IE 477 – IE 478 Systems Design

Suggested Steps to Perform Verification and Validation

Last Revised on September 7, 2026

In a typical industrial engineering project, after defining the problem and determining assumptions, constraints, and objectives, one first develops a high-level conceptual model that will show inputs, outputs and all major components and their interactions. Some (or all depending on the boundaries of the problem defined) of these major components and their interactions are expected to require mathematical models to describe the complexity, as well as to achieve better (or optimal) solutions using the objectives determined. These mathematical models are described and then analytically realized (mostly represented parametrically) when converted into a computer code of various kind (such as Python, C, Java, VBA, MATLAB, R code) or via a special-purpose software (such as CPLEX, Mosel, Gams, SIMAN, Arena, Microsoft Excel, and many others). Given any input data, results are obtained as the proposed solution method (or as a part of the solution approach). Before these results are used, one should check credibility of the method, as well as its generality for practically any input data. There are various names given to this credibility check in the literature of different fields, all serving the same purpose - see Balci (2016), Hillston (2003 and 2017), and Landry et al (1983)[1] as examples for the ones this document shares a lot of commonalities in approach.

In this document, we are going to use two separate words to describe the steps for this check: *verification* and *validation*. We consider *conceptual validation* in parallel with the construction of the conceptual model. The schematic model in Figure 1 explains roughly what we mean by those two words. We further describe details and give examples. **Please note that the examples are not meant to cover all design projects – the needs of your project might require different ways that prescribed in the examples stated in this document.**

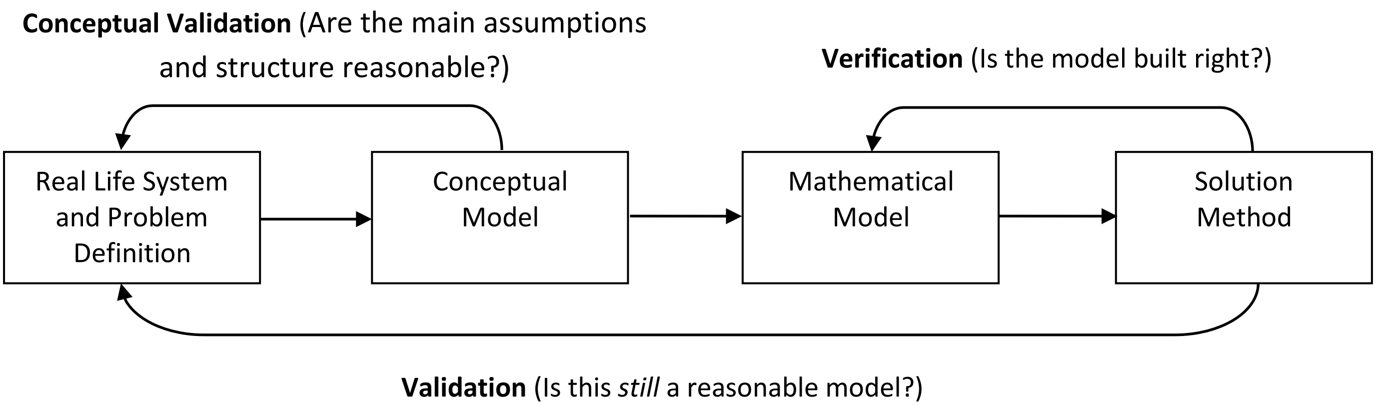

Figure 1: Schematic representation of verification and validation steps

[1] In Landry et al (1983) the term verification is not explicitly mentioned but takes its place within several categories of validity specified. Regardless of the definitions stated, the union of verification and validation mentioned in this note is identical to the union of several categories of validation mentioned in the reference.

Verification

When a mathematical model is built, one needs to check whether the model works as designed. This stage is sometimes straightforward for an experienced analyst. However, there could be problems rooted at two different sources:

- The mathematical model may be incomplete or erroneous as it does not cover all possible scenarios that may happen in real life.
- The translation of your mathematical model to a code/software may be incomplete or erroneous as you obtain unexpected results.

Verification asks, "Is the model working as intended or is it built right?" and is expected to demonstrate the model's potential usefulness. At this stage, you **do not need real data**, as fabricated data (not necessarily randomly generated) will suffice for the analyses. Actually, in most cases, when only real data is used for verification, the verification tests will not encompass all the numerical possibilities that can occur in real life. As a result, you are asked to **fabricate data sets for detailed verification purposes**.

Verification is like debugging a program; make sure that there are no mistakes in the way it is written. Basic tactics can be informally listed as: tracing the model flow; checking the outputs, and making sure that they make sense; dumping arbitrary data (with extremes) to check reaction. There are numerous ways to verify a model under different types of models. See Hillston (2003) for more formal descriptions of the possible generalized tests stated as antibugging, structured walk through/one step analysis, analysis of simplified (reduced) models, continuity testing, degeneracy testing, consistency testing and other simulation specific methods.

Verification Examples

Here are some examples – note that these are only examples and are not exhaustive.

Mathematical Programming Models

These include linear and mixed integer programming models, dynamic programming models, multiple criteria problems, etc. One can implement some of the below-mentioned methods:

- **Analysis of simplified (reduced) models:** For example, take some of the constraints out. Remaining problem can be a simple (trivial) problem that can be solved via other means. Make sure that you obtain the same solution from your model.
- **Continuity testing:** For example, change the RHS values consistently, and check whether the optimal objective function changes as expected.
- **Degeneracy testing:** For example, increase, or decrease some objective function coefficients indefinitely. The result you obtain should be consistent with what you expect.
- **Consistency testing:** For example, increase demand indefinitely. Make sure that you obtain the result you expect.

Stochastic Models

One can implement some of the below-mentioned methods:

- **Analysis of simplified (reduced) models:** For example, eliminate, or reduce randomness and check whether you obtain the result you expect.
- **Continuity testing:** For example, change a parameter (such as customer arrival rates, number of servers, service rates) consistently and check whether this test is satisfied.

- **Degeneracy testing:** For example, increase or decrease a parameter indefinitely. The result you obtain should be consistent with what you expect.
- **Consistency testing:** For example, increase or decrease demand input indefinitely. Make sure that you obtain the result you expect.

Forecasting Models

One can implement some of the below-mentioned methods:

- **Continuity testing:** For example, change one of the inputs consistently and the result you obtain should be consistent with what you expect.
- **Degeneracy testing:** For example, increase or decrease a parameter indefinitely. The result you obtain should be consistent with what you expect.
- **Consistency testing:** For example, increase or decrease parameters of the obtained model indefinitely. Make sure that you obtain the result you expect.

Simulation Models

Any of the methods and more (see Sargent, 2020, specific for simulation models) can be devised to make sure that the simulation model prepared is working as intended. A general rule of thumb is to run the simulation model including different scenarios considering the above-mentioned criteria and check if the model provides a reasonable output. Do not forget to make independent replications or use batch means method that will provide the output measure of interest with a confidence interval. Recall that the length of the confidence interval can be reduced via more replications (for independent replications) or a longer simulation run (for batch means).

Heuristic Algorithms

One can implement some of the below-mentioned methods:

- **Analysis of simplified (reduced) models:** Special (easy, trivial) cases may be solved analytically, and we expect the heuristic algorithm to give same or similar results (or on the average similar results if the algorithm utilizes a randomized search process)
- **Degeneracy Testing**: For example, increase or decrease some input indefinitely. The result you obtain should be consistent with what you expect.
- **Consistency testing:** For example, increase or decrease parameters of the obtained model indefinitely. Make sure that you obtain the result you expect.

For other models, one can devise methods (deduct from the general idea) to do the verification.

Validation

For a model to be acceptable, one should show that, under a given set of conditions, the model results are identical (or sufficiently close) to those obtained by the real system under the same conditions. Of course, this is ideal, and sometimes, it would be difficult to attain this level for comparison.

As described in the introduction of this note, it may not be easy to separate verification and validation. We approach this in a simpler way: Verification is the step that ensures the model acts as intended, whereas validation assesses the model's credibility. (By credibility, we imply that a decision maker (or makers) who are going to use the output either directly as decisions, or use it as input information for further decision making, will utilize the output without questioning the underlying mechanism. Of course, the input data and other features of the model can always be questioned, but the logic behind the framework is expected not to be questioned for a credible model.) We suggest that one should first verify the solution method's output, then proceed to validate.

Validation asks the question “Is this a reasonable model?” and checks the model's credibility, namely, whether it contains key elements for addressing the primary problems. Note that this step starts with conceptual modeling (see Figure 1). The assumptions made about the boundaries, as well as the structure of the approach considered, constitute the initial step of validation: conceptual validation. Usually, in IE education, this type of validation is included in model formation.

Before we go into the details of validation, one needs to realize that the validation step may be specific to the project undertaken (as compared to the verification step, which might utilize some generic techniques). A model (or an approach) is developed to analyze (and solve) a particular problem, and hence the model's abstraction level depends on how the problem is defined. From this perspective, validation becomes the task of demonstrating that the approach under consideration is a reasonable representation of what is done (or what needs to be done) in the actual system.

There are mainly three aspects to validate:

1. Model boundaries, including constraints of the approach considered (sometimes this aspect is called checking the assumptions made) - Needed at an early stage of the development as a part of conceptual validation (see the introductory parts of this document).
2. Inputs to the approach considered (sometimes this aspect is called calibration of the approach).
3. Outputs of the approach considered (sometimes this aspect is called the conclusive part, as we check some measured metrics of the model and make sure that these metrics are similar or close to the values observed in the real system).

Full validation is usually very difficult and tedious to achieve in practice. Hence, depending on the system under consideration, analysts identify the most important aspect and validate the system through that aspect. Nevertheless, the other aspects are expected to be discussed in some detail, and a reasonable consensus on their validity should be reached.

Four broad approaches can be utilized to check the validity with respect to each of the aspects considered above:

- **Face validity or expert opinion:** Validity confirmation by using the understanding of the decision makers involved in the situation modeled.
- **Operational validity on real system measurements:** Validity confirmation by running an actual pilot study in the system.
- **Operational validity on theoretical analysis or data:** Validity confirmation by comparing results of the model with the historical results observed by the system under controlled experiments.
- Any combination of the above and/or other possibilities specific to the model/environment considered.

Effects of Conceptual Model Construction: Examples for Validation

Almost all experts say that validation should start at the modeling stage to challenge the conceptualization. In other words, one should develop a reasonable model to start with (conceptual validation requires the project group to communicate with the AA and IA). Hence, the validation process starts as soon as analysts begin modeling the system. So, we first give those examples where one should be concerned with some checks at the model formation step. This is one of the reasons why you should share your thought process within the group, as well as with your AA and IA.

- **Face validity:** For example, is it reasonable to assume stationarity over time? What are the consequences? Is it clear what to do (how to manage) if the assumptions do not hold? How can such a situation be detected? We do not need detailed responses to each question; simply acknowledge that practical solutions are needed. As a summary, the expert is needed to respond to "What if" and "what are the consequences" type questions without numerical exercises.
- **Face validity:** Is it reasonable not to have some of the parts of the real system considered in the approach (or the model)? What if those parts were consistently left out of the current system? Each such consideration needs to be checked independently at the model formation stage. As a summary, the expert is needed to respond to "What if" and "what are the consequences" type questions without numerical exercises.
- **Operational validity:** For example, is it possible to obtain inputs as desired by the model at the desired time? A model is prepared that requires certain data and information (up-to-date and updated) at defined intervals. If the information and data updating cycles do not coincide in time, this weakens the model, making it less useful.
- **Operational validity:** For example, how frequently should the model run? Is it possible to achieve it in practice? Is it meaningful to have the requested frequency?

These examples are simple but capture the essence when setting up the modeling approach to be used for the remainder of the study.

Validation Examples for the Almost-complete Frameworks or Models

Validation should be devised regardless of the modeling approach. So here we do not list possible models; instead, we assume any model and present possible approaches as examples. These are a very limited set of examples, far from being an exhaustive list:

- **Face validity:** As an example, what is the overall effect of including (or not including) certain aspects in the solution approach?
- **Operational validity:** For example, is it possible to set a pilot study where one can implement the proposed methodology? If so, what performance indicator levels would one expect, and can we measure them in the pilot study for comparison? The values obtained by the model and by measuring them in the pilot study (examples: actual length of the trip versus the one stated by the model; actual cost of overtime versus the value stated by the model; total cost measured by the pilot versus as stated in the model) being compared are not expected to be exactly the same, but they are expected to be within a desired accuracy level (and relative error is important).
- **Theoretical analysis:** For example, estimate the input parameters (and input distributions) needed to operationalize the approach. Check whether the input estimates yield the same objective function (or performance measures) in the model as in the actual under current operational decisions. One may iterate a few times to obtain a relevant set of inputs where the objective values are close to each other – this process is called parameter calibration, and as a result, one can assess data validity. As an example, consider a simple routing problem in which one set of input parameters is the distances among nodes. There is a current route, and the distance (or time) to cover the route has been

measured in practice (several times). The model with the same route selected should yield a similar or close value for the distance traveled or the time it took.

- **Theoretical analysis:** For example, once data validity is reached, various scenarios that have been observed in the past can be tried to obtain operational validity by comparing the objective function value (or, in general, performance measure values) of the model with the realized. Note that you can 6 also do the same analysis for those constraints which are crucial and make sure that the amount used from this resource is similar (close to) in theoretical analysis with the actual value observed.
- **Face validity – expert opinion:** Systematic checks to be controlled by a panel of experts can be used to validate the overall approach. Of course, planning these systematically may be challenging. See Taghikhah et al. (2021) and Vainola (2021) as interesting and recent examples in which the validation process is not straightforward. These examples constitute very strong indications that validation is ultimately case-specific.

Validation Examples for Different Solution Methodologies with Further Comments Related to Benchmarking

- **Optimization Model:** To validate an optimization model, the following controlled approach may give satisfactory results:
  1. Keep track of the firm's historical decisions and the performance indicators for those decisions.
  2. In the proposed mathematical model, show that the decisions made by the firm (for each instance obtained) are within the feasible solution set of the model (no need to optimize).
  3. Once you show that each incidence corresponds to a feasible solution, simply compute the performance measure values (performance measures can be part of the objective function, the objective function or any other function that can be computed as a function of the decision variables).
  4. Ask the IA to compare the actual values of the performance measures with the ones calculated in part c). If the comparison yields reasonable results, you have validated your model.
  5. Note that no optimization has been used up to this point. The task is simply to show that the model can produce the current decision within its feasible set and that the measurements (made by the model) are acceptable. In benchmarking, one optimizes the model and compares it with the current solution (decision variables, objective function or any other performance measures). Some sensitivity for benchmarking is needed.
- **Heuristic Model:** To validate a heuristic model, the controlled approach considered in the above methodology can be utilized. Here are the parts that will differ:

  In part 2., the feasible solution concept may be harder to consider in a heuristic. Nevertheless, in most cases, you can obtain the solution directly and proceed with parts 3. and 4. Benchmarking is more challenging, as you may need to use a sufficiently large test bed (in terms of sensitivity) to show that the proposed heuristic is effective.
- **Predictive Model:** To validate a predictive model (like a forecasting model) is challenging. As a result, a pilot study may be required in such situations. A good approach would be to keep the current and proposed systems in parallel for some time. Comparisons of predictions across different bases should be done by experts. In those sessions, it might be enlightening to specify the causes of these differences and provide explanations. Note that it would be even better if one could obtain these comparisons with past data (with information on the predictions that came out of the current system,

and corrections made subjectively, if any) in which the realized outcome is also observed. Note that face validity of the general approach under consideration is a very important tool in validating predictive models.

- **First-time designs and one-time plans:** In these cases, there is no past or current system to be compared. Hence, carefully designing face validity for each subsystem and the system as a whole is necessary. For these models, validation work is expected to start very early, almost simultaneously with the conceptual models being designed. Use of simulation animation to facilitate the expert opinion may be utilized. Furthermore, these simulation models can be used to compare a small number of alternatives

Conclusions

Verification and validation are two important steps in using conceptual models in real-world applications. Conceptual validation starts together with the initial thoughts of the conceptual model – you are expected to be familiar with this type of validation, as it constitutes the principles in IE education. Verification does not require actual data, but the approach type determines which methods to use. Validation, on the other hand, may require actual data (to some extent), and the type of modeling framework is not important. Hence, one can treat the methodology as a black box, focusing on the inputs and outputs only – validation is achieved using these inputs and outputs and their consistency with the current system.

Note that the verification step is expected to involve the industrial advisor (or decision-makers), as the model's possible merits (as noted above, we used the word "usefulness") will become apparent even at the verification stage. Similarly, a requirement for starting and completing validation is ensuring that the actual decision-makers (including, in our case, at least the project's industrial advisor) are satisfied and ready to use the model.

References

Balci, O., (Revised version), "Verification, Validation and Testing of Models," In *Encyclopedia of Operations Research and Management Science*, S.I. Gass and C.M. Harris, Eds., Revised version published in the Encyclopedia's Third Edition, 2011, by Springer Science + Business Media

Hillston, J., "Model Validation and Verification", September 19, 2003 (Access no longer available – see your course WEB site for a copy)

Hillston, J., "Performance Modelling — Lecture 16: Model Validation and Verification" School of Informatics, The University of Edinburgh, Scotland, Presentation Slides, 16th March 2017

Landry, M., Malouin, J. L., & Oral, M., "Model validation in operations research", *European Journal of Operational Research*, *14*(3), 207-220, 1983.

Sargent, R. G., "Verification and validation of simulation models: an advanced tutorial", In *Proceedings of the Winter Simulation Conference* (pp. 16-29), 2020.

Taghikhah, F., Voinov, A., Shukla, N., Filatova, T., & Anufriev, M., "Integrated modeling of extended agro-food supply chains: A systems approach", *European Journal of Operational Research*, *288*(3), 852-868, 2021.

Vainola, V., "Sales Plan Framework for Operational and Performance Qualification Service", Master's Thesis, Helsinki Metropolia University of Applied Sciences, 2021