A FRAMEWORK FOR SIMULATION VALIDATION COVERAGE

Megan Olsen
Mohammad Raunak

Department of Computer Science
Loyola University Maryland
Baltimore, MD 21210, USA

ABSTRACT

Although verification and validation have been studied for modeling and simulation for many decades, we do not yet have a quantitative measure of the level of validation performed on a simulation model. Validation is especially important as it determines whether or not the results from the simulation model can be trusted and used to make statements about the studied system. We propose a validation coverage metric to quantify the validation performed on a simulation model based on the possible validation that could be performed on it. This metric takes into account the aspects of the simulation model that should be validated. To show how such a metric could be utilized, we propose a version of the metric specific to agent-based models, and analyze three example models. We find that the coverage metric can be used to quantify validation on a variety of simulation models.

1 INTRODUCTION

Creating a model and simulation is a well-established and effective approach to analyze or propose changes to a system. Complex systems are widely studied using the tools of Modeling and Simulation (M&S), including natural systems such as population mobility or cancer cell growth (Abbott 2002); systems that are yet to be built such as missile defense or nuclear reactors (Ender et al. 2010); or systems for which making changes in the real-life/runtime environment to study their impact is very expensive or dangerous, such as the process of patient care in a hospital emergency department (Raunak et al. 2009) or automotive control (Ray, Cleaveland, and Shelton 2009). The techniques proposed in the field of modeling and simulation are no longer solely used by computer scientists and engineers, but are now utilized by scientists, researchers, and professionals in many other domains. Since modeling is becoming ubiquitous, a common mechanism to clearly assess the level of validation is necessary.

A crucial step of any simulation-based study, whether the System Under Simulation (SUS) is mechanical/electrical, natural, or theoretical, is to ensure that the simulation model (a) is internally consistent with no known errors (verification) and (b) mimics the SUS's behavior to a level of confidence necessary for making the model useful for its intended application (validation). Without proper verification and validation, predictions cannot be made about the SUS based on the simulation results. Simulation verification and validation (V&V) have been studied for decades, primarily to determine their process and principles (Sargent 2011; Balci 2010; Kleijnen 1995) as well as to develop different V&V techniques (Sargent 2010; Balci 2010). Since the computerized simulation model is a software system, it benefits from many years of software verification research (Bertolino 2007).

There is currently a gap between the validation recommendations of researchers and the actions of the rapidly growing population of simulation practitioners. Although there are simulations that are adequately validated, this is not true for all. Without a metric to capture and communicate the level of validation performed on a simulation model in comparison with the amount that could be performed, it is difficult

to ascertain an appropriate level of confidence in decisions made about the original system based on the findings of a simulation model.

This paper represents the first known attempt to quantify the level of validation performed on a simulation model. Our goal is to develop a metric measuring the extent to which validation activities have been performed on a simulation model, as well as guidelines for applying such a metric. We call this metric *validation coverage*. Validation coverage measures how well *validation aspect elements* have been validated in the system. A validation aspect element is an element of the model that should be validated against the SUS; for agent-based models, an example could be the rate of birth or death of an agent. This set includes both emergent behaviors and input data.

We first define guidelines to enable the creation of these aspect elements for any simulation model. They are developed as a set of aspects that each contain aspect element categories defining the types of elements that belong in each aspect. Using this categorization, a simulation practitioner will develop their set of aspect elements, and determine which validation techniques can be used on each. The validation coverage will define how well a simulation's aspects are validated with these techniques, where each aspect is weighted by its importance in the system. These aspect elements could be developed by a domain expert instead of the model or simulation creator.

In this paper we propose a generic coverage criterion that can be applied to all simulation model types, such as *discrete*, *continuous*, *monte carlo*, *system dynamics*, *gaming-based*, *agent based modeling*, and *AI(Knowledge)-based* (Balci 2012). We also define aspects and aspect element categories for categorizing elements of agent-based models. We evaluate our proposed coverage criterion with case studies of three different agent-based simulation models. We argue that our coverage metric will motivate researchers and practitioners to validate their simulation models more thoroughly by providing a better way to quantify their validation work, and compare the validation of related models. Thus, this research is providing a first framework for quantifying the answer to the question: how much can one trust the result of a simulation model as indicative of the SUS behavior?

2 RELATED WORK

It is generally well accepted that it is difficult to validate large-scale simulation models sufficiently to trust their results (Sargent et al. 2000). In many cases, the cost of trying to achieve complete validation is neither practical nor worthwhile (Shannon 1975). Sargent suggests solving this problem by focusing on smaller simulations in combination to represent the larger simulation (Sargent et al. 2000). However, not all systems can be broken into smaller simulations, so we must have some way to determine how well validation has been achieved.

More than 75 verification, validation, and testing (VV&T) techniques have been presented in the Handbook of Simulation (Banks 1998). These techniques have been categorized into four different groups: *informal*, *static*, *dynamic*, and *formal* (Balci 1998). Informal techniques such as review, inspection, visualization, and face validation are some of the most commonly used approaches for validating simulation models. Static techniques are primarily based upon software static verification, and some are useful for analyzing operational validity of the simulation model. Dynamic validation techniques are extensions of more general software verification activities. Thus the dynamic techniques are primarily useful for verification (checking accuracy between the conceptual and simulation model) of simulation code and are usually not directly applicable for model validation. Dynamic techniques that are specifically used for validation purposes, e.g. statistical techniques, rely heavily on the data available from the SUS (Sokolowski and Banks 2010; Sargent 2010). Formal VV&T techniques are based on mathematical proof of correctness (e.g., logical inference, inductive assertions, etc.) and have limited applicability in operational or behavioral validation.

Balci draws a well defined parallel between a software development process life cycle (Pressman 2010) and simulation development process (Balci 2010; Balci 2012). He proposed that Verification and Validation (V&V) as well as Quality Assurance (QA) activities are an integral part of each of the development

phases of a simulation system (Balci 2012). However, there is no mechanism presented to measure the level of validation performed on a model. Depending on the time of application, the VV&T techniques applied to a simulation system are classified into four groups: validation of the conceptual model to ensure that acceptable assumptions have been made (Conceptual Model Validation); verifying that the simulation software is correct (Computerized Model Verification); validating that the software's behavior matches what would be expected in the SUS (Operational Validation); and ensuring that the necessary data is available and correct (Data Validation) (Sargent 2010). The work presented in this paper focuses on operational validation and data validation. We have considered some of the well established operational validation techniques identified by Sargent including: animation, result validation through comparison to other models, degenerate tests, event validity, extreme condition test, historical data validation, historical methods, internal validity, multistage validation, operational graphics, parameter variability (sensitivity analysis), predictive validation, and trace-based validation (Sargent 2010). We have also considered the most commonly used data validation technique: statistical matching through some goodness-of-fit test (Banks 1998). Our research presents a framework describing how these techniques can be applied systematically on elements of a simulation model and how their application or lack thereof can lead to measuring the level of validation performed on a simulation system.

Simulation researchers have studied additional validation approaches for different types of simulation models. Birta proposed a creation of a validation knowledge base (VKB), captured as a set of relationships between input and output variables of a simulation model (Birta and Özmizrak 1996). Kleijnen presents different statistical techniques to be used for simulation model validation based on the available data (Kleijnen 1999). Reynolds developed a framework based on morphological analysis for effectively using expert analysis of complex systems simulation (Reynolds 2010). Murphy showed the applicability of metamorphic testing in verifying simulation systems with potential for using this technique in model validation (Murphy et al. 2011).

We are unaware of any research on the measurement or quantification of the validation effort, neither for general use, nor for specific types of simulation modeling. That is the focus of this paper. We propose a validation coverage criterion to help researchers and practitioners quantify their validation effort for a simulation model. The concept of coverage criteria is related to the software engineering principle of covering different coding related elements, such as statement coverage, branch coverage, or path coverage (Miller and Maloney 1963). The coverage metric will help us define how much validation is necessary and actually achieved on a particular simulation model.

We note that the act of validating a simulation model cannot be divorced from the available data because the choice of techniques must be based on the available real-life data (Sargent et al. 2000). Data are needed not just for validating the model, but also during the creation of the conceptual model, and for running experiments. However, the data for running experiments is usually not considered during data validation (Sargent 2010). The validation coverage metric discussed in this paper considers data validation as part of the overall validation activity.

3 APPROACH

3.1 Validation Process

A simulation model is an abstraction of the real world system, which we have termed SUS. The purpose of simulation validation is to ensure that the modeled system represents the behavior of the SUS closely enough for the intended purpose of using the simulation. Our primary goal is to develop a metric measuring the extent to which validation activities have been performed on a simulation model, as well as the process for applying such a metric. We call this metric *validation coverage*. Validation coverage is measured based on how well *validation aspects* have been validated in the system. A validation aspect defines a primary dimension of the simulation. Each aspect contains a set of aspect elements, i.e. the elements of the model that must be validated. For agent-based models, an example of aspect elements could be life cycle events

such as birth rate and death rate of an agent. While running a simulation model, all values and simulation behavior that are provided and observed can be divided into three primary categories:

- the values that are used to create the conceptual model
- the values used to run the simulation
- the values that emerge as a result of simulation operation and are observable or measurable

The first two are examples of *input data* (*id*), which must be validated. Input validation is an important step in validating any simulation system, where the modelers need to ensure that the data used in the simulation are indeed representatives of the SUS, as discussed in section 2.

The last category of values is what we term as *Observable Emergent Information* (*oei*), which is information that emerges as part of the simulation model execution. Other than input validation, what we traditionally term as simulation validation activities are generally focused on ensuring that the *oei* values match the SUS. An *id* or an *oei* is an element of some validation aspect of the simulation model. Thus, as part of measuring the level of validation activities performed on a simulation model, we propose identifying all input data (*id*) and observable emergent information (*oei*) of a simulation system and then measuring how well each of these aspect elements has been validated.

Each aspect is thus defined as a set of aspect elements (*ae*s), i.e. observable emergent information (*oei*s) and input data (*id*) that represent specific details of a simulation, and must be shown to match the real world (Figure 1). We do not assume that every potential aspect is relevant for a given simulation, nor that every aspect category is relevant, but provide the means to determine how well a simulation has been validated based on which aspects and aspect elements are relevant.

The aspect elements are validated using accepted validation techniques, and the entire process is illustrated in Figure 2. We propose that depending on the simulation approach, one can identify a set of aspects related to the modeling approach, including a classification of the relevant *oei*s and *id*s. Different *validation techniques* can be applied to validate these *ae*s. The measurement metric will provide a mechanism for quantifying the level of validation that has been applied on a simulation model. Section 3.2 discusses these ideas more formally to develop and specify the concept of a validation coverage criterion.

3.2 Validation Coverage

As already discussed, our proposed validation coverage concept is aimed at establishing a standard way to measure and communicate how well validation activities were performed on a simulation model. To accomplish this goal, we start by recognizing that there are a number of different approaches one can take to develop a simulation model. Thus we first define a set of simulation modeling approaches:

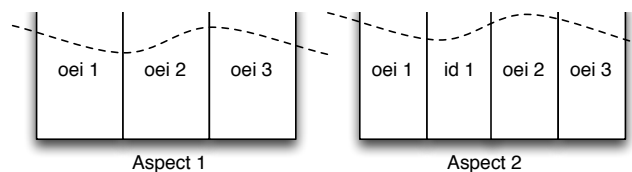$$SMA = \{sma_1, sma_2, ..., sma_n\} \tag{1}$$

Figure 1: An overview of the relationship between aspects, *oei*s, and *id*s. Both *oei*s and *id*s are instances of *ae*s. The dotted line represents the desired level of validation coverage for each.
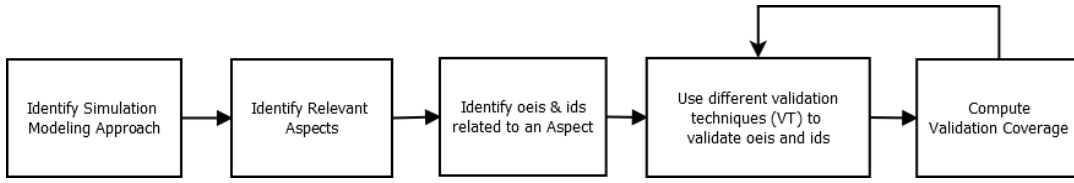
Figure 2: The validation coverage process. This process must be performed for each simulation model for which validation coverage is applied. However, we propose guidelines for each step and a framework for agent-based models that can be adapted for additional simulation modeling approaches.

where sma_i represent a specific modeling approach (e.g., agent-based, monte-carlo, etc). For each of the sma_i we will determine what validation techniques can be utilized. Let us define VT as the set of all potential validation techniques:

$$VT = \{vt_1, vt_2, ..., vt_n\} \tag{2}$$

where vt_i is a particular simulation validation technique or data validation technique. The simulation validation techniques include the validation techniques listed in section 2.

We need to determine the aspects of the simulation model that must be validated, and the applicable validation techniques for that aspect based on the relation between aspect elements and validation techniques. We define VA as the set of all possible validation aspects for all simulation modeling approaches:

$$VA = \{va_1, va_2, va_3, ..., va_n\} \tag{3}$$

For any particular simulation approach, certain aspects are relevant. Thus we define $VA(sma_k) \subseteq VA$ to be the set of validation aspects that are potentially meaningful for a simulation modeling approach sma_k. Each individual simulation model within sma_k will be based upon a subset of these $VA(sma_k)$.

As a reminder, when validating a simulation model, it is the aes that are being validated against the SUS. Our approach recognizes that each ae (oei or id) can potentially be validated and can be grouped categorically by aspects of different SMAs. Thus, a particular aspect va_k is actually a set:

$$va_k = \{oei^1_{va_k}, oei^2_{va_k}, ..., oei^n_{va_k}, id^1_{va_k}, ..., id^m_{va_k}\} \tag{4}$$

Let us denote the set of validation techniques that can be used to validate a particular $oei^j_{va_k}$ as $VT(oei^j_{va_k})$, and the techniques that can validate a particular $id^j_{va_k}$ as $VT(id^j_{va_k})$. We can now define $VAL(oei^j_{va_k})$ as the validation level for a particular $oei^j_{va_k}$ and $VAL(id^j_{va_k})$ as the validation level for a particular $id^j_{va_k}$. VAL is the percent of $VT(oei^j_{va_k})$ or $VT(id^j_{va_k})$ that have been used to validate that ae.

Using the above definitions we now quantify validation activities performed on a simulation model by aspect and then for the entire model. As we described in section 3.1, this process involves first determining the aes by aspect, and then determining the validation techniques that are relevant for each ae in a specific aspect and taking into account what fraction of those techniques have been utilized for validation for each ae, as already defined. Next we must define the validation percentage for a particular aspect va_k as

$$VP_{va_k} = \frac{\sum_{j=0}^{n} VAL(oei^j_{va_k}) + \sum_{j=0}^{m} VAL(id^j_{va_k})}{|va_k|} \tag{5}$$

where $|va_k| = m + n$, and $0 \leq VP_{va_k} \leq 1$. Finally, we define Validation Coverage VC for a particular simulation model of type sma_k based on the validation percentage achieved for each of its aspects:

$$VC = \frac{\sum_{i=0}^{z} w_i * VP_{va_i}}{\sum_{i=0}^{z} w_i} \tag{6}$$

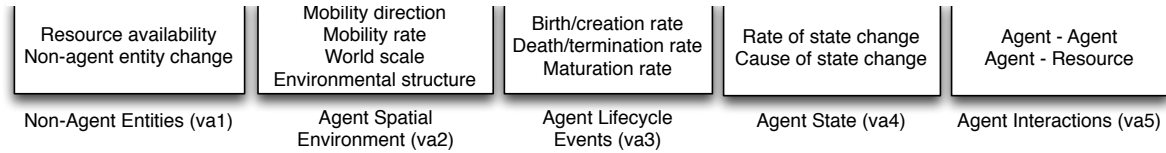| Resource availability Non-agent entity change	Mobility direction Mobility rate World scale Environmental structure	Birth/creation rate Death/termination rate Maturation rate	Rate of state change Cause of state change	Agent - Agent Agent - Resource
Non-Agent Entities (va1)	Agent Spatial Environment (va2)	Agent Lifecycle Events (va3)	Agent State (va4)	Agent Interactions (va5)

Figure 3: The potential aspects and aspect element categories in agent-based simulations.

where $w \in [0,1]$ is an importance weighting for each $va_i \in VA(sma_k)$, and $z = |VA(sma_k)|$.

4 APPLICATION OF VALIDATION COVERAGE

To evaluate the applicability of our approach, we have applied the proposed validation coverage framework to the simulation modeling approach (*SMA*) known as Agent Based Modeling (*ABM*). For agent based models we have identified five aspects (Figure 3) that can accommodate all *oei*s and *id*s related to *ABM*:

1. *Non-agent entities*: Agent based models often include entities that are non-agents. Some of these non-agents can be considered resources (entities that are contended-for and used by agents) in the model. The validation of the creation, destruction, and availability of such resource entities will fall under this aspect.
2. *Agent spatial environment*: Many agent-based models have some form of space, although it may not be the traditional 2D or 3D grid. Elements of the model that deal directly with the spatial representation belong in this aspect. Where the agents are also mobile, elements regarding their mobility direction, mobility rate, and the model environment's scale in relation to the SUS also belong here.
3. *Agent life-cycle events*: This aspect will validate such events as construction (birth), destruction (death), and maturation (change from one phase to another) of agents.
4. *Agent state changes*: Agents usually also have non life-cycle related states. Those states and their changes belong to this aspect.
5. *Agent Interactions*: Agents may interact either with other agents or with non-agent entities in the model. Both types of interactions fall under this aspect. Any resulting state change or resource change would be placed in the appropriate aspect (Agent state and Non-agent entities, for instance).

The aspect element categories shown in Figure 3 denote the types of *ae*s that fit within each aspect. These aspect element categories serve as a guideline for determining a model's *oei*s and *id*s, as well as further defining each aspect. There may be multiple *oei*s or *id*s for each aspect element category for a particular simulation model, and not all aspect element categories or even aspects may be relevant for a model.

These aspects and aspect element categories are broad enough to capture many different dimensions of agent-based models. To demonstrate its usage, we analyze three different simulation models with these aspects for validating ABMs: a model simulating interaction behavior of Tasmanian devils, a model simulating gossip propagation in a social network, and a model of tumor growth. Each of our case studies represent a different use of agent-based modeling and a different type of real world problem, has varying levels of available data for validation, and needs to utilize different subsets of the aspects described above. For each case study we will briefly discuss the simulation model, followed by the relevant aspects and aspect element categories, and then the application of validation coverage. These case studies will demonstrate that the proposed aspects and aspect element categories can represent most ABMs.

4.1 Case Study: Tasmanian Devils

4.1.1 Model Description

For our first case study we examine an agent-based predator-prey model of Tasmanian devils, a carnivorous marsupial found only in the Tasmanian island of Australia. In this simulation, devils move in a two-dimensional grid either randomly or toward prey if the prey's scent is within their neighborhood. As devils are primarily scavengers, prey appear in random locations at a fixed rate to mimic roadkill and carcasses. Preys emit a scent gradient that decays over time, which devils follow when hungry. Devil hunger decreases as they eat, and is of a fixed level each day.

The purpose of this simulation is to study the rate of interactions between devils, as in the wild they suffer from a deadly cancer known as Devil Facial Tumor Disease. It is transmittable from devil to devil whenever they bite each other, which primarily occurs during eating. The simulation therefore logs how frequently devils meet while eating, to help determine the overall rate of disease transmission through biting (Fay et al. 2011).

4.1.2 Validation Coverage

This simulation model fits easily within our framework for validation coverage. Figure 4 shows the relevant aspects and *ae*s. This simulation does not model any of the agents' life cycle events. Devils neither procreate, nor die, nor evolve in the simulated period, as the model focuses on a six-month period of time not during mating season. Thus none of the *ae*s related to agent life cycle aspect is relevant for this model. However, the other four aspects are relevant and sufficient to represent all elements of the simulation that should be validated.

To determine validation coverage, we first analyze how well each *oei* and *id* is validated. To do that we will need to identify the set of validation techniques applicable for a particular *ae*. We propose that $oei^1_{va_4}$ *rate of hunger change* from the *Agent State* aspect can be validated using historical data validation and sensitivity analysis. Let us assume it was validated using both of these validation techniques. Since there are two methods in $VT(oei^1_{va_4})$, and both have been applied, we compute the validation level $VAL(oei^1_{va_4})$ to be 1, giving maximum coverage for $oei^1_{va_4}$. As this was the only *ae* in this aspect, we compute the coverage of the va_4 as 1 or 100%.

The next aspect we validate is *agent spatial environment*, va_2. We again propose that both $oie^1_{va_2}$ and $oie^2_{va_2}$ should be validated using historical data when such data is available, as well as animation. Let us suppose that devil movement ($oie^2_{va_2}$) is validated using historical data validation and animation, giving 100% coverage. The validation of *world size* ($id^1_{va_2}$) is related, as the two are dependent; it is validated to match the SUS in scale, giving it 100% coverage. During our validation of *following scent* ($oie^1_{va_2}$), however, we only used the validation technique of animation. Since there is one relevant validation technique that has been unused, we calculate $VAL(oei^1_{va_2})$ to only have 50% coverage. Therefore, the *agent spatial environment* aspect validation VP_{va_2} is 83.3% (Equation 7).

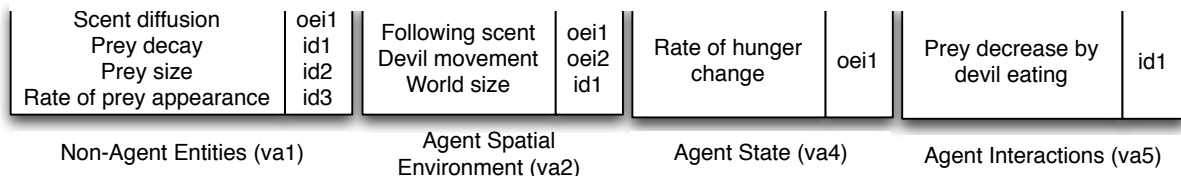| Scent diffusion Prey decay Prey size Rate of prey appearance	oei1 id1 id2 id3	Following scent Devil movement World size	oei1 oei2 id1	Rate of hunger change	oei1	Prey decrease by devil eating	id1
Non-Agent Entities (va1)		Agent Spatial Environment (va2)		Agent State (va4)		Agent Interactions (va5)	

Figure 4: Relevant aspects and *ae*s for the Tasmanian devils simulation model.

$$VP_{va_2} = \frac{0.50 + 1 + 1}{3} = 0.833 \tag{7}$$

Finally, the *non-agent entities* aspect, va_1, must be analyzed. All *ae*s of va_1 except *scent diffusion* are validated as *id*s. In some simulation models, these input data (*id*s) are often traces from the SUS.

Let us suppose all *id*s ($id_{va_1}^1, id_{va_1}^2, id_{va_1}^3$) have been validated using the 'statistical match' validation with real life data from SUS. Let us also suppose that the $oei_{va_1}^1$ of *scent diffusion* is validated using animation, which was the only relevant validation technique. Thus, VP_{va_1} is 100% as all of its elements have been validated with all relevant validation techniques.

The final aspect *agent interactions* (va_5) only has one *id* which is also validated by statistical match, giving $VP_{va_5} = 1$.

The above calculations will now allow us to compute the validation coverage for the devil simulation. For the sake of simplicity of description, we use equal weight for all aspects, giving a coverage percentage of 95.8% (Equation 8).

$$VC(devil) = \frac{1 + 0.833 + 1 + 1}{4} = 0.958 \tag{8}$$

4.2 Case Study: Gossip Propagation

4.2.1 Model Description

Our second case study is an agent-based model of gossip propagation in a social network, which is naturally modeled as a graph. Each node in the graph represents a person capable of spreading gossip about some target node. Some nodes are liars, which mutates the bit string that is passed as gossip. The gossip propagates from node to node based on the strength of connection between those nodes and how long the gossip has been propagated so far. Gossip that has been propagating for some time has a weaker strength than fresh gossip, which will cause nodes to no longer wish to spread the information. Each node must determine what to believe about a particular piece of gossip, and the simulation compares the overall belief of the network based on various decision strategies. After a piece of gossip is no longer spreading through the network, nodes acquire a fitness score which denotes how close their beliefs were to the truth. The purpose of this simulation is to study human gossip decision rules and gossip propagation (Laidre et al. 2013).

4.2.2 Validation Coverage

As shown in Figure 5, we posit that only four aspects are relevant for this model. The spatial environment in this model is defined as the connections between agents that form a network, and thus the environmental structure *ae* is the agent network, including liar location. The world size is defined by the number of agents and liars. The gossip being shared and modified is a non-agent entity in the system, and thus its fidelity decrease is a non-agent entity change . The primary agent state of note is the agent's current state of belief, which is potentially modified each time it makes a decision about received gossip. The agent memory determines how much previous gossip agent can recall. Finally, agents interact with each other through gossip in two ways: modifying the gossip before sharing it, and choosing who to share the gossip with. Based on our analysis, we find that the proposed aspects adequately cover all elements of this simulation model that require validation.

Following the general process we proposed in section 3.1 to compute validation coverage, we will analyze coverage for each aspect by taking into account its *oei*s and *id*s one by one. Let us first consider va_1, the *non-agent entities* aspect. Let us say that $VT(oei_{va_1}^1)$ (validation techniques for *gossip fidelity decrease*) has four elements: 1) animation, 2) degenerate tests, 3) extreme condition test, and 4) sensitivity analysis through parameter variability. Let us assume that we validated $oei_{va_1}^1$ using the first two of these techniques
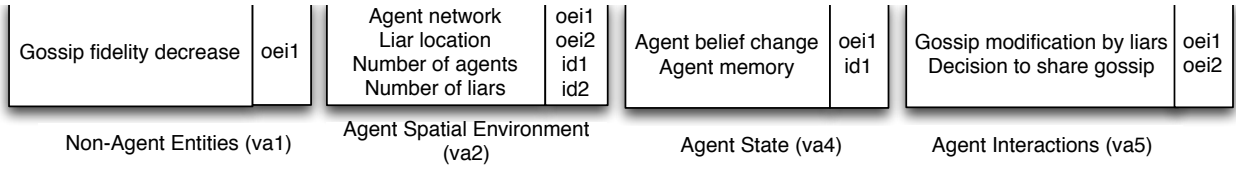
Gossip fidelity decrease	oei1		Agent network Liar location Number of agents Number of liars	oei1 oei2 id1 id2		Agent belief change Agent memory	oei1 id1		Gossip modification by liars Decision to share gossip	oei1 oei2
Non-Agent Entities (va1)			Agent Spatial Environment (va2)			Agent State (va4)			Agent Interactions (va5)	

Figure 5: Relevant aspects and *ae*s for the gossip propagation model. Although the same four aspects are relevant as in the devils model, the *ae*s are different.

('animation' and 'degenerate tests'). Thus, $VAL(oei^1_{va_1}) = 2/4 = 0.5$, that is the coverage achieved for $oei^1_{va_1}$ is 50%. We can now compute $VP(va_1)$ simply as $0.5/1 = 0.5$.

Next we examine the *agent spatial environment* aspect. Let's say that the environmental structure *oei* $oei^1_{va_2}$ can be validated with animation and face validity, and that we have applied them both. Thus we achieve a $VAL(oei^1_{va_2})$ of 1, or 100% coverage. Let us also suppose that $oei^2_{va_2}$ was not validated at all. There are also two *id*s, the *number of agents* and the *number of liars*. Let us assume that both are validated to match the SUS. Thus the overall coverage for validation aspect, va_2 is seen in Equation 9.

$$VP_{va_2} = \frac{1+0+1+1}{4} = 0.75 \tag{9}$$

Next, we must validate the *agent state* aspect va_4. Let us suppose that we validated $oei^1_{va_4}$ with all validation techniques in the set $VT(oei^1_{va_4})$, but we were not able to validate $id^1_{va_4}$, resulting in VP_{va_4} being 0.5. Finally, for *agent interactions* aspect va_5 we assume that $VT(oei^1_{va_5})$ includes historic data validation and sensitivity analysis, and we have validated $oei^1_{va_5}$ *gossip modification by liars* using both these techniques. Let us suppose we validated $oei^2_{va_5}$, *decision to share gossip*, using historical data, out of four validation techniques in $VT(oei^2_{va_5})$: 1) historic data, 2) degenerate test, 3) extreme condition test, and 4) sensitivity analysis. With only one of four techniques used, $VAL(oei^2_{va_5}) = 0.25$, and therefore $VP_{va_5} = 0.625$.

With the values we have computed so far, we can now compute the total validation coverage for the gossip model. From the discussion of the domain experts, it was deemed that va_5 has more weight than the other aspects, which is reflected in the computation in Equation 10 that results in coverage of 59.6%.

$$VC(gossip) = \frac{(.8)(.5)+(.8)(0.75)+(.8)(0.5)+(1)(0.625)}{3.4} = 0.596 \tag{10}$$

4.3 Case Study: Multi-scale Cancer Model

Our third case study is the most complicated model of the three: an agent-based multi-scale model of cancer growth at the molecular, cellular, and tissue level (Olsen and Siegelmann 2013). This model represents tissue cells, including both normal cells and cancer cells. Due to the interactions between normal cells, cancer cells, and the environment, there are many aspects within this simulation that must be validated for the results to be trustworthy. We show these aspects in Figure 6. As the main difference between this simulation and the prior two case studies is the number of *oei*s and *id*s, we will not discuss the coverage but instead focus on showing that this model fits within our proposed agent-based aspects and aspect element categories.

The cells in this simulation have the potential to create new cells (proliferate), die (apoptosis), mutate, and repair mutations. Cancer cells have decided advantages over normal cells, with higher proliferation, lower apoptosis, and the ability to squeeze more cells into a given space. Additionally, the simulation examines angiogenesis, which is the creation of new blood vessels. All cells need enough nutrient to

Oxygen depletion	oei1
Oxygen diffusion	oei2
VEGF diffusion	oei3
Vessel size	oei4
Spatial chemical diffusion	oei5

Non-Agent Entities (va1)

Normal cell mobility direction	oei1
Normal cell mobility rate	id1
Cancer cell movement	oei2
Cellular space requirements	id2
Tumor growth pattern	oei3
Vessel system structure	oei4
Vessel sprout growth rate	oei5
Vessel sprout growth direction	oei6

Agent Spatial Environment (va2)

Normal proliferation rate	id1
Normal apoptosis rate	id2
Cancer proliferation rate	id3
Cancer apoptosis rate	id4

Lifecycle Events (va3)

Genetic mutations	oei1
Genetic repair	id1
Hypoxic/Necrotic/Normal	oei2

Agent State (va4)

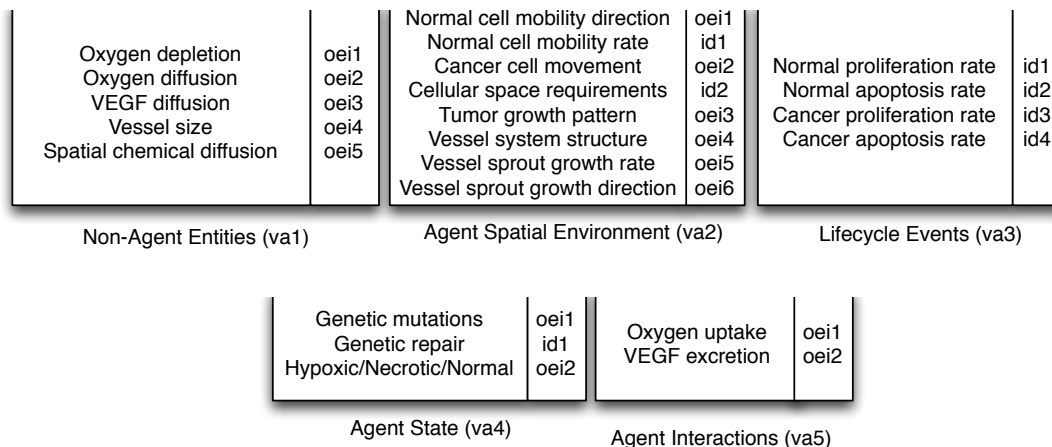| Oxygen uptake | oei1 |
| VEGF excretion | oei2 |

Agent Interactions (va5)

Figure 6: Relevant agent-based modeling aspects for the cancer growth model. All five aspects are relevant.

survive, otherwise they become hypoxic and cannot proliferate. Hypoxic cancer cells are given the ability to diffuse VEGF, a chemical, which will stimulate new blood vessel growth when it reaches a blood vessel. These new blood vessels growth along the gradient of VEGF, and if they form a loop with another sprout can become a viable blood vessel and begin diffusing nutrients. Usually this change increases the ability of cancer cells to grow in an environment that could not otherwise support their numbers.

For applying the agent aspects to this simulation model, we see that the agents are the cells, and the environmental structure includes all chemicals diffused as well as the blood vessels. Additionally, the agent state is primarily related to the genes that control overall behavior, and whether or not the cell has enough oxygen to function properly. This model has lifecycle events, as both normal and cancerous cells are able to proliferate under certain conditions. Thus, despite the complexity of the simulation model, the agent aspects can describe the emergent behaviors and input data that should be validated.

5 DISCUSSION

This work demonstrates the usefulness of a simple framework to capture the level of validation performed on a simulation model. We have proposed a basis for creating a standard validation coverage metric, have proposed a set of aspects specific to agent-based models, and have shown its applicability in measuring the level of validation performed on three agent-based simulation models. The metric determines the validation percentage for each aspect, as defined by how well its observable emergent behaviors and input data have been validated. Each aspect is not assumed to be of equal importance, but are instead weighted in the final coverage metric calculation.

The aspects proposed for agent-based models are intended to be general purpose and applicable to all agent-based models. To demonstrate applicability and their use in coverage, we examined three simulation models as case studies. The case studies demonstrated how the *oei*s and *id*s are determined for an aspect based on the aspect element categories, and then how the validation performed on them determines the validation coverage. For the Tasmanian devils simulation we demonstrated how a simple agent-based model may acquire high validation coverage. We then used a networked model of gossip propagation to show how a different type of agent-based modeling may fit within the proposed agent-based modeling aspects, and how coverage may be calculated for such a model. Finally, we used a substantially more complex agent-based model of cancer growth to show how a large and complex model can also utilize the proposed agent-based aspects and aspect element categories for coverage.

Note that a desired level of coverage may also be determined for a simulation model. It seems likely that only a range may be proposed in a general framework to define the acceptable level of validation, and this range is likely to be different for different simulation approaches. Although we have assumed a coverage goal of 100% while showing the applicability and usefulness of our framework, it is unlikely that every simulation will have the required data available to attain complete coverage.

There are a few potential weaknesses of this approach. The simulation practitioner may be able to take advantage of the system by misrepresenting either the aspect weights, what validation has been performed, or the desired coverage level. However, we feel that this issue is already at hand without a framework for quantifying validation. Therefore, our system enables this issue to become more apparent and more easily caught than with current qualitative analyses of validation level. Additionally, our aspect categorization enables a clearer description of what must be validated, and may simplify the process of ensuring all elements that should be validated have been considered. With our framework, the conversation can be started in a quantitative manner about how well a simulation is validated versus how well it could be validated in an ideal world.

Another weakness may be pointed out that we currently have a loose recommendation on how to identify different aspect elements to be validated using the guidelines provided by aspect element categories, which in turn are used to quantify validation coverage. We believe that this categorization is loose enough to enable all models to fit within it, but tight enough to guide the creation of sufficient elements. As a first step in a problem that is not currently being addressed in the community, our framework provides a new approach to determining how well the many different dimensions of a SUS are represented in a simulation model. We have provided aspects and aspect element categories for agent based models. We strongly encourage the community to try this approach to quantify their agent-based model validation effort.

This framework is an important step toward formalizing a more general framework for establishing a comprehensive validation coverage metric. Thus there are multiple directions in which this work will continue to grow. One is to add structure to the recommendations about how to identify *oei*s and *id*s. Another is to define which validation techniques are applicable for each aspect element category, as not all validation techniques would be applicable for a particular simulation modeling approach or an aspect element. Moreover, there may be prioritization of validation techniques for each aspect element categories or aspect elements, as future work may show that some are more important than others. Another future direction involves creation of aspects and aspect element categories for other simulation modeling approaches such as discrete event simulations. Over time the creation of these sets will develop a framework that can be readily applied to any simulation model as long as the practitioner understands the aspects of the SUS. Our proposed framework has the potential to revolutionize the analysis of validation for simulation models across all fields, and enable a robust discussion of validation coverage.

REFERENCES

Abbott, R. 2002. "Cancersim: A computer-based simulation of hanahan and weinberg's hallmarks of cancer". Master's thesis, University of New Mexico.

Balci, O. 1998. *Verification, Validation, and Testing*. John Wiley & Sons.

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.

Banks, J. 1998. *Handbook of Simulation:Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons.

Bertolino, A. 2007. "Software Testing Research: Achievements, Challenges, Dreams". In *Future of Software Engineering*.

Birta, L. G., and F. N. Özmizrak. 1996, Jan. "A knowledge-based approach for the validation of simulation models: the foundation". *ACM Transactions on Modeling and Computer Simulation* 6 (1): 76–98.

Ender, T., R. Leurck, B. Weaver, P. Miceli, W. Blair, and P. West. 2010. "Systems-of-Systems Analysis of Ballistic Missile Defense Architecture Effectiveness Through Surrogate Modeling and Simulation". *System Journal* 42:156–166.

Fay, G., M. Olsen, J. Gran, A. Johnson, V. Weinberger, and O. Carja. 2011. "Agent-based model of Tasmanian Devils examines spread of Devil Facial Tumor Disease due to Road Construction". In *Proceedings of the International Conference on Complex Systems*.

Kleijnen, J. 1995. "Verification and validation of simulation models". *European Journal of Operations Research* 82:145–162.

Kleijnen, J. 1999, Dec. "Validation of Models, statistical techniques and data availability". In *Proceedings of the 1999 Winter Simulation Conference*.

Laidre, M., A. Lamb, S. Shultz, and M. Olsen. 2013. "Making sense of information in noisy networks: human communication, gossip, and distortion". *Journal of Theoretical Biology* 317:152–160.

Miller, J. C., and C. J. Maloney. 1963, February. "Systematic mistake analysis of digital computer programs". *Commun. ACM* 6 (2): 58–63.

Murphy, C., M. Raunak, A. King, S. Chen, C. Imbriano, and G. Kaiser. 2011. "On Effectve Testing of Healthcare Simulation Software". In *Software Engineering in Health Care (SEHC)*. Honolulu.

Olsen, M. M., and H. T. Siegelmann. 2013. "Multiscale Agent-based Model of Tumor Angiogenesis". *Procedia Computer Science* 18 (0): 1016–1025. 2013 International Conference on Computational Science.

Pressman, R. 2010. *Software engineering: a practitioner's approach*. New York: McGraw-Hill.

Raunak, M., L. J. Osterweil, A. Wise, L. A. Clarke, and P. L. Henneman. 2009. "Simulating Patient Flow through an Emergency Department Using Process-Driven Discrete Event Simulation". In *Proceedings of the Software Engineering in Health Care (ICSE 2009)*. Vancouver, Canada.

Ray, A., R. Cleaveland, and C. Shelton. 2009. "alidating Automotive Control Software using Intrumentation-based Verification". In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*.

Reynolds, W. 2010. "Breadth-Depth Triangulation for Validation of Modeling and Simulation of Complex Systems". In *IEEE Intelligecne and Security Informatics Conference, Workshop on Current Issues in Predictive Approaches to Intelligence and Security Analytics (PAISA)*, 190–195.

Sargent, R. 2011. "Verifying and Validating Simulation Models". In *Winter Simulation Conference*. Colorodo.

Sargent, R., P. Glasow, J. Kleijnen, A. Law, I. McGregor, and S. Youngblood. 2000. "Strategic Directions in Verification, Validaiton, and Accreditation Research". In *Proceedings of the 2000 Winter Simulation Conference*, 909–916.

Sargent, R. G. 2010. "Verifying and Validating Simulation Models". In *Proceedings of the 2010 Winter Simulation Conference*.

Shannon, R. 1975. *The Art and Science*. Prentice Hall.

Sokolowski, and Banks. 2010. *Modeling & Simulation Fundamentals*. Wiley.

AUTHOR BIOGRAPHIES

MEGAN OLSEN is an Assistant Professor in the Computer Science Department at Loyola University Maryland. Her research interests are in simulation validation, complex systems, and stigmergy. Much of her work is in simulating biological complex systems, primarily of predator-prey systems and cancer cell growth. She recently received her PhD in Computer Science from the University of Massachusetts Amherst. Her e-mail and web addresses are mmolsen@loyola.edu and http://www.cs.loyola.edu/~olsen.

MOHAMMAD RAUNAK is an Assistant Professor in the Computer Science Department at Loyola University Maryland. His research interests are software process and simulation; verification, validation, and analysis of complex software and simulation systems. His email and web addresses are raunak@loyola.edu and http://www.cs.loyola.edu/~raunak.