IE 477 – IE 478 Production Systems Design

Suggested Steps to Perform Verification and Validation

Last Revised on November 17, 2021

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.**

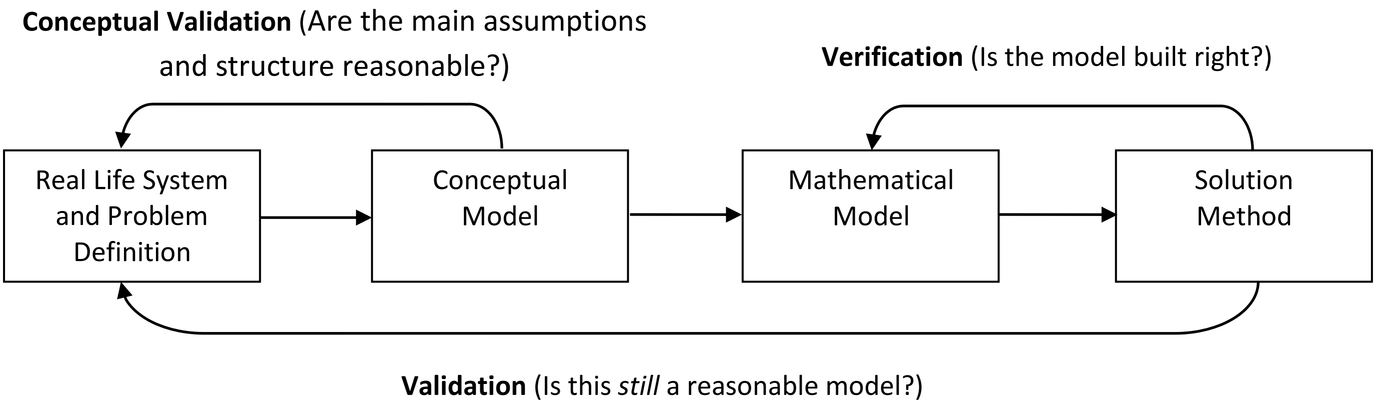

Figure 1: Schematic representation of verification and validation steps

[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 is asking the question "Is the model working as intended or is it built right?" and is expected to show the possible <u>usefulness</u> of the model. At this stage you **do not need real data**, as fabricated (not necessarily randomly generated) data will suffice for the analyses to be performed.

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

In order a model to be acceptable, one should show that under a given set of conditions, model results are identical (similar, close enough) to the results obtained by the real system under the same set of 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 is not easy to separate verification and validation. We approach in a more simplistic way: Verification is defined as the step to make sure that the model acts as intended, whereas validation shows the credibility of the model. We suggest that one should firstly verify the output of the solution method, then proceed to perform validation.

Validation asks the question "Is this *still* a reasonable model?" and will check the credibility of the model; namely whether the model contains key elements in addressing the primary problems. Note that this step starts with conceptual modeling (see Figure 1). The assumptions made on the boundaries as well as on 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 to the details of validation, one needs to realize that validation step is very specific for the project undertaken (as compared to 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 abstraction level of the model depends on the way that problem is defined. Looking with this perspective, validation becomes the task of demonstrating that the approach considered is a reasonable representation of what is done (or what is needed 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 as checking the assumptions made).
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 as the conclusive part, as we check some measured metrics of the model and make sure that these metrics are of similar or close to the values observed in the real system).

Full validation is usually very difficult and tedious level to achieve for practical purposes. Hence, depending on the system considered, analysts consider the most important aspect and validate the system through that aspect. Nevertheless, the other aspects are expected to be discussed in a certain level of detail and reasonable consensus on the validity of those aspects should be confirmed.

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.

Conceptual Validation Examples

Almost all experts say that conceptual validation should start at the modeling stage. In other words, one should come up with a model which is reasonable to start with. Hence, validation process starts as soon as analysts start to model the system. So, we first give those examples where one should be concerned with some checks at the model formation step.

- **Face validity:** For example, is it reasonable to assume stationarity over time? What are the consequences?
- **Operational validity:** For example, is it possible to obtain inputs as desired by the model at the time desired?
- **Operational validity:** For example, how frequently should the model run? Is it possible to achieve it in practice?

These examples are simple, but give the essence at the time of setting up the modeling approach to be utilized for the remaining part 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, but consider that we have any model and state possible approaches as examples. These are a very limited set of examples, far from being an exhaustive list:

- **Face validity:** For example, is it reasonable not to have some of the parts of the real system considered in the approach (or the model)? Are these parts left out consistently? Note that, at this stage one needs to check the overall effect of the approach rather than checking each part independently (actually, each were checked independently at the model formation stage).
- **Operational validity:** For example, is it possible to set a pilot study where one can implement the proposed methodology? If so, what are the possible levels of performance indicators one would expect, and can we measure those from the pilot study for comparison?
- **Theoretical analysis:** For example, estimate the input parameters (and input distributions) needed to operationalize the approach. Check whether the input estimates are yielding the same value of objective function (or performance measures) in the model as well as the actual under current operational decisions. One may iterate a few times to come up with a relevant set of inputs where the objective values are close to each other – this process is called *calibration* of parameters and as a result one can obtain *data validity*. As an example, consider a simple routing problem where one set of input parameters are 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 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

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 not be straightforward and may be challenging. See Taghikhah et al (2021) and Vainola (2021) as interesting and recent examples when the validation process is not straightforward. These examples constitute very strong indications that validation is ultimately case specific.

Conclusions

Verification and validation are two important steps in making use of conceptual models in real life. 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 primers in IE education. Verification does not need actual data, but the type of approach affects the methods to be used. Validation, on the other hand, needs actual data (up to a certain extend), and the type of the 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 possible merits of the model (above we used the word "usefulness") will become apparent even at the verification stage. Similarly, one requirement of starting and completing validation is to make sure that the actual decision-makers (including at least the industrial advisor of the project in our case) 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