Verification and Validation of Agent-based Scientific Simulation Models

Xiaorong Xiang Ryan Kennedy Gregory Madey
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556
xxiang1@nd.edu, rkenned1@nd.edu, gmadey@nd.edu
Steve Cabaniss
Department of Chemistry
University of New Mexico

Abstract

Most formalized model verification and validation techniques come from industrial and system engineering for discrete-event system simulations. These techniques are widely used in computational science. The agent-based modeling approach is different from discrete event modeling approaches largely used in industrial and system engineering in many aspects. Since the agent-based modeling approach has recently become an attractive and efficient way for modeling large-scale complex systems, there are few formalized validation methodologies existing for agent-based model validation. In this paper, we design, develop, adapt, and apply various existing verification and validation techniques to an agent-based scientific model and investigate the sufficiency and importance of these techniques for the validation of agent-based models.

Keywords: Agent-based modeling, Model verification, Model validation. Scientific simulation models

1 Introduction

One of the main valuable aspects of a simulation study is to explain and understand real world phenomena that are costly to perform in the laboratory or difficult to collect in field experiments. A successful simulation that is able to produce a sufficiently credible solution can be used for prediction. Since it is impossible (performance concern) and unnecessary (including elements that do not have much effect on the system) to construct a simulation model that represents all the detail and behavior of the real system, some assumptions must be made about the system to construct a simulation model. Therefore, a simulation model is an abstract representation of a physical system and intended to enhance our ability to understand, predict, or control the behavior of the system. However, the abstractions and assumptions introduce inaccuracies to the simulation model. One of the important tasks for a simulation study becomes determining how accurate a simulation model is with respect to the real system. The difficulty for the validation task is that there is no a universal approach. There are many principles and techniques of model verification and validation that have been presented. However, it is difficult and time consuming to use all possible techniques for validating every model that is developed. Modelers are depended on to choose appropriate techniques that can assure the acceptable accuracy and credibility of their model. The objective of our work is exploring the new validation schemes for agent-based scientific models based on the existing verification, validation, and calibration techniques presented in various literatures.

The organization of this paper is as follows. We first introduce a few classical verification and validation techniques that are used in industrial and system engineering fields in Section 2. The development of the NOM simulation model, a test project for our agent verification and validation research, is illustrated in Section 3. In Section 4, we demonstrate the verification and validation processes for the NOM model. Summary and proposed works are described in Section 5.

2 Model Verification and Validation

Model verification is a process that determines whether the programming implementation of the abstract or conceptual model is correct. This process includes debugging software, looking for incorrect implementation of conceptual models, and verifying the calculations. Model validation is a process that determines whether the conceptual model is a reasonably accurate representation of the real world [11] and the output of simulations is consistent with real world output. Calibration is an iterative process adjusting the unmeasured or poorly characterized experimental parameters in the programming implementation. The purpose of the calibration process is to improve the agreement with experimental data. Model verification, validation, and calibration are generally considered to be parts of the model development process. Model validation is achieved through the calibration of the model until model accuracy is acceptable.

2.1 Model Verification and Validation Techniques

Balci [2] [3] presents 75 validation, verification and testing techniques that are largely used in validating the models of manufacturing, engineering, and business processes. These models are mostly discrete event simulation models and are intended to control the total costs of the system, problem solving, and decision-making. There is a taxonomy that classifies the techniques into four primary categories: informal, formal, static, and dynamic. Some of these techniques come from the software engineering discipline that is not our focus in our verification and validation process. We next present some relevant techniques.

Face validity is asking the domain experts whether the model behaves reasonably and makes subjective judgments on whether a model is sufficiently accurate. There are two ways to allow the experts to give the correct judgments easily:

1) *Animation* is the graphical display of the behavior of the model over time. Some simulation software, such as Swarm and Repast, have built-in features for animation and can even track the individual's properties while the simulation is running.

2) *Graphical Representation* is representing the model's output data (mean, distribution, and time series of a variable) with various graphs. These graphs can help in making subjective judgments.

Model developers also use *Animation* and *Graphical Representation* for code verification in the model implementation process. Face validity is the first step of the three-step approach formulated by Naylor and Finger [14] and widely followed in industrial and systems engineering.

Tracing is a technique similar to *Animation*. The behavior of entities in the model is followed to determine if the logic of the model is correct. Although tracing is extremely useful in isolating the strange behavior of the model, it causes considerable additional processing overhead.

Internal Validity involves comparing the results of several replications of a stochastic simulation model using different random seeds. If the random seeds used for the random number generators cause the inconsistency (large variability) of the sample points, the model is questionable either in the programming model or the conceptual model.

Historical Data Validation is used when historical data exists (or if data is collected on a system for building or testing the model). Part of the data (the training sets) is used to build the model and the remaining data (test sets) is used to determine if the model behaves as the system does.

Parameter Variability - Sensitivity Analysis is a validation technique where one changes the values of the input and internal parameters of a model to determine the effect upon the model and its output. The same relationship should occur in the model as in the real system. Those parameters that are sensitive, i.e., cause significant changes in the model's behavior, should be made sufficiently accurate prior to using the model.

Predictive Validation is used to compare the model's prediction with actual system behavior. The system data may come from an operational system or specific experiments. For instance, the data may come from a laboratory or from field experiments.

To perform *Turing Tests*, experts of a system are given both real system and model outputs and asked if they can discriminate the real system output and the model outputs.

2.2 Model-to-Model Comparison

Model-to-model comparison, also known as back-to-back testing or docking, is a technique that compares various results of the simulation model to results of other models [1]. Different conceptual models can represent a real world phenomenon. For instance, the cell behaviors in the biological process, such as the cell growth and division process, can be modeled using either an agent-based modeling approach [22] or a cellular potts model (CPM) [9]. A real world phenomenon can be represented by one conceptual model, but different research groups or individuals can implement conceptual models using different programming languages or different simulation toolkits. These computational models may be run on different platforms. The output comparison between different simulation models with the same input data is conducted using this technique. Differences in the outputs reveal problems with model accuracy. When the compared model is a valid model, the agreement between these two models can infer the validity of the model. When the compared model has not been validated, the comparison between these two models can support the code verification process if these two models are implemented from the same conceptual model. One of the big advantages for the model-to-model comparison technique is for the non-observable system, i.e., all data required for model validation cannot be collected from the real-world system.

2.3 Statistical Tests

Compared to subjective validation techniques, such as face validation and turing test, statistical analysis is a quantitative method. Incorporating statistical analysis techniques in the validation process can significantly increase the credibility of the model. Model validation is conducted by using the statistical techniques to compare the model output data with the corresponding system or with the output data of other models when the model is run with the same input data. Some discussions on applying the statistical techniques to the validation process can be found in Sanchez [18] and Kleijnen [10].

The first step for starting the statistical test is determining a set of appropriate output measures that can answer user questions. After a set of output measures is collected, various statistical techniques can be applied to complete the validation process. Time series, means, variances, and aggregations of each output measures can be presented as a set of graphs for model development, face validation, and Turing tests. Confidence intervals (c.i.) and hypothesis tests can be used in the comparison of parameters, distributions, and time series of output data for each set of experimental conditions. These statistical tests can help model developers to determine if the model's behavior has an acceptable range of accuracy.

3 Case Study: Agent-based Modeling for NOM Evolution

Natural organic matter (NOM) is a polydisperse mixture of molecules with different structures, compositions, functional group distributions, molecular weights, and reactivities, that forms primarily as the breakdown product of animal and plant debris. NOM is ubiquitous in terrestrial ecosystems and has been widely reported in marine environments. Its structure, chemical composition, and reactivity vary both spatially and temporally. NOM plays a crucial role in ecological and bio-geochemical processes such as the evolution of soils, the transport of pollutants, and the global bio-geochemical cycling of elements [5]. NOM is a primary food source to microorganisms, and it can act as a natural "sunblock," attenuating potentially damaging UV radiation [20] in lakes and streams. While passing through an ecosystem, NOM may be acted upon and potentially altered by a wide array of processes, such as microbial biodegradation, adsorption to mineral surfaces, redox reactions, photochemical processes, and aggregation or coagulation. The evolution of NOM over space and time from precursor molecules to eventual mineralization (primarily as CO_2) is an important research area in a wide range of disciplines, including biology, geochemistry, ecology, hydrology, soil science, and water resources. Given the widespread abundance and importance of NOM to many hydro- bio- geochemical processes, predictive modeling of its evolution in structure, composition, and reactivity are fundamental to many areas of environmental research.

Because of the complex nature of NOM, and the multitude of possible reactions it may undergo in natural environments, we have only a limited knowledge of the detailed mechanisms by which it forms from precursor molecules, or how its structure, composition, and reactivity evolve over space and time. Perhaps most notably, the fact that NOM is a polydisperse mixture of molecules which themselves have complex structures means that it defies characterization by established analytical methods. Previous models of NOM formation and evolution have been important for predicting certain types of reactions or interactions, but they have not been able to describe both the quantitative aspects of organic carbon transfer and the semi-quantitative or qualitative aspects of NOM structure and functional heterogeneity. Predicting NOM interactions and their consequences to other environmental processes requires a clear understanding both of how a single NOM component behaves and how the entire NOM mixture at a given site evolves over space and time.

Agent-based modeling (ABM) is a method used to track the actions of multiple agents that can be defined as objects with some type of autonomous behavior [12] [7]. By using the ABM approach, the higher-level behaviors of a system, called "emergent behaviors," can be discovered without being explicitly coded into the simulation. The technique of building and using agent-based models is an essential tool for understanding complex systems and is increasingly applied to biological, ecological, and sociological issues.

3.1 The Conceptual Model

Based on the observational and experimental studies on the behavior of NOM in the environment, we designed a conceptual agent-based stochastic model. In this model, NOM, microorganisms, and their environment form a complex system. The evolution of NOM over discrete time and space from precursor molecules (such as cellulose, lignin, and protein) to eventual mineralization involves various molecular transformations. These transformations involve chemical reactions, adsorption, aggregation and physical transport in soil, ground, or surface waters. NOM is presented as a large number of discrete molecules with varying chemical and physical properties. Individual molecules, modeled as individual agents, are given a set of simple rules on how to move through soil pores and how to interact with each other.

Agent: In the NOM complex system, the agents are NOM molecules. These NOM molecules are hypothesized to be derived from macromolecules such as proteins, polynucleotides, cellulose, lignin, or small organic molecules such as phospholipids, sugars, and amino acids. It is impractical to define precise molecular representations that involve descriptions of atomic location, electron density maps, and force field constants given our goal of modeling thousands or even up to one million individual molecules. We therefore need an intermediate level molecular representation, which is more specific than simply "percent carbon" but less detailed than a precise molecular connectivity map. Also, the representations of structures should be detailed enough to illustrate the heterogeneity of the NOM. The data used for constructing representations of the NOM molecules includes the following components:

- Elemental formula, i.e. the number of **C, H, O, N, S, and P** atoms in the molecule. The molecular weight **MW** can be easily calculated from the elemental formula.
- Functional group count, e.g. **carboxylic acid, alcohol, and ester** groups. There are a total of 19 possible functional groups in our model for each molecular structure.
- A record of the molecular "origin," i.e. the initial molecule, its starting position, and its time of entry into the system. This allows for the calculation of separate "turnover times" and apparent ages for individual molecules or fractions. The x- and y-coordinates of a 2-dimensional lattice represent the location of each molecule in the simulation.

Behaviors: In the environment, NOM can move with percolating water down a soil column and into the groundwater, and interact with other molecules and their environment. As molecules are transported by water through soil pores, they can sorpt to and desorpt from mineral surfaces. Sorption behavior of molecules is not a transformation of molecular structure; however, it will affect the probability of other reactions by changing the environment of a given molecule. Chemical reactions result in structural changes in the molecule, such as the addition of groups to a NOM molecule. New molecules can be generated from the predecessor molecules and those predecessor molecules may leave the system. Twelve types of chemical reactions, including first-order and second-order chemical reactions are modeled as described in Table 1. Others will be added as needed.

These twelve chemical reactions are separated into four categories:

- *First order reactions with a split:* The predecessor molecule **A** is split into two successor molecules **B** and **C**. Molecule **B** occupies the position of molecule **A**, and one of the empty cells nearest molecule **B** is filled with molecule **C**.

Table 1. Chemical Reactions in the Conceptual Model

Reaction Name	**Reaction Type**
Ester condensation	Second order
Ester hydrolysis	First order with split
Amide hydrolysis	First order with split
Microbial uptake	First order with molecule disappear
Dehydration	First order with split
Strong C=C oxidation	First order with split (50% of the time)
Mild C=C oxidation	First order without split
Alcohol(C-O-H) oxidation	First order without split
Aldehyde C=O oxidation	First order without split
Decarboxylation	First order without split
Hydration	First order without split
Aldol condensation	Second order

- *First order reactions without a split:* The transformation only changes the structure of the predecessor molecule **A**.
- *First order reactions with the disappearance of a molecule:* The predecessor molecule **A** disappears from the system. This reaction occurs when the molecule is small enough that a microorganism like fungi or bacteria can envelop it.
- *Second order reactions:* Two molecules **A** and **B** are combined to form a new molecule **C**. The new molecule **C** replaces molecule **A** while the other predecessor molecule **B** disappears from the system.

Space: In this conceptual model, the individuals are associated with a location in geometrical space and can move around their environment. The geometrical space is described as a discrete 2D grid space represented by integer values. Each molecule can occupy one grid cell. Each grid cell can host multiple molecules up to a certain threshold.

Reaction Probabilities: The probability for each reaction type is expressed in terms of intrinsic and extrinsic factors. Intrinsic factors are derived from the molecular structure including the number of functional groups and any other structural factors. Extrinsic factors arising from the environment include concentrations of inorganic chemical species, light intensity, availability of surfaces, presence of microorganisms, presence and concentration of extracellular enzymes, and the presence and reactivity of other NOM molecules. The intrinsic and extrinsic factors are combined in probabilistic functions.

Molecular Properties: The reactivity of the resulting NOM over time can be predicted based on the distribu-

Table 2. Molecular properties

Name	Comments
Number of Molecules	Number of molecules in the system. It changes as molecules condense, split, or are consumed.
MWn	The number-average molecular weight.
MWw	The weight-average molecular weight.
Z average	The average charge on each molecule at pH 7.
Element Mass	Weight of C, O, H, etc. in the system.
Percent of Element	The weight percentages of each element.
Reactions (1...n)	Number of reactions that occur for each type of reaction.

tions of molecular properties. Some molecular properties of NOM can be calculated and predicted. These properties are quantities that can be calculated from the elemental composition and functional group data. They represent a measurable quantity that can be used as a predictor for an environmental function. They are also useful and interesting both as part of scientific inquiry and for the calibration and verification of our conceptual model and simulation. Some properties are easy to calculate, such as molecular weight, molecular charge, and charge density. On the other hand, some involve non-trivial computations. In addition, some properties are exact calculations, (e.g., molecular weight) while others must be estimated by empirical relationships that will introduce additional uncertainty (e.g., pK_a). Table 2 summarizes some molecular properties that can be used for answering user questions.

Simulation Process: The conceptual model is a stochastic synthesis model of NOM evolution. It serves as a design framework for the computer simulations described later in this paper. In a stochastic process, the state of the system is represented by a set of values with a certain probability distribution, such that the evolution of the system is dependent on a series of probabilistic discrete events. At each time step, for each molecule, a uniform random number is generated. This number determines whether a reaction will occur, and if one does occur, it determines the reaction type. After a reaction takes place, the attributes for the current molecule are updated and reaction probabilities are recalculated. The molecule structure is changed to affect the outcome of the reaction and a new probability table entry is added for newly formed molecules, if there are any.

3.2 Implementations

There are five implementations derived from the conceptual model.

AlphaStep is a reference implementation that is coded in Delphi 6 and runs under Windows XP. It is a demonstration of the comprehensive conceptual model that doesn't have web or collaboration features. AlphaStep simulates a variety of chemical and biological transformations, but does not simulate any type of transport and does not represent the spatial properties of NOM. Hence, it represents a batch or closed system without porous media. AlphaStep is intended as a stand-alone application to allow ecologists, geochemists, and environmental scientists to explore possible routes of NOM transformation. It is available for download from http://www.nd.edu/ñom/software.

The other four implementations are coded using the Java programming language (Sun JDK 1.4.2) and Swarm [21] [13] and Repast[16] software. Swarm is a software package for simulating complex systems that was developed at the Santa Fe Institute. It is a set of libraries that facilitates the implementation of agent-based models. RePast is a Swarm-like agent-based simulation toolkit written in Java. It was developed by the University of Chicago and Argonne National Laboratory. Swarm and Repast provide versatile random number generators and distributions that are essential to stochastic computer simulations. Also, these toolkits provide high-level visualization capabilities. A control panel allows a running model to be stopped, restarted, or executed step by step. They also allow modelers to visualize the current state of the running model and probe every object in the model. An animation window shows the location of individual agents in a space. Line graphs and histograms are used to illustrate changes in collections of model objects that occur during the simulation. Additionally, simulation data, such as population size over time, can be reported periodically as the model executes.

The **Flow** and **No-Flow** implementations respectively model laboratory column and batch experiments. **Flow** implementations simulate the column experiments. Molecules are continually added from the top of the grid at a constant rate to simulate their mobility with water flow including advection and dispersion. These molecules move along with the water flow, experience physical and chemical reactions, and leave the system from the bottom of the grid. **No-Flow** implementations simulate the batch experiment with all molecules added to the system at the beginning and then diffusing throughout the medium. There are no molecules entering into or leaving the system. Furthermore, according to the scientific interests of different reactions, sorption behaviors and chemical reactions of molecules are implemented separately.

4 Validation of NOM simulation model

Verification and validation (V&V) are processes used to increase confidence in simulations. Verification concerns solving “the problem right” while validation concerns solving “the right problem .” Although neither process guarantees absolute confidence, we used numerous V&V techniques on the NOM simulation. We describe these techniques using an adapted version of Sargent’s V&V process shown in Figure 1 [19].

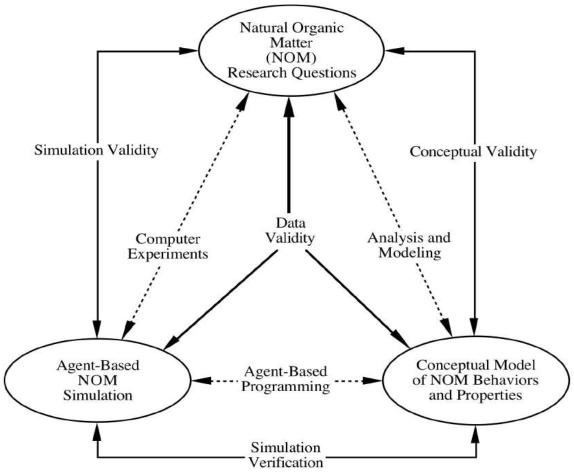

Figure 1. Verification and Validation Processes of the Agent-based Stochastic Model

The modeling process starts by identifying research questions of interest. Through analysis and modeling, we developed a conceptual NOM model that includes the features relevant to the questions. The conceptual model is based on theory and domain knowledge from environmental chemistry, soil science, and geomicrobiology. This theoretical knowledge which guided the model development includes the following as examples: 1) the heterogeneity of NOM molecules, 2) the important NOM interactions with mineral surfaces such as adsorption, hemi-micelle formations, acid or complexion dissolution, and reductive dissolution, 3) NOM interaction with pollutants, 4) relationships between NOM adsorption to mineral surfaces and the molecular weight of the NOM molecules, and 5) probabilistic reaction kinetics based on elemental composition and the nature of functional groups in the molecules The incorporation of such theory and domain knowledge provides us initial face validity, i.e., the logic of the conceptual model appears to domain experts to include appropriate mechanisms and properties of the research problem. Six scientists on the project - two biologists, a chemist, a geomicrobiologist, and two soil scientists - evaluated the conceptual model for face validity.

Once the conceptual model achieved its initial validation, coding of the agent-based simulation took place. In this step, verification methods such as code walk-through, trace analysis, input-output testing, pattern tests, boundary testing, code debugging, and calculation verification were used to verify the correctness of the simulation. Visualization of the behavior of the simulation is another useful technique used for simulation validation [8]. A snapshot of an animated visualization of the flow of molecules through a soil column depicting the adsorption and desorption of the molecules to mineral surfaces in the simulation is shown in Figure 2. In addition to the color-coding of molecules by molecular weight, the adsorbed or desorbed states are depicted by solid and hollow circles respectively. A corresponding animated graph of the molecular weight distribution shows how the molecular weight distribution shifts with time: initially favoring lower weight molecules in the early stages of the simulation and gradually shifting to larger molecular weights as the simulated time passes. These same behaviors were observed in laboratory experiments, increasing the confidence in the simulation. Figure

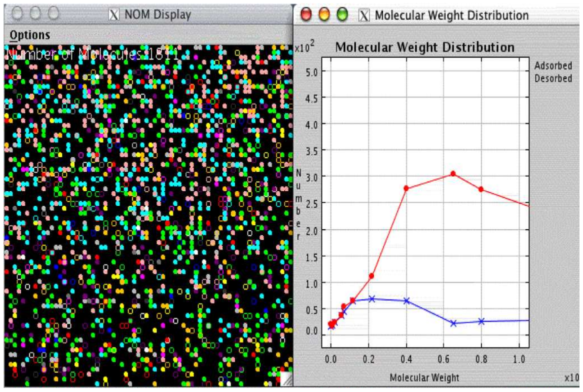

Figure 2. Example of a snapshot from a FlowSorption simulation. Left side displays molecules moving through the column: adsorbed (closed circles) or in solution (open circles). In a colored version, colors represent different MW intervals. Right side displays the corresponding MW distribution of adsorbed (higher peak) versus in-solution molecules. Adsorption is preferential for intermediate-to high MW components. In this example, the model input was molecular weight distribution of the NOM with adsorption controlled by molecular weight. Specific chemical reactions were not explicitly considered.

2 is a screen shot of a **FlowSorption** simulation used by the authors for verification and validation of the simulation.

4.1 Internal Validity

A simulation model that uses random seeds must have statistical integrity in that independent simulations with the same input data should have similar results. If the simulation model produced large variabilities because of the random seeds, there would be a considerable problem with it. To test this, we performed 450 simulations with our NOM simulator, each with a different random seed. We chose the total number of molecules in the system after the simulation had completed as a point of comparison. We found that our simulations produced the expected normal curve upon analysis of the data. Figure 4 shows the histogram for this data. By verifying the independency of the random seeds in the NOM simulator, we were able to conclude that it is statistically robust in terms of repeatability. Further statistical analysis needs to be performed to verify how reliably our simulator conforms to a normal distribution.

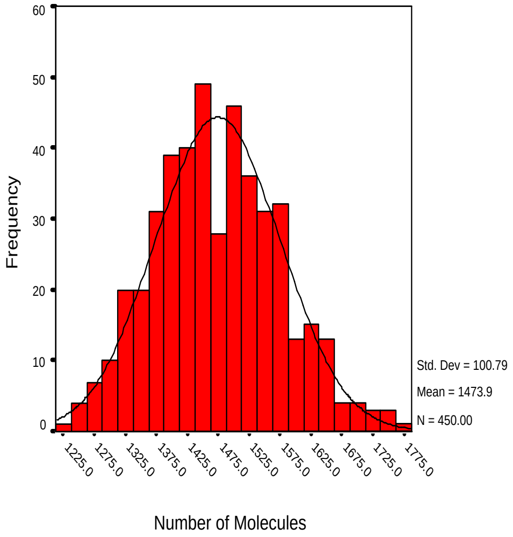

Figure 3. Histogram showing the distributions of the number of molecules after 1000 simulated hours. This histogram is based on 450 individual simulations, each with a different random seed.

Table 3. Differences of features in Alpha step and No-flow Reaction implementations

Features	Alpha Step	No-flow Reaction
Programming Language	Delphi 6, Pascal	Java (Sun JDK 1.4.2)
Platforms	Windows	Red hat Linux cluster
Running mode	Standalone	Web based, Standalone
Simulation packages	None	Swarm, Repast libraries
Initial population	Actual number of different molecules	Distribution of different molecules
Animation	None	Yes
Spatial representation	None	2D grid
Second order reaction	Random pick one from list	Choose the nearest neighbor
First order with split	Add to list	Find empty cell nearby

4.2 Alpha Step vs. No-Flow Reaction implementations

As we stated before, Alpha step and No-flow Reaction are independent implementations that follow most features of the conceptual model we have described in previous sections. We summarize some differences between these two implementations in Table 3. It is valuable to conduct the model-to-model comparison. At the beginning of this docking process, we experienced big differences in the outputs of these two models. The main reason that caused the results to be different was that the techniques used to calculate reaction probabilities were not the same.

Since these two implementations are stochastic simulations, it is not enough to compare both implementations with one simulation run. We conducted 25 replications using different random seeds for both implementations given the same initial condition. There are many molecular variables, as shown in Table 2, that we can choose as measures for this comparison purpose. Some of them can be derived from others using simple mathematic calculations. We chose **Number of Molecules**, **MWn**, **MWw**, **Carbon Mass**, and **Carbon percentage** for comparison. Also, the main purpose for NOM simulations is to mimic the NOM evolution over time, as the comparison of properties at one time point is not enough to capture the trend of the evolution. The comparison of ensemble averages of each measured over time for two implementations is shown as Figure 4. These ensemble averages are computed from 25 replications using different random seeds.

Despite the good agreement between these two implementations visually, the operational validity requires statistical significance tests. Some Goodness-of-Fit Tests (such as Chi-Square testing, Kolmogorov-Smirnov test) and factorial ANOVA test need to be conducted to assure that an observed difference are statistically different.

5 Conclusion and Future Work

In this paper, we applied various techniques for validation of an agent-based stochastic simulation model. We achieved the preliminary validation of the model by conducting a model-to-model comparison.

Verification and validation techniques used in industrial and systems engineering fields (such as queuing models) are also discussed in the computational science field [6] [15] [17] [4]. We observe that discrete event modeling approach is the most closely approach as agent-based modeling approach with differences in several aspects:

1) Entities in the system. Agent-based modeling approach focus on systems that contain large number of heterogeneous objects (people, business units, animals, vehicles) and time events ordering or other kind of individual behavior associated with them. Discrete event modeling approach tends to model relatively homogenenious entities with variability captured in the service time of the system's servers.

2) Spatial representation. In most of agent-based models, the individuals are associated with a location in a geometrical space. This geometrical space can be described as either continuous space (represented by partial differential equations) or as discrete grid space (represented by integer values). In industrial and systems engineering field, the spatial representation is typically a network of servers and queues.

3) Discovery of global behavior. Industrial and systems engineering models know the behavior of a system in advance. This is either because a system already exists from which observational data can be collected, or the system is being designed for desired behavior (e.g., performance level). In a complex adaptive system, the emerging behavior cannot be known in advance. Whenever there is strange behavior, it is hard to distinguish if it is a program bug or the real world phenomena.

4) Degree of interactivity and communication among the entities. Agents in the agent-based models have many activities. They can interact with each other and their environment. In the industrial and systems engineering field, entities may only have low interactivity among each other.

We believe that model-to-model comparison is an efficient validation method for agent-based scientific models. Comparing the output of simulations with empirical data is the next step that we intend to do.

Although agent-based modeling approach is becoming more and more attractive and has been adopted for modeling various complex systems, it is still lacking mathematical foundations as well as a formalized validation methodology. The formalized verification and validation techniques are

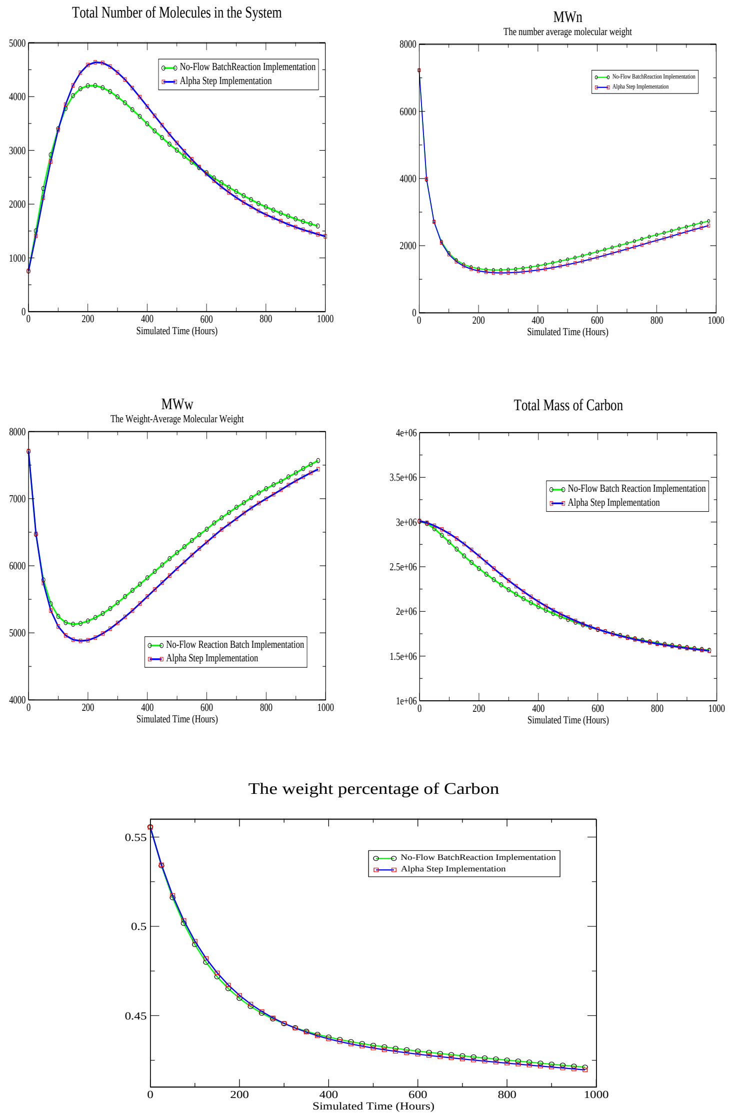

Figure 4. The figure shows a comparison of five molecular properties for two implementations. Since they are stochastic simulations, the output is ensemble average of 25 replications for each implementation.

derived from the industrial and system engineering. These systems are structured systems with well-understood rules and the behavior of the system is transparent to modelers. Complex systems that are suitable for using an agent-based modeling approach are unstructured with incomplete understanding of behavior by modelers. Validation methods that are less important for one modeling approach maybe more important and useful for another one. The objective of our work is systematically evaluate a subset of these techniques and answer the following questions: (1) Are the classical verification and validation techniques sufficient for validating agent-based models? (2) Which subset of these techniques that are more cost-effective for validating of agent-based models?

References

[1] R. Axtell, R. Axelrod, J. M. Epstein, and M. Cohen. Aligning simulation models: A case study and results. *Computational and Mathematical Organization Theory*, 1996.

[2] O. Balci. *Handbook of Simulation: Priciples, Methodology, Advances, Applications, and Practice*, chapter Verification, Validation and Testing. John Wiley & Sons, New York, 1998.

[3] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol. *Discrete-event system simulation*. Industrial and systems engineering. Prentice Hall, third edition edition, 2001.

[4] R. F. Benjamin. An experimenter's perspective on validating codes and models with experiments having shock-accelerated fluid interfaces. *Computing in Science & Engineering*, 06(5), 2004.

[5] S. E. Cabaniss. Modeling and stochastic simulation of nom reactions, working paper. http://www.nd.edu/ nom/Papers/WorkingPapers.pdf, July 2002.

[6] A. Calder, J. Dursi, B. Fryxell, T. Plewa, G. Weirs, T. Dupont, H. Robey, J. Kane, B. Remington, F. Timmes, G. Dimonte, J. Hayes, M. Zingale, P. Drake, P. Ricker, J. Stone, and K. Olson. Validating astrophysical simulation codes. *Computing in Science & Engineering*, 06(5), 2004.

[7] V. Grimm. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modeling*, 115:129–148, 1999.

[8] V. Grimm. Visual debugging: A way of analyzing, understanding and communication bottom-up simulation models in ecology. *Natural Resource Modeling*, 15(1):23–38, Spring 2002.

[9] J. A. Izaguirre, R. Chaturvedi, C. Huang, T. Cickovski, J. Coffland, G. Thomas, G. Forgacs, M. Alber, G. Hentschel, S. A. Newman, and J. A. Glazier. Compucell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, 2004.

[10] J. P. C. Kleijnen. Validation of models: Statistical techniques and data availability. In *Proceddings of the 1999 Winter Simulation Conference*, 1999.

[11] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, second edition, 1991.

[12] J. G. Louis. Agent-based modeling in ethnobiology: A brief introduction from outside. http://www.tiem.utk.edu/ gross, April 2002. Talk to NSF Workshop on Priorities in Ethnobiology.

[13] N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The swarm simulation system: A toolkit for building multi-agent simulations. Technical report, Santa Fe Institute Working Paper 96-06-042, 1996.

[14] T. H. Naylor and J. M. Finger. Verification of computer simulation models. *Management Science*, 2:B92–B101, 1967.

[15] D. W. Pierce. Beyond the means: Validating climate models with higher-order statistics. *Computing in Science & Engineering*, 06(5), 2004.

[16] Repast. http://repast.sourceforge.net/.

[17] P. J. Roache. Building pde codes to be verifiable and validatable. *Computing in Science & Engineering*, 06(5), 2004.

[18] S. M. Sanchez. Abc's of output analysis. In *Proceddings of the 2001 Winter Simulation Conference*, 2001.

[19] R. G. Sargent. Validation and verification of simulation models. In P. A. Farrinton, H. B. Nembhard, D. Sturrock, and G. W. Evans, editors, *Proceedings of the 31st Winter Simulation Conference*, Phoenix, AZ, December 5-8 1999. ACM Press.

[20] N. M. Scully and D. R. S. Lean. The attenuation of ultraviolet radiation in temperate lakes. *Ergeb. Limnol.*, pages 135–144, 1994.

[21] Swarm development group. http://www.swarm.org.

[22] D. Walker, J. Southgate, G. Hill, M. Holcombe, D. Hose, S. Wood, S. M. Neil, and R. Smallwood. The epitheliome: agent-based modelling of the social behaviour of cells. *BioSystems*, (74):89–100, 2004.