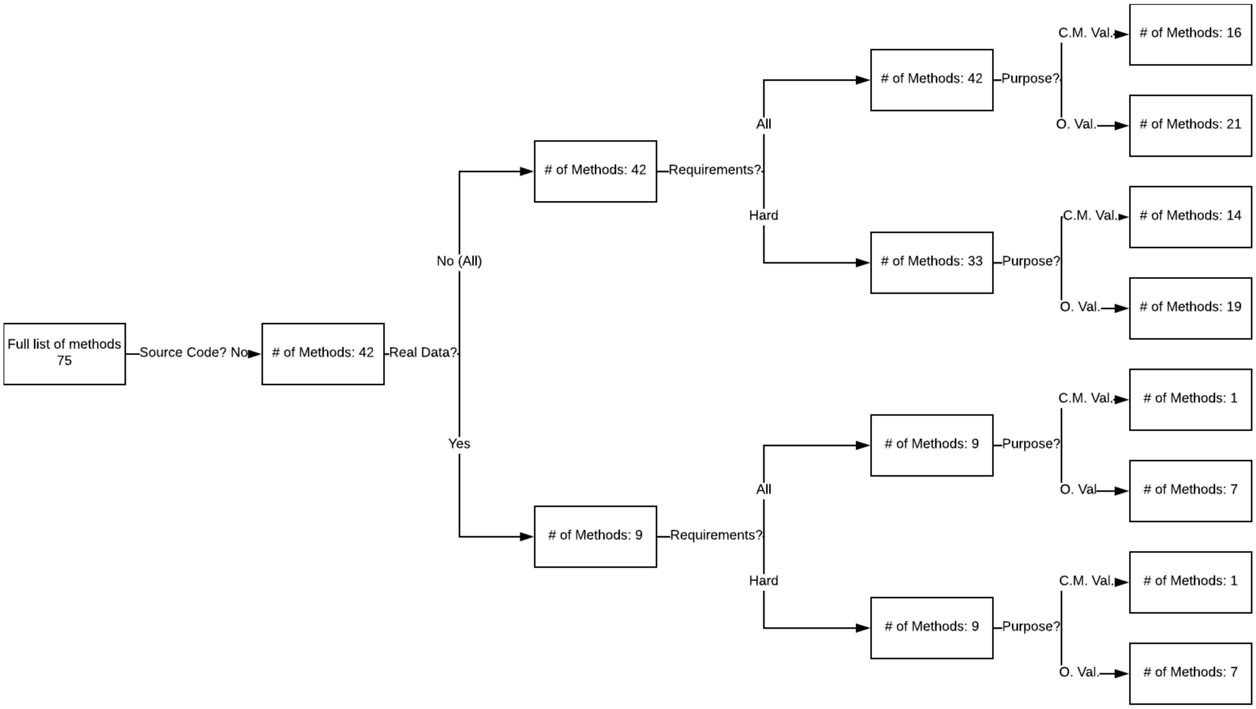

Figure 2. A tree graph of the method selection process.

interest of the study, the final list of the seven methods is shown in Table III. Out of this list, in total four methods were used, namely: the *Face Validation*, *Graphical Comparisons*, *Predictive Validation*, and *Turing Test*. *Predictive Validation* was first used to handle the initial datasets (simulation dataset & operational dataset) and to produce results for the different statistical tests. Then, a combination of the remaining three methods was used to ascertain the validity of the simulation.

With regards to the techniques, the initial list, as shown in Table II, consists of 18 techniques. Then with every step, the list is narrowed down. For this particular study, the selection process for each characteristic was as follows:

1) The model's and reality's output were examined;
Answer: 1 or 2. Using this criteria reduced the available techniques to 16.
2) The amount of delays was the focus;
Answer: 1. Using this criterion eliminated 4 more techniques totaling in 12 available techniques.
3) The study was concerned with whether the delays between the model and reality were similarly distributed and whether the averages were significantly different;
Answer: Mean equality and Goodness-of-fit. Using this criterion resulted in 2 suitable techniques for mean equality and 6 for goodness-of-fit.
4) The sample parameters (μ, σ^2) were known;
Answer: Yes. This is a criterion that only influences the results if the answer is *No*, since the non-parametric techniques can still be used when the mean and variance are known. Therefore the number of techniques remained the same.
5) The delays were in seconds, hence numerical;
Answer: Numerical. Using this criterion eliminated 1 techniques for the mean equality, resulting in 1 technique for mean equality and 6 for goodness-of-fit.
6) Each sample was larger than 100;
Answer: Large. This last criterion did not further reduced the number of techniques, since the only techniques suitable for small datasets (*Mann-Whitney-Wilcoxon Test*) had been eliminated in a previous step.

For testing the equality of means, the only suitable techniques, i.e., t-test, was used. Whereas for testing the goodness-of-fit, from the 6 suitable techniques, the Kolmogorov-Smirnov and chi-squared test were used.

TABLE III. REFINED LIST OF V&V METHODS OF THE CASE STUDY.

Method	**1**	**2**	**3**	**4**
Face Validation	No	Yes	Both	O. Val.
Field Testing	No	Yes	Both	O. Val.
Graphical Comparisons	No	Yes	Both	O. Val.
Predictive Validation	No	Yes	Hard	O. Val.
Real-Time Input Testing	No	Yes	Hard	O. Val.
Turing Test	No	Yes	Both	O. Val.
Visualization/Animation	No	Yes	Both	O. Val.

In this section, the use of the proposed framework demonstrates clearly its effectiveness. As shown in Table III, the initial list of 75 methods was narrowed down in a matter of minutes to the manageable level of seven; and similar reduction occurred in the techniques. By all means, the effectiveness of the framework is not only evident due to its time-saving nature but also due to the fact that it ensures that the chosen methods and techniques are appropriate for the simulation and the system at hand as well as for the purpose of the V&V study.

V. CONCLUSION & FUTURE WORK

In this paper, a framework for simulation validation and verification method and statistical technique selection was proposed. Various properties and characteristics of simulations and systems were taken into account and it was shown that indeed some of these influence the method and technique selection and thus, the final results of the simulation study.

Moreover, the framework was applied on a case study, as a first step towards verifying its effectiveness. The case study showed that the framework is an effective time-saving tool, which also provides a safety net for choosing the methods and techniques that best serve the intended purpose of the simulation and the V&V study.

With regards to future work, additional simulation properties should be identified that may potentially influence the method selection, or some of the discarded properties, identified in Section II-B, might prove to be more influential than initially acknowledged. Moreover, there is a need to further verify the connection of each method to the simulation model's properties and the purpose for which they are more suitable; in other words, it should be verified that the answers on columns 2-6 in Table I are correct. With regards to the techniques, a more extensive list analyzed in the same way as in Section III-B would provide for an improved guide towards selecting the most effective techniques given the problem at hand. Finally, more case studies, from the authors and more importantly from researchers unrelated to the authors, both in pure simulations and in games, would further strengthen the validity and applicability of the framework.

Nevertheless, this paper paves the way for future research in the topic, and as discussed earlier, the main contribution of the framework does not lie in the results presented in Table I and Table II, but is related to the identification of the relationships between the methods, the techniques, the simulation's and system's properties, and the purpose of the V&V study. Therefore, it is of utmost importance that any future research be focused on these relationships.

Acknowledgment

This research is funded through the Railway Gaming Suite 2 program, a joint project by ProRail and Delft University of Technology.

References

[1] B. Roungas, S. Meijer, and A. Verbraeck, "A framework for simulation validation & verification method selection," in SIMUL 2017: The Ninth International Conference on Advances in System Simulation, Athens, Greece, 2017, pp. 35–40.

[2] J. W. Forrester, World dynamics. Wright-Allen Press, 1971.

[3] D. H. Meadows, D. L. Meadows, J. Randers, and W. W. Behrens III, The limits to growth. New York, U.S.A.: Universe Books, 1972.

[4] D. L. Meadows, W. W. Behrens III, D. H. Meadows, R. F. Naill, J. Randers, and E. Zahn, Dynamics of growth in a finite world. Cambridge, Massachusetts: Wright-Allen Press, 1974.

[5] M. Janssen and B. De Vries, "Global modelling: Managing uncertainty, complexity and incomplete information," in Validation of Simulation Models. SISWO, Amsterdam, The Netherlands, 1999, pp. 45–69.

[6] W. D. Nordhaus, "World dynamics: Measurement without data," The Economic Journal, vol. 83, no. 332, 1973, pp. 1156–1183.

[7] O. Balci, "Validation, verification, and testing techniques throughout the life cycle of a simulation study," Annals of Operations Research, vol. 53, no. 1, 1994, pp. 121–173.

[8] S. Schlesinger, R. E. Crosbie, R. E. Gagné, G. S. Innis, C. S. Lalwani, J. Loch, R. J. Sylvester, R. D. Wright, N. Kheir, and D. Bartos, "Terminology for model credibility," Simulation, vol. 32, no. 3, 1979, pp. 103–104.

[9] O. Balci, "Verification, validation, and certification of modeling and simulation applications," in Proceedings of the 35th Conference on Winter Simulation, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Eds. New Orleans, Louisiana, USA: Winter Simulation Conference, 2003, pp. 150–158.

[10] R. G. Sargent, "Verification and validation of simulation models," in Proceedings of the 37th Conference on Winter Simulation, S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Eds. Orlando, Florida, USA: Winter Simulation Conference, 2005, pp. 130–143.

[11] O. Balci, "Verification, validation, and testing," in Handbook of Simulation, J. Banks, Ed. Engineering & Management Press, 1998, ch. 10, pp. 335–393.

[12] W. E. Perry, Effective methods for software testing. Wiley Publishing Inc., 2007.

[13] B. Beizer, Software testing techniques (2nd edition). Van Nostrand Reinhold Company Limited, 1990.

[14] L. A. Miller, E. Groundwater, and S. M. Mirsky, "Survey and assessment of conventional software verification and validation methods," in No. NUREG/CR–6018; EPRI-TR–102106; SAIC–91/6660. Nuclear Regulatory Commission, Washington, DC (United States). Div. of Systems Research; Science Applications International Corp., Reston, VA (United States), 1993.

[15] G. J. Myers, T. Badgett, T. M. Thomas, and C. Sandler, The art of software testing. John Wiley & Sons, Inc., 2011.

[16] R. S. Pressman, Software engineering: A practitioner's approach (8th edition). McGraw-Hill, New York, NY, 2015.

[17] U.S. Department of Defense, "DoD Modeling and Simulation (M&S) Glossary," Tech. Rep., 1997.

[18] C. Rattray, Specification and verification of concurrent systems. Springer-Verlag London, 1990.

[19] O. Balci, "The implementation of four conceptual frameworks for simulation modeling in high-level languages," in Proceedings of the 20th Conference on Winter Simulation, M. A. Abrams, P. L. Haigh, and J. C. Comfort, Eds. San Diego, California, USA: ACM, 1988, pp. 287–295.

[20] R. H. Dunn, "The quest for software reliability," in Handbook of Software Quality Assurance. New York: Van Nostrand Reinhold, 1987.

[21] R. W. Adrion, M. A. Branstad, and J. C. Cherniavsky, "Validation, verification, and testing of computer software," ACM Computing Surveys (CSUR), vol. 14, no. 2, 1982, pp. 159–192.

[22] C. F. Hermann, "Validation problems in games and simulations with special reference to models of international politics," Behavioral Science, vol. 12, no. 3, 1967, pp. 216–231.

[23] R. Shannon and J. D. Johannes, "Systems simulation: The art and science," IEEE Transactions on Systems, Man, and Cybernetics, vol. 6, no. 10, 1976, pp. 723–724.

[24] Systems and software engineering – Vocabulary. IEEE, 2011.

[25] M. S. Martis, "Validation of simulation based models: A theoretical outlook," Electronic Journal of Business Research Methods, vol. 4, no. 1, 2006, pp. 39–46.

[26] I. M. Copi, C. Cohen, and D. E. Flage, Essentials of logic. Taylor & Francis, 2016.

[27] S. R. Schach, Classical and object-oriented software engineering (8th edition). McGraw-Hill, 2011.

[28] H. P. Barendregt, The Lambda Calculus: Its syntax and semantics, 1984.

[29] I. Burnstein, Practical software testing: A process-oriented approach. Springer-Verlag New York, 2006.

[30] E. W. Dijkstra, "Guarded commands, non-determinacy and a calculus for the derivation of programs," in Language Hierarchies and Interfaces. Lecture Notes in Computer Science, vol 46, F. Bauer et al., Ed. Springer, Berlin, Heidelberg, 1976, pp. 111–124.

[31] J. R. Emshoff and R. L. Sisson, Design and use of computer simulation models. MacMillan, New York, 1970.

[32] R. C. Backhouse, Program construction and verification. Prentice-Hall International, 1986.

[33] R. B. Whitner and O. Balci, "Guidelines for selecting and using simulation model verification techniques," in Proceedings of the 21st Conference on Winter Simulation, E. MacNair, K. Musselman, and P. Heidelberger, Eds. Washington, D.C., USA: ACM, 1989, pp. 559–568.

[34] J. C. King, “Symbolic execution and program testing,” Communications of the ACM, vol. 19, no. 7, 1976, pp. 385–394.

[35] I. Sommerville, Software engineering (9th edition). Addison-Wesley, Reading, MA, 2004.

[36] L. W. Schruben, “Establishing the credibility of simulations,” Simulation, vol. 34, no. 3, 1980, pp. 101–105.

[37] D. L. Hahs-Vaughn and R. G. Lomax, An introduction to statistical concepts. Routledge, 2013.

[38] B. L. Welch, “The generalization of ‘Student’s’ problem when several different population variances are involved,” Biometrika, vol. 34, no. 1/2, 1947, p. 28.

[39] H. Hotelling, “The generalization of Student’s ratio,” in American Mathematical Society, Berkeley, CA, USA, 1931.

[40] T. H. Naylor and J. M. Finger, “Verification of computer simulation models,” Management Science, vol. 14, no. 2, 1967, pp. B–92–B–101.

[41] R. G. Lomax and D. L. Hahs-Vaughn, Statistical concepts: A second course. Taylor & Francis Group, 2013.

[42] R. T. Warne, “A primer on multivariate analysis of variance (MANOVA) for behavioral scientists,” Practical Assessment, Research & Evaluation, vol. 19, no. 17, 2014, pp. 1–10.

[43] O. Balci and R. G. Sargent, “Validation of simulation models via simultaneous confidence intervals,” American Journal of Mathematical and Management Sciences, vol. 4, no. 3-4, 1984, pp. 375–406.

[44] I. T. Jolliffe, Principal Component Analysis and Factor Analysis. Springer, 1986.

[45] I. M. Chakravarty, J. D. Roy, and R. G. Laha, Handbook of methods of applied statistics. John Wiley & Sons, 1967.

[46] R. A. Fisher, “The conditions under which χ^2 measures the discrepancey between observation and hypothesis,” Journal of the Royal Statistical Society, vol. 87, no. 3, 1924, pp. 442–450.

[47] T. W. Anderson and D. A. Darling, “Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes,” The Annals of Mathematical Statistics, vol. 23, no. 2, 1952, pp. 193–212.

[48] D. A. Darling, “The Kolmogorov-Smirnov, Cramer-von Mises tests,” The Annals of Mathematical Statistics, vol. 28, no. 4, 1957, pp. 823–838.

[49] H. Cramér, “On the composition of elementary errors,” Scandinavian Actuarial Journal, vol. 1928, no. 1, 1928, pp. 13–74.

[50] N. H. Kuiper, “Tests concerning random points on a circle,” Indagationes Mathematicae, vol. 63, 1960, pp. 38–47.

[51] H. Theil, Economic forecasts and policy. Amsterdam. The Netherlands: North-Holland.

[52] H. B. Mann and D. R. Whitney, “On a test of whether one of two random variables is stochastically larger than the other,” The Annals of Mathematical Statistics, vol. 18, no. 1, 1947, pp. 50–60.

[53] H. White, “A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity,” Econometrica, vol. 48, no. 4, 1980, pp. 817–838.

[54] H. Glejser, “A new test for heteroskedasticity,” Journal of the American Statistical Association, vol. 64, no. 325, 1969, p. 316.

[55] J. A. Fitzsimmons, “The use of spectral analysis to validate planning models,” Socio-Economic Planning Sciences, vol. 8, no. 3, 1974, pp. 123–128.

[56] G. M. Jenkins and D. G. Watts, Spectral analysis and its applications. Holden-Day, San Fransisco, 1969.

[57] J. Durbin and G. S. Watson, “Testing for serial correlation in least squares regression: I,” Biometrika, vol. 37, no. 3/4, 1950, p. 409.

[58] W. F. van Gunsteren and A. E. Mark, “Validation of molecular dynamics simulation,” The Journal of Chemical Physics, vol. 108, no. 15, 1998, pp. 6109–6116.

[59] J. P. Kleijnen, “Verification and validation of simulation models,” European Journal of Operational Research, vol. 82, no. 1, 1995, pp. 145–162.

[60] H. Vangheluwe, J. de Lara, and P. J. Mosterman, “An introduction to multi-paradigm modelling and simulation,” in Proceedings of the AIS’2002 Conference (AI, Simulation and Planning in High Autonomy Systems), F. Barros and N. Giambiasi, Eds., Lisboa, Portugal, 2002, pp. 9–20.

[61] J. H. Byun, C. B. Choi, and T. G. Kim, “Verification of the DEVS model implementation using aspect embedded DEVS,” in Proceedings of the 2009 Spring Simulation Multiconference. San Diego, CA, USA: Society for Computer Simulation International, 2009, p. 151.

[62] H. Saadawi and G. Wainer, “Verification of real-time DEVS models,” in Proceedings of the 2009 Spring Simulation Multiconference. San Diego, CA, USA: Society for Computer Simulation International, 2009, p. 143.

[63] M. D. Di Benedetto, S. Di Gennaro, and A. D’Innocenzo, “Diagnosability verification for hybrid automata,” in Hybrid Systems: Computation and Control, A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. Springer, Berlin, Heidelberg, 2007, pp. 684–687.

[64] J. Jo, S. Yoon, J. Yoo, H. Y. Lee, and W.-T. Kim, “Case study: Verification of ECML model using SpaceEx,” in Korea-Japan Joint Workshop on ICT, 2012, pp. 1–4.

[65] Y. Barlas, “Model validation in system dynamics,” in Proceedings of the 1994 International System Dynamics Conference, Sterling, Scotland, 1994, pp. 1–10.

[66] J. W. Forrester and P. M. Senge, “Tests for building confidence in system dynamics models,” in System Dynamics, TIMS Studies in Management Sciences, 14, 1980, pp. 209–228.

[67] M. C. Overstreet and R. E. Nance, Characterizations and relationships of world views, R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, Eds. Washington Hilton and Towers, Washington, D.C., U.S.A.: ACM, 2004.

[68] D. Liu, N. D. Macchiarella, and D. A. Vincenzi, “Simulation fidelity,” in Human Factors in Simulation and Training, 2008.

[69] J. E. Morrison and L. L. Meliza, “Foundations of the after action review process,” Alexandria, VA, p. 82, 1999.

[70] R. Fanning and D. Gaba, “The role of debriefing in simulation-based learning,” Simulation in Healthcare, vol. 2, no. 2, 2007, pp. 115–125.

[71] J. van den Hoogen, J. Lo, and S. Meijer, “Debriefing in gaming simulation for research: Opening the black box of the non-trivial machine to assess validity and reliability,” in Proceedings of the 2014 Winter Simulation Conference, A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, Eds. Savannah, Georgia, USA: IEEE Press, 2014, pp. 3505–3516.

[72] J. Lo, J. van den Hoogen, and S. Meijer, “Using gaming simulation experiments to test railway innovations: Implications for validity,” in Proceedings of the 2013 Winter Simulation Conference, R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, Eds. Washington, D.C., USA: IEEE Press, 2013, pp. 1766–1777.

[73] B. Zevin, J. S. Levy, R. M. Satava, and T. P. Grantcharov, “A Consensus-based framework for design, validation, and implementation of simulation-based training curricula in surgery,” Journal of the American College of Surgeons, vol. 215, no. 4, 2012, pp. 580–586.e3.

[74] P. J. Morgan, D. Cleave-Hogg, S. DeSousa, and J. Tarshis, “High-fidelity patient simulation: Validation of performance checklists,” British Journal of Anaesthesia, vol. 92, no. 3, 2004, pp. 388–392.

[75] S. I. Gass, “Decision-aiding models: Validation, assessment, and related issues for policy analysis,” Operations Research, vol. 31, no. 4, 1983, pp. 603–631.

[76] R. R. Nemani and S. W. Running, “Testing a theoretical climate-soil-leaf area hydrologic equilibrium of forests using satellite data and ecosystem simulation,” Agricultural and Forest Meteorology, vol. 44, no. 3-4, 1989, pp. 245–260.

[77] A. Mavin and N. Maiden, “Determining socio-technical systems requirements: Experiences with generating and walking through scenarios,” in Proceedings of the 11th IEEE International Conference on Requirements Engineering. IEEE Comput. Soc, 2003, pp. 213–222.

[78] F. Nilsson and V. Darley, “On complex adaptive systems and agent-based modelling for improving decision-making in manufacturing and logistics settings,” International Journal of Operations & Production Management, vol. 26, no. 12, 2006, pp. 1351–1373.

[79] M. A. Louie and K. M. Carley, “Balancing the criticisms: Validating multi-agent models of social systems,” Simulation Modelling Practice and Theory, vol. 16, no. 2, 2008, pp. 242–256.

[80] F. Landriscina, Simulation and learning. Springer, 2013.

[81] J. Mylopoulos, L. Chung, and E. Yu, “From object-oriented to goal-

oriented requirements analysis," Communications of the ACM, vol. 42, no. 1, 1999, pp. 31–37.

[82] O. Balci, "Quality assessment, verification, and validation of modeling and simulation applications," in Proceedings of the 36th Conference on Winter simulation, R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, Eds. Washington, D.C., USA: Association for Computing Machinery, 2004, pp. 122–129.

[83] D. K. Pace, "Modeling and simulation verification and validation challenges," Johns Hopkins APL Technical Digest, vol. 25, no. 2, 2004, pp. 163–172.

[84] W. Feller, "The fundamental limit theorems in probability," in Selected Papers I. Springer International Publishing, 2015, pp. 667–699.

[85] R. G. Sargent, "Verification, validation, and accreditation: Verification, validation, and accreditation of simulation models," in Proceedings of the 32nd Conference on Winter Simulation, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, Eds. Orlando, Florida, USA: Society for Computer Simulation International, 2000, pp. 50–59.

[86] S. Meijer, "Gaming simulations for railways: Lessons learned from modeling six games for the Dutch infrastructure management," in Infrastructure Design, Signalling and Security in Railway, X. Perpinya, Ed. InTech, 2012, ch. 11, pp. 275–294.

[87] G. van Lankveld, E. Sehic, J. C. Lo, and S. A. Meijer, "Assessing gaming simulation validity for training traffic controllers," Simulation & Gaming, vol. 48, no. 2, 2017, pp. 219–235.

[88] S. A. Meijer, "The power of sponges – high-tech versus low-tech gaming simulation for the Dutch railways," in CESUN 2012: 3rd International Engineering Systems Symposium, Delft, The Netherlands, 2012, pp. 18–20.

[89] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 231, no. 694–706, 1933, pp. 289–337.

[90] D. A. Middelkoop and L. Loeve, "Simulation of traffic management with FRISO," WIT Transactions on the Built Environment, vol. 88, 2006.

[91] B. Roungas, S. Meijer, and A. Verbraeck, "Validity of railway microscopic simulations under the microscope: Two case studies," International Journal of System of Systems Engineering, p. in press.