PARAMETER VALIDATION USING CONSTRAINT OPTIMIZATION FOR MODELING AND SIMULATION

Guodong Shao, Charles McLean
Manufacturing Simulation and Modeling Group
National Institute of Standards and Technology (NIST)
100 Bureau Drive, MS 8260
Gaithersburg, Maryland 20899-8260, U.S.A.
{gshao, mclean}@nist.gov

Alexander Brodsky, Paul Ammann
George Mason University
Fairfax, VA 22030, U.S.A.
{Brodsky, Pammann}@gmu.edu

KEYWORDS
Modeling and Simulation (M&S), Verification and Validation (V&V), Model Parameter, Optimization.

ABSTRACT

Modeling and simulation (M&S) techniques are increasingly being used to solve problems and aid decision making in many different fields. Results of simulations are expected to provide reliable information for decision makers. But potential errors may be introduced during the M&S development lifecycle. It is critical to ensure to build the right model and the model is built right. M&S community has had intensive Verification and Validation (V&V) research. But V&V activities are often not formally performed in most of the cases. For those who perform V&V activities, they normally wait until development of the simulation modeling is finished. Practical and solid validation techniques are hence needed. In this paper, the authors propose a validation methodology that allows parallel simulation development and model parameter validation, i.e. first the simulation model can be built with unknown parameters included; and then, those parameters can be estimated using a built-in constraint optimizer. Finally the initially unknown parameters are replaced with the found optimal values. The model is then ready for future output prediction. As an example application, a simple supply chain cost simulation model was discussed using the proposed methodology.

INTRODUCTION

In order to perform the study of the real world problem scientifically, we often have to make a set of assumptions about how it works. These assumptions, which usually take the form of mathematical or logical relationships, constitute a model that is used to gain some understanding of how the corresponding system behaves. If the relationships are simple enough, one may just use an analytic solution that is a mathematical function to express it and obtain exact information on questions of interests. Unfortunately, most real-world problems we are trying to solve are too complex for an exact mathematic function to represent and there may be many parameters that are unknown. They must be studied by means of simulation. That's why simulation is regarded as second only to "math programming" among 13 operations-research techniques (Law and Kelton 2000).

M&S is the process of constructing a model of a system that contains a problem and conducting experiments with the model for a specific purpose of solving the problem and aiding in decision-making. M&S is particularly valuable for Department of Homeland Security (DHS) applications and manufacturing applications, because it can provide a non-destructive and non-invasive method of observing a system and also provide a way to test multiple inputs and evaluate various outputs (Jain and McLean 2006). For example, in DHS applications, simulations allow users to reconstruct a comprehensive representation of real-world features during disaster response. Simulation models can help the decision makers determine staff and resource levels in hypothetical terrorist attack scenarios (Shao and McLean 2008) (Shao and Lee 2007). But for the developers and users of the simulation models, the decision makers using the results of these models, and individuals affected by decisions based on such models are all concerned with whether a model and the simulation results are correct (Sargent 2007). Even though M&S community has had intensive V&V research (DOD 2001), V&V activities are often not formally performed in most of the cases. Validation efforts have often been limited to the use of less rigorous techniques, such as face validation and traceability assessment (Sargent, et al., 2000). Practical and solid validation techniques are hence needed to make sure the simulation model is validated and the simulation results are credible.

The contribution of this paper is to propose a novel validation methodology that allows parallel simulation development and model parameter validation. The technique integrates constraint optimizer that performs the parameter validation for M&S. First the simulation model can be built with unknown parameters included; and then, those parameters can be estimated using a built-in constraint optimizer. Finally the initially unknown parameters are replaced with the optimal values. After validating the simulation results using corresponding set of input data, the model is ready for future output prediction. The constraint optimizer uses Constraint Optimization Regression in Java (CoReJava) that implements Regression Analysis (RA) to estimate the parameters based on a training data that could either be historical data or experimentation data (Brodsky, et al. 2008).

The rest of the paper is organized as follows: next section identifies the V&V needs and issues. Then related work and technologies are discussed. The parameter validation technique is presented. A methodology to validate the technique, and a simple supply chain example modeled using CoReJava and a simulation tool are introduced. Finally the paper is concluded with future works and discussion.

VERIFICATION AND VALIDATION NEEDS AND ISSUES

During the development lifecycle of M&S, risks associated with potential errors in creating the model (programming errors) and inadequate fidelity (errors in accuracy when compared to real-world results) may be introduced (Cook and Skinner 2005). To guarantee that you have a valid model and simulation that produces correct results, V&V of the model and data used for the simulation must be employed throughout the life cycle of an M&S application.

Balci (Balci 2007) defines the model V&V as follows:
"Model validation is substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the study objectives. Model validation deals with building the right model. It is conducted by running the model under the "same" input condition that drive the system and by comparing model behavior with the system behavior. Model verification is substantiating that the model is transformed from one form into another, as intended, with sufficient accuracy. The accuracy of transforming a problem formulation into a model specification or the accuracy of converting a model representation in micro flowchart into an executable computer program is evaluated in model verification."

Figure 1 is the Sargent's circle - a simplified version of the M&S process (Sargent 2007). The problem entity shown in the figure could be a real or proposed system, idea, situation, policy, or phenomena to be modelled.

1. Conceptual model validity should answer the questions: Is the description of the system sufficient and correct? Is it valid for the intended use?
2. Computerized model verification deal with the questions: Is the numerical implementation of the model correct? Are the numerical algorithms employed correct and fully converged?
3. Operational validity answers the questions: Are we able to predict the experiment(s) in sufficient detail? How do we formulate quantitative validation metrics given a specific application?
4. Data validity answers the questions: Is the experimental data used in the comparisons a sufficiently accurate description of reality? How do experimental uncertainties affect predictive performance? Are the experiments used in the validation exercise appropriate?

To perform a complete validation of the model, appropriate validation techniques need to be applied to each step. Figure 2 shows the simulation modeling process steps and each of them may be a source of errors that will influence the validity of the model. For example, incorrect conceptual modeling will make the model invalid for the intended use. Lack of calibration of the parameters will not produce an accurate function that sufficiently describes the problem. Implementation error will make the simulation model invalid even if the conceptual model is valid. The use of poor quality input data will increase the risk of providing incorrect results to the users, i.e. trash in will have trash out. Unsatisfied operational conditions will cause wrong estimates. Results comparison is to compare the observed data and the simulation outputs. A large number of data are needed to have a meaningful evaluation of model performance in statistical terms. We should note that the model predictions and measured data will never match exactly; treads over time are one of the most useful tools to evaluate model performance (Donatelli and Stockle 1999).

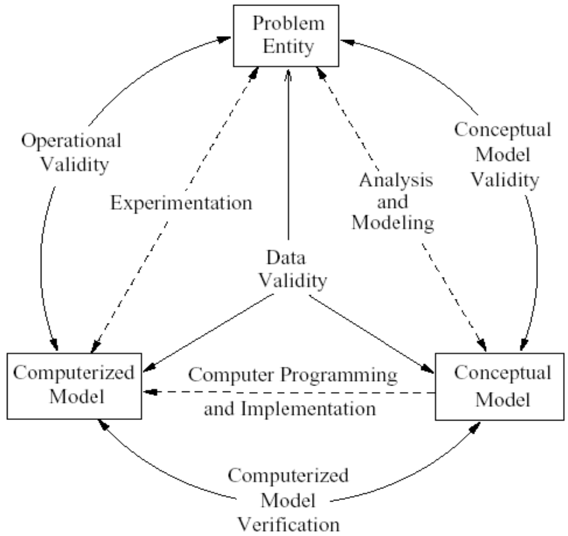

Figure 1. Simulation modeling and validation process (Sargent's circle) (Sargent 2007)

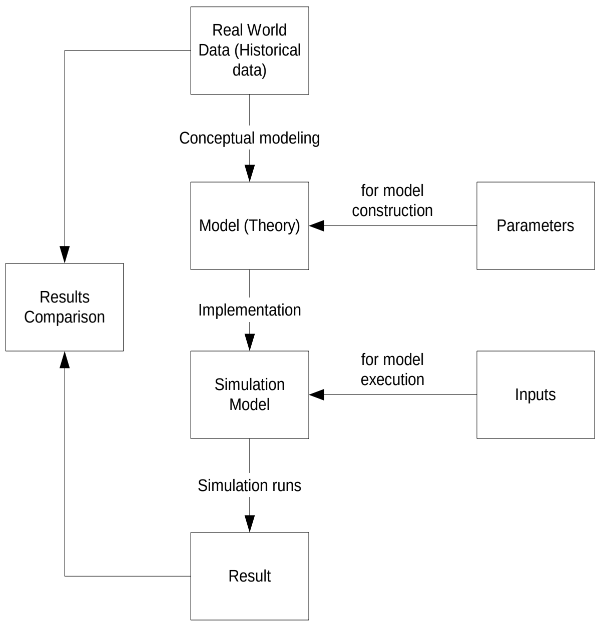

Figure 2. Simulation modeling and executing steps

RELATED WORK AND TECHNIQUES

This section discusses the related research work, techniques, and tools for the proposed methodology.

Law and Kelton suggested that quantitative techniques should be used whenever possible to test the validity of various components of the overall model (Law and Kelton 2000). An important technique for determining which model factors have a significant impact on the desired measures of performance is sensitivity analysis, the factors may be the value of the parameters, or the choice of a distribution, etc. One approach to determine the sensitivity of the factors is to use statistical experimental design. In experimental-design terminology, the input parameters and structural assumptions composing a model are called factors; the output performance measures are called responses. Factors can be quantitative or qualitative, controllable or uncontrollable (Law and Kelton 2000).

In (Doebling and Hemez 2007), model validation is also defined as “The process of assessing and improving confidence in the usefulness of computational predictions for a particular application” and “Solving the right equations.” Model validation is an application-specific process. Fidelity of the model relates to agreement with real world/test data, validity relates to suitability for the specific application. Validity of a model is defined over a region of the parameter space.

Model validation supporting technologies include:

- Metamodeling - simplified relationship between model parameters and response features,
- Design of Experiments - generate metamodels & plan validation tests,
- Parameter Optimization - quantify unmeasured variables, calibrate surrogate mechanics models, and
- Data Compression - extracting features from simulation and test data.

Machine learning techniques enable us to estimate the system parameters with specified confidence intervals using historical data, predict the outcome by given a new input, identify adjustment to the system parameters to meet performance requirements (Zabaras 2003).

Brodsky, Luo and Nash proposed and implemented the language CoReJava, which extends the programming language Java with Regression Analysis (RA), i.e. the capability to perform parameter estimation for a function. In a Java program, some parameters are not a priori known, but can be learned from training sets provided as input. Existing RA software typically requires inputting a data structure that describes the parametric functional form, or assumes this data structure to be fixed. The problem, however, is that in many applications, a functional form is not explicitly available. CoReJava allows the user to encode complex computational processes in Java, in which some parameters used are not a priori known. Unknown parameters can be learned from a training data set. The CoReJava compiler analyzes the structure of the learning function method to automatically generate a constraint optimization problem, in which constraint variables correspond to parameters that need to be learned. The objective function to be minimized is the summation of squares of errors with respect to the training set, and then solves the optimization problem using the non-linear optimization solver - A Mathematical Programming Language (AMPL)/SNOPT. (Brodsky, et al. 2008) provides detail descriptions of language syntax, use, and semantics of CoReJava.

AMPL is a comprehensive and powerful algebraic modeling language for linear and nonlinear optimization problems, in discrete or continuous variables. A few solvers such as CPLEX 11, SNOPT, and MINOS are free to download from (Bell 2008).

PROPOSED PARAMETER CALIBRATION AND VALIDATION METHODOLOGY

The proposed novel validation methodology will focus on the parameter calibration and validation as shown in Figure 2. A validated set of system parameters for a specific problem function will allow users to properly characterize the system under study. There is no universal model that would work with an unaltered set of parameters for all conditions. Adjustment of parameter values must be done within the range known for the parameters. Calibration assumes the availability of observed data to adjust model parameters in order to match model outputs to measured data.

Depicted in Figure 3 is the model parameter calibration and validation approach. In practice, most of the validation processes do not start until the completion of M&S development. In our proposed methodology, the validation of the simulation model can be parallel to the development. First we build the simulation template with unknown key parameters included. We can think of it as a black-box model. To do this, we need to first analyze the data available and decide what we should measure and what parameters we do not know and need to be estimated. Then we start the parameter validation process, that is to find out the “correct” model parameters. This can be done by applying an RA tool. The RA module will find the best estimate of the unknown parameters by learning from the available training data sets. The training data sets may be either real world historical data or experiments data. RA is one of the metamodeling techniques for investigating and modeling the relationship between variables. As input to RA, a parametric functional form can be either linear or non-linear, e.g., $f(x_1, x_2, x_3) = p_1x_1 + p_2x_2 + p_3x_3$, and a set of training examples, e.g., tuples of the form (x_1, x_2, x_3, f), where f is an experimental observation of the function f value for an input (x_1, x_2, x_3). The goal of RA is to find the unknown parameters, e.g., p_1, p_2, p_3 that “best approximate” the training set. Once we find the optimal set of the parameters, we replace the variables in the simulation template with the parameters values. The simulation model then becomes a deterministic model or a white-box model. By feeding in new inputs, the simulation can produce and predict valid outputs. When compare the simulation results to the historical data, we hope that the simulation will duplicate as closely as possible the collected data within a confident interval. If the comparison results are not satisfied, simulation template needs to be verified, or the parameters needs to be re-estimated using more data or new data, even the simulation model execution condition including input data needs to be checked to make sure sufficient runs are performed. After iteratively comparing with the training data and modifying the model, we can eventually obtain a more validated model over that particular parameter domain and valid data range.

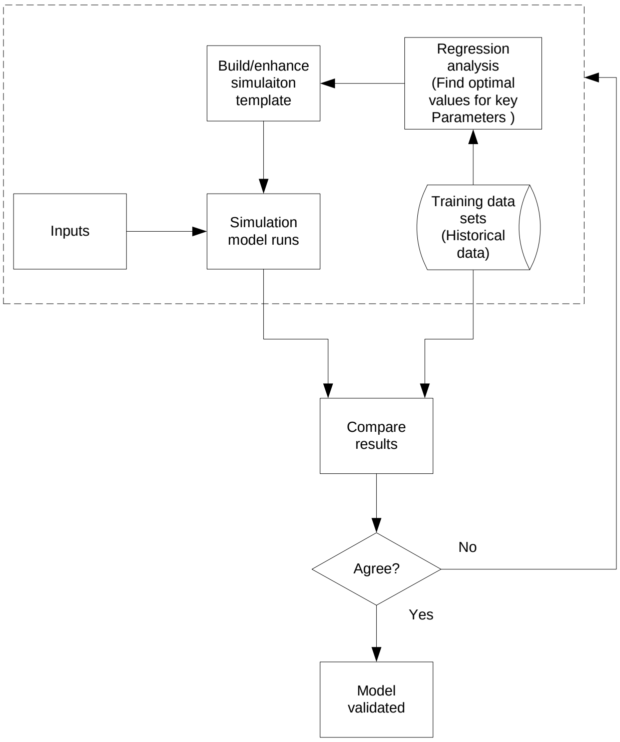

Figure 3. Simulation model parameter validation approach

VALIDATION METHOD FOR THE PROPOSED TECHNIQUE

In order to validate the proposed technique, we need large sets of historical data. Based on the different category of the collected data, we can divide the data into at least two groups, for example, use one set of data to train and next set of data to validate the result. We should use one of the groups as the RA training data, once we derived the best estimate model parameters values, the other data sets can be used to verify the result. We can check if the differences between the simulation outputs and the real collected data are within a confident interval.

We use a simple example to explain the proposed methodology. Figure 4 depicts a simplified supply chain cost model, a functional form may be given that computes the total cost of manufacturing, given three products to be produced. This functional form may have unknown parameters, e.g., the unit costs and the required quantities of component materials to produce specific products.

The manufacturer produces three products using three components. The quantities of component materials needed are functions of the required quantities of products. The cost of the produced products is the total cost of the required components. Thus, the cost of manufacturing is a function of the required quantities of products. However, the coefficients of this function, i.e. the unit cost of each component and the amount of each component material to produce 1 unit of each product may be unknown and subject to RA, which is provided by CoReJava.

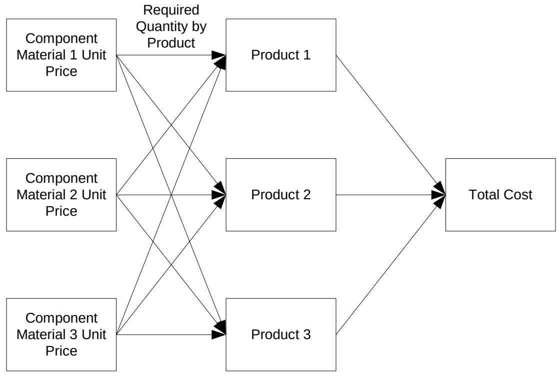

Figure 4. A Simple supply chain cost example

The example historical data as training data set is listed in Table 1. Each row is a learning set. Each learning set includes three product quantities and the actual total cost recorded. The size of the table is decided by the information the user has collected (Brodsky, et al. 2008).

Table 1. Input learning set

Product 1 Quantity	Product 2 Quantity	Product 3 Quantity	Actual Total Cost
8	5	9	20
9	7	6	18
7	6	14	25
10	11	12	29
5.5	12.1	9.8	31.1
…	…	…	…
11.2	9.6	6.5	25.33

Figure 5 shows the simulation result using CoRejave for a product set of (64.2, 50.4, 35.5) as inputs, i.e. the product1 quantity is 64.2, product2 quantity is 50.4, and product3 quantity is 35.5. We need to determine the actual total cost for this set of products. In Figure 5, the two-dimensional array reqMatQty represents the quantity of each required material to produce one unit of every product. The array matUnitCost represents the unit cost of every material. The data includes the values of coefficients (matUnitsCost and reqMatQty arrays) and the total cost (Brodsky, et al. 2008).

For comparison purpose, a simulation model of the same problem was also developed using a simulation software, which does not have a RA module. Data in Table 1 alone is not sufficient for constructing the model. Important data such as component unit price, and amount of components material needed is not available. A triangular distribution is typically used in a model for a source of randomness when no system data are available. T (0.3, 1, 3) is chosen for that, we used the min_Bound value in CoReJava example as the minimum value of the triangular distribution, which is 0.3 and used the max_Bound value in CoReJave example as the maximum value, which is 0.3, for the triangular distribution, then arbitrarily choose a mode value as 1. From the results showed in Figure 6, we can see that the results are far apart from the results of CoReJava. This is because the model did not incorporate the existing data sets. From the results comparison, we can see this model did not accurately describe the problem. By using the proposed technique, the parameters are ensured to be learned based on the past historical data. The model will properly represent the problem. The results will be within a confident interval of the existing data. That makes the proposed model more valid for the specific application.

```
[echo] #### 4. Interpreting the optimized results...
[java] objective: 5533.03269354264
[java] matUnitCost[0]:0.3
[java] matUnitCost[1]:0.3
[java] matUnitCost[2]:0.3
[java] reqMatQty[0][0]0.741492
[java] reqMatQty[0][1]0.3
[java] reqMatQty[0][2]0.45798
[java] reqMatQty[1][0]0.741492
[java] reqMatQty[1][1]0.3
[java] reqMatQty[1][2]0.45798
[java] reqMatQty[2][0]0.868433
[java] reqMatQty[2][1]0.3
[java] reqMatQty[2][2]1.03503
[java] The cost to produce a set of products:19.067327489999997
```

Figure 5. Simulation result using CoReJava.

Report1 for products_v2 (Avg. of 999 replications)

General | Locations | Location States Multi | Location States Single/Tank | Resources | Resource States | Node Entries | Failed Arrivals | Entity Activity

Variables for products_v2 (Avg. of 999 replications)

Name	Total Changes	Avg Time Per Change (MIN)	Minimum Value	Maximum Value	Current Value	Avg Value
s sa 3 21	0.00	0.00	0.00	0.00	0.00	0.00
number of product1	0.00	0.00	64.20	64.20	64.20	64.20
number of product2	0.00	0.00	50.40	50.40	50.40	50.40
number of product3	0.00	0.00	35.50	35.50	35.50	35.50
product1 cost	1.00	0.00	0.00	6.09	6.09	6.09
product2 cost	1.00	0.00	0.00	6.13	6.13	6.13
product3 cost	1.00	0.00	0.00	6.23	6.23	6.23
total cost	1.00	0.00	0.00	920.96	920.96	920.96

Figure 6. Same example modeled using distribution

CONCLUSION

M&S techniques are increasingly used to solve problems and aid decision making in many different fields. Results of simulations are expected to provide reliable information for the decision makers. However potential errors may be introduced in the process of the M&S development lifecycle. It is critical to make sure to build the right model and that the model is built right.

This paper demonstrated a novel approach of unknown parameters calibration and validation through constraint optimization based on training data sets within a data range. The technique proposed a parallel process of developing and validating the simulation model, before every parameter is known. A simulation template can be built with unknown variables in it. By using a built-in RA module that learns the training data sets, the best estimates of the variable values can be derived. This will help to ensure the accurate relationship between indepedent inputs and dependent output. Future work may include the development generic UML model for the RA module. That will provide a software independent design of the validation technique. Any interested simulation software vendor can implement it as a module of their product. Also more implementation for real world problems may be needed to validate the technique.

REFERENCES

Sargent, R. 2007. "Verification and validation of simulation models," In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton. 124--137.

Law, A. and W. Kelton. 2000. *Simulation Modeling and Analysis*, McGraw-Hill Higher Education, Boston.

Balci, O. 2007. "Validation, Verification, and Testing Techniques throughout the Life Cycle of A Simulation Study," *Annals of Operation Research*, 53 (1994), New York, pp. 121-173.

Jain, S. and C. R. McLean. 2006. "An Integrating Framework for Modeling and Simulation for Incident Management," *Journal of Homeland Security and Emergency Management*, Vol. 3, Iss. 1, Article 9. Available at: http://www.bepress.com/jhsem/vol3/iss1/9.

Cook, D. and J. Skinner. 2005. "How to Perform Credible Verification, Validation, and Accreditation for Modeling and Simulation."

Sargent, R. P. Glasow, J. Kleijnen, A. Law, I. McGregor, and S. Youngblood. 2000. "Strategic Directions in Verification, Validation, and Accreditation Research," In *Proceedings of the 2000 Winter Simulation Conference.* J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.

Brodsky, A. J. Luo, and H. Nash. 2008. "CoReJava: Learning Functions Expressed as Object-Oriented Programs," International Conference on Machine Learning and Applications (ICLMA) Forum 2008.

Zabaras, N. 2003. "Simulation matching and model estimation using statistical learning techniques," *Advanced Mechanical Technologies*, GE Global Research Center, NY.

Bell Laboratories. 2008. AMPL. Http://www.ampl.com

Department of Defense. 2001. "Department of Defense Verification, Validation and Accreditation (VV&A) Recommended Practices Guide," Defense Modeling and Simulation Office, Alexandria, VA. (Coauthored by: O. Balci, P. A. Glasow, P. Muessig, E. H. Page, J. Sikora, S. Solick, and S. Youngblood.) http://vva.dmso.mil/Ref_Docs/VVTechniques/vvtechniques.htm

Doebling, S. and F. Hemez. 2008. "Finite Element Model Validation, Updating and Uncertainty Quantification, A 2-day short course." http://www.la-dynamics.com/courses/validation_sample_vg_0302.pdf

Shao, G. and C. McLean, 2008. "Emergency Room Simulation Prototypes for Incident Management Training," In *Proceeding of Industrial Simulation Conference 2008*, pp. 323-327.

Shao, G. and Y. Lee. 2007. "Applying Software Product Line Technology to Simulation Modeling of Emergency Response Facility," *Journal of Defense Modeling and Simulation*, The Society for Modeling and Simulation International, Vol. 4(4), October 2007. http://www.scs.org/pubs/jdms/vol4num4/vol4num4.html.

Donatelli, and Stockle. 1999. "Model calibration and validation, short course on the model cropsyst," Middle East Technical University – Dept. of Economics, Ankara, Turkey. http://www.bsyse.wsu.edu/cropsyst/Documentation/lectures/acrobat/lect2_Calibration.PDF