COMPLEX ADAPTIVE SYSTEMS AND THE THRESHOLD EFFECT:
TOWARDS A GENERAL TOOL FOR STUDYING DYNAMIC PHENOMENA
ACROSS DIVERSE DOMAINS

by

Theodore David Carmichael

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2010

Approved by:

____
Dr. Mirsad Hadzikadic

____
Dr. Moutaz Khouja

____
Dr. Zbigniew Ras

____
Dr. Joseph Whitmeyer

____
Dr. Xintao Wu

ABSTRACT

THEODORE DAVID CARMICHAEL. Complex Adaptive Systems and the threshold effect: towards a general tool for studying dynamic phenomena across diverse domains. (Under the direction of DR. MIRSAD HADZIKADIC)

Most interesting phenomena in natural and social systems include transitions and oscillations among their various phases. A new phase begins when the system reaches a threshold that marks a qualitative change in system characteristics. These threshold effects are found all around us. In economics, this could be movement from a bull market to a bear market; in sociology, it could be the spread of political dissent, culminating in rebellion; in biology, the immune response to infection or disease as the body moves from sickness to health. Complex Adaptive Systems (CAS) has proven to be a powerful framework for exploring these and other related phenomena. Our hypothesis is that by modeling differing complex systems we can use the known causes and mechanisms in one domain to gain insight into the controlling properties of similar effects in another domain. To that end, we have created a general CAS model; one that is flexible enough so that it can be individually tailored and mapped to phenomena in various domains, yet retains sufficient commonality across applications to facilitate a deeper, cross-disciplinary understanding of these phenomena. In this work, we focus on the threshold effect. We show that the general model successfully replicates key features of a CAS. And we demonstrate its general applicability by adapting the model to three domains: cancer cells and the immune response; political dissent in a polity; and a marine ecosystem.

DEDICATION AND ACKNOWLEDGEMENTS

I would like to dedicate this work to my parents, Ann and Don Carmichael, for instilling in me the belief that I am fully capable of finding the answers to any of life's puzzles; and to my brother, Taylor Carmichael, for constantly challenging that belief and making me work harder for those answers.

I would also like to gratefully acknowledge: Dr. Mirsad Hadzikadic, who is a gifted and dedicated mentor, and whose extraordinary encouragement, support, and guidance over the years has been invaluable; the members of my dissertation committee, who have shown intelligence, patience, and good humor throughout this process; and the UNC Charlotte CAS Research Group, in particular Drs. Didier Dréau and Joseph Whitmeyer for their collaboration on biological and sociological processes which are found herein.

The research that led to this dissertation was financially supported by UNC Charlotte's Graduate Assistant Support Plan, by the College of Computing and Informatics through Teaching Assistantships and special funding for travel to various conferences, and by the Defense Advanced Research Projects Agency.

All of the models described in this work were created using the NetLogo modeling environment; designed and supported by the Center for Connected Learning and Computer-Based Modeling at Northwestern University, Evanston, Illinois.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Most interesting phenomena in natural and social systems include transitions and oscillations among their various phases. A new phase begins when the system reaches a threshold that marks a qualitative change in system characteristics. These threshold effects are found all around us. In economics, this could be movement from a bull market to a bear market; in sociology, it could be the spread of political dissent, culminating in rebellion; in biology, the immune response to infection or disease as the body moves from sickness to health.

Companies, societies, markets, or humans rarely stay in a stable, predictable state for long. Randomness, power laws, and human behavior ensure that the future is both unknown and challenging. How do events unfold? When do they take hold? Why do some initial events cause an avalanche while others do not? What characterizes these events? What are the thresholds that differentiate a sea change from insignificant variation?

Complex Adaptive Systems (CAS) has proven to be a powerful framework for exploring these and other related phenomena. As the name implies, a CAS is a system of agents that interact among themselves and/or their environment, such that even relatively simple agents with simple rules of behavior can produce emergent, complex behavior. The key to CAS is that the system-level properties cannot be understood, or often even defined, at the level of the individual agent description. Therefore, these systems must be studied holistically, as the sum of the agents and their interactions.

We characterize a general CAS model as having a significant number of self-similar agents that:

- Utilize one or more levels of feedback;
- Exhibit emergent properties and self-organization;
- Produce non-linear dynamic behavior.

The CAS framework can be used to describe systems that encompass phenomena across many diverse environments and a wide range of disciplines. These systems are present at all scales of inquiry: from the movement of markets and economies to individual knowledge acquisition; from large-scale social interaction to small-scale cellular behavior. Advances in modeling and computing technology have not only led to a deeper understanding of complex systems in many areas, but have also raised the possibility that similar fundamental principles may be at work across domains, even though each of these systems manifest themselves differently due to the peculiarities of their environments.

Our hypothesis is that by modeling differing complex systems we can use the known causes and mechanisms in one domain to gain insight into the controlling properties of similar effects in another domain. To that end, we have created a CAS-based model so that it can be individually tailored and mapped to phenomena in various domains. This model encompasses all the key characteristics of CAS described above.

1.1 Research Goals

While there have been many implementations of CAS models used in various domains, these models are usually created in isolation, such that an economics model is used only to study market effects and dynamics, or a population model is used to describe and predict the causes and limits of population growth. There have also been examples of ideas and concepts inherent in CAS being transferred from one field to another. However, there has not been a general CAS model developed that is intended to replicate phenomena simultaneously in more than one field or area of inquiry. As Neil Johnson writes:

> "In particular, the connections between such systems have not been properly explored – particularly between systems taken from different disciplines such as biology and sociology. Indeed it is fascinating to see if any insight gained from having partially understood one system, say from biology, can help us in a completely different discipline, say economics" (John07, 16).

Put another way, Epstein writes: "Generality, while a commendable impulse, is not of paramount concern to agent-based modelers at this point" [Epst06]. Therefore, the main research goal of this work is to pursue such generality; to look closely at some of these connections across domains by working towards a general CAS model, one that can serve as a common language, even for fields that are far apart. We do this in the context of threshold effects, a phenomena common to many domains. By specifically identifying general properties in a common framework this CAS tool aims to: 1) facilitate the transfer of knowledge from one domain to another; and 2) stimulate a deeper understanding of the properties in one system by using the general model for mapping like properties from another system.

This is, by necessity, a challenging and long-term goal. In its current state, CAS modeling is more art than science, and it is difficult to capture all the salient features of even a single complex system. However, there is a tremendous amount of potential in moving towards a "language of CAS" that is widely applicable. What we show here is not intended to be a final solution, but rather a significant first step towards the development of this language.

1.2 Background Information

There have been numerous examples of using ABM (Agent-based Modeling) to implement a CAS framework, in order to further understanding of the dynamics found within particular complex systems. In [Midg07] the authors construct a model that aims to reproduce a typical market structure by utilizing the properties of a supermarket setting. Their model incorporates three types of agents: consumers, retailers, and manufacturers. They have chosen ABM over more traditional methods of model construction that use game theory or analytical equations of system dynamics, due to the power and flexibility of CAS. "[O]ne can more easily incorporate the existing knowledge about the nature of human-decision-making processes into AB models than into analytical equations. […] AB models allow a flexibility of representation that is not present in more traditional approaches."

But this model was not designed with general applicability in mind. It may be that some of the agent attributes could reasonably be applied to other domains: "chance of observing a store promotion" might be a stand in for "vision;" "number of best promotions remembered" may be generalized as "memory;" and perhaps "satisfaction threshold" for an agent could represent any state-change threshold for any agent. But it is not explicitly explored how these translations may be realized, or to what advantage, in a different system. Other attributes such as “range of advertising levels” or “quarterly increment/decrement to mark-up” may not have any obvious analogues. Furthermore, the rules governing calculations that utilize these attributes also suffer for a lack of generalizability or an explicit method for applying these rules to a new domain.

In [Tay05] the authors present a more general economic model, utilizing only two agents: buyers and sellers. While this work is intended to demonstrate the utility of ABM in this context, it is quite clear that these agents may be easily applicable to many types of markets. However, as with [Midg07], there is no discussion or representation of this model’s applicability to systems that are outside of economics.

Examples from other domains also follow this common pattern. Vries and Biesmeijer have created an ABM of honeybee foraging [Vrie98], which they expanded upon in [Vrie02]. While this work is intended to utilize enough flexibility to represent a broad range of variable values found in real-world honeybee colonies, it does not purport to show general adaptability to other fields. Similarly, ABM has been used to develop sophisticated tools for the study of traffic flow under a wide spectrum of environmental factors, such as weather, infrastructure, and changing demographics. [Erol10] describes one such system; but again, limited to only a single domain.

There have also been examples of ideas or concepts of CAS taken from one domain and applied to one or more others. Schelling’s classic model on segregation [Sche71] is an example of a fundamental property, one that may be readily applied to many systems, informing models found in sociology, biology, or economics. Flocking behavior has been studied in birds, fish, and crowds of people, and simple analogies between these diverse systems can be drawn [Sump06]. Also, the collective intelligence of ants for determining the shortest path has proven to be useful in the engineering of decentralized flow control, such as in computer networks. In general, these examples illustrate how one system can inform study of another: either by drawing comparisons from one model to another, or by using certain properties found in one model to inform the construction of a second model. Our work here is distinct, in that we will use a single CAS tool to replicate key threshold properties as found in multiple domains. In other words, we are working towards a single model that is capable of simultaneous simulation in diverse fields, rather than using common properties to develop multiple models.

This endeavor is similar in scope to the work of Nicolis and Prigongine [Nico77]. As described by [Sump06], they were attempting to develop a rigorous theory of self-organizing behavior, and they were successful in showing that mathematical equations used to describe chemical reactions could also apply to the cyclical dynamics of a predator-prey model. However, their approach did not use a stochastic ABM method, but rather relied on idealized equations which – though useful – are difficult for representing a diversity of agents and agent-attributes.

Finally, our approach is most similar to that used previously by Axelrod, et. al. [Axel06b]. In this work, a model of political state-level alliances during World War II was successfully applied to an economics system of company-level alliances. In the political arena, five attributes – such as shared religion or border disputes – were used as either attractors or repulsers in a pair-wise calculation of affinity across 17 countries. These affinity calculations – 65,536 in total – would then determine the alliances of each country (subsequently labeled either Allies or Axis). No matter what the initial conditions, only one of two final configurations appeared each time, one of which was correct for all 17 countries save one. This was then applied to the case of eight computer companies choosing which coalition to support between two competing versions of the UNIX operating system. This application used the same theory as that for the political model, simply adapting the attributes and relative sizes of each actor, and the model successfully predicted the real-world strategic alliances that the computer companies formed.

Although the success of their model represents an important instance of a single model being used in two domains, it varies significantly from our work here. The primary difficulty with [Axel06b] is that there are so few agents in each system: seventeen for the political case and eight for the business case. This opens up the model to criticism in terms of attributes that can, perhaps, be easily calibrated to predict a known result. Furthermore, this system is not intended to simulate the machinations of the countries or the companies over time; rather, it merely searches for a single end state.

While their model may certainly be useful in categorizing and understanding an important past event, and may therefore be applicable for predicting similar future events, such an endeavor would require repeated iterations as applied to multiple similar events. In this way, their model could be more finely tuned in terms of the weights of each attribute as either an attractor or a repulser. However, it is unclear how a set of these weights in one domain – political alliances – would help inform similar weights in another domain, such as corporate alliances.

Nevertheless, the strength of their work is that the model's interactions are translatable from one domain to another, particularly in terms of the underlying theory used in both cases. We extend these results in three important ways. First, our model uses hundreds or thousands of actors, rather than just a few. This necessarily gives more emphases to the common traits among actors, and the resultant emergent properties that represent the collective behavior of the overall system, not the particular behaviors of a handful of actors. Second, the attributes of the agents in our model are much more generalizable and more easily mapped from one domain to another. For example, our agents have attributes such as speed, number of connections, and lifetime, whereas the agents in Axelrod, et. al., have what are essentially five measures of "stress" to calculate the pair-wise affinity between actors. Finally, our model uses the temporal simulation outputs to show not just an end result, but rather the system dynamics over time. This is in many ways a more difficult process, as the model outputs reflect not just the final configuration of the agents, but also the path those agents collectively take to reach the final state.

We achieve these advances by focusing on the threshold effect: a phenomena found across systems and across disciplines. Section (2) discusses our approach in broad terms: a focus on threshold effects and the realization of CAS properties to simulate these effect. This section therefore describes and defines thresholds, and characterizes thresholds into three different types. Here we also present the background of CAS and complexity studies, and define in more depth the key properties of a CAS as we have interpreted it. In section (3) we summarize previous research we have undertaken with colleagues that was used to inform our approach. Section (4) represents an overview of the development of the general model, gives the specifications for this tool and our methodological principles, and details how its implementation demonstrates key features found in section (2).

In section (5) we present the evaluation of the general tool by presenting our results in mapping it to three domains: soft-tissue cancer; political dissent in a polity; and a marine ecosystem. Section (6) is used to summarize the key results and outcomes of this research, discusses its intellectual merits and significance, and presents future directions for research based upon this work.

Section (7) concludes with an overview of our key motivations, highlighting the importance of cross-disciplinary tools and methods, and summarizing how our general model fulfills this role, facilitating cross-disciplinary communication by providing a common language and methodology that can be applied to threshold effects across many different domains.

CHAPTER 2: APPROACH

2.1 Thresholds

We define a threshold effect as a change in sign or abrupt change in magnitude (either enduring or a spike) in the first or second derivative of a system variable. We characterize three distinct threshold processes: 1) the ratchet mechanism, 2) cumulative causation, and 3) contagion.

The ratchet (or "lock in") mechanism is defined as follows: once an increase in X produces a change in Y, it is easier to continue to increase Y than to decrease Y. Example: an increase in X of one unit, in time T1, produces an increase of Y of one unit. In time T2, X decreases by one unit, but Y does not decrease.

The mechanism of cumulative causation follows the following rules: 1) the full effect of X on Y is not immediate; 2) below the threshold, the influence of X on Y is small; and 3) a threshold is reached when an additional change in X results in a large change in Y.

In the contagion mechanism, agents choose between options X and Y. The agent's choice is influenced by the choices of other agents in its neighborhood or network.

A clear example of a threshold effect can be found in the behavior of cancer cells. Once a cancerous cell is produced, it begins to proliferate, creating more cancer cells in its neighborhood. We identify the growth of cancer as the ratchet mechanism. Initially, the cancer cells have few negative health consequences. They are limited partially by the immune cell response, and partially due to reduced angiogenesis (that is, the growth of new blood vessels to feed the cancer cells). But when a threshold is reached, the cancer cells have an increased growth rate, begin to adversely affect overall health, and this growth overcomes the body's natural negative feedbacks on the cancerous growth, overwhelming the immune system's role.

A second example regards the trajectory of political dissent in a population. Here, the contagion model is appropriate. Each citizen is either dissenting or not dissenting. If an agent does protest, then that affects the other agents in his neighborhood by encouraging them to dissent. This represents the key feature of a contagion model – the positive feedback (influence) from one agent to another. There are also government agents that work to quell dissent; the more government agents there are, the more dissent is suppressed. However, the number of government agents is constrained by the total resources available to the government, which in turn is negatively affected by the amount of dissent. Therefore, if the level of dissent becomes sufficiently high, then the government lacks the resources to deploy suppression agents.

Thus, this model allows for analysis of multiple potential thresholds, including: 1) a start-up threshold of dissent, at which point contagion-type spreading occurs; 2) a turning point threshold of the relative numbers of dissenters and government agents, representative of the "tipping point" threshold as with the ratchet effect in soft-tissue cancer; 3) a government success threshold of dissent, which reverses the contagion in (1); and 4) a dissent success threshold, when the tipping point of (3) is surpassed and the system collapses.

2.2 Complex Adaptive Systems

2.2.1 Historical Context

In the 1960's researchers were trying to better understand the dynamics of slime mold: in particular, there was a persistent mystery in how it could transition between its active and its dormant states (John02, 12-17). Biologists had long known about slime mold's strange behavior, acting as a single organism under some conditions, and devolving into individual cells under other conditions. They knew that the chemical acrasin was somehow involved, and speculated that there were "pacemaker" cells which would produce acrasin and thereby attracted the other cells to it. Years of study were conducted in the vain search for these pacemakers.

In the late 1960's a physicist and a mathematician (Evelyn Keller and Lee Segal) came across a paper by Alan Turing that described what he termed "morphogenesis:" the idea that organisms can form great complexity from simple roots. Published in 1954, it was one of the last papers he produced, and in it he described a mathematical model whereby simple organisms, following just a few simple rules, could produce strikingly complex patterns.

Keller and Segal took the ideas in Turing's paper and developed the mathematics to describe a system of slime mold, demonstrating that it is not necessary to account for pacemaker cells in such a model. Rather, all that was required to reproduce the properties of the system were two rules: that each cell simultaneously produces (rule #1), and is attracted to (rule #2), acrasin. These two simple rules were sufficient to account for the mold's strange behavior, showing how this collective interaction could allow numerous individual cells to form a multi-cellular organism, one that could move about its environment and act like a single living being.

In this way, the description of a slime-mold model exhibits all the classic properties of a CAS: the agents, or cells, of the slime-mold affect each other via the feedback mechanisms as represented by the two rules; they also react to the influence of the changing environment, which is sufficient to activate these rules; once activated, the cells self-organize as an emergent property of this system; and finally, the threshold change in behavior of the slime-mold organism represents the non-linear dynamics necessary to adapt to new environmental conditions.

This re-framing of the slime-mold behavior is indicative of a systems-level approach to studying complex phenomena. This framework was recognized as a new way to approach other system-level phenomena in many other fields, such as the classic “invisible hand” that governs the marketplace, as found in the work of economist Adam Smith. The subsequent founding of the Santa Fe Institute in 1984 by Murray Gell-Mann, a physicist; John Holland, a biologist; and others, is seen by many as the beginning of CAS as an explicit field of study [ref. Waldrop]. They recognized the multidisciplinary nature of these phenomena, and thus brought together scholars from many different areas to begin the process of applying CAS to a wide variety of research questions.

2.2.2 Complexity

There is not yet a single, agreed-upon theory that describes “complexity” or a “complex system” equally for every situation. As with many things, it is often a matter of degree or perspective, rather than clear distinction, as to what is complex and what is not. However, we can distinguish some key characteristics of a complex system for our purposes here.

The most general distinction we use refers to Warren Weaver's division of complexity into two types: disorganized complexity and organized complexity [Weav48]. Disorganized complexity refers to a system of many – even millions – of parts that interact at random, producing aggregate effects that can be described using probability and statistical methods. The example he gives is that of a very large billiard table with millions of balls rolling in different directions, colliding with each other and with the walls. Even though the path of a single ball may be erratic, or even unknown, the system itself has measurable average properties. Clearly, there is feedback in such a system: one billiard ball strikes another, and then that ball can bounce around and strike back. But this does not suffice. There is something missing in this system that would cause the feedback amongst the billiard balls to produce self-organizing behavior.

What we are concerned with here, then, is *organized complexity*. Organized complexity refers to a system with a sizable number of factors which have *correlated* interactions; furthermore, these correlated interactions produce emergent, global properties. "An average quantity alone is not an emergent feature. Yet statistical quantities which define properties of an aggregation can be regarded as simple emergent properties, if they depend on a relation of the particles to each other, i.e. if they do not make sense for a single particle" [From05, 8].

Correlation among the interactions in such a system implies two things: 1) that the agents of the system exhibit feedback mechanisms; and 2) that these feedback mechanisms are, by definition, endogenous to the system itself, so that the agents affect each other in a correlated manner.

2.2.3 Agents

The term 'agent' tends to be an overloaded one. Some researchers, therefore, may use an alternative, such as "particle" [Kenn01] to describe the individual objects of a complex system. While logically sound in the way this is presented, it doesn't seem to capture the autonomy, or intent, of many agents, particularly those found in social systems. Thus we use the more conventional term 'agent' in our description, but we distinguish between the – somewhat overlapping – conceptions of agents found in CAS relative to those generally described in a MAS (Multi-Agent System) [Wool02]. We define CAS agents as possessing simple rules and attributes; as being largely autonomous with only local knowledge; and as being components of a system that could be replaced by similar components without disrupting the emergent features of that system. In contrast, MAS agents are generally more autonomous and intelligent, more complicated, and fewer in number. Furthermore, emergent properties of most MAS models are usually highlighted as something to avoid, rather than some inherent, key property of the system.

In our work we also consider CAS agents to be self-similar, to use a term common in the literature; i.e., the agents are largely homogenous. It is worth noting that many published works refer to these not as homogenous agents, but as heterogeneous agents, such as in Epstein [Epst07, 5-6]. We believe the discrepancy is largely a difference in semantics and emphasis. Nevertheless, for the sake of clarity, it is worth exploring various levels of distinction between agents as found in a CAS.

CAS agents must be, at the very least, different spatially or temporally. Without these differences, there would clearly be no meaningful interaction; nor would there be a way to differentiate among them. So using the term ‘homogenous’ simply indicates great similarity – even exact similarity – among the agents’ rules and attributes, while it is understood that each agent represents a different current state. These agents can be identical in every way except their current state. Even thus, they are still quite capable of producing emergent features, based on the correlated differences across these various agent-states, and the aggregate or global properties that result from these agents’ interactions. Computer simulations of traffic patterns, flocking birds, or Axtell’s The Game of Life exhibit various emergent properties even though the specifications for the agents are exactly the same across all cars, and birds, and for each grid-cell in the Game of Life.

As Epstein uses the term “heterogeneous,” he is generally referring to a differentiation in terms of the agent specifications themselves; that is, a difference in their rules and/or their attributes. While others may say that these agents are “largely” homogenous, we use the term “self-similar” simply to avoid ambiguity, while recognizing that the agents can be different - but not *too* different – in terms of the rules and attributes that relate to the emergent property in question.

These differences across agents do matter, in their variety, because a particular emergent property depends upon some degree of self-similarity within the system. Consider a simple traffic flow example, with the agents as cars moving along a highway. Each agent has two rules: slow down if the car ahead is too close, and speed up if it is too far away. Under some conditions, a wave-like pattern can emerge across the ebb and flow of the cars, as one car slows, causing the next in line to slow, also. In simulations this can occur whether the rules for slowing down and speeding up are exactly the same across all cars, or if there is some slight variation for the activation of each rule.

But if some agents have rules that allow them stop completely, or crash, or drive off the road, then this chaotic behavior would disrupt the emergent patterns of traffic. There is a breakdown in the system at the point where an agent diverges too far. In a similar manner, if the flocking example found in [Wile98] were adjusted so that some agents have wildly different attributes, then "flocking" may not be a reachable state for the system.

The degree to which agent must be similar depends upon the characteristics of the model being studied; specifically, it depends on the emergent behavior that is of interest. For example, the agents in the traffic pattern may be made much more complex, with many more attributes, than two simple rules of when to speed up and when to slow down. Each agent's perceptions, disposition, reactive ability, and etc., could be included in the specifications But these this case – in terms of the emergence of waves of congestion along the highway – these attributes, and many more, only matter to the degree that they relate to the two conditions that produce the emergent behavior. The agents themselves, therefore, may be described as quite heterogeneous, but the relevant attributes must still be self-similar enough to produce a traffic pattern that can be analyzed and compared to real-world data.

2.2.4 Agent-level vs. System-level Adaptation

Notice also that if all the agents in a particular model have exactly the same rules and attributes, then they cannot be thought of as adaptive at the individual level. (At least, they are not adapting as long as these rules and attributes remain completely homogenous across the population.) "Adaptation" implies some sort of fitness function or selection of agents based on their attributes, which implies at least some difference or capacity for change among these attributes.

Agent level adaptation becomes hard to distinguish under certain conditions, however. In order to illustrate the potential difficulty, one could imagine an economics model where agents buy or sell a certain good at a certain price. The agents each have a rule that states: buy product *X* if it costs no more than *Y* units of money. Thus on one level, the agents are exactly the same, in that their internal rules are the same. But one agent's current state for the value of *Y* may be 10 units, while another agent may have his *Y* set to 11 units. The difference between them is simply a matter of variation of local conditions between the two agents. In one sense, these agents are still homogenous, because they have the same type of rules, and they apply these rules in the same way; as with spatial or temporal properties, the agents differ only in their current state for the value of *Y*. In another – but very real – sense, these agents are adapting individually, since the price point for each agent may vary.

Agents can adapt individually on a higher level as well. For example, the rules themselves may change for individual agents, so that even if two agents are in exactly the same local situation, they may react to that situation differently.

In general, we will use changes in agent attribute-values to be related to system-level adaptation – i.e., the system as a whole reacts and adapts to its environment – and changes in the set of agent rules or attributes to be related to agent-level adaptation. In this way, the complex system is said to adapt to an environment simply because each individual agent reacts to its local environment in a pre-determined way. Our model uses only this system level adaptation currently; however, more complex behavior resulting from agent-level learning – i.e., adapting or changing rules as well as attributes – can easily be incorporated into this model for future study or added flexibility.

2.2.5 Feedback

Feedback, simply defined, means that the outputs of a system at time *t* affect the inputs of that system at time *t*+1. As the agents in a complex system interact, the results of some interactions may influence future interactions. It is this influence that represents the feedback within the system itself. In the previously mentioned model of traffic patterns along a highway, one car that slows down in response to the car in front of it may then produce a similar effect in the next car in line. This action/response that can easily produce a wave of congestion along the highway is due to feedback between the cars, from one to the next in line. It is worth pointing out that the term "wave" is apt in this case, as it describes a pattern of behavior across multiple agents, much like a wave in the ocean, even though the agents participating in the pattern change over time. This matches well with how Holland and others have described emergence in complex systems: "Emergent phenomena in generated systems are, typically, persistent patterns with changing components" [Holl99, 225].

Note also the distinction between this organized feedback as compared to the disorganized complexity of our billiard table. While it is true that one collision between two balls alters the course of future collisions, it does not affect the course of future collisions in a persistent way; that is, if one colliding ball happens to bounce to the north, it does not mean that the next ball struck will also bounce northward. "Relationships in these systems are mutual: you influence your neighbors, and your neighbors influence you. All emergent systems are built out of this kind of feedback" [John02, 120].

The key point here is that such reciprocal influence among neighbors is more significant when it creates measurable, global properties. The action/reaction patterns represent the correlations within the system that make up these global properties. While our traffic pattern example may have measurable statistical properties – such as how many cars traverse the highway in a given day – these measurements do not fully capture the wave-like behavior of the system. It is by identifying the correlated feedback that we find a richer, and therefore more interesting, description of the system.

2.2.6 Endogenous effects

One may want to consider the first action that sets the pattern in motion -- is it an endogenous or exogenous instigator? While the pattern is certainly endogenous to the system, the initiation of that pattern may be either. It can sometimes be difficult to characterize effects as one or the other, and how the system itself is defined may further confuse the distinction. However, by defining correlated feedback as a key property of a CAS, we bypass this argument in favor of defining what the feedback represents, and what it tells us about the system.

If an external effect sets off a chain reaction of persistent patterns, then the underlying properties that allow this chain reaction to occur are of distinct interest for understanding the system. If, however, there is a persistent and recognizable feedback that comes from outside of the system, then we consider this feedback to be significant in terms of our understanding of the system properties. Therefore, when we define a system, we use the method and type of feedback as a key attribute.

Consider the example of a marketplace. Such a system may encompass agents that buy and sell products, or stock in companies; it may include the concept of wealth, earnings, inflation, etc.; and it may also be affected by regulatory bodies, such as the Federal Reserve acting to tighten or loosen the conditions for borrowing. Clearly, if one defines the system as only the agents and how they interact with each other, then the actions of a Federal Reserve would be exogenous to this system. However, these actions by the Federal Reserve – whatever they may be – are clearly influenced by the state of the market. Furthermore, they are likewise designed to influence the future state of that market. This is a significant level of feedback that should be accounted for when studying the "system," i.e., the market.

2.2.7 The Environment of the System

Another way of stating the idea of exogenous factors is to say that feedback goes both ways: the agents affect the environment even while the environment affects the agents. This is distinct from a model of, say, an ecology which has sunlight as an external factor. The sun cycles through day and night, as well as annual cycles of summer and winter, and these cycles generally affect the behavior of most ecological systems. But the agents in this system cannot likewise affect the behavior of the sun. So while defining what encompasses a "system," and what potential factors are internal or external to that system, it is more important to note the level of feedback that exists between those factors, as this is both definitional and functional to the system being studied.

The type or existence of feedback suffices even with very broad definitions of "environment." If the environment for one driver-agent is defined as the road as well as all the other agents, it is the distinction between levels of feedback that is the more germane characteristic. In the models that we have developed and are described in subsequent sections, these characteristics are explicitly defined.

2.2.8 Emergence and Self-Organization

The term "emergence," like complexity, has not yet reached a consensus definition. Some researchers distinguish between weak emergence and strong emergence, and use this definition as representing a fundamental law.

"If there are phenomena that are *strongly emergent* [emphasis added] with respect to the domain of physics, then our conception of nature needs to be expanded to accommodate them. That is, if there are phenomena whose existence is not deducible from the facts about the exact distribution of particles and fields throughout space and [time] (along with the laws of physics), then this suggests that new fundamental laws of nature are needed to explain these phenomena" [Cham02].

This idea would seem to indicate that a strongly emergent property is similar to the idea of gravity: gravity is a fundamental law, a property of matter; but gravity is only apparent as one particle relates to another. In this view, it is not that the rule cannot be modeled by the agent, but rather it cannot be understood except in terms of other agents.

In our definition of emergent behavior we adopt this idea of relations among agents in the system, as in the way we have previously defined correlated interactions. A traffic "pattern" cannot really exist with only one car, and a colony of ants cannot be said to find food if there is only one ant. In this way, emergent behavior is a property of a system that is at a different scale than the parts of the system [Ryan07]. In a similar vein, emergence is the macro-level behavior that is not defined at the macro-level, but rather depends upon the rules and interactions of agents defined at the micro-level.

Consider a few examples of typical emergent behavior, in respect to the systems they stem from. There are the cars as agents, in the example cited previously. There is also the example of bees or ants, following simple rules to forge for food or build a nest. Johnson talks at length about the city of Manchester, England, during the 19th century [John02, 33-40]. He uses it to illustrate how a city with tens of thousands of people, yet absolutely no central planning, still managed to organize itself in distinct patterns, such as areas of the working class separate from the nicer middle-class neighborhoods.

"The city is complex because … it has a coherent personality, a personality that self-organizes out of millions of individual decisions, a global order built out of local interactions" [John02, 39].

The brain is also often cited as a complex, adaptive system, with intelligence (or even some sub-set of intelligence, such as vision) as an emergent feature. In our CAS model, we will look at a number of emergent features, such as the self-organization of the agents and the aggregate behavior of the system.

The "self" in self-organization refers to the state of an individual agent in a complex system. This agent follows its own local rules, and uses its own attributes in applying those rules. Let us consider a simple model of an ant colony. For the purposes of illustration, this model need not be realistic. Assume each individual ant has the same three rules: 1) search randomly across the environment for food; 2) if you find food, carry it back to the colony and leave a "food" trail; 3) if you find a food trail, follow it until you find food.

If one ant finds food, then this new attribute – "I have food" – activates the rule to carry a piece of the food back to the colony and leave a food trail. Now, by leaving the food trail, this ant can affect the current state of any other ant that happens upon that trail. A new ant, finding the food trail, will activate its own rule to follow that trail to the food source, at which point it will also carry a piece back to the colony, and add to the trail. In this way a significant sub-set of the ant colony organizes itself to systematically collect the food and bring it back to the colony. The individual agents – in this case, the ants – are acting with limited knowledge and simple rules. But by providing feedback to other agents, and influencing them to act in similar ways, they produce the correlations of behavior that represent the organization of the overall system; i.e., the self-organization that emerges from these interactions, defining the local increase in complexity.

2.2.9 Natural Bias of Complex Systems

The framework of CAS directly challenges two distinct biases that tend to affect our understanding of the agents in a complex system: 1) a hierarchical bias; and 2) a complexity bias. A hierarchical bias can be illustrated by the tendency to view a complex system in terms of a leader directing the action of all the other agents. As Johnson points out, colonies of ants have previously been viewed as the queen controlling the colony as a whole; however, this fails to capture the amount of autonomy present among the other ants [John02]. In much the same way, the growth of Manchester was deeply surprising to those who thought that such distinct patterns of growth could only be achieved by directed action of some sort of governing body.

These strange phenomena – global properties of systems as represented in the growth of Manchester, or Smith's "invisible hand" theory – did not go unnoticed or unstudied. However, as with the peculiar behavior of the slime mold, researchers struggled to frame a model that could explain these global effects. The development of CAS tools and models, therefore, represent a new methodology to remedy the shortcomings of previous methods. We no longer have to assume that the behavior is directed in a hierarchical fashion, and the continued refinements in understanding these systems give us more flexibility in understanding complex systems.

CAS methods of analysis also help resist the complexity bias for agents often found when studying complex systems. This is closely related to the hierarchical bias, in that leader-agents are assumed to be more complex to account for the level of control needed in a leadership model. Also, the network amongst the agents is necessarily more robust and long-reaching, to allow for instructions to be passed to each agent in the system. If one is to assume a hierarchical system in, say, an ant colony, then the modeler must answer the question: how are orders conveyed to each worker ant?

A CAS is inherently simpler. Each ant does not need instructions; rather, they can simply be “programmed” with a few rules of behavior. In such a model, the ants don’t even have to be aware of the state of the colony as a whole; they only need to know their own current state and apply that information to their current environment. Similarly, slime-mode doesn’t need a complex pacemaker cell if a CAS model is able to replicate the organism’s complex behavior without them.

This release from both the hierarchical bias and complexity bias in the agent-level description of a system is more satisfying, as it follows Occam’s Razor: the simplest explanation for successfully describing a particular phenomenon is generally the preferred one. As with Occam’s Razor in general, the simplest explanation for a complex system is not necessarily the correct one. Their may be many model specifications that produce the correct behaviors, at least under prescribed conditions. But the simplest of these ‘correct’ models, the one that relies on the fewest number of complications, is generally a good starting point.

Furthermore, challenging the hierarchical bias also leads to a focus on agent primitives, such that features and phenomena are allowed to emerge, rather than be dictated in a top-down, complicated manner. This gives the model the inherent flexibility to simulate systems even when conditions are different than what was expected. Only if our models are inherently flexible in this way will we have a simulation that can capture results we didn’t already expect to see.

.

CHAPTER 3: PREVIOUS WORK TOWARDS A GENERAL MODEL

Here we present summaries of two projects we have undertaken with our colleagues, which have been used to inform our current work towards a general CAS tool. Each one highlights key attributes of a CAS and is implemented via an ABM. The first is a social theory testbed, realized as an agent simulation for Afghanistan; the second is a detailed model of soft-tissue cancer progression.

3.1 A Computer Simulation Laboratory For Social Theory

The Afghanistan model utilizes two agent-types: civilian agents and fighter agents [Whit08, Fern08]. The fighter agents are further sub-divided according to their particular attribute values into: government forces, coalition forces, and Taliban forces. The primary output of the simulation was a measurement of mind-share across the population of civilians. Each civilian-agent performed a calculation every simulation time step that would take into account various parameters, such as: economic health, local violence, loyalty to local leadership, and inherent predisposition. Based on the combination of these factors, each civilian would determine whether they supported the government, the Taliban, or remained neutral.

The decisions of the citizenry affect each other, as well as the ability of the two factions – the pro-government fighters and the Taliban fighters – in their relative strengths. And the fighting between these two forces can also affect the civilians, producing multiple levels of correlated feedback. Various weights, both pre-sets and customized, allow the operator to initialize the model according to different theories of social interaction. Figure 1 shows a portion of the model interface, and figure 2 illustrates the sliders used to initialize three social theory pre-sets from the social science literature: coercion theory, legitimacy theory, and representative theory.

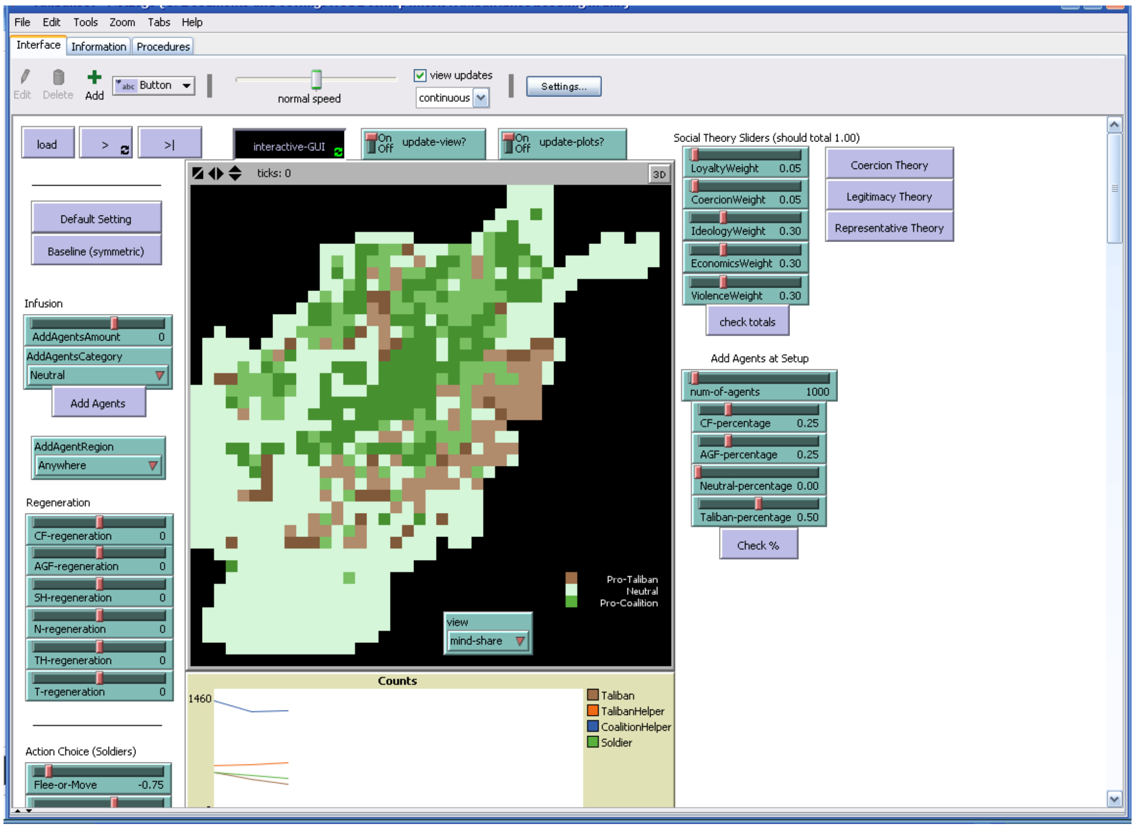

Figure 1. A snapshot of a portion of the Afghanistan model interface. The entire interface extends farther than pictured here, incorporating ~200 control mechanisms. In the top right corner five sliders are shown (figure 2) that represent one method of incorporating various social theories.

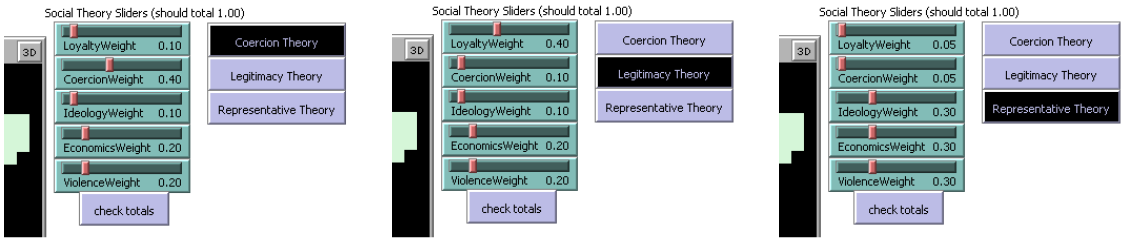

Figure 2. Details of a group of sliders used to incorporate three different pre-set social theories: Coercion Theory, Legitimacy Theory, and Representative Theory.

Only about 25 of the sliders and controls are shown in figure 1; the full interface utilizes over two hundred input control mechanisms, as well as various visualization tools and graphs. This gives the simulation environment operator flexibility, to design and test various scenarios, to input additional social theories at multiple levels of scale, and to control aspects of the model based on real-world changes of inputs. From [Whit08]: "The final result of evaluating social theories is not likely to be as simple as stating that one particular theory is best. It may be that which theory or combination of theories works best will depend on the context, differing, for example, according to historical background, religion, level of development and affluence, or some other variables."

The Afghanistan model is thus designed to be generalizable to other countries or environments. However, it is not applicable to fundamentally different systems, i.e., it is only intended to apply specifically to social science theories in the context of large populations or polities.

3.2 An Agent-Based Model of Solid Tumor Progression

This section describes previous work in developing and testing an agent-based model simulation intended to mimic the progression of solid tumors [Dréa09]. The model includes a hierarchy of active objects and attributes, including: cancer cells (initial number, division rate, cell death rate), immune cells (number of lymphocytes and macrophages, and their interactions), energy availability based on levels of vascularization, and chains of communication and memory based on cytokines and antibodies. Figure 3 is an image of the simulation interface.

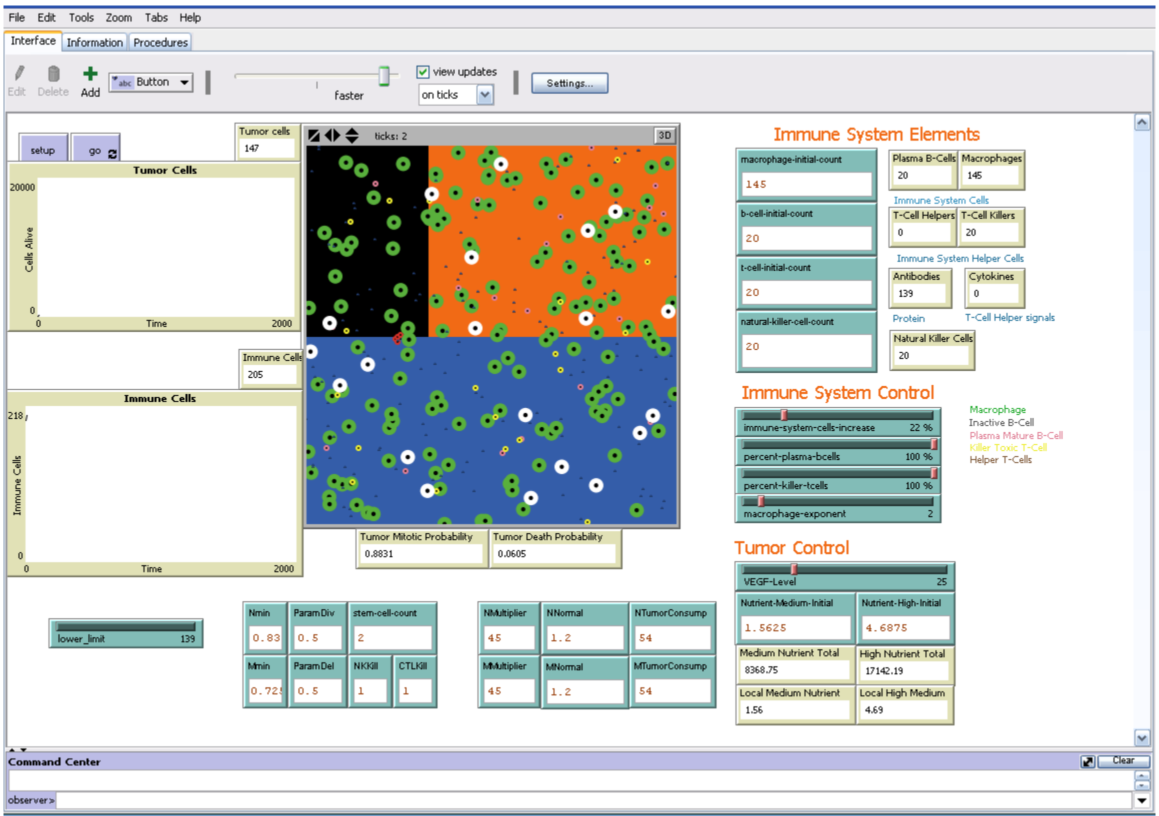

Figure 3. The tumor model interface, illustrating various controls, inputs, monitors, and visualization tools.

In figure 4 we see two close-up images of the main visualization window, which shows three levels of available nutrients based on vascularization, as well as the existent tumor and immune system cells.

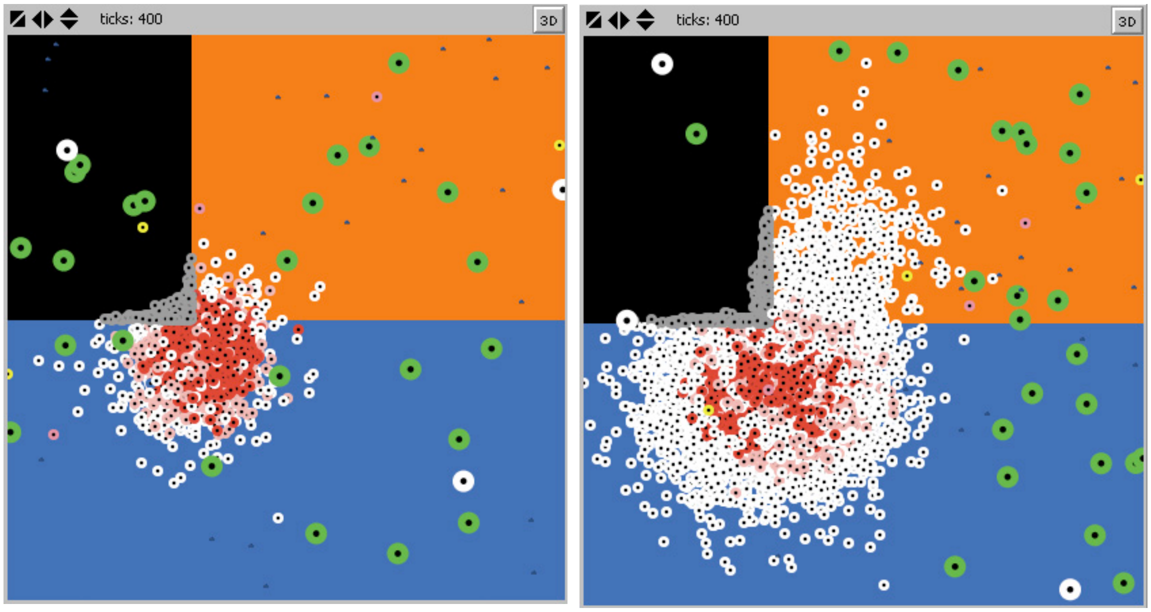

Figure 4. Two representative visualizations of the tumor growth after 400 iterations (ticks) of the simulation. The left side represents tumor growth with high nutrient needs; the right represents the same, but with low nutrient needs. The background represents zones of low (black), medium (blue) and high (orange) nutrient levels. Larger cells are immune cells (macrophages in green and lymphocytes in yellow). Tumor cells are small circles of different colors (white, pink, red) depending on their age and current state.

The novelty of this work is that it simulates the discrete interactions among all three system sub-sets – tumor cells, immune cells, and vascularization – to provide a more dynamic representation of tumor development over time. Furthermore, this system exhibits self-regulating and self-organizing characteristics that are the hallmark of CAS. Simulation results were able to demonstrate: 1) variability of the tumor growth based on higher or lower nutrient needs; 2) differences in tumor growth based on various scenarios of immune system response; and 3) the positive effects of repeated immunotherapy treatment over time.

3.3 Departure from Previous Work

These two previous applications of CAS and ABM – the population dynamics of an armed insurgency and the dynamic relationship between tumor cells, immune cells, and vascularization – share two characteristics that helped to inform our current work. The first is that these simulations were not designed to be applicable to other domains. This makes it difficult to draw lessons about complex system properties in general. It also limits researchers' ability to apply new results or fundamental principles to other domains, particularly when the fields are far apart. Thus our focus here on a model that does have this general applicability.

The second characteristic that these models share is that they were both designed to account for a multitude of variables, taking into account many of the known real-world properties of these systems. This is shown to be successful for both allowing maximum flexibility for the operator to design useful simulation experiments, as well as giving a greater degree of verisimilitude in the simulation environment and results. However, this focus on "everything but the kitchen sink" makes it difficult to deconstruct which variables and relationships have the greatest impact, and which are functionally inefficient in producing interesting outputs.

Therefore, our current work focuses on agent primitives in an attempt to build up towards the minimum number of attributes and relationships needed to: 1) exhibit the key properties of a CAS; and 2) represent the most salient characteristics of a particular complex systems to which the general model is mapped. This approach is further enhanced by the fact that focusing on primitives encourages the bottom-up generation of known global-level properties, rather than dictating these properties in a way that may unduly restrict the model. While both the models from previous work succeeded in generating these effects to some degree, our general CAS tool aims to go even farther in this regard, giving this model even more flexibility and a greater likelihood of generating surprising results.

CHAPTER 4: DEVELOPING THE GENERAL CAS MODEL

4.1 Design Principles

For the design and implementation of our general tool we followed an iterative process, adhering to the basic principles of CAS throughout each step. The first of these is that emergent properties are a feature, not a bug. If known properties of a system are allowed to emerge, rather than being constrained by a top-down approach, then we are that much closer to understanding the controlling dynamics of the system. Furthermore, such an approach is more flexible, as unanticipated behavior that is the result of low-level interactions among the agents is more likely to occur. As stated in [Midg07] (in reference to economic agent-based models):

> "[M]any current approaches are top-down, imposing analytical structures on markets that are useful to the researcher. Historical markets, however, are built bottom-up from the actions of independent agents of different types. Assuming structures, rather than allowing interactions, might artificially constrain the system in ways that are difficult to understand and which might not reflect the historical dynamics or behavior of the system."

Furthermore, this approach represents an expansion to the explanatory power of more traditional models. As Epstein describes it:

> "[I]t does *not* suffice to demonstrate that, if a society of rational (*homo economicus)* agents were placed in the pattern, no individual would unilaterally depart—the Nash equilibrium condition. Rather, to explain a pattern, one must show how a population of cognitively plausible agents, interacting under plausible rules, could actually arrive at the pattern on time scales of interest" [Epst06].

Or put another way: “If you didn’t grow it, you didn’t show it” [Epst99]. Thus our process favors a focus on agent primitives and the emergent properties of simple interactions among them, in order to “value simplicity more than theoretical sophistication in model specification” [Midg07].

The second principle that we followed was to restrict the model to only a few types of self-similar agents. In most cases, there are two distinct agents; in the final mapping of the general model – the marine ecosystems – we expanded this to three, and then four agent-types.

Finally, at each stage of development, the model has been extensively tested to ensure that all the properties of CAS, as outlined above, are present: one or more levels of feedback, emergent or self-organizing behavior, and non-linear dynamics (threshold effects). At each stage, these various properties are explored either in general terms or in regards to the salient features of the domain being simulated.

4.2 Implementation

We implemented our general CAS model, as well as the specific domain models, using the NetLogo programmable modeling environment [Wile98]. NetLogo is a Java-based framework for rapid prototyping of any agent-based simulation model. It thus acts as its own programming environment, with its own language of standard functions and descriptors. It also has a standard toolset of interface features that are modular, and can be easily incorporated into the graphical user interface of the environment. These features include: controls (sliders, buttons, and switches); outputs (monitors, plots); and the main agent environment window, which exposes the spatial positions and movements of all the agents, in a manner defined by the programmer.

NetLogo also includes: functionality for running multiple simulations in series for repeated runs with changing scenario values; a process for including Java extensions for integration with external modules; and allows for exporting models into Jave Applets, for web-based integration and dissemination.

The general framework of NetLogo includes two basic agent-types: "patches" and "turtles." The patches – square-shaped, immobile agents – form a grid and the turtles generally move across this grid. A library of functions in the NetLogo language allow the programmer to define patches, turtles, or any number of customized agents, as well as their many attributes and interactions.

In our model, the patches are the A-agents (immobile), and the turtles are the B-agents (random movement adjacent to the patches). All of our models are also modular in nature, in that the set of attributes, functions, and interactions that define an agent-type apply to every instance of that agent-type, regardless if there are fifty, or five hundred, or five thousand. We also define, using the NetLogo programming language, a number of customized functions that allow us to realize the salient features of our general model within the NetLogo framework.

4.3 First Iteration

In the first iteration of the general model, the A-agents (velocity = 0) are aligned on a grid, such that there are 125 grid cells on a side, for a total of 15,625 cells. The grid is in the form of a torus, such that the left-most cells connect continuously to the right side cells, and the top wraps around to the bottom. Each A-agents has two end states: 0 and 1; and we define the progression between the 0-state and the 1-state as having ten intermediate steps. The neighborhood for each A-agent is defined as the eight grid cells that surround that A-agent. Thus, when an A-agent reaches the 1-state, it is able to then affect neighboring A-agents towards the 1-state as well.

The B-agents affect the A-agents *away* from the 1-state and towards the 0-state. They move about the simulation randomly, until it happens upon an A-agent in the 1-state. When this happens that agent will remain there, continuing to affect that agent in steps until the 0-state is reached; at this point the B-agent is freed.

As we will see, complexity is not a stable condition in this model: either the A-agents will eventually all be changed into the 0-state by the B-agents, or the 1-state A-agents will spread the 1-state across the entire grid. (When this second condition is reached, a B-agent is still able to affect the A-agent it is directly adjacent to; however, this work is ineffective, and is continually undone by all the surrounding A-agents, so that the B-agent becomes effectively trapped in place.)

The outcome of the model stochastically depends upon the initial conditions: how many A-agents begin the simulation already in the 1-state, their initial configuration, the initial number of B-agents, and the efficiency of both types of agents.

Figure 5 shows a sample of the agent interaction. The A-agents are light-blue in the 0-state, black in the 1-state, and shades of red to show the steps between the 0- and 1-states. The B-agents are colored in shades of yellow, although there is no functional difference among them in this version of the model. Figure 6 and Figure 7 show the two inevitable outcomes: either all the A-agents end up in the 0-state and the B-agents are still free to move around (which is not reversible), or all the A-agents are in the 1-state and the B-agents are trapped.

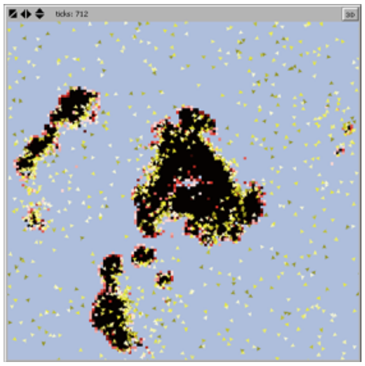

Figure 5. Dynamic agent interaction.

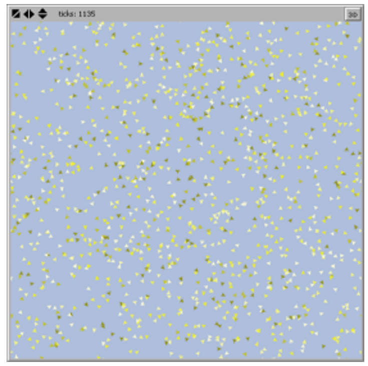

Figure 6. Outcome 1: all A-agents in the 1-state; B-agents move freely.

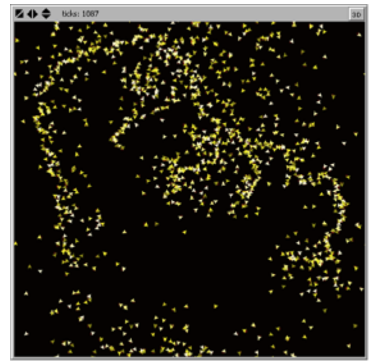

Figure 7. Outcome 2: all A-agents in the 0-state; B-agents are trapped.

Even in this example of an unstable system, the general model provides remarkably complex behavior. For example, due to the stochastic behavior of the B-agents, there are many cases where the end result cannot be predicted by the initial conditions. A small run of experiments were performed on this model across three initial configurations of 1-state A-agents: a “ring” with radius of 17 A-agents and thickness of 2; a “line” (vertical, extending from top to bottom) with thickness of 1; and a “cross” (one vertical line and one horizontal line) with thickness of 1. Note that the ring configuration initializes with a total of 196 1-state A-agents; the line configuration has 125 1-state A-agents; and the cross has 249 1-state A-agents. Each configuration was run 30 times for each level of B-agent populations. The following graphs in Figure 8 show the results:

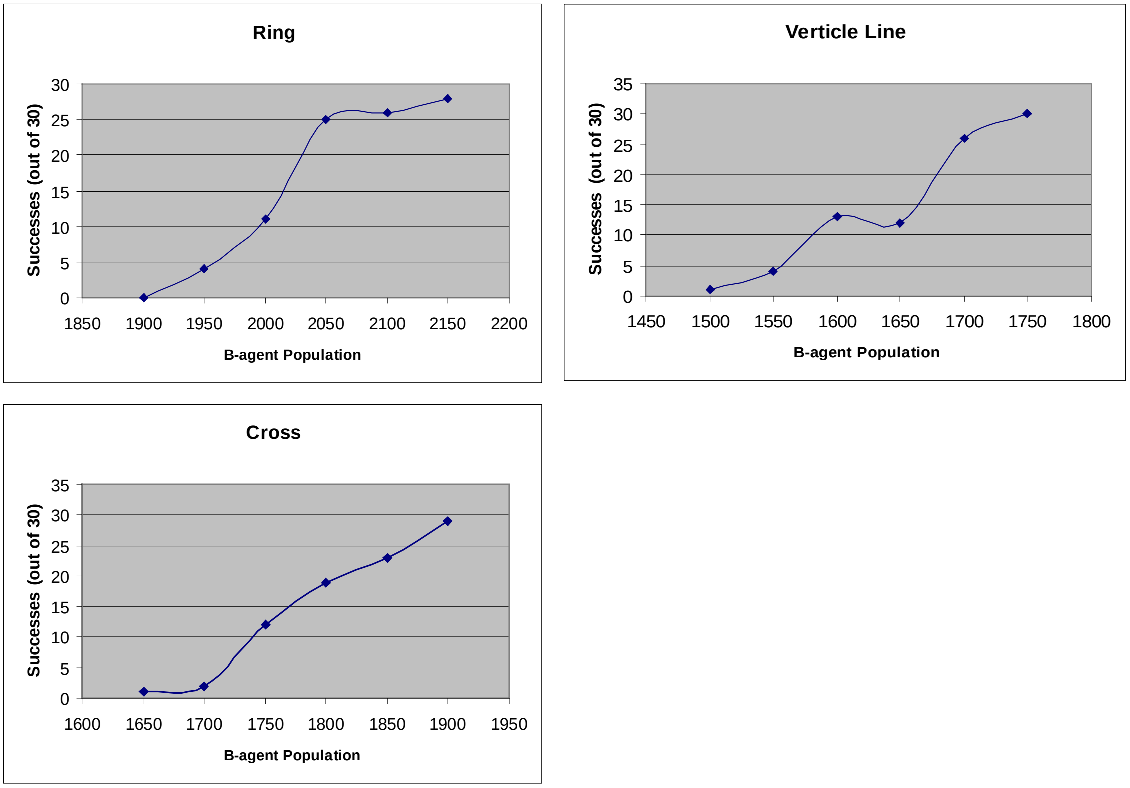

Figure 8. Each graph shows: 1) the size of initial B-agent populations (x-axis); and 2) the number of “successful” outcomes (all A-agents in the 0-state) out of 30 experiments for each level of population (y-axis).

The ring configuration is the most difficult for the B-agents to overcome; 2200 B-agents were needed for a “success” result (all A-agents in the 0-state and all B-agents remain free) in all 30 runs. The cross configuration needed only 1900 B-agents for 29 out of 30 successes, even though the cross has more initial 1-state A-agents than the ring. Not surprisingly, the vertical line required only 1750 B-agents for 30 out of 30 successes, given that the initial number of 1-state A-agents is lower and that the configuration is inherently weaker than the ring.

The variability of the outcomes is not the only complex behavior exhibited by this model. In Figure 9 below, a typical ring configuration is shown during progressive states of a single run. Note the eventual formation of a "crescent" shape of 1-state A-agents; this pattern is one that repeats itself time and again. (Although the ring configuration is shown here, the crescent shape will appear with any other initial configuration, given a sufficient number of 1-state A-agents.)

Also note how the B-agents tend towards higher concentration on the inner edge of the crescent and lower concentration on the outer edge. This is an example of the emergent, self-organizing behavior of the B-agents. What is remarkable is that this pattern occurs consistently across multiple runs, even though a B-agent cannot "see" beyond the A-agent it is directly adjacent to. Nor can the B-agents interact directly among themselves, but only indirectly, by how they individually affect the A-agents they are adjacent to.

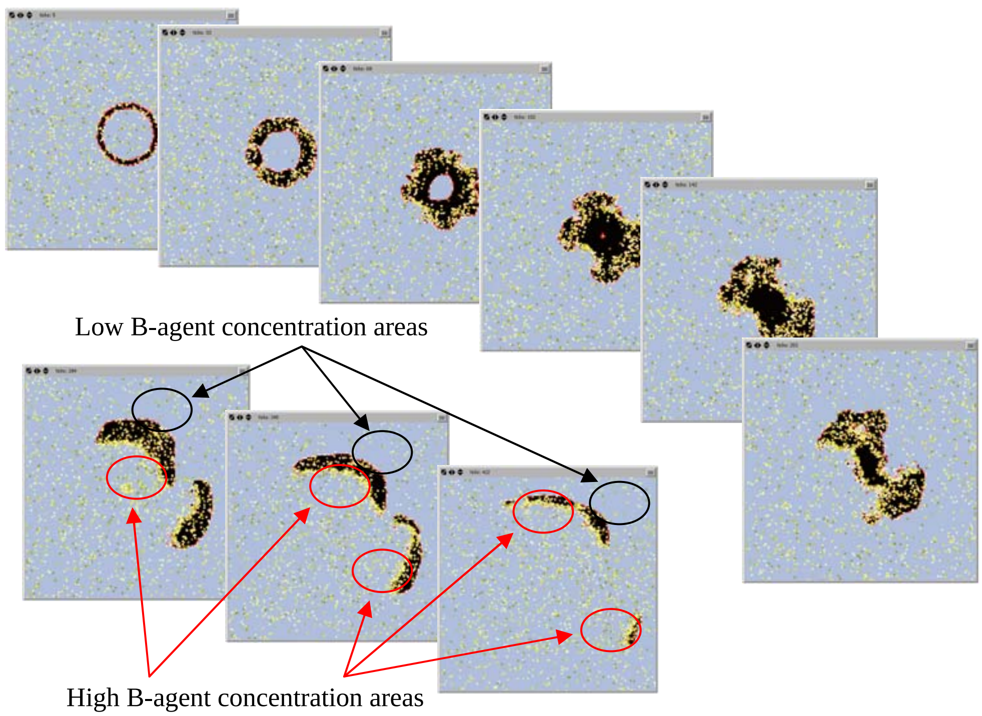

Figure 9. Illustration of nine stages of a sample simulation run. Low vs. high concentrations of B-agents (yellow) are indicated by the blue and red circles.

We compare this emergent behavior to that found in [Hawi06] and [Wile98]. Both of these models found in the literature show the phenomenon of "flocking" behavior among the agents. In particular, [Hawi06] uses a predator-prey model, and categorizes the resulting patterns of behavior between these two populations. These patterns are illustrated in Figure 10. The predator population in red are foxes and the prey population (blue) are rabbits. As they describe it: "Wave-fronts usually form after the collapse of a blob or spiral. The predators (foxes) spread out into a line to maximise [sic] their chance of finding prey (rabbits) and the rabbits become spread out in a corresponding line as they flee from the foxes. Thus two semi-parallel lines of animats [*agents*] merge into a cluster and travel in a wave across the area of interest."

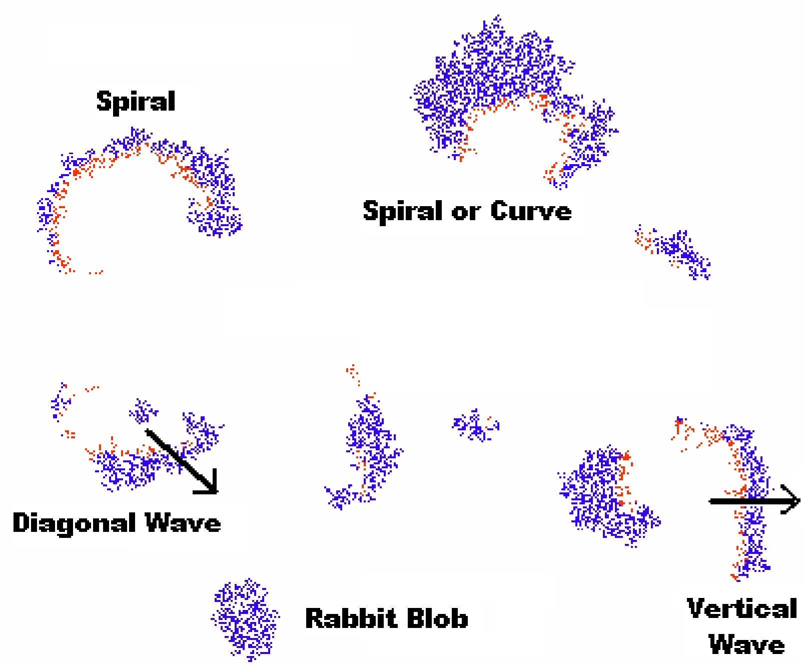

Figure 10. A catalogue of patterns from [Hawi06]. The blue agents represent the prey species, while the predator species is depicted by the red agents.

Note the similarity between the resultant crescent shape found in our model output (Figure 9), as compared to the "spiral" and "vertical wave" from this illustration. These emergent shapes are common to both models, even though the specifications are quite different.

In order to draw a comparison between the two models, we can consider the A-agents in the 1-state from our model to be the prey. Then we can imagine the B-agents "feeding" on these agents, as if 1-state A-agents represent food. Once a particular A-agent has been transformed into the 0-state, the "food" is gone and the mobile B-agent moves on in search of other food. This is where the models start to diverge: our model does not have mobile "prey" as in [Hawi06]. Yet both the crescent shape and the *movement* of the crescent shape is remarkably similar in both models. Figure 8 illustrates how the concentration of 1-state A-agents "moves" towards the low-concentration area of B-agents and away from the high-concentration area.

Of course, these agents aren't really moving – they remain frozen in the grid-pattern of the simulation environment. But their current state is passed from agent to agent, giving the illusion of movement that corresponds to the actual movement of prey in the rabbit-fox model.

Another key difference is in the programmed behavior in the agents. The rabbit-fox model specifies that these populations of agents prefer to be near each other. As the flocking birds in [Wile98], the foxes and rabbits are explicitly defined as "flocking" agents. In [Hawi06], they attribute the "blobs" of agents to these explicit instructions: "[T]he animat rules ensure that successful animats will always seek the company of others. […] This built-in 'seeking out of other animats' leads to the emergence of spatial clusters of animats."

While we cannot state for certain that their model would show flocking behavior even without the flocking rules, it is a significant result that such behavior does not always require explicit instructions. As we mentioned previously, feedback of some sort is the key to self-organized behavior. Our model shows that even indirect feedback, such that the B-agents are completely blind to other B-agents, is enough to allow these agents to form into persistent clusters.

Thus, the first iteration of our general model is able to simulate important features of a CAS: a simple feedback mechanism which allows for self-organizing behavior among the two agent types. It compares favorably to other general CAS models in the literature, replicating key properties even though the specifications of our model are simpler and require fewer top-down constraints. But this model is not a stable system. It's inherent complexity is only a temporary condition and not self-regulating in any way. We address this deficiency in the next section, primarily by adding two additional properties to the mobile B-agents.

4.4 Second Iteration

In order to stabilize the complexity of our model, our second iteration (and all subsequent models) includes two refinements to the specifications of the B-agents: that they have only a limited lifetime, and a method of reproduction. The lifetime is controlled by the operator and measured in terms of the number of simulation time-steps. Once the prescribed lifetime limit is reached, that agent is simply removed from the system. The method of reproduction is defined as a positive function of the number of A-agents affected; that is, the number of "successes" that a B-agent collects. A success is defined as moving a 1-state A-agent one step towards the 0-state. (Recall that the difference between the 0-state and the 1-state is sub-divided into steps; in this case, ten steps.) The number of successes needed to spawn a new B-agent is also controlled by the operator.

Other small modifications were also made to expand the range of possible experiments. We added a "susceptibility" attribute to the A-agents, such that the 1-state A-agents now have only a *probability* of moving an adjacent A-agent one step towards the 1-state. The susceptibility of each A-agent is set in the range [0, 1], inclusive, as a normal distribution (mean = 0.5, s.d. = 0.25). Also, in this model, both the efficiency of the A-agents and that of the B-agents can be adjusted. In the case of the A-agents, increased efficiency means there can be multiple opportunities to affect a neighboring A-agent (although still limited by the neighboring A-agent's susceptibility). For the B-agents, increased efficiency here means that they can each act multiple times during a single time-step. Finally, we also changed the model to allow for a small random chance of state-change in an A-agent each time-step, meaning there is a slight probability for each agent to move towards either the 0-state or the 1-state, even without the actions of any other agents.

With these changes, the model can now run continuously under most scenarios. The refinements of reproduction and a limited lifetime for the B-agents creates a situation where the size of 1-state A-agent populations and B-agent populations become self-regulating, mimicking a typical predator-prey dynamic [Yi08, Carn07]. The random movement towards either the 0- or 1-state for the A-agents is the catalyst. Once a single

A-agent reaches the 1-state, it becomes autocatalytic, producing more 1-state A-agents in a positive feedback dynamic. An increase in these agents, however, also increases the opportunity for B-agents. Therefore, the autocatalytic A-agents also produce more inhibitors, i.e., B-agents.

If the number of B-agents is growing faster than the number of 1-state A-agents, then the B-agents will eventually be able to overwhelm the A-agents and start to reduce their numbers. This reduces the autocatalytic effect, but it also reduces the production of inhibitors. Therefore, the growth rate of B-agents will slow down as well. Eventually, the system produces oscillating behavior (Figure 11) typical of the predator-prey model, which has similar dynamics. In Figure 12 we see that the relationship between the number of 1-state A-agents and B-agents is fairly stable over time. This stability represents the "basin of attraction" for this system.

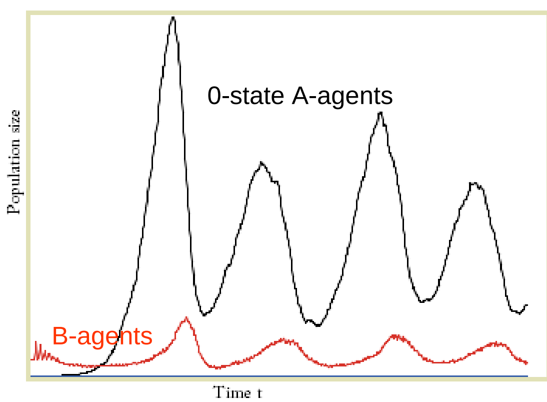

Figure 11. Typical predator-prey oscillations.

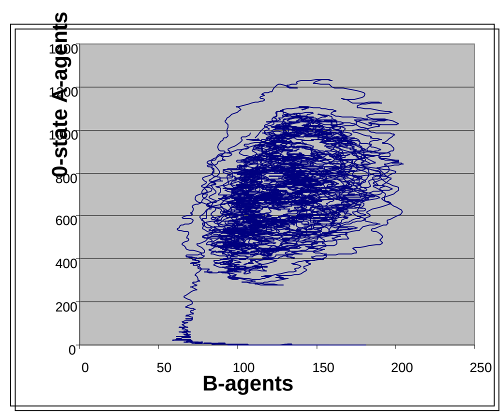

Figure 12. Cyclical "basin of attraction."

Note that even though the autocatalytic effect for the A-agents, and the functionally similar reproduction of the B-agents, only produces a stable system under certain parameters. In some scenarios, the fluctuations of either the 0-state A-agents or the B-agents can reach zero, at which point the system's self-regulating mechanism would collapse. The random change in state for the A-agents – if turned on by the operator – ensures a steady supply of dynamic A-agents. Likewise, a minimum number of B-agents can also be imposed on the system to prevent their population from reaching zero. The degree to which these two functions are in effect depend on the particular mapping of the simulation, and the known properties of the system being studied. Nevertheless, most scenarios in this model produce a long-term stable system, such that the collapse of either population is remote, allowing for extended observations of the general system dynamics.

4.5 Specifications of the General CAS Model

4.5.1 The Agents

Our general CAS tool utilizes two types of agents: A-agents, representing the environment; and B-agents, which act within that environment. Note that a single-agent simulation might seem to be the more general choice. However, in that case, a continuum of these agents would have to be defined. Therefore, a binary model comprising two general types of agents is more consistent with a minimum definition of a CAS. Furthermore, this paradigm allows us to consider multiple levels of feedback that would not exist in a single-agent simulation. In order to express feedback between the B-agents and the environment in a meaningful way, the A-agents have to have at least two distinguishable states, as well as a defined threshold to demark the change between these two states.

Therefore, the general CAS model that we have developed utilizes these two types of agents, A-agents and B-agents, with the following rules:

A-agents: 1) A-agents have two polar states, labeled 0-state and 1-state. 2) The progression between 0 and 1 goes in steps. 3) Once an A-agent reaches the 1-state, it can affect the state of other A-agents within its neighborhood towards the 1-state. 4) There is a chance of random movement toward either the 0-state or the 1-state. 5) A-agents have velocity = 0, and lifetime = infinity.

B-agents: 1) B-agents are mobile and affect the adjacent A-agent towards the 0-state. 2) If the adjacent A-agent is already in the 0-state, it will move randomly. 3) They spawn new B-agents as a positive function of the number of A-agents affected. 4) They have a limited lifetime.

In this way the dynamic properties of this system are a result of the opposing forces between the A- and B- agents: A-agents can move each other towards the 1-state while B-agents can move them towards the 0-state.

There are many attributes of both A- and B-agents that can be adjusted in the model to produce various effects. Examples include the neighborhood for each A-agent; the degree of random movement towards either the 0-state or the 1-state; the distance between the 0-state and 1-state; and the efficiency of the B-agents, in terms of speed (number of turns per simulation time-step), distance traveled per turn, lifetime, and spawn-rate for new B-agents. A minimum number of B-agents can also be set if desired, and their ‘vision’ can be adjusted. Currently, B-agents can only detect A-agents that are directly adjacent: i.e., within the same simulation grid point. They cannot detect other B-agents.

Due to the large number of adjustable features, the state space of this computational model is extremely large. Here we only partially explore this state space; however, it should be noted that there is ample room for a great deal of flexibility in the model, and a rich environment exists for future experimentation and applicability. Nevertheless, the limited environment addressed here displays all the classic properties of a CAS, including self-similar agents, feedback, emergence, self-organization, and non-linear dynamics.

4.5.2 Feedback in the CAS Model

We previously defined feedback as a circular system of causality, whereby some portion of the output of the system is returned as input in subsequent simulation time-steps. In this model there are only a few feedback mechanisms, in keeping with the desire to use the simplest model possible. The immobile A-agents can affect, or be affected by, other A-agents. As described in the specifications, this only occurs once the A-agent has reached the 1-state, and they only affect other A-agents in the direction of the 1-state. The potential feedback is represented by the fact that once a neighboring A-agent has been successfully transformed to the 1-state, that neighbor is now capable of affecting the original A-agent in turn. (If, that is, the originating agent is ever somehow moved away from the 1-state.) In this way, current action from one A-agent to another is likely to be returned in a future time-step.

The mobile B-agents also exhibit feedback mechanisms, both amongst themselves and between the two types of agents. Once a B-agent successfully transforms an adjacent A-agent back to the 0-state, its local environment is now changed; thus, the B-agent is free to move randomly away. Furthermore, the spawning of new agents is a positive function of how many A-agents are affected; so the population growth of the B-agents is controlled by the current state of the A-agents.

The feedback *between* B-agents is much more indirect, however, since in the version of the model as described in the following experiments, they cannot detect each other in any way. The indirect feedback, then, is achieved via the status of the adjacent A-agents. If a common A-agent is transformed by one B-agent, then all the other B-agents also adjacent to it have their mobility returned. can affect A-agents towards the 0-state, and spawn new B-agents as a positive function of this behavior. As we will see later, this indirect feedback mechanism is necessary for the self-organizing behavior amongst all the B-agents.

4.5.3 Flexibility of the Model

As stated previously, one of the most important aspects of this model is its inherent flexibility. This is a key property that allows for mapping the model to a particular domain, so that the generated outputs can be adjusted to match the known system outputs of that domain.

The CAS model we describe here has such flexibility in a number of ways. For example, the B-agents can be made more efficient in four distinct ways: 1) their lifetime can be increased; 2) the number of affected A-agents needed to spawn a new B-agent can be reduced; 3) the distance traveled each turn can be adjusted; and 4) the number of turns the B-agents have for every simulation time-step can be attuned. Each of these adjustments makes the B-agents more efficient in different ways, so that the time-series outputs of the model have different characteristics.

CHAPTER 5: EVALUATION

5.1 Mapping the CAS model

The mapping of the general CAS tool is an iterative process. First, a conceptual model in the problem domain is created that can be represented by the specifications of the general model. Then the model is adjusted so that time-series output data matches what we would expect to be true, or what we can show to be true, in this domain. Using the remaining flexibility, we can then fine-tune the model to preserve correct outputs. In this way, the model's ability to both explain and predict will be enhanced. As stated in [Simo99]:

> "By altering the model in various respects, we can investigate the effects of experimental manipulations, and thereby derive empirical predictions for empirical studies. A computational model can thus give rise to new theoretically motivated experiments, and to reexamination of existing experimental data – further examples of the complementary relation between cognitive modeling and empirical studies."

Domain experts, whose deep knowledge of the field helps to establish and preserve the ground-truth for the inputs and outputs of each model, lead this iterative process.

In the following sections we map our CAS model to the growth of aggressive soft-tissue tumors and to a model of political dissent in a polity.

5.2 The Development of Aggressive Tumors

The growth of a tumor and the immune response to that growth can easily be modeled within the framework of out CAS model. A-agents represent tissue cells, and B-agents are immune cells; the tissue cells are "healthy" or "cancerous," depending on their current state (0-state = healthy, 1-state = cancerous). Once a cell becomes cancerous, it begins to proliferate more cancer cells in its neighborhood. The immune cells can attack cancer cells; as they do, they attract more immune cells to the cancer (by spawning new B-agents).

The aggressive tumors threshold exhibits characteristics of the ratchet effect. Initially, cancer cells have few negative health consequences in part because the immune cells and reduced angiogenesis limit their growth and activities. However, once the threshold is reached, the growth of the aggressive tumor is no longer limited through communication with surrounding cells or the actions of the immune cells. The CAS model is used to: 1) define the parameters associated with the aggressive tumor growth threshold; 2) suggest and generate models suitable for individual tumor modeling; and 3) better understand relationships between the different agents in tumor development that suggest new targets for diagnosis and treatment.

Computer-based models are not yet reliable enough to substitute for randomized clinical trials in decision making [Beer07]. However, the CAS paradigm moves us towards more realistic models, allowing for a more complete understanding of a biological system because it can take into account multiple features that interact in complex ways, including tumor intrinsic features, the net tumor cell growth, and the influence of both the immune system and the vascularization [Gate07]. Modeling of toxin effects and anti-tumor efficacies in vivo provides opportunities to tailor combination therapies to the aggressiveness of malignant tumors [Axel06, Okte06, Gate04]. However, these models fail to account for architectural complexity of the tumor and angiogenesis. Fractal geometry and mathematical models [Gate04, Kozu07] have had better success, but they provide only partial representations of the events associated with solid tumor growth and development [Geri07]. Consequently, they are of limited use in determining tumor aggressiveness thresholds.

For the aggressive tumor threshold, the net growth of the tumor mass is modulated by: 1) intrinsic tumor events; 2) interactions with extra-cellular matrix; 3) nutrients; and 4) interactions with other cell types [Gate07]. We have completed a preliminary study of this model, one that captures much of the complexity involved in the formation of an aggressive tumor. The tumor cells vary in their ability to grow, ability to escape the immune system. The immune cells vary in their ability to detect cancer cell space and to destroy cancer cells. Furthermore, the system can be adjusted to provide a minimum number of immune cell agents, replicating either an inherently strong or weak immune system.

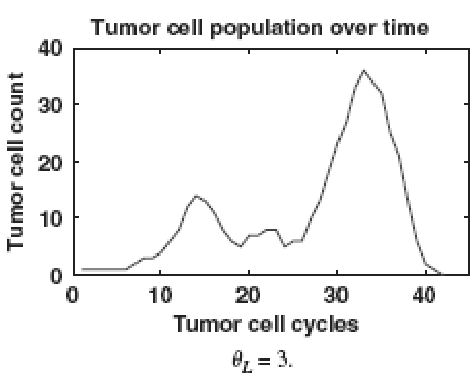

Figure 13. Mallet: Tumor growth failure.

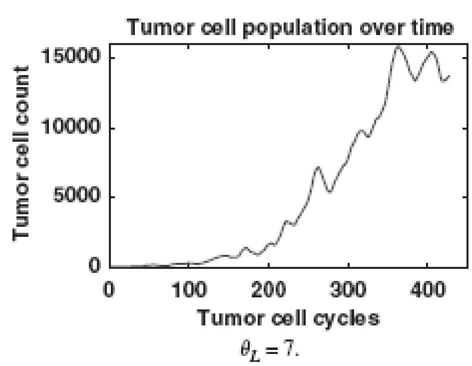

Figure 14. Mallet: Tumor growth success.

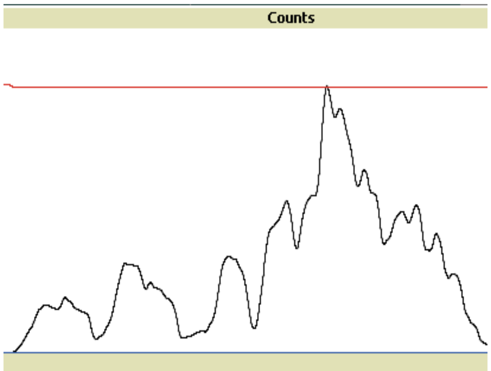

Figure 15. CAS model: tumor growth failure. As with Figure 13 and Figure 14, tumor cells are in black; immune cells are in red.

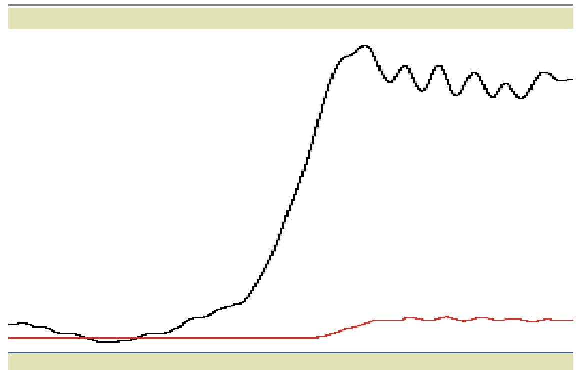

Figure 16. CAS model: tumor growth success. Tumor cells are in black. The level of immune cells (red) begin the same as in Figure 15; thus the scale of the black line indicates dramatic tumor growth.

For validation of this model, we compared the outputs to a previously published theoretic data set [Mall06]. Figure 13 shows the growth and reduction over time in the number of tumor cells. The immune response suppresses this growth, preventing a threshold in tumor size from being reached. Figure 14 shows the same model, but with adjusted parameters that allows the tumor to reach and exceed the threshold level of growth. In the top-right corner of this graph, a new stable level in the number of tumor cells is shown.

Our CAS model outputs, as shown in Figure 15 and Figure 16, can easily be adjusted to match the characteristics of this time-series data. Figure 15 matches the cyclical growth and repression of the number of tumor cells found in Figure 13, and Figure 16 mimics the dramatic growth – after crossing a threshold – to a new steady state, as in Figure 14.

Note that our model does not directly induce a strong tumor growth as in [Mall06]; rather, the development of an aggressive tumor is allowed to stochastically appear based on the small chance of random movement for the A-agents, towards either the 0-state or the 1-state. Due to this, the simulation can run quite a long time before the threshold change between the two steady states: that of a relatively low number of cancer cells and a high number of cancer cells. In one experiment, over the course of 500 simulation runs, the smallest number of time steps to produce the tipping point was 1456 simulation time steps; the largest number was 98,380. Approximately 79% of the time, however, the critical threshold was reached in less than 20,000 time steps, and the relationship of number of time steps to reach this threshold is such that it becomes increasingly unlikely to have simulation runs with an extended number of time steps.

The settings used to produce these outputs were as follows: immune-cells have 10 turns per time step; they can move 0.12 the distance of one grid cell each time they move; they can attract a new immune cell after moving a cancerous cell towards the healthy state 15 times; and their lifetime is 65 turns. Furthermore, a minimum number of immune cells were added, to mimic the body's natural state. This number was set to 270 immune agents.

In Figure 16 as we've seen, the number of cancer cells reaches a new steady state at a dramatically higher number of cancer cells than found before the threshold was reached. (In terms of studying cancer, this is the end of the useful duration of the model, as this level of sickness represents the death of the host.) Here, the number oscillates back and forth a bit, but doesn't have much volatility. We let this simulation run for an extended period of time (approximately 700,000 time steps) without seeing a reversal back to the lower stable number of cancer cells. However, by making small changes to a few of the parameters, the model will show random movement between the low number of cancer cells and the high number of cancer cells.

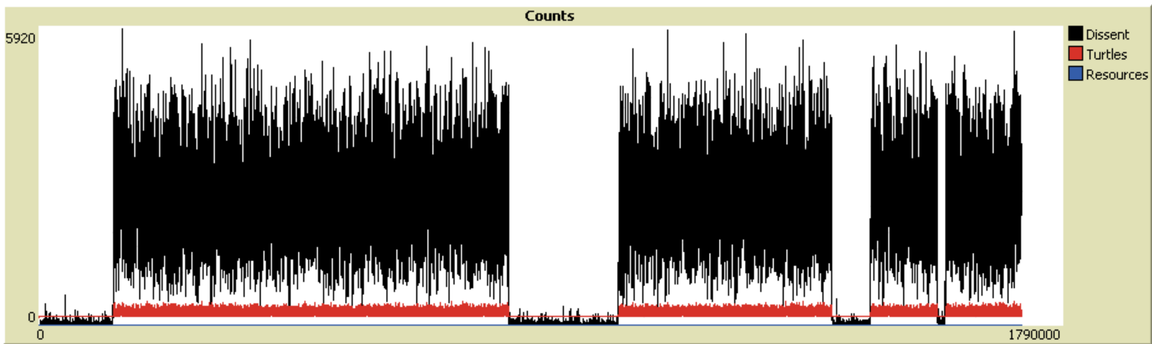

Figure 17. Number of cancer cells over 1,790,000 time steps.

The number of turns per time step for the immune cells, as well as the attraction rate, remained the same. The distance an immune cell can travel was increased to 0.18, and lifetime was increased to 75 turns, and the minimum number of existent immune cells was reduced to 170. In Figure 17, the low steady state and the high steady state can clearly be seen. The thickness of the black line indicates the high degree of variability in the number of cancer cells. Note also that the number of immune cells – here in red – is also elevated, in response to this. Occasionally – three times, in this graph – the immune response is successful in suppressing the cancer cells, pushing it back to a low steady state. Repeated experiments have shown that the movement between the low and high steady states is stochastic and – within a certain range – it is unpredictable as to whether the agent populations will remain in their current state or transition to the other.

5.3 Political Dissent in a Polity

We can also use this CAS model to trace the trajectory of political dissent within a population. In this mapping, A-agents represent ordinary citizens, who take on a dissent state ranging from the 0-state, indicating no dissent, to the 1-state, indicating dissent. B-agents represent government agents, who suppress dissent. Dissent increases in some ordinary citizens and spreads to others.

This model introduces the idea of resources into the CAS framework. The number of government agents can change in response to the change in dissent, but is constrained by the total resources available to the government, which in turn is negatively related to the total amount of dissent (i.e., as more people dissent, they also withhold their share of resources from the government). When almost all ordinary citizens comply, the government economizes by putting few government agents on the ground, but – as with the immune cells in the cancer model – there is a minimum number of government agents.

Dissent by the citizens increases and has a contagion effect on the dissent level of those near by. That is, a citizen is a more likely to dissent if nearby citizens are already dissenting. This is similar to the cancer spreading to nearby cells. However, unlike the cancer model, the dissent spreads not to adjacent cells, but a random number of nearby cells. These nearby cells represent the A-agent's "neighborhood," that is, the people that this agent would normally come into contact with or be able to influence.

In response to rising dissent, the government increases the supply of government agents. The more government agents that are deployed, the more dissent is suppressed. If the level of dissent in the population becomes sufficiently high, however, the government lacks the resources to deploy sufficient agents. The model allows us to analyze multiple potential thresholds, including (a) a start-up threshold of dissent, (b) a turning point threshold of the relative numbers of dissenters and government agents, (c) a government success threshold of dissent, and (d) a dissent success threshold. These processes are consistent with current empirical and theoretical work on dissident social movements [Oliv03, Rosc01]. The key parameters of this preliminary model of dissent include: heterogeneity in the susceptibility of ordinary citizens to social influence; speed at which government agents are generated in response to dissent; and the resource constraints on the government.

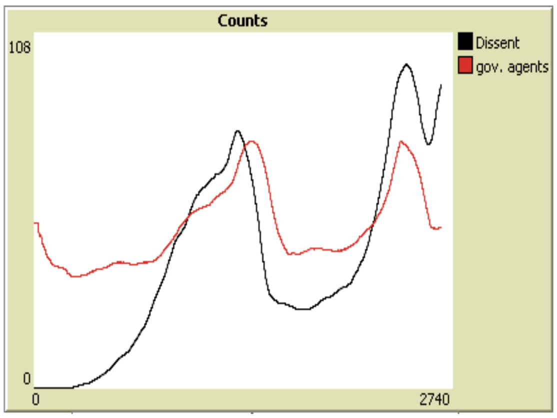

Figure 18. Government agents and dissenters

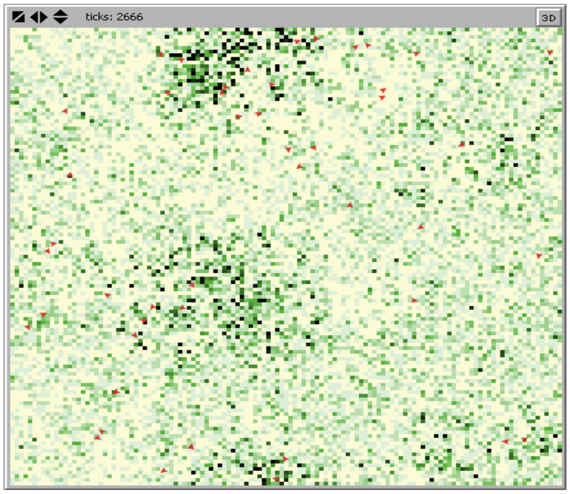

Figure 19. Graphic representation of dissent.

Figure 18 shows the response in the number of government agents to the changes in dissent, and Figure 19 shows graphically the spreading dissent in the population: the levels between the 0-state (no dissent) and the 1-state are shown in green, with those agents close to the 1-state being darker. The government agents are shown in red.

We compare this data to Figure 20, which shows protests and detentions in South Africa, from 1970 to 1986 [Olza05]. We can clearly see that there is some correlation between the change in number of protests and the change in number of detentions. Of course, we do not expect the graph to exactly match any real-world data set point-for-point. Rather, we try to find settings in the simulation that produce realistic outputs in terms of how we can characterize the graphs.

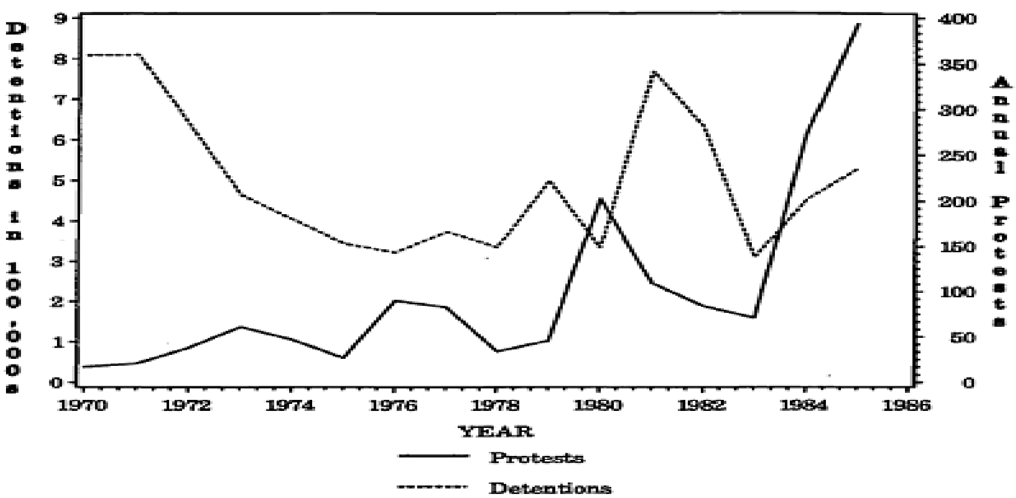

Figure 20. Detentions and protest in South Africa, 1970-1986 [Olza05].

In our model of dissent, once the government resources run out, the number of dissenters increases dramatically, eventually leading to total breakdown of control in the population. Figure 21 shows the number of dissenters relative to the government agents after resources reach zero, and Figure 22 shows graphically the contagion model of dissent spreading through the population.

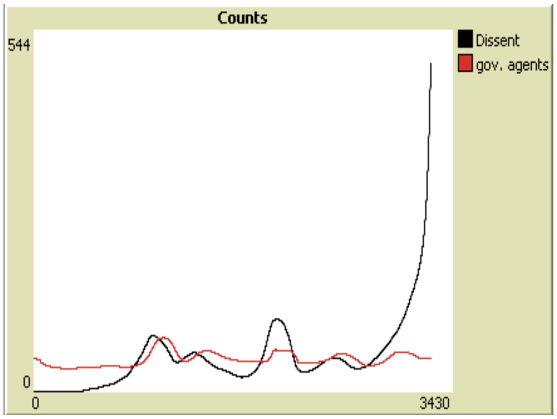

Figure 21. Growth of dissent, no resources.

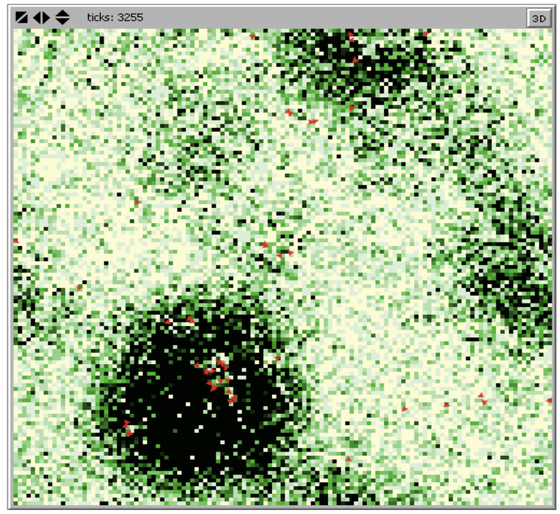

Figure 22. Graphic, contagion of dissent.

The contagion effect illustrated in Figure 22 reflects the difference in this model compared to previous ones, in terms of the network of agent interactions. Since each agent's network is a random sample from within a constrained radius, the spread of dissent in this model is more diffuse than the spread of cancer in the previous model.

Nevertheless, these two models share another important characteristic, relative to other models of contagion dynamics. In a archetypical contagion model, a single contact from a single source is sufficient to spread the influence, whether it be the transmission of a disease or the passing along of a piece of information [Cent07]. This is true regardless of the probability of transmission; even a low-probability transmission is either passed along or it isn't. In contrast, for both the cancer and the dissent models, transmission occurs in an additive fashion, such that the agent is affected by each contact, but the transformation threshold (from no dissent to dissent, or from healthy to cancerous) requires repeated contacts to occur. These contacts may originate from a single source or multiple sources in the agent's network. Thus both of these models display the same type of contagion as described in [Cent07], opening up intriguing possibilities for the transfer of domain models of social contagion to the spread of cancer, and vice-versa.

Even the second main refinement – the concept of resources – has an obvious analogy in the case of an immune system: overall resources of the immune system could be a measure of how healthy the host is, and an increase in the immune system's response (exhibited by, for example, an increase in the number of immune cells) can tax that system. Further, as the cancer grows and adversely affects more of the host body, the body's ability to replenish its resources may be degraded.

5.4 Summary of the General CAS Model

To summarize the main features of the general model, Figure 23 illustrates the order of operation in the form of a flow chart: from initialization, to the A-agent ("patch") and B-agent ("turtle") cycles, to the user-defined stopping condition.

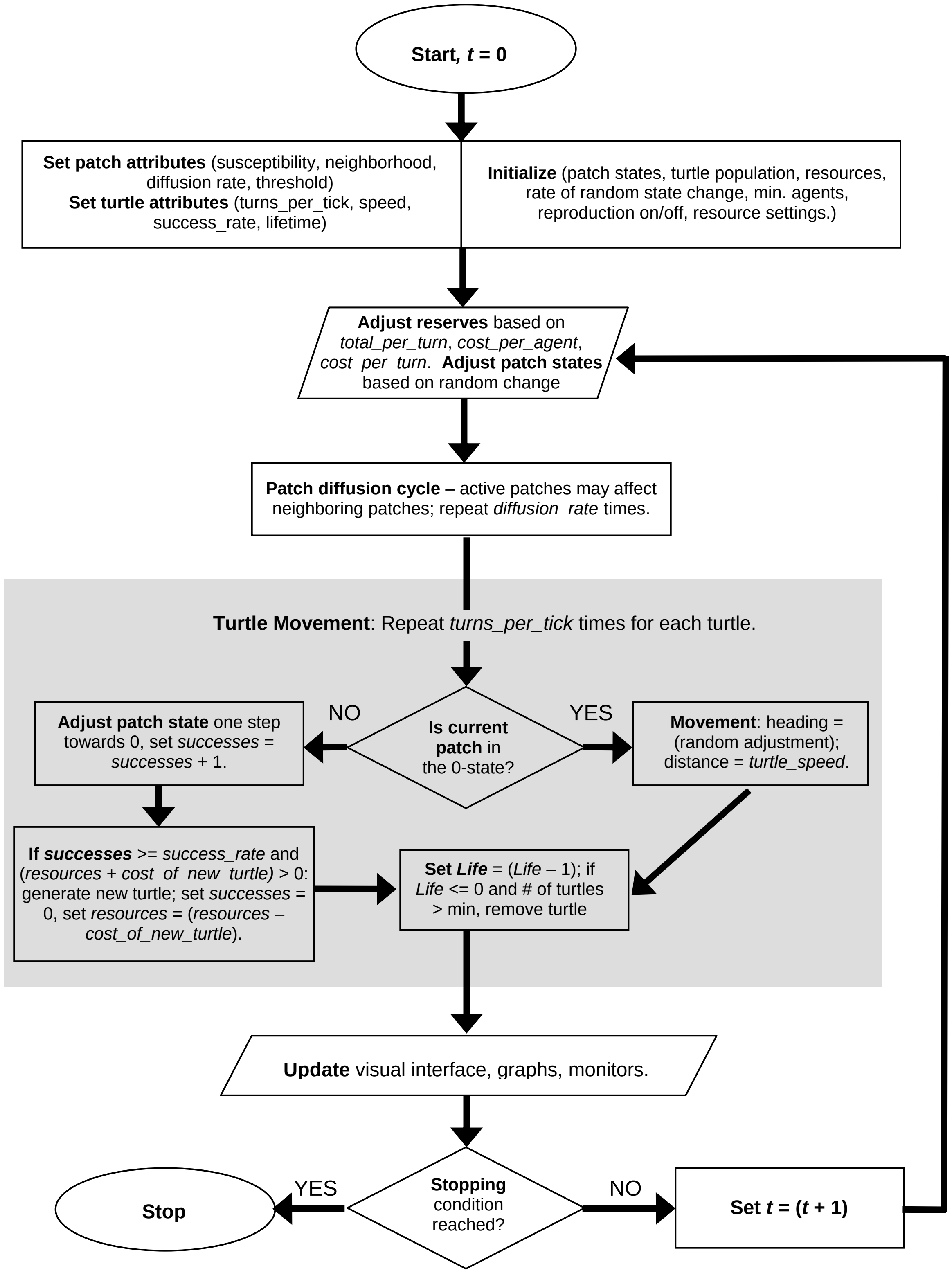

Figure 23. Flow Diagram illustrating the general model's order of operation.

The order of operation is the following: 1) Start the simulation at t = 0; 2) Set A-agent attributes: susceptibility, neighborhood, diffusion rate, and threshold(s) for change between 'active' and 'inactive'; 3) Set B-agent attributes: turns, speed, *success_rate*, and lifetime; 4) Initialize: A-agent states, B-agent population, resources, rates for random A-agent state change, minimum number of B-agents, reproduction (on/off), and resource settings (if used); 5) Adjust reserves based on *total_per_turn*, *cost_per_agent*, and *cost_per_turn*; 6) Adjust A-agent states based on random change setting (probability); Run the A-agent "diffusion cycle" (this is run *diffusion rate* number of times); 7) Run the B-agent movement cycle (this is run *turns_per_tick* number of times – each turn, a B-agent will either affect the adjacent A-agent or move randomly); 8) Update the interface monitors and graphs; 9) Check for the stopping condition, repeat steps 5 through 9 as necessary. (For a more detailed description of all the attributes, functions, and agent-types, please see Appendix A.)

The stopping condition varies according to the operator's preference. In the case of an unstable scenario, the stopping condition is when either the A-agents are all in the 0-state (unless there is the possibility of a random change) or the B-agent are all removed from the system (which can be avoided with a minimum requirement of B-agents). Many scenarios result in stable system-level behavior, even when the random state change or minimum B-agent levels are not used. In this case, an alternative stopping condition may be defined. It may be after a cyclical, oscillating phenomena has been observed, such as in Figure 11, or it could be when the system changes dramatically, as in the stochastic phase shifts observed in Figure 17.

5.5 Preliminary Results: Marine Ecosystems

The general model and its application to both soft-tissue cancer and political dissent exhibit many of the characteristics of the classic predator-prey model. Figure 11 illustrates the typical oscillations between the two populations found in the predator-prey dynamic. One of the foundations of ecology dynamics is the Lotka-Volterra equations for predator-prey populations. These equations are both mathematically robust and widely accepted, but are also general in nature. Thus, they are limited by the assumptions imposed upon them, including, for example, the assumption of unlimited resources available to the prey population.

First proposed in 1925-1926, the Lotka-Volterra equations are a pair of first-order, non-linear differential equations that govern the relationship between two types of interacting species. The equations have periodic solutions, such that an increase in the prey population generates a temporary increase in the predator population, which increases predation levels. Increased predation reverses the growth of the prey population, which in turn reduces the predator population. Once the prey reverses again to a growth phase, the cycle is complete.

These dynamics are well-understood and have been validated in both computer simulations and real-world studies. By utilizing the CAS framework and the general CAS tool for simulating this relationship, we can easily incorporate more realistic, stochastic elements than one would find in a purely mathematical solution to these equations. Nevertheless, this model easily captures the cyclical nature of this well-understood dynamic.

The key assumptions of the Lotka-Volterra equations are: 1) unlimited food availability to the prey population; 2) the predator population depends entirely on the prey for food; 3) the natural growth rate for both populations are proportional to their sizes; and 4) the environment doesn't change to the benefit of either population.

Our investigation into a deeper understanding of the predator-prey dynamics began by changing assumption (1) above: we tailored the general simulation model so that the food available to the prey population is adjustable. The simulation environment is a torus grid with 151 * 151 grid cells, for a total number of 22,801 cells.

We also define four populations – four agent-types – in this model rather than two: food (generated by the simulation stochastically as a constant rate per grid cell); fish (the prey population); eggs (generated by the fish as a positive function of the amount of food consumed); and predators (which reproduce as a positive function of the number of fish consumed). In this mapping, "resources" are not needed as an additional system-level property. Unlike the mapping to cancer or political dissent, the A-agents represent food; and the actions that B-agents (the fish) perform on the A-agents represents eating the food. Thus A-agents provide resources to the fish even as they are affected by them. Further, the fish also provide resources to the predators.

Of course, this model is not intended to be exhaustively realistic, but rather simply capture the basic properties of the predator-prey-food relationship. As such, the environment is currently homogeneous, without any variations in sea temperature, depth, or ocean currents. Also, each tropic level is represented by a single species, without the complex dynamics of functionally similar, individual species. These refinements can be selectively added to future models in an iterative process, to ensure that the basic dynamics at each level are well-understood before proceeding to the next level of complexity.

As with the environment, both the fish and the predator populations are also homogeneous, different only in their current state variables: individual age, x-y coordinates, and current amount of food consumed. When the simulation is run with a baseline test-case (food production set to 20% chance of positive growth per cell, per simulation time step) it settles to an equilibrium relationship between the fish and predator populations. The fish population is somewhat more variable than the predators, stabilizing generally between ~1100 and ~1200 individuals. The predator population stabilizes at ~170 individuals.

This model was designed to de-emphasize the cyclical volatility of that found in Figure 11, in order to more clearly see the overall population trends of each species. In terms of age, the equilibrium age for predators is about 50% higher than that of fish. These outcomes can be adjusted by changing the parameters and the environment to more realistically capture real-world species. However in the current simulation, what is important to note is how the food supply – the lowest trophic level – affects the relationship between the mid- and high-trophic level populations. Figure 24 shows the change in population counts as the food supply is increased from 20% to 30%, and again to 40%.

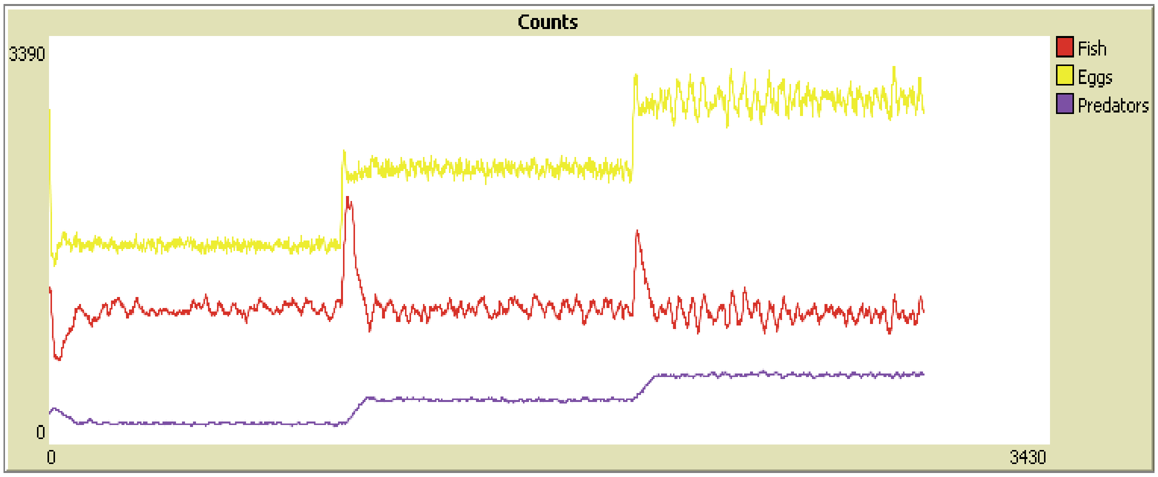

Figure 24. 3000 simulation time-steps showing population counts at 20%, 30% and 40% food levels (1000 steps per level).

Initially the fish population increases in response to an increased food supply, as more resources allow them to produce more eggs. The predator population subsequently also increases as their food supply (the fish) becomes more abundant. Remarkably however, the gains in fish population are only temporary, and are quickly offset by the increased predation rates. Thus, the fish population returns to the same equilibrium level as that found with a lower supply of food: only the predator population remains elevated. This result indicates that all the gains resulting from the increased food supply are transferred to the high-trophic-level predators.

When we examine the changes in the average age for each population, we see that the fish – though reproducing at a faster rate – don't live as long as they do at a lower food supply. Even as they reproduce faster, they are also consumed faster, so that their average age is much lower, thus preserving the equilibrium population size. The predators, faced with an increased food supply, are much more efficient in catching the fish; thus, their population increases. Unlike the fish however, their average age – after stabilizing at a higher population – is essentially unchanged. Figure 25 shows changes in the mean population age for each of the three populations, at the three different food supply levels as found in Figure 24.

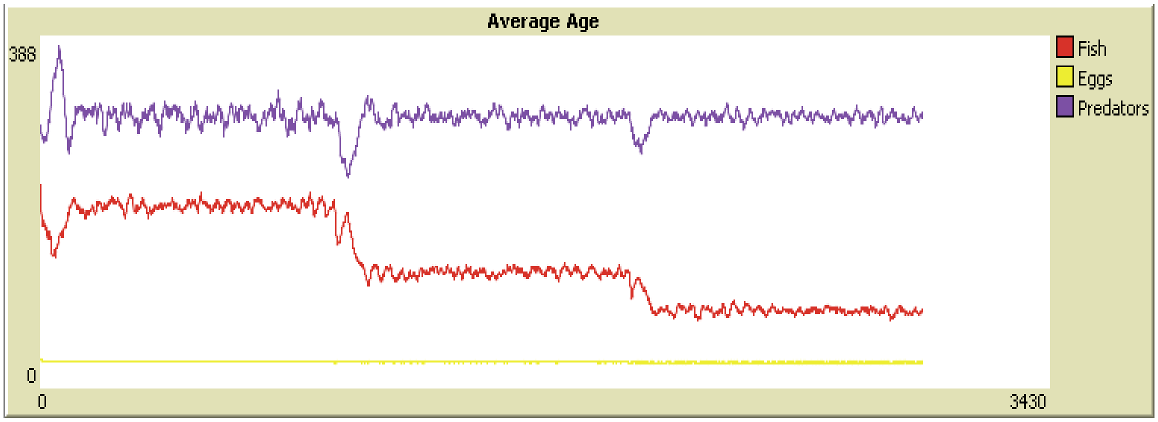

Figure 25. 3000 simulation time-steps showing mean population age at 20%, 30% and 40% food levels (1000 steps per level).

These same results have been replicated in experiments that exclude the population of fish eggs, substituting instead a simple spawn rate based on food consumption that exactly mimics the method used for predator reproduction. Thus, incorporating the fish egg agents into the model only adds a slight delay in generating new fish, one that can easily be accounted for by adjusting the "reproduction rate" attribute of the fish agents. With the fish eggs added the model is more complicated, and more realistic, adding verisimilitude as well as an added degree of stochasticity; however, functionally, it has proven to produce nearly identical outputs.

These results have also proven to be robust across other modest refinements, such as allowing the predators to predate upon fish eggs as well as the fish themselves. Of course the exact outputs vary across these experiments, in terms of exact population levels and range of setting that produce equilibrium; however, the general finding remains the same.

Though we suspect that other factors will ultimately influence these dynamics to some degree, the overall picture is clear: the highest trophic level – the predators – are greatly affected by the food available at the lowest trophic level, while the species in the mid-trophic level – the fish – are affected only in average age, not in population. This is a surprising and new result not found in the literature for a three-trophic-level simulation. As such, we must ensure that these results are robust, and that they comport with experimental and real-world observations.

CHAPTER 6: SUMMARY OF RESULTS AND FUTURE WORK

In designing our general CAS model, we strived for two things: simplicity and generality. At each stage of development we evaluated our tool in terms of its ability to replicate important features of CAS in general as well as simulating known effects in specific domains.

The first, simplest iteration of our model clearly showed self-organizing behavior, as the mobile B-agents tend to cluster together, even though they are not able to see or detect each other directly. This is contrasted with other flocking or clustering models, in particular the work of Hawick, et al. [Hawi06]. In their work the clusters of agents showed a remarkable similarity to those found in our model, even though: 1) both of their designated agent-types were mobile; and 2) the agents were explicitly programmed to seek out like agents. This doesn't mean, of course, that such behavior is not found in nature; only that it is not a necessary condition for such behavior to exist.

This new result could have interesting implications not only for ecological modeling and simulations, but also for application in a general sense to any model that currently assumes agents have to have some sort of direct link to each other in order to influence each other. For example, in a cancer/immune system model, it may be assumed that clusters of immune agents are interacting with each other; this example indicates that that is not necessarily the case, which could help us understand the physical processes involved in cellular activity. In social science, the idea of discernable indirect influence has been shown in the case of happiness, obesity, and smoking, across two and even three degrees of separation [Fowl09].

Our second model iteration added, primarily, a limited lifetime for the mobile B-agents, as well as a method of reproduction. These two refinements allow the model to run continuously, rather than always degenerate into one of two end states. In this way the model is now self-regulating. This model was successful in replicating the typical predator-prey dynamics that are found in many domains. This is true even though the agents and their interactions are stochastically variable across the simulation environment, rather than defined with a mathematically idealized approach.

The general model, therefore, consists of self-similar agents that exhibit all the properties of a CAS as we have defined it: the system is self-regulating; the agents produce emergent properties and self-organizing behavior; and the system shows non-linear dynamics, producing threshold effects that are both user-defined (such as the transition between the 0-state and the 1-state) and emergent (such as the system-level oscillations between two states). Furthermore, these oscillations may occur at different scales, and can be either cyclical in nature (Figure 11) or occur stochastically, as seen in Figure 17. Finally, the model provides a great deal of flexibility, due to the focus on simple primitives, such that a wide variety of behaviors can be realized simply by making small adjustments to the agent attributes, or to the rules governing their interactions.

We first applied this general model to a generalized representation of soft-tissue cancer and the immune system response. The outputs of the model successfully mimic known properties of this system, and were shown to compare favorably to previously published theoretic data, in particular that found in [Mall06]. In addition, our model was able to produce the growth of cancer dynamically, rather than requiring its explicit introduction into the simulation. This aspect allows for future experimentation of cancer growth at larger time scales, such as over the lifetime of the patient rather than simply the lifetime of the cancer.

The general model was then mapped to political dissent in a polity, which represents a domain not usually compared so explicitly to that of cancer growth and suppression. This model also introduced the concept of resource availability for the B-agents, and allows for greater variation in the defined networks of A-agents. Thus, the model allows us to analyze multiple potential thresholds. The outputs of the model are consistent with current empirical and theoretical work on dissident social movements [Oliv03, Rosc01, Olza05].

Preliminary work has been performed in applying the general model to a marine ecosystem. Additional agent-types were added to allow experimentation with three tropic levels rather than just two. We also allowed for a more complex method of agent reproduction, with the introduction of 'fish-egg' agents rather than simple, asexual reproduction, which adds additional verisimilitude to the simulation. This additional feature may introduce important dynamics under certain conditions, but has so far proven to be functionally similar to that found in the simpler model. This result underlines the utility of our approach in emphasizing, as Einstein said, that everything should be as simple as possible, but not simpler.

This ecological simulation is considered preliminary until sufficient consultation with domain experts and a more thorough literature review is undertaken; however, the model exhibits surprising effects in terms of the distribution of benefits that result from a higher resource availability, and is a key area for future research.

Table 1 summarizes the key attributes for each mapping of the general CAS model.

Table 1. Key parameters and attributes for each model.

	1st iteration	2nd iteration	cancer	dissent	marine
number of agent-types	2	2	2	2	3-4
self-regulating?	no	yes	yes	yes	yes
A-agent neighbors	adjacent only	adjustable	adjacent	nearby, random	n/a
B-agent reproduction basis	none	based on positive function of successes	based on positive function of successes	based on positive function of successes/ limited by resources	based on consumption levels
B-agent lifetime	infinite	limited	limited	limited, possibly resource related	limited or predation
min. # of B-agents?	initialized	adjustable	yes, adjustable	yes, adjustable	no

6.1 Intellectual Merit

Over the last twenty-five years, CAS and ABM have grown as increasing numbers of researchers have used these tools to explore complex phenomena found in nearly every field imaginable. On one side we have models that are extremely detailed and specific, capturing every possible attribute or agent interaction that is salient to the general system dynamics being replicated. These may be anything from simulations of vast markets, to ecological systems, to social contagions, to even biological and chemical interactions at the microscopic level.

On the other side we have models that illuminate very specific and simple behavior that is known to occur in many different fields. Schelling's segregation model, which we discussed earlier, is one particular effect that may manifest in many different systems. Another is the flocking behavior of birds, or fish, or even microbes.

As Johnson, Epstein, and others have pointed out, we have not previously seen a model that is specific enough to capture the dynamic properties of a complex system, yet still general enough to be applied to multiple domains. Thus, what we have accomplished here is to increase the range of complexity available to very simple, general models, while simultaneously increasing the generality of very domain-specific models, in order to meet somewhere in the middle. It is not, as we have stated, intended to be a final solution; rather, it is an important first step.

Our general CAS model has successfully reproduced a particular class of phenomena, defined herein as threshold phenomena, found in disparate domains. Furthermore, the few key attributes as listed in Table 1 do not represent all the variables, functions, and interactions that our model represents and that the three domain mappings – cancer, political dissent, and marine ecosystems – all have in common. Thus, there are certainly many more potential systems that can be simulated with this general tool, as the extent of its range and flexibility has not yet been fully explored.

There may be instances of complex systems that are not easily mapped to our general model, or instances where such mapping is not the ideal framework for studying a system. Nevertheless, this model and the process for developing it can serve as a template for how to approach other classes of problems from different fields that also share key characteristics.

6.2 Significance

The ultimate goal for this research is to build more and better bridges across the disciplines. In discussing reports on interdisciplinary creativity, van Raan states that "Eminent scientists strongly emphasize the crucial role of instruments for the progress of science, particularly the 'bridging' role between disciplines, by transferring instruments from one discipline to another" [VanR00].

The general tool described here is not simply intended to allow researchers from multiple fields to work on a single issue (although it may certainly be useful to that end). Rather, as we illustrated with the multiple systems that can be simulated, it is a way to bridge the gap between different complex systems themselves. There are many particular problems that require a multi-disciplinary approach to solve, but these are often done by splitting a multi-faceted problem into parts: the biologists tackle the biology parts, the economists study the economic parts, and so on, bringing the individual contributions back together for the final solution.

Our model instead acts as a common language: a detailed method for describing and discussing threshold phenomena. By emphasizing a common language rather than a common problem, our model allows research in one area to inform research in other, distant areas; further, it can lend insight into domain-specific issues simply by providing a methodology for reframing a research question in terms of our general model specifications.

For example at UNC Charlotte, in the CAS Research Group, Marvin Croy has been working on a mapping of the general model as it applied to deductive proof construction. Dr. Croy, a professor of philosophy, is interested in how students learn the rules of proof construction based on particular instances of solutions that are provided to them. Biology professor Didier Dréau, who contributed greatly to the cancer model [Dréa09, Carm09] and is also a member of our research group, has stated that working with this model has given him ideas for new lines of inquiry into soft-tissue cancer. In particular he noted that real-world data is not nearly as granular as the time-series outputs of our and other *in silico* models. Rather, measurements of immune cell levels may produce a few, rather than thousands, of data points over the course of treatment. Therefore what is needed are experiments that measure immune cell levels continuously, so that more data might lead to a better understanding of the particular dynamics of these cells.

6.3 Future Work

Other members of the CAS Research Group are also exploring ways to apply the general model to their domains. We are collaborating with UNC Charlotte political science professor Jim Walsh on an application of ABM to terrorism and human rights, to be presented at the annual meeting of the APSA (American Political Science Association) in September of 2010. Also, theatre professor Mark Pizzato has undertaken a study of the cognitive processes and interactions of audience members as they watch a performance on stage, using the CAS principles exemplified by our general model.

We have also taken steps to further explore the ocean ecosystem model results, as presented in section 5.5. This model not only seems to present a significant new result, in terms of which tropic levels receive the benefits of greater resources. It may also have important applications to economic markets, similar to the ‘trickle-up’ theory of wealth distribution [Degn03]. Thus, further exploration of this mapping for the general tool is warranted, and may involve researchers in economics; mathematics (to fully describe these new dynamics as a corollary to the Lotka-Volterra equations); population dynamics; and industry experts familiar with real-world ocean fisheries.

Along with Drs. Min Jiang (communications), Martha Kropf (political science), and Anita Blanchard (psychology) – all researchers at UNC Charlotte – we have undertaken a project to use these CAS tools to help understand local communities, and to build an online platform that can help to increase local political awareness and engagement.

Finally, we have begun a study for applying this CAS general model to the domain of patient care in the complex, dynamic environment of a hospital. There are a wealth of possible non-linear dynamics as a patient travels from first admission to final discharge, and we hope that applying these tools will both improve patient care and reduce the costs of such care. This work is being done in cooperation with Dr. Ognjen Gajic, MD, and others experts at the Mayo Clinic in Rochester, Minnesota.

CHAPTER 7: CONCLUSIONS

A greater understanding of threshold effects can have a positive impact on many aspects of society, across many fields of endeavor. It may be that we want to use this understanding to prevent or, at least, mitigate a threshold effect, such as with monetary policy and recessions; or perhaps as an aid to diplomacy, for more efficient intercession in a failing state. Conversely, others may want to encourage a positive threshold crossing, to help our immune system beat back cancer, or to enable students to reach a higher level of understanding when presented with new material.

Regardless of the aims for studying threshold effects, different disciplines tend to provide their own standard language in the framing and methodology used for solving problems. Researchers are trained within their discipline both to use this language as a research tool and to communicate with other members of their discipline. Interdisciplinary communication is often stifled because of these language differences. Not only are there technical terms peculiar to particular disciplines, but disciplinary languages implicitly or explicitly express ontologies that enforce categories, distinguish essential from inessential characteristics of phenomena, determine what questions need addressing, and that legitimize various methods for answering those questions.

Thus, our CAS model is intended to act as its own language, regardless of the field. Continued refinement of the general model presented, and its application to different domains, will help determine the minimum number of necessary components needed to adequately describe threshold effects in any field applicable to CAS. With the use of a general CAS tool, computer scientists can collaborate with, and learn from, researchers in many different fields. Such collaboration will allow the computer researchers to better understand both the potential and the limitations of current technology, and – more importantly – gain a clear idea of how such tools can be refined for greater use and impact. Conversely, researchers in the natural, physical, and social sciences can gain from using the common language of CAS in order to provide a new way of modeling and analyzing familiar dynamics. Designing a CAS model requires computational thinking about a complex system: What are the agents? How do they interact? Can the top-down global constraints of a model be induced – and therefore, explained – by bottom-up emergent properties?

Thinking about complex systems in terms of agents is not always easy, and it takes practice to become adept at conceptualizing a system within this framework. Often, emergent features are well known, sometimes for decades, before an agent-based explanation is forthcoming. Adam Smith wrote about the “invisible hand” to describe market’s natural tendency towards efficiencies over two hundred years ago. Yet this description was merely an observation of the system-level properties of the marketplace, not an explanation of their source.

Today there is a greater emphasis on, and understanding of, how individual attributes and agent-level interactions drive these system-level dynamics. Nevertheless, most economists were surprised by the intensity of the 2008-09 world economic crisis. Even though mathematical and computational models were in widespread use for forecasting trends and calculating risk, these models were insufficient to the task of predicting the depth of the recession, or understanding the degree of systemic danger of this system. Such models can easily suffer from too many assumptions and too many clever arguments.

If we can simulate an environment or a particular phenomenon with fewer assumptions, then we can explain that phenomenon with a greater degree of generality. By utilizing a greater simplicity in the agents' descriptions and their interactions, we can create models which are likely to be much more robust. With fewer top-down constraints, they are more flexible and, therefore, more likely to have the ability to exhibit unexpected and surprising properties. This is why CAS modeling is sometimes referred to as the "science of surprise."

The work previously mentioned on slime mold illustrates another extremely useful aspect of CAS as a common language. The solution to the pacemaker problem was not found in biology, but rather mathematics (Turing, Segal) and physics (Keller). Solutions to problems can often be found in surprising places, and it is the nature of creativity that we look to synthesize what we already know in new and sometimes unexpected ways.

A general CAS tool can help foster this creative process. But there is much more work to be done in describing a general model that is widely applicable. CAS as a former field of study is often traced back to the founding of the Santa Fe Institute twenty-five years ago. Yet even today, most models are domain-specific, and much of this research advances in a 'stove-pipe' manner, where advances in one field are seldom applied to other fields. By modeling soft-tissue cancer, political dissent, marine ecosystems, and other threshold effects, we move closer to defining a common language and grammar that transcend the particular disciplines from which the models come. In this way, a deep understanding of one domain can lead to surprising insights in another domain. The act of mapping the general model to a particular problem not only gives researchers a new way of looking at their fields, but potentially also opens up exciting possibilities so that insights and knowledge in one field can increase our understanding in others.

REFERENCES

[Axel06] Axelrod R, Axelrod DE, Pienta KH: "Evolution of cooperation among tumor cells." Proceedings of the Natl Academy of Science, 103, 36, 13474-12479, 2006.

[Axel06b] Axelrod R: "Agent-based modeling as a bridge between disciplines," in Handbook of Computational Economics, Tesfatsion L and Judd KL (eds.), Elsevier, 1st ed., vol. 2, 1565-1584, 2006.

[Beer07] Beerenwinkel N, et. al.: "Genetic progression and the waiting time to cancer." PLoS Computational Biology, 3, 11: e225, 2007.

[Carm09] Carmichael T, Hadzikadic M, Dréau D, Whitmeyer J: "Towards a General Tool for Studying Threshold Effects Across Diverse Domains," in Advances in Information and Intelligent Systems, Springer, Koronacki J, Wirzchon S, Ras Z, Kacprzyk J, eds., 2009.

[Carn07] Carneiro MV and Charret IC: "Spontaneous emergence of spatial patterns in a predator-prey model." Physical Review E., 76, 6, id. 061902, 2007.

[Cent07] Centola D and Macy M: "Complex contagion and the weakness of long ties." American Journal of Sociology, 113, 3, 702-734, 2007.

[Cham02] Chambers DJ: "Varieties of Emergence," http://consc.net/papers/granada.html. Internet, accessed Sept., 2008.

[Degn03] Degnbol-Martinussen J and Engberg-Pedersen P: Aid: Understanding International Development Cooperation. Zed Books, London, UK, 2003.

[Dréa09] Dréau D, Stanimirov D, Carmichael T, Hadzikadic M: "An agent-based model of solid tumor progression." BICoB, LNBI 5462, 187-198, 2009.

[Epst99] Epstein J: "Agent-based computational models and generative social science." Complexity, 4, 5, 41-57, 1999.

[Epst06] Epstein J: "Remarks on the foundations of agent-based generative social science," in Handbook of Computational Economics, Tesfatsion L and Judd KL (eds.), Elsevier, 1st ed., vol. 2, 1585-1604, 2006.

[Epst07] Epstein J: Generative Social Science: Studies in Agent-Based Computational Modeling. Princeton University Press, Princeton, NJ, 2007.

[Erol10] Erol K, Levy R, Wentworth J: "Application of agent technology to traffic simulation." http://www.tfhrc.gov/advanc/agent.htm. Internet, accessed Feb., 2010.

[Fern08] Fernandez SJ, et al.: "Actionable Capability for Social and Economic Systems (ACSES)." Technical report, http://www.osti.gov/bridge/product.biblio.jsp?osti_id= 933058, 2008.

[Fowl09] Fowler JH and Christakis NA: Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives. Little, Brown and Company, New York, NY, 2009.

[From05] Fromm J: "Types and Forms of Emergence." Cornell University arXiv e-print service, http://arxiv.org/ftp/nlin/papers/0506/0506028.pdf. Internet, accessed Aug., 2008.

[Gate04] Gatenby RA, Frieden BR: "Information dynamics in carcinogenesis and tumor growth." Mutation Research, 568, 2, 259-273, 2004.

[Gate07] Gatenby RA, et. al.: "Cellular adaptations to hypoxia and acidosis during somatic evolution of breast cancer." British Journal of Cancer, 97, 5, 646-653, 2007.

[Geri07] Gerisch A, Chaplain MA: "Mathematical modeling of cancer cell invasion of tissue: Local and non-local models and the effect of adhesion." Journal of Theoretical Biology, 2007.

[Hawi06] Hawick KA, James HA, and Scogings CJ: "A Zoology of Emergent Patterns in a Predator-Prey Simulation model." Computational Science Technical Note CSTN-0015, Massey University, March 2005.

[Holl99] Holland JH: Emergence: From Chaos to Order. Perseus Publishing, Cambridge, MA, 1999.

[John02] Johnson S: Emergence: The Connected Lives of Ants, Brains, Cities, and Software. Simon & Schuster, New York, NY, 2002.

[John07] Johnson N: Two's Company, Three is Complexity. Oneworld Publications, Oxford, 2007.

[Kenn01] Kennedy J, and Eberhart RC: Swarm Intelligence. Morgan Kauffman Publishers, San Francisco, CA, 2001.

[Kozu07] Kozusko F, Bourdeau M: "A unified model of sigmoid tumour growth based on cell proliferation and quiescence." Cell Proliferation, 40, 6, 824-834, 2007.

[Mall06] Mallet DG and DePillis LG: "A Cellular automata model of tumor-immune system interactions." Journal of Theoretical Biology 239, 334-350, 2006.

[Midg07] Midgley D, Marks R, Kunchamwar D: "Building and assurance of agent-based models: An example and challenge to the field." Journal of Business Research, 60, 8, 884-893, 2007.

[Nico77] Nicolis G, Prigogine I: Self-Organization in Non-Equilibrium Systems. Wiley, New York, 1977.

[Okte06] Oktem G, et. al.: "Role of intercellular communications in breast cancer multicellular tumor spheroids after chemotherapy." Oncology Research, 16, 5, 225-233, 2006.

[Oliv03] Oliver PE and Myers DJ: "Networks, Diffusion, and Cycles of Collective Action," in Social Movement Analysis: The Network Perspective, Diani M and McAdam D (eds.), Oxford University Press, Oxford, UK, 2003.

[Olza05] Olzak S and Olivier JL: "Racial Conflict and Protest in South Africa and the United States." European Sociological Review 14, 3, 255-278, 2005.

[Rosc01] Roscigno VJ and Danaher WF: "Media and Mobilization: The Case of Radio and Southern Textile Worker Insurgency, 1929-1934." American Sociological Review 66, 21-48.

[Rose85] Rosen R: Anticipatory Systems. Pergamon Press, UK, 1985.

[Rose99] Rosen R: Essays on Life Itself. Columbia University Press, New York, NY, 1999.

[Ryan07] Ryan AJ: "Emergence is coupled to scope, not level." Complex, 13, 2, Nov. 2007, 67-77. DOI= http://dx.doi.org/10.1002/cplx.v13:2.

[Sche71] Schelling TC: "Dynamic Models of Segregation." Journal of Mathematical Sociology, 1, 143-186, 1971.

[Simo99] Simon HA, Wallach D: "Cognitive Modeling in Perspective." Kognitionswissenschaft, 8, 1, 1-4, 1999.

[Stan01] Standish RK: "On Complexity and Emergence," Complexity International, Volume 09, Paper ID: standi09, URL: http://www.complexity.org.au /vol09/standi09/, 2001.

[Sump06] Sumpter DJT: "The principles of collective animal behavior." Philosophical Transactions R. Soc. B, 361, 5-22, 2006.

[Tay05] Tay NSP, Lusch RF: "A preliminary test of Hunt's General Theory of Competition: using artificial adaptive agents to study complex and ill-defined environments." Journal of Business Research, 58, 9, 1155-1168, 2005.

[VanR00] van Raan A: The Interdisciplinary Nature of Science: Theoretical Framework and Bibliometric-Empirical Approach. In *Practicing Interdisciplinarity*, edited by Weingart P & Stehr N, 2000.

[Vrie98] Vries H, Biesmeijer JC: “Modeling collective foraging by means of individual behavior rules in honey-bees.” Behavioral Ecology and Sociobiology, 44, 2, 109-124, 1998.

[Vrie02] Vries H, Biesmeijer JC: “Self-organization in collective honeybee foraging: emergence of symmetry breaking, cross inhibition and equal harvest-rate distribution.” Behavioral Ecology and Sociobiology, 51, 557-569, 2002.

[Weav48] Weaver W: “Science and Complexity,” American Scientist, Vol. 36: 536, 1948.

[Whit08] Whitmeyer J, Carmichael T, Eichelberger C, Hadzikadic M, Khouja M, Saric A, Sun M: “Computer Simulation Laboratory for Social Theories.” Proceedings of the 2008 IEEE/WIC/ACM International Conference on Intelligence Agent Technology, Sydney, Australia, 2008.

[Wile98] Wilensky U: NetLogo Flocking model, 1998. http://ccl.northwestern.edu/netlogo/models/Flocking. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[Wile99] Wilensky U: NetLogo, 1999. http://ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

[Wool02] Wooldridge M: An Introduction to MultiAgent Systems. John Wiley and Sons, Chichester, West Sussex, UK, 2002.

[Yi08] Yi F, Wei J, and Shi J: “Bifurcation and spatiotemporal patterns in a homogeneous diffusive predator-prey system.” Journal of Differential Equations, 2008. doi:10.1016/j.jde.2008.10.024.

APPENDIX A: DETAILED GENERAL MODEL DESCRIPTION

A-agent (Patch) Attributes:

Position: Whole number x- and y-coordinates; each position unique and static.

State: [0, 1], inclusive; intermediate values can be anything, but usually constitutes ten discrete, equal steps.

Active?: If true, the patch can affect other "neighborhood" patches towards the 1-state in pre-defined steps.

Threshold: Can be a single, bi-directional threshold, or two unidirectional thresholds. In the case of a bi-directional threshold, above this threshold means the patch is active, and below it is inactive. In the case of two unidirectional thresholds, then crossing one threshold (in one direction) makes the patch active, while crossing the second threshold (in the opposite direction) makes the patch inactive.

Neighborhood: The defined set of patches that a single patch can affect, or attempt to affect. Can be set by the operator, usually as either: a) the set of eight patches directly in contact; or b) a random selection of *x* patches within *y* radius. Membership in a defined neighborhood is not necessarily reciprocal.

Susceptibility: The value of this attribute determines the % chance that this patch can be affected by another patch. Can be set as either: a) a normal distribution across all patches (with user-defined mean and s.d.); b) a uniform distribution; or c) homogenous – the same for all patches (as set by the user).

B-agent (Turtle) Attributes:

Position: Floating-point x- and y-coordinates; multiple turtles can be on (adjacent to) a single patch.

Life: Whole number from [0, *lifetime* (slider)]; *Life* is initialized to *lifetime*, and reduced by 1 for each turn of the turtle; when *Life* reaches 0, the turtle is removed from the system.

Successes: Whole number count of the accumulated actions performed on A-agents, i.e., how many times this agent has moved any A-agent one step from the 1-state towards the 0-state. When this number reaches *success_rate* (slider), a new B-agent is generated and *Successes* is reset to 0.

Environment (sliders and controls):

Setup: (button) initializes all patches and turtles.

“>|”: (button) advances the simulation 1 step.

“>>”: (button) continuously advances the simulation, 1 step at a time.

Patch controls – general:

Patch Diffusion Rate: determines how many turns each patch is allowed during one simulation time step. If set to 0, patches cannot affect any neighboring patches.

Random_increase_state: [0.000, 0.100], inclusive. Chance that a single patch will randomly change its state a single step towards the 1-state, per turn: e.g., a value of 0.050 would equal a 5% chance of increasing by one step per turn.

Random_decrease_state: [0.000, 0.100], inclusive. Same as previous, except random movement is towards the 0-state.

Patch controls – susceptibility:

Distribution: one of three types: homogenous, uniform, or normal. If set to homogenous, all patches have their susceptibility set to *S_mean* (slider); if uniform, all patches have susceptibility set to a uniform distribution with min = 0 and max = *S_mean*; if normal, all patches have susceptibility set to a random number [0, 1] inclusive, with mean = *S_mean* and s.d. = *std_div* (slider).

S_mean: see Distribution.

Std_div: see Distribution.

Patch controls – neighborhood:

PatchNetwork: on/off. If off, the neighborhood of each patch is defined as the eight adjacent patches; if on, the neighborhood is defined by the *NetworkRadius* (maximum distance from patch for any neighbor) and *NetworkSize* (maximum number of patches in neighborhood).

NetworkRadius: see PatchNetwork.

RandomRadius: on/off. If on, the maximum radius for a patch's neighborhood is set between [0, *NetworkRadius*]; if off, the maximum radius is set to *NetworkRadius*.

NetworkSize: see PatchNetwork.

RandomSize: on/off. If on, the number of neighbors for any patch is a random number between [0, *NetworkSize*]; if off, each patch has exactly *NetworkSize* neighbors.

Turtle controls:

Population: initial number of turtles.

Lifetime: initial setting for Life for each new turtle.

Success_rate: the number of Successes each turtle must accumulate in order to generate a new turtle.

Reproduction: on/off. If on, then *Lifetime* and *Success_rate* are activated. If off, then the B-agents have unlimited lifetime and do not generate new B-agents.

Turns_per_tick: determines how many turns each turtle is allowed during one simulation time step.

Turtle_speed: Determines how far a turtle moves during movement. (Note: a turtle will only move if the underlying patch is fully in the 0-state.) Distance of movement is measured by the length of one patch: e.g., “1” equals the vertical or horizontal length of one patch, and “1.4” approximates the diagonal length. Heading is the current heading plus random change between [-45, +45] degrees.

Resources:

Resources were added for additional flexibility in experimenting with the dissent model, to incorporate the idea of state resources being collected and spent each turn on B-agents. If resources run out, B-agents cannot be created or maintained.

Reserves (measurement): the accumulated amount of resource units.

Total_per_turn: the units of resources collected each time step, added to reserves.

Cost_per_turn: the units of resources deducted each time step due to dissent, deducted from reserves.

Cost_per_agent: the units needed to produce one new B-agent, deducted from reserves for each new B-agent produced.

Cost_per_turn: the units needed to sustain one B-agent, deducted from reserves for each B-agent in existence during each time step.

Min_agents: the minimum number of agents maintained in the simulation. If used in combination with resources, then a negative accumulated reserve represents debt due to sustaining B-agents.

Remaining controls allow for altering visual monitors (such as patch and turtle colors, or a "running average" graph), for experimenting with certain scenarios (Behavior space), or for adding or subtracting A- and B-agents.

Figure 26 illustrates the full interface of the general simulation tool, after mapping to the ocean ecosystem. The green sliders and purple buttons are generally in four clusters. Counter-clockwise from the top-left they are: 1) Fish settings; 2) Predator settings; 3) fish-egg settings; and 4) Patch settings (food). The top left contains the main simulation window, which shows the spatial interactions and movements of the four agent-types. Along the bottom are two representative graphs: 1) on the left, the population size for the predators (purple), the fish (red), and the fish-eggs (yellow); 2) on the right, the average age, in simulation ticks, for each of these three populations.

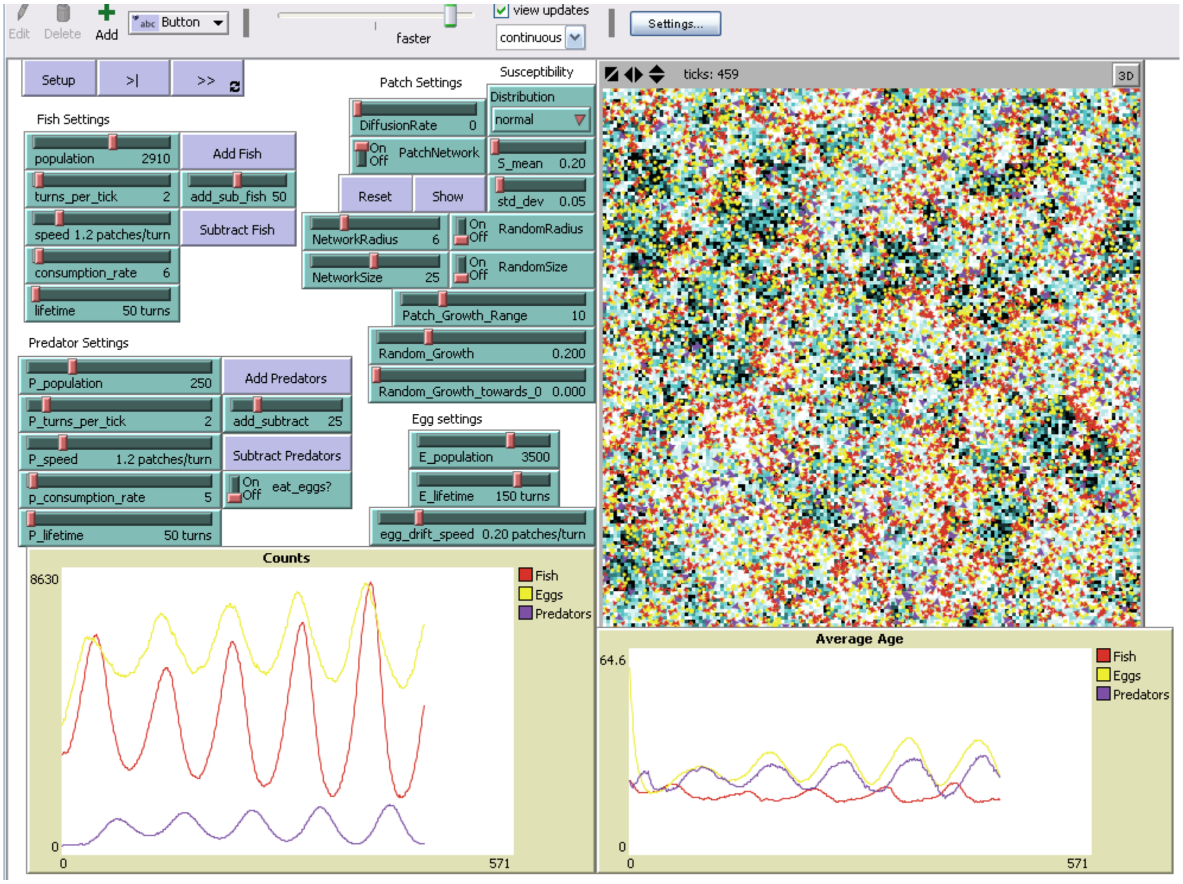

Figure 26. The general model interface, after mapping to the ocean ecosystem model. Not shown are additional controls for adjusting colors, Behavior Space, graph smoothing, and other output adjustments.