

A Method for Quantified Confidence of DEVS Validation

Megan Olsen, Mohammad Raunak

Department of Computer Science

Loyola University Maryland

4501 N Charles St., Baltimore, MD 21210 USA

mmolsen@loyola.edu, raunak@loyola.edu

Abstract

The Discrete Event System Specification (DEVS) framework provides a formal approach for defining a conceptual model that represents some real system. For the model to be useful in analyzing the real system, it must be validated. Generally DEVS models are validated in terms of behavior and structure, utilizing the experimental frame. Although there has been discussion on approaches to building valid DEVS models, and techniques for validating the model, there is no way to quantify our confidence in the model's validity. We propose a technique to quantify our confidence in the validity of DEVS models. In addition, this metric can be utilized to discuss the level of validity of a model, and compare competing models. Our metric therefore continues the DEVS trend of making modeling and simulation more formal, and increasing standardization of the modeling and simulation process.

Keywords: DEVS, Validation, verification

1. INTRODUCTION

The field of Modeling and Simulation (M&S) has been growing rapidly in its usage and influence. Different types of simulations are being used in a wide array of application areas. Computational simulations are now created to study many different fields, by both computer scientists and engineers, and by experts in the studied domain: cancer biology, ecology, psychology, and mechanical engineering to name a few. Regardless of the process that is followed and the tools that are used to develop these models and simulations, the models and simulations must be properly verified and validated before their results can be trusted. Validation, in particular, becomes a crucially important part of building confidence in the results derived from the use of simulation model.

Verification and validation (V&V) are defined a number of ways in both the software engineering and modeling and simulation communities. The definition we find to be most accurate and most widely referenced is that verification determines whether or not the model solves the problem correctly, whereas validation determines whether or not the model solves the correct problem. In essence, verification is testing *how* the model was built, and validation is assessing *what* has been modeled and whether the model can sufficiently reproduce the behavior of the real system for its simulation pur-

pose. There are many steps within the modeling and simulation process where each must be applied to achieve high confidence in the model and simulation (Balci 2010).

This definition of V&V has also been commonly accepted and applied to the special subclass of simulations defined using the primary formalism in the field, known as DEVS (Discrete Event System Specification) (Zeigler, Kim, and Praehofer 2000). DEVS provides a mechanism by which a model can be explicitly defined separately from a simulation, which is often lacking in simulation practice. As Tolk recently showed, one of the issues plaguing the field of Modeling and Simulation is a lack of explicit terminology, including a definition of a conceptual model (Tolk, Heath, Ihrig, Padilla, Page, Suarez, Szabo, Weirich, and Yilmaz 2013). We argue that the DEVS model is in essence the conceptual model of the simulation, and should be treated as such for V&V.

DEVS has increased the rigor in the field and addressed issues with explicit model definitions. However, the formalism has not been fully extended to incorporate verification and validation. In general, validation is performed to increase one's confidence in the model. In DEVS, as with other M&S approaches, there is no mechanism to quantify confidence in the validity of a simulation model based on the performed validation. Thus the users of the simulation model, who may not be the ones who developed it, may find it difficult to decide which available model is the best to use for their specific purpose. The DEVS community has increased emphasis on standardization (Wainer, Al-Zoubi, Hill, Mittal, and Martin 2011), and we propose the standardization should extend to include a process to quantify confidence in simulation models built using the DEVS formalism. We argue that following such a standard process will make the level of validation performed and corresponding confidence in the model clear to both the model's creator and any other party considering the model, which in turn will help increase the reuse of models.

In this paper we present an approach to calculate an explicit confidence level for a DEVS simulation model, based on validation performed on that model. We focus on behavioral, structural, and data validation of the simulation model. In the next section we discuss related work in validation within and outside of DEVS, then provide an overview of V&V in DEVS, followed by a description of computing behavioral confidence level, structural confidence level, and confidence

on the model data. We show how our proposed framework combines these three confidence levels to produce an overall quantified level of confidence of the whole model. We end with an example of applying our approach on a DEVS model and conclude with a discussion on the potential impact and future direction of this research.

2. RELATED WORK

For DEVS models, it can be difficult to make a careful distinction between whether the DEVS model is coded correctly (verification) and whether it represents the source system closely enough for its purpose (validation). DEVS discusses validation through the use of an experimental frame, which defines the purpose, behaviors, structure, and data of the model (Zeigler, Kim, and Praehofer 2000). Structural validation focuses on the static parts of the model and ensures, through the application of specific validation techniques, the structural consistency and integrity. Behavioral validation focuses on the dynamic part of a DEVS model.

The M&S community in general considers the validation process to include a) conceptual model validation, b) operational validation of the simulation model, and c) data validation. They primarily focus on operational validation, as the conceptual model is not generally as well defined. They have identified a number of validation techniques that can be applied to increase one's confidence in a model, such as extreme condition tests, results validation, and face validation (Sargent 2010; Forrester and Senge 1980). It is well accepted that these techniques should be applied throughout the modeling process, in addition to performing verification checks (Balci 2012). However, less focus has been placed on quantifying the V&V effort applied on simulation models in general.

Labiche investigated the application of software engineering testing techniques for performing V&V on DEVS models (Labiche and Wainer 2005). This paper discusses selecting test cases based on some criterion, but does not specify any mechanism for quantifying the coverage achieved through that selection. Zengin examines the question of how much validation is enough by performing a case study on a DEVS-Suite model (Zengin and Ozturk 2012). They examine available validation techniques and discuss the difficulties of validating simulation models. They find DEVS-Suite to be a useful tool for validation, and suggest that a tool is needed to aid in successful validation approaches.

Wang proposed a method of assessing the V&V activities based on detection, classification, and documentation of model defects during the simulation model construction period (Wang 2011). The focus of Wang's work is properly classifying the defects, akin to IBM's orthogonal defect classification (ODC) (Chillarege et al. 1993). Later Wang proposed a framework for developing a catalog of V&V techniques and selecting specific V&V technique for an M&S project based

on their applicability (Wang 2013), which could be used in tandem with our work.

Hollmann has attempted to formalize the verification and validation process of DEVS modeling (Hollmann, Cristia, and Frydman 2014). They propose a method to choose the most significant set of configurations to use for validation of a DEVS model to increase confidence in the model's results. Our work builds on this work as we focus not on choosing parameters for running the simulation, but instead on criteria for measuring confidence in the model afterward.

Although validation is performed on most models, it is not generally discussed sufficiently (Elele, Hall, Farid, Turner, Davis, and Keyser 2014; Raunak and Olsen 2014b). There is an imminent need to improve the documentation of validation activities performed on simulation models, which could be accomplished by standardizing and quantifying validation. Within DEVS there have been numerous steps to standardize, such as standardizing the simulation of DEVS models in tools (Li, Vangheluwe, Lei, Song, and Wang 2011), or formalizing models and their context through experimental frames to improve validation (Traore and Muzy 2006).

In our earlier works, we have developed approaches to quantify validation coverage in agent-based models (Olsen and Raunak 2013) and in discrete-event system models (Raunak and Olsen 2014a). In this paper we build on our earlier works by overhauling the approach to focus on calculating a level of confidence in the model through the validation of behavior, structure, and data for DEVS, leading to an overall quantified level of confidence in the entire model.

3. VALIDATION PROCESS

The process of modeling and simulation is better defined in DEVS than in many other approaches, as can be seen in the overview in Figure 1. The DEVS formalism provides a concrete conceptual model, and thus conceptual model validation can be more rigorously performed. The formalism is ideally built using data about the real system that is being studied, known as the source system (also known as System Under Study). Generally an executable simulation model is built to experiment with the conceptual model, using one of the many frameworks available, such as CD++ (Wainer 2002), DEVS-C++ (Cho and Cho 1997), or PowerDEVS (Bergero and Kofman 2010). These experiments are run using an experimental frame, which is essentially a mechanism for clarifying the data, parameters, and structure of the simulation model. The experimental frame performs two roles: providing data for the creation of the model, and including all data and parameters necessary for running the simulation. For the results of the simulation model to be trusted, it must be verified against the conceptual model, as well as operationally validated using the experimental frame.

We define the experimental frame as in (Zeigler, Kim, and

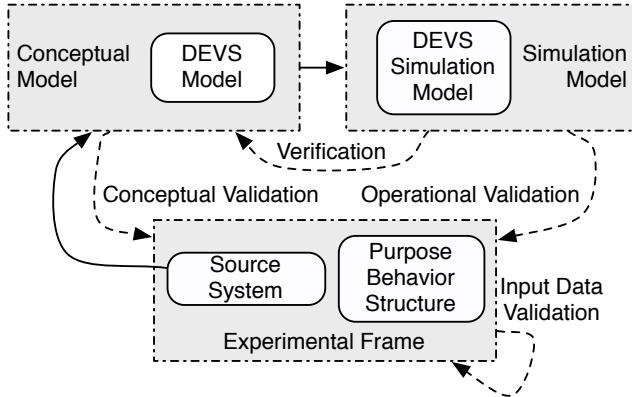


Figure 1. Overview of the DEVS modeling process, with a focus on verification and validation.

Praehofer 2000), that it can be utilized toward two related goals: defining the types of data to be used, and interacting with the model to either measure or observe results. We use the experimental frame for both purposes, to define our data and as an oracle when an oracle is possible.

Although DEVS provides a useful formalism for defining a conceptual model, it does not necessarily aid in operational validation (how well does the model mimic the source system's behavior). To determine the extent of validation, and to properly validate a simulation model, we must be able to determine the aspects of the real-world system that are validatable within the simulation model. The fact that DEVS is the underlying modeling approach is irrelevant at this stage; validation does not care about how the simulation model is created, only what is being modeled and how it is being assessed for operational accuracy. For a single DEVS model, whether atomic or coupled, any aspect specific to the formalism (input/output of an atomic model, links between coupled models, etc) is a part of the simulation that needs to be verified for transformational accuracy between the conceptual model and the simulation model. In the case of verification, quantifying the success is similar to any software system; we could use a coverage metric that counts whether each of the formalism's statements works as planned, for instance. However, that metric does not help us compare the simulation to the broader real-world picture of the system being modeled.

We consider validation of DEVS to encompass the three main properties in the experimental frame: purpose, behavior, and structure (Zeigler, Kim, and Praehofer 2000). The purpose guides us, as it defines why the model was built and what type of questions may be asked of it. The behaviors and structure of the model are based on that purpose. The behaviors are essentially pairs of input and output; given a certain state/input of the model, how should it act? When validating behavior we are validating structure indirectly, but to have full confidence in a model, we also need to perform explicit

structural validation. Therefore, to calculate the level of confidence in the model, we must track the validation of both behaviors and structure, with regard to purpose, and validate the data used in simulation model execution.

4. OVERVIEW OF CALCULATING MODEL CONFIDENCE

We have argued that to measure the confidence level on a DEVS simulation model completely, the behavior, structure, and data of the model need to be validated and the level of that validation needs to be assessed. In this section, we propose a process and present techniques for calculating confidence in the model's validity for each of these aspects.

The first step in behavioral validation is to identify all of the behaviors captured by the model that require validation. The second step involves determining what validation techniques are applicable to each of the validatable behaviors that have been identified. The next step is to perform validation on the simulation model by applying those validation techniques. When a particular validation technique is applied on a simulation model, it may validate more than one behavior, as many behaviors may be interconnected in such a way that they may not be validated separately. As validation is performed, the validation techniques applied and their level of success needs to be recorded. The final step in is to calculate our confidence in the model. This confidence level quantifies the amount of validation performed successfully versus the amount that could be performed on the model behaviors.

From the perspective of the experimental frame, the structural aspect of a DEVS model incorporates elements of both the conceptual model and simulation model. Quantifying structural validity requires fewer steps, however. There is a set of techniques already identified to be applicable for performing structural validation (Forrester and Senge 1980). We propose applying these techniques and using the percentage of these techniques that have been successfully applied to determine our level of confidence in the structural validity of the model. Similarly, we propose calculating data validation based on the successful application of a set of validation techniques on each set of input data to ensure it properly matches the source system and is free of outliers and other errors.

We present a mechanism for combining all three quantified aspects into an overall confidence level for the validity of the model. All levels of confidence are in the range of 0 to 100, with 100 representing the maximum possible confidence in model but not that the simulation is 100% correct. However, it is often impossible to fully validate a simulation model due to constraints such as budget, time, and the absence of required data. A desired level of confidence should therefore be determined at the requirements stage, to guide later validation efforts. Additionally, the confidence calculation can also guide the choice of validation techniques to apply as the goal

should be to perform the validation that will increase model confidence most efficiently.

5. BEHAVIORAL VALIDATION

To calculate our confidence in the validity of a model, we first determine the behaviors of the model that need to be validated. We consider behaviors to encompass a single high-level input/output pair in the DEVS model, related to a definable expectation in the source system. Multiple behaviors will generally be necessary to define a specific module. For instance, in a hospital emergency department (ED) simulation, a behavior could be the process by which a patient is placed inside the ED upon arrival in a simple case, or pieces of that process flow (e.g., registration) in a more complex model. Similarly, while simulating the operations of an ATM machine, a behavior could be the process of withdrawing money or a piece of that process such as checking whether the entered PIN is correct or not and the result of that check.

To aid in determining the behaviors of the model, we suggest utilizing the previously published categories presented as *aspects* and *oeis* (observable emerging information) of a discrete event simulation (Raunak and Olsen 2014a). Although the coverage metric proposed in that work is different from our confidence level in this paper, the simulation behaviors being discussed here are identical to that concept of *oeis*.

Once we have identified all behaviors that must be validated, we must determine the applicable validation techniques for each of them. A single technique may apply to many different behaviors, and each behavior will often be validated with more than one applicable technique.

In our proposed framework we take into account the fact that not all validation techniques will contribute equally toward building confidence on the simulation model. By applying any validation technique successfully we increase our confidence in the validity of the model. However, which validation technique(s) should be applied for each behavior? We propose utilizing the rankings in Table 1. Each commonly accepted technique is listed with its relative level of possible confidence increase. This proposed ranking is based on a survey of the most commonly used validation techniques (Raunak and Olsen 2014b), a discussion of their uses (Sargent 2010), and the authors’ experience. In general we rate Animation the lowest as it can be misleading and hide underlying issues, but rank results validation and trace data the highest as they can be very powerful techniques when the required data exists. We rate expert validation slightly below the validation involving data due to the potential for human error, and model comparison slightly less due to difficulty in knowing the validity of the other model without a standard confidence calculation. However, additional techniques exist and can be added and ranked as necessary given the purpose of the model, as well as to encompass additions to DEVS such

Technique	Maximum Confidence
Animation	3
Degenerate Tests	4
Internal Validity	5
Comparison to other model	6
Turing Tests	7
Face validation (expert)	8
Results validation (historic data)	10
Trace data	10

Table 1. Ranking of the most common validation techniques with the maximum level of confidence they can provide for a given behavior.

as Cell-DEVS (Wainer, Frydman, and Giambiasi 1997).

Given a simulation model to validate, techniques are chosen based on their usefulness in creating the highest possible confidence in the model. There may be more validation techniques that are applicable than are necessary to achieve a reasonable level of high confidence in the model. For the highest model confidence, however, we must apply every applicable validation technique successfully. If it is not possible to apply all applicable techniques for any reason, we can determine our level of confidence as described in the next section.

5.1. Confidence Calculation

Given a simulation model’s list of behaviors, the list of applicable validation techniques by behavior, and the list of successfully applied validation techniques, we can quantify our confidence in the model. For some DEVS simulation model M , let $B = \{b_0, b_1, \dots, b_n\}$ be the set of behaviors to be validated, and $T = \{t_0, t_1, \dots, t_m\}$ be the list of available validation techniques for operational validation such as those in Table 1. Each of these elements has an associated confidence weighting $w(t_i)$, defined as in Table 1. For b_i , the weight $w(b_i)$ is determined based on the importance of this behavior for model validation, and $0 < w(b_i)$. In the simplest case all behavior weights are equal.

When performing validation it is possible that a single validation technique could be applied successfully and validate multiple behaviors simultaneously. Therefore a validation technique must be paired with some set $D \subseteq P(B)$, $|D| \geq 1$. Here $P(B)$ denotes the power set of B . It is also true that a validation technique t_i may be used in multiple ways when validating a single simulation, each time used to validate a different set of behaviors by focusing on a different aspect of the simulation model and/or using different sets of available data or experts. We therefore must consider a single validation approach $v_i \in V$ as a triplet,

$$v_i = \{t_i, d_i, c_i\}, \quad (1)$$

where $t_i \in T$, $d_i \in D$ is a set of b_j s, and c_i is the corresponding

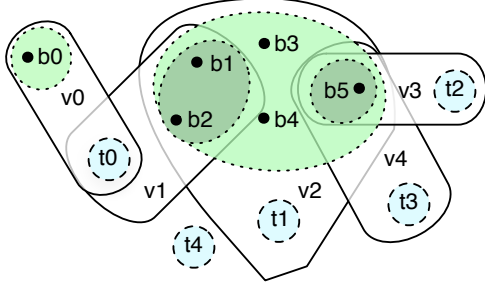


Figure 2. A visualization of each element of a v_i . Each t_i is a validation technique, each b_j is a behavior, each dotted circle around a set of b_j is a d_i , and each full shape is a v_i . Note that each b_j can be in multiple v_i . The size of the shape represents c_i .

confidence gain from applying t_i to d_i . There may only be one $t_i \in v_i$, as each validation technique should be considered independently. Each b_j and t_i may be duplicated as many times as necessary in V . The relationships between v , t , and b are demonstrated in Figure 2.

The $c_i \in v_i$ represents the confidence gain from t_i 's successful application. It is based on the weight of each behavior in d_i represented as $w(b_j)$, and the weight of the validation technique $w(t_i)$ as seen in Equation 2. Therefore, higher importance behaviors contribute more to our confidence in the system. Additionally, more powerful validation techniques contribute more to our confidence than less powerful techniques.

$$c_i = w(t_i) * \sum_{j=0}^{|d_i|} w(b_j), \text{ where } b_j \in d_i \quad (2)$$

Recall that our level of confidence is based on what validation techniques have been applied successfully. Each validation technique successfully applied raises our confidence in the validity of the model. To quantify this level, we must define the highest possible confidence. We base the maximum confidence on the potential validation approaches

$$P = v_0, v_1, \dots, v_p, \quad (3)$$

where $P \subseteq V$. Additionally, we must know which validation approaches succeeded,

$$A = v_0, v_1, \dots, v_a, \quad (4)$$

where $A \subseteq P$. We assume that any validation approach v_i that succeeded, did so for all $b_j \in d_i$. Finally, we may calculate our confidence in the behaviors across the entire simulation model bc as in Equation 5.

$$bc = 100 * \frac{\sum c_i |c_i \in v_i \wedge v_i \in A}{\sum c_i |c_i \in v_i \wedge v_i \in P} \quad (5)$$

Our level of confidence in the behavioral aspect of the simulation model is therefore a real number, $0 \leq bc \leq 100$, where

100 represents the highest possible confidence. Note that confidence does not equal correctness, and this value does not imply that the simulation model is 100% correct. A confidence level of 100 only implies that the simulation model is as correct as it is possible to determine.

6. STRUCTURAL VALIDATION

The structure of a model is indirectly validated through behavioral validation, but for the highest possible confidence in a model it should be directly validated as well. Forrester and Senge (Forrester and Senge 1980) recommend the tests listed in Table 2 for validating structure.

Each technique plays an important role in structural validation. Although many have the word “verification” in the name, they are validation tests as they are comparing the model to the real system. The structure verification test verifies that any structure found in the model is also found in the source system; it is the easiest test to pass. The parameter verification test evaluates whether the parameters adequately match the real system, which is only true if the correct structure underlies the model. Extreme condition tests are necessary to determine that the model will function in abnormal conditions; if it does not, then more structure is needed, but if it does then the correct structure has been chosen. The boundary adequacy test for structure essentially tests if the correct level of abstraction has been chosen, which correlates to structure. The dimensional consistency test can reveal incorrect structure assumptions when run with the parameter verification test, and is another simple and easy way to increase model confidence.

To have full confidence in our simulation model, all five structural validation tests should be successfully administered. However, there may be constraints that prevent a test from being applied. In that case, the confidence in the model should be calculated as discussed in the following section.

6.1. Confidence Calculation

As with behavioral validation, each structural validation technique may be applied at different levels of success. In this case we propose that each structural validation technique st_i has the confidence level $c(st_i)$ attainable via successful application, as shown in Table 2. This confidence level is based on (Forrester and Senge 1980; Zengin and Ozturk 2012).

Each structural validation test may validate a specific percentage of the model. For instance, our model may pass 90% of the tested extreme conditions. We have thus gained some confidence in our model, but not full confidence as we are aware of extreme condition test failures that must be fixed before trusting the model's results in all cases. We therefore propose that our overall confidence in the structure of the model may be quantified as in Equation 6, where $p(st_i)$ is the level of successful application of st_i (0.9 for extreme condition testing

Technique (st)	Potential Confidence (c(st))
Structure verification test	10
Boundary adequacy test	15
Parameter verification test	20
Dimensional consistency test	20
Extreme condition test	35

Table 2. Ranking of the most common structure validation techniques with their maximum possible level of confidence.

in our example above).

$$sc = \frac{\sum_i c(st_i) * p(st_i)}{\sum_i c(st_i)} \quad (6)$$

Thus if all tests have been applied thoroughly and successfully, we will have the highest possible confidence in the structure of our model, which we quantify as 100%. If all were successful except extreme condition tests, which had a 90% successful rate, our confidence in the model’s structure becomes 94%. Recall that this value is our level of confidence in the correctness of the structure matching the source system, not the percentage of the system we know to be correct.

7. DATA VALIDATION

A primary aspect of validation is to ensure the validity of the data used both for model creation and for model execution. Sargent describes three necessary processes for data validity: “(1) collecting and maintaining data, (2) testing the collected data using techniques such as internal consistency checks, and (3) screening the data for outliers and determining if the outliers are correct” (Sargent 2010).

However, we can also validate our input data specifically to ensure that it is consistent with the source system. In DEVS, this input data is part of the experimental frame, and used for running the behavioral validation. A key differentiation here is that there are two types of data: the data used to build the model, and the data used to run the model. Here data validity refers to the validation of data used to run the model, as the data used to build the model is validated through structural validity as discussed in section 6..

We propose that the following validation techniques are necessary for data validation, based on (Sargent 2005): Goodness of Fit Test and Face Validity.

7.1. Confidence Calculation

All applicable validation tests need to be run on all input data for achieving maximum confidence that we are using the correct data for our simulation. To calculate confidence in the data, we need to enumerate the list of data $I = \{i_0, i_1, \dots, i_r\}$. This list is part of the model’s experimental frame, and we as-

sume that the modelers will collect it as part of their modeling and validation work.

Let $DP = \{dp_0, dp_1, \dots, dp_p\}$ denote the set of applicable data validation techniques. For each dataset i_j we have the set of successfully applied data validation techniques $ADP_j \subseteq DP$. If we assume that all data validation techniques are equally important, then $c(i_j)$, the confidence in each dataset i_j , can be computed by Equation 7.

$$c(i_j) = 100 * \frac{|ADP_j|}{|DP|} \quad (7)$$

A simulation modeler may decide that some input data is more important to validate than others. In that case, we propose incorporating the relative importance of the input data by attaching weights to each i_j denoted as $w(i_j)$. The overall confidence in data validity is then defined as in Equation 8.

$$dc = \frac{\sum c(i_j) * w(i_j)}{\sum w(i_j)} \quad (8)$$

8. OVERALL CONFIDENCE

Once we have confidence in the behavior, structure, and input data of the simulation model, we can define our confidence in the model as a whole. We propose that behavioral and structural validation are equally important. However, as behavioral validation indirectly validates some aspects of structure, their validation confidence levels should not equally influence our overall confidence in the simulation model. Additionally, data validation is crucial for correct behavioral validation, but very hard to perform completely across all data for the system. In reality, assumptions of data validity based on data source and collection are more common. We therefore propose the following calculation for overall model confidence mc :

$$mc = 0.5bc + 0.3sc + 0.2dc, \quad (9)$$

where bc , sc , and dc are as defined previously. Thus, our confidence in model validity is slightly more affected by behavioral validation than structural or data validation.

9. EXAMPLE

To demonstrate the confidence calculation we use a model of an Automated Teller Machine (ATM), such as the publicly available one on the CD++ Sample Models page (Wainer), as an example. Our only modification is to allow PIN and Amount to be input instead of assumed rates. The purpose of this model is to represent a cash machine.

9.1. Behavioral Validation

To validate the behaviors, we must first determine what behaviors of an ATM should be validated to match the source system. Let’s assume that the behaviors are as follows:

1. b_0 = An incorrect PIN number causes card return and no additional screen options, $w(b_0) = 3$
2. b_1 = Requested money is given if account has sufficient fund, otherwise not, $w(b_1) = 4$
3. b_2 = The user interface question order follows the correct workflow, $w(b_2) = 2$
4. b_3 = The card is always returned after completed use, $w(b_3) = 2$

Note that most behaviors do not have an obvious correlation to a single atomic model within the DEVS conceptual model. One confusion in this model is that many of the behaviors to be validated are also aspects of the model that must be verified as well. In a more complex model that would not be the case, as behaviors would be further removed from the atomic and coupled models. However, verification would involve looking at finer grained detail about individual aspects of the model and how they work to accomplish the goals.

Next we must determine the sets of potential validation approaches, $P \subseteq V$. Let's assume that the validation techniques from Table 1 are the only available validation techniques, and that they are applicable as follows:

- $p_0 = \{\text{Animation}, \{b_0, b_1, b_2, b_3\}, 33\}$
- $p_1 = \{\text{Face Validation}, \{b_0, b_1, b_2, b_3\}, 88\}$
- $p_2 = \{\text{Results validation}, \{b_1, b_2\}, 60\}$
- $p_3 = \{\text{Results validation}, \{b_0, b_3\}, 50\}$

The next step is to perform the validation. If all tests were run successfully, then our set of applied validation techniques would be $A = P$, and our confidence in the model's behavior would trivially be 100.

Alternatively, if we were unable to run a validation approach, such as p_0 , we would have $A = \{p_1, p_2, p_3\}$ and our confidence in the model behavior would be

$$bc = 100 * \frac{88 + 60 + 50}{33 + 88 + 60 + 50} = 85.7\%. \quad (10)$$

9.2. Structural Validation

Let's assume that of the five available structure tests, we were able to successfully perform all but the structure verification test, due to time and budget constraints. Our confidence in the structure of our model would therefore be

$$sc = 20 + 35 + 15 + 20 = 90\%. \quad (11)$$

9.3. Data Validation

The input data necessary to this model are PIN inputs, Withdrawal amounts, and corresponding account balances. The data needs to be validated relate to the rates at which PINs are correctly entered, and the rates at which accounts

may be overdrawn. Since the model and data are simple, we can assume that all data were validated with all techniques, giving us $dc = 100\%$.

9.4. Overall Confidence

Our overall confidence in this simulation model assuming that the behavioral validation is given by Equation 10 and the structural validation is given by Equation 11, is

$$mc = 0.5 * 85.7 + 0.3 * 90 + 0.2 * 100 = 89.85. \quad (12)$$

This confidence level should be interpreted as high confidence in the model, although it also makes it clear that we are not yet completely confident. For a simple model such as the ATM, 100% confidence may reasonably be expected before the model's use. However, most models are significantly more complicated, and thus a confidence level less than 100% will often be sufficient for judging the model as adequate.

10. CONCLUSIONS

In this paper we have presented an approach for computing confidence in DEVS model validity. Prior to this paper, we are unaware of any attempt to quantify the level of validation in a DEVS model. In our approach, confidence in a model increases with each successful application of validation technique, and decreases with each unsuccessful or applicable, yet unused one. Although Balci has previously argued that confidence need only be discussed qualitatively, we propose that a quantified level of confidence can both guide our choice in model and aid us in convincing those who did not build the model of its accuracy. Additionally, a quantified model confidence can increase the standardization of validation.

This approach builds on our previously proposed metric of validation coverage for agent-based and discrete-event simulation models, sharing the idea of quantifying validation by focusing on behaviors. In addition to behaviors, we now calculate model confidence in terms of structure and data as well, and utilize the experimental frame in DEVS. Behavior validation should be utilized to replicate known system behavior, and to test its ability to predict unknown behavior. Our current confidence calculation does not differentiate between these two needs, although we propose including it in our future work along with additional conceptual model validation.

Many extensions to DEVS exist, such as Cell-DEVS. The approach for quantifying confidence levels can be applied to these DEVS extensions by only extending the types of expected behaviors to validate.

In large models it is generally understood that 100% validation is impossible due to time or budget constraints, or a lack of required data. Our framework not only provides a mechanism for calculating confidence in the model based on validation performed, but it also provides a mechanism for making

tough decisions about what validation will be most beneficial on the model to maximize confidence given constraints.

In future work we will examine how to calculate confidence such that not all applicable validation techniques must be applied to have full confidence in a particular behavior.

REFERENCES

- Balci, O. (2010). Golden rules of verification, validation, testing, and certification of modeling and simulation applications. *SCS Modeling & Simulation Magazine*.
- Balci, O. (2012, July). A life cycle for modeling and simulation. *Simulation: Transactions of the Society for Modeling and Simulation* 88(7), 870–883.
- Bergero, F. and E. Kofman (2010). Powerdevs: a tool for hybrid system modeling and real-time simulation. *Simulation*.
- Chillarege, R., I. Bhandari, J. Chaar, M. Halliday, D. Moebus, B. Ray, and M. Wong (1993, Nov). Orthogonal defect classification—a concept for in-process measurements. *IEEE Transactions on Software Engineering* 18(11), 943–956.
- Cho, H. and Y. Cho (1997). Devs-c++ reference guide.
- Elele, J. N., D. H. Hall, A. R. Farid, D. J. Turner, M. E. Davis, and D. R. Keyser (2014). Lessons learned from independent verification and validation of models and simulation (wip). In *SpringSim TMS DEVS*.
- Forrester, J. and P. Senge (1980). Tests for building confidence in system dynamics models. In *TIMS Studies in the Management Sciences*, Volume 14, pp. 209–228. North-Holland.
- Hollmann, D. A., M. Cristia, and C. Frydman (2014). A family of simulation criteria to guide devs models validation rigorously, systematically and semi-automatically. *Simulation Modelling Practice and Theory*.
- Labiche, Y. and G. Wainer (2005). Towards the verification and validation of devs models. In *Proceedings of the First Open International Conference on Modeling and Simulation*.
- Li, X., H. Vangheluwe, Y. Lei, H. Song, and W. Wang (2011). A testing framework for devs formalism implementations. In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, TMS-DEVS '11, San Diego, CA, USA, pp. 183–188. Society for Computer Simulation International.
- Olsen, M. and M. Raunak (2013, Dec). A framework for simulation validation coverage. In *Proceedings of the 2013 Winter Simulation Conference*, Washington, D.C.
- Raunak, M. and M. Olsen (2014a, Dec). Quantifying validation of discrete event simulation models. In *Proceedings of the 2014 Winter Simulation Conference*.
- Raunak, M. and M. Olsen (2014b, Dec). A survey of validation in health care simulation studies. In *Proceedings of the 2014 Winter Simulation Conference*.
- Sargent, R. (2005). Verifying and validating simulation models. In *Proceedings of the 2005 Winter Simulation Conference*.
- Sargent, R. G. (2010). Verification and validation of simulation models. In *Proceedings of the 2010 Winter Simulation Conference*.
- Tolk, A., B. Heath, M. Ihrig, H. Padilla, E. Page, E. D. Suarez, C. Szabo, P. Weirich, and L. Yilmaz (2013). Epistemology of modeling and simulation. In *Proceedings of the 2013 Winter Simulation Conference*, pp. 1152–1166.
- Traore, M. K. and A. Muzy (2006). Capturing the dual relationship between simulation models and their context. *Simulation Modelling Practice and Theory* 14, 126–142.
- Wainer, G. (2002). Cd++: A toolkit to develop devs models. *Software - Practice and Experience* 32, 1261–1306.
- Wainer, G., C. Frydman, and N. Giambiasi (1997). An environment to simulate cellular devs models. In *SCS European Multiconference on Simulation*.
- Wainer, G. A. Cd++ sample models. http://www.sce.carleton.ca/faculty/wainer/wbgraf/doku.php?id=model_samples:start. Last Accessed: 11-09-14.
- Wainer, G. A., K. Al-Zoubi, D. R. Hill, S. Mittal, and J. Martin (2011). An introduction to devs standardization. In G. Wainer and P. Mosterman (Eds.), *Discrete-Event Modeling and Simulation: Theory and Applications*. CRC Press.
- Wang, Z. (2011). Towards a measurement tool for verification and validation of simulation models. In *Proceedings of the 2011 Winter Simulation Conference*.
- Wang, Z. (2013). Selecting verification and validation techniques for simulation projects: A planning and tailoring strategy. In *Proceedings of the 2013 Winter Simulation Conference*.
- Zeigler, B., T. Kim, and H. Praehofer (2000). *Theory of Modeling and Simulation*. Academic Press.
- Zengin, A. and M. M. Ozturk (2012). Formal verification and validation with devs-suite: {OSPF} case study. *Simulation Modelling Practice and Theory* 29(0), 193 – 206.