Model Mechanisms and Behavioral Attractors

H. Van Dyke Parunak[0000-0002-3434-5088]

Parallax Advanced Research, Beavercreek, OH 45431
van.parunak@gmail.com

Abstract. In social modeling, a *computational environment* runs a *model* that represents the *world*. The states the model explores (its *behavioral attractor*) are typically fewer than its description suggests. The mapping between model and attractor depends not only on its *parameters* (exploring variants of the world) and its *conventions* (imposed by the computing environment), but also its *mechanisms* (components of the model representing selected dimensions of the world). We illustrate the impact of different mechanisms on the attractor. In our case, in general, the more mechanisms one implements, the smaller the attractor ("the more you model, the less you see"), but with unexpected twists.

Keywords: Agent Based Modeling, Social Simulation, Complex Dynamics, Model Parameters, Model Mechanisms, Behavioral Attractor.

1 Introduction

The user of a social model expects the model to generate a range of behaviors. For example, how many distinct behaviors can the actors manifest? How does their spatial distribution vary over time? The range of behaviors generated by a running model (the system's *behavioral attractor*) is usually smaller than the static model suggests.

The mapping between a model and its attractor can depend on three different sets of variables: *parameters*, *conventions*, and *mechanisms*. Each of these describes a different component of the modeling enterprise, in which a *computational environment* runs a *model* that represents the *world* (Fig. 1).

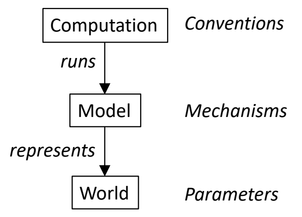

Fig. 1: Central Relations

- *Parameters* describe the world that the model represents. Varying them explores how the world might behave if its characteristics (e.g., relative group sizes) change.
- *Conventions* are unrelated to the real world but imposed by the computational environment (such as agent execution order on a von Neumann machine or agent behavior at arena boundaries), and varying them explores the degree to which the behavioral attractor is an artifact of that computational environment.
- *Mechanisms* are model components that reflect facets of the world. For example, real social actors have short term preferences and strategic goals that guide their choices, subject to constraints among available options and the actions of other ac-

tors. Not every social model has a mechanism for each of these (preferences, goals, option constraints, interactions), and no social model has a mechanism for every possible dimension.

Modelers assume that a model with fewer mechanisms than the world's facets can still give useful information. Most modeling frameworks offer few alternative mechanisms, seducing modelers to ignore the impact of mechanism choice. SCAMP (Social Causality using Agents with Multiple Perspectives) [13], a causal language and simulator for social scenarios, has a rich array of mechanisms that can be activated independently of one another. In general, the more mechanisms we activate, the smaller the attractor ("the more you model, the less you see"), but interactions among mechanisms lead to anomalies. For instance, a more constrained attractor may lie partly outside less constrained ones with the same conventions and parameters. Adding mechanisms can not only sharpen the model's focus, but also shift its location.

These results are of immediate interest to teams who are using SCAMP. In addition, our methods should be helpful to other modelers in understanding the implications of their choice of mechanisms. Our exploration of SCAMP's dynamics is a concrete example of what might be done in other frameworks.

Section 2 summarizes related work. Section 3 describes the mechanisms of SCAMP that these experiments vary. Section 4 describes our methodology. Section 5 presents the experimental results. Section 6 discusses their implication for interpreting the results of a SCAMP run, highlights implications of this experiment for other social modeling systems, and outlines future work.

2 Related Work

We expect behavior to vary with model *parameters*, which are the focus of most studies of the dynamics of agent-based systems (e.g., [2-4,20]), including studies of tipping points (parameter values where behavior changes discontinuously, leading to a phase shift) and lever points (parameters whose change has a lasting, directed effect) [1,15]. Wolfram [19] identifies four distinct classes of one-dimensional 0-1 nearest-neighbor cellular automata, varying only the update rule, the key model parameter. Verification methods such as sensitivity analysis [5] (p. 24) or comparison of agent trajectories with observed data also explore behavioral changes when parameters change, but not the impact of changing conventions or mechanisms.

Studies of the impact of *conventions* imposed by the computing environment are less common, but revealing. For example, a differential equation model and an agent-based model can yield qualitatively different results for the same parameters [16,18]. Restricting ourselves to agent-based modeling, on a von Neumann machine, agents can run only one at a time, and different scheduling disciplines for entities that in reality execute concurrently have repeatedly been shown to lead to different results [6,8]. [9] reviews the extensive literature on the impact of scheduler synchrony.

This study focuses neither on the parameters that vary the world explored by a model nor on the conventions imposed by computation, but on differing sets of mechanisms that the model uses to represent facets of the world. Naively, one hopes that even a primitive model will be useful, and that adding more mechanisms will add more detail to the results of the initial model. Unexpectedly, such refinements can also move the focus, and cause other anomalies. We know of no other ABM work that demonstrates this effect, because most modeling frameworks do not offer multiple mechanisms that can be activated independently of one another.

3 The SCAMP Causal Modeling System

This section explains enough of SCAMP's structure to motivate our experiments. For further details, see [11,13]. Our experiments use two of SCAMP's four perspectives.

1. A *causal event graph* or CEG is a directed graph whose nodes represent types of events in which agents can participate.
2. A *hierarchical goal network* or HGN is a directed acyclic graph that models the goals of a group of agents and how those goals are related to the levels of participation on events in the CEG. Leaf nodes in the HGN are linked, or *zipped,* to event nodes that either support or block them.

The CEG has two sorts of edges:

1. An *agency edge* from node A to node B means that an agent currently participating in an event of type A may consider an event of type B as its next activity. A chain of agency edges defines a plausible *narrative* of the agent's experience. Depending on its group membership, an agent has *agency* for a subset of the nodes in the CEG, and can move between two nodes only if it has agency in both of them. Most nodes have multiple successors, making the CEG a *narrative space* [14] that captures many possible narratives. The main output from a SCAMP model is the history experienced by each agent. Agency edges are obligatory.
2. Sometimes one event causally constrains another even though no agent has agency for both events. For example, an act of God such as a pandemic may hinder events in which people gather together, or enhance hospitalization events. SCAMP captures these relationships with *influence edges*. Influence edges are optional.

When an agent completes one event in the CEG, it selects the next based on two vectors. Each event has a *feature vector* that describes the event's effect on agent wellbeing, how urgent the event is to satisfying the HGNs to whose leaf goals it is zipped, and how extensively agents of each group have participated in it recently. Each agent carries a *preference vector* over the same space. To choose its next event, the agent

1. computes the dot product of its preference vector and the feature vector of each accessible event type in the CEG,
2. exponentiates each dot product so that it is non-negative, defining a roulette,
3. adjusts the presence and size of segments with incoming influence edges based on the participation levels on events at the origins of those edges,
4. and spins the roulette.

In step 3, a *prevent* or *enable* influence edge can remove or add an event to the roulette that guides agent choice, changing the structure of the CEG dynamically as participation levels on influencing events vary.

Each agent adjusts its roulette before spinning by raising the size of each sector to its personal *determinism* level, modeling human deviation from pure rationality. An agent with determinism 0 makes completely random choices, while determinism 100 models a utility optimizer. Our experiments set agent determinism to 100, while our baselines set it to 0 to generate a random walk.

SCAMP uses polyagents [10], which represent each domain entity by a single *avatar* that can deploy a swarm of *ghosts*. The ghosts explore their avatar's possible next choices by looking ahead a fixed distance (here, two events). At each step, they form a roulette over all nodes in the CEG that are immediate successors to their current node, choose one node, and increment the node's presence feature for their group proportional to the value of the position in which they find themselves. The avatar chooses its next step by choosing probabilistically based on the presence features deposited by its ghosts. This mechanism simulates the well-documented psychological process of evaluating actions by mental simulation of possible outcomes [7].

We base our experiments on a model of civil strife inspired by recent history in Syria. The CEG in this model includes 460 event nodes with 1106 agency edges and 400 influence edges. The six HGNs, one for each group, include 122 goals or subgoals. 77 leaf goals are zipped to 177 event nodes.

4 Experimental Methodology

Our methodology has three parts.

1. Define how to measure *the space of behaviors*.
2. Identify the *mechanisms* that an instance of the model supports. A given set of mechanisms defines a *configuration*. We are interested in how the size of behavior space varies with the configuration.
3. Identify a configuration to represent an unconstrained *baseline*.

4.1 Defining Behavior Space

An analyst constructing a SCAMP model starts with the CEG, defining types of events that might occur in the domain and linking them into reasonable narratives for agents belonging to different groups. One useful measure of behavior space is how many of these event types the system actually explores. Two levels of exploration are meaningful. The first counts node *coverage*, in several ways:

1. How many nodes do *ghosts* visit in evaluating possible futures for their avatars?
2. How many nodes do ghosts *consider* in evaluating possible futures for their avatars? These are successor nodes to those nodes that the ghosts actually visit.
3. How many nodes do the *avatars* visit in carrying out their decisions?

We measure these values for multiple runs of each configuration, with different random seeds. In this paper, we run at least six runs per configuration.

We also look at how similar the sets of nodes under each measure are for repeated runs with different random seeds. Let Q and R be the sets of nodes explored (under one of the options above) for two runs of the same configuration, and let S be the union of the sets explored by both runs. Then the *overlap* between Q and R is $|Q \cap R|/(|Q| + |R| - |Q \cap R|)$.

We hypothesize that as we add mechanisms, the numbers of distinct nodes in each category will drop (the attractors will shrink) while the overlaps will increase, because the system will be attracted into the same region of state space. As we will see, the data hold some surprises that yield important insight into the system's behavior.

These measures do not by any means exhaust those that could be considered. Since a commonly used output of SCAMP is the set of behavioral trajectories followed by the agents, one very relevant measure is the number of distinct trajectories that avatars execute. We leave that analysis for future work.

4.2 SCAMP's Mechanisms

SCAMP offers several mechanisms to capture different dimensions of the world.

The most basic is the *structure of the agency edges* in the CEG, which record the meaningful behavioral trajectories available to agents. Even if agents execute random walks, the branching factors differ along different paths, so that nodes only accessible along highly branched paths will have a lower probability of being sampled in a run of a given length than those with less ramified approaches.

The mean node degree restricted to agency edges in our example CEG is 4.74, not much more than the limit of 4 for an infinite square lattice, but degree in the CEG is highly variable. Consider a synthetic baseline of 460 integers randomly selected from [3, 6]. The mean is 4.5, comparable to our data, but Pearson's kurtosis for this synthetic baseline is 1.64, well below the threshold of 3 associated with normally distributed data. For our CEG, the kurtosis of node degree is 8.7, reflecting the heavy tail of nodes with high degree (up to a maximum degree of 21).

For comparison, we construct a rectangular directed lattice of 21*22 = 462 nodes, over which we do a random walk (with both ghost and avatar determinism set to 0). A random walk on a regular lattice with restart will visit every node, if it runs long enough. We expect the CEG to perform similarly. We also do a random walk over the CEG model itself, augmented with a single START and a single STOP node.

Psychological preference is modeled by the *feature space* that defines agent preferences and event features. Without preferences, ghosts perform a random walk in laying down the presence features that guide avatars. With preferences, ghosts will favor some nodes over others, based on the features that the model builder has defined for those nodes. We expect a) agents using preferences will explore fewer nodes than those walking randomly, b) overlap across runs will be greater with preferences than without, and c) the longer the model runs, the more nodes will be visited.

SCAMP's HGNs model strategic reasoning. Each HGN monitors the recent participation level on event types to which it is zipped to assess its current *satisfaction*, then computes the *urgency* feature of each of these events. Agents respond to urgency according to their preferences. If an agent is running without preferences, the HGN is irrelevant. But if preferences are active, we expect HGNs to focus the agents' attention, reducing the number of nodes explored and increasing their overlap.

Influence edges model causal influences among event types between which agents do not move directly, modulating the probability of destination nodes dynamically based on participation levels on source nodes. Again, including this mechanism should reduce the number of nodes visited and increase their overlap.

A *configuration* is a binary string indicating active mechanisms. The first position shows whether (1) or not (0) preferences are active. The second position shows HGNs, and the third, the use of influence edges. Thus in 000, the only mechanism is the structure of agency edges, 100 indicates the use of preferences alone, 110 adds HGNs, and 001 is the use of influence edges alone. The decimal values of these strings identify configurations 0 (no mechanisms active) to 7 (all mechanisms active). Configurations 2 and 3 (HGNs without preferences) violate the assumptions of the model and are not included. Configurations 4-7 include preferences, configurations 6 and 7 include HGNs, and odd configurations include influence edges. Our configurations thus form a partial lattice (Fig. 2). All configurations use the same parameters to describe the world and run with the same conventions.

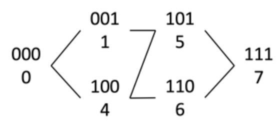

Fig. 2. Configuration lattice: 001 = influences, 010 = HGN, 100 = preferences

4.3 Random Baseline

In addition to a space within which the attractor is defined and mechanisms that might impact the attractor, we need a baseline against which to compare their impacts. We provide two baselines, L (the 21*22 lattice) and R (the CEG), with both ghost and avatar determinism set to 0 so that they ignore the roulette entirely. In configuration 0, unlike R, avatars have determinism 100, and follow their (randomly moving) ghosts.

5 Results

Our experiments illustrate how studying the behavioral attractor as a function of model mechanisms can yield valuable insights that confirm or correct our intuitions and call attention to behaviors that invite further study. Our study is exploratory, and we present most results as boxplots [17].[1] In some cases, we compute the significance of pairs of results using the one-sided Mann-Whitney U test. p-values greater than 0.05 are reported as not significant.

We begin with summary plots that characterize the data and show the impact of run length on our measures. Then we examine how visits and overlaps vary with configu-

[1] The box extends from the upper to the lower quartile of a data series. The bold line marks the median. The whiskers extend to the most distant data points within 1.5 times the inter-quartile range of the quartile limits, and circles mark outliers. Comparing the inter-quartile boxes for two series is a good heuristic for whether they are the same or different.

ration, to see how adding mechanisms affects agent activity. Finally, we offer some summary statistics on the impact on our metrics of the three mechanisms we are studying: preferences, HGNs, and influence edges.

5.1 Making Friends with the Data

First, compare the coverage and overlaps (Fig. 3) of each measure (avatar visits (av), ghost visits (gv), and successors considered by ghosts (sc)) for the baseline configuration (000 ~ 0) and the most constrained (111 ~ 7). Ghosts visit fewer nodes than they consider, and avatars visit only a small fraction of those explored by ghosts. Added mechanisms reduce the number of nodes that the ghosts consider and visit, as expected, but the number of nodes visited by avatars is unchanged. Additional mechanisms focus the ghosts' attention more closely, but however broadly or narrowly the ghosts explore, an avatar chooses one path from those explored by its ghosts, and in a run of fixed length visits only a limited number of nodes. The avatar nodes are not the same in the two configurations, but by the structure of the program the coverage is the same size.

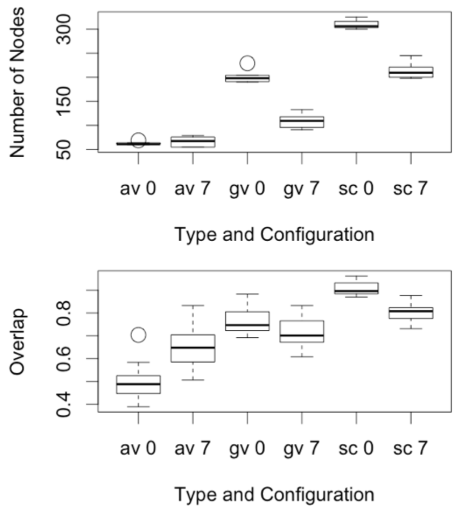

Fig. 3. Nodes Visited (Left) and Overlaps (Right) by Types and Configurations

We expect overlap to increase with mechanisms, as agents focus their attention on fewer nodes. Fig. 3 confirms this intuition for avatar visits, but overlaps for ghost visits and successors actually *decrease*, a phenomenon we discuss in Section 5.3.

In a regular directed lattice, coverage increases with run length. Most of our results are runs of 1000 Repast ticks. Fig. 4 shows the effect of increasing run length to 2000. We compare configuration 0 with 4, which (we will see) is particularly influential. In x-axis labels, the first digit (0, 4) is configuration, and the second (1, 2) is run length in k-ticks.

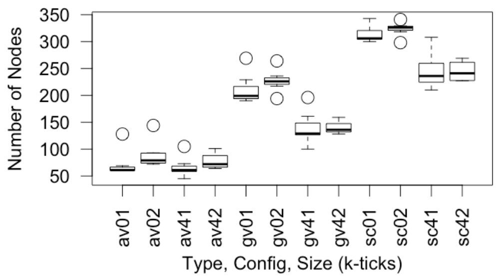

Fig. 4. Effect of Run Length on Coverage

The intuition is correct for avatar visits, and for ghost visits and successors in configuration 0. But for configuration 4, the preference mechanism leads the system to converge, and longer runs do not increase ghost visits or successors.

The median value of sc01, 306, leaves 162 event types in a typical run that the ghosts never consider. However, these 162 CEG nodes are not the same in each run. The median overlap is about 90%, and many runs show lower overlaps between pairs of runs for each configuration. For example, while the maximum successor nodes in any single run of configuration 0 is 325, all of the runs together explore 343 nodes. This still misses 117 nodes of the complete CEG, but suggests that multiple runs are at least as important as run length in sampling the causal graph adequately.

5.2 Impact of Adding Mechanisms

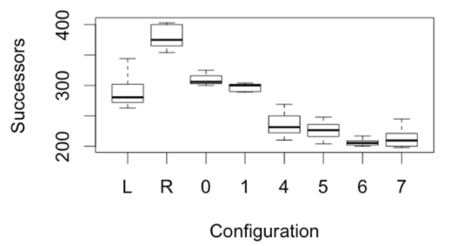

Fig. 5. Successors by configuration

Fig. 3 shows a clear reduction in coverage for successors and ghost visits between runs with no mechanism except the CEG, and all mechanisms. Fig. 5 shows intermediate configurations. Ghost visits show the same pattern.

In the baselines, random walk on a lattice (configuration L) offers fewer successors to consider (and thus fewer ghost visits) than on the CEG (configuration R), reflecting the long tail in our model's degree distribution. Successors and ghost visits on baseline R are higher than with any mechanisms, which is not surprising.

As expected, both measures tend to decrease as we add mechanisms. Two details are particularly interesting.

1. The drop from configurations 0 and 1 (no preferences) to configurations 4-7 (with preferences active) is particularly large, suggesting that preferences have more influence on the system than do HGNs or influence edges.
2. Configuration 6 (preferences and HGNs without influence edges) appears to be *lower* than the more highly constrained configuration 7 (which adds the influence edges). This unexpected result shows an unanticipated but realistic interaction between the two mechanisms. An agent's goals (in life and in SCAMP) guide its actions by identifying high-priority events in which the agent should participate, and the usefulness of goals will decrease if other events block access to those urgent events through influence edges.

In contrast to successors and ghost visits, avatar visits do *not* change with more mechanisms. This observation is consistent with Fig. 3: while ghosts can explore more or less narrowly, each avatar follows only one path, and thus visits only a relatively constant number of nodes for runs of a given length.

Adding constraints not only decreases attractor size (for gv and sc), but also shifts its location. Define the *alignment* of one configuration with another with a subset of its mechanisms (say, 101 with 100, 001, or 000) as the percent of events in its attractor with that of the less constrained con-

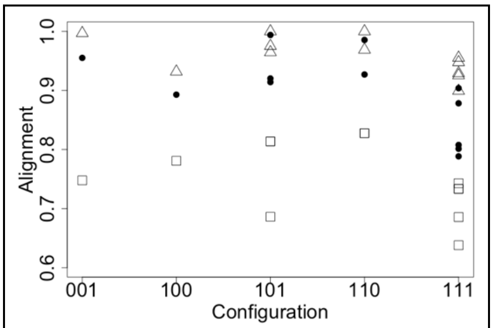

Fig. 6. Alignment of Successive Constraints: square = av, dot = gv, triangle = sc

figuration. Fig. 6 shows the alignment of each configuration more constrained than 000 with all of its predecessors in the configuration lattice (Fig. 2), for all three measures (triangle = sc, dot = gv, square = av). (The top two squares for 101 have the same value.) As it happens, the highest alignment for each configuration (including av) is not with its immediate predecessor in Fig. 2, but with 000. We expect alignment of 1 (indicating that the attractor for more mechanisms falls entirely within that for fewer), but except for sc for 001, 101, and 110, the more constrained attractors have lower alignment. This result challenges the common assumption that unmodeled facets of the real world result in a more fuzzy but still essentially correct outcome. In fact, adding mechanisms for these facets could shift the model's output.

5.3 A Closer Look at Overlap

In addition to monitoring the coverage of nodes considered or visited (our approximation of a model's attractor), it is also useful to study the variation among the sets of nodes visited in different runs of the same configuration. Intuitively, we expect overlap to increase with number of mechanisms. This intuition must be qualified.

With significance p = 2E-16, avatar visits have lower overlap than ghost visits, and ghost visits have lower overlap than successor coverage. We hypothesize that this difference reflects the fact (Fig. 3) that there are far more successors than ghost visits, and far more ghost visits than avatar visits, out of a fixed number of nodes. Higher coverage of the CEG leaves fewer nodes on which runs can differ with each other.

Fig. 7 shows how avatar and successor overlaps vary with configuration. Avatar overlap satisfies our intuition that with more mechanisms guiding agents into similar regions of the CEG, overlap should increase. Consistent with this dynamic, the baseline configurations L and R, with both ghosts and avatars executing random walks, have the lowest overlaps. Configuration 6 yields the highest overlap. Adding influence edges in configuration 7 reduces overlap, reflecting their interaction with HGNs.

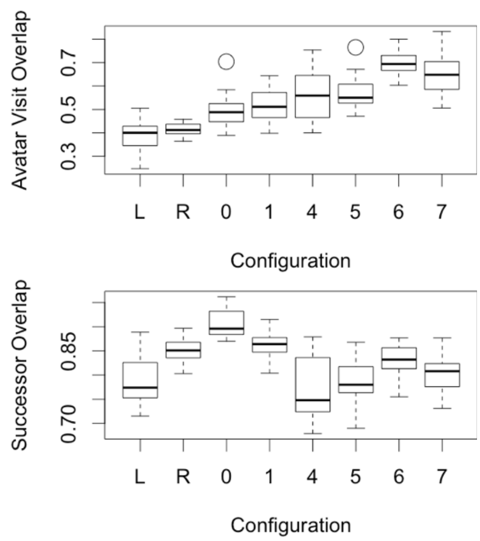

Fig. 7. Overlaps by Configuration

Ghost overlap (not shown) is less intuitive. Overlaps between the sets of nodes visited by ghosts in different runs of the same configuration are invariant with configuration, and do not significantly differ from the baselines.

Overlaps in the successor metric are even more complex. Setting aside the random walks L and R, the overlaps actually *decrease* with added mechanisms! As with the successor and ghost visit coverage metrics, there appears to be a particularly sharp drop with configuration 4, when agent preferences become active. Again, the power of HGNs in drawing agents together is clear in the increased overlap in configuration 6, but faces interference from influence edges in configuration 7.

The overall negative correlation between successor overlap and number of mechanisms is surprising. Perhaps the mechanisms lead the agents into parts of the CEG that they otherwise would not visit. Preferences in particular can lead agents to prefer highly branching regions that otherwise would be relatively inaccessible. In such a region with high node degree, SCAMP's stochastic roulette selection can push different runs in different directions, increasing successor coverage and thus reducing overlap. Modelers who assign favorable features to some events may unconsciously focus more attention on them and ramify the paths to which they lead more than they do for other events, a form of modeling bias of which they should be aware.

5.4 Impact of Individual Mechanisms

Table 1 compares the effect of the preference and HGN mechanisms, aggregated over all configurations, against the baseline. In this table, an entry of the form ">, 0.04" means that the row variable is larger without the column mechanism operating than with it, with significance p=0.04. "<" means that the unconstrained value is smaller than the constrained one, and "NS" means that the p-value is greater than 0.05. The aggregate impact of influence edges for all variables is NS.

Table 1. Mechanism impact on coverage

Variable	Preferences	HGNs
Ghost Visits	>, 7E-7	>, 1E-5
Ghost Overlaps	NS	<, 0.01
Avatar Visits	>, 2E-4	NS
Avatar Overlaps	<, 1E-9	<, 1E-9
Successors	>, 7E-7	>, 2E-5
Successor Overlaps	>, 8E-8	NS

Preference has the most widespread impact of all mechanisms, affecting every measure except ghost overlaps, and it reduces all measures that it affects except overlap among avatar visits, which it increases. The increase of avatar visit overlap is in line with our initial hypothesis: the more mechanisms constrain the model, the fewer different nodes agents will visit, and the more those sets of nodes will resemble each other across runs. The decrease of successor overlap is puzzling, but confirms the impression we drew from Fig. 7.

HGNs are the next most influential mechanism, increasing visits (except for avatar visits) and decreasing overlaps (except for successor overlaps).

Though in the aggregate influence edges do not have significant impact on these measures, they can reduce the contribution of HGNs by limiting agent access to types of events that the HGN identifies as urgent.

6 Discussion and Future Work

Our specific results are of great interest to users of SCAMP, but our message is important for the responsible use of any agent-based modeling framework, in two ways. 1) Programmers and modelers have a sense of the range of possibilities covered by their models, based on the static structure of those models. The actual attractor visited by the model when it runs may be much smaller. Users need to understand the effective coverage of a model under different conditions, and modelers need to understand how adding mechanisms is likely to impact that coverage. Sometimes users will want to increase coverage to consider more possible outcomes; in other cases they will want to decrease it to focus on the most likely outcomes. 2) Adding mechanisms to capture more dimensions of the real world can not only provide a more focused result, but also shift the location of that result in state space.

For ourselves, these results suggest several directions for future work that would not be evident in the absence of this analysis.

- Validate our hypotheses about what features of SCAMP (e.g., widely varying branching factors?) lead to premature focusing, and develop guidelines for modelers who use SCAMP to avoid unrealistic expectations about how aware the system actually is of all the alternatives they are constructing.
- Explore in more detail what leads to some of the counterintuitive behaviors we have discovered, such as the interaction of HGNs and influence edges, and the decrease in successor overlap as we add mechanisms.
- Formally, a system's attractor is the region of state space to which it is constrained *after initial transients have died out*. Our data comes from complete runs, and ignores possible noise from start-up conditions. The start-up period can be identified by plotting the entropy of the roulette constructed by each agent as a function of time [12]. Applying this measure to the analysis in this paper is more challenging than in our previous application of it, but would refine our results.
- The notion of an attractor is only one of several physics-based concepts that can elucidate the dynamical behavior of a social simulation. We are exploring others, such as the graph spectra of emergent social networks .
- At several points, coverage and overlap measured by avatar visits behave very differently than ghost visits and successor counts. Most reports we generate for users concern the movement of avatars, and we have viewed the ghost mechanism and successor structure of the CEG as internal details that are not relevant to analysts, but clearly they are important in assessing the model's dynamic coverage, and we will explore ways to communicate this information to users.

References

Publications by the author are available at https://www.abcresearch.org/abc/papers.

[1] Brueckner, S., Parunak, H.V.D.: Information-Driven Phase Changes in Multi-Agent Coordination. In Proceedings of Workshop on Engineering Self-Organizing Systems (ESOA, at AAMAS 2005), pp. 104-119, Springer (2005)

[2] Butner, J.E., Wiltshire, T.J., Munion, A.K.: Modeling Multi-Agent Self-Organization through the Lens of Higher Order Attractor Dynamics. Frontiers in Psychology, 8:380 (2017)

[3] Cenek, M., Dahl, S.K.: Geometry of behavioral spaces: A computational approach to analysis and understanding of agent based models and agent behaviors. Chaos: An Interdisciplinary Journal of Nonlinear Science, 26(11) (2016)

[4] Falandays, J.B., Smaldino, P.: The Emergence of Cultural Attractors: An Agent-Based Model of Collective Perceptual Alignment. Annual Meeting of the Cognitive Science Society, vol. 43, 2021)

[5] Gilbert, N., Troitzsch, K.G.: Simulation for the Social Scientist. 2 ed. Buckingham, United Kingdom, Open University Press (2005)

[6] Huberman, B.A., Glance, N.S.: Evolutionary Games and Computer Simulations. Proceedings of the National Academy of Science USA, 90(16):7716-7718 (1993)

[7] Kahneman, D., Tversky, A.: The Simulation Heuristic. In Kahneman, D., Slovic, P., Tversky, A. (eds.) Judgment under Uncertainty: Heuristics and Biases, pp. 201-208. Cambridge University Press, Cambridge, UK (1982)

[8] Mudigonda, S., Núñez-Corrales, S., Venkatachalapathy, R., Graham, J.: Scheduler dependencies in Agent-Based Models: A case-study using a contagion model. Computational Social Science Society of the Americas, Springer, Santa Fe, NM (2021)

[9] Núñez-Corrales, S., Friesen, M., Srikanth, M., Venkatachalapathy, R., Graham, J.: In-Silico models with greater fidelity to social processes: towards ABM platforms with realistic concurrency. Computational Social Science Society of the Americas, Springer, Santa Fe, NM (2020)

[10] Parunak, H.V.D., Brueckner, S.: Concurrent Modeling of Alternative Worlds with Polyagents. the Seventh International Workshop on Multi-Agent-Based Simulation (MABS06, at AAMAS06), pages 128-141, Springer, Hakodate, Japan (2006)

[11] Parunak, H.V.D., Morell, J.A., Sappelsa, L., Greanya, J.: SCAMP User Manual. Parallax Advanced Research, Beavercreek, OH (2020). https://www.abcresearch.org/abc/papers/SCAMPUserManual.zip

[12] Parunak, H.V.D.: Learning Actor Preferences by Evolution. Computational Social Science (CSS21), CSSSA, Santa Fe, NM (2021)

[13] Parunak, H.V.D., Greanya, J., McCarthy, M., Morell, J.A., Nadella, S., Sappelsa, L.: SCAMP's Stigmergic Model of Social Conflict. Computational and Mathematical Organization Theory, (2021)

[14] Sappelsa, L., Parunak, H.V.D., Brueckner, S.: The Generic Narrative Space Model as an Intelligence Analysis Tool. American Intelligence Journal, 31(2):69-78 (2014)

[15] Savit, R., Brueckner, S.A., Parunak, H.V.D., Sauter, J.: General Structure of Resource Allocation Games. Altarum, Ann Arbor, MI (2002). https://www.abcresearch.org/abc/papers/RAGpaper.pdf

[16] Shnerb, N.M., Louzoun, Y., Bettelheim, E., Solomon, S.: The importance of being discrete: Life always wins on the surface. Proc. Natl. Acad. Sci. USA, 97(19 (September 12)):10322-10324 (2000)

[17] Tukey, J.W.: Exploratory Data Analysis. Addison-Wesley (1977)

[18] Wilson, W.G.: Resolving Discrepancies between Deterministic Population Models and Individual-Based Simulations. American Naturalist, 151(2):116-134 (1998)

[19] Wolfram, S.: Cellular Automata and Complexity: Collected Papers. Reading, MA, Addison-Wesley (1994)

[20] Zia, A., Koliba, C.: The emergence of attractors under multi-level institutional designs: agent-based modeling of intergovernmental decision making for funding transportation projects. AI & Society, 30(3):315-331 (2015)