

## **A ROBUST EVOLUTIONARY STRATEGY FOR GENERATIVE VALIDATION OF AGENT-BASED MODELS USING ADAPTIVE SIMULATION ENSEMBLES**

Levent Yilmaz  
Guangyu Zou

Auburn University  
Computer Science and Software Engineering  
3116 Shelby Center  
Auburn, AL 36849, USA

Osman Balci

Virginia Tech  
Department of Computer Science  
3160B Torgersen Hall  
Blacksburg, VA 24060, USA

### **ABSTRACT**

Few studies examine distinct characteristics of problems studied by agent-based models and their implications on operational validation. This paper focuses on exploratory and generative modeling perspective advocated by the agent-based modeling paradigm. The significance of robustness is emphasized, and a robust generative validation strategy is proposed for models used in scientific problems in which ambiguity and deep uncertainty pervade. The strategy is predicated on the premise of creative evolutionary systems perspective that enables viewing validation within the scientific method of falsification. That is, the strategy mimics the way scientific knowledge is constructed and validated by groups of scientists within a scientific discipline. The utility and feasibility of the method is demonstrated using a case study.

### **1 INTRODUCTION**

Agent-Based Models (ABMs) are often criticized for relying on informal, subjective, and qualitative validation procedures (Edmonds and Chattoe 2005). Because most ABMs are highly abstract and are built from bottom up, their emergent behavior is often unpredictable. Also, a single authoritative model does not often capture critical aspects of a multidimensional phenomena due to factors excluded during the abstraction process (Schreiber 2002). Furthermore, ABMs are often developed for studying complex adaptive phenomena, which involve uncertainty and ambiguity in terms of their underlying behavioral mechanisms (Yilmaz and Ören 2009). Models that focus on human and social dynamics are especially prone to ambiguity and uncertainty (Bharathy and Silverman 2010).

By developing a model of a complex adaptive system with large number of diverse entities that not only interact with each other, but also adapt and learn through feedback and self-organization, one can gain empirical insight while exploring emergent behavior. The underlying behavioral rules in such complex systems are often unclear; hence, scientists and model developers hypothesize theoretically grounded rules to explore their implications to discover underlying plausible mechanisms that help explain the behavior of such systems (Yilmaz and Hunt 2011). In such problems, there can be one or more valid models that can be considered as plausible. Therefore, while developing a single authoritative model and validating its behavior against theory, system, and/or empirical data is feasible, such a strategy may not be robust in the presence of uncertainty and ambiguity that pervade complex adaptive systems.

Figure 1 depicts three major types of agent-based models used in complex systems simulations. The area marked with *variability* assumes the existence of an authoritative model. Using this model, experiments can be conducted by varying the *experiment space*, which is comprised of parameters that define the system configuration and the context in which the experiment takes place. By selecting parameters, as well as their range and constraints, one can observe and test the consequences of various assumptions for the purpose

of understanding a system or predicting its behavior. Substantial validation research has been conducted and principles of validation are well understood in this domain (Balci 2003, Balci 2004, Yilmaz 2006).

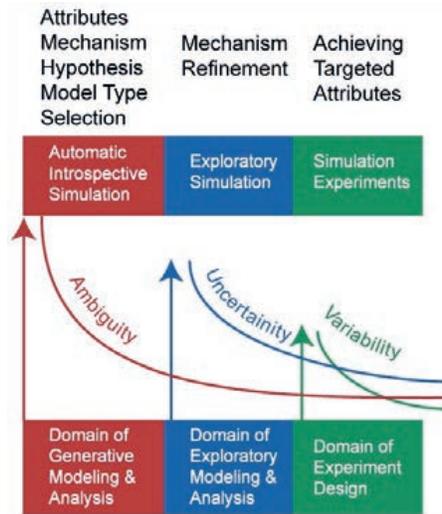


Figure 1: Generative and Exploratory Simulation.

The domain of exploratory modeling focuses on conducting computational experiments in the presence of deep uncertainty (Davis and Bigelow 2000). Seeking strategies that perform reasonably well across a large number of plausible, yet unknown states is especially critical. Similarly, in scientific discovery, scientists seek to discern mechanisms that are capable of explaining as many targeted observable measures as possible. Exploratory modeling and analysis considers both structural and parametric uncertainty. Constraints and structural relations on selected operating principles are varied to refine mechanisms to explore implications of the constraints on the *hypothesis space*. However, in early phases of foresight, there is often lack of clarity about which operating principles are applicable. This is the domain of *generative modeling*, for which agent-based models are widely used (Epstein 2007).

To introduce the Robust Generative Validation (RGV) strategy, we use the metaphor of traditional cycles of scientific advancement, where multiple hypotheses are formulated from a body of knowledge and their testable consequences are evaluated. The proposed work is predicated on the conjecture that creative evolutionary systems (Bentley and Corne 2002) coupled with the principles of complex adaptive systems thinking facilitate reconsidering the use of simulation so that creative discovery of a diverse portfolio of sufficiently valid models can be discovered. In our strategy, generation of models and their validation testing are coupled together, and the feedback from the validation process affects generation of increasingly accurate models. By doing so, we intend to mimic the way scientists collectively construct and validate increasingly precise and accurate knowledge in an incremental and iterative manner.

The rest of the paper is structured as follows. In Section 2, we overview validation and verification of simulation models, including agent-based models. The conceptual basis for RGV is introduced in Section 3. Section 4 demonstrates its applicability using a case study. We conclude in Section 5 by summarizing the method and discussing potential avenues of future research.

## 2 BACKGROUND

A *model* is a representation or abstraction of something such as an entity, a system or an idea. Simulation is the act of experimenting with or exercising a model or a number of models under diverse objectives including problem solving, training, or acquisition. Since a model, by definition, is an abstraction or

approximation, its representation is never expected to be perfect. Therefore, substantiating the sufficient accuracy of a simulation model continues to pose significant technical challenges (Balci 2010).

Verification and Validation are intended to assess the *accuracy* quality characteristic of a simulation model (Balci 2004). *Simulation Model Verification* deals with the assessment of transformational accuracy of the model and addresses the question of Are we creating the model right as intended? *Simulation Model Validation* deals with the assessment of behavioral or representational accuracy of the model and addresses the question of Are we creating the right model? (Balci 2003).

ABM enables conceptualizing systems in terms of interacting autonomous entities, called agents. Complex adaptive systems are comprised of such interdependent, networked, and distributed agents that can adapt (Miller and Page 2007). Agents have perception and social ability to perform goal-directed knowledge processing over time, on behalf of humans or other agents in software and physical environments. The core knowledge processing abilities of agents include: reasoning, motivation, planning, and decision making. The factors that may affect decision making of agents, such as personality, emotions, and cultural backgrounds can also be embedded in agents. Since agent-based modeling advocates a bottom-up view of system representation, the underlying mechanisms and their explanation in terms of agent behavior and interaction rules are crucial. The emergent behavior can be counterintuitive, and such behavior can be construed as novelty and new knowledge, instead of inaccuracy. ABM also provides a framework for tuning the complexity of agents by allowing model developers to refine agents' behavior, interaction rules, ability to learn, and evolve across multiple levels of resolution. As ABMs are increasingly being used in computational science, especially in biomedical, therapeutic (Hunt, Ropella, Lam, Tang, Kim, Engelberg, and Bahaei 2009), and social sciences (Miller and Page 2007), scientists are developing incremental and iterative refinement protocols (Hunt, Ropella, Lam, Tang, Kim, Engelberg, and Bahaei 2009) to discover plausible explanations of scientific phenomena, not simply for prediction, but also for understanding and discovery.

Our focus in this article refers to challenges of validating ABMs involved in mechanism discovery. For such problems, the underlying behavioral rules and interaction mechanisms are often unclear, imprecise, and uncertain. To gain empirical insight into such problems and to be able to generate behavior that mimics expected behavior or theoretical regularities, model development and refinement should synergistically be coupled with evaluation and validation.

### 3 CONCEPTUAL BASIS FOR ROBUST GENERATIVE VALIDATION

In developing a conceptual basis for the proposed *Robust Generative Validation* method, we refer to the strategies used by scientists in generating and validating knowledge. Falsification, originally promoted by Karl Popper, is a scientific method that views knowledge creation and validation as a formal activity, where theories and models that make failed predictions are ruled false and those that make good predictions are provisionally accepted and selectively retained, until new evidence comes along. Scientific method also involves an active process that facilitates construction of knowledge. As such, we consider validation as an incremental and iterative generative (i.e., constructive) search and decision-making process and seek a method to partially automate it.

#### 3.1 Structuring Robust Generative Validation

As advocated in (Klahr and Dunbar 1988), scientists coordinate search in two spaces to construct and find valid models: a *space of experiments* and a *space of hypotheses*. Hypotheses are the tentatively assumed propositions in regard to either operating principles and mechanisms or outcomes with testable consequences. Once generated, hypotheses are evaluated for their plausibility. In this study, we view hypotheses as the mechanisms that are comprised of activities within and interactions between agents. Hypotheses are generated and evaluated as a result of experimentation; in this case, simulation. The experiment space involves selecting both the endogenous and exogenous variables and their values.

Figure 2 presents critical components of a hypothetical interactive, simulation-based generative validation system that explores both spaces. Using initial information about basic building blocks and configuration of system's processes and activities, scientists employ computational models and scenario generators to create ensembles of plausible models. Each scenario consists of a particular choice of (adaptive) mechanism embedded within a single model generated from the hypotheses space and a specific point in the experiment space (parameters and their values), coupled with an expected regularity or outcome. Following initial claims about robust mechanisms generated by the RGV system, scientists can test and revise mechanisms through setting new experimental conditions that invalidate current mechanisms (path b) and then generate and explore alternative promising mechanisms (path c). Furthermore, scientists can interactively inject new targeted attributes (path e) or environmental conditions and search for significantly different mechanisms (models) (path d).

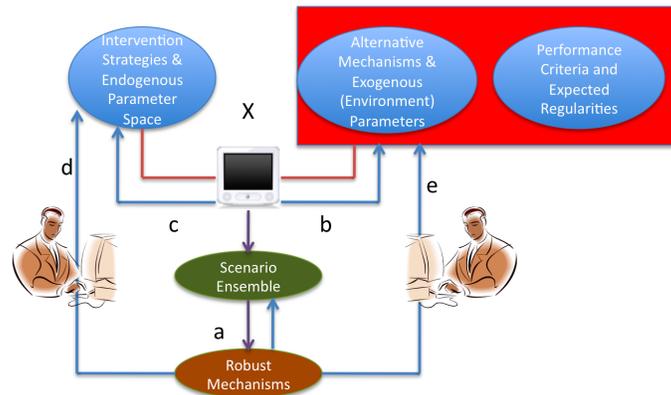


Figure 2: Robust Generative Validation Process.

**Exploratory modeling software** enables model developers to navigate through the large numbers of scenarios required to make up a model portfolio and to formulate rigorous arguments about intervention choices based on these explorations. A **scenario generator** uses the relationship among the variables to create members of scenario ensembles. In contrast to a traditional model that is typically designed to produce a comparatively small number of predictive conclusions, a scenario generator should yield a full range of plausible alternatives. Next, we examine a strategy that has the potential to enable designing such exploratory modeling and creative scenario generation systems.

### 3.2 Designing RGV as a Creative Evolutionary System

To facilitate creative construction of scenarios and explore their consequences within the aforementioned hypotheses and experiment spaces, we design RGV as a *Creative Evolutionary System* (CES) (Bentley and Corne 2002). To formalize RGV, we define the structure of the domain of models as a graph of model ensembles,  $G = (V, R)$ , where  $V$  is the set of nodes, and each node  $v \in V$  denotes an ensemble of models, and  $R$  is the set of relations depicting affinity (e.g., similarity in terms of function and form) between the ensembles. Each ensemble  $E$  refers to a single hypotheses space and has a neighborhood  $N(E)$ , which refers to a connected subgraph of  $G$  containing  $E$ . For our purposes, each ensemble contains a collection of metaobjects, each one of which specifies the schema of a corresponding model. Models within an ensemble follow the structural and behavioral constraints and assumptions underlying the mechanisms hypothesized. Figure 3 depicts the structure of graph of ensembles. The strength of relations (e.g.,  $w(i, k)$  or  $w(k, i)$ ) between ensembles signify the degree of similarity analogues in the source ensemble exhibit with respect to phenotypic attributes of the target ensemble's referent.

In accordance with the evolutionary dynamics used in CES, to evolve model schemas and their metaobjects, we need an encoding scheme. Although the encoding of a schema depends on the purpose of

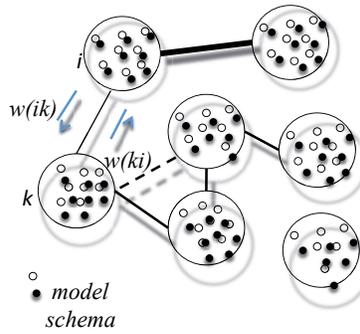


Figure 3: Graph of Model Ensembles.

the study and experimentation space (i.e., factors to vary and their levels) that will be evolved during the process of generative simulation, for simplicity and purpose of demonstration, we may assume that each schema is a binary sequence of length  $n$  from the space  $\{0, 1\}^n$ . We denote the set of schemas with  $S$  and define a population function,  $P$ , as

$$P : V \rightarrow S \tag{1}$$

The population function can be extended via a neighborhood function  $N : V \rightarrow 2^V$ , which returns for a given node  $x \in V$ , all nodes,  $y \in V$  in  $G$ , where  $(x, y) \in R$ . Hence,  $P(N(E))$ , where  $E \in V$ , returns all schemas contained in the neighboring ensembles. One can however also access schemas within a given ensemble by taking neighborhood equal to a specific ensemble. At any given round of generation,  $T$ , the schema population of the neighborhood of an ensemble,  $E$ , is given by  $P(N(E), T)$ . Similarly, the population of a specific ensemble is defined as  $P(\{E\}, T)$ , and the number of schemas in ensemble is  $N = N_{E;T}$ .

### 3.2.1 Evaluation

Given the above specification, we need to define evolutionary aspects of the ensemble. Three major factors are of interest. First, models that exhibit behaviors similar to those targeted must be favored, as they generate sufficiently valid behavior, if a similarity measure value exceeds a prespecified threshold. Second, those models that use divergent mechanisms yet have a sufficient level of validity may facilitate discovery of novel mechanisms. Consequently, they should be retained. Finally, those models that demonstrate success in generating behaviors and features imposed by neighboring ensembles (i.e., hypotheses spaces) should be favored, as they may be able to extend their usefulness and scope. Models that satisfy the constraints of multiple phenotypic attributes will relate to schemas from more than one ensemble: they are expected to have larger impact. The similarity for a given ensemble  $v$  is defined as  $F_a : S \rightarrow [0, 1]$ .  $F_a(s)$ , where  $s \in S$ , returns similarity of the analogue with schema  $s$  within the ensemble  $v$ . For a given schema, its performance is the degree of similarity of corresponding analogue behaviors to targeted phenotypic attributes. The *extent* function,  $F_e : N(E) \times S \rightarrow [0, 1]$ , measures the degree of relevance of the schema with respect to ensembles in  $N(E)$ . The total similarity of schema  $s$  in ensemble  $E$  is the weighted sum of its validity and extent:

$$f(s) = \alpha_a F_a + \alpha_e F_e \tag{2}$$

where  $\alpha_a + \alpha_e = 1$ . Adjustment of these parameters enables examination of alternative population evolution strategies. So doing may suggest effective and efficient methods for discovering analogues that are qualified to mimic selected sets of attributes.

### 3.2.2 Generation

The similarity functions specify how the schemas in the ensemble will be judged. Schema generation uses general schema *transformation operators*. A transformation operator is defined as a function  $t : S^m \rightarrow S$  for some integer  $m$ . The generative simulation system has a collection of operators,  $R = \{t_i, i = 1, 2, \dots, J\}$ , that enable generation of alternative forms and structures. *Elaboration* involves refinement of a schema and is similar to a mutation operation with  $m = 1$ , while *combination* is analogous to a crossover operation, with  $m = 2$ . Each transformation operator,  $t_i$ , is associated with a weight,  $w_i$ , that determines frequency of its application. The generation of new schema involves a stage of schema selection, followed by interactions to create new schemas. At each round, schemas with similarity values less than a predefined threshold are dropped. The remaining population in  $P(\{E\}, t)$  representing ensemble  $E$  is replaced with a new interim population  $\phi$  consisting of schemas in  $P(\{E\}, T)$  with different frequencies. Using the conventional GA fitness proportionate parent selection mechanism, we compute for each schema its probability of selection:  $\frac{f(s)}{\sum_{j \in E} f(j)}$ . The expected number of copies of each schema (metaobject) in the interim population is then given by  $\frac{f(s)}{\bar{f}}$ , where  $\bar{f} = \frac{1}{N} \sum_{j \in E} f(j)$  the average similarity measure of the population. The frequency of each schema is therefore approximately proportional to its total similarity. The interaction between selected schemas proceeds as follows. A schema is selected from the interim population  $\phi$ . A transformation operator  $t_j$  is selected with a probability proportional to its weight  $w_j$ . If the arity of the transformation operator is  $m$ , then the remaining  $m - 1$  schemas are randomly selected from the interim population  $\phi$ . Following application of the transformation operator the produced schema is included in the new population:  $P(\{E\}, T + 1)$ . The interaction process is repeated  $N$  times to generate a new full population.

### 3.2.3 Transfer

Given the set of edges,  $R \subseteq V \times V$  of the ensemble graph,  $G$ , each edge  $(E_i, E_j)$  is associated with two components:  $w_{ij}$  and  $w_{ji}$ . These components, shown in Figure 3 as the strength of relations, are positive integers that define transfer rates from  $E_i$  to  $E_j$  and  $E_j$  to  $E_i$ , respectively, and  $Q = \sum_j w_{ij} \leq N$ . Each schema  $s$  in the ensemble has a propensity to transfer  $\mu(s)$ , which is a monotonic function of the change of similarity over  $k$  iterations. Initially,  $w_{ij}$  for each ensemble  $i$  is set to a low value. These transfer rates, which emulate conceptual transfer and analogy-based discovery, may change over time. Learning takes place as information about the similarity of copied and transferred models is gathered. If models that are transferred from  $E_i$  to  $E_j$  improve their average similarity, the transfer rate for migration from  $E_i$  to  $E_j$  is updated to increase number of transfers; otherwise, the transfer rate is decreased. At each round of evaluation, for every  $(E_i, E_j)$  in the analogue ensembles graph, the population in ensemble  $i$  is scanned to locate  $K$  schemas with  $\mu(s) \geq \gamma_{transferThreshold}$  and from these schemas a subset of size proportional  $\frac{m_{ij}}{Q}$  schemas are selected for transfer to  $E_j$ . The generation and transfer mechanisms update the contents and structure of the ensemble graph as a complex adaptive system mimicking an evolving ecology. The network of ensembles enables interaction between models. The boundaries may denote separate attributes and targeted objectives in the referent. Ensembles communicate with each other and share models across their boundaries. Valid models are discovered through continual flows of models so that ensembles in the graph can sustain themselves and improve the impact and usefulness of local solutions.

## 4 CASE STUDY: APPLICATION OF RGV TO COLORSCAPE MODEL

To illustrate the use of the evolutionary dynamics of coupled generation and evaluation (i.e., validation) processes, we developed an agent-based model, called ColorScape. The model aims to capture the formation and development of scientific disciplines. Both traditional and virtual scientific communities can be simulated using ColorScape.

#### 4.1 Validation Framework

To generate and validate mechanisms and experiment spaces capable of exhibiting behavior consistent with available empirical data for both types of communities, we conducted two experiments. To gain preliminary insight about the use of RGV, we simplified the method presented in Section 3. Instead of using a network of ensembles, in each experiment we applied the strategy with a single ensemble containing a large population of models. In the first case, our aim is to discover sufficiently valid models that are capable of generating the structural characteristics of the overlay map of traditional scientific communities. The second experiment is intended to discover plausible models that can generate regularities observed in the empirical network structure of a virtual scientific organization, called Open Biomedical Ontologies (OBO) Foundry.

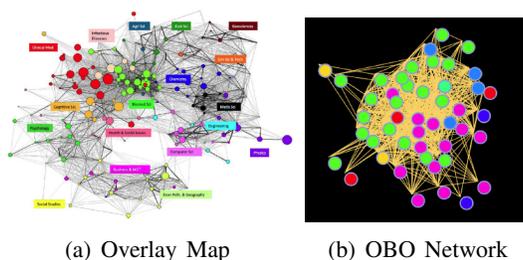


Figure 4: Empirical Collaboration Networks.

Figure 4 illustrates the two empirical network structures against which we validate generated models. Network metrics such as density, clustering coefficient, core-periphery ratio, and degree centrality are used as validation criteria. The generic and customizable simulation model and its associated validation framework are implemented by RePast. RePast provides mechanisms of both single run and batch run. However, the configuration parameters cannot be evolved during either single or batch run. So, a new runner that inherits the default runner of RePast is introduced so as to dynamically translate the gene (model schema) to configuration parameters and collect the outputs that are required for the computation of similarity (validation) measures.

As shown in Figure 5(a), the abstract GA class implements all other functions except fitness function. The fitness function is overridden by a child class that runs the simulation model after converting the gene to the configuration parameters. Figure 5(b) shows the sequence diagram of the validation framework. The main class invokes the genetic algorithm embedded within the RGV method, which in turn invokes simulation model for similarity (validation metric) value.

In the main function, the class `GAColorscape` is instantiated with four parameters: population size, gene length, maximum generations, and the runner object for `Colorscape`. Then the genetic algorithm implemented in the class `GAColorscape` is started. When the fitness value of a specific gene is needed, the gene is converted to configuration parameters and then the `RunColorscape` is invoked with the configuration parameters as input arguments. In `RunColorscape`, the configuration parameters loaded from default batch run XML file of the `Colorscape` model are updated. Next, the `Colorscape` model is replicated 30 times with configuration parameters. For each single run of the `Colorscape` model, output metrics such as density, centrality, clustering coefficient are stored. Finally, the reciprocal of the distance between the output metrics and expected values is calculated as the fitness of the gene.

#### 4.2 The ColorScape Model

*ColorScape* is an agent-based model used to study collective creativity in epistemic communities. Three major components are used to specify growth and development of scientific communities: *domain*, *maturity*, and *resources*. Domain refers to discipline, whereas maturity indicates the degree of development in that specific domain. Resources held by a community are vital to undertake scientific activities. In order

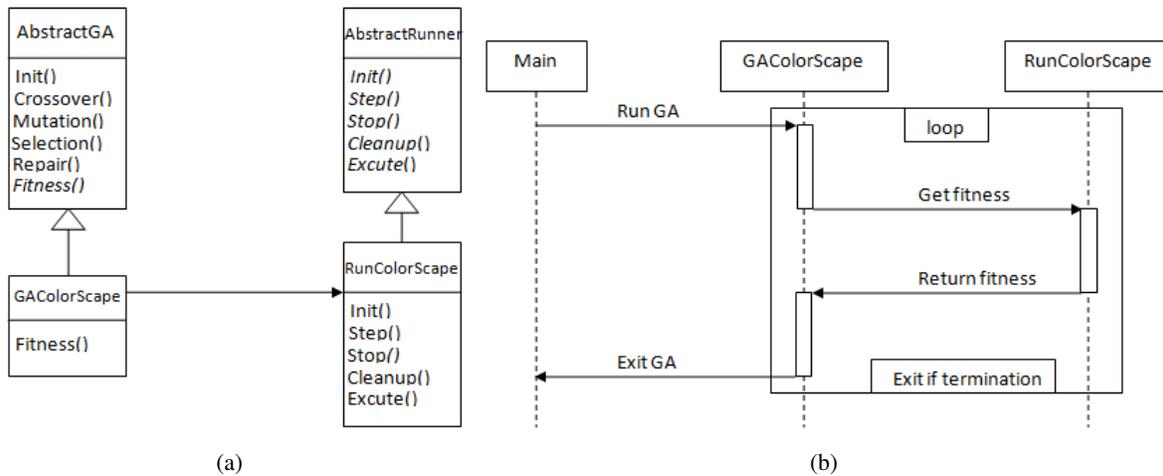


Figure 5: Class and Sequence Diagram of Framework.

to visually depict the evolving states of communities, the HSB (Hue, Saturation, and Brightness) color model is used. Hue indicates the domain of a community. Saturation within the color spectrum represents maturity. The degree of brightness corresponds to the level resource. Figure 6 depicts the snapshots of our model with grid and network topology respectively, where each cell represents a community whose color corresponds to its internal state. As shown in Figure 6, the state space of communities exhibits a color landscape.

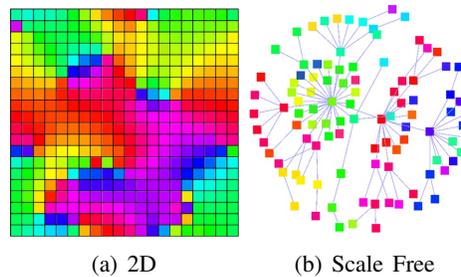


Figure 6: Snapshots of ColorScape Model.

The behavior of scientific communities in ColorScape is comprised of six subprocesses: resource allocation, interaction within community, learning, innovation, growth, and fade. Resource allocation refers to strategies that distribute resources to communities. Two factors are considered for distribution: *resource size* and *allocation strategies*. Interaction within a community refers to scientific activities at the macro level i.e., community is driven by funding to improve its maturity. Learning and innovation between communities mimic the boundary processes among communities, i.e., communities affect and are influenced by peer communities. There are two options of innovation: reorganization and specialization. A parameter named *reorganization tendency* is used to determine the type of innovation. Learning and innovation describe the influences from peers, so there is a parameter called *receptivity* that is defined as the ratio of the sum of influences of neighbors to its own resistance. Growth is defined as the process through which communities extend and increase their influences, where *growth threshold* defines the frequency of growth operation. Fade refers to disappearance of the community due to loss of resources. Besides these parameters, there are two additional ones: *population* and *communication frequency*. Population refers to the initial number of communities, whereas communication frequency denotes frequency of communication

between communities. In the next section, we discuss how to encode these configuration parameters into genes that constitute the model schema.

### 4.3 Model Schema Encoding and Decoding

In an earlier study, we identified twelve resource allocation strategies. Figure 7 shows a sample of gene after encoding the configuration parameters of the ColorScape model, where the total number of bits is 21.



Figure 7: Gene Encoding.

To calculate the fitness of each gene, the gene has to be translated (decoded) into configuration parameters of the *experiment space*. The ColorScape model is then batch run with the given the parameters and value assignments. The results returned from each simulation are then used as input to the fitness function. Within the 21 bit gene, the first two bits are used to decode *population* variable, denoting different range of community numbers (e.g., 00→10, 01→50, 10→100, 11→200). Bits 2 through 5 designate the *allocation strategy* variable that refers to resource distribution styles such as uniform allocation with fixed external resource, allocation proportional to contribution etc. The *resource* variable controls the amount of external resource (e.g., expertise, funding) allocated to a community at each time step, and bits 6 and 7 are used to denote resource levels that vary from low to high. *Tolerance* is decoded by bits 8 and 9. It denotes ability of a community to innovate; whereas, *receptivity* considers ability of a community to learn from peer communities. Bits 10 and 12 are used to specify different levels of *reorganization* tendency. This variable is used to control propensity to create new sub-communities that is similar to speciation tendency in ecological systems.

### 4.4 Fitness

The fitness is the indicator showing how close simulation outputs produced by a model defined by an instance of a schema are against the target system i.e., OBO data. As a general measure of the degree of socio-technical interaction, we use and interpret density, centrality, clustering coefficient, average path length and core/periphery ratio so as to identify the OBO network. The core/periphery network pattern is considered as a stable, sustainable, and innovative structure (Krebs and Holley 2002). Given the same number of core members, increasing level of periphery members is beneficial for bringing new external ideas. The core/periphery ratio is used to measure the percentage of the members in the core to the members in the periphery. The fitness is a function of the distance between metric values generated by the ColorScape Model under an evolving set of configuration parameter values and the corresponding values of the same metrics observed within the OBO network.

### 4.5 Preliminary Results

In this preliminary experiment, instead of using a network of ensembles, we instantiate two independent ensembles of model schemas. These two ensembles are used to discover valid models capable of generating structural regularities observed in the overlay map and OBO data, respectively. In this experiment, we did not allow transfer between two ensembles. Future work will extend this study to facilitate convergence to a set of model schemas that are successful in generating phenotypes (attributes) of both empirical data.

Table 1 presents a comparison of network metrics generated by RGV system against the corresponding metrics derived from the OBO data (expected values in the table). Since the confidence intervals of metrics

Table 1: Simulation vs. OBO Data.

Metrics	Mean Value	Confidence Interval at 90%	Expected Values
Number of Communities	55.633	[46.076, 65.190]	49
Density	0.605	[0.521, 0.689]	0.549
Clustering Coefficient	0.846	[0.812, 0.881]	0.880
Centrality	0.355	[0.302, 0.407]	0.405
Average Path	1.404	[1.317, 1.491]	1.406
Core/Periphery Ratio	50.9	[37.2, 64.6]	23.5

generated from the simulation data contain the corresponding values observed in the OBO network, we conclude that the ColorScape model can generate network structures similar to OBO.

In addition, the best configuration parameters against the network of OBO are recorded in Table 2. From the table, we can observe that the best configuration has a medium level tolerance (0.6), high receptivity (0.9), and high degree of communication frequency (1.0). These are peculiar characteristics of open source science communities.

Table 2: The Best Configuration against OBO.

Name	Value
Population	30
Resource Size	1
Tolerance	0.6
Reorganization Tendency	0.5
Receptivity	0.9
Allocation Strategy	Uniform allocation with technology transferring
Communication Frequency	1.0
Threshold to Grow	0.5

Table 3 summarizes the comparison of network metrics generated by the ColorScape model with the corresponding metrics from the science overlay map (expected values in the table). Although the metrics of the overlay map do not fall into the confidence interval of the ColorScape model, we can still observe sufficiently similar simulation outputs for the purpose of our study. Considering the relatively large size of the network (222 nodes and 6820 edges), high degree of randomness and multiple fitness objectives (i.e., metrics), it is reasonable that the ColorScape model does not produce exact same outputs for both open source and traditional scientific enterprise structures. However, if the objective metrics are reduced, then the ColorScape model is able to produce sufficiently similar outputs to the science overlay map as well.

Based on the exploration within the experiment space, we instill confidence that the ColorScape model is a general purpose model that is capable of creating network patterns similar to both traditional science (e.g., overlay map) and open source communities (OBO).

Table 3: Simulation vs. Overlay Map.

Metrics	Mean Value	Confidence Interval at 90%	Expected Values
Number of Nodes	296	[272, 320]	222
Density	0.214	[0.194, 0.233]	0.139
Clustering Coefficient	0.574	[0.559, 0.589]	0.648
Centrality	0.354	[0.342, 0.366]	0.216
Average Path	1.847	[1.821, 1.873]	2.415
Core/Periphery Ratio	77.3	[65.5, 89.1]	73.0

## 5 CONCLUSIONS

In this study we introduce a validation framework for agent-based models by taking advantage of operational principles of creative evolutionary systems to facilitate exploration and exploitation within the hypothesis and experiment spaces of model specifications. Based on the experiments conducted using the ColorScape model, we observe that the framework can successfully be leveraged in practice to synergistically integrate validation and generative modeling processes. The ability to instantiate, generate, transform, execute, and if necessary, evolve multiple models of interacting mechanisms, in parallel, all of which take similar but slightly different perspectives on the same referent system, opens the door to the automatic generation and selection (by falsification) of many somewhat different hypothetical, including non-intuitive mechanisms for a referent. Such an exponential increase in model and hypothesis throughput could promote scientific discovery while increasing opportunities for creative leaps.

## REFERENCES

- Balci, O. 2003, December. "Verification, validation, and certification of modeling and simulation applications". In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 150–158. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Balci, O. 2004, December. "Quality Assessment, Verification, and Validation of Modeling and Simulation Applications". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 122–129. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Balci, O. 2010. "Golden Rules of Verification, Validation, Testing, and Certification of Modeling and Simulation Applications". *SCS Modeling and Simulation Magazine* 1 (4).
- Bentley, P. J., and D. W. Corne. 2002. "An Introduction to Creative Evolutionary Systems". In *Creative Evolutionary Systems*, edited by P. J. Bentley and D. W. Corne, Chapter 1, 1–75. Mogan Kaufmann.
- Bharathy, G. K., and B. Silverman. 2010, December. "Validating Agent-based Social Systems Models". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hukan, and E. Yücesan. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Davis, P. K., and J. H. Bigelow. 2000, December. "Exploratory Analysis Enabled by Multiresolution, Multiperspective Modeling". In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A.

- Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 127–134. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Edmonds, B., and E. Chattoe. 2005, July. “When Simple Measures Fail: Characterising Social Networks Using Simulation”. Technical report, Center for Policy Modeling. Manchester Metropolitan University.
- Epstein, J. M. 2007. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.
- Hunt, C. A., G. E. P. Ropella, T. N. Lam, J. Tang, S. H. J. Kim, J. A. Engelberg, and S. S. Bahaei. 2009. “At the Biological Modeling and Simulation Frontier”. *Pharmaceutical Research* 10 (1): 2369–2400.
- Klahr, D., and K. Dunbar. 1988. “Dual Space Search during Scientific Scientific Reasoning”. *Cognitive Science* 12:1–55.
- Krebs, V., and J. Holley. 2002. “Building Sustainable Communities through Network Building”. Technical report, Orgnet.com, <http://www.orgnet.com/BuildingNetworks.pdf>.
- Miller, J. H., and S. E. Page. 2007. *Complex Adaptive Systems, An introduction to computational models of social life*. Princeton University Press.
- Schreiber, D. 2002. “Validating agent-based models: From metaphysics to applications”. *Annual Conference of the Midwestern Political Science Association, Chicago, IL:235–241*.
- Yilmaz, L. 2006. “Validation and Verification of Social Processes within Agent-Based Computational Organization Models”. *Computational and Mathematical Organization Theory* 12 (4): 283–312.
- Yilmaz, L., and C. A. Hunt. 2011. *Advanced Concepts and Generative Simulation Formalisms for Creative Discovery Systems Engineering*, Chapter Intelligence-Based Systems Engineering, 233–258. Springer.
- Yilmaz, L., and T. Ören. 2009. “Rethinking Modeling and Simulation to Enhance Creativity and Computational Discovery”. In *Proceedings of the 2009 Summer Computer Simulation Conference*, edited by L. Yilmaz, O. Balci, X. Hu, and M. Sierhuis, 66–74: SCS Press.

## AUTHOR BIOGRAPHIES

**LEVENT YILMAZ** is Associate Professor of Computer Science and Software Engineering and holds a joint appointment with the Industrial and Systems Engineering at Auburn University. He received M.S. and Ph.D. degrees from Virginia Tech. His research interest are in Modeling & Simulation, Agent-directed Simulation, and Complex Adaptive Systems with a focus in advancing the theory and methodology of modeling and simulation and developing computational models of socio-technical, cognitive, and cultural systems with applications to science of science and innovation policy. He is a member of the Board of Directors of SCS and is currently serving as the Editor-in-Chief of *Simulation: Transactions of the Society for Modeling & Simulation International*. He is member of ACM, IEEE Computer Society, Society for Computer Simulation International (SCS), and Upsilon Pi Epsilon. His email and web addresses are [yilmaz@auburn.edu](mailto:yilmaz@auburn.edu) and <http://www.eng.auburn.edu/users/yilmale>.

**GUANGYU ZOU** is a Ph.D. candidate at the Computer Science and Software Engineering Department at Auburn University. He received his M.S. degree from Industrial and Systems Engineering at Auburn University. His research interests are in modeling and simulation, agent-based modeling, and complex adaptive systems. His email address is [gzz0001@auburn.edu](mailto:gzz0001@auburn.edu).

**OSMAN BALCI** is a Professor of Computer Science at Virginia Polytechnic Institute and State University (Virginia Tech). He received B.S. and M.S. degrees from Boğaziçi University (Istanbul) in 1975 and 1977, and M.S. and Ph.D. degrees from Syracuse University in 1978 and 1981. Dr. Balci currently serves as an Area Editor of *ACM Transactions on Modeling and Computer Simulation*, Modeling and Simulation (M&S) Category Editor of *ACM Computing Reviews*, and Editor-in-Chief of *ACM SIGSIM M&S Knowledge Repository*. He served as the Editor-in-Chief of two international journals: *Annals of Software Engineering*, 1993–2002 and *World Wide Web*, 1996–2000. He served as Chair of ACM SIGSIM, 2008–2010; and Director at Large of the Society for M&S International, 2002–2006. Most of Dr. Balci’s work has been funded by

*Yilmaz, Zou, and Balci*

the U.S. Navy since 1983. His current areas of expertise center on software engineering, software/system architecting, and M&S. His e mail and web addresses are [balci@vt.edu](mailto:balci@vt.edu) and [manta.cs.vt.edu/balci](http://manta.cs.vt.edu/balci).