### Old Dominion University

# **ODU Digital Commons**

Computational Modeling & Simulation Engineering Theses & Dissertations Computational Modeling & Simulation Engineering

Winter 2012

# Partitioning Method for Emergent Behavior Systems Modeled by Agent-Based Simulations

O. Thomas Holland Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/msve\_etds

Part of the Computer Sciences Commons

#### **Recommended Citation**

Holland, O. T.. "Partitioning Method for Emergent Behavior Systems Modeled by Agent-Based Simulations" (2012). Doctor of Philosophy (PhD), Dissertation, Computational Modeling & Simulation Engineering, Old Dominion University, DOI: 10.25777/wqf0-ke57 https://digitalcommons.odu.edu/msve\_etds/36

This Dissertation is brought to you for free and open access by the Computational Modeling & Simulation Engineering at ODU Digital Commons. It has been accepted for inclusion in Computational Modeling & Simulation Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

# PARTITIONING METHOD FOR EMERGENT BEHAVIOR SYSTEMS

# MODELED BY AGENT-BASED SIMULATIONS

by

O. Thomas Holland B. S. E. E. 1983, Tennessee Technological University M. S. E. E. 1996, Virginia Polytechnic Institute and State University

> A Dissertation Submitted to the Faculty of Old Dominion University in Partial Fulfillment of the Requirements for the Degree of

## DOCTOR OF PHILOSOPHY

## MODELING AND SIMULATION

OLD DOMINION UNIVERSITY December 2012

Approved by:

John Sokolowski (Director)

Frederic D. McKenzie (Member)

Andreas Tolk (Member)

Bruce Copeland (Member)

#### ABSTRACT

# PARTITIONING METHOD FOR EMERGENT BEHAVIOR SYSTEMS MODELED BY AGENT-BASED SIMULATIONS

O. Thomas Holland Old Dominion University, 2012 Director: Dr. John Sokolowski

Used to describe some interesting and usually unanticipated pattern or behavior, the term *emergence* is often associated with time-evolutionary systems comprised of relatively large numbers of interacting yet simple entities. A significant amount of previous research has recognized the emergence phenomena in many real-world applications such as collaborative robotics, supply chain analysis, social science, economics and ecology. As improvements in computational technologies combined with new modeling paradigms allow the simulation of ever more dynamic and complex systems, the generation of data from simulations of these systems can provide data to explore the phenomena of emergence.

To explore some of the modeling implications of systems where emergent phenomena tend to dominate, this research examines three simulations based on familiar natural systems where each is readily recognized as exhibiting emergent phenomena. To facilitate this exploration, a taxonomy of Emergent Behavior Systems (EBS) is developed and a modeling formalism consisting of an EBS lexicon and a formal specification for models of EBS is synthesized from the long history of theories and observations concerning emergence. This modeling formalism is applied to each of the systems and then each is simulated using an agent-based modeling framework. To develop quantifiable measures, associations are asserted:

- 1) between agent-based models of EBS and graph-theoretical methods,
- with respect to the formation of relationships between entities comprising a system and
- concerning the change in uncertainty of organization as the system evolves.

These associations form the basis for three measurements related to the information flow, entity complexity, and spatial entropy of the simulated systems. These measurements are used to:

- 1) detect the existence of emergence and
- 2) differentiate amongst the three systems.

The results suggest that the taxonomy and formal specification developed provide a workable, simulation-centric definition of emergent behavior systems consistent with both historical concepts concerning the emergence phenomena and modern ideas in complexity science. Furthermore, the results support a structured approach to modeling these systems using agent-based methods and offers quantitative measures useful for characterizing the emergence phenomena in the simulations. Copyright, 2012, by O.Thomas Holland, All Rights Reserved.

To Linda, Kathryn, and Daniel. And to my father, Orgal, for his 90<sup>th</sup> birthday.

### ACKNOWLEDGMENTS

Much like the advent of computer science in the middle of the 20<sup>th</sup> Century, we have not yet seen the full maturity of the discipline of modeling and simulation. I am certain that the 21<sup>st</sup> Century will see tremendous strides in this field and I am thrilled to be part of it, even if only in a small way. For the opportunity to be part of something revolutionary I am sincerely grateful to my professors, who taught me that modeling and simulation is indeed an academic discipline.

I want to thank sincerely each member of my committee for your time and guidance; I truly admire each of you and the accomplishments in your careers.

I owe a great debt of gratitude to Dr. Bruce Copeland, a mentor and colleague, who has been a great encouragement to me when the demands of work, family and health have challenged the completion of this dissertation.

Finally, I am deeply grateful to Dr. John Sokolowski for his guidance, efforts, patience and support over the years of this work.

Thanks to you all.

# TABLE OF CONTENTS

Page

vii

LIST OF	TABLES	ix
LIST OF	FIGURES	xi
Chapter		
1. INTRO	DUCTION	1
1.1	PROBLEM STATEMENT	3
1.2	MOTIVATION	6
1.3	SCOPE AND LIMITATIONS	11
1.4	DISSERTATION ORGANIZATION	14
1.5	SIGNIFICANCE OF THIS WORK	17
1.6	DISSERTATION PRODUCTS	18
2. LITER	ATURE REVIEW AND SYNTHESIS	19
2.1	A BRIEF HISTORY OF EMERGENCE	22
2.2	MODERN INTERPRETATION OF EMERGENCE	25
2.3	CONCEPTS AND LEXICON	26
2.4	COMPLEXITY AND THE ROLE OF MODELING AND SIMULATION	ON
		30
2.5	EMERGENT BEHAVIOR AND THE AGENT METAPHOR	32
2.6	CHALLENGES TO MODELING EMERGENCE AND EMERGENT BEHAVIOR SYSTEMS	38
3. METH	OD AND APPROACH	49
3.1	ENTITY COMPLEXITY IN EBS	52
3.2	GRAPH THEORY CONCEPTS AND RELEVANCE TO EBS	55
3.3	INFORMATION THEORY (ENTROPY)	65
3.4	MECHANISMS OF FEEDBACK (SOURCES OF CONSTRAINT)	69
3.5	MEASURING CONSTRAINT	78
3.6	A WORKING TAXONOMY FOR EMERGENT BEHAVIOR	
	SYSTEMS	80
3.7	THE ROLE OF CONSTRAINT IN THE EBS TAXONOMY	89

3.8	DEFINING EMERGENT BEHAVIOR SYSTEMS
4. ANALY	YSIS FRAMEWORK102
4.1	FRAMING THE PROBLEM AND DEFINING METRICS
4.2	THE INFORMATION FLOW METRIC - CHARACTERISTIC PATH LENGTH
4.3	THE COMPLEXITY METRIC - CLUSTERING COEFFICIENT
4.4	THE ORGANIZATION METRIC - SHANNON ENTROPY115
4.5	HYPOTHESIS TESTS
SIMULA	FION, ANALYSIS AND RESULTS129
5.1	APPLYING THE EBS MODELING FORMALISM
5.2	NETLOGO MODELS
5.3	EBS METRICS IN MATLAB152
5.4	PRELIMINARY ANALYSIS
5.5	DESIGN OF EXPERIMENTS
5.6	ANALYSIS AND RESULTS
CONCLU	SIONS
6.1	SUGGESTED NEAR-TERM APPLICATIONS OF THIS WORK
6.2	OBSERVATIONS AND AREAS FOR CONTINUED RESEARCH 204
6.3	OTHER CONSIDERATIONS
REFERE	NCES
APPEND	ICES
A. C	SLOSSARY
B. N	1ATLAB CODE & SCRIPTS22
C. A	A HEURISTIC MEASURE OF ENTITY SOPHISTICATION IN EMERGENT BEHAVIOR SYSTEMS24
VITA	

viii

# LIST OF TABLES

Table	Page
1. Ferber's Agent Definition	36
2. EBS Modeling Perspectives	45
3. Predator – Prey Rules	47
4. States of the British Traffic Signals	70
5. Traffic Signal with Independent States	70
6. Bar-Yam's types of emergence	80
7. Fromm's Four Types of Emergence	81
8. The Five Types of EBS	82
9. The Characteristics of the Different Types of Entities in an EBS	98
10. EBS Specification Terms	101
11. Meso-scale System State Plots	120
12. A Minimally Constrained Case Data Analysis	123
13. Factors and Conditions for the Particle System Simulation	132
14. Factors and Conditions for the Flocking Simulation	
15. Factors and Conditions Affecting the Stigmergy Simulation	142
16. Experiment Considerations for the Inherently Constrained Particle System	172
17. Factorial Experiment for Inherent Constraint in the Particle System	173
18. Experiment Considerations for the Flocking System	175
19. Factorial Experiment for the Constrained Flocking System	176
20. Experiment Considerations for the Constrained Stigmergy System	178

21. Factorial Experiment for the Constrained Stigmergy System	179
22. H1 Results for the Flow Metric (CPL) in the Particle System	181
23. H1 Results for the Complexity Metric in the Particle System	184
24. H1 Results for the Entropy Metric in the Particle System	185
25. H1 Results for the Flow Metric in the Flocking System	186
26. H1 Results for the Complexity Metric in the Flocking System	188
27. H1 Results for the Entropy Metric in the Flocking System	191
28. H1 Results for the Flow Metric in the Stigmergy System	192
29. H1 Results for the Entropy Metric in the Stigmergy System	192
30. ANOVA Results for CPL	196
31. ANOVA Results for GCC	198
32. ANOVA Results for SE	199

# LIST OF FIGURES

Figure Pag	<u></u> ge
1. History Timeline of Emergence Related Interests	4
2. Local to Global to Local Feedback in an EBS	6
3. Dissertation Research Process	.13
4. Approach Mapped to Dissertation Chapters	.14
5. Ablowitz's Resultant and Emergent	.24
6. Nwana Agent Typology (Nwana, 1996)	.34
7. Communication in Port Automata	.41
8. A Pile of Rocks Depicted as a Graph	.56
9. An Undirected Graph and its Corresponding Adjacency Matrix	.57
10. Removing the Centermost Vertex	.58
11. Regular Graph	.60
12. Complete Graph	.61
13. Directed Graph	.62
14. A Hamiltonian Path Through a Simple Regular Graph	.63
15. Tree Graph	.63
16. Bipartite Graph	.64
17. Disconnected Graph	.65
18. Schematic of Shannon's General Communication System	.66
19. Micro- to Meso- Scale Relationship	.74
20. Inherent Constraints Are Built-in Restrictions	.75

21. Contextual Constraints Result From Objects Constituting the Environment76	5
22. Entity Constraints Result From Direct Interaction Between Entities	7
23. Stigmergic Constraint Results When Entities Communicate Indirectly	8
24. Guerin's and Kunkle's Pheromone Following Ants is an Example of Meso-scale Inherent and Stigmergic Constraints	9
25. Type 1– Nominal Emergence	5
26. Type 2– Moderated Emergence	7
27. Type 3 – Multiple Emergence	8
28. Type 4 – Evolutionary Emergence	9
29. The Context of Observation	7
30. EBS Metrics	5
31. Communication Events Indicated by Energy Changes	7
32. Inflow and Outflow	7
33. Persistent and Non-Persistent Information Flows	8
34. Some Structures Shown as Graphs and Their Characteristic Path Length11	1
35. Graphs for Clustering Coefficient	4
36. Spatial Gridding Method for System State Representation	7
<ul><li>37. Initial and Final Configurations of a Sample System Showing Meso-Scale Boundaries</li></ul>	9
38. Effect of Gridding Size on Count of Unique States $\psi$	9
39. State Occurrence of a Minimally Constrained System	2
40. The Particle System Simulation Is an Interpretation of Hooke's Law	3
41. Reynolds' Alignment, Separation, and Cohesion Behaviors	4
42. Interpreting the Components of the Flocking Simulation in Terms of the EBS Simulation Specification	7

43. Simulating Pheromone Following Ants	139
44. Information Flow in the Stigmergic System	140
45. Super-node Representation of the Stigmergic System	141
46. Interpreting the Components of the Stigmergic Simulation in Terms of the EBS Simulation Specification	: 143
47. NetLogo's BehaviorSpace	145
48. The Particle Simulation GUI	149
49. The Flocking Simulation GUI	150
50. The Stigmergy Simulation GUI	151
51. The Effect of Increasing the <i>shear-point</i> Factor	160
52. An Example of the Minimally Constrained Flocking System	161
53. An Example of the Nearly Random Flocking System	162
54. Effects of Increasing max-align-turn in the Flocking System	163
55. Flocking Nearly-Random Scenarios	164
56. Stigmergy Nearly-Random Scenario	165
57. Inefficiency in the Nearly-Random Stigmergy Scenario	166
58. Minimally Constrained Stigmergy Scenario	167
59. Effects of Increasing Pheromone Level in the Stigmergy System	168
60. Five Instances of the Minimally Constrained Case	170
61. Structure Count and NMD for the Minimally Constrained Particle System	171
62. Inherent Constraint – High I <sub>c</sub> on Left, Low on Right	174
63. Typical System Results for Increasing rest-length	174
64. Typical Flocking System Response to Increasing max-align-turn	177
65. Comparing Flow (CPL) of the Particle System as Constraint Increases	182

66. Comparing CPL at rest-length = 4 (top), 7, and 9 (bottom)
67. Plot of 1- p-value for CPL
68. Comparing Complexity (GCC) of the Nearly-Random Particle System to the System as Inherent Constraint Increases
69. Comparing Entropy (SE) in the Particle System as Constraint Increases
70. Comparing Flow (CPL) of the Flocking System as Constraint Increases
71. Comparing CPL for Varying Constraint in the Flocking System
72. Comparing Global Clustering Coefficient ( $\Omega$ ) as Constraint Increases
73. Comparing GCC at Varying Constraint in the Flocking System190
74. Comparing Entropy (S) of the Flocking System as max-align-turn Increases191
75. CPL Plot for the First 600 Time-steps at <i>pheromone</i> = 70193
76. CPL in the Stigmergy System as pheromone Increases
77. Comparing Flow in the Stigmergy System as $I_C$ Increases
78. Comparing Entropy of the Stigmergy System as $I_C$ Increases
79. CPL for Each System in Emergence196
80. CPL for Particles & Stigmergy197
81. GCC for Each System in Emergence198
82. SE for Each System in Emergence
83. SE for Flocking and Stigmergy200
84. Plot of Degree Distributions for Figure 51205
85. Poisson Distributions Generated from the Mean Degrees of Figure 84206
86. Comparison of the Minimally Constrained Particle System to the Poisson Distribution
87. Correlation of Mean Degree Distribution of the Flocking System

88. The Normalized Flow Metric.	
89. Progression of Emergence	214
C-1. The Agent Taxonomy of Moya and Tolk	
C-2. Architecture Property of the Reasoning Characteristic	250
C-3. Beliefs Property of the Reasoning Characteristic	251
C-4. Goals Property of the Reasoning Characteristic	252
C-5. Reactivity Property of the Reasoning Characteristic	253
C-6. Access Property of the Perception Characteristic	255
C-7. Accuracy Property of the Perception Characteristic	256
C-8. Memory Property of the Perception Characteristic	258
C-9. Communication Property of the Action Characteristic	259
C-10. Protocol Property of the Action Characteristic	260
C-11. Entity Sophistication Derived from the Agent Characteristics of Moya and Tolk	261
C-12. A Graph View of the Entity Sophistication Heuristic	262

### CHAPTER 1

### INTRODUCTION

From a systems perspective, the term *emergence* is used to describe some interesting and previously unanticipated patterns or behaviors while observing the evolution of a system over time. Large numbers of relatively simple interacting entities often characterize these systems and the measurements used to describe them are fundamentally different from those used to describe the entities that comprise them. For example, consider the ideal gas law. The state of an ideal gas is specified by its pressure, volume, and temperature; however, those measures are not applicable to describe a single molecule of the gas. In that case, measurements like molecular mass and bond energy are appropriate. Similarly, a house is typically measured in square feet, number of floors, etc., whereas the components that comprise the house are measured in board feet, pennies, and a myriad of other units applicable to the specific component. In either example, it is obvious that the simple collection of parts does not give rise to interesting and useful structure until those parts exist in relationship with one another. The components are the entities that comprise the system and the resulting system of interest is called the emergent.

In its simplest form, the relationships between entities are well specified and linear, such as gears that give rise to a clock. Typically, when we refer to emergence, we are more interested in the resultant behavior that arises from entities that interact in nonlinear ways, i.e., complex systems. Such complex systems abound in both the natural and man-made world. A significant amount of previous research has emphasized methods of systems implementation or specification with regard to many expected applications such as collaborative robots (Shell, et al., 2005), software design (Jennings & Wooldridge, 2000), supply chain analysis, social science, economics and ecology. While these are traditionally disparate problem domains, these and many other domains often demonstrate interesting and unexpected ensemble behaviors of patterns in space and time. The computer simulation of such systems are increasingly receiving greater interest as improvements in computational technologies combined with new modeling paradigms such as agent-based modeling support the simulation of ever more dynamic and complex systems specified at the entity level. It is now plausible for the emergence researcher to study in arbitrary detail, through computer simulation, the evolution of both real and notional systems that exhibit the emergence phenomena, with the prospect of revealing the underlying phenomenology, i.e., cause and effect relationships, which enable such systems. These systems, the behavior of which is the result of emergent phenomena, we refer to as Emergent Behavior Systems (EBS).

This research posits that there exist several types, or classes, of EBS, which can be modeled using agent-based methods and distinguished by a tuple of the form

$$E = [f(u(t)), \Omega(u(t)), S(u(t))]$$
(1)
where:

u(t) is a sample path over time t for a given set of initial conditions,

f is a measure of the principal information flows in the system,

 $\Omega$  is a measure of the complexity of the entities comprising the system, and

*S* is a measure of the Shannon Entropy of the system.

#### 1.1 Problem Statement

Although the modern understanding of emergence acknowledges that it manifests itself in different forms depending on the domain of study, the fundamental characteristic of the phenomena of emergence is that relatively sophisticated ensemble behaviors tend to arise from the interactions of relatively simple entities. Although the concept of emergence is old indeed, with the first modern scientific interest beginning in the 19<sup>th</sup> Century, the progression of computing power has fueled an explosion of research into the nature of such systems and the complexity that characterizes them. The rapid rise in the interest in emergence appears to parallel the rise in the computing capability of the latter half of the 20<sup>th</sup> Century to the present. The ability to explore large-scale interactions in simulation has even caused some to reconsider the meaning of complexity, complex systems, and simulations that model them (Varenne, 2009).

A renewed interest in the emergence phenomena has resulted with many examples of complex systems exhibiting emergence observed in both the natural and manmade world. Figure 1 depicts the rapid growth of the interest in emergence by featuring many of the notable researchers who have published related materials.



Figure 1. History Timeline of Emergence Related Interests

Modern modeling paradigms that emphasize the specification of individual components such as agent-based modeling (ABM) expediently support the computer simulation of these complex systems. Some popular examples of some natural systems modeled with ABM include aggregates of particles forming interesting material characteristics, the flocking/herding/schooling behaviors of many animals and the nearoptimal solutions achieved by many chemical trail forming insects such as ants (Bak, 1996), (Barton, 2005), (Kassner, 2009). Telecommunications, military combat, transportation, logistics and other sophisticated systems-of-systems form an ever growing list of the man-made systems that exhibit emergence and are increasingly being modeled by entity-based methods such as ABM. To explore some of the modeling implications of systems where emergent phenomena tend to dominate, this research examines three simulations based on natural systems where each is readily recognized as exhibiting emergent phenomena. To facilitate this exploration, a modeling formalism consisting of an emergent behavior *lexicon* and a formal specification for models of Emergent Behavior Systems (EBS) are synthesized from the long history of theories and observations concerning emergence. This modeling formalism is applied to each of the systems and each is then simulated using the agent-based modeling framework NetLogo (Wilensky, 1999).

To develop quantifiable measures, associations are asserted:

- between agent-based models of emergent behavior systems and graphtheoretical methods that support analysis related to information flow,
- with respect to the complexity of the relationships between entities comprising a system, and
- concerning the change in uncertainty of information, i.e. spatial entropy, observed as the system evolves.

These associations form the basis for three measurements related to the information flow, entity complexity, and spatial entropy of the simulated systems and are used to:

1) detect the existence of emergence

and

2) differentiate amongst these three systems.

### 1.2 Motivation

With increased computational capability, and with the increasing desire to create autonomous or near-autonomous systems that can exhibit complex life-like behaviors, challenges arise in applying existing system formalisms to describe the constitution of those systems that achieve their objectives through the complex interaction of a multitude of simple parts from which a higher-order behavior arises. Complexity physics, artificial intelligence, computer science, statistics, biology, and social psychology (as well as some other disciplines) are rapidly converging upon a common observation: that as analysts and designers further explore their systems of interest, the robustness, richness, and diversity observed in both natural and manmade complex systems are more than the simple sum of the components.

Interactions amongst constituent components at the local level affect, to varying degrees (both directly and indirectly), the behavior of the system of which they are a part. The resulting global level behavior in turn affects the local interactions of the components as shown in Figure 2.



Figure 2. Local to Global to Local Feedback in an EBS

This observation of the complex interaction between local relationships and global feedback, which was previously theorized but more recently substantiated through advances in simulation science<sup>1</sup>, suggest that the level to which such interactions can be observed are in fact degrees of resolution, or scale. Depending on the perspective of the observer there is some level of emergent behavior, global at that scale, at which interesting and useful behaviors are observed. Consider an insect social system such as the beehive: at a certain scale, the beehive itself can be considered as a single entity with several measurable properties such as weight, volume and amount of honey. Clearly, modeling the beehive is driven by the perspective of the modeler: for example, it can be cast from a System Dynamics (SD) perspective as a simple set of difference equations that relate the quantity of honey produced to the presence and quantity of suitable flowers, the season of the year, or geographical location. However, since SD is prefaced on the assumption that internal causal structure of a system determines its dynamic tendencies, such an approach necessarily requires an aggregate perspective(Schieritz & Milling, 2003). It is not difficult to appreciate that modeling systems through feedback relationships without sufficient understanding of the causal relationships can obscure many of the underlying components (in this example those critical to honey production) and so result in a naïve and inaccurate model. Such a "top-down" perspective applied without a satisfactory understanding of the causal relationships assumed can incorrectly subsume many critical details about the system, the environment, and their interaction. For example, we know from life-experience (because we can easily observe at the appropriate scale) that the beehive is not a single entity but rather the result of many

As opposed to computer science – the term simulation science is increasingly common in the literature where computer simulations are the primary means of discovery and encompasses issues associated with computing hardware, software, and associated analysis methods.

interacting individuals governed by a relatively small rule set. Although, at one scale of interest, it might be convenient to view the beehive as a single honey producing entity, there are limits to what this scale really tells us about the system. Difference equations can readily be stated that relate honey production to seasons of the year, but it is only by peering within the hive that we can observe that the system is actually the result of the collective behavior of many other subsystems - its components - and that these components appear to function without any knowledge or regard to a greater plan, goal, or design of the hive<sup>2</sup>. Consequently, a parametric "bottom-up" alternative approach to describing the beehive system considers the beehive at a level of resolution below that at which the beehive behavior is observed, that is at the scale of the individual bee. When we do so we see that the bee has various characteristics quite apart from those characteristics that describe the beehive. In fact, it is not even clearly discernible at all from examining a bee that at some level of population of bees a functioning beehive will come into being, i.e., emerge. This very apparent dichotomy between what is observable at the local level (bottom-up) and what is observed at the global level (top-down) is a key characteristic of emergent-based systems. Although a single bee cannot constitute a beehive, a sufficient number of bees do. This and other characteristics of such systems lead to a plethora of challenging questions such as, "How many bees does it take to make a hive?" When we turn this question around many applications, cost savings, and new capabilities are suggested. The speculative question, "How does nature specify a bee in

<sup>&</sup>lt;sup>2</sup> As another bee-related example, consider the formation of honeycomb. The hexagonal geometry of the honeycomb cell creates a nearly optimal solution to storage; maximizing area while minimizing energy used to produce the cell. It is highly unlikely that bees have embarked on a mathematical analysis to find this optimum geometry. Instead, it is far more likely that the individual goals of energy use minimization and space use maximization (combined with the increased strength of the hexagon over say the square or triangle) have resulted in this geometry after many generations of natural selection.

order to design a system to make honey?" is not so dissimilar from: "How should we specify an unmanned aerial vehicle such that a single operator can control scores of them?" or "How can we specify the number of nano-bots, given the practical limits of functionality that we can design into each, such that a tumor can be excised in a patient?" (Freitas, 2005).

It is important to recognize that, as a modeling paradigm, there is historical debate as to the necessity or even the appropriateness of entity-based models with some arguing that highly complex systems should only be represented by SD methods in order to achieve a simplification of the more complex system (Scholl, 2001), (Parunak, et al., 1998). Although SD provides a mathematically mature means to represent non-linear interactions, many systems of interest are not adequately understood in terms of causal relationships and this can lead to unsuccessful application of system dynamics. Individual-based paradigms such as ABM provide a complementary capability by emphasizing instead the specification of the components comprising a system. This is attractive to the modeler since in many systems more confidence exists in the understanding of the individual behavior of the components than in the causal mechanisms and feedback paths within the system. Macal and North emphasize that, "...systems that we need to analyze are becoming more complex in terms of their interdependencies," and they suggest that assumptions used to make certain problems "analytically and computationally tractable" lead to over simplification. They further observe that rapid advances in computational power make the individually-based models increasingly attractive (Macal & North, 2005). Schieritz and Größler 2003 consider the practical limitations of SD with regards to modeling supply chains and suggest

combining SD with ABM as complimentary methods as a simultaneous top-down, bottom-up approach (Schieritz & Größler, 2003). Schieritz and Milling 2003 compare SD and ABM, noting that practitioners in each area have in large part ignored the other. They liken SD as "modeling the forest" while ABM is "modeling the trees." Concerning SD they note that "Only rarely information used for decision making is complete, unbiased and actual." (Schieritz & Milling, 2003). The behavior of an information-feedback system is highly sensitive to the kind of information used to make decisions and the accuracy of that information." Parunak, Savit, and Riolo (1998) discuss the forms and execution of the two modeling approaches and in arguing for an agent-base approach to supply network modeling assert, "Supply networks, like most systems composed of interacting components, exhibit a wide range of dynamical behavior that can interfere with scheduling and control at the enterprise level. Data analytic approaches based on assumptions such as stationarity are not generally effective in understanding these dynamics, because the commercial environment changes too rapidly to permit the collection of consistent data series long enough to support statistical requirements" (Parunak, et al., 1998). Nevertheless, ABM is often criticized for lacking the accepted theory and mathematical rigor inherent in system dynamics. Conversely, strong arguments can be made that entity-based models such as ABM are necessary to correctly model systems such as swarms that are finding greater application ranging from unmanned mobile platforms for military purposes to microscopic devices for medical treatment. Rouff et. al. considered the problem of verifying emergent behaviors in swarm-based systems, noting that there are potentially exponential interactions required to produce desired results and as such errors can result that are difficult to predict. They

point out that to include more advanced capabilities such as learning in such a system compounds the problem, causing the system to constantly change with implications not only to behavior assurance but to testing as well (Rouff, et al., 2004). As greater capability is demanded of systems, the desired highly complex and interactive characteristics required will likely result from the emergent behavior of interacting entities. The technology exists today to build such emergent behavior systems (EBS), but the theoretical framework to support modeling, analysis and validation of such systems is still in its infancy.

### 1.3 Scope and Limitations

To restrict the potential scope of this effort, we consider those systems where the principal entities of interest comprise a homogeneous population of entities that can be specified by finite state automata and where their state transitions are governed by a Finite Markov Decision Processes. A significant contribution of this research is in the formulation of an axiomatic taxonomy of emergent behavior and subsequent metamodels fundamental to establishing a modeling theoretic for the specification of the different classes within the taxonomy. Although excursions to systems composed of entities represented by non-Markovian processes (such as entities that learn from their actions) are tempting, this is considered an unnecessary complication to gain insight into the phenomena of emergence and consequently outside the scope of this research. Nevertheless, these results can pave the way toward continued research to develop a modeling theoretic by which heterogeneous populations with more sophisticated entities can be explored.

The phenomenon of emergence is by its very nature far reaching across many disciplines and schools of thought. Figure 3 depicts the research process of this dissertation beginning with a study of the historical context and literature search, which reveals the significance of System Dynamics, Agent-based Modeling, Graph Theory, and Statistics. This is enabled by the technology of Computer Simulation whereby insight to the nature of emergence supports a working definition of Emergent Behavior Systems including supporting lexicon and taxonomy. From here, axiomatic statements concerning the nature of such systems enable the development of analytical methods that can support a modeling theoretic for EBS.

Continuing in reference to Figure 3:

- Exploring previous researchers' efforts across many application domains reveals both computational and analytical methods that support the investigation of EBS. As a first step, this effort explores the history and background of emergence, computational science, modeling, physics, and mathematics to synthesize pertinent aspects of those domains into a consistent and fundamental set of definitions and lexicon expressing terms such as *agent*, *emergence*, etc., within an EBS modeling and simulation context.
- From this synthesis of previous works, a working definition of emergence is derived and a modeling formalism for EBS is developed. Concurrently, the synthesis of previous works supports a notional taxonomy of EBS concerning sources of information flow, energy, and complexity.
- The working definition, taxonomy and modeling formalism support the utilization of an agent-based modeling and simulation (ABMS) framework to

explore the emergence phenomena and add to the understanding of EBS by demonstrating methods that examine the relationships between measures of entity complexity within the system, the flow of information, the energy state of the system, and the measurement of emergent properties.



Figure 3. Dissertation Research Process

Subsequent to the literature search and formulation of emergent behavior system taxonomy, canonical emergent behavior systems identified in the literature search are examined in simulation. Measures on the simulated systems (derived from the mathematical development related to variety, constraint, energy, etc.) are made and statistics gathered to explore characteristics that relate to classes of emergent behavior systems. Using agent-based modeling this research will consider familiar forms of emergent behavior systems with observed emergent behavior, namely;

Particle Systems

**Flocking Systems** 

Stigmergy Systems

These three systems are considered frequently in the literature and represent three recognized classes of emergent behavior systems. These are subject to the measures described above to form a categorical characterization G(u(t)). It is expected that E will form a feature set by which these systems can be partitioned such that  $E(u(t)) \rightarrow G(u(t))$ .

#### 1.4 Dissertation Organization

This dissertation is organized into six principal chapters that discuss the problem being considered, the background in complex systems science, the description of the research, foundational elements of lexicon and taxonomy, simulations, and results.

Figure 4 depicts the approach of this research and its mapping to the organization of the dissertation.



Figure 4. Approach Mapped to Dissertation Chapters

Each block corresponds to a step in the experimental process. The following describes each step of the experimental approach.

**Chapter 1, Introduction**, includes the first step of the dissertation research, i.e., *Statement of Problem.* It addresses the thesis statement, motivation, approach, dissertation organization, and significance of the research. The problem statement presented in Section 1.1 is derived from the synthesis of the background and literature reviewed in detail in Chapter 2 and can be stated in general as, "How can we explore systems where emergent phenomena dominate through modeling and simulation?"

Chapter 2, Literature Review and Synthesis presents the history and relevant research related to earlier efforts to understand, represent and analyze the nature of systems exhibiting emergent phenomena. As the second step in the research process, it examines the relevant historical context of the phenomena of emergence, past researchers' theories considering the nature of the phenomena, fundamental concepts leading to the language used to describe systems exhibiting emergence and analysis methods especially from graph theory and statistical mechanics. The ideas gleaned from previous researchers are synthesized in Section 2.3, Concepts and Lexicon, and Section 2.6, Challenges to Modeling Emergence and Emergent Behavior Systems, to establish a foundation and context for the subsequent simulation development and analysis. This foundation is built to develop a taxonomy for understanding emergent systems that leads to a working specification for modeling emergent behavior systems. To facilitate the exploration of emergent behavior systems through simulation, a modeling formalism for emergent behavior systems is derived and challenges arising from the nature of complex systems are considered.

**Chapter 3, Method and Approach** establishes the boundaries of the dissertation research, describes the development of the simulations, analysis measurements and design of experiments used in the pursuit of the hypothesis. In this chapter, concepts are further refined to support the development of the models and analysis methods to follow. Here the basis for a modeling framework used in subsequent steps is applied and allows for the formulation of a concise hypothesis explored in the dissertation.

**Chapter 4, Analysis Framework** frames the problem and defines metrics, i.e., the components required to state and test the hypotheses to examine the problem statement. The development of the three metrics to measure complexity and information flow are derived from graph-theory and the energy of the system based on Shannon Entropy is developed.

**Chapter 5, Simulation, Analysis and Results** presents the application of the modeling formalism to the simulation of the three systems using the NetLogo agent-based modeling language. The outcome of the simulations runs and subsequent analysis of data are examined. Discussion of the results of the hypothesis tests are presented and discussed with respect to the success of the approach and metrics.

**Chapter 6, Conclusions,** summarizes observations and conclusions based on the analysis of the results in Chapter 0. It also presents insights gained while conducting the research, observations on the methods and tools used, and topics for possible future research.

Following the Bibliography are three appendices. Appendix A: Glossary, presents a brief collection of terms useful for quick look-up. Appendix B: MATLAB Code & Scripts, although not inclusive of all the MATLAB code developed in this study, presents the source code for the NetLogo data reading, graph transformations, and data analysis. Appendix C: A Heuristic Measure of Entity Sophistication in Emergent Behavior Systems, presents a digression into a method to assess the sophistication of entities in an emergent behavior system (also see Chapter 3). Although not a critical component of the analysis done here, it was a result of the considerations of entity complexity and thought a worthy finding to include.

#### 1.5 Significance of This Work

This research explores the relationship between the specifications of entities comprising systems and the characteristics that emerge as those entities interact with each other and their environment. Of particular interest are the definition of *entity complexity*, the role of entity population, and the significance of relationships between entity measures and system performance measures.

There are several contributions to simulation science that result from this research, not least of which being the development of a taxonomy for emergent behavior systems based on specifications associated with complex systems, namely entity complexity, paths of information feedback, and information entropy. This lays the groundwork not just for the detection of emergence, but also for the analysis of such systems with potential application to validation of models of such systems. Another significant contribution is the comprehensive literature search and synthesis leading to a consistent lexicon for emergent behavior systems. It is not the intent of this research to offer new definitions of such terms as agent, emergent, etc., but rather to distill from earlier researchers the essence of those concepts into unambiguous and potentially axiomatically expressible definitions particularly suited to the modeling of emergent behavior systems.

### 1.6 Dissertation Products

There are six principal products resulting from this research. These are:

- 1. An emergence-focused Lexicon consistent with previous research, including a nomenclature describing the types of constraint in systems in Section 3.4
- 2. A taxonomic categorization of systems characterized by primary feedback paths leading to emergent phenomena in Section 3.6
- 3. A Formal Specification for modeling Emergent Behavior in Section 3.8
- A Heuristic to quantify the sophistication of entities comprising Emergent Behavior Systems in Appendix C: A Heuristic Measure of Entity Sophistication in Emergent Behavior Systems
- 5. Three simulations of systems that demonstrate emergent phenomena written in the NetLogo framework described in Section 5.2
- 6. Metrics useful to characterize emergence in the simulated systems described in Chapter 4 with their MATLAB implementations in Section 5.3

## **CHAPTER 2**

## LITERATURE REVIEW AND SYNTHESIS

Since Turing, researchers have envisioned computing machines with human-like qualities such as language, pattern recognition, and reasoning (Turing, 1950). Various computational models inspired by biological structures, e.g., artificial neural systems, cochlea models, artificial retina, and those inspired by information theory, e.g., Bayesian networks, simulated annealing, etc., have been implemented in the hope of imparting biological capabilities to machines. Early attempts applied a system dynamics approach, describing complex interactions through feedback loops and stocks and flows. Such models have produced notable success in aspects of pattern recognition and signal processing but typically fail to reproduce the more complex, adaptive, and collaborative behaviors observed in biological systems (Shmulevich, et al., 2002).

Seeking to explore some of the more complex behaviors observed in nature, more recent efforts have applied parametric approaches to complex systems, contemplating the group behaviors of social structures, e.g., social insects, human organizations, etc., by describing the individual entities that comprise such systems, their interaction with each other and their environment (Tosic, 2006), (Carley, 2002), (Hudlicka & Zacharias, 2004). Consequently, there has been an increase of effort within the modeling and simulation community emphasizing the representation of social systems and the autonomous entities that comprise them.

Significant advances in computer science, such as the advent of object-oriented design, multi-threading and the distributed information network of the internet, have provided a fertile soil for the growth of software agents, i.e., programming constructs that

maintain their own rule-base and instantiate themselves interactively within the software environment. This has resulted in many autonomous constructs such as "bots" and "crawlers" used in search engines and other network-based information harvesting (Bradshaw, 1997), (Nwana & Ndumu, 1999). Similarly, the field of distributed artificial intelligence (DAI) focuses on the application of cognitive agents, mini-expert systems if you will, with the ability to communicate with each other, share information and learn in order to solve complex problems (Jennings & Wooldridge, 1998).

Models that are simulations of multiple interacting entities are often interchangeably referred to in the literature as individual, entity, or agent –based models. The entity-based paradigm is receiving greater attention as a method to model complex interactive systems such as biological systems and military operations (Wong, 2006). Probabilistic Boolean networks, cellular automata, genetic algorithms, and neural networks are some traditional examples of entity-based systems designed to exhibit complex life-like behavior. Recent progress in the implementation of multi-agent systems (MAS), made possible by the ever-increasing speed and computational capabilities of digital computers, has contributed to the multi-agent metaphor as a modeling technique supporting the concept of emergence as an engineering goal (De Wolf, et al., 2004). Increasing computational capability combined with the desire to create highly interactive systems that exhibit complex life-like behaviors, including cognitive ability, has driven ever increasing complexity in such approaches and poses challenges to existing modeling formalisms.

Typically, the term emergence is used to describe some interesting and usually unanticipated state or sequence of states while observing a complex system over time.

20
This leaves the system modeler faced with the question, "What is emergent behavior and how should I deal with it?" Although there have been some significant efforts to describe emergence, there remains no universally accepted definition that results in axiomatic representation of the phenomena. As such, researchers adopt working definitions closely aligned with their domain of interest. Furthermore, the concept of emergence seems to elicit conflict between reductionist and holistic perspectives to modeling (Scholl, 2001), (Parunak, et al., 1998). Nevertheless, both camps admit that there seems to be some phenomena whereby quantitative interactions between entities in a system can lead to qualitative changes in that system which are different from, and irreducible to, the entities comprising it. The late 19th Century / early 20th Century psychologist C. Lloyd Morgan expresses this observation in his 1923 text "Emergent Evolution" (Morgan, 1923). Peter Corning summarizes the history of the concept of emergence in an excellent article in Complexity (Corning, 2002) and presents the many facets regarding definition (or the lack thereof) of the term. Previous research has frequently emphasized methods of systems implementation or specification with regard to specific applications such as robotics (Shell, et al., 2005), (Nicolescu & Mataric, 2002), supply chain analysis (Schieritz & Größler, 2003), social science (Leon, 2005), and biology (Hawick, et al., 2004). Although greatly different in application, these areas share in common the emergence of patterns of behavior and are complementary to the progress made in the related fields of multi-agent systems (MAS), DAI (Ferber, 1999) and Complex Dynamical Systems (CDS) (Boccara, 2004). Modeling and simulation provides a mathematically disciplined computational environment to describe the constitution of such systems, however without the existence of axiomatic formalisms and the associated

theoretical underpinnings, designing such systems will be similar to designing bridges before Newton wherein some designs which worked well in some applications did not always work in other applications. This was because the fundamental relationships were not sufficiently developed or understood. Just as Newton's physics provided the theoretical basis to describe the relationship between force and mass and so allowed the design of bridges with predictive properties, we seem to be at the forefront of a similar breakthrough that is based on the burgeoning discipline of modeling and simulation. It would seem that modeling and simulation stands poised to provide both the theoretical and computational faculties to explore the relationship between entity complexity and population and to enable the development of methods whereby locally simple, yet globally complex systems can be designed to exhibit intentional emergent behaviors.

#### 2.1 A Brief History of Emergence

Any modern discussion of emergence would be remiss without considering the contributions of John Holland. Known as the originator of genetic algorithms, Holland has done much to popularize the wonder and promise of emergence in his first text "Hidden Order: How Adaptation Builds Complexity", which deals primarily with the concept of agents and "Emergence: From Chaos to Order" (Holland, 1998), (Holland, 1995). With the broadest brush, Holland presents emergence as "...much coming from little" and notes that "it is unlikely that a topic as complicated as emergence will submit meekly to a concise definition...", and so he does not present a concise definition. Rather Holland's text, although rich in content and perhaps the best of any yet written, is more of a call to further research into the nature and analytical representation of emergence. Holland states, "In short, we will not understand life and living organisms until we

understand emergence." It is within the descriptions of emergence that Holland sets forth that a study (and broad definition) of emergent behavior systems begins to take shape, specifically that such systems fundamentally are "...composed of copies of a relatively small number of components that obey simple laws."

The casual reader might be inclined to attribute to Holland the notion of emergence or the hypothesis that emergence may be a fundamental natural phenomena. However, the literature survey reveals that the concept has rather deep roots in philosophy, biology, social sciences, and mathematics. Not to disregard the significant contributions of Holland, in order to unearth a consistent intuition of emergence and to eventually scope a definition by which emergence phenomena can be quantitatively and parametrically represented, it is beneficial to understand the very beginnings of the interest in emergence. Searching beneath the works of Lorenz (Lorenz, 1963), Ashby (Ashby, 1956), and Turing (Turing, 1950), leads to the efforts of Reuben Ablowitz in the early 20<sup>th</sup> Century and who, in 1939, set forth a philosophical theory of emergence (Ablowitz, 1939), and, indirectly at least, introduced the key concepts of scale and observation to this theory. Citing references as early as 1843 that addressed philosophical issues such as causation. Ablowitz drew on the works of 19th and early 20<sup>th</sup> Century scientists and philosophers. In the collection of his thoughts, we can see the beginnings of what will later become known as *complexity theory*. Ablowitz asserts a definition of emergence that begins by making the distinction between "emergents" and "resultants". He states that an "emergent" is a "new quality of existence which results from the structural relation of its component parts." A "resultant" is a "property of the combination that can be foretold exhaustively from the individual elements."

As depicted in Figure 5 below, Ablowitz illustrates the distinction between the two metaphorically by considering bricks used to build a house: "...the weight of the house is a resultant of the individual weight of the bricks, but the peculiar characteristic of being a new entity called a 'house' is an emergent; you could not possibly tell from looking at a single brick what manner of object a house would be, unless you considered the structural relation the bricks were to assume." He points out that resultant properties are additive, whereas emergent properties are not. Indeed, this observation foreshadows the advent of modern chaos theory in the mid-Twentieth Century with its nonlinearity and sensitivity to initial conditions which became apparent with the advent of computers capable of many repeated calculations.



Figure 5. Ablowitz's Resultant and Emergent

Upon consideration, we can see that Ablowitz's philosophical study of emergence suggests a mathematical treatment; we can observe that the properties of the resultant are necessarily linear, whereas the structure imparted by interactions amongst components results in the inherently non-linear properties of the emergent. Albowitz's theory of emergence foreshadows the modern scientific interest in emergence and lays a philosophical foundation on which the application of chaos and complexity theory intertwine with computer science in the modern discipline of modeling and simulation.

### 2.2 Modern Interpretation of Emergence

While naturalists were considering the intricacies of the interactions of many entities, engineers were considering the practical aspects of statistical mechanics to electronic communications. In particular, Claude Shannon showed in 1948 that there was a duality between energy and information (Shannon, 1948). This discovery has resulted in a measure of uncertainty for the correct transference of information between transmitters and receivers, now known as Shannon, or Information Entropy. Information Entropy becomes particularly salient in regards to the study of EBS when considered from the perspective that such systems are essentially communication systems, owing to Ablowitz's assertion that that emergence is about the interrelationships between system components since, in most systems of interest, those interrelationships are manifested by the communication of some information. Within an EBS, communication can occur either directly between entities, or indirectly by the entities modifying their local environment i.e. stigmergy (Grasse', 1959), (Theraulaz & Bonabeau, 1999), (Chira, et al., 2007). Shannon's Theory does not rely upon a particular means of communication aside from the conceptual model of there being a transmitter, a receiver, noise source, and a channel by which communication can occur. The information shared amongst entities should be of great interest to the modeler charged with representing an EBS, because what the entities in the system do in response to that information has a great deal to do with the overall system behavior.

Shortly after Shannon's formulation, Ashby (Ashby, 1956) explored the nature of controllability in a complex system and noted that within any system, there is a possible variety of processes, but it is by the selective constraining of that variety by which useful results are obtained. Ashby's Law of Requisite Variety relates the complexity of an

overall system to the complexity of the entities comprising that system. Statistical mechanics relates states of individual entities to overall system states. The Law of Requisite Variety similarly provides a means to express the constraint required on the entities of a system to achieve system performance. The two together suggest underlying phenomena critical to system behavior that is manifested by observable, i.e., measurable, self-organization. This idea has continued to be developed in more modern philosophical perspectives with regard to the validity of deductive processes; the idea here is that simpler elements are not sufficient to understand complex systems, but perhaps the dynamism of those components with dependence on their context gives rise to emergent explanations of the perceived phenomena (Baas & Emmeche, 1997). This perspective relies on the presence of an observer and suggests that emergence is a phenomena reliant on the experience of the unanticipated or "surprise". A counter perspective is that emergence is a phenomena that either exists or does not, regardless if an observer is present. For example, one could argue that humanity did not understand gravity for many millennia and perhaps is still does not understand it today. Nevertheless, gravity remains prevalent and so far, nothing has been observed to fall up. Other modern thinkers argue that emergence can be thought of as a quality or property of a system that exists whether the observer is present or not and the surprise of an observer has nothing to do with the existence of a property (Abbott, 2006).

# 2.3 Concepts and Lexicon

The first step toward a theoretical framework for the modeling and simulation of EBS is to establish a working glossary wherein the language of description translates into mathematical expression. This is needed not only to maintain consistency within a

26

disciplined study but any axiomatic expressions will need to be consistent with the mathematics to be developed. Largely, researchers in system dynamics, complexity, chaos, and computer science have made, when looked at in total, significant inroads to the mathematical definitions relative to emergent behavior systems. Unfortunately, because of little cross-disciplinary interchange (Scholl, 2001), a fair amount of energy must be expended to recognize that the varied terminology of traditionally distinct domains frequently express essentially similar ideas.

This student has set about to view the varied definitions for emergence related terms in light of their mathematical basis whenever possible, and as such those terms are stated with careful consideration to an eventual mathematical expression. Definitions are vital to developing any theory; however, it is not the intent of this research to introduce just another set of definitions. Instead, the definitions developed in this research are a synthesis of the historically significant work of noteworthy researchers with the hope of reducing the present ambiguity in the science and establishing a foundational glossary useful to the modeling and simulation of emergent behavior systems. These definitions, drawing on the commonalities that have persisted over time, reflect the current language of complex systems, system dynamics, and modeling and simulation. In some cases, a simple one-sentence definition is not sufficient to communicate clearly the implications of the definition. In those cases, amplifying information is provided. It should be stressed however, that wherever possible, the foremost goal is the pretext that whatever is defined in words must be ultimately expressible mathematically. As an example, consider a general definition of an *autonomous agent* as given by Franklin and Graesser:

"An autonomous agent is a system situated within and a part of an environment

that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future." (Franklin & Graesser, 1996)

They even admit that this definition is "too large to be useful as is." Consider in contrast, the mathematically concise definition offered here, based in large part on the work of Crutchfield (Crutchfield, 1994):

An agent is an autonomous stochastic dynamical system that attempts to build and maintain a maximally predictive internal model of its environment within the context of its inherent sensors, behavior sets, and effectors.

Similarly, the definition of emergence asserted in this dissertation is based on concepts in the works of Crutchfield (Crutchfield, 1994), Shalizi (Shalizi, 2003), and Boschetti et al (Boschetti, et al., 2005):

Emergence is that phenomena in which patterns that are observed at a globallevel arise solely from interactions among lower-level components acting on rules that are executed using only local information without reference to the global pattern.

Armed with such definitions, a modeling specification is offered in Chapter 3 to facilitate the exploration of emergent phenomena through a structured simulation framework.

We will see in Chapter 3 that the EBS Taxonomy and the range of constraints on the variety of the system illustrate the breadth of the emergence phenomena; however, some reasonable generalizations can be made here leading to a convenient, if not technically precise, definition of emergence. A wealth of previous thought on emergence suggests consensus that an EBS should consist of multiple entities and that these entities share information in such a way as to constrain their inherent variety and to dissipate their energy (or information) to achieve a (perceived useful) equilibrium. This way of thinking about EBS is couched in statistical mechanics and system dynamics and so suggests a sense of mathematical formalism to describe them. Additionally, the ideas explored by some recent researchers, benefiting from the parametric simulations made possible with modern computing technologies, offer a complementary view of EBS. However, nowhere in the discussions of EBS related works can we find the requirement that the entities comprising such systems must be of a certain kind, or be of a certain degree of complexity. With these concepts in mind, we are emboldened to assert a *context-free definition* of an Emergent Behavior System:

An Emergent Behavior System is a natural or synthetic system that produces observable changes in state in the form of either spatial or temporal patterns resulting from interactions between the components comprising the system.

This high-level definition would seem to address all the ideas noted by earlier researchers and spans the breadth of systems suggested by the Taxonomy from those comprised of very simple components to those where the individuals are varied and very sophisticated in their inner workings. This definition speaks to the nature of individual entities that have definition independent from their system (micro-level), the behaviors that govern the relationships between entities (the meso-level) and the global scale behavior that emerges (macro-level). The key word is "interactions" because that is what separates the linear resultant characteristics of a collection of entities from the complex emergent characteristics of a system of entities. It includes the thinking of many modern researchers that natural systems and increasingly manmade systems are increasingly heterogeneous in their composition with their constituents interacting without consideration of an overarching goal (Mittal, 2012).

This is called a "context-free definition" because it is highly abstracted; useful for conceptual considerations but still too broad for a working definition within a specific context. We will see that we cannot escape the dependence on context (because of mechanisms of constraint and feedback), and that there exist classes of EBS that lend themselves to different modeling approaches (Holland, 1998). Nevertheless, the context-free definition supports the continued development of a modeling and simulation lexicon from which we may develop a useful modeling formalism by which simulations of EBS can be described. Furthermore, we can begin with this definition to show that such systems can be abstracted to a useful degree by considering the complexity associated with the entities comprising the system, their interactions with each other, and the major paths of information feedback in the system. In this way, the context-free definition does help us develop an intuition about modeling EBS.

## 2.4 Complexity and the Role of Modeling and Simulation

Michel Baranger of the MIT Department of Physics asserts in "Chaos, Complexity, and Entropy" (Baranger, 2008) that "...the enormous success of calculus is in large part responsible for the decidedly reductionist attitude of most twentieth century science, the belief in absolute control arising from detailed knowledge." In short, Chaos was developed because the determinism of calculus fell apart with the discovery of certain equations with hypersensitivity to initial conditions such as the now famous Lorenz equations of weather (Lorenz, 1963). This has great impact in the area of dynamical systems as many examples exist of systems where uncertainties in initial conditions lead to exponential changes in system state. Chaos in the spatial domain manifests itself in fractals, whereas chaos in the time domain results in a complex system-dynamics. Barringer connects spatial chaos (fractals) with temporal chaos and observes, "Every chaotic dynamical system is a fractal-manufacturing machine. Conversely, every fractal can be seen as the possible result of the prolonged action of time-chaos." The mathematical relationship amongst components, whether we study chaos or complexity, is inherently non-linear. Interestingly, the concepts of statistical self-similarity and chaotic dynamical systems align nicely with the ideas of scale (or resolution) in the study of emergent phenomena. The field of complexity then can shed a great deal of light on the nature of emergent phenomena, and therefore the properties of complex systems can provide a basis for a deeper understanding of emergent behavior systems. At this juncture it is valuable to enumerate the properties of complex systems as this will have direct bearing on the approach taken in developing an emergent behavior systems taxonomy. Although the notion of a complex system remains to be further refined, Baranger (Baranger, 2008) identifies the following properties that characterize such systems:

- 1. Complex systems contain many constituents interacting nonlinearly.
- 2. The constituents of a complex system are interdependent.
- 3. A complex system possesses a structure spanning several scales.
- 4. Complexity involves interplay between chaos and non-chaos.
- 5. Complexity involves interplay between cooperation and competition.

Although complex systems are often associated with what might be called emergent phenomena, Baranger's statements do not necessarily imply that EBS must always be complex. Indeed, although the behaviors exhibited by an EBS can be very complex, such complexity need not be the result of complicated components. For example, Rodney Brooks (Brooks, 1991) has shown that highly adaptive, seemingly intelligent behavior can arise from simple stimulus/response entities or "reactive agents".

In his seminal text "Introduction to Cybernetics", Ashby, defines *variety* in a complex system by considering the question of, given a set, how many distinguishable elements does it contain? Ashby's *Law of Requisite Variety* can be loosely stated that for appropriate regulation in a system as, "the variety of a regulator must be equal to or greater than the variety of the system being regulated." The implication is that a controller must have at least as many degrees of freedom as what it is controlling. The greater challenge may be found in the application of Ashby's Law to the discovery or specification of the variety of a system and its components.

The upshot here is that the complexity of a system can be analyzed with respect to measures of variety and constraint of the system and that the Law of Requisite Variety provides a means to relate the complexity of constituent components, i.e., the entities that comprise the system, to the complexity of the system. Ashby's work defined both variety and constraint and relates the variety possible to the variety expressed through his Law of Requisite Variety.

## 2.5 Emergent Behavior and the Agent Metaphor

As is seen in the previous discussion, it is difficult to understand what it means to be emergent. Typically the terms emergence, emergent, or emergent phenomena come up in discussions of complex adaptive systems and multi-agent systems. Current literature seems to suggest that emergence results when agents interact, but what do researchers really mean by "agent" and consequently what level of entity complexity or sophistication distinguishes agents from other components of a system? No modern treatment of systems composed of interacting entities can ignore the concept of agency. Germaine to our consideration of the term "agent" in regards to the primary entities in a system exhibiting emergent phenomena, we must draw some distinction between terms such as "agent-based", "entity-based", and "individual-based" models, especially where this applies to the modeling of EBS.

Clearly, Ablowitz had no idea of what the modern thinking on agents would be and considered even very simple entities as components of an EBS (e.g., bricks). A review of more recent literature would suggest that almost any component of a system can be represented by an agent metaphor; the most simple components being modeled as agents with profoundly limited sense and respond behaviors (reactionary agents) and the most sophisticated components modeled as cutting-edge artificially intelligent engines (cognitive agents). One must ask, "Why agents?" and along with that, "Is then everything some kind of agent?" The answer to both questions necessarily degenerates to a philosophical debate founded on inconsistent and poorly defined terms used in rather broad applications. Broad and inconsistent terms give system modelers intense headaches. Since so many like to create systems of agents and then marvel at the often unexpected yet interesting ensemble behavior, let us first consider the components of a system in general and then the concept of agent specifically.

The current literature is replete with references to the term "agent" whether it is with regard to "agent-based" (Parunak, et al., 1998), "agent-oriented" (Kim, et al., 1999), "autonomous agent" (Franklin & Graesser, 1996), rational agent (Russell & Norvig, 1995), "multi-agent" (Ferber, 1999), etc. Unfortunately, one must struggle a little to identify the unique characteristics of these terms when reading current literature, and many working definitions seem to be tightly coupled to the specific problem domain that is the area of interest of the respective researchers. This can (and has) lead to much inconsistency and confusion amongst the varied domain perspectives. For example, Graesser and Franklin presented a formal definition of an "autonomous agent" as a means to distinguish software agents from other programming constructs. Researchers have made some progress toward a more general description of such behavior based entities, such as Ashlock's and Kim's examination of representation of "finite state agents" by Cellular Automata (CA) which found that using the CA representation resulted in less cooperation in solving the prisoner's dilemma problem (Ashlock & Kim, 2005). With regards to "software agents" Nwana states, "We have as much chance of agreeing on a consensus definition for the word 'agent' as AI researchers have of arriving at one for 'artificial intelligence' itself - nil!"(Nwana, 1996). As such, she presented her own agent typology derived from three minimal characteristics as shown in Figure 6.



Figure 6. Nwana Agent Typology (Nwana, 1996)

Emergent phenomena are often associated with MAS. Not to be omitted, Ferber observes in his treatise on Multi Agent Systems that the term 'agent' "is used in rather a vague way" (Ferber, 1999). He then presents what he says is a "minimal definition" as summarized in Table 1. Key characteristics have been emphasized in bold. Being a "minimal definition" means that there is essentially no limit on what can be considered an agent. However, is the agent metaphor and the specification of MAS the best choice for the modeling of EBS?

Beginning with Ferber's definition of agent within MAS, perhaps we can synthesize a workable definition of the fundamental entities comprising a model of an EBS. First, it is reasonable that an agent must exist within some **environment**: that would imply it has no meaning outside of its environment (or at least not the same meaning in different contexts) and the agent is itself something apart from its environment; so, an agent is not the system itself, but rather a part of a greater system. Secondly, not only do agents **communicate** according to Ferber, but they also communicate directly with other agents. This implies that an agent must not only be able to transmit information but also be able to receive it. Notice that this does not preclude communication with something other than agents as well. So, what does it really mean for agents to communicate? At its lowest level, communication can be defined as the transference of data. An agent is a physical or virtual **entity** that

- a. is capable of acting in an environment,
- b. can communicate directly with other agents,
- c. is driven by a set of tendencies (in the form of individual objectives or of a satisfaction/survival function which it tries to **optimize**),
- d. possesses resources of its own,
- e. is capable of **perceiving its environment** (but to a limited extent),
- f. has only a **partial representation of this environment** (and perhaps none at all),
- g. possesses skills and can offer services,
- h. may be able to **reproduce** itself,
- i. has behavior that tends towards **satisfying its objectives**, taking account of the resources and skills available to it and depending on its perception, its representations and the communications it receives.

Table 1. Ferber's Agent Definition (Ferber, 1999)

Ferber suggests no restriction on what form that communication must take or by what medium it must be achieved, but he does indicate that it is direct. This implies that for something to be an agent it must be able to transfer data directly to other agents and not only by some consequential or secondary means; that is, agent-A sends a message intended to be received by agent-B; however, do entities comprising and EBS need such awareness? This is not a trivial consideration since communication in natural systems manifests both directly and indirectly. In some cases, one entity purposefully communicates to another entity with the expectation of a response. In other cases, the communication is simply the passing of information through secondary media such as the posting of traffic signs by road designers or the pheromones deposited by certain social insects. In either case, information is transmitted and received, but in the first case, there is intent by the transmitting agent to communicate and an expectation that a receiving agent will respond. In the second case there is no such intended recipient or expectation, but data rather appears as a part of the environment whether it is created or received by entities in that environment (Theraulaz & Bonabeau, 1999). A key characteristic of EBS is the balance of variety and constraint that leads to organization, so in the EBS there is no requirement that an entity need know that the information it has just received is from another entity; it only needs to act on (or in response to) the information it receives. So this begs the question, what constitutes information in an EBS? The EBS modeler might consider that information is anything that imposes some constraint on an entity in a system such that there is elicited a change of state in that entity, i.e., any actionable data that facilitates a change of state in an entity. Either form of this working definition of information is necessarily recipient-oriented, that is the recipient's response (or lack of it) determines whether in fact received data is information, but the form of the data or its media is inconsequential. Either a radio transmission or a stone to the head may be sufficiently termed information if the receiving entity perceives it as such.

Ferber's definition requires an agent to be **goal seeking** or seeking to optimize some utility function. To do so would require that it have some way of ascertaining the extent to which its actions allow it to attain its goal. Interrelated, **perception** relates to an agent's ability to sense (or measure) to some degree its environment or other entities and so leads to an internal representation within the agent of its world. Ferber points out that this perception is typically incomplete; a reasonable assertion given the practical limits and uncertainties associated with any sensorium. This is fitting with our previous observation that in an EBS, responses are based on local information and entities have no understanding of the larger system. Having **skills**, and **offering services** implies that something an agent can do is of use to something else and that it can communicate (or it is understood) what it can offer to others. Reproduction is the only "may" meaning that it is not critical to the definition – just if it can, it can still be an agent.

Precisely what is an agent, or constitutes agency, is not then a matter of assertion, but rather a matter of value. With regard to EBS for example, particle systems establish complex relationships and give rise to synergies of scale and threshold effects producing emergent phenomena such as avalanches (Corning, 2002). Typically one would never refer to a pebble as an agent, but agent methods can be (and increasingly are) utilized in the implementation of models of such systems. Unfortunately, given the many available definitions of agent, any and all of these components are often implemented as agents, leading to the need for innumerable modifiers. To facilitate this dissertation, we suggest that an EBS modeling lexicon needs to distinguish between the "agents" in a system and the "agent-based methods" of implementing them. (This is addressed in Chapter 3.) Therefore, for our purposes, we will speak generally of the *entities* that comprise a system, as opposed to some modified term of agent, and relegate the term *agent* to that of a particular programming paradigm.

## 2.6 Challenges to Modeling Emergence and Emergent Behavior Systems

It is good to recognize that emergent behavior (especially where we are concerned with the discovery of emergence or with the design of systems with intentional emergent behavior) can result from either collectives of interacting simple entities, such as particles of sand that make up a dune, or from vastly more sophisticated entities such as people in a society. Similarly, computer simulations of entities give rise to emergence phenomena and range from simpler cellular automata (CA) to collectives of more sophisticated computational entities such as the afore discussed agents with all the complexity that the modern term connotes. In both naturally occurring and synthetic emergent systems we typically observe interactions amongst three readily recognizable components; 1) the boundaries of the system or the environment, 2) the less sophisticated entities within that environment (which we will later define as objects), and 3) the more sophisticated entities, or Actors, that tend to affect objects and other actors in the environment.

Although it is common to assume that emergent behavior systems have their beginnings in the study of complexity, it is rather a "chicken-or-the-egg" issue. Clearly, the idea of emergence (at least philosophically) predates modern thoughts of complexity, yet it has only been the advances in recent history, primarily the advent of modern computing capability, that has fueled the explosion of research into complexity and as such lead to, if not the rediscovery, at least the renaissance of emergence. This new interest in the emergence phenomena has taken hold both from the perspectives of intentional emergence, i.e., where a system of entities are created with the intent to achieve an emergent system behavior, and unintentional emergence, where some unexpected system characteristic or behavior is observed or measured as data is garnered about the system. In either case, the ability to explore the emergence phenomena has been primarily facilitated by the rapid increase in computing capability.

It is no wonder then that the complexity made possible with modern computers

would lead to emergent behaviors, whether intentional or not. Whether the goal is to detect, observe, or predict emergent phenomena in real systems, or to design systems with specific emergent properties, a great deal is being done without a working "calculus of emergence" - a situation not unlike civil engineering in the absence of Newtonian Physics as emphasized by Lyons & Arkin: "There is an analogy with the history of civil engineering: bridges and other major structures were constructed for thousands of years before the necessary mathematical tools were developed to guarantee their performance. In the 19th and 20th centuries, as such projects became more ambitious; some spectacular failures ensued due to the absence of effective performance guarantees." (Lyons & Arkin, 2003)

As computing capability has increased, our ability to create ever increasingly complex systems has increased. As early as the mid-1980s systems engineers observed the problems associated with systems of increasing complexity. As Harel states, "The literature on software and systems engineering is almost unanimous in recognizing the existence of a major problem in the specification and design of large and complex reactive systems. A reactive system, in contrast with a transformational system, is characterized by being, to a large extent, event-driven, continuously having to react to external and internal stimuli." (Harel, 1987)

Given the rise of complexity in such systems, assuring the correct emergent behavior poses a challenging task. Lyons and Arkin explore this very problem with regard to robot-environment interaction and develop an analytical approach (based on the formalism of port automata) that reduces the combinatorial scale between robot actions and the environment (Lyons & Arkin, 2003). Lyons and Arkin applied the formalism of Port Automata (PA) to the problem of robot-environment interactions with the hope of achieving some kind of performance guarantees. The authors describe the "open-ended" nature of robot-environment interactions and the need to assure behavior in this system. Their approach was to utilize the PA formalism emphasizing the role of communication events. Figure 7 depicts a way one PA can communicate with another in a single communication event. The small circles represent communication ports and the directed links communication events.



Figure 7. Communication in Port Automata

The similarities of this representation with that of both agent-based simulations and system dynamics are obvious: the similarities to agent-based simulations include the individual description of each PA, and similarities to system dynamics arise from the communication links, especially those cases of self-communication, i.e., loops, that in PA theory allow the storage of internal variables (a form of memory). Furthermore, the PA shares some obvious similarities with current definitions of agents such as the ability to communicate with each other and actions taken based on input. The A formalism for PAs is given by:

$$PA = (Q, L, X, \delta, \beta, \tau)$$
 (2)  
where

- Q is the set of states
- *L* is the set of ports
- X is the event set for each port  $X = (X_i | i \in L)$
- $\delta$  is the transition function
- $\beta$  is the output map for the ports
- $\tau$  is the set of start states

More recently, Varenne has explored the role of computer simulation with regard to complex systems modeling. Varenne notes that prior to the ability to create sophisticated computer simulations, the very understanding of complexity has heretofore striven to represent complexity by modeling with simple notations and allowing the model complexity to manifest in the implementation (Varenne, 2009).

The works of Wolpert et al address the need to relate individual agents to the performance of the collective they form (Wolpert, et al., 2000). The authors' approach was to create a top-down (system level) feedback of the global system (the world) to the entity level, i.e., assert a global utility function as a means to assess each agent's contribution toward achieving the desired collective goal according to:

$$WLU_{\eta}(z) \equiv G(z) - G(z_{-\eta}, CL_{\eta}^{\nu})$$
(3)

where,

- $\eta$  is the designation of a specific agent,
- z is the state of all agents, and
- $z_{-\eta}$  is the state of all agents other than  $\eta$

The "wonderful life utility" described by  $WLU\eta$  serves to moderate the affect of

an agent  $\eta$  on the system by measuring the difference between the world as it is and the world without the presence of the agent  $\eta$ . In Wolpert's work, *G* is a global utility function and *CL* is a kind of "clamp" that effectively keeps an agent from changing. In this way, the agents are continuously assessed as to their contribution to achieving the global goal. This approach achieves a practical method by which agents of a certain type can be systematically adjusted, driving them toward a global behavior and has been successfully applied to multiple domains, e.g., autonomous rovers, constellations of communication satellites, and data routing. However, this approach does not for instance; provide a measure of the relationship between agent specification and the number of agents needed to achieve the global behavior.

Whether the intent is to produce a simulation of an existing system, or to achieve an advanced behavior, e.g., autonomous weapons, simulation is critical. In the latter case, the simulation is the system. Lyons and Arkin assert, "Without the establishment of strong formalisms to describe emergent-based systems, advances in this area will not keep pace with needs, and perhaps more importantly, it will be very difficult to assure the behavior of such systems, which could lead to unintended and in some cases devastating results."

Common usage of the term *emergence* often refers to the observation of higherlevel properties and behaviors in a system that, while obviously originating from the collective dynamics of that system's components, are neither to be found in nor are directly deducible from the lower-level properties of that system. To some extent, this research attempts to reconcile what might appear as opposing perspectives of emergent behavior, each perspective previously suggesting a different and often conflicting modeling approach. The first perspective sees simulations of systems that are characterized by observable system behaviors but where the underlying entity characteristics are generally unobservable. This perspective encourages models that are top-down, closed, and generally lending themselves to some form of a *global utility function*. Causality is presumed and the system model emphasizes causality. The second perspective takes the view that a good deal can be known about the entities comprising a system but the full nature and extent of the interactions between the entities is not known. The key distinction between these perspectives is that the first is top-down whilst the second is bottom-up, i.e., generally specified and observable at the entity level. The second perspective typically lacks sufficient global-level observations to support statistical representations of the behavior of the system, as is often the case in many natural occurring events. Table 2 summarizes the distinction between the two perspectives.

Top-down Perspective	Bottom-up Perspective
<ul> <li>System specified at the global level</li> <li>Global behaviors are what they are</li> <li>Entity characteristics unknown</li> <li>Entity interaction rules unknown</li> <li>System Dynamics</li> </ul>	<ul> <li>System specified at the entity level</li> <li>Desirable Global behaviors are specified</li> <li>Entity characteristics are known</li> <li>Entity interaction rules are known</li> </ul>
Boundless trade space	<ul><li>Agent-based Simulations</li><li>Bounded trade space</li></ul>
	-

Table 2. EBS Modeling Perspectives

To illustrate these two modeling perspectives, consider the oft-cited classic system dynamics example of the predator-prey modeling of foxes and rabbits (Hawick, et al., 2004). An example of the top-down perspective results from the naturalist's observations of an ecosystem consisting of two populations, one of rabbits, another of foxes. The top-down perspective would involve some estimation of both populations over some time in which the increases and decreases would be noted with a basic assumption that the dynamics are correlated. In fact, this is exactly the system dynamics approach in which a set of difference equations can be stated that describe the observed behavior such as:

$$P_{Rn} = P_{R(n-1)} + B_R * P_{R(n-1)} - A * P_{R(n-1)} * P_{F(n-1)}$$
(4)

$$P_{Fn} = P_{F(n-1)} - D_F * P_{F(n-1)} + A * P_{F(n-1)} * P_{R(n-1)}$$
(5)

In (4) and (5),  $B_R$  is a constant representing the rabbit birth rate,  $D_F$  is the fox death rate,  $P_R$  is the rabbit population,  $F_R$  is the fox population, and A is an empirically defined interaction constant. The state of the system is incremented with each n step. As can be readily seen, the rabbit population will decrease in the presence of foxes, with a greater number of foxes causing a greater decrease in rabbits. Arguments can be made as to the validity of this model, but no one can deny that a great deal of ecological science is aggregated in the assignment of the interaction constant *A*. Additionally, the birth and death rates are assumed to be effectively independent of any other factors, which is a modeling simplification that may or may not be valid. Although a system dynamics approach as this one has clear advantages (such as computational speed resulting from mathematical simplicity), an argument could be made that such an approach does not "faithfully represent" the system of interest and so can lead to many false conclusions about the problem domain. At the very least, real rabbit and fox populations would have to be observed over time to determine valid values for the constants *A*, *B*, and *D*. Also, one could argue that this modeling approach is inherently limited and potentially misleading: *A*, *B*, and *D* could in fact aggregate many unknown variables in the environment. Of course, such arguments are always subject to the intended use of a model.

From the bottom-up perspective, the naturalist would observe the characteristics of the individual entities, i.e., foxes and rabbits, and specify a behavioral model at the level of the individual. Such a modeling approach allows the modeler to focus on the entities (with the assumption that it is easier to specify the components of a system than the phenomena governing the interaction of those components) and emphasizes the rules that govern their interaction. Hawick, Scogings, and James used just such an approach to discover interesting emergent characteristics of predator-prey systems, specifically the emergence of what they termed "defensive spirals", i.e., spatially spiral patterns formed by the prey as they flee the predators (Hawick, et al., 2004). The following table lists the characteristics that they used in building fox and rabbit models, resulting in a predatorprey system at the entity level.

Prey or "rabbit" rules	Predator or "fox" rules
Move away from a fox if the fox is adjacent	Eat a rabbit if the rabbit is adjacent
Breed if a rabbit is adjacent and less than 5 rabbits are nearby	Move towards a rabbit if it is nearer than 80 spatial units and this fox is hungry
Move towards a rabbit if the rabbit is nearer than 20 spatial units	Breed if a fox is adjacent and less than 3 foxes are nearby
Move to a randomly selected adjacent position	Move towards a fox if it is nearer than 80 spatial and this fox is not hungry;
	Move to a randomly selected adjacent position

Table 3. Predator - Prey Rules (Hawick, et al., 2004)

With regards to models of emergent behavior systems, we can ascribe either the top-down or the bottom-up perspectives depending on the domain of the model, that is, whether it is representing what is observed in an existing system, or whether a system is being designed where there is some aggregate complex behavior as a goal (as is often the case of models implemented as agent-based simulations.)

At first blush, the quantitative study of systems comprised of multiple interacting entities appears intractable. For example, consider that a system composed of 100 interacting entities each with 10 possible behaviors is but a collection of 1000 behaviors, yet when they interact the possible combinations of those behaviors produces a behavior space as high as of  $10^{100}$ . Such magnitudes of computation are problematic even in light

of modern computer capabilities. At the time of this writing, Japan's K Computer<sup>3</sup> holds the record as the world's fastest super computer at  $10.51 \times 10^{15}$  floating-point operations per second (Anon., 2011). Assuming a best possible ability of one decision process per flop, the K Computer would require something on the order of  $10^{77}$  years to exhaustively compute such a decision space. Intractable indeed considering that the estimated age of the universe is about  $17 \times 10^{9}$  (17 billion) years (van den Bergh, 1995). Although we cannot explore every possible state for even reasonably small EBS, it is not uncommon at all to identify entity-based systems or to build agent-based simulations with such a population and behavior set. Amazingly, models of such systems exist all around us and achieve high degrees of success and organization.. Since we are not able to examine exhaustively the behavior space of such systems, we must consider new metrics consisting of non-exhaustive measures.

<sup>&</sup>lt;sup>3</sup> RIKEN Advanced Institute for Computational Science, http://www.aics.riken.jp/en/over/riken.html

### CHAPTER 3

# METHOD AND APPROACH

In the journal Nature, Tamas Vicsek stated, "If a concept is not well defined, it can be abused. This is particularly true of complexity, an inherently interdisciplinary concept... with no underlying, unified theory." (Vicsek, 2002) . The USMC Doctrinal Publication, MCDP 1, Warfighting (U. S. Marine Corps, 1997), discusses complexity from the perspective of military operations where war is described as a "complex phenomena" and each "belligerent" is described as a "complex system." The belligerents are complex because "...each belligerent is not a single, homogeneous will guided by a single intelligence. Instead, each belligerent is a complex system consisting of numerous individual parts." It goes on to say, "Each element is part of a larger whole and must cooperate with other elements for the accomplishment of the common goal. At the same time, each has its own mission and must adapt to its own situation."

MCDP 1 Warfighting further states that, "As a result, war is not governed by the actions or decisions of a single individual in any one place but emerges from the collective behavior of all the individual parts in the system interacting locally in response to local conditions and incomplete information."

Cosma Shalizi, in his chapter on Methods and Techniques of Complex Systems Science: An Overview in Complex Systems Science in Biomedicine by Thomas S. Deisboeck, J. Yasha Kresh. (Deisboeck & Kresh, 2006), begins by stating that a complex system "is one with many parts, whose behaviors are both highly variable and strongly dependent on the behavior of the other parts." Hubler (Hubler, 2005) looks at complexity from a physics perspective and defines complex systems as "...open systems, where the flow of a medium through the system is large. To make a system complex we increase the throughput until something unexpected occurs: a pattern emerges in the system as the system starts to oscillate." He defines a complex system as being composed of many subunits with nonlinear interactions and feedback. He also notes, "In contrast to many other physical systems, the emerging behavior of complex systems often depends on historical events." Michael Baranger (Baranger, 2001) notes that "the constituents of a complex system are interdependent." Bar-Yam observes in "About Engineering Complex Systems: Multiscale Analysis and Evolutinary Engineering (Bar-Yam, 2005), that "...the behavior of complex systems is ultimately unstable and that as a system becomes more complex the interfaces between its parts become increasingly important."

Clearly, there is great importance placed on the notion of complexity not just of a system but even that of its constituents. The various manifestations of complexity tend to produce many varied measures of this important characteristic. Depending on one's interest or perspective, complexity measures exist for computational (Steinman, et al., 2010), stochastic (Rissanen, 1989), statistical (Shalizi, 2001), structural (Kowaliw, 2008), and many other kinds of complexity. Shalizi (Shalizi, 2001) examined measures of complexity from a theoretically general perspective of complex systems science. Beginning from a statement of Occam's razor that "…entities are not to be multiplied beyond necessity", Shalizi observed that a model should be no "rougher" than necessary. As such then, complexity is that set of minimum characteristics that describe the intricacy of a system. Attempting to identify a useful measure for agent complexity is no less problematic and one should avoid the introduction of yet another measure, but the literature is rather vague at best. A literature survey reveals many approaches and

conflicting definitions of complexity with regard to agent-based, emergent, or adaptive systems. Dominique Chu, in his article "Criteria for Conceptual and Operational Notions of Complexity" (Chu, 2008), asserts that complexity as a "formal notion" should be abandoned in regards to such systems.

Is there a useful measure for complexity or something like it that can be applied to EBS, especially those that are represented by ABMs? A good starting point is to consider what we mean when we talk about agent complexity. Concerning systems represented by agent-based methods, there are at least two kinds of complexity to consider, the first is the complexity of the individual agent(s) comprising the system and the second is the complexity imparted to the system by an individual agent. Recognize that in an ABM there is the issue of the ABM itself as a model of some known or speculative system and that some entities in the system are in fact formed of multiples of agents working in concert. However, the agents comprising the ABM are in themselves models, designed to function in a suitably representative manner ascribed to some notional or real entity. With regards to ABMs (which is the approach taken here to study EBS) it is intuitive to reason that the capabilities of an entity affect its ability to interact with its environment and other entities in it. As such, complexity could be as simple as the number of states an entity can make available to the system. How should that be measured? What would that be for say a system of particles vice a system of advanced robots, or even a system of social entities such as terrorists? For the system of particles, the states might be simply: 1) bumping into a boundary, 2) bumping into another particle, 3) position and 4) velocity. Which ones are important? If all the particles are the same then it seems that their complexity should be the same even though they do not have to be in the same state at all times. Now compare the particle system to a swarm of autonomous military robots. In both cases, we are dealing with a collection of many like entities but it is easy to appreciate that the design of the latter is significantly more complicated than the representation of the former. However, if the latter are so constrained that their capabilities are so inhibited as to reduce them to particles effectively, then just with what complexity should we, as modelers of EBS, be concerned?

#### 3.1 Entity Complexity in EBS

One might begin an argument that complexity in an EBS is related to *inherent constraint*, i.e., the limits an entity enforces on itself based on "the rules of the road". The ratio of inherent constraint to an entity's possible abilities might be a useful metric, but is this what we mean by entity complexity? Another perspective that might be of interest is the number of relationships an entity can support, i.e., the number of communication pathways by which the entity can receive actionable data. For a simple particle system this would simply be the energy exchanged during collisions; for the military swarm, this could be any number of communications related information exchange characteristics e.g., packet length, data rates, information content, etc.; for a terrorist network, this could be telephone conversation events, financial transactions, etc. Complexity from this perspective might be relative to the sensorium of the entity as these define the fundamental interaction pathways. For a system of particles, this would be the result of direct contact. For ants, this might be touch and smell. For birds in a flock, it might be sight and sound. This thinking seems to suggest that an entity's complexity is in some way related to the selection of responses available to the entity based on information it

can receive. In this way of thinking about complexity the states the entity can provide to the system, compounded by the relationships the entity can sustain by its inputs and outputs might in some way define the complexity of the entity.

Another perspective of complexity might consider the "autonomy" an entity has within the system. This notion is attractive in that it seems intuitive that the more autonomous an entity is the more complex or sophisticated it is likely to be. Clearly, the particles are more autonomous than say the barriers in a particle simulation. Similarly, birds are more autonomous than particles, etc. However, this might merely be a secondary result from the complexity described in the paragraph above. What is desired is some way of combining the various notions of autonomy, communication and an entity's ability to affect its environment in a quantitative specification.

The foregoing discussion suggests that *agent complexity* is not the true measure of interest with regard to Emergent Behavior Systems, or at least is not the only measure. Here it is important to return to the lexicon of EBS and recognize that the use of an agent as a modeling metaphor or programming method is merely a means to represent the entities that comprise the EBS. Typically, our entities of interest are *actors* in the EBS and are subject to all the constraints imposed by the relationships in the EBS. The extent to which they interrelate is one kind of complexity while their individual internal workings are another kind. Therefore, we consider our definitions again and choose the term *sophistication* when we wish to describe the inner workings of an entity and the term *complexity* when we wish to speak of an entity's significance in the system with regard to the relationships formed in the EBS. These terms and their definitions make immanent sense when we consider that the word *sophistication* comes from the Greek

root *sophos* meaning wise and *complexity* comes from the Latin *complexus* meaning entwined. With this distinction in mind, we can recognize that the *sophistication* of the entities, which are the actors in an EBS, is an individual property of the actors, whilst the complexity of that *actor* is dependent on the environment of which it is a part, that is, within its *context of observation*. Now we can see how a sufficiently constrained actor may provide very limited complexity regardless of its sophistication. So with this perspective, the complexity associated with an entity in an EBS is more related to the effect or importance that entity has in the structure of which it is a part, i.e., how it is entwined with others, and not in the actual specification of the entity itself. This *entity complexity* is measured at a local level but is meaningful only in context of the global system. The entity complexity is subject to change in an EBS as the system evolves; the *entity sophistication* remains a constant. Therefore, when we speak of complexity within an EBS we are referring to the concept of entity complexity. We will use the term *sophistication* when we speak of the design of a particular entity. Stating these definitions succinctly:

*Entity Sophistication*: the measure of the intrinsic processes required by an entity necessary for it to manifest its range of functionality.

*Entity Complexity*: the measure of an entity's influence on the system arising from sustained relationships with other entities.

Appendix C: A Heuristic Measure of Entity Sophistication in Emergent Behavior Systems, presents a heuristic of entity *sophistication* derived from the characteristics used to describe agents and applied to the EBS concept of entity. This heuristic is derived from the agent taxonomy of Moya and Tolk but takes the perspective that "agents" are a descriptive architecture or implementation method, as opposed to a class of entity to be represented by some other paradigm. This distinction is reasonable when one considers that objects as inanimate as a rock or as elaborate as a human can be represented by an agent with the appropriate characteristics. However, in the EBS context a rock might indeed be an entity that is part of an EBS such as an avalanche. Given such consideration, the avalanche is the EBS comprised of rock entities that can be modeled using agent-based methods and specified with the appropriate properties. In Section 4.3 we will present a measure of entity complexity within the system that will serve as a metric of complexity of the EBS.

#### 3.2 Graph Theory Concepts and Relevance to EBS

Recalling Ablowitz's bricks, keep in mind the distinction made previously between *entities* and *agents* where "agents" is viewed as a descriptive architecture, or implementation method as opposed to a class of entity to be represented by some other paradigm. This distinction is reasonable when one considers that objects as inanimate as a rock can be represented by an agent with the appropriate characteristics. However, in the EBS context a rock might indeed be an entity that is part of an EBS such as an avalanche. Given such consideration, the avalanche is the EBS comprised of rock entities that can be modeled using agent-based methods and specified with the appropriate properties. Still, the influence of any one rock is limited because of its limited connectedness to the other rocks. In fact, its only connectedness is in the touching of adjacent rocks. This connectedness can be represented graphically by vertices representing the rocks and edges indicating whether they are touching or not. Figure 8 depicts a pile of rocks as a simple graph of vertices and edges.



Figure 8. A Pile of Rocks Depicted as a Graph

In consideration of Ablowitz's "structural relation" assertion, one can think of the information that describes the pile of rocks in terms of vertices and edges of a graph. Later we will describe some specific mathematical relationships wherein the entities comprising a system are represented by vertices and the information shared between them is represented by edges. To facilitate this understanding, here we review some fundamental graph terminology and notation.

If we define a graph as a pair G = (V, E), where V = V(G) is a finite set  $\{v_1, ..., v_n\}$ , that is, a set of vertices, and E = E(G) is a finite set  $\{e_1, ..., e_m\}$ , that is, a set of edges, we usually write vw for the edge  $\{v, w\}$ . We refer to the number of vertices n as the order of the graph and the number of edges m as its size. We can encode the graph G in terms of the order of the vertices. The label  $v_iv_j$  represents the edge from  $v_i$  to  $v_j$  and similarly  $v_jv_i$  is the edge from  $v_j$  to  $v_i$ . In the case where  $v_iv_j$  represents the same edge as  $v_jv_i$ , we say that G is an *undirected graph* and the edge is bidirectional. Otherwise, we say that the graph G is a *directed graph* or *digraph*. Although there are several distinct ways to represent a graph, here we will only consider a binary matrix A where each row and column correspond to a vertex. We call this matrix the *adjacency matrix* and it is of the form  $A = (a_{ij})$ , where  $a_{i,j} = 1$  if and only if there is an edge to itself so  $a_{ii} = 0$ . In the
case of the undirected graph, the adjacency matrix is symmetric since  $a_{ij} = a_{ji}$ . Figure 9 shows an undirected graph and its corresponding adjacency matrix.



Figure 9. An Undirected Graph and its Corresponding Adjacency Matrix

The set of neighbors of a vertex v in G can be denoted by  $N_G(v)$ . (This can be more generalized by letting  $U \subseteq V$ , that is, the subset of all vertices. We can then call N(U) the set of neighbors of a vertex.) The degree  $d_G(v) = d(v)$  of a vertex is simply the number of edges at v which is also the number of neighbors. This way of representing entity relationships is straightforward when considering a system such as a pile of rocks. Clearly the more relationships, i.e. greater d(v), of a vertex, the greater its influence in the system. In Figure 8, the centermost vertex has d(v) = 6 and because of its presence there are four other vertices with d(v) = 4. If it is removed as in Figure 10, then the maximum degree in the system is d(v) = 3. In fact, if any other vertex had been removed instead the result would be  $d(v) \ge 5$ ; so clearly that central-most vertex is of significant value to the system, i.e., that particular vertex is critical to the definition of the system.



Figure 10. Removing the Centermost Vertex

The average degree of G is given by:

$$d(G) \equiv \frac{1}{|V|} \sum_{v \in V} d(v) \tag{6}$$

The average degree quantifies globally what is measured locally by the vertex degrees, i.e., the number of edges of *G* per vertex (Diestel, 2005). In this regard, we now have a local measure, i.e., at the entity level, of a global influencer that exists not just in a Euclidean space, but also more abstractly in a relation space, i.e., at a system level. This can be considered one variable associated with the state of the system. Compare Figure 8 where d(G) = 36/10 = 3.6 with Figure 10 where d(G) = 24/9 = 2.67

The removal of the central rock suggests that perhaps that rock is more significant to the system than the others are; however, we do see from the graph of Figure 10 that there is still relationship among the remaining rocks. From an information flow perspective, we can see that there exists at least one path between any pair of vertices. This causes us to ponder whether the complexity of an entity involves not just degree in the graph but the extent to which the connectedness of the graph depends on a particular vertex. We will further develop this thought in Section 4.3 and develop a measure for entity complexity within the system as well as a measure related to information flow in the system. In advance of these developments, it is useful here to take a short digression to review some fundamental graph definitions and discuss their relevance to EBS.

**Simple Graph**: As depicted in Figure 9, a simple graph is an undirected graph with no loops, i.e., no vertex has an edge to itself, and no more than one edge between any two vertices.

**Relevance**: The simple graph represents the entity-relative configuration of a structure in an EBS. A structure, in general, is minimally represented as a simple graph. For example, in Figure 9 we could say that the entity represented by  $v_1$  is in relationship with entities represented by  $v_2$  and  $v_6$ .

**Regular Graph**: A regular graph as shown in Figure 11 is a simple graph in which every vertex has the same degree.

**Relevance**: Entities in an EBS that are described by a regular graph are assured at least two paths of information flow. No entity is more important than another is. For the simplest of regular graphs, like that shown in Figure 11, the loss of any entity, or the loss of information flow to or from any entity, fundamentally changes the structure of which it is a part.



Figure 11. Regular Graph

**Complete Graph**: Figure 12 shows a regular graph in which every pair of vertices is joined by an edge. This is called a complete graph. The number of edges |E| in a complete graph is given by:

$$|E| = \frac{|V|(|V| - 1)}{2} \tag{7}$$

This is the maximum number of edges that can exist in a simple graph.

**Relevance**: A structure in an EBS could be represented by a complete graph if every entity sustains a relationship with every other entity in the structure. Maximally connected, this kind of structure would be robust concerning information flow through the system as there would be a maximal set of information paths available.



Figure 12. Complete Graph

**Directed Graph**: A directed graph is one in which the edges exist in only one direction. **Relevance**: The directed graph represents information flow between entities in a structure. The arrows are in the direction of the relationship. Referring to Figure 13, we can read this graph in EBS terms as, "4 and 5 are in a relationship with 6," or "4 is in relationship with 6 and 2," etc. We could also say, "5 and 4 take information from 6." The convention taken in this study is that the direction of the arrow is in the direction of the relationship; information flow is then in the direction against the arrow.

With directed graphs, we consider the number of inward directed edges at a vertex as the *indegree* and the number of outward directed edges at a vertex the *outdegree*. For example,  $v_6$  of Figure 13 has an indegree of 2 and an out-degree of 1. We can also say that the degree of  $v_6$  is,  $d(v_6) = d_{in}(v_6) + d_{out}(v_6) = 3$ , which is the same degree of  $v_6$  if the graph was undirected. As such, when we speak of "the degree of a vertex" unless stated otherwise we mean the sum of the indegrees and the outdegrees.



Figure 13. Directed Graph

**Path**: A path is a sequence of vertices such that from each of the vertices there is an edge to the next vertex in the sequence. For the tree graph of Figure 15 the set of vertices  $V_p = \{1, 2, 3\}$  and the edges connecting them form a path from  $v_1$  to  $v_3$ . We usually refer to a path simply by the sequence of its vertices. In this case we could write  $P = \{v_1, v_2, v_3\}$ . The number of edges in a path is its *Path Length*. A cycle is a simple path except that beginning and ending vertices are the same.

Figure 14 depicts a special type of path where each vertex is visited exactly once. The path  $P = \{v_1, v_3, v_5, v_2, v_6, v_4\}$  is a *Simple Path*, meaning that no vertices (and consequently no edges) are repeated. This example has a path length of 5. Since all vertices are used exactly once, we call this path a *Hamiltonian Path* or *H-Path*. We can also say that the path *P* is a *spanning tree* of the graph since it contains no cycles. If the first vertex of the path and the last vertex of the path are the same, that is, if there is an edge  $v_1v_4$ , then a cycle does exist and that path is called a *Hamiltonian Cycle*.

Other paths from  $v_1$  to  $v_4$  include  $P = \{v_1, v_4\}, P = \{v_1, v_5, v_2, v_4\}$ , etc. The *characteristic path length* is the mean over all pairs of vertices of the number of edges in

the shortest path between two vertices. This is also known as the mean minimum (or average) path length.

**Relevance**: The characteristic path length suggests a measure of information flow in a structure in an EBS.



Figure 14. A Hamiltonian Path Through a Simple Regular Graph

**Tree Graph**: Referring to Figure 15, a tree is an undirected simple graph that has no cycles, but would if an edge was added.

Relevance: A tree can represent possible paths of information flow in an EBS.



Figure 15. Tree Graph

**Bipartite Graph**: A bipartite graph is one in which the vertex set can be partitioned into two sets such that every vertex in one set is adjacent only to vertices in the other set. Although different spatially, the two graphs shown in Figure 16 are equivalent relationally.

**Relevance**: Although structures in an EBS might be geometrically different, the relational equivalence suggests similar information flow characteristics. Stigmergic systems would appear to suggest representation by bipartite graphs.



Figure 16. Bipartite Graph

**Disconnected Graph**: A disconnected graph is one in which the vertex set can be partitioned into two or more sets such that every edge in one set shares no edges with the others. In Figure 17, the graphs  $V(G_0) = \{1 \ 3 \ 5\}$  and  $V(G_l) = \{2 \ 4 \ 6\}$  are disconnected graphs of  $V(G) = \{1 \ 2 \ 3 \ 4 \ 5 \ 6\}$ , also known as subgraphs.

**Relevance**: In an EBS *Context of Observation* in which the entities are represented by the graph *G*, there may exist many disconnected graphs. Usually these are the structures in the EBS. In graph theory terminology, these are called subgraphs or components. The leftmost graph in Figure 17 suggests a spatially single structure, however there are actually two distinct structures since there is no relationship between the subgraphs given by  $V(G_0)$  and  $V(G_1)$ . Although the spatial configuration of the structures defined by  $V(G_0)$  and  $V(G_1)$  may be relevant to the EBS of which they are part, there is no relationship between the two.



Figure 17. Disconnected Graph

#### 3.3 Information Theory (Entropy)

In 1948 Claude Shannon published his now famous "A Mathematical Theory of Communication" (Shannon, 1948) wherein he developed *information entropy* as a measure of uncertainty for the correct information transference between a transmitter and receiver. Information entropy becomes particularly salient in regards to the study of multi-agent systems and to emergent behavior systems in general when considered from the perspective that such systems are essentially communication systems. Recalling Ablowitz's assertion, that emergence is about the interrelationships between components in a system, in most systems of interest, those interrelationships are manifested by the communication of some information. Either, this communication can occur directly between entities, or indirectly by the entities modifying their local environment<sup>4</sup>; Shannon's Theory does not rely upon a particular means of communication aside from the conceptual model of there being a transmitter, a receiver, noise source, and a channel by which communication can occur.



Figure 18. Schematic of Shannon's General Communication System

Shannon showed that there is information – energy duality that can be understood by means of statistical mechanics, i.e., thermodynamics. Thermodynamics in essence is about the relationship between ordered and disordered energy. In physics and chemistry, disordered energy is heat. Statistical mechanics strives to derive the laws of thermodynamics using statistics. Entropy (the Second Law of Thermodynamics) is a

<sup>&</sup>lt;sup>4</sup> The later method is termed *stigmergy* and is characteristic of many natural agent systems, e.g., pheromone trails of ants, territorial markings of canines, etc.

measure of heat, or more specifically disorder in a system and observes that closed systems progress to disorder over time. Fundamentally, entropy S is defined in terms of temperature T and heat Q by:

$$\Delta S = \frac{\Delta Q}{T} \tag{8}$$

This is called the macro definition of entropy commonly used in physics and chemistry. Statistical Mechanics takes a micro view of physics and identifies the macro definition of entropy with the number of microscopically defined states  $\Omega$  accessible to a system, that is

$$S = k \ln \Omega \tag{9}$$

where k is Boltzmann's constant  $(1.4 \times 10^{-16} \text{ erg/deg})$ .

. .

The Second Law of Thermodynamics might seem counter to the observation of emergent phenomena that appear to become more organized as time progresses; in fact, we observe natural emergent behavior systems organizing with great efficiency. However, this is achieved without violation of the Second Law because of the coupling between the macro levels of the system with the disorganizing process at the micro levels. Parunak (1997) referred to this as an "entropy leak" that drains disorder away from the macro level to the micro level and observed that insect colonies leak entropy by depositing pheromones whose molecules evaporate and spread through the environment under Brownian motion.

Shannon's formulation of the Second Law considers the rate at which information is produced and by taking a statistical mechanics approach considers a set of possible events  $p_1, p_2, ..., p_n$ . The question to be asked is, "How much choice is involved in the

- 1. *H* should be continuous in the  $p_i$ .
- 2. Uncertainty should increase with equally likely events as the number of events increase. That is, if all the  $p_i$  are equal, then  $p_i = \frac{1}{n}$  and H should be a monotonically increasing function of n.
- 3. In addition, if *H* is decomposed, the result is a weighted sum of the decomposition of *H*.

Shannon showed that the only H that can satisfy all three properties is of the form

$$H(x) = -K \sum_{i=0}^{N-1} p_i \log p_i$$
 (10)

where x is a chance variable and K is a positive constant which amounts to a selection of units of measure. From the similarity of his result with the measure of entropy in statistical mechanics, Shannon refers to (10) as the entropy of the set of probabilities and which we now call *Information or Shannon Entropy*.

Such a relationship between entities, their complexity, and system measures finds merit in work such as that by Parunak and Brueckner (Parunak & Brueckner, 2001), who observed that the Shannon Entropy of a multi-agent system is a measure of coordination of agents within a system. If we consider the case of entities that can only be in one state at a time, then Shannon's entropy equation can be stated as:

$$H = -\sum_{i=0}^{N-1} p_i \log_2 p_i$$
(11)

where  $p_i$  is the probability that entity *i* is in a specific state.

3.4 Mechanisms of Feedback (Sources of Constraint)

Ashby defined *variety* in relation to a set of distinguishable elements as either 1) the number of distinct elements or 2) the logarithm to the base 2 of the number, the context indicating the sense used.

Consider the set {a,a,b,b,b,c,c,c,a,b,c}. Regardless of the quantity of elements in the set, the number of distinct elements is 3 {a,b,c} and the measure of variety can be expressed logarithmically as

$$v = \log_2 m \tag{12}$$

where *m* is the number of distinct elements in the set, or in the case of finite state automata the number of unique states possible. (Base 2 is reasonable since a system represented as finite state automata can be in one and only one state at a time.) For the case of 3 distinct elements,  $v = \log_2 3$  so v = 1.585. In essence then, it can be said that the complexity of a system is directly related to the number of unique states that describe the system.

Ashby also defines the concept of *constraint*, which can be useful in describing the relationship between two sets. Constraint relates the variety of a system's components to the variety of the system itself. This would suggest that constraint is an essential aspect of system complexity (dealt with extensively in Ashby.) His example of British Traffic Signals (a simple state machine) illustrates the concept:

	(1)	(2)	(3)	(4)	(1)	
Red:	+	+	0	0	+	•••
Amb:	0	+	0	+	0	•••
Grn:	0	0	+	0	0	•••

Table 4. States of the British Traffic Signals (Ashby, 1956)

The British Traffic Signal has 4 distinct states, red, red and amber, green, and amber. Notice that if the lights were allowed to activate independently the states would be as shown in Table 5.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
Red:	0	0	0	0	+	÷	+	+	
Amb:	0	0	+	+	0	0	+	+	
Grn:	0	+	0	+	0	+	0	+	

Table 5. Traffic Signal with Independent States

The second table presents the set of all possible states the traffic signal could attain, but the first table presents only those states allowed by the system controller. This is an example of how the controller *constrains* the variability of a system. In the first, i.e., constrained, case the variety of the system is  $v = \log_2 4$  or v = 2. In the second case  $v = \log_2 8$  or v = 3.

It can be said that the system controller *constrains* the state space. For general discussion, let's define  $V_s$  as the variety found in the constrained system and  $V_p$  the variety available to the system if it were unconstrained. If the traffic signal system were completely unconstrained, then the system variety  $V_s$  would equal that of the total possible variety  $V_p$ . If it were only allowed one state then  $V_s$  would be zero (since  $\log_2 1=0$ ). We can define the intensity of constraint *I* to be proportional to the reduction in the number of possible arrangements that the controller allows. The portion of the possible variety that is allowed by the controller can be defined as the fraction of the

system variety,

$$I = \frac{V_p - V_s}{V_s} \tag{13}$$

Therefore the intensity of constraint of the British Traffic Signal system is  $\frac{3-2}{2} = 0.5$ . If the traffic signal system were unconstrained then I = 0 and if it were allowed only one state, i.e., maximally constrained, then  $I \rightarrow \infty$ .

Another interesting aspect of this system is to observe that each light can be either on or off, thus each have exactly 2 states which gives a v = 1 for each light. Observe that the variety of the non-constrained system is simply the sum of the varieties of the components, i.e., 3, which is the total possible variety of the system. So in this case,

$$V_p = Nv \tag{14}$$

where v is the variety of each component. As is the case with the traffic signal it is assumed that each component is identical and therefore of the same variety, but in general for a system

$$V_p = \sum_{n=1}^{N} v_n \tag{15}$$

where N is simply the number of components comprising the system. This generalization implies that only the states of the components that influence the state of the system are of concern and consequently supports Ashby's assertion that, "It will be noticed that a set's variety is not an intrinsic property of the set: the observer and his powers of discrimination may have to be specified if the variety is to be well defined." (Ashby, 1956)

Ashby's observations suggest that how much can be known about a system is directly related to the level of detail, or scale, at which the system can be observed: this will become paramount in the analysis of emergent behavior systems, both from the perspective of observing such systems and specifying systems designed to exhibit emergent behavior.

If we only consider a system composed of identical components then,  $m_1 = m_2 = \cdots m_n$  and by substituting (12) into (14),

$$V_{p} = \sum_{n=1}^{N} \log_{2}(m_{n}) = \log_{2}(m_{1}) + \log_{2}(m_{2}) + \log_{2}(m_{3}) \cdots$$
  
=  $\log_{2} m^{N} = N \log_{2} m$  (16)

The states that the constrained system can take on comprise the variety of the system. In that case

$$V_{\rm S} = \log_2 M \tag{17}$$

where M is the number of unique states of the constrained system. Rearranging (13) and substituting (16) and (17) we can state the intensity of constraint I for a system as:

$$I = \frac{V_p}{V_s} - 1 = \frac{N \log_2 m}{\log_2 M} - 1$$
(18)

With regard to a system comprised of a number of finite state automata, m is the possible number of states of each automata, and M is the number of states exhibited by the system. Applied to the traffic signal example where N = 3 for the three lights that comprise the traffic signal, m = 2 since each light can be in one of two states (either on or

off), and M = 4 is the number of allowed states for the traffic light system, the intensity of constraint is  $I = \frac{3}{2} - 1 = 0.5$  as observed earlier. Equation (18) then is a general definition of intensity of constraint and the value of I is a characteristic of the system.

Constraint can originate from various sources within an EBS. As was described previously, the reduction in variety i.e., increasing constraint implies a decrease in uncertainty, which can be expressed by measures of entropy. Neither Shannon's nor Ashby's assertions require a specific source for constraint. When we talk about constraint within an EBS we are referring to any mechanisms that reduce the variety available to the entities comprising the system. Depending on the kind of EBS, we can observe that there are predominant sources of constraint in play. These predominant sources of constraint are those mechanisms that reduce the available variety through globally created actions on the locally defined entities. This is depicted at a high level of abstraction in Figure 19 showing the global or macro-scale influence on the local or meso-scale entities which themselves are defined by micro-scale properties. Correspondingly, the local actions at the meso-scale influence the global (macro-scale) system. This can be thought of as information pathways related to the primary mediums of communication amongst entities that create feedback loops between the meso- and macro- scales of the system.



Figure 19. Micro- to Meso- Scale Relationship

In natural systems exhibiting emergent phenomena such as social insects, we observe that interactions between entities occur both directly and indirectly. These include;

1) Inherent constraint (self-imposed limits),

- 2) Contextual constraint (objects in the environment reduce variety),
- 3) Entity obtrudent constraint (direct entity to entity communication),
- Stigmergic constraint (entities communicate indirectly by affecting the environment).

These modes of constraint limit the variety available to the entities and tend to determine what characteristics of the micro-level are manifested at the meso-level.

This idea of constraint within a system leading to some kind of ordered or desired system states is valuable to the goals of the EBS modeler. The sources of those constraints become fundamental to the representation of the entities comprising the system. The following sections discuss the different types of constraints and their role in an EBS.

# 3.4.1 Inherent Constraint

Foremost we observe that the very nature of an entity can restrict it in the variety from which it might otherwise avail itself. In this case, we are referring to constraints that are defined at the micro-scale, manifest at the meso-scale and are not dependent on the context of the entity. These are most often characteristics or abilities possessed by an entity that which we often think of as defining it. These built-in restrictions are referred to as *inherent constraints* as depicted in Figure 20. Examples of such inherent constraints include physical size, the field-of-view of an animal, the flight ceiling of an airplane, scan rate of a radar, etc.



Figure 20. Inherent Constraints Are Built-in Restrictions

For example, modeling a flock of birds might likely require the modeler to specify the field of view of a bird as a meso-scale property of the actor; the micro-scale components of the bird, such as position of eyes on its head, and other aspects of avian physiology that give rise to the meso-scale specification, are usually not of interest. Of course, this all depends on the context of observation and the assertion that the bird, and not its eyeball, is the actor in the EBS.

#### 3.4.2 Contextual Constraint

Contextual constraints arise from limits imposed by entities comprising the environment. Examples are physical boundaries or barriers; objects that require an actor to take some action. Contextual constraints are related to the positions occupied by the entities within the context of observation, which are often fixed both spatially and temporally.



Figure 21. Contextual Constraints Result From Objects Constituting the Environment

Particle systems are often strongly constrained by their context as illustrated in Figure 20. In such cases, objects within the context of observation form persistent structures that are not affected in general by the actors in the system. Intensity of constraint tends to remain constant over time and energy tends to be dissipative. As such, entropy is reduced as the system seeks equilibrium.

# 3.4.3 Entity Constraint

Actors may directly communicate with each other providing information that leads to constraint. This communication is usually the result of some obvious interaction and may be as simple as one actor "seeing" another. Both entities need not be of the same type. An actor might "read" information from an object (such as a book). The information transfer is from the object to the actor in this case.



Figure 22. Entity Constraints Result From Direct Interaction Between Entities

It is important to bear in mind that information, i.e., actionable data, is relayed by the actor's sensorium and effectors and so can take on many forms. The constraint may be mutual when information is exchanged such as when ants engage in antennation, or obtruded such as when a sniper kills his enemy (in which case the intensity of constraint is admittedly rather severe).

## 3.4.4 Stigmergic Constraint

Stigmergy, a term coined by French biologist Pierre-Paul Grasse, results in entity behaviors that are a consequence of the effects produced in the local environment by previous behavior (Grasse', 1959). It is the indirect transfer of information from one actor to another that results when one modifies the environment and the other responds to the new environment later. This indirect information transfer is common in biological societies such as ants and other social insects. Stigmergy provides a coordination mechanism at the meso-scale whereby entities of the system need have no knowledge of the macro-scale. In this way, such systems appear very organized, yet this organization arises without any planning or central control.



Figure 23. Stigmergic Constraint Results When Entities Communicate Indirectly

Although sometimes referred to as "cooperation without communication" (Cao, et al., 1997) in the EBS sense, actionable information is transferred so there is indeed communication. The higher information content of the Stigmergic structure intensifies constraint within the EBS. In this way, a form of macro-scale memory emerges.

# 3.5 Measuring Constraint

Examination of the constraints described here is possible using the Information Entropy of Equation (11) if we can identify

1) the states available to the system and

2) the probability of finding the system in each of those states.

In all cases in the previous section, the representative systems are specified in a discretized space. Although the actors (indicated by the circles) may move about more or less in a continuous manner, the system states can be regarded with respect to actor location on a superimposed grid. This is the approach taken by Parunak and Brueckner (Parunak & Brueckner, 2001), where they describe two ways of measuring entropy: one being location-based and the other direction-based. Both approaches are discrete and rely on defining a grid to cover the system. This is the approach used in this research and is

further described in Chapters 4 and 5.

The challenge for the EBS modeler is specifying a model to determine  $p_i$  in Equation (11). Guerin and Kunkle (Guerin & Kunkle, 2004) did this for pheromone following ants. This is a good example of meso-scale considerations. Figure 24 depicts the possible next-state transitions of Guerin and Kunkle's pheromone following ant.



Figure 24. Guerin's and Kunkle's Pheromone Following Ants is an Example of Meso-scale Inherent and Stigmergic Constraints.

In this model, the ant moves from its current position on the grid to any one of five available positions. This limit of five is the result of an inherent constraint of the ant in that it cannot move backwards. The probability of moving to a possible position  $p_j$  is given by,

$$p_{j} = \frac{\mu_{j}^{\alpha} + \beta}{\sum_{\alpha=1}^{N} \mu_{j}^{\alpha} + \beta}$$
(19)

where  $\mu^{\alpha}_{j}$  is the pheromone level at position j and  $\alpha$  is used to increase the probability that the next position will be the one with the greatest pheromone level. The  $\beta$  term is a random base, which introduces uncertainty to the system and can be thought of in physical terms as heat (or noise).

#### 3.6 A Working Taxonomy for Emergent Behavior Systems

As we have seen, Ashby's Law of Requisite Variety relates the number of control states of a system to the number of variations in control to achieve an effective response. Simply stated, requisite variety says that even if there is a sufficient number of control states designed into a system (variety), it still won't achieve its goals unless the system can execute a sufficient number of actions. Ashby's seminal work is foundational to modern cybernetics and provides a departure for establishing a means to classify systems exhibiting emergent phenomena with respect to constraining factors in the system. Perhaps the best recent works that have explored the classification of emergent behaviors are papers by Yaneer Bar-Yam (Bar-Yam, 2004), (Bar-Yam, 2004) and a paper by Jochen Fromm (Fromm, 2005). Bar-Yam generalizes Ashby's law to develop a relationship between the available behaviors of the system's subcomponents as they are defined at different scales. This concept of multi-scale variety suggests that emergent behavior is a matter of resolution; i.e., a hierarchy can exist wherein EBSs comprise EBSs. The challenge then in detecting, modeling, or designing EBSs is in determining what resolution, i.e., scale, is appropriate to the observer's, modeler's or designer's intended use.

**Type A: Emergent Behavior (Micro to macro)** Type 0: Parts in isolation without positions in the whole Type 1: Parts with positions to whole (weak emergence) Type 2: Ensemble with collective constraint (strong emergence) Type 3: System to environment relational property (strong emergence)

Type B: Dynamic emergence of new types of systems "new emergent forms"

Table 6. Bar-Yam's Types of Emergence (Bar-Yam, 2004)

Fromm presents four categories or types of emergence based on different feedback types wherein the classification of emergent behavior systems according to feedback mechanisms serves as a fundamental discriminate amongst such systems. Table 7 cites the characteristics of Fromm's categories.

**Type I Simple/Nominal Emergence without top-down feedback** Type Ia Simple Intentional Emergence Type Ib Simple Unintentional Emergence

**Type II Weak Emergence including top-down feedback** Type IIa Weak Emergence (Stable) Type IIb Weak Emergence (Instable)

**Type III Multiple Emergence with many feedbacks** Type IIIa Stripes, Spots, Bubbling Type IIIb Tunneling, Adaptive Emergence

**Type IV Strong Emergence** 

Table 7. Fromm's Four Types of Emergence (Fromm, 2005)

Neither Bar-Yam nor Fromm were considering emergence from the perspective of a systems modeler. In this dissertation, Fromm's and Bar-Yam's ideas on feedback and scale are modified by considering some of the ideas of pattern formation and complexity detection developed by Crutchfield (Crutchfield, 1994) and Shalizi (Shalizi, 2003) to form a taxonomy of emergent behavior systems helpful to the modeler. This results in:

 Dispensing with Fromm's concepts of "strong emergence" and "supervenience" as both are essentially a catchall for the unknown and rather qualitative in nature. Instead, we introduce the use of evolutionary agents at the local level where feedback from either the global or local levels can cause individual entities to add or delete from their governing rule set, and 2) The definition of five types of emergent behavior systems with four subtypes, in which the principal discriminating features are feedback types and pattern formation at the scale of observation. The taxonomy presented here builds on the thoughts of both Bar-Yam and Fromm with ideas about complex systems, with intent to model such systems in simulation.

Type 0 Constituent (Non- Emergence)
<b>Type 1 Nominal Emergence</b> Type 1a Intentional Emergence Type 1b Unintentional Emergence
<b>Type 2 Moderated Emergence</b> Type 2a Moderated Stable Type 2b Moderated Unstable
Type 3 Multiple Emergence
Type 4 Evolutionary Emergence

Table 8. The Five Types of EBS

3.6.1 Type 0: Constituent (Non-Emergence)

Type 0 emergence is in fact no emergence at all. It is the order of emergence assigned to an isolated part of a whole. In 1939, Ablowitz described how a pile of bricks is not a house until the relationship between the bricks has been defined (Ablowitz, 1939). A brick or a single gear of a watch in isolation is Type 0 emergent: not a system in itself, but rather a component, or potential component, of a system. Concerning feedback mechanisms, stated earlier as the key discriminator that is foundational to this taxonomy, Type 0 identifies a part observed in isolation without regard to its location or connectedness to the system or whole of which it is a part. Although this zero order of emergence may seem trivial at first, it is necessary in order to establish the fundamental precept that "collective behaviors are not contained in the behaviors of the parts" (Bar-Yam, 2004). Ablowitz's pile of bricks is representative of Type 0 emergence – the potential is there for emergence, but the connectivity between entities is non-existent.

#### 3.6.2 Type 1: Nominal Emergence

Type 1 Emergence is characterized by feedback that is local-to-global only. It is governed by feed-forward processes, i.e. there are no feedback mechanisms from the global-to-local involved. In Type 1 Emergence, positions, momenta, and interactions between the components comprising the system describe the behaviors of the system wholly. In Type 1 Emergence each component always has the same behavior, i.e., has a single state, and each components' state is independent of the:

- 1) other components' states,
- 2) global state of the system, and
- 3) environment.

Since the behavior of each component is independent of the other components' states, the system is completely deterministic, measurable by aggregate statistics, and therefore predictable. Causality is from the "bottom-up" as each subcomponent's behavior in aggregate forms the system behavior.

Type 1 is further subdivided into Type 1a in which the emergence is intentional, e.g., a designed machine such as a clock in which each component is specifically designed to carry out a specific function. Some examples include clocks (components being gears, springs, etc.), engines (with pistons, valves, crankshafts), and traditional software (composed of various code segments). Type 1a can be referred to as Nominal *Intentional* Emergence and typically exhibits the following characteristics:

- Typical of "ordinary machines"
- Constituent components function independently
- No top-down feedback (global to local)
- Results in Brittle Systems (Inflexible, not adaptable)
- Locally dependent (e.g., one gear turns the next).

Whether either Type 1a Emergence has varied or identical components, the behavior of the resulting system emerges from the completely specified interaction of individual components. Type 1b Emergence is distinguished by the interaction of many loosely coupled unorganized but equal components. Since the interaction is unspecified, Type 1b can be called Nominal *Unintentional* emergence. Type 1b systems are readily represented by statistical quantities. In Type 1b emergent systems, it is convenient to consider each component as a constituent "particle" that comprises the system. Examples include gases, which can be described by properties like pressure, volume, temperature, and avalanches, e.g., a line of dominoes falling or the critical slope of a sand pile (Bak, 1996). The key to identifying a system as Type 1b emergent is whether the aggregate can be described by properties that do not apply to the particles or not. For example, a single domino has width, height, etc., but the falling line of dominoes has a wave front that propagates at a rate dependent on the spacing of the dominoes. Likewise, the thermodynamic properties of a volume of gas are not properties of the individual gas molecule. Some characteristics of Nominal *Unintentional* Emergence include:

• Typical of particle systems (e.g., gasses) describing a large number of "agents" as

the sum of an average property

- Constituent components are "particles" that in aggregate are described by properties inapplicable to the constituents
- No top-down feedback, but peer-to-peer interaction is present (particles interact with each other locally, e.g., one domino knocks down its neighbor)
- Scale preserving interaction results in waves, chain-reactions, avalanches, etc.



Figure 25. Type 1-Nominal Emergence

## 3.6.3 Type 2: Moderated Emergence

Type 2 Emergence is primarily distinguished by feedback from the system to the components, i.e., top-down feedback from the global-level to the local-level. This is the form of emergence typically associated with Multi-Agent Systems, wherein the interactions of many agents react to global influences yet at the local-level give rise to patterns observable at the global-level. These resultant patterns are the emergent phenomena: they are not directly specified in the agent interaction rules but can usually be revealed via simulation. Causation is much more complex than in Type 1 emergence since Type 2 Emergence includes both direct and indirect interaction between the agents. Direct interaction amongst the agents (at the local-level) can lead to the formation of clusters that in turn can influence the behavior of the agents. Indirect interaction occurs

when agents (whether individually or by clusters) change the state of the system at the global-level that in turn affects the agents at the local-level. Flocking, schooling, or herding are examples of Type 2 emergent behavior with direct feedback as the individuals of the group attempt to stay close to each other (attraction) but not so much so that they collide (repulsion). Indirect feedback can only occur if the agents can manipulate the system at the global-level via persistent local changes. The pheromone trail following behavior of ants is an example of Type 2 emergent behavior with indirect feedback.

Type 2 Emergence, whether it is direct or indirect, can be further classified according to the net effect of the feedback. In Type 2a emergent behavior, the feedback from the global to the local is net negative and so imposes constraint, or moderating influence, on the actions of the agents and generally results in a stable global system. The bottom-up influences of the agents are regulated by the top-down feedback from the global system including the environment. In this way, Type 2a – Moderated Stable Emergent Behavior depends on complementary bottom-up, top-down processes. Flocking birds or schooling fish that rely on moving close, but not too close, and free-market economies in consumer-production balance are examples of Type 2a emergent systems. Some key characteristics of Type 2a *Moderated Stable* Emergent Behavior are:

- Typical of many social-biological systems, e.g., insects
- Stable: balance between exploration, diversity, and randomness (like the ant example)
- Top-down Feedback is net negative local affects global which tends to regulate local

• Results in stable patterns and grouping.



Figure 26. Type 2- Moderated Emergence

In Type 2b emergence, feedback from the global to the local is net positive and typically results in rapid, often exponential, changes in the system and general instability. Type 2b emergent behavior systems often exhibit runaway spirals, race conditions, or explosions. As in Type 2a, interactions between entities can cause the temporary formation of clusters at the local level, but without the stabilizing influence of net-negative feedback these clusters are not persistent therefore making Type 2b unstable. Some characteristics of Type 2b *Moderated Unstable* Emergent Behavior systems are:

- Typical of social catastrophes
- General instability
- Top-down (global) feedback is net positive and does not regulate local interactions
- Results in the rapid rise of unstable patterns or groupings and catastrophic events

## 3.6.4 Type 3: Multiple Emergence

Type 3 Emergent Behavior combines the characteristics of Type 2a and Type 2b emergent behaviors: it describes systems that exhibit both net positive and net negative feedback. Such systems can be chaotic, yet maintain long-term stability. This can be described by short-term positive and long-term negative feedback. Foraging ants that rely on the replenishment and evaporation of pheromone trails and similar reaction-diffusion systems (also called activator-inhibitor systems) typify Type 3 Emergence. Some characteristics of Type 3 *Multiple* Emergence include:

- Typical of "reaction-diffusion" systems
- Adaptive, turbulent, chaotic, exhibiting both stable and unstable characteristics
- Short-term positive feedback checked by long-term negative feedback
- Results in biological-like pattern formation, e.g., "Game of Life" behaviors.



Figure 27. Type 3 – Multiple Emergence

## 3.6.5 Type 4: Evolutionary Emergence

This last class of emergence introduces a fundamental difference in the underlying assumptions of the entities comprising the system. Emergent behavior systems of Types 1 through 3 are premised on constituent entities that are governed by an internal set of rules that are immutable. Although the entities need not be identical, their individual properties essentially do not change: the values of those properties change, but there is no addition or deletion of the number of properties each entity possesses. The entities comprising Types 0 through 3 can be described by a finite set of state changes. Type 4 Evolutionary Emergent Behavior is distinguished by changes in the governing rule sets within the local entities, i.e. from a state space perspective, each entity may, by various means, add to or delete from its governing rule set. Very dynamic indeed, this fourth order emergence allows individual entities to change their very definition in response to either local or global influences. Genetic algorithms are a form of Type 4 emergence. Many aspects of biological systems would fall into this category. Some characteristics of Type 4 *Evolutionary* Emergent Behavior systems include:

- Typical of biological entities
- Entities can learn from experience
- Adaptive at both global and local levels
- Feedback can be of any type and multiple in sources
- Results in highly complex adaptive systems.



Figure 28. Type 4 – Evolutionary Emergence

3.7 The Role of Constraint in the EBS Taxonomy

We now see that there is a relationship between the variety available to a system and the paths of information. This is expressed in terms of the Type 1a (T1a) and Type 1b (T1b) classes respectively: a system of a single state can be thought of as infinitely constrained, as is the T1a EBS and an unconstrained system exhibits the characteristics of the T1b EBS.

In Type 0 emergent behavior, there is no emergence at all. Since the system transitions to no state, the system exhibits exactly one state and  $V_s = 0$  in Equation (18). Likewise there is no change of state of the components so the variety of each component v is also 0. Intensity of constraint for such a system is undefined.

In a T1a emergent behavior system, the interactions between entities is strongly directed; relationships between entities comprising the system are strictly ordered. Consider two gears, one with ten teeth and the other with twenty. The interaction of these two components can be arranged in only one way and the only achievable gear ratio of the system is 1:2. To have a functioning system each component can have only one of two states. The two states are engaged or not engaged. Variety for the components from (1) is therefore 1. The unconstrained variety of the system comprised of the two gears is consequently  $V_p = 2$ . For the system to work it can only have one state so  $V_s = 0$ . So for a T1a emergent system we see that  $I \rightarrow \infty$ , which is the ultimate constraint. This is to be expected given that this geared system can only be defined for what it is when constructed in one way.

T1b emergent behavior is distinguished by interaction between like-entities where there is no strict ordering. There is no feedback from the global to the local, i.e., no outside influence affects the T1b system. Variety for such a system is simply the sum of the v's of the components so  $V_s = V_p$ , therefore there is no intensity of constraint, i.e., I = 0. The Type 2 (T2) emergent behavior system includes feedback from the global to the local level. The system constituents are affected not only by interactions amongst themselves but also by influences from outside the system that tend to moderate the system. In T2 emergence, the global to local feedback can have a net negative (T2a) or net positive (T2b) influence. T2a moderation tends to stabilize the system. T2b tends to drive the system toward instability. Intensity of constraint in such a system varies as the system affects the entities comprising the system and the entities affect the system. For the case of a system composed of finite state automata (the assumption is that the components state space is static), this makes I a function of time as the states of the system vary in time. So (18) becomes,

$$I(t) = \frac{V_p}{V_s(t)} - 1 = \frac{N \log_2 m}{\log_2 M(t)} - 1$$
(20)

and

$$\frac{d}{d(t)}I(t) = \frac{d}{d(t)} \left(\frac{V_p}{V_s(t)} - 1\right)$$
$$= \frac{d}{d(t)} \left(\frac{N\log_2 m}{\log_2 M(t)} - 1\right)$$
$$\frac{d}{d(t)}I(t) = \frac{N\log(m)M'(t)}{M(t)\log^2(M(t))}$$
(21)

If I is increasing then the system is tending toward T2a, if decreasing then the system is moving toward T2b. Type 3 emergent behavior would indicate a damped oscillation, at times increasing I and at times decreasing I, with an overall tendency to increase I. In (21) this could only be caused by a decreasing number of states available to

the system.

The problem with (20) and (21) is that it is assumed that the only influence on intensity of constraint is the number of states of the system. However, *I* could have just as easily been increased by increasing  $V_p$ : for simplicity it has been assumed that  $V_p$  is constant. However we must consider under what conditions  $V_p$  can change; if the components of the system are identical finite state automata then *m* does not change (at least not in T0 – T3). However, as the components interact in response to influences of others and their environment they will at times combine to form structures within the system, i.e., new *meta-components* with influences affecting the system being some combination of the states of the components. This is an interesting observation that is beyond the scope of the current study, but perhaps worthy of further investigation.

## 3.8 Defining Emergent Behavior Systems

Recalling Section 2.3, can we simply apply Ferber's definition of agent to all entities in an EBS? At first blush, one might answer "Yes", but the more thoughtful answer would have to be "No". Although Ferber's definitions of agent and MAS go far in describing EBS, these definitions exclude many forms of EBS unless Ferber's definitions are modified for each case. Consider that Ablowitz's brick can be made to fit the agent definition if we were to dial down certain values of Ferber's characteristics. However, a brick is very different from a sophisticated cognitive agent. To reconcile this point we must depart from the discussion on agents and return to what we mean by an EBS; then with some understanding of what might be expediently addressed by agent methods within the EBS context we can move toward a more formal definition of EBS useful to the modeler.
Although Ablowitz's definition of emergence suggests a non-linear quality stemming from entity interactions, continued research for the subsequent six decades has led to additional insight to the nature of emergent phenomena. Holland (Holland, 1998) points out that a key characteristic of emergent systems is the persistence of patterns even though the components change. Bonabeau et al (Bonabeau, et al., 1995) observed that "there is no real agreement on what it should imply for a phenomenon to be emergent." They examined emergence from the perspective of Artificial Life and Cognitive Science, presenting examples from modern biology, chemistry, economics, and physics. Their examples identified additional characteristics of emergent systems such as the existence of levels whereby lower level components give rise to new characteristics at a higher level, global coherence arising from local rules, and the necessity of an observer by which emergence can be recognized. The upshot of the modern understanding of emergence is that it takes on different forms depending on the domain of study, but can indeed arise from very simple entities. Bonabeau et al reasoned the need for a framework for characterizing emergence; a result in keeping with Holland's assertion that, "The uncertainty of definition forces us to rely on partial descriptions, which in turn rely heavily on context."

In a subsequent paper, Bonabeau et al present the concept of emergence through a framework built on levels of organization, levels of detection, and theories of complexity. The authors argue that these reveal some characteristics of emergent phenomena and assert that any recognition of a phenomena to be emergent is in itself subject to the existence of an observer (Bonabeau, et al., 1995). They include the local perception of actors and their ability to act locally, initial states of organization, evolution over time, etc and note, "...the emergent aspect of a phenomena is related to the point of view of an observer of this phenomena: it is not intrinsic to the phenomena, but related to the global system (phenomenon + observer)." This insight has presents an important consideration for the EBS modeler; specifically, what is (or what should be) the point of view of the observer, i.e., the necessary perspective and level of resolution required in the model? For the modeler, we must ask, can we formalize the definition of an EBS that is consistent with the fundamentals discussed earlier and building on more recent progress in MAS? To do so will require us to further decompose the concept of an EBS and revisit our context-free definition. However, in doing so, we will arrive at a modeling formalism for EBS that is useful for identifying the critical elements of an EBS model and a basis for communicating EBS model characteristics.

First, to call something a system it must be distinguishable apart from other systems, i.e. it must be bounded or rather it must exist within some context. In general, we are quite comfortable with the notion of observing systems spatially within a volume. Typically, we call such a context the *environment*. If we were to ask a child (the observer) to identify the parts of a scene of fish schooling in the water, he might identify a school of fish that seem to stay together as they move in the water. The water (or at least the child's perception of it) is the environment in which the school exists. The fish do not really do fishy things if taken out of the water, so they only have meaning in the water, i.e., in their environment. They certainly cannot school on the sidewalk! The child will also observe that there might be octopi and seahorses in the water, as well as jellyfish and seaweed. The child will tell you they are not the same as the school of fish, but the fish must swim around them. Those entities are not the fish, but they also occupy the environment. Therefore, there is the school of fish, and their environment composed of water along with the other things in the water. Although the fish swim in the water and respond to it, they are unable to affect substantially the water. However, they can bump into, avoid, communicate with, etc., the other animals in the water. Nevertheless, from their perspective, and recognizably from the perspective of the child-observer, these are just *entities* in the *environment*. Therefore, it seems that the concept of an EBS is meaningful only within the context of the environment perceived by the observer. We must conclude that a formal definition of an EBS to support modeling must include some definition of *environment*.

Second, by definition, a system is composed of discrete units that we can call entities. Entities can occupy the environment, but do they have characteristics that can distinguish them apart from the environment? Asked simply, what is an entity? Referring again to Ferber, an agent is obviously an entity, but should all entities be treated the same or referred to the same way in specification? Clearly, the fish in the previous example are agents in the Ferber sense, and that can be easily extended to the octopi and seahorses, but what about the seaweed or a piece of flotsam? Surely, neither would qualify as an agent. Still, the flotsam has characteristics that govern how it behaves in the water, and the fish can influence it by pushing it, eating it, etc. The question to be considered then is not so much how something is represented, but rather what role does it play?

If we assume that entities are the atomic components of an EBS that are observable at the scale of observation and that entities through some means can interact with other entities in the environment, then how they interact and those means of interaction become crucial to the EBS modeler. Entities must be observable in the environment. Entities can influence each other. Consider if one of the fish dies; then that fish is now flotsam. It is still an entity within the system, but from the modeler's perspective, it now belongs more to the environment than to the set of agents in the school. Additionally, the fish, by virtue of maintaining their positions relative to and in concert with each other, form the school. That school is a system within the system, i.e., a structure with aggregate characteristics not readily deducible from the parts.

Finally, recall that we have made the argument that the term "agent" refers more to a method of implementation than to a specific role in an EBS. To emphasize this distinction, we adopt the term actor for a fish in the school. For the modeler, objects apart from those that are the primary actors in the system would seem to define the environment of an EBS. It is the difference between a fish in a school and a dead fish; an entity that was once an agent has become an object. The flotsam, whether it is a dead fish or something else, is still contained in the environment and if it were to fall to the ocean floor and become encrusted with barnacles it might then be considered part of, that is an object making up, the environment. With this trite example, we conclude that within any context of observation an entity may be an actor, or object, within the environment. Additionally we observe, that although these entities can form structures that persist, as time progresses the roles of entities can and do often change. Furthermore, Ferber's definition of "agent" no longer confuses the discussion since we now recognize "agent" as an implementation means of EBS entities. The notions of actor, object, and environment define the modeler's context of observation. Figure 29 depicts the context of observation and dynamics of the relationships between entities and the environment of

which they are a part.

Context of Observation



Figure 29. The Context of Observation

The modeler cannot discount the context of observation and the roles the entities play within it. All entities comprising the system are observable at the scale of observation. Objects do not use information but can react to actions (you bump into a wall, and the wall makes a noise; you push a ball and it does not argue with you). Objects can be a source of information (like a book). Objects are entities and actors are entities – what distinguishes objects from actors is the form of interaction they can participate in, i.e., the role they can play. This means that actors can at times act as objects for other actors, depending on the state they have assumed. The distinction arises from the perceived relationships and the formation of structure at the time of observation. Consequentially, these entities, i.e., actors and objects, exist within the environment and are at various times considered part of it. Now we have at least a conceptual idea of what roles constitute an EBS for modeling purposes; we readily see that there is an observer, an environment, actors, and objects. We also observe that these roles change in time, i.e.,

there is an evolution of the system. Table 9 summarizes the characteristics of the entities comprising an EBS.



Table 9. The Characteristics of the Different Types of Entities in an EBS

Ferber presented a formal specification identifying as separate kinds of entities the agents, objects, and environment comprising the general MAS. Here we suggest a similar specification for the purposes of modeling EBS. Compared to MAS, the EBS formalism distinguishes the distinct functions of actors, objects, and structures, which nevertheless are entities of the system. Therefore we state a basic specification for an EBS as

$$EBS \equiv \langle C, \{E\}, \{R\}, \{\Psi\}, T \rangle$$
where
$$(22)$$

$$C(t) = \{e: P_e(t)\}$$
(23)

is the spatial and temporal context of observation defining the environment that contains all entities, i.e., where  $P_e(t)$  is the position of an entity e at time of observation t

*E* is the collection of all entities that are *actors A* and *objects O* or *structures S* within the context of observation such that,

$$E = \{e : e \in (A \cup O \cup S) \cap C\}$$
where
(24)

O is the subset of entities called *objects* where  $\forall o \notin A : O \in E$ 

A is the subset of entities called *actors* where  $\forall a \notin O : A \in E$ 

S is the set of structures formed by entities in relationship such that

$$S \equiv \{e: e_i \mapsto E \times R\}$$
(25)

These structures can take on the roles of actors or objects within the context of observation.

R is the set of *relationships* forming the links between entities such that

$$R = \{r : r_{i,j} \mapsto (B_i \times B_j)\}$$
(26)

where  $B_i$  and  $B_j$  are the *behavior sets* governing the  $i_{th}$  and  $j_{th}$  entities forming a relationship such that B is the collection of entity *behaviors* that can potentially affect other entities within the context of observation such that

$$B = \{ \forall e_i : e_i \in E, \exists \{b_i\} : \{b_i\} \subseteq B \}$$

$$(27)$$

 $\Psi$  is the set of *states attainable by the system* which emerges from the relationships amongst the entities such that,

$$\Psi \equiv \{\psi \Longrightarrow (E \times R)\}$$
(28)

and T is the time base.

This attempt at a formal definition produces a declarative model that views the EBS as a sequence of changes in state. It also serves to define the components of an EBS and their relative roles formally. In light of this formalism, we restate the definition of some key terms relative to EBS in Table 10.

EBS Term	Definition		
Entities	the collection of objects and actors comprising an EBS.		
Context of Observation	that portion of a system perceived by an observer that defines the environment of interest.		
Environment	the spatial and temporal context of observation defined by objects and containing all actors		
Actor	an entity comprising an autonomous stochastic dynamical system that attempts to build and maintain a maximally predictive internal model of its local environment within the context of its sensorium, behavior sets, and effectors.		
Object	an entity contributing to the environment of an EBS that is not an actor within the context of observation.		
Structure	A group of entities within the context of observation, which are in relationship and persist for some time.		
Relationship	an information flow between entities forming interactions governed by the behaviors of the entities		
Behavior	the set of rules inherent to an entity that govern its relationships to other entities		
Information	any data that elicits a change of state in an entity.		

Table 10. EBS Specification Terms

.

#### CHAPTER 4

## ANALYSIS FRAMEWORK

The previous three chapters provided some background and rationalizations to:

- 1) describe the concept of emergence in systems of interacting entities,
- 2) understand some of the factors that affect emergent behavior systems, and
- 3) express a system model with the intent of examining emergence

phenomena through simulation.

In the remaining chapters, we will apply the ideas of the first three chapters to the development and implementation of three simulations of emergent behavior systems. We will develop and apply metrics to these three simulations to explore the concepts described earlier.

Recall that this research posits that the emergence phenomena in simulations of EBS can be described by system properties related to 1) the complexity associated with the role of the entities comprising a system, 2) the information (or energy) flow between the entities, and 3) the information uncertainty associated with the system. We set forth to examine these properties by relating the entities in a simulation to a graph representation and defining metrics for principal information flow, complexity of entities within the system, and organization uncertainty in the form of information entropy.

#### 4.1 Framing the Problem and Defining Metrics

Ashby's Law of Requisite Variety as discussed in Chapter 3, can be interpreted to state that a complex system can only be effectively regulated if the variety of the regulator is equal to or greater than the possible variety of the system being regulated. This regulation of variety, i.e., constraint is manifest as phenomena that tends to reduce the variety at the meso-scale structure and are related to the primary mediums of communication amongst the entities in the system. In natural systems exhibiting emergent phenomena such as social insects, we often observe that interactions between entities occur both directly and indirectly. These interactions are regulated by the following constraining modes:

- 1) Inherent Constraint (self-imposed limits),
- 2) Contextual Constraint (objects in the environment reduce variety),
- 3) Peer Constraint (obtrudent, i.e., direct entity to entity communication), and
- 4) Stigmergic Constraint (entities communicate indirectly by affecting the environment).

The dissertation touches on these different constraint mechanisms both in regards to the taxonomic delineation of EBS types and the principal paths of information flow in the system. The metrics described here examine the three systems where different constraints dominate, namely inherent, peer, and Stigmergic constraints. Foundational to the modeling of EBS, is the distinctions between data, information, and communication. In this dissertation, we have defined information as any actionable data, i.e., data that causes an entity to change state, within the context of observation. Communication is the sharing of information; relationships, as evidenced by the formation of structures, are established when communication occurs and are sustained.

We have seen that graph theory methods provide a means to measure the ability of data to flow in a structure. In particular, representing the EBS in graph terms supports the examination of information flows and feedback, i.e., paths that form loops. The flow metric is related to the sequence of vertices that can represent the principal actors in a structure and is determined by the characteristic path length, i.e., the mean over all pairs of vertices of the number of edges in the shortest path between two vertices.

The complexity metric deals with the relative importance an actor maintains in the emergent structures in the system. Again, graph theoretic measures readily apply. Stated another way, the entity complexity metric represents a measure of the significance of an entity to the structure of which it is a part and can be thought of as a measure of "relatedness" of the structure. This metric is developed from the local clustering coefficient  $c(v_i)$  for a vertex  $v_i$  and is given by the proportion of links between the vertices within its neighborhood and the number of links that could possibly exist between them.

Shannon Entropy is a measure of uncertainty or disorganization of the system. In its essence, it provides a measure of system disorder as it relates to the entities comprising the system. In the course of this dissertation study, two approaches to the estimation of Shannon Entropy were implemented. The first dealt with representations of the system state in terms of the systems spatial geometry. This approach is in agreement with that described by Parunak and Brueckner (Parunak & Brueckner, 2001) and is well suited to systems where system state is easily defined as some spatial relationship between actors. A more general approach developed here is in keeping with the emphasis on relationship within the EBS. As opposed to a Euclidean-space perspective, this approach depends on the relation-space of the system. That is, the state vector defining the system at any time is a measure of the number of structures in the system and normalized mean degree of the structures in the system. The three systems simulated in this study lend themselves to spatial state representations. We will focus on the spatial state form of the entropy metric in explaining its application, however in the analysis of the systems we consider the relation-space concept as well and discuss it in Chapter 6. Figure 30 summarizes pictorially the three metrics and their relationship through graph methods. The following sections detail the mathematical implementation of these concepts into EBS metrics



Figure 30. EBS Metrics

### 4.2 The Information Flow Metric - Characteristic Path Length

As it has been established that structure formation is dependent on relationship and that relationship is sustained information flow, one way to characterize an EBS is by some measure of the information flow in the system. Recalling Chapter 3, we see that a strong case is made for the representation of relationship by graph-theoretic methods. We saw that there is a distinction between directed and undirected representations. At this point, a concrete example will aid in the rationalization of the choice of a form of graph path length. Although numerous domain examples can be found in the literature, we will now speak in terms more closely related to the simulations in the next chapter with the intent to facilitate clarity of the concept behind the flow metric.

In the Particle System Simulation, particles can interact with each other either directly through collision, or through the formation of spring-like connections. If the spring connections are able to persist, these particles form structures that in turn can interact with other entities in the environment. The spring relationship is based on Hooke's Law and the spring constant is given as k. The higher the value of k, the stronger the spring is. (A damping factor represents a contextual constraint that is somewhat representative of the viscosity of a fluid.) They can also interact through collision with the boundaries of their space or with objects that can be placed in the context of observation. A plot of the internal energy is an indicator of communication events as it identifies whenever any particle contacts another object. The exponential decay arises from the dampening factor. Each communication event is detectable by an abrupt change in this energy as shown in Figure 31 where the left image shows a structure falling to the floor and the right shows communication events indicated by energy changes. Here there are approximately 5 communication events, one for each time a particle encounters the floor as the structure bounces. This data is communicated throughout the system since it is fully connected with a k = 50 and rest-length of 4. The communication events are propagated through the structure in a time depending on the value of k and other constraints in the system. The pushing and pulling of the springs between the particles form communication events (information flow). In the absence of any dampening this

would be seen as an oscillation of the structure. There is information outflow from the floor to the colliding particles and this is propagated to the other particles through their relationship to each other.



Figure 31. Communication Events Indicated by Energy Changes

This is further illustrated in Figure 32 where we can see the inflow and outflow of information (or energy.)



Figure 32. Inflow and Outflow

It is easy to see that a communication event can create a change of state in an entity. What is interesting is when a communication event is propagated through a structure as depicted in Figure 33. Here the characteristic of interest is the length of path, i.e., the route by which a number of entities can share information and the interaction of inflow and outflow. The principle information flow is given by the path available to the information at the time of the communication event. In Figure 33, entity e3 strikes entity e4 initiating a communication event at time t0. At time  $t_0$  e3 and e4 are in relationship with communication event C0. However, the path available to C0 is e3, e1, e2, i.e., three entities. Note that to have meaning, a communication event must occur between two or more entities since there must be inflow and outflow. If there were only a single entity in a system then there is nothing for it to communicate to so no path exists, no state change can occur.



Figure 33. Persistent and Non-Persistent Information Flows

In the EBS, entities form sustained relationships manifesting as structures within the context of observation. Information circulates as the structure responds to various information from the macro to meso-scales. These flows of information through the actors comprising the structures form networks that vary in time and precipitate the evolution of the structures.

Erdos and Renyi are recognized as having laid the foundation for the study of random graphs (Erdos & Renyi, 1959), (Erdos & Renyi, 1960). A random graph is one in which given N vertices, any two vertices have a finite probability p of being joined by an edge. The random graph then has an average degree  $\approx pN$ . Newman, Strogatz, and Watts observed that random graphs have a long history of representing coupled dynamical systems such as social networks, epidemiology, food webs, etc., and can offer insight into patterns characteristic of the social structures formed by self-organizing systems (Newman, et al., 2001), (Watts & Strogatz, 1998). However, they point out that measurements derived from network models based on random graphs differ significantly from the graphs derived from real data; an observation that suggests that additional social structure exists that is not captured by the random graph. Whereas truly random graphs produce degree distributions that are Poisson distributed, Newman et al observed that many real-world networks exhibited degree distributions that are power-law distributed (Newman, et al., 2002). Newman et al showed that for many large, sparse networks found in nature, the number of edges in the shortest path between two vertices averaged over all pairs of vertices is a metric with intuitive meaning to these dynamical systems. For instance, the average of the shortest path lengths of a network of friends is the average number of people connecting two friends. (We will discuss Newman et al again in the following presentation of the clustering coefficient.) Newman et al observed, "the average distance, which is a global property, can be calculated from a knowledge only of the average numbers of first- and second-nearest neighbors, which are local properties. It

would be possible therefore to measure these numbers empirically by purely local measurements on a graph such as an acquaintance network and from them to determine the expected average distance between vertices" (Newman, et al., 2001).

.

Their observations suggest that the average of minimum path length, which is also called the characteristic path length (CPL), is a good measure of information flow in a system, yet still a measure based on local properties; a concept attractive to the nature of EBS. One idea explored in this dissertation is that the relationships established as an EBS evolves can be represented by graphs and that these graphs can characterize emergence from non-emergence. As such, the CPL of the graph representing the EBS should be distinguishable from a similar random graph, i.e., a random graph of the Erdos and Renyi type with the same average vertex degree. This supports the formulation of an information flow metric based on the CPL for graphs representing an EBS. Figure 34 shows some example graphs and the result of applying the flow metric of characteristic path length developed for this study. Section 5.3 presents the MATLAB design for this measure and Appendix B: MATLAB Code & Scripts, presents the actual MATLAB code.

In summary, the flow metric is related to the sequence of vertices that can represent the principal actors in a structure and is determined by the CPL, i.e., the mean over all pairs of vertices of the number of edges in the shortest path between two vertices.



Figure 34. Some Structures Shown as Graphs and Their Characteristic Path Length

## 4.3 The Complexity Metric - Clustering Coefficient

Here we again observe that graph representations of EBS are not completely random as relationships are subject to the constraints at play and neither are they completely regular since at any given set of observations the graph can be seen to change. (However, they may become regular subject to constraints as well.) Watts and Strogatz introduced the measure of the *clustering coefficient* as a measure of connectedness in networks that "rewire" and exhibit varying amounts of disorder (Watts & Strogatz, 1998).

Referring to the examples of Figure 35, the local clustering coefficient  $c(v_i)$  for a vertex  $v_i$  is given by the proportion of links between the vertices within its neighborhood and the number of links that could possibly exist between them. For a directed graph,  $e_{ij}$  is distinct from  $e_{ji}$ , and therefore for each neighborhood  $N_i$  there are  $k_i(k_i - 1)$  links that

could exist among the vertices within the neighborhood ( $k_i$  is the total (in + out) degree of the vertex). Thus, the local clustering coefficient for directed graphs is given as

$$c(v_{i}) = \frac{\left|\{e_{jk}\}\right|}{k_{i}(k_{i}-1)} : v_{j}, v_{k} \in N_{i}, e_{jk} \in E$$
(29)

An undirected graph has the property that  $e_{ij}$  and  $e_{ji}$  are considered identical. Therefore, if a vertex  $v_i$  has  $k_i$  neighbors, edges could exist among the vertices within the neighborhood. We can then consider the idea of a *clustering coefficient* of a particular vertex v defined for the neighborhood of v denoted as c(v) (Newman, et al., 2001). We define c(v) as the fraction of edges that actually exist in the neighborhood of v relative to the total possible number of edges that can exist in the neighborhood of v. If  $k_i$  is the number of neighboring vertices of  $v_i$ , then, if the graph is assumed to be undirected, the maximum number of edges that can exist in the neighborhood of v is  $\frac{k_i(k_i-1)}{2}$  and if the number of existing edges in the neighborhood of v is given by  $|\{e_v\}|$  then we can define the local clustering coefficient of the vertex  $v_i$  as

$$c(v_{i}) = \begin{pmatrix} \frac{2|\{e_{jk}\}|}{k_{i}(k_{i}-1)} & ,k_{i} \ge 2 \\ & & :v_{j}, v_{k} \in N_{i}, e_{jk} \in E \\ 0 & ,k_{i} = 1 \end{cases}$$
(30)

The significance of c(v) is that it indicates the extent to which the vertices in the neighborhood of v are also neighbors to each other. The smaller c(v) is, the more dependent those in the neighborhood of v are on the existence of v. If c(v) is 1, then the neighbors of v are still connected even if v is removed. If c(v) is 0 then all the neighbors of v depend on it as illustrated by  $v_0$  in the third graph of Figure 35.

One implication of the clustering coefficient to EBS is that if neighbors of a single vertex are neighbors to each other as well, then they do not wholly depend on that vertex for their relationship. Stated another way, if the clustering coefficient associated with an entity in an EBS is high, then its neighbors do not rely on it very much. The corresponding entity in an EBS does not contribute significantly to the structure of which it is a part. If the clustering coefficient is low, that entity plays a more significant role in the structure of which it is a part and as such, its presence in the structure is more important to the existence of the structure. In studies of scale-free networks, i.e., networks in which the degree distributions exhibit a power-law distribution, Barabasi and Albert observed that the clustering coefficient of scale-free networks were about five times higher than that of random graphs(Barabasi & Albert, 2002).

Consider the three simple graphs shown in Figure 35, all of which have the same number of vertices but different edge sets. G1 is a complete graph and can be thought of as representative of an EBS in which each entity has a direct relationship with every other entity. G2 is incomplete with several subgraphs, and G3 is a tree, representative of an EBS where the structure, i.e. all relationships, is dependent on the single entity  $v_0$ .

We can further define a global clustering coefficient C(G) of a graph G as the mean of the local clustering coefficients in G.

$$C(G) = \frac{1}{|V|} \sum_{i=0}^{|V|-1} c(v_i)$$
(31)

where |V| is the number of vertices in the graph with  $d(v) \ge 2$ .



Figure 35. Graphs for Clustering Coefficient

Another measure of complexity associated with a vertex is its degree. Indeed, both degree and clustering coefficient are both local graph measures of a vertex's connectedness. Earlier in the development of this dissertation, the normalized mean degree was identified as a complexity measure and is included in the definition of the state vector based on the relation-space perspective. However, the clustering coefficient factors in not only the connectedness of a vertex but also the importance of a vertex to those to which it is connected. The global clustering coefficient then provides insight into both the connectedness and the strength of dependencies of entities in the system; this is a measure not only of the information paths associated with an entity, but of how reliant the emergent structure is on that entity. Chapter 5 presents the MATLAB algorithm to compute the clustering coefficient and Appendix C lists the MATLAB code.

#### 4.4 The Organization Metric – Shannon Entropy

Recall that in the previous chapter, we discussed how Claude Shannon showed that the only H that can satisfy all three properties of the macro definition of entropy with the number of microscopically defined states is a function of the form

$$H(x) = -K \sum_{i=0}^{N-1} p_i \log p_i$$
(32)

Where x is a chance variable and K is a positive constant that amounts to a selection of units of measure. From the similarity of his result with the measure of entropy in statistical mechanics, Shannon referred to this as the entropy of the set of probabilities, which we now call information entropy. An information source provides large entropy (|H(x)| large) if its contents are of random nature; entropy is small if the source contains regular structures. This phenomenon can be observed in time varying systems of interacting entities when some spatial patterns appear more frequently than others do. If we consider this formation of structure with regard to Ashby's Law of Requisite Variety (Ashby, 1956), a cause of decreasing disorder is increasing constraint; in a system, this reduction of the uncertainty is indicative of increasing structure. Therefore, we should expect to see decreasing entropy as an EBS evolves to an orderly state. The challenge to the modeler is to determine what defines the state of a system. For the three systems explored here, we consider the state of the system described by the positions of the entities in Euclidean space. In the particle and stigmergic EBS, the locations of the emergent structures are taken to be the state of the system. For the flocking system, we consider the direction of the flock. Both of these measures are based on the work of Parunak and Brueckner (Parunak & Brueckner, 2001).

Computationally implementing (32) to analyze emergent behavior systems requires observations of the states accessible to the system and the determination of the probability of finding the system in each of those states. This can be accomplished parametrically with a simulation of a system and using the Monte Carlo method to build an estimate of H(x) by counting the occurrence of each observed state. Since many EBS manifest as spatial-temporal patterns, the states of the EBS can be expressed in terms of the locations of the entities comprising the system. This location-based approach and the sampling in time to produce the necessary observations suggest an approach that is discrete in both space and time. The following steps explain the location-based approach. Step 1: Gridding

Parunak and Brueckner developed a gridding method to specify system states in multi-agent systems. This method is adopted here with slight modification for the computation of (32). Figure 36 depicts the method. Here an  $m \ge m$  context of observation (8 x 8 in Figure 36) is gridded into  $n \ge n$  cells (here 4 x 4) to produce a state vector of the form

Each c in the vector is the count of entities in that cell at time t. For the 4 x 4 grid shown, each state  $\Psi$  is given by a vector of 16 elements. Each element is the entity count in that cell.



Figure 36. Spatial Gridding Method for System State Representation

### Step 2: Determine Probability of States

Multiple simulation runs beginning with random initial conditions can be used to generate the set of  $p_i$  needed for (3). At each time-step of the simulation, the state vector  $\Psi$  is produced. The count of each  $\Psi$  across the simulation runs at each time-step can be used to estimate the probability  $p_i$  at each time-step t in the manner,

$$p_{i}(t) \approx \frac{count\_of\_\Psi_{i}(t)}{R}$$
(34)

where R is the number of simulation runs.

#### Step 3: Compute Entropy

Once the states are identified and counted for each time step, (32) can be computed. In computing (32), recall that K is a constant. For this analysis, K is assumed to be 1. The form of (32) as used in this analysis is

$$S(t) = \frac{-\sum_{i=0}^{R-1} p_i(t) \log p_i(t)}{\log R}$$
(35)

Equation (35) is the Shannon Entropy estimate at time *t* for the system but normalized by dividing by  $\log R$  so that the case S = 1 indicates maximum disorder. <u>Application of the Method</u>

Applying equation (35) to calculate the entropy of a system gives rise to two questions:

- 1) Should the extents of the grid be the extents of the context of observation?
- 2) What size grid should be used?

The first question is more a matter of choice based on the goals of the analysis. The research here is concerned with the meso-scale entropy; therefore, the gridding is limited to the extents of the spatial positions of the particles, i.e. the structure formed by the entities in relationship. Figure 37a shows the initial configuration for a sample system and Figure 37b shows the same system after one-thousand time-steps. In both figures, a box has been drawn to indicate the meso-scale area. The meso-scale system state plots for the first state and last state are shown at grid sizes of 3, 4, 5, 6, 7, 8, 9, and 10 in Table 11. Grid sizes beyond 8 do not add significant resolution as can be seen on the plot of the count of unique states versus grid size n shown in Figure 38. This data suggests that grid sizes above about six or seven do not provide additional resolution to the measure of the Entropy for this system. At finer grid sizes, the grid covers the structure at about the same scale as the context of observation. The knee in this plot at 4 and the essentially linear slope to 8 suggests that a gridding somewhere between 4 and 8 is sufficient. A coarser gridding would tend toward greater generalization of the system. It may be possible to

determine analytically this gridding, but that is a topic suggested for future study.



a. Initial Configuration b. Final Configuration

Figure 37. Initial and Final Configurations of a Sample System Showing Meso-Scale Boundaries



Figure 38. Effect of Gridding Size on Count of Unique States  $\psi$ 

Grid size	First State	Last State	# of unique states
3	-1 		53
4			82
5			94
6			108
7			126
8			144
9			148
10			150

Table 11. Meso-scale System State Plots

If a system changes randomly then no one state should be more likely to occur than any other, i.e., the occurrence of a specific state should be nearly a random variable. However, it is important to recognize that to be truly random the motion of the particles would have to be random in their positions from observation to observation, i.e., a random walk. Since the goal of any simulation is to be representative of some real or notional system, a random walk of particles is a significant deviation from the nature of the system being simulated; so in this analysis we make use of "the minimally constrained case" as opposed to the "random case". In terms of EBS, an unconstrained case would remove any sustained information flows from the system but would still allow particles to collide with each other and entities comprising the environment; in this way, the nature of the system is maintained. If a system is indeed minimally constrained, then the entropy should be very nearly 1 according to equation (35) at each time-step. To verify this observation, a system of particles with randomly generated initial positions, velocities and headings was constructed. The particles' heading and velocity maintain their initial values unless affected by another particle or barrier by collision. This represents a minimally-constrained system by not allowing any relationship between entities but still allows the fundamental characteristics of the system to be present. Four runs were made, each with 10,000 time steps. The only difference being the initial positions, velocities, and headings. The following plots in Figure 39 show the occurrence of states for each of the four runs. The horizontal axis is the number of unique states and the vertical axis is how many times that state occurred. For example, the first plot shows that there were just over 600 unique states in the system and that some of them occurred as much as about 100 times. The last plot shows the number of unique states (2007) for all four runs.



Figure 39. State Occurrence of a Minimally Constrained System

Table 12 shows the mean, standard deviation, coefficient of variation, and Shannon Entropy for each of the data sets of Figure 39.

	Mean	std	cv	S
OC1	16.4745	14.4769	0.8787	0.9459
OC2	25.6410	21.8984	0.8540	0.9445
OC3	20.1613	18.7476	0.9299	0.9375
OC4	16.6113	15.0589	0.9065	0.9451
OCX	19.93	18.5365	0.9301	

Table 12. A Minimally Constrained Case Data Analysis

OC1 is the count of states in run 1 (state 1 is OC1(1) which occurs 16 times.) The mean of OC1 is the mean of the occurrences of all states that occur during run 1. We can observe that the standard deviation of run 1 differs from the mean by 12.13%. The coefficient of variability indicates the occurrence of states is highly variable; therefore, the system does not occupy any one state significantly more than any other does, which is to be expected of the unconstrained system.

#### 4.5 Hypothesis Tests

A hypothesis is a proposition that is consistent with known data, but has been neither verified nor shown to be false.

A null hypothesis  $(H_0)$  is a statistical hypothesis that is tested for possible rejection under the assumption that it is true, usually that the observations are the result of chance.

The hypothesis contrary to the null hypothesis  $(H_1)$ , usually that the observations are the result of a real effect, is known as the *alternative hypothesis*. In this study, we are asking two questions about the three simulated emergent behavior systems. This suggests two hypotheses must be tested using observed data from the EBS with the

appropriateness of the second hypothesis dependent on the success of the first. These two hypotheses, call them H1 and H2, can be simply stated in question form as:

H1: For each type of EBS, can emergent phenomena be detected?

H2: Assuming the presence of emergence in each of the three EBSs, can the three types of EBS be discriminated?

The bulk of this section will concern itself with the statistical definitions of these questions.

#### 4.5.1 Digression on Constraint and Randomness

The first hypothesis, H1, must seek to determine if a system that is exhibiting emergence can be recognized apart from one that does not. We will accept, for the sake of discussion, that if the state of a system at any given observation is explainable by random assignment of the state variables then it is likely that no emergence phenomena are present. In terms of our three measures, we would expect a high amount of variability in the measure of system complexity  $(\Omega)$ , no persistent information flows (f) leading to formation of structures, and no decrease in entropy (S) of the system over the set of observations. However, the term *random* is not straightforward when we are talking about a model of an EBS and deserves some discussion. As identified earlier, the various types of constraint that influence an EBS are of fundamental consideration. At first blush, it might seem appropriate to consider a random assignment of the state variables. For example, the particle system would be just a random scattering of particles absent of all constraints. However, in the case of a particle system, does a simple random scattering make the system truly random? At least in the strictest sense the answer would seem to be "No" since there are certain contextual considerations that if removed would fundamentally change the system. Recall that in the specification of an EBS, entities that are not actors as well as the context of observation define the EBS. For instance, even randomly distributing the particles is subject to the bounds of the space in which they are contained; change the container and you have changed the system. This fundamental spatial limit is an example of a contextual constraint discussed earlier. Similarly, if the particle system of consideration is affected by gravity, do we consider the random system to be without gravity? If these types of constraints are removed, is the random system still representative of the true system? Perhaps a more realistic *random case* for a particle system would maintain the context of observation and retain contextual and certain inherent constraints, i.e., all particles (in the particle system) must still exist within the spatial bounds, cannot occupy the same point in space, etc., but in short the system should still be recognizable as a particle system. This suggests that it is likely better to consider instead a form of the system where certain constraints remain but others are minimized. This form of the system would be one in which the mechanisms that enforce sustained relationships between actors are missing; as such the actors cannot form structures as described in the EBS specification. We can call this form of the system the minimally constrained system. The minimally constrained system will minimize the effects of variable constraints (or those that emerge during the evolution of the system) yet still maintain the characteristics governed by non-varying constraints, e.g., gravity. To an observer, the minimally constrained system will still be recognized as the system under study but will exhibit the nearly-random behavior we have in mind and not exhibit

the emergence phenomena. In this way, we can examine the EBS across the three measures by comparing the constrained case (emergence) to the minimally constrained case (non-emergence) and still have confidence that we have not fundamentally changed the nature and context of the system. From a model-based analysis perspective, this does place the responsibility of choosing which constraints to be minimized on the analyst. We will subsequently describe the minimally constrained condition for each EBS under study. Assuming of course that a system demonstrates emergence phenomena and as such is an EBS, the second hypothesis will examine the proposed measures of the EBS with regard to the three types of EBS of this study and see if these measures form a feature set to distinguish one type of EBS from another.

4.5.2 Hypothesis H1

Hypothesis: A system exhibiting emergence phenomena is indistinguishable from that system when it is not.

Based on the discussion in the previous section, we can refine this statement as:

- Null Hypothesis H<sub>0</sub>1: For the systems under study, each is indistinguishable from its minimally constrained version.
- Alternative Hypothesis H<sub>1</sub>1: each is distinguishable from its minimally constrained version.

As is typical, we will be seeking to accept the alternative hypothesis by rejecting the null hypothesis. Therefore, we will be looking to see if there is a statistical difference between a system that is minimally constrained and the same system when it exhibits emergence. Hypothesis: There exist several types, or classes, of emergent behavior systems which can be modeled using agent-based methods and that each class can be specified by a tuple of the form

$$E = [f(u(t)), \Omega(u(t)), S(u(t))]$$
(1)
where:

u(t) is a sample path over time t for a given set of initial conditions,

f is a measure of the principal information flows in the system,

- $\Omega$  is a measure of the complexity of the entities comprising the system, and
- *S* is a measure of the Shannon Entropy of the system.

Null Hypothesis  $H_02$ : The three types of emergent behavior systems in this study are indistinguishable by the measures specified by (1).

Alternative Hypothesis  $H_12$ : The measures specified by (1) can distinguish the emergent behavior systems used in this study.

Here we are interested in the cases where emergent phenomena are observed and we consider whether the three measures are sufficient to distinguish amongst the systems. Specifically, we are interested in the systems when emergence has manifested.

# 4.5.4 Hypothesis Test for Principal Information Flow f

The principal information flow metric is the average of the minimum path lengths, i.e., the characteristic path length (CPL), of the system within the context of observation. The null-hypothesis can be stated as  $H_0: f_C = f_U$ , where  $f_C$  is the CPL for the constrained system and  $f_U$  is the CPL for the minimally constrained system. The alternative hypothesis then is,  $H_1: f_C \neq f_U$ . In this case we are comparing means of the data from matched subjects (the constrained and minimally constrained systems are observed at each time t, subject to the same initial conditions), which suggests the use of a paired t-test. If we consider a test parameter  $\theta = f_C - f_U$  then we can rewrite the previous as  $H_0: \theta = 0$  and  $H_1: \theta \neq 0$ . Since it is unreasonable to expect  $f_C$  to be exactly  $f_U$  we are interested instead where they differ, i.e., within the alpha of the t-test.

#### 4.5.5 Hypothesis Test for Complexity $\Omega$

The global clustering coefficient (GCC) presented in Section 4.3 is examined in both the constrained and minimally constrained cases. The approach is the same as in the previous section; the null-hypothesis can be stated as  $H_0:\Omega_C = \Omega_U$ , where  $\Omega_C$  is the GCC for the constrained system and  $\Omega_U$  is the GCC for the minimally constrained system. We consider as the alternative hypothesis  $H_1:\Omega_C \neq \Omega_U$ . As above, if we consider a test parameter  $\theta = \Omega_C - \Omega_U$  then we can rewrite the previous as  $H_0: \theta = 0$  and  $H_1: \theta \neq 0$ .

#### 4.5.6 Hypothesis Test for Entropy S

The Shannon Entropy estimate given by Equation (35) and as with the previous metrics, we will consider the measure in comparison of the constrained and minimally constrained cases such that  $H_0: S_C = S_U$  and  $H_1: S_C \neq S_U$ . As in the others, we will consider the test parameter  $\theta = S_C - S_U$  so that  $H_0: \theta = 0$  and  $H_1: \theta \neq 0$ .
#### CHAPTER 5

#### SIMULATION, ANALYSIS AND RESULTS

To this point, the examination of emergent behavior systems has been kept to an abstract or theoretical level. In this chapter we will now exercise the ideas developed in previous chapters to produce the simulation models of the three types of EBS of this study. This chapter will describe:

- the motivation for and the design of the study simulations based on the previously presented EBS modeling formalism,
- 2) the instantiation of those designs in the NetLogo ABM framework,
- 3) the design of the EBS metrics in MATLAB, and
- 4) the analysis of the data produced by the EBS simulations

The three EBS selected for this study represent three fundamentally different classes of systems where emergence phenomena characterize the systems. These three systems are particles, flocking (or herding), and stigmergy. The actors in each EBS are representative of a different level of entity sophistication and each derives its principal information from distinct paths.

The particle system simulation is an interpretation of Hooke's Law where unit mass particles form structures through spring-like connections. Hooke's Law has found wide application in domains as diverse as robotics and biology (Hamdi, et al., 2005), (Oddershede, et al., 2002). Particles interact directly and the flow of information is very physical, either through direct contact with each other or their environment, or by the formation and breaking of the spring connections. Without any global to local feedback mechanisms, the particle system simulation can be an example of Type 1b emergence. We will conduct these experiments using a damping factor that insures that the net feedback is negative and as such is an example of Type 2a emergence. Additionally, the use of traditional "agents" to represent such a physical system emphasizes the assertion that "agent-based" is more of programming method or paradigm as each actor is very simple in its sophistication.

The Flocking system simulation is an implementation based on Craig Reynold's Boids algorithm (Reynolds, 1987), (Reynolds, 1999), which has found broad application such as to crowd behavior modeling (Chiang, et al., 2008). Three simple behaviors based on each actor's observation of other actors near it suggest a Type 2a emergence. The "boids" of the flocking simulation and the ants in the Stigmergic simulation are examples of more sophisticated traditional agents.

Stigmergy can be loosely defined as communication through the environment. The simulation examined here is that of trail forming ants and demonstrates the traits of Type 3 emergence described in Chapter 3. In this case, a spatiotemporal structure in the form of an ant trail emerges rapidly from positive feedback as ants deposit pheromone when they are carrying food. It then is moderated with negative feedback as the chemical trail disperses and evaporates.

#### 5.1 Applying the EBS Modeling Formalism

For each of the simulated EBS systems, we identify the actors and objects, the environment, and how these form the context of observation according to the ideas of Chapter 3. We make the distinction between what is micro-scale, i.e., belonging to the definition of the actors and what is meso-scale, i.e., what belongs to the interaction between actors.

# 5.1.1 Particles

The Particle System simulation is a demonstration of agent-based modeling applied to representing a well-defined physical system of masses and springs governed by Hooke's Law. The actors are points of unit mass. Relationships are governed by the spring-like behavior. Constraints include effects of gravity, objects in the environment, damping, etc. as shown in Table 13.

Factor	Value Range	Description			
particle-number	1 – 100	the number of particles in the system			
spring-constant	0 - 100	the value of spring constant			
rest-length	0 – max shear-point	the rest length of the spring separating two particles that occurs when the force on the spring is zero			
shear-point	0-120	the distance at which a spring will break			
damping	0 – 10	a decay factor, representing the viscosity of a fluid medium. Higher values cause quicker decay of energy			
Conditions					
damping-only	yes/no	particles are subject only to the decay of damping			
gravity-only	yes/no	particles are affected by gravity only			
neither	yes/no	particles are affected by neither damping nor gravity			
both	yes/no	particles are affected by both damping and gravity			
initial-condition	uniform-no-velocity uniform-rnd-velocity random-no-velocity random-rnd-velocity	particles can be initially uniformly distributed in a circle about the center of the context of observation with a diameter equal to the rest-length, or randomly placed within the context of observation, either with or without an initial random velocity			

Table 13. Factors and Conditions for the Particle System Simulation

The actors in this simulation are not typically associated with the common usage of the term "agent". Indeed, this simulation was developed in part to demonstrate that the agent metaphor might perhaps be more properly considered a programming paradigm (much like object-oriented programming) as opposed to a specific modeling construct only useful for certain types of representations. Information flows in the particle system when one particle interacts with another in such a way as to influence its behavior. The idea that information is "actionable data" is maintained; specifically, data is transferred from one particle to another by means of their relationship. For example, billiard balls transfer data when one strikes another. The data (velocity, momentum) is transferred on striking and the resulting reaction is the effect of the "actionable data". For the system of masses and springs here, the relationship is given by F=-kx, so any action taken by one particle is transferred to particles to which it is connected (in this example, all have the same k but x varies). As such, any time a particle strikes an object, information from that strike propagates throughout the system. In this way, the state of the particle system is affected whenever any particle of the system strikes another particle, the floor (ceiling) or wall. Figure 40 identifies the entities in the particle system simulation as well as how energy can propagate through the structures.



Figure 40. The Particle System Simulation Is an Interpretation of Hooke's Law

# 5.1.2 Flocking

The Flocking System simulation demonstrates self-organization of a group of entities like a flock or herd. Based on only a few simple rules of behavior, namely Reynolds' three rules of "alignment", "separation", and "cohesion", the flocking simulation illustrates a system where entities make decisions based on their perception of other entities. Figure 41 illustrates the three behaviors. "Alignment" means that a bird tends to turn so that it is moving in the same direction that nearby birds are moving. "Separation" means that a bird will turn to avoid another bird that gets too close. "Cohesion" means that a bird will move towards other nearby birds (unless another bird is too close). When two birds are too close, the "separation" rule overrides the other two, which are deactivated until the minimum separation is achieved. A unique extension developed in this simulation is the addition of the rule of "color". "Color" means that a bird will prefer others with his color. All these rules are set as preferences that are applied to all the birds in the simulation.



Figure 41. Reynolds' Alignment, Separation, and Cohesion Behaviors

Table 14 shows all the factors and conditions that affect behaviors in the flocking simulation. Figure 42 shows how the components of the flocking system are interpreted within the EBS specification.

Factor	Value Range	Description			
population	1 – 200	the number of birds in the system			
colors	1 – 10	the number of distinct colors. Each bird is assigned a color at the time it is created.			
affinity	0 – 100	the degree of preference a bird has for birds of its color. 0 is attracted to all birds equally 100 is only attracted to birds of same color.			
fov	10 - 360	fov in combination with the vision factor defines a cone of awareness for a bird. A value of 90 yields a fov of 45 degrees on either side of the direction of travel of the bird. A value of 360 means that a bird is aware of all other birds around it, ahead or behind.			
vision	0 – 35	the distance (in patches) that a bird can see. It is the radius of the fov.			
min-separation	0 – 5	the minimum distance (in patches) a bird will attempt to maintain from other birds			
max-align-turn	0 - 180	the maximum turn in degrees a bird will make to remain aligned in the direction of its flock mates			
max-cohere-turn	0 - 180	the maximum turn in degrees a bird will make to steer toward the center of its flock mates			
max-separate- turn	0 - 180	the maximum turn in degrees that a bird will make to avoid colliding with another bird			
mean-bird- speed	0-2 in 0.1 increments	the mean speed assigned to a bird at the time of its creation. Only used if Vary_speed? is true			
Vory speed?	l veo/no	if you hinds an appianed a gread shout the			
vary_speed?	yes/no	mean-bird-speed. Also, if true, bird speeds change based on various other conditions during the simulation run.			
Allow_Leaders?	yes/no	if yes, a bird that follows no other bird is a leader. A leader can have greater range and variation of speed if Vary_speed? is true.			

Table 14. Factors and Conditions for the Flocking Simulation



Figure 42. Interpreting the Components of the Flocking Simulation in Terms of the EBS Simulation Specification

#### 5.1.3 Stigmergy

The Stigmergy System simulation illustrates information flow between actors through the environment. It is also an example of collective intelligence, in this case where the ant-hill "knows" where the food is located, but the individual ants do not. Ants search randomly for food until they come upon a chemical trail deposited by ants that were carrying food. A form of stability is achieved when the number of ants is sufficient to sustain a path to the food. This is the popular pheromone following behavior that is central to ant-inspired self-organizing and path following algorithms (Dorigo, et al., 1999), coordination of mobile robots (Beckers, et al., 1994), and control of unmanned military robotic vehicles (Parunak, et al., 2001).

Figure 43 illustrates the conceptual representation of ants bringing food back to their anthill. The flow of information in this stigmergic system is from an ant, to an object in the environment, then to another ant when it traverses that location in the environment. In the approach taken here, ants wander about in search of food (red patches.) When they find food they pick some up and begin dropping pheromone into each patch (green patches) while they make their way back to the nest (black patches). This pheromone tends to spread out (diffuse) and to decrease in strength over time (evaporate). Wandering ants (brown) move about randomly searching for food or a trail to follow. When a wandering ant encounters a trail, it attempts to determine the direction that the pheromone is strongest (sniffing). The implication is that there will exist an increase in activity, and therefore pheromone deposits, closest to the found food supply. Ants that find a trail (blue) follow it in the direction of increasing intensity (uphill), which will usually be toward the food. The ants hold a limited supply of pheromone that is replenished each time they pick up food. They can exhaust their supply if the nest is too far from the food. A nest-scent helps the ants find their direction to the nest once they pick up food.

Information flow is from a carrying-ant to a patch, then from a patch to a sniffingant. There is also information flow from patch to patch, as the pheromone diffuses.

Figure 44 shows the stigmergy system of Figure 43 described as a graph. Notice that the green vertices share information with each other in a bidirectional way. This is in analogue to the dispersion of the pheromone across patches. The red vertices (numbers 1, 10, and 6), representing ants carrying food, have information flow from themselves to the green vertices. Likewise, the blue vertices represent ants following the trail; they take information from the green vertices. The numbering of the lattice in Figure 43 and the labeling of the graph in Figure 44 is somewhat in keeping with the numbering convention used in NetLogo for its World and Turtles.



Figure 43. Simulating Pheromone Following Ants

The resulting graph of the stigmergy system is fundamentally different from that of the previous systems. In both of those cases, relationships are primarily between like actors and the structures that result are comprised of those primary actors. Information flow is characteristically bidirectional in the particle simulation and directional in the flocking simulation. In the stigmergy example here, we see directional information flow associated with one mechanism, namely between the ants and the patches, and bidirectional flow associated with the patches. Notice that there is no direct communication between the ants in this model, but instead communication is through the intermediation of the patches with pheromone as shown in Figure 44.



Figure 44. Information Flow in the Stigmergic System

Observe that by identifying each patch as an actor, very large adjacency matrices can result. For example, a context of  $100 \times 100$  patches with 100 ants produces an adjacency matrix that is  $10,100 \times 10,100$ ; computing graph metrics such as those used in this study on matrices of this order is challenging to available computing resources. We can reduce this computational demand if we observe that the spread of pheromone in the patches produces structures that we can aggregate into a single vertex, or super-node in the graph representation of the system. In doing that, Figure 44 becomes like that of Figure 45.



Figure 45. Super-node Representation of the Stigmergic System

This greatly reduces the computational size of the resulting system graph to the number of ants and the number of contiguous pheromone patches (structures).

Table 15 lists the factors and conditions that affect behaviors in the stigmergy simulation. Figure 46 shows the components of the stigmergy system interpreted within the EBS specification.

Factor	Value Range	Description
Ants	1 – 200	the number of available to the system.
pheromone	1 – 10	This is the amount of pheromone an ant gets each time it is replenished. If it is not high enough, the ant will run out before it reaches the anthill.
use-nest-scent?	on/off	A nest-scent, similar to the pheromone trail, is used to help ants find the anthill.
Conditions		
diffusion-rate	0 - 10	This governs how fast the pheromone spreads to adjacent patches. 0 is never spreads while 10 is very rapid spreading.
evaporation-rate	0 – 10	This governs how readily the pheromone evaporates. 0 is no evaporation while 10 is very quick evaporation.
endless-food?	on/off	When set on, food does not diminish as it is picked up.

Table 15. Factors and Conditions Affecting the Stigmergy Simulation



Figure 46. Interpreting the Components of the Stigmergic Simulation in Terms of the EBS Simulation Specification

# 5.2 NetLogo Models

First introduced in 1999, NetLogo is a open source product produced by the Center for Connected Learning and Computer-Based Modeling at Northwestern University (Wilensky, 1999). NetLogo is designed for the simulation of multi-agent systems combining both a rich agent-oriented programming language and a simple but convenient graphical user interface in a tool for prototyping simulation developed for research applications (Sklar & Davies, 2005), (Sklar, 2007). NetLogo offers a combination of features that exemplify the EBS formalism described in Chapter 3. For example, the idea of NetLogo's "turtles" and "patches" facilitates the EBS entity concepts of actors and objects.

NetLogo includes a tool called "Behavior Space" to facilitate Design of Experiments (DOE). BehaviorSpace automates model runs so that a model can be run many times, systematically varying the model's settings and recording the results. As shown in Figure 47, experiments can be constructed with control of any factor in a NetLogo simulation, including full control of the random seed. Reporters are programming structures within a simulation that process and return data in a way prescribed by the programmer. The two key reporters in all three simulations are "plist" and "link-list". The reporter "plist" returns the position of every actor in the simulation. The reporter "link-list" returns a list of each actor and the others actors with which it is in relationship. This link list is used to create the adjacency matrix and other graph representations in later analysis. Using Behavior Space, data from any reporter can be saved at each time-step, i.e., NetLogo tick, or at the end of a run.

Behavior Space	Experiment			ĨX
Experiments:	Experiment name PS061610-1	1		
PH 18 14 20 11 20 12 630 / 2013	Vary variables as follows (note	brackets and quo	lation marks):	
PS0654060.308 (10 runs) PS0654060.308 (10 runs) PS0654060.308 (10 runs) PS05202010-1 (12 runs) ▼ New Edit Dupicate Delete Run	['damping' 0 1 2 3] ['shear-point' 3 5 16] ['random-seed' 23456] ['particle-number' 4] Exher for Julius to use, for examples ['mondfor 12 7 8] ['mondfor 12 for and and, for examples ['mondfor fin 1 10] [form additional bandward		K	
	to go from 0. 1 at a time, to 10. You may also vary men-caror, min		e-marter, random-seed,	
	Repetitions 2			
	run each combination this many times			
	Measure runs using these reporters:			
	link-list			17
	one reporter per lines you may not	split a reporter		<u> </u>
Run options	across multiple lines			
Spreadsheet output	Measure runs at every ste	ep 	_	
	Sebio commandis:	Wy Writen tangy are ove	r Go.commands;	
	setup	•	90	
Simultaneous runs in parallel 4			1	
Defaults to one per processor core.		*		~
OK Cancel	Stop condition:		Final commands:	·
		•		4
		~		*
	the run stops if this reporter become	es due	run at the end of each run	·····
	Time limit 10 stop after the menu steps (0 = pp	lemiz)		
		¯ 💌 (	Cancel	

Figure 47. NetLogo's BehaviorSpace

A sample portion of an output file created with this setup produces data in the

format shown below.

```
"BehaviorSpace results (NetLogo 4.1)"
"Particles_100407.nlogo"
"PS061610-1"
"06/16/2010 12:32:08:936 -0400"
"min-pxcor", "max-pxcor", "min-pycor", "max-pycor"
"0", "50", "0", "50"
"[run number]", "damping", "shear-point", "random-
seed", "particle-number", "[step]", "plist", "link-list"
"4", "0", "5", "23456", "4", "0", "[[12 42] [1 7] [22 4] [18
2]]", "[[0] [1] [2] [3]]"
"3", "0", "5", "23456", "4", "0", "[[7 17] [42 41] [22 13] [47
45]]", "[[0] [1] [2] [3]]"
```

"4","0","5","23456","4","1","[[12.04 42.05] [1 6.87] [21.899812069678646 3.9699064082561013] [18.030187930321354 2.060093591743899]]","[[0] [1] [2 3] [3 2]]"

Here we see some of runs number 3 and 4. This is actually in order of completion of a segment of a run. We can see that Run 4, damping = 0, step = 0 occurred before Run 3, damping = 0, step = 0. This is to be expected given that this experiment was on a fourprocessor workstation. Behavior Space assigns runs to processors, so we see Run 4, step = 1 output after Run 3, damping = 0, step = 0.

The first bracketed set of data is that from the plist reporter. In this simulation, the plist reporter reports the ordered list of particle positions. Here, Particle-0 is at position xcor = 12, ycor = 42, Particle-1 is at [1,7], Particle-2 at [22,4], and Particle-3 at [18,2]. "4", "0", "5", "23456", "4", "0", "[[12 42] [1 7] [22 4] [18 2]]", "[[0] [1] [2] [3]]"

The second bracketed set is the data from the link-list reporter. This reporter produces an ordered list of particles that are linked. In the previous line we see that no particles are linked. However, looking at Run 4, step =1 we see

]","[[0] [1] [2 3] [3 2]]"

Showing that Particles 0 and 1 have no links, Particle 2 is linked to Particle 3 and (as to be expected) Particle 3 is linked to Particle 2.

Understanding how the data output is formatted is critical to the MATLAB functions that are used to process and analyze this data.

### 5.2.1 Particles NetLogo Model

This model illustrates an EBS where the concept of entropy change is a result of

the redistribution of energy in a system to available microstates. These are subject to three forms of constraint; a) inherent, b) contextual, and c) entity. Inherent constraint arises from each entity's subjection to damping and gravity. Contextual constraint arises from barriers such as the boundaries of the environment. Entity constraint arises from the direct entity-to-entity communication governed by spring-constant, shear-point, and restlength.

The attributes of this simulation can be set to represent a number of stuctures composed of a number of particles that are bound together with Hookian springs, with particular equilibrium length and spring constants. For instance, if gravity is permitted in the environment, a ball of particles starts above the "ground" and when allowed to fall under the influence of gravity rebounds off the ground or any other barriers it encounters. Energy is conserved in this process, nevertheless the height of the ball of particles decreases with each bounce as the initial potential energy of the system is transferred to disordered vibrations of the internal particles. External mechanical energy, such as potential energy and ordered kinetic energy is transferred to internal potential energy of the springs and disordered kinetic energy. It is possible to investigate how the rate at which this energy is redistributed relates to factors such as the strength and length of the springs, the number of particles, and the barriers it encounters. Also of interest are the final equilibrium configurations of the structures and when this equilibrium occurs. Particles are created randomly or in a prescribed manner around the center of the context of observation. In the simulation each particle interacts pair-wise with every other particle via a Hooke's Law F = -kx force, where x is the distance between the two interacting particles minus the equilibrium length of the spring. The particles will not

pass through each other but collide fully elastically. The motion is simulated numerically based on Newton's second law F = ma. In order to have precision with the simulation and to conserve energy we use a fourth order Runge Kutta. There are options to include a number of other forces in the simulation. A damping force serves to gradually dampen out the motion so that the particles can reach an equilibrium configuration, much as if the particles were in a viscous medium. The simulation is interactive. For example, when an equilibrium is established, damping can be turned off and gravity turned on to allow the structure to fall. It is also possible to have both damping and gravity at the same time, and neither. The elastic, gravitational and kinetic energy of each particle is calculated at each time step. The internal energy is set equal to the total elastic energy of the particles and the kinetic energy that is in excess of the center of mass kinetic energy. The total energy is the total gravitational potential energy and the center of mass kinetic energy. Particles form structures if they are closer than the shear-length. If their separation exceeds the shear-length then the spring is broken. Figure 48 shows the graphical user interface (GUI) of the particle simulation. The upper left portion of the GUI has buttons and sliders to set factors that describe particle characteristics and simulation controls. On the middle left are the controls for initial conditions and the environment, e.g., gravity on/off, etc. The lower left are user selectable visualization settings. These settings allow for the visualization of the connections between particles, indication of particle-to-particle collisions, etc. The right side provides run-time plots and update of data. Included in these plots are the kinetic and dynamic energy in the spring system as well as EBS related metrics of then number of structures, the size of structures, and the normalized mean degree of the graph associated with the system. The center of the GUI is NetLogo "World

View" which is the EBS context of observation. The dots are the particles, i.e., the actors of interest. The background is composed of NetLogo patches. A patch is the unit of spatial measure for the system and is in this case a 100 x 100 patch area. Just like the turtles that are the agents in a NetLogo simulation, each patch is an agent as well but unable to move like the turtles. The patches along the borders have traits that the particles recognize. Particles bounce off the borders in elastic collision. The buttons along the bottom are intended to allow the user to explore the placement of barriers within the environment or to add particles interactively.



Figure 48. The Particle Simulation GUI

## 5.2.2 Flocking NetLogo Model

The flocking simulation is somewhat based on the flocking of birds where birds of different colors prefer grouping together. In this model, the flocks that emerge appear to identify a de facto leader. This leader is simply the only bird in a flock that does not see another bird to follow. In this way, a leader "emerges". In a sense, there are not any leaders: only birds that are not following.

These rules affect not only a bird's heading but its speed as well. Each bird tries to move at the average speed of its flock mates but even this speed can be statistically varied. (This can be thought of as adding noise to the speed property.)



Figure 49. The Flocking Simulation GUI

# 5.2.3 Stigmergy NetLogo Model

The stigmergy simulation represents the considerations addressed in Section 5.1.3. As can be seen in Figure 50, the simulation allows factors to be varied and scenarios to be run. Since the patches are actors of interest in this simulation, the resulting link lists are much larger than those of the previous simulations are. The computation of the structures and the normalized mean degree are very computationally intensive with matrices of this scale. Instead, system status information is plotted in real-time, showing remaining food and the percentage of ants carrying food. The link lists produced for analysis implement the concept of the super-node described in Section 0.



Figure 50. The Stigmergy Simulation GUI

# 5.3 EBS Metrics in MATLAB

Appendix B contains MATLAB code listings for all the analysis functions developed for this study. The following sections detail the algorithms and use of the three EBS metrics. In each case, the basis for the measurements are related to the representation of the interacting entities in the form of graph vertices and edges. To facilitate that, the simulations produce output of relations in the form of an adjacency, or link list as described previously in Section 5.2. This link list is used to create the graph adjacency matrix of the form described in Section 3.2. The MATLAB function make\_adjacency\_lists.m extracts the link list from data files created in NetLogo and the function list2mat.m produces the corresponding adjacency matrix. At the MATLAB command line, this looks like

>> SAM = list2mat(LL)

Where LL is the link list and SAM is the resulting adjacency matrix.

5.3.1 Information Flow (Characteristic Path Length)

The argument for use of the characteristic path length (CPL) as a measure associated with information flow in an EBS was made in Section 4.2. To compute the CPL for a graph given by an adjacency matrix, two functions are used:

pathlength.m returns a matrix of the distances between vertices in a graph.

charpath.m finds the mean distance (path length) between every pair of vertices.

The MATLAB command line expression looks like

>> charpath(pathlength(SAM))

A characteristic of the adjacency matrix is that the distance between any pair of vertices can be found by raising the matrix to a power corresponding to the distance. For example, the tree graph shown in Figure 15 of Section 3.2 has easy to recognize paths. Its adjacency matrix is given by,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$
(36)

The matrix A shows the paths of length 1 between every pair of vertices. Squaring A will show the paths of length 2 between pairs of vertices and the diagonals show the degree of a vertex (if the graph is undirected.) In general, raising a graph to the  $k^{th}$  power will identify the number of distinct paths of length k between vertices.

$$A^{2} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 3 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$
(37)

If we let  $S_k = A + A^2 + ... A^k$ , the first k for which the (i, j) entry in  $S_k$  is the count of the paths from vertex i to vertex j in k steps or less. If the entry is 0, then there is no path of length k between the two.

The function pathlength.m computes the shortest path lengths between all vertices in a graph by incrementally raising the adjacency matrix of degree n to increasing powers until either there are no elements equal to zero, or the  $(n-1)^{st}$  power

has been reached. It then records the first power at which the (i,j) element became nonzero. Self-loops are not allowed. For the example used here, the function is applied as,

>> pathlength(A)

Which produces the output

ans =					
Inf	1	2	3	2	3
1	Inf	1	2	1	2
2	1	Inf	3	2	3
3	2	3	Inf	1	2
2	1	2	1	Inf	1
3	2	3	2	1	Inf

This result is the minimum path length between any pair of vertices. This is called a distance matrix. The function charpath.m operates on the distance matrix to produce the average shortest path length, i.e., the global mean of the distance matrix, excluding any Infs. The function includes some other graph statistics as well that are not used in this discussion. To compute the flow metric for this study, i.e., the CPL, given any adjacency matrix A, the CPL is computed using the MATLAB functions developed here as

>> CPL = charpath(pathlength(A))

For the example of Figure 15 of Section 3.2, CPL = 1.9333.

5.3.2 Complexity (Global Clustering Coefficient)

In Section 4.3 we described the foundation for the choice of the global clustering coefficient (GCC) as a measure of complexity of an EBS. Recall that the clustering coefficient for each node in a simple graph is given by the ratio of the number of triangles that can be formed on that node and the number of triads in which that node is central. As

we discussed in the previous section, we know that the square of A will give the number of paths of length 2 and raising it to the third power will give the number of paths of length 3. Again, we look to another practical quality of the adjacency matrix, specifically that we can find the number of triangles with a vertex v in the center by looking at the diagonal elements of  $A^3$  as these will be the paths of length 3 that start and stop on vertex v. We have to divide that result by 2 since the triangles are counted both directions. The number of triads with v central is given by d(v)\*(d(v)-1)/2 where d(v) is the degree of v. The clustering coefficient for the graph is the average of the clustering coefficients for those nodes which have a clustering coefficient, i.e., those with d(v)>1.

The MATLAB function that computes the clustering coefficient is clusco.m which computes the clustering coefficient from the adjacency matrix of a simple graph. It returns both the clustering coefficient of each individual vertex as well as the global clustering coefficient for the graph.

The MATLAB code for clusco.m can be seen in Appendix B: MATLAB Code & Scripts. Referring to the middle graph of Figure 35 in Section 4.3, the corresponding adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$
(38)

The clustering coefficients for this graph are computed with

>> [cg,cn]=clusco(A)

which returns

cg = 0.7667 which is the GCC for A, and

$$cn = 1.0000$$
 0.6667 0.6667 1.0000 0.5000

which is the clustering coefficient at each vertex in A.

Clustering coefficient is defined for simple graphs only. The function clusco.m will take a directed graph and convert it to a simple graph, i.e., undirected and no self-loops.

#### 5.3.3 Organization (Shannon Entropy)

In Section 4.4 we presented the foundational argument for the use of Shannon Information Theory, i.e., Shannon Entropy as an information measure of organizational uncertainty in the EBS. Recall that computing the form of Shannon's expression used as the entropy metric (Equation 35) requires observations of the states accessible to the system and the determination of the probability of finding the system in each of those states. This is accomplished parametrically from the simulation of a system and making repeated runs with varying initial conditions. At each time-step of the simulation, the probability of each observed state is estimated by counting the number of times that state was observed.

Whereas Section 4.4 addressed the states of the system as a gridding of the 2D context of observation, it is incumbent on the simulationist or systems engineer to specify what constitutes the system states. Since in practicality the system states are dependent on the nature of the system and the intent of the study of that system, the implementation of the entropy metric in this study is generalized so that any state representation can be used for the computations. In particular, the system states monitored in the simulations, namely *structure count* and *normalized mean degree*, form a convenient state vector that

is independent of some problematic system characteristics. For example, in the flocking simulation, flocks are the structure of interest and the richness of their interaction is represented in the normalized mean degree. This is invariant with respect to the direction (or orientation) of the flock or the particular position of the flock. This seems to be a reasonable assumption given that a flock may change speeds and directions many times during observation. However, if positions and orientations were the characteristics of interest, the same MATLAB code developed here could be used. In that case, the state vector would consist of such measurements such as the spatial positions of each bird, the orientation of the flock, the number of birds in a flock, etc. The gridding technique described in Section 4.4 is perfectly acceptable to the MATLAB code developed here; the computation of the entropy metric is applicable to any form of state representation. As described in Section 5.2, the simulations produce a real-time readout of system state defined by structure count and normalized mean degree. For this study, the system state vector is of the form given by

$$S(t) = [C(t), Dn(t)]$$

where,

C(t) is the number of graph components (EBS structures) at time t Dn(t) is the normalized mean degree at time t of the system considering all the actors so that Dn(t) = mean(d(t))/(N-1)

where N is the number of Actors (N-1 is the maximum degree of a vertex if the system were fully connected.)

The MATLAB functions that enable this analysis consist of the following principal functions. Note that there may be other functions called in these functions that aren't necessarily identified here. Please refer to Appendix B: MATLAB Code & Scripts. ebssv.m creates an EBS state vector of the form S(t) = [C(t), Dn(t)] and statecountx.m examines all states of the system in an experiment, comparing each to each other to count how many unique states are in the system and how many times they occur. findstatex.m is a helper function (see Appendix B: MATLAB Code & Scripts.) N is the set of row vector observations of the form:

> N(1,:) = sv1 sv2 ... svm -- first observation N(2,:) = sv1 sv2 ... svm -- second observation N(n,:) = sv1 sv2 ... svm -- nth observation

shannonx.m takes a matrix N as one observation set of states that result from Monte Carlo experiments, typically for a given time step in a simulation. N is of the form N(r,c,s) where r and c are the rows and columns identifying each state vector of the state matrix and s is the sample index. This function depends on statecountx.m.

The following command line examples describe the four steps for computing the estimate of the Shannon Entropy. The test data set is the result of a Monte Carlo approach where the simulation is run repeatedly. The example shown here is for 10 runs, i.e., 10 different random seeds each with 1000 time-steps.

Step 1: Get the filename and the data.

```
>> TMCPS = 'C:\Users\Thomas\Emergent Intelligent
Behavior\Dissertation\Experiments\Particle
System\Particles_120817x Test-MCP-shannon-table.csv`
```

Step 2: Get the adjacency lists for each run.

>> TMCPSal= make adjacency lists(TMCPS,1:10);

Step 3: get the state vector matrices for all runs. Keep them in a cell array for easy access.

>> for n=1:10;TMCPSsv{n}=EBSSV(TMCPSal,1:1000,n);end

The state vector matrix for just one run, in this case the first random seed run can be found with the following:

>> TMCPSsv1=EBSSV(TMCPSal,1:1000,1);

Step 4: Compute the Shannon Entropy for each run.

>> for n = 1:10; TMCPSse(n)=shannonx(TMCPSsv{n});end

#### 5.4 Preliminary Analysis

In this preliminary analysis of the systems, we are simply exploring the nature of the systems as they evolve in order to gain insight into what might be expected when simulation factors are varied. System behavior is easy to observe given the visualization incorporated into the simulations. A useful observation is that since it appears that the systems are increasingly non-random as constraint increases, a comparison of the metrics f,  $\Omega$ , and S may be made between the minimally constrained and constrained systems, i.e., between non-emergent and emergent forms of the systems. We are particularly interested in inspecting the states of the different systems in the non-emergent and emergent forms.

#### 5.4.1 Preliminary Inspection of the Particle System

To explore this conjecture in the particle simulation, we consider the EBS under study when the actors form random relationships. In the case of the particle system simulation, this would be when the particles are distributed randomly and with varied values of the *shear-point* factor. In the particle simulation, increasing the *shear-point* has the effect of increasing the likelihood that one particle is in relation to another. As such, the EBS in the minimally constrained case can be considered a *nearly-random* graph with the probability of a vertex adjacent to another being dependent on the value of the shearpoint factor. The effect of increasing shear-point for a single random arrangement of particles in the EBS can be seen in Figure 51 where the value of *shear-point* is increased from 5 to 25.



shear-point = 15

shear-point = 25

Figure 51. The Effect of Increasing the shear-point Factor

# 5.4.2 Preliminary Inspection of the Flocking System

Similar to the preliminary analysis of the particle system, we identify the actors of the flocking system as the birds and assess the effect of increasing constraint on the system. In this case, the constraint factor(s) that are varied are those that affect the birds' ability to maneuver. As Reynold's behaviors consider alignment (max-align-turn), cohesion (max-cohere-turn), and avoidance (max-separate-turn), we construct a scenario with these factors maximized to create the minimally constrained case. In this manner, the birds have the most variety available to them. Note that this is not just the random

flight of birds, but the system allowed the greatest variety. To achieve something nearly random, we can constrain the factors governing the birds' behaviors such that one bird cannot influence another. Figure 52 shows a screen shot of the minimally constrained system while Figure 53 shows the system with *max-align-turn*, *max-cohere-turn*, and *max-separate-turn* set to 0 to produce a nearly random system. A factorized study of the factors would identify combinations of values leading to specific system characteristics but in this study, we are only considering the presence of emergence. Therefore, to facilitate analysis, we consider only varying one of the factors, namely *max-align-turn*. Figure 54 shows some final configurations of the system as *max-align-turn* is progressively increased. In both Figure 52 and Figure 54, the simulation is shown with links between birds visible; this allows for the structure formed by the relationships to be observed during runtime.



Figure 52. An Example of the Minimally Constrained Flocking System



Figure 53. An Example of the Nearly Random Flocking System



Figure 54. Effects of Increasing max-align-turn in the Flocking System

To investigate the nature of the effect of increasing variety, we also examine scenarios of increasing vision; although the birds are not affected by seeing the other birds, the information pathways, and in a graph the edges, are present. Figure 55 shows three nearly random scenarios, each with the same initial conditions. In the top screenshot, the birds' range of vision is 5 patches. The middle is 10 and the bottom is 15.



Figure 55. Flocking Nearly-Random Scenarios
We observe that the nearly random scenario results when the values governing the birds' ability to align and turn are minimized, i.e., highly constrained. The minimally constrained system, i.e., the one with the most variety, consistently produces a single structure system with a normalized mean degree of 1.

## 5.4.3 Preliminary Inspection of the Stigmergy System

As was pointed out, the stigmergy system depends on indirect interactions between entities. Unlike the particle and flocking systems where we are mostly concerned with a single type of actor, the stigmergy system demonstrates information flow between different types of actors, namely between the ants and the patches.

To achieve a nearly-random scenario, we set the *pheromone* of the ants to 0 and set *use-nest-scent*? to off.



Figure 56. Stigmergy Nearly-Random Scenario

Figure 56 shows the nearly-random system with 152 ants. Ants cannot drop pheromone and so no trails can be formed. They are unable to sense the nest so they encounter food randomly, and take it to the nest randomly. Figure 57 shows the Stigmergy simulation after over 1000 time-steps in the simulation. In this nearly-random scenario resulting from highly constraining the system by not allowing any pheromone or nest scent, ants encounter food and pick it up, but they are unable to efficiently find the nest. As they randomly come across the nest, they deposit food, resulting in approximately 70% of the ants carrying food.

In Figure 58, we show the results of a minimally constrained scenario. Here, variety is maximized and ants have unlimited pheromone. The presence of so much pheromone overwhelms any useful path formation. The resultant behavior is much like the nearly-random scenario with high constraint, exhibiting nearly 70% of ants carrying food at dynamic equilibrium.





Figure 57. Inefficiency in the Nearly-Random Stigmergy Scenario



Figure 58. Minimally Constrained Stigmergy Scenario

Setting *use-nest-scent*? on allows ants that have found food to efficiently return to the nest. About 10% of ants are carrying food since they quickly deposit it at the nest as shown in the top image of Figure 59. Subsequent images in Figure 59 show the effect of varying the amount of pheromone an ant has available to deposit on the patches. As *pheromone* is increased, ants have a greater likelihood of finding a path the food.











With *use-nest-scent*? on, ants return food to the nest more efficiently, but finding food is still random. Approximately 10% of ants are carrying food.



A *pheromone* setting of 25 allows some trail to be formed. About 20% of ants carry food.



With *pheromone* = 50, approximately 25% of ants are carrying food.





Figure 59. Effects of Increasing Pheromone Level in the Stigmergy System

When the ants cannot deposit pheromone in the patches, no relationships are formed. No communication occurs and so no information network is established between the ants. Sensing the nest and allowing the formation of a trail improves the efficiency of the ants getting food to the nest.

The unique nature of this stigmergy system has significant implications for the analysis metrics. Since the information flow is from the ants carrying food, i.e., those depositing pheromone to the super-node, then from the super-node to those ants who are following the trail, the path lengths between ants will always be either 0 or 2. The characteristic path length for the system will then always be between 0 and 2. Similarly, since all information flow is through the structure represented by the super-node, the global clustering coefficient will always be 0.

### 5.5 Design of Experiments

For the particle system, H1 depends on the statement of the minimally constrained version of the system. Given the factors associated with constraint, one way of specifying the minimally constrained version of this system is to consider a random placement of particles with a spring constant of zero. This reduces entity constraint while maintaining the context of the simulation. Exploratory experiments with the system when the damping factor is zero yields a constantly changing location of particles where the normalized mean degree and structure count continuously changes but maintains an approximate average values of 0.03 and 11 respectively. Changes in spring constant or rest length do not change these values. This indicates that a minimally constrained version of this system can be represented by a randomization of the position of the particles. In this case, the critical factor that affects system metrics is the *shear-point* as

was seen in Section 0. Figure 60 shows five instances of the minimally constrained case. Figure 61 shows the EBS relation-space state vector parameters of structure count and normalized mean degree for the minimally constrained case.



Figure 60. Five Instances of the Minimally Constrained Case



Figure 61. Structure Count and NMD for the Minimally Constrained Particle System

Now we construct an experiment to produce data for the particle system where constraint is systematically increased. In this experiment, we will start the system with the same initial conditions and allow it to evolve until it reaches equilibrium. The fifty particles are randomly distributed about the context of observation. An initial random velocity is assigned to each particle. We will look at the system as an inherent constraint varies. Table 16 shows the factors for the inherently constrained particle system simulation and considerations for the experiment.

Factor	Value	Notes
particle-number	50	The number of particles in the system is the same as in the minimally constrained case.
spring-constant	50	The initial energy in the system is produced by the initial placement of the particles and the resulting force of the springs connecting them.
rest-length	1 to 10	The particles will seek this length. varied from 1(high constraint) to 10 (no constraint)
shear-point	10	This is the distance at which a spring will break and release its particles. For this experiment it is held constant and kept small enough to reasonably reduce the influence of contextual constraint attributed to the boundaries of the context.
damping	1	A damping value of 1 allows the entities in the system to interact for a longer time.
		Conditions
damping-only	Ves	Damping is an environmental constraint
gravity-only	no	
neither	no	
both	no	
initial-condition	random-rnd-velocity	Initial state of the system is randomized both in particle position and velocity (high disorder to start)

Table 16. Experiment Considerations for the Inherently Constrained Particle System

Table 17 identifies the enumerations for the simulation runs, referred to as Test-IC. Rest-length is the single factor that is varied. The system is allowed to evolve until equilibrium. Each factor value is iterated over 100 random seeds to produce 1000 enumerations.

	Enumerations (run number) for Test-IC										
	Factor (rest-length)										
		1	2	3	4	5	6	7	8	9	10
	1	1	101	201	301	401	501	601	701	801	901
dom seed	2	2	102	202	302	402	502	602	702	802	902
	3	3	103	203	303	403	503	603	703	803	903
	:	:	:	:		:	:	:	:	:	:
	99	99	1 <b>99</b>	299	399	499	599	699	799	899	999
Ran	100	100	200	300	400	500	600	700	800	900	1000

Table 17. Factorial Experiment for Inherent Constraint in the Particle System

In this experiment, the inherent constraint increases as the *rest-length* increases. That is to say, when the *rest-length* is very small relative to the *shear-point*, the particles have more opportunity to establish relationships without breaking relation when the *shear-point* is exceeded. As the *rest-length* increases, there is less opportunity to move without exceeding the *shear-point*. Also, note that a particle that exceeds the *shear-point* may be freer to form bonds with other particles, but it has very little leeway in doing so. The intensity of constraint,  $I_C$ , for this experiment is then stated as,

$$I_C = \frac{rest - length}{shear - point}$$
(39)

If *rest-length* is equal to *shear-point* then particles are beyond the influence of other particles and the  $I_C = 1$ , the maximum constraint. As *rest-length* decreases, there is more variety available to the particles, i.e., the intensity of constraint is lower.

Figure 62 shows the extremes of the rest-length influence and Figure 63 shows

sample system response with increasing rest-*length*<sup>5</sup>. By varying only the *rest-length*, the system goes from creating neatly uniform balls to non-symmetrical dragons.



 $I_c = 0.9.7$  structures, NMD = 0.12163

 $I_c = 0.2.7$  structures, NMD = 0.04082

Figure 62. Inherent Constraint – High  $I_c$  on Left, Low on Right



Figure 63. Typical System Results for Increasing rest-length

<sup>&</sup>lt;sup>5</sup> An interesting observation that is evident in the analysis of the next section as well, in running this simulation there appears to be a value of *rest-length* where the resultant structures rapidly become characteristically different from those of lesser *rest-lengths*.

## 5.5.1 Flocking System

In Section 0 we created a nearly random scenario of the flocking system by reducing the factors that allow the birds to take information and act on it. We also considered a scenario where the variety available to the system was maximized. We observed the effect of increasing the value of the *max-align-turn* factor. For this experiment, we select factor values that produce satisfying flocking and vary just one factor, *max-align-turn*, while keeping others constant. Table 18 lists the considerations for the flocking experiment. Table 19 enumerates the runs. Each run is for 1000 time-steps and data is collected at each time-step.

Factor	Value	Notes
population	50	The number of birds in the system is the same as in the minimally constrained case.
colors	1	We limit the experiment to a single color of bird
affinity	100	The birds have strong attraction to their color. (Since there is only one color this doesn't matter.)
vision	10	Birds can see for 10 patches.
fov	240	A good value similar to that of most real flocking birds.
min-separation	2	Birds can get close.
max-align-turn	0 to 24	This is the factor we choose to vary in steps of 2
max-cohere-turn	4	
max-separate-turn	12	
seedval	1 to 10	We will run the experiment 10 times with different random initial conditions
		Conditions
Vary_speed?	false	Birds fly at a constant speed.
Allow Leaders?	false	Leaders not allowed

Table 18. Experiment Considerations for the Flocking System

	Enumerations (run number) for Flocking Experiment									
	Factor (max-align-turn)									
		0	2	4	6		22	24		
p	1	1	11	21	31		111	121		
n See	2	2	12	22	32		112	122		
nobn	3	3	13	23	33		113	123		
Ra	:	:	;	:		:	•	:		
	9	9	19	29	39		119	129		
	10	10	20	30	40		120	130		

Table 19. Factorial Experiment for the Constrained Flocking System

In this experiment, the intensity of constraint decreases as the value of *max-alignturn* increases, i.e., more variety is available to a bird with increasing ability to turn to align itself. For convenience, we state the intensity of constraint as

$$I_C = 1 - \frac{m}{\max(m)} \tag{40}$$

where m = max-align-turn.

Figure 64 shows the system for the same initial conditions at 500 time-steps with a sampling of values for *max-align-turn*. Observe that the normalized mean degree increases as the variety in the system increases.



Figure 64. Typical Flocking System Response to Increasing max-align-turn

#### 5.5.2 Stigmergy System

As was seen in Section 0, the scenario that is most like random movements of ants the one with *use-nest-scent*? = off and *pheromone* = 0. In this case, no information flows and consequently the patches do not change state or make connections to other patches. To determine the H1 hypothesis for the stigmergy system, we will use the minimally constrained version of the system and compare to higher values of pheromone with usenest-scent? = true. Recall Figure 59 where pheromone is increased progressively; Table 20 shows the factors settings for the H1 test. We will compare the effects of increasing *pheromone* to the case where *pheromone* = 0.

Factor	Value	Notes
ants	100	The number of ants in the system is the same as in the minimally constrained case.
pheromone	0 - 100	The amount of pheromone available to each ant is increased with each experiment from 0 to 100 in increments of 10.
use-nest-scent?	on	The ants will follow the scent of the nest once they pick up food.
diffusion-rate	5	Pheromone diffuses to surrounding patches.
evaporation-rate	10	A faster rate of evaporation is used.
Conditions		••••••••••••••••••••••••••••••••••••••
endless-food?	on	Food supply does not decrease

Table 20. Experiment Considerations for the Constrained Stigmergy System

Table 21 records the enumerations for the simulation runs, referred to as Test-SC. The only factor that is varied is *pheromone*. The system is run for a warm-up period of 500 time-steps. This allows for all ants to leave the nest and for trails to be established. With endless food, this sustained trail is considered an equilibrium condition. Each factor value is repeated for 100 random seeds to produce 1000 enumerations.

	Enumerations (run number) for Stigmergy Experiment										
				Fa	ctor (ph	heromo	ne)				
		10	20	30	40	50	60	70	80	90	100
b	1	1	101	201	301	401	501	601	701	801	901
n see	2	2	102	202	302	402	502	602	702	802	902
iopur	3	3	103	203	303	403	503	603	703	803	903
<sup>22</sup>	:	:	:	:		:	:	:	:	:	:
	99	99	199	299	399	499	599	699	799	899	999
	100	100	200	300	400	500	600	700	800	900	1000

Table 21. Factorial Experiment for the Constrained Stigmergy System

The intensity of constraint,  $I_C$  for this experiment is simply,

$$I_{C} = 1 - \frac{pheromone}{\max(pheromone)}$$
(41)

When *pheromone* is large, the  $I_C$  is small. When *pheromone* = max(pheromone) in the experiment, the  $I_C$  is minimized and indicated as 0. Figure 56, Figure 57, and Figure 59 in Section 0 showed the effect of increasing the value of *pheromone*.

#### 5.6 Analysis and Results

As described in Chapter 4, we look to see how the metrics associated with the definition of the system scenarios that produce nearly-random behavior compare to the metrics when the system is constrained to exhibit emergence. For the H1 analysis, where we seek to discriminate emergence from non-emergence, we use a t-test with an alpha of 0.05 on each of the metrics and using MATLAB's statistical analysis capabilities. To examine the metrics' ability to differentiate the three systems, i.e., the H2 analysis, we examine the data across all three metrics using a one-way analysis of variance (ANOVA).

5.6.1 Presence of Emergence in the Particle System - H1

Table 22, Table 23, and Table 24 present the results of the t-test against data produced by the simulation experiment. The mean of the nearly-random scenario is shown in the upper left of the tables. We observe that for each of the three metrics the null-hypothesis  $H_01$ , that the system in emergence is unrecognizable from when it is not, is strongly rejected.

In the case of the flow metric, i.e., the characteristic path length (CPL), we report p-values = 0. Reporting p-values of 0 is indicating that we would not observe the minimally constrained value in the data. However, when we examine this data, which is measured when the system is in equilibrium, we see consistent formation of structure with the system always producing the same result for strong constraints.

$\hat{f}_{MC}$ =	=1.8354		
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)
0.1	0	1 1	reject
0.2	0	1 1	reject
0.3	0	1 1	reject
0.4	0	1 1	reject
0.5	0	0.9999 1.0003	reject
0.6	0	0.9998 1.0010	reject
0.7	0	1.3322 1.3786	reject
0.8	0	2.8614 3.2885	reject
0.9	0	2.3638 2.6494	reject

Table 22. H1 Results for the Flow Metric (CPL) in the Particle System

Figure 65 is a box plot showing the minimally constrained results for CPL and the data for the system as the factor *rest-length* ranges from 1 to 10. Here we can see that between the 8<sup>th</sup> and 9<sup>th</sup> increments, i.e., 7 < rest-length < 8, a transition occurs that is characteristically different from when the constraint was higher. We believe that this can be attributed to the fundamental physical characteristic of the particle simulation since the rest-length in this region is approaching the shear-point value, when exceeded will yield no path between particles (see the last two images of Figure 63).

Figure 66 shows typical plots for three values of *rest-length*. The top plot is lower constraint while the bottom is highest. Here we can see that when the difference between *shear-point* and *rest-length* is greater (top), there is more opportunity for the system to share information and form structures quicker. In the top plot, stability is achieved by about time-step 300, while in the more constrained systems it is not achieved until around 500.



Figure 65. Comparing Flow (CPL) of the Particle System as Constraint Increases



Figure 66. Comparing CPL at rest-length = 4 (top), 7, and 9 (bottom).

Figure 67 plots the 1- p-value for CPL in the region where the system transitions to stability. Here we can clearly see when the null-hypothesis begins to be rejected and the stable structure emerges.



Figure 67. Plot of 1- p-value for CPL

Similar results are recorded for the complexity metric of global clustering coefficient (GCC). Figure 68 shows the GCC for the experiment as  $I_C$  increases in columns 1 through 10. Observe that there is a transition point between 9 and 10 where the system's complexity would likely be similar to the nearly-random complexity if data were taken at 8 < rest-length < 9. Column 11 is the data for the nearly random scenario.

$\widehat{\Omega}_{_{MC}}$	= 0.2553		
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)
0.1	4.0068e-011	0.8735 0.9545	reject
0.2	0	0.9043 0.9233	reject
0.3	0	0.8958 0.9142	reject
0.4	0	0.8996 0.9188	reject
0.5	0	0.8868 0.9063	reject
0.6	0	0.8808 0.9021	reject
0.7	0	0.6580 0.6841	reject
0.8	0	0.3729 0.4012	reject
0.9	0	0.1115 0.1357	reject

Table 23. H1 Results for the Complexity Metric in the Particle System



Figure 68. Comparing Complexity (GCC) of the Nearly-Random Particle System to the System as Inherent Constraint Increases

The data analysis for the Shannon Entropy (SE) is shown in Table 24 and in Figure 69. Notice how this metric reveals a condition in the system where there is a

sudden drop in entropy after a period of increasing entropy. The SE for the nearlyrandom scenario is in column 11 of the plot.

$\hat{S}_{MC}$	= 0.8118			
Ic	p-value	95% confiden	ce interval	H <sub>0</sub> (accept/reject)
0.9	3.2311e-009	0.4094 (	).4829	reject
0.8	3.3760e-011	0.3974 (	).4445	reject
0.7	9.9117e-008	0.4278 (	).5271	reject
0.6	5.1240e-008	0.4453	0.5340	reject
0.5	1.5253e-007	0.4969	0.5820	reject
0.4	4.0634e-005	0.5697	0.6829	reject
0.3	0.0015	0.6366	0.7541	reject
0.2	7.0165e-010	0.5100	0.5571	reject
0.1	2.3690e-009	0.5023	0.5571	reject

Table 24. H1 Results for the Entropy Metric in the Particle System



Figure 69. Comparing Entropy (SE) in the Particle System as Constraint Increases

5.6.2 Presence of Emergence in the Flocking System – H1

As with the particle system data, we make a similar analysis of the flocking system, comparing the nearly-random scenario to the emergence scenario. Table 25 summarizes the t-test results. Here we can see that  $H_01$  is rejected when  $I_C \leq 0.75$  but not for large values. This is not unexpected since the preliminary analysis suggested that when the actors in the flocking system were highly constrained the system appeared to exhibit random-like behavior.

$\hat{f}_{MC}$ =	=1.9272		
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)
0	0.0023	2.7876 4.7959	reject
0.08	4.8490e-006	3.3794 4.2698	reject
0.16	0.0062	2.5508 4.7422	reject
0.25	3.9435e-005	2.9513 3.8465	reject
0.33	4.8404e-004	2.7015 3.8490	reject
0.42	2.7555e-005	2.6668 3.2728	reject
0.5	0.0239	2.1500 4.3895	reject
0.58	0.0052	2.5747 4.6722	reject
0.67	0.0149	2.3022 4.5963	reject
0.75	0.0013	2.6897 4.1697	reject
0.83	0.0641	1.9015 2.6632	accept
0.92	0.3849	1.7110 2.4363	accept
1	0.2651	1.6745 2.0059	accept

Table 25. H1 Results for the Flow Metric in the Flocking System

Figure 70 is a box plot of the flow metric, CPL, for the flocking system as  $I_C$  increases. CPL for the nearly-random scenario is in column 14. Here it can be seen that the system becomes more random-like in a transition between columns 10 and 11, corresponding to  $I_C$  of 0.75 and 0.83. Figure 71 shows the value of the flow metric over 1000 time-steps at three different values of *max-align-turn*. Here the random-like behavior of the highly constrained case (bottom plot) is evident.



Figure 70. Comparing Flow (CPL) of the Flocking System as Constraint Increases



Figure 71. Comparing CPL for Varying Constraint in the Flocking System

# Table 26 and Figure 72 show the results of the statistical analysis for the

complexity metric, i.e., GCC. For the experiment here, the  $H_01$  for the GCC is rejected.

$\widehat{\Omega}_{_{M\!C}}$	=0.2525		
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)
0	3.2965e-012	0.6810 0.7228	reject
0.08	1.1680e-013	0.6931 0.7223	reject
0.16	1.3505e-012	0.6967 0.7357	reject
0.25	2.4269e-013	0.6896 0.7211	reject
0.33	2.9932e-013	0.6874 0.7195	reject
0.42	1.1418e-012	0.7046 0.7436	reject
0.5	5.3948e-010	0.6615 0.7348	reject
0.58	4.3183e-012	0.6710 0.7131	reject
0.67	2.7115e-010	0.6465 0.7115	reject
0.75	9.6625e-010	0.6163 0.6863	reject
0.83	1.7550e-009	0.6003 0.6724	reject
0.92	2.6812e-007	0.4465 0.5242	reject
1.0	8.9379e-005	0.3421 0.4336	reject

Table 26. H1 Results for the Complexity Metric in the Flocking System



Figure 72. Comparing Global Clustering Coefficient ( $\Omega$ ) as Constraint Increases

Figure 73 shows the progression of GCC over 1000 time-steps of the simulation. Similarly to the flow metric, we observe random-like behavior at higher constraint.



Figure 73. Comparing GCC at Varying Constraint in the Flocking System

Table 27 and Figure 74 present the results of the t-test for the organization metric SE. Column 14 of the box plot of Figure 74 shows results for the nearly-random scenario. For the experiment, the null-hypothesis is rejected.

$\widehat{S}_{MC} = 0.9452$							
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)				
0	1.0173e-004	0.6913 0.8212	reject				
0.08	5.1353e-005	0.7112 0.8232	reject				
0.16	2.9789e-004	0.6928 0.8364	reject				
0.25	1.9269e-005	0.7467 0.8331	reject				
0.33	1.3928e-006	0.7450 0.8123	reject				
0.42	1.1165e-005	0.7796 0.8479	reject				
0.5	8.0970e-004	0.7259 0.8638	reject				
0.58	2.6657e-004	0.7393 0.8552	reject				
0.67	3.5028e-004	0.7865 0.8782	reject				
0.75	3.3297e-006	0.7955 0.8503	reject				
0.83	4.7705e-009	0.8101 0.8358	reject				
0.92	3.2655e-010	0.7930 0.8150	reject				
1.0	5.6393e-011	0.7524 0.7755	reject				

Table 27. H1 Results for the Entropy Metric in the Flocking System



Figure 74. Comparing Entropy (S) of the Flocking System as max-align-turn Increases

5.6.3 Presence of Emergence in the Stigmergy System - H1

As has been discussed, the nature of the stigmergy system is such that when no pheromone can be deposited, no communication between actors can occur. In this situation, f = 0 and  $\Omega = 0$ . The results of the t-test on the stigmergy simulation for the flow and organization metrics are shown in Table 28 and Table 29 respectively.

$\hat{f}_{MC} = 0$					
Ic	p-value	95% confidence interval	H <sub>0</sub> (accept/reject)		
1.0	4.0604e-007	1.0666 1.5188	reject		
0.9	3.9655e-009	1.4763 1.8153	reject		
0.8	0	1.8354 1.8848	reject		
0.7	0	1.8714 1.9059	reject		
0.6	0	1.9014 1.9268	reject		
0.5	0	1.9239 1.9314	reject		
0.4	0	1.9371 1.9427	reject		
0.3	0	1.9402 1.9467	reject		
0.2	0	1.9518 1.9542	reject		
0.1	0	1.9513 1.9544	reject		

Table 28. H1 Results for the Flow Metric in the Stigmergy System

$\widehat{S}_{MC} = 0.9977$					
Ic	p-value	95% confidence inte	rval H <sub>0</sub> (accept/reject)		
1.0	9.6777e-011	0.4892 0.5538	reject		
0.9	3.1467e-011	0.5385 0.5903	reject		
0.8	8.3307e-011	0.5828 0.6347	reject		
0.7	7.1815e-011	0.6067 0.6549	reject		
0.6	9.3517e-012	0.6205 0.6579	reject		
0.5	1.7504e-011	0.6384 0.6765	reject		
0.4	1.9224e-010	0.6684 0.7134	reject		
0.3	2.8457e-010	0.6867 0.7310	reject		
0.2	2.1604e-012	0.7104 0.7349	reject		
0.1	2.8332e-010	0.6937 0.7369	reject		

Table 29. H1 Results for the Entropy Metric in the Stigmergy System

The plot shown in Figure 75 shows the change in CPL for a *pheromone* level of 70 during the first 600 time-steps of one simulation run. Figure 76 shows how the CPL changes as *pheromone* increases. The first approximate 100 time-steps correspond to the time it takes the ants to move from the nest to the food. The increasing value of *pheromone* supports increasingly larger patch structures. With smaller *pheromone* values, a frequency of increasing and decreasing CPL can be seen as in the top plot of Figure 76. Here, ants rapidly deplete their pheromone before new ants arrive and relationships are quickly established but then broken.



Figure 75. CPL Plot for the First 600 Time-steps at *pheromone* = 70



Figure 76. CPL in the Stigmergy System as pheromone Increases

Figure 77 shows how increasing  $I_C$  affects CPL.



Figure 77. Comparing Flow in the Stigmergy System as  $I_C$  Increases



Figure 78. Comparing Entropy of the Stigmergy System as  $I_C$  Increases

## 5.6.4 Discerning the Three Systems – H2

Recall from Section 0, that the  $H_02$  is that the three EBS in this study are indistinguishable by the measures. The alternative hypothesis  $H_12$  is that the metrics can distinguish them.

We look at the results of the three metrics across the three systems when they are in equilibrium and examine the three metrics for each of the three simulations using a one-way analysis of variance (ANOVA) where the null hypothesis is that all samples are drawn from populations with the same mean and thus indistinguishable.

The ANOVA results for the flow metric (CPL) is shown in Figure 79 with summary statistics in Table 30.



Figure 79. CPL for Each System in Emergence

Source	SS	df	MS	F
Columns	24.5015	2.0000	12.2507	16.9594
Error	19.5036	27.0000	0.7224	
Total	44.0050	29.0000		
p-value	<b>p-value</b> 1.6954e-005			
	Particles	Flocking	Stigmergy	
mean	1.8051	3.7918	1.9529	

Table 30. ANOVA Results for CPL

We can see that the F-statistic is larger than 1 and the p-value is very small, thus we reject  $H_02$  and surmise that the three systems are not indistinguishable by the CPL. However, the ANOVA test essentially says the three are not the same but we observe in Figure 79 that the mean for Stigmergy is within the upper quartile of the data for Particles. We examine just those two and obtain the plot shown in Figure 80. An ANOVA analysis here yields p = 0.306 and we conclude that the H<sub>0</sub>2 hypothesis is accepted, i.e., the CPL is not able to discern Particles from Stigmergy in this case.



Figure 80. CPL for Particles & Stigmergy.

Similar ANOVA analysis is performed for the complexity metric (GCC). The results are shown in Figure 81 and Table 31. Here we see that the F-statistic is very much larger than 1 and the p-value is 0, thus we reject  $H_02$  and conclude that the systems are distinguishable by the GCC.



Figure 81. GCC for Each System in Emergence

Source	SS	df	MS	F
Columns	4.577	2.0000	2.288	1691.6
Error	0.03653	27.0000	0.001353	
Total	4.613	29.0000		
p-value	0			
	Particles	Flocking	Stigmergy	
mean	0.9140	0.7019	0	

Table 31. ANOVA Results for GCC

Figure 82 and Table 32 present the results of the ANOVA analysis for the organization metric SE. We observe an F-statistic larger than 1 and a p-value that is very near 0 and thus reject  $H_02$ .



Figure 82. SE for Each System in Emergence

Source	SS	df	MS	F
Columns	0.5676	2.0000	0.2838	72.1104
Error	0.1063	27.0000	0.0039	
Total	0.6739	29.0000		
<b>p-value</b> 1.4805e-011				<u> </u>
	Particles	Flocking	Stigmergy	
mean	0.4461	0.7563	0.7153	

Table 32. ANOVA Results for SE

However, similar to the analysis between the Particle and Stigmergy with the CPL, here we see that mean of Stigmergy falls within the lower quartile of Flocking. Comparing just those two, we see the plot shown in Figure 83 we get a p = 0.1923 and must accept the null hypothesis, i.e., SE does not discern Flocking and Stigmergy.



Figure 83. SE for Flocking and Stigmergy
# CHAPTER 6

# CONCLUSIONS

This study presented a workable definition of emergence couched in information flow and relationships between entities. The definition emphasized the description of system components in a way that is useful to the modeler of systems that exhibit emergence phenomena. The ambiguities often associated with agent-based modeling terminology were reduced by treating ABM as a programming method and defining a language for EBS based on the concepts of entities, actors, objects, etc. These concepts readily translated to the implementation of some sample simulations that exhibit emergence. It offered a classification scheme (taxonomy) for EBS based on the type of information feedback predominant in a system The three simulations illustrated three system types according to the defined taxonomy, namely Type 1b (particles), Type 2a (flocking), and Type 3 (stigmergy). Beyond just illustrating the EBS type according to the taxonomy, each simulation demonstrated a particular interesting aspect of information flow. It is important to note that these are but three examples of many varied systems where emergence is commonly recognized and are in no way intended to be conclusive or complete with regard to representing emergent phenomena. The particle simulation was developed as an example of information flow that is direct between entities and where the information flow is always bidirectional. The flocking simulation demonstrated information flow that at times is directional and at other times bidirectional but where the behavior of each entity is not related to direct interactions as in the particles, but is based instead on each entity's sensing of those within its sensorium. The stigmergy simulation demonstrated a system where the flow of information between entities is intermediated

through entities comprising the environment. Whereas some may consider the flocking system as stigmergic as well, we observe a key distinction between the flocking system and the stigmergy system used here. In the flocking system, information is shared between similar actors. In the stigmergy system, information is shared between actors (ants) through an intermediary object (patches) different from the perceived actors. We believe that this is more in keeping with the definition of stigmergy offered by Grasse` and is somewhat a different perspective from some other works that would likely consider flocking simulations as stigmergic (Burbeck, 2004-2007).

Three metrics based on measures related to: 1) paths of information flow, 2) relative complexity, and 3) organization of the system, were explored within each of the simulated EBS. These metrics were developed around ideas based in graph theory and information entropy and the results suggest that emergence may be quantifiable in simulations of EBS.

By identifying the EBS as a distinct class of systems with unique modeling requirements, this research, through the definitions put forth, the interpretation and implementation of simulations based on those definitions, and the system metrics developed, points to a general method for measuring emergence in simulations of emergent behavior systems built from an entity-based perspective. Description ambiguities arising from too much dependence on agent formalisms are avoided by considering agents not as the fundamental components of an EBS but rather as an implementation paradigm for constructing the entities of the EBS. The approach demonstrated in this study allows the EBS modeler to describe the EBS in terms like actor, object, and structure, each with clearly definable roles.

#### 6.1 Suggested Near-Term Applications of this Work

Although emergent phenomena has been observed throughout history, advances in computing technology now make possible the simulation of systems formed of many interacting entities from which complex spatial and temporal patterns and behaviors emerge. Such Emergent Behavior Systems present unique challenges to the modeling and simulation community.

It is hoped that this research will stimulate further interest within the modeling and simulation community to exert leadership concerning EBS by 1) attempting to resolve the ambiguities and vagaries resulting from the heretofore unrecognized commonalities across varied problem domains, and 2) supporting a more generalized approach to the modeling of systems where emergence is of interest. In this way, it is hoped that through computer simulation we may come to a greater scientific understanding of the phenomena of emergence.

An initial step in this direction is offered here in the beginnings of an EBS Lexicon. Much remains to be done to better develop the working vernacular that will lead to axiomatic representation of emergent phenomena, but we believe this effort is a good start. Continuing efforts will no doubt produce axiomatic definitions based on ideas from statistical mechanics, complexity, and information sciences. As computing capabilities continue to increase rapidly, simulation seems poised to provide a virtual laboratory where arbitrary system parameters can be represented and measured; as such, modeling and simulation will serve as a kind of Petri dish to cultivate and study emergent phenomena. Progress in simulation science will provide the needed discipline and consistent forms of specification to achieve greater understanding of emergent phenomena and the modeling, analysis, and design of Emergent Behavior Systems.

## 6.2 Observations and Areas for Continued Research

This has been an arduous study. Much like the nature of emergence itself, the beginning was very broad and high-level, then diving deep into the representation of interactions between entities, and returning again to the surface to better understand the manifestation of phenomena of meso-level components at the macro-level. This study is foundational and provides a starting point for a great deal of further research. The following sections present some additional thoughts that occurred during the conduct of the study as well as some suggested departure points for additional research.

## 6.2.1 EBS Relationships to Small World Networks

In their study of natural networks, Newman, Watts, and Strogatz observed that large random graphs have Poisson degree distributions (Newman, et al., 2002). This suggests that graphs that are non-random are not likely to exhibit degree distributions that are Poisson. This would seem to be a valid premise for addressing the detection of emergence such as Hypothesis H1. The approach that was taken in this study considered the EBS if the factors that contribute to emergence were minimized. If the EBS is really just a random occurrence, then it should yield something close to a random graph. If it is a random graph, then it should be Poisson in its degree distribution. If it is not random, then it will be something other than Poisson.

The following figure shows plots of the degree distributions for each of the values of *shear-point* for the particle system EBS shown in Figure 51 of Section 0.



Figure 84. Plot of Degree Distributions for Figure 51

The Poisson distribution is defined as function

$$f(k,\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{42}$$

where

- e is the base of the natural log
- k is the number of occurrences of an event
- $\lambda$  is the expected value of k, i.e., the mean

Figure 85 plots the Poisson random data generated for k = 50 samples and a mean

 $\lambda$  equal to each of the average degrees of Figure 84.



Figure 85. Poisson Distributions Generated from the Mean Degrees of Figure 84

Just from this small sampling, it appears as if the minimally constrained EBS has the characteristics of a nearly-random graph. Figure 86 plots data from 100 simulation runs and compares that to the Poisson distribution.



Figure 86. Comparison of the Minimally Constrained Particle System to the Poisson Distribution

Upon inspection of the plots in Figure 86, for the three values of *shear-point* of the particle system shown, we can see that the nearly-random case is indeed similar to the Poisson distribution. This suggests that the nearly-random case approaches the

characteristics of a random graph. This would seem to support the supposition that a minimally constrained system should exhibit the characteristics of a random graph.

However, when this same approach is applied to the flocking system simulation, we see a different result with degree distribution that is not Poisson for the nearly-random case. Figure 87 shows the correlation of the mean degree between some samples of the minimally constrained case to the Poisson distribution of the same mean. The top row is the test distributions for vision equal to 5, 10, and 15 patches. The bottom row is the corresponding Poisson distribution with same mean as the samples. Shown at the bottom are the p-values for testing shown for the hypothesis of no correlation. Each p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero.



Figure 87. Correlation of Mean Degree Distribution of the Flocking System

Whereas the mean degree of the particle system seems very Poisson distributed when minimally constrained, the flocking system does not. This observation might be attributable to the fact that the particle system shares information in a bidirectional manner as opposed to the more mixed directional and bidirectional processes of the flocking system. It may be that certain types of EBS demonstrate non-Poisson characteristics. Continued study in this area might identify additional details to refine the EBS taxonomy and offer a kind of validation of models of certain systems, or metrics applicable to the engineering of certain complex systems.

# 6.2.2 The Information Flow Metric

One reason for the choice of the characteristic path length for the information flow metric used in this study was that the measure of the average path length scales with the number of vertices. We believe that it might be more meaningful, especially when comparing dissimilar systems (such as the H2 analysis) to normalize the metric such that the average path length for the system is compared to the graph diameter given by the Erdos-Renyi random graph diameter estimate,

$$D_{ER} \approx \frac{\ln(N)}{\ln(k)}$$

where N is the number of vertices and k is the average of the vertices' degrees. Like the average path length,  $D_{ER}$  scales with the number of vertices as well. The diameter estimate has some interesting qualities concerning random graphs (Barabasi & Albert, 2002):

- If k < 1 then the graph is composed of isolated trees
- If k > 1 then there is a giant cluster

• If  $k \ge ln(N)$  then the graph is fully connected

It might be possible to associate these characteristics to certain types or certain aspects of complex systems by this *normalized flow metric*.

It is envisioned that computing this metric would be as follows: As before, the average path length for the system is numerically determined by first generating the adjacency matrix for the system, and then successively raising the matrix in power either until there are no elements equal to zero or the  $n-1^{st}$  power has been reached. This results in a distance matrix from which the characteristic path length is computed as was done in this study. The *normalized flow metric* is then computed as

$$f = \frac{CPL}{D_{ER}}$$

where  $D_{ER}$  is computed using the values of N and k from the EBS.

Figure 88 shows the normalized flow metric applied to four sample graphs. Compare this to Figure 34 in Section 4.2.



Figure 88. The Normalized Flow Metric.

## 6.2.3 The Complexity Metric

In 1976, Thomas McCabe published a technique to assess the complexity of software programs (McCabe, 1976). McCabe observed that a software program could be represented by a directed graph where the vertices represented functional blocks of the program and the edges represented the transitions between blocks, i.e., branches to other blocks of code. Representing software code in this manner is referred to as the *program control graph*. He observed that the program control graph is a strongly connected, that is a path exists from any arbitrary vertex, to any other arbitrary vertex (we can also say that a vertex is *reachable* from any other vertex). McCabe's complexity relates to the number of control paths through a program and is similar to the clustering coefficient applied in this study. This is likely relatable to the ideas of local connectivity and

hierarchical linkages (Waters, 2006) and (Berge, 2001).

#### 6.2.4 The Entropy Metric

Although Parunak and Brueckner identified the Shannon Entropy of a multi-agent system as a measure of coordination of agents within a system (Parunak & Brueckner, 2001), recall that in general, entropy relates the number of microscopically defined states accessible to a system to the probability of finding the system in a specific state. The critical question is how to identify system states? Their approach considered a spatiotemporal system (specifically, robots on a 2D plane) and superimposes a grid. The number of agents occupying squares in the grid at any time distinguishes states of the system. The identification of system state then is the position of entities in the Euclidean space resulting in a location-based entropy. In this study, we sought a more general solution derived from a graph-theoretic approach. This is a reasonable approach in light of the axiom that interesting structure arises from the relationships established through the interaction of components: the formalism of vertices and edges provides an analytical means to represent system states as relationships amongst agents in a general sense, or at least in a sense that is invariant to physical geometry. Granted, this is not always appropriate, or at least may not be sufficient. For instance, if the intent is to drive a set of entities to a specific geometrical configuration, a Euclidean state space may be best. In the prosecution of this dissertation, both approaches were examined with comparable results especially in the nearly-random scenarios (as would be expected). However, the relation-space approach was very applicable to the flocking system, where the flock direction can change frequently and the flocks' positions change continuously. In that

case, the actual direction was not of interest – instead, the size and organization of flocks were. Similarly, the particle simulation when examined in the relation-space is invariant to structure position or orientation. This may be an advantage to some systems. However, if the positions of the structures within the context of observation are important, then the location –based entropy applies.

This could be particularly useful in the modeling and analysis of communication, logistic, and social systems, i.e., those where the structures of interest are dynamic and of high dimension. We expect that by looking at the relation-space as system state, certain relationships will be manifested at that level of abstraction that might otherwise be unrecognized with more traditional Euclidean organization measurements. In the analyses of this dissertation, where relation-space was considered, it consisted of the number of structures in the system and the normalized mean degree. This could have just as well been the characteristic path length or any other graph metric. More work needs to be done to determine what features should describe system states for certain domains. It may be that certain classes of systems are best described by specific features.

# 6.3 Other Considerations

Equation (21) in Section 3.7 expresses intensity of constraint as a function of time. The metrics developed in this study looked at the systems when they were organizationally stable. We believe that a great deal of insight to EBS could be obtained if the trends of these metrics were analyzed over time. It might be that certain patterns in the metrics could be indicative of nascent or eminent behaviors. The CPL of the Stigmergy system shown in the top plot of Figure 76 shows a periodicity related to the depletion of ant pheromone that results in a physical gap between the trail to the food and

the anthill. Other, more useful systems might demonstrate similar characteristics.

The computational requirements of systems comprised of many entities increases as  $N^2$ . Although hundreds of entities do not pose a particular problem to today's even modest computers, many systems of interest could easily result is many thousands of entities. As an example, the Stigmergy system developed here considers both ants and patches as entities. Using a context consisting of a 50 x 50 grid of patches and 100 ants produces an adjacency matrix that is 2600 x 2600 (6,760,000 elements). Although easily within the abilities of high-performance parallel processors, such contexts often produce sparse matrices. That was included in the rationale of the super-node approach in this study. Still, mixed entities with very large computational spaces are likely to become of more interest. The observation here of the super-node, i.e., the coalescing of the many patches in relationship to a single structure may be indicative of a special characteristic of such systems. We suggest that continued research into the nature of the emergence phenomena may reveal computational shortcuts, like the super-node, that can reduce the computational demand and be revealing of causal mechanisms in other systems.

It is hoped that this research would offer some steps toward reconciling the systems dynamics and entity-based perspectives by establishing metamodels that allow the top-down perspective to measure or regulate the bottom-up perspective. Starting with the fundamental case of a collection of homogeneous agents, a question that stands to be examined is that of emergent progression, i.e., must systems comprised of such simple interacting entities progress through a sequence of distinct behavioral classes like that shown in Figure 89 (if they progress at all)? If so, can the onset of emergence be detected and can the patterns of emergence be predicted? The results of this research may

contribute to the formal specification of certain complex systems and the validation of the designs of those systems.

Zeroth Order	First Order	Second Order	Third Order	Fourth Order	
Components in Isolation	No global to local feedback	Global to local feedback	Combined feedback	Evolutionary	-
T0=	$\rightarrow^{T1a}_{T1b}$		<b>→</b> T3=	<b>-⇒</b> T4	
	Unchangin	ig Component Pro	perties	Changing Co	mponent Properties

Figure 89. Progression of Emergence

# REFERENCES

Abbott, R., 2006. Emergence Explained: Abstractions. Complexity 12,1.

Ablowitz, R., 1939. The Theory of Emergence. Philosophy of Science, Issue 6, pp. 1-16.

Anon., 2011. Japan's K Computer Tops 10 Petaflop/s to Stay Atop TOP500 List. [Online] Available at: http://www.top500.org/lists/2011/11/press-release

Anon., n.d. s.l.:s.n.

Ashby, W. R., 1956. An Introduction to Cybernetics. s.l.: Chapman & Hall.

Ashlock, D. & Kim, E.-Y., 2005. The Impact of Cellular Representations on Finite State Agents for Prisoner's Dilemma. Washington, DC, s.n.

Baas, N. A. & Emmeche, C., 1997. On Emergence and Explanation. Intellectica 2, pp. 25: 67-83.

Bak, P., 1996. How Nature Works: The Science of Self-Organized Critciallity. New York: Copernicus.

Barabasi, A.-L. & Albert, R., 2002. Statistical Mechanics of Complex Networks. Reviews of Modern Physics 74, pp. 47-97.

Baranger, M., 2001. Chaos, Complexity, and Entropy: a Physics Talk for Non-Physicists. [Online]

Available at: http://necsi.edu/projects/baranger/cce.pdf

Baranger, M., 2008. Chaos, Complexity, and Entropy: A Physics Talk for Non-physicists. [Online] Available at: http://necsi.org/projects/baranger/cce.html [Accessed 2008].

Barton, A., 2005. NRC/ERB-1122: Modelling the Foraging Patterns of a Colony of Cooperating Army ant Agents Using NetLogo, s.l.: NRC.

Bar-Yam, Y., 2004. A Mathematical Theory of Strong Emergence Using Multiscale Variety. Complexity 9, pp. 37-45.

Bar-Yam, Y., 2004. Mulitiscale Variety in Complex Systems. Complexity 9, pp. 15-24.

Bar-Yam, Y., 2005. About Engineering Complex Systems: Multiscale Analysis and Evolutinoary Engineering. In: Engineering Self-Organizing Systems. Berlin: Springer-Verlag, pp. 16-31.

Beckers, R., Holland, O. E. & Deneubourg, J. L., 1994. From Local Actions to Global Tasks: Stigmergy and Collective Robotics. Artificial Life IV, pp. 181-189.

Berge, C., 2001. The Theory of Graphs. s.l.:Dover Publications.

Boccara, N., 2004. Modeling Complex Systems. s.l.:Springer-Verlag.

Bolllobas, B., 1985. Random Graphs. New York: Academic Press.

Bonabeau, E., Dessalles, J. & Grumbach, A., 1995. Characterizing emergent phenomena (2): a conceptual framework. *Revue Internationale de Systemique*, pp. 347-371.

Bonabeau, E., Dessalles, J. & Grumbach, A., 1995. Characterizing emergent phenomena: a critical review (1). *Revue Internationale de Systemique 9(3)*, pp. 327-346.

Boschetti, M., Prokopenko, I., Macreadie, A. & Grisogono, M., 2005. Defining and detecting emergence in complex networks. s.l., s.n.

Bradshaw, J., 1997. Introduction to Software Agents. In: Software Agents. s.l.:AAAI Press / The MIT Press.

Brooks, R. A., 1991. Intelligence Without Representation. *Artificial Intelligence Journal* (47), pp. 139-159.

Burbeck, S., 2004-2007. Complexity and the Evolution of Computing: Biological Principles for Managing Evolving Systems. [Online] Available at: <u>http://www.evolutionofcomputing.org/Complexity and Evolution of Computing v2.pdf</u> [Accessed 2012].

Cao, Y., Fukunaga, A. & Kahng, A., 1997. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots* 4(1), pp. 7-23.

Carley, K., 2002. Computational organization science: A new frontier. 14 May, Volume 99.

Chiang, C., Hoffman, C. & Mittal, S., 2008. Emergent Crowd Behavior. Computer-Aided Design and Applications Vol. 6 Num 6., pp. 865-875.

Chira, C., Pintea, D. & Dumitrescu, D., 2007. Stigmergic Collaborative Agents. Journal of Universaly Computer Science, vol. 13, no. 7, pp. 922-932.

Chu, D., 2008. Criteria for Conceptual and Operational Notions of Complexity. Artificial Life, Summer 2008, Vol. 14, No. 3, pp. 313-323.

Corning, P. A., 2002. Emergence: A Venerable Concept in Search of a Theory. *Complexity* 7(6), pp. 18-30.

Corning, P. A., 2002. The Re-emergence of "Emergence": A Venerable Concept in Search of a Theory. *Complexity*, 7(6).

Crutchfield, J., 1994. The Calculi of Emergence: Computation, Dynamics, and Induction. *Physica D*, pp. 11-54.

De Wolf, T., Holvoet, T. & Berbers, Y., 2004. Emergence as a paradigm to engineer distributed autonomic software, Belgium: s.n.

Deisboeck, T. & Kresh, Y. J., 2006. Complex Systems Science in Biomedicine. New York: Springer Medical Sciences.

Devore, J. L., 2008. Probability and Statistics for Engineering and the Sciences, 7th Edition. Belmont, CA: Thomson-Brooks/Cole.

Diestel, R., 2005. Graph Theory, Third Edition.. New York: Springer-Verlag.

Dorigo, M., Di Caro, G. & Gambardella, L., 1999. Ant Algorithms for Discrete Optimization. *Artificial Life, Vol. 5, No. 2.* 

Erdos, P. & Renyi, A., 1959. On random graphs, I. Publ. Math. Debrecen 6, pp. 290-297.

Erdos, P. & Renyi, A., 1960. On the evolution of random graphs. *Magyar Tud. Adak. Mat. Kutato Int. Kozl. 5*, pp. 17-61.

Ferber, J., 1999. Multi-Agent Systems, An Intorduction to Distributed Artificial Intelligence. s.l.:Addison-Wesley Longman.

Franklin, S. & Graesser, A., 1996. Is it an Agent or Just a Program? A Taxonomy for Autonomous Agents. s.l., s.n., pp. 21-35.

Freitas, R. A., 2005. Nanotechnology, nanomedicine and nanosurgery. *International Journal of Surgery*.

Fromm, J., 2005. Types and Forms of Emergence. Complexity Digest 25.

Grasse', P.-. P., 1959. La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs.. *Ins. Soc.*, *6*, pp. 41-83.

Guerin, S. & Kunkle, D., 2004. Emergence of Constraint in Self-organizing Systems. Nonlinear Dynamics, Psychology, and Life Sciences, Vol 8. No. 2.

Hamdi, M., Sharma, G., Ferreira, A. & Mavroidis, C., 2005. Prototyping Bio-Nanorobots Using Molecular Dynamic Simulation. Paris, s.n.

Harel, D., 1987. Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming, Issue 8, pp. 231-274.

Hawick, K. A., Scogings, C. J. & James, H. A., 2004. Defensive Spiral Emergence in a Predator-Prey Model. Cairns, Australia, s.n.

Holland, J. H., 1995. Hidden Order: How Adaptation Builds Complexity. s.l.:Addison-Wesley.

Holland, J. H., 1998. Emergence: From Chaos to Order. s.l.:Addison-Wesley.

Hubler, A. W., 2005. Predicting complex systems with a holistic approach: The throughput criterion. *Complexity, Vol. 10, Issue 3*, 9 February, pp. 11-16.

Hudlicka, E. G. & Zacharias, G., 2004. Approaches for Modeling Individuals within Organizational Simulations. s.l., s.n.

Jennings, N. R. & Wooldridge, M., 2000. Agent-Oriented Software Engineering. Proceedings of the 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99).

Jennings, N. R. & Wooldridge, M. J., 1998. Applications of Intelligent Agents. In: N. R. Jennings & M. J. Wooldridge, eds. *Agent Technology Foundations, Applications, and Markets*. s.l.:Springer-Verlag.

Kassner, M., 2009. *Swarm Intelligence: Are digital ants the answer to malware?*. [Online] Available at: <u>http://blogs.techrepublic.com.com/security/?p=2757</u> [Accessed 3 December 2009].

Kim, M. et al., 1999. Agent-Oriented Software Modeling. s.l., s.n., pp. 318-325.

Kowaliw, T., 2008. Measures of Complexity for Artificial Embryogeny. s.l., s.n., pp. 843-850.

Leon, D., 2005. Reduction, Emergence, And Downward Causation In International Relations Theory. Washington, D.C., s.n.

Lorenz, E. N., 1963. Deterministic Nonperiodic Flow. *Journal of Atmospheric Science*, Issue 20, pp. 130-141.

Lyons, D. M. & Arkin, R. C., 2003. Towards Performance Guarantees for Emergent Behavior. s.l., s.n.

Macal, C. M. & North, M. J., 2005. *Tutorial on Agent-based Modeling and Simulation*. s.l., M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds..

Magedanz, T., Rothermel, K. & Krause, S., 1996. Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?. San Francisco, s.n.

McCabe, T. J., 1976. A Complexity Measure. *IEEE Transactions On Software Engineering, Vol SE-2, No. 4*, December.

McDonald, J. H., 2009. *Handbook of Biological Statistics, 2nd Edition*. Baltimore, MD: Sparky House Publishing.

Mittal, S., 2012. Emergence in Stigmergic and Complex Adaptive Systems: A Formal Discrete Event Systems Perspective. *Journal of Cognitive Science Research, Special Issue on Stigmergy*.

Mizuta, H. & Nakamura, F., 2005. Agent-Based Simulation of Enterprise Communication Network. s.l., s.n.

Morgan, L. C., 1923. Emergent Evolution. London: Williams and Norgate.

Newman, M. E. J., Watts, D. J. & Strogatz, S. H., 2002. Random Graph Models of Social Networks. s.l., s.n., pp. 2566-2572.

Newman, M., Strogatz, S. & Watts, J., 2001. Ranom graphs with arbitrary degree distributions and their applications. *Physical Review E 64*.

Nicolescu, M. N. & Mataric, M. J., 2002. *A hierarchial architecture for behavior-based robots.* s.l., ACM Press.

Nwana, H. S., 1996. Software Agents: An Overview. *The Knowledge Engineering Review*, 11 (3), pp. 205-244..

Nwana, H. S. & Ndumu, D. T., 1999. A Perspective on Software Agents Research. *The Knowledge Engineering Review*, 14(2), pp. 1-18.

Oddershede, L. et al., 2002. The Motion of a Single Molecule, the  $\lambda$ -Recepter, in the Bacterial Outer Membrane. *Biophysical Journal*, Vol. 83, pp. 3152-3161.

Parunak, H., Brueckner, S., Sauter, J. & Posdamer, J., 2001. *Mechanisms and Military Applications for Synthetic Pheromones*. Montreal, ERIM, pp. 58-67.

Parunak, H., Savit, R. & Riolo, R., 1998. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and User's Guide.. s.l., Springer, LNAI 1534, pp. 10-25.

Parunak, H. V. D., 1997. "go to the ant": Engineering principles from natural agent systems. *Annals of Operations Research*, 75, pp. 69-101.

Parunak, H. V. D. & Brueckner, S., 2001. Entropy and Self-Organization in Multi-Agent Systems. s.l., s.n.

Parunak, H. V. D. & Brueckner, S., 2001. Entropy and Self-Organization in Multi-Agent Systems. Montreal, Quebec, Canada, ACM, pp. 124 - 130.

Prokopenko, M., Mahendra, P. & Wang, P., 2005. On Convergence of Dynamic Cluster Formation in Multi-Agent Networks. Canterbury, UK, s.n.

Prokopenko, M., Wang, P. & Price, D., 2005. Complexity Metrics for Self-monitoring Impact Sensing Networks. Washington D.C., s.n.

Reynolds, C., 1987. Flocks, Herds, and Schools: A Distributed Behavior Model.. s.l., s.n., pp. 25-34.

Reynolds, C., 1999. Steering Behaviors for Autonomous Characters. s.l., s.n., pp. 763-782.

Rissanen, J., 1989. Stochastic Complexity in Statistical Inquiry. Singapore: World Scientific.

Rouff, C. et al., 2004. Verification of Emergent Behaviors in Swarm-based Systems. s.l., s.n., pp. 443 - 448.

Russell, S. & Norvig, P., 1995. Artificial Intelligence: A Modern Approach. s.l.:Prentice-Hall, Inc..

Schieritz, N. & Größler, A., 2003. Emergent Structures in Supply Chains - A Study Integrating Agent-Based and System Dynamics Modeling. s.l., s.n.

Schieritz, N. & Milling, P. M., 2003. Modeling the Forest or Modeling the Trees. Comparison of SD and AB Simulation. s.l., s.n.

Schieritz, N. & Milling, P. M., 2003. Modeling the Forest or Modeling the Trees: A Comparison of System Dynamics and Agent-Based Simulation. New York, s.n.

Scholl, H., 2001. Agent Based and System Dynamics Modeling: A Call for Cross Study and Joint Research. Maui, s.n.

Shalizi, C., 2003. Methods and Techniques of Complex Systems Science: An Overview. In: *Complex Systems Science in Biomedicine*. s.l.:s.n., pp. 33-114.

Shalizi, C. R., 2001. Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata. Madison: The University of Wisconsin.

Shannon, C. E., 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*.

Shell, D. A., Jones, C. V. & Mataric, M. J., 2005. Ergodic Dynamics by Design: A Route to Predictable Multi-Robot Systems. Washington, DC, Naval Research Laboratory, pp. 291 - 297.

Shmulevich, I., Dougherty, E. R. & Zhang, W., 2002. From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks. *Proceedings of the IEEE*, November.90(11).

Sklar, E., 2007. Software Review: NetLogo, a Multi-agent Simulation Environment. *Artificial Life 13*, pp. 303-311.

Sklar, E. & Davies, M., 2005. *Multiagent simulation of learning environments*. New York, ACM Press, pp. 953-959.

Steinman, J., Lammers, C., Valinski, M. & Steinman, W., 2010. Scalable Publish and Subscribe Data Distribution. s.l., s.n.

The MathWorks, Inc., 1984-2011. *MATLAB - The Language of Technical Computing*. [Online] Available at: <u>http://www.mathworks.com/products/matlab/index.html</u> [Accessed September 2011].

Theraulaz, G. & Bonabeau, E., 1999. A Brief History of Stigmergy. Artificial Life 5, pp. 97-116.

Theraulaz, G. & Bonabeau, E., 1999. A Brief History of Stigmergy. Artificial Life 5, pp. 97-116.

Tosic, P., 2006. Distributed Coalition Formation for Collaborative Large-Scale Multi-Agent Systems, s.l.: Univ. of Illinois at Urbana-Champaign.

Turing, A. M., 1950. Computing machinery and intelligence. *Mind*, Issue 59, pp. 433-460.

U. S. Marine Corps, 1997. *MCDP 1, Warfighting, PCN 142 000006 00, Foreword by C. C. Krulak,* Washington, DC: Department of the Navy, Headquarters United States Marine Corps.

van den Bergh, S., 1995. Ages of the Oldest Clusters and teh Age of the Universe. *Science*, 22 December, pp. 1942-1943.

Varenne, F., 2009. Models and Simulations in the Historical Emergence of the Science of Complexity. In: *From System Complexity to Emergent Properties*. Berlin: Springer-Verlag, pp. 3-21.

Vicsek, T., 2002. Complexity: The Bigger Picture. Nature, 11 July, p. 131.

Waters, N., 2006. Network and nodal indicies - Measures of complexity and redundancy: A review. In: *Spatial Dynamics, Networks and Modelling*. Cheltenham, UK & Northampton, MA: Edward Elgar, pp. 13-33.

Watts, D. & Strogatz, S., 1998. Collective dynamics of 'small-world' networks.. *Nature*, pp. 409-10..

Wilensky, U., 1999. *NetLogo*. [Online] Available at: <u>http://ccl.northwestern.edu/netlogo/</u> [Accessed September 2011].

Wilensky, U., 1999. *NetLogo*. Evanston(IL): Center for Connected Learning and Computer-Based Modeling, Northwestern University.

Wolpert, D. H., Wheeler, K. R. & Turner, K., 2000. Collective Intelligence for Control of Distributed Dynamical Systems. *Europhysics Letters, Vol. 49, No. 6*, March.

Wong, Y. H., 2006. Ignoring the Innocent: Non-combatants in Urban Operations and in Military Models and Simulations, s.l.: Pardee RAND Graduate School.

# **APPENDICES**

# APPENDIX A: GLOSSARY

This glossary provides a quick definition of some of the terms in this dissertation. Some terms, particularly those that are fundamentally related to the dissertation, are more technically defined elsewhere in the dissertation text.

Actor:	In an EBS an <i>Actor</i> is an <i>Entity</i> comprising an autonomous stochastic dynamical system that attempts to build and maintain a maximally-predictive internal model of its <i>Environment</i> within the context of its <i>Sensorium</i> , behavior sets, and <i>Effectors</i> . See Section 3.8.
Agent:	an autonomous stochastic dynamical system that attempts to build and maintain a maximally-predictive internal model of its environment within the context of its inherent sensors, behavior sets, and effectors. See Section 2.5.
Communication:	the transference of data. See Section 2.5
Complexity:	In EBS, this is related to the selection of responses available to an entity based on information it can receive and is measure of the an entity's influence on the system arising from sustained relationships with other entities. Compare to Sophistication. See Section 3.1.
Complex System:	a system of many mutually interacting dynamical parts which are coupled in a nonlinear fashion. Such a system may be discrete (such as a cellular automata system or set of difference equations), or it may be continuous as in a system of differential equations. Because they are nonlinear, complex systems are more than the sum of their parts because a linear system is subject to the principle of superposition, and hence is literally the sum of its parts, while a nonlinear system is not. Most biological systems are complex systems, while most traditionally engineered systems are not. Many research disciplines are becoming interested in this branch of mathematical analysis because the digital computer has made theoretical exploration of such systems possible.
EBS:	Emergent Behavior System. A system that achieves its objectives primarily through the (often dynamic) interaction of a multitude of simple parts (or entities) through which a higher-order behavior arises. Such systems are often characterized by the fact that units used to describe the system are unrelated to the units used to described the components of the system. See Section 2.3.

Emergence:	a complex system phenomena in which patterns that are observed at a global level arise solely from interactions among lower-level components acting on rules which are executed using only local information without reference to the global pattern.
Emergent:	a new quality of existence which results from the structural relation of its components parts. See Section 2.1.
Entity:	Any Actor or Object within the Context of Observation. See Section 3.8.
Environment:	The collection of Entities within an EBS Context of Observation. See Section 3.8.
Information:	in an Emergent Behavior System, any actionable data that facilitates a change of state in an Entity. See Section 2.5.
Random graph:	a graph in which the vertices and edges form connecting pairs at random. See Section 4.2 as well as (Bolllobas, 1985).
Resultant:	a property of the combination that can be foretold exhaustively from the individual elements. See Section 2.1.
Sensorium:	The totality of those parts of an Actor that receive, process and interpret sensory stimuli.
Sophistication:	A form of complexity associated with the micro-scale of an entity and measurable independently of the context of that entity. See Section 3.1 and Appendix C.
Stigmergy:	communication that occurs indirectly between entities through the modification of their local environment. See Sections 3.4.4 and 5.1.3.
Variety:	The degrees of freedom available to an entity in an EBS. Also, Law of Requisite Variety refers to degrees of freedom possible in a system to the degrees of freedom available to a ssytem. See Section 3.4.

# APPENDIX B: MATLAB CODE & SCRIPTS

Name	Туре	Purpose	
LBE	MATLAB Function	EBS Shannon Entropy Measure	
Convention: [SE,N,BX,BY] = LBE(XC, YC, Csize, dim) Description: LBE is used to compute the Location Based Entropy of the dataset produced by <i>Hookean Springs</i> , <i>OfaFeather</i> , and <i>Ants</i> or other agent-based models with data formats processable by PREMESH. LBE uses the position of all actors (A) in a 2D context of observation (CO). LBE invokes a 2D histogram (see hist2d) with a square grid of bins of order <i>d</i> . Return Values: N is the dim x dim x (# of observations) matrix giving the state of the system at each observation. Example; N(: , : , 1) is the state matrix for the first observation. Related Functions: SHANNON,			
<pre>function [SE,N,BX,BY] = % [SE,N,BX,BY] = LBE(XC %</pre>	LBE(XC, YC, Csize, dim) , YC, Csize, dim)		
<pre>%Location-based Entropy computation. %Convention: [SE,N,BX,BY] = LBE(XC, YC, Csize, dim) %Description: LBE is used to compute the Location Based Entropy of the dataset %produced by Hookean Springs, OfaFeather, and Ants or other agent-based</pre>			
<pre>%with data formats proc actors (A) %in a 2D context of obs biot2d) with</pre>	<pre>models %with data formats processable by PREMESH. LBE uses the position of all actors (A) %in a 2D context of observation (CO). LBE invokes a 2D histogram (see</pre>		
hist2d) with %a square grid of bins of order d. %Return Values: SE is the Shannon Entropy. %N is the dim x dim x (# of observations) matrix giving the state of the system at			
<pre>%each observation. Exam first observation. %</pre>	ple; N(: , : , l) is the	state matrix for the	
<pre>%XC and YC are rxc matrices where r is the number of observations %and c is the number of particles in the system. %XC is the matrix containing the x-coordinates and YC the y-coordinates %corresponding to each particle's position. Note that XC and YC must be of %the same dimension.</pre>			
<pre>%Che same dimension. %Csize is the dimension %dim is a scalar which %measure. %SE is a column vector</pre>	of the Csize x Csize con defines the dim x dim gri where each row is the ent	text of observation. d resolution of the cropy measure for each	
<pre>%observation. %The Context of Observa partioned</pre>	tion is assumed to be a p	positive valued	

%space with (0,0) being the lower left corner and (maxXp,maxYp) being the Supper right corner. %Return Values: [SE, N, BX, BY] %SE is the computed entropy. %N is the dim x dim x length(XC) state matrix of the system. There are as %many Ns as there are observations (length(XC)). %BX and BY are the centers of the X and Y bins. %Example: SE = LBE(XC,YC,10,2) will return the Shannon Entropy measure %based on the particle positions as described by XC and YC in a 10 x 10 Scontext of observation and summed in a 2 x 2 grid of bins. % >>> Should check that XC and YC are same dimension <<<</pre> % If dimensions of XC and YC are not identical return on error. [rx,cx]=size(XC); [ry,cy]=size(YC); if  $(rx \sim = ry)$ disp('Error: XC and YC must be of same dimensions: row agreement.') disp('.') return else if  $(cx \sim = cy)$ disp('Error: XC and YC must be of same dimensions: col agreement.') return end end % >>> Context of Observation is divided into bins based on DIM <<<</p> % Find bin centers BX and BY. bcc = Csize/dim; BXc = zeros(1,dim); %preallocate the matrix BXc(1) = bcc/2;% This is the first bin's center. for m = 2:dimBXc(m) = BXc(m-1)+bcc;end % Since this is a square gridding BY and BX are the same. BYC = BXC;tic % Nitt = flipud(hist2d(XC(1,:),YC(1,:),BXc,BXc)); % for itt = 2:length(XC) Nitt = cat(3,Nitt,flipud(hist2d(XC(itt,:),YC(itt,:),BXc,BXc))); 8 % end Nitt=zeros(dim,dim,length(XC)); % Preallocate array length(XC) for itt = 1:length(XC) % The next line of code is used when gridding the entire space and you % want to specify the exact bin centers, i.e., macro-scale system states.

```
%Nitt(:,:,itt) = flipud(hist2d(XC(itt,:),YC(itt,:),BXc,BXc));
% The next line of code is used when gridding only the space demarked
by
% maximum positions of the entities in the system. Makes the measure
% idependent of scale; i.e., meso-scale system states.
     Nitt(:,:,itt) = flipud(hist2d(XC(itt,:),YC(itt,:),dim));
end
toc
if nargout == 0
   imagesc(Nitt(:,:,1))
else
   N = Nitt;
   BX = BXc;
   BY = BYc;
   SE = shannon(Nitt);
end
toc
```

Name	Туре	Purpose
STATECOUNT	MATLAB Function	Count unique states

#### Convention: [ST,OC,OB] = STATECOUNT(N)

Description: Examines all states of the system, comparing each to each other to count how many unique states are in the system and how many times they occur. N is the dim x dim observations of the observation matrix.

```
Return Values: ST is the (dim:dim:#ofstates) matrix of all the unique states. OC is the vector of counts of each unique state. OB is the original number of samples examined.
```

```
function [st,oc,ob] = statecount(N)
[ST, OC, OB] = STATECOUNT(N)
웅
%STATECOUNT examines all states of the system, comparing each to
each
Sother to count how many unique states are in the system and how
many times
%they occur. N is the dim x dim x observations observation
matrix.
8
%Example: [ST,OC,OB] = STATECOUNT(N) will return (ST) the unique
states in N
%a count of their occurance (OC), and the total number of
observations (OB)
õ
%Example: C = STATECOUNT(N) will return the number of unique
states in N.
% >>> Should check that N is a valid Observation Matrix <<<</pre>
[r,c,dim]=size(N);
if (r \sim = c)
    disp('Error: Invalid observation matrix, r & c must be of
same dimension.')
    disp('.')
    return
else
    if (\dim == 1)
        disp('Warning: Only one observation.')
    end
end
octmp = 0;
stc = 1;
st = N(:,:,1); %The first state is the first unique state
oc(1) = 0;
                %keep building this list and compare all the rest
[dr,dc,dd]=size(st); %to it and count them.
```

```
for m=1:dim
    [dr,dc,dd]=size(st);
    octmp=zeros(dd,1);
    for n=1:dd
        if isequal(N(:,:,m),st(:,:,n))
            octmp(n)=octmp(n)+1;
            oc(n)=oc(n)+1;
        end
    end
    end
    if sum(octmp)== 0
        stc = stc+1;
        st(:,:,stc)=N(:,:,m);
        oc(stc)=1;
    end
end
end
```

end

end

Name	Туре	Purpose
STATECOMPARE	MATLAB Function	Count a specific state

Convention: STC = STATECOMPARE(ST1,ST2) Description: Compares the dim x dim ST2 to ST1. Return Values: STC(n) is the vector where each element is the count of ST1(n) that occurred in ST2.

```
function stc=statecompare(st1, st2)
%STC=STATECOMPARE(ST1,ST2)
å
%STATECOMPARE Compares the dim x dim x m state matrix ST2 to the
dim x dim
%x n statematrix ST1.
%STC is the n x l vector where each element is the count of
ST1(n) that is
%found in ST2. The index of STC is the matching index of ST1.
8
% >>> check that the rows and columns of ST1 and ST2 match
[rst1, cst1, dst1] = size(st1);
[rst2,cst2,dst2]=size(st2);
if (rst1 ~= rst2)
    disp('Error: State matrix row length mismatch.')
    return
end
if (cst1 \sim = cst2)
    disp('Error: State matrix column length mismatch.')
    return
end
stc=zeros(dst1,1); %preallocate stc
for m=1:dst1
    for n=1:dst2
        if isequal(st1(:,:,m),st2(:,:,n))
            stc(m) = stc(m) + 1;
        end
    end
```

Name	Туре	Purpose
FINDSTATE	MATLAB Function	Find a specific state

Convention: FS = FINDSTATE(STATE,STATELIST) Description: Used to search a state list for a specific state Return Values: returns the index in the list of states STATELIST of the state STATE .

```
function fs = findstate(state,statelist)
% FS = FINDSTATE(STATE,STATELIST)
% used to search a state list for a specific state
% returns the index in the list of unique states STATELIST of the
state STATE
% If the state STATE is not found in the list STATELIST a 0 is
returned.
```

```
ş
[rl,cl,dl]=size(statelist);
[rs,cs,ds]=size(state);
if (rl \sim = rs) or (cl \sim = cs)
    display ('Error: the rows and columns of the sample state
must match that of the state list')
    return
end
for n = 1:dl
    if isequal (state, statelist(:,:,n))
        fs = n;
        return
    else
        fs = 0;
    end
end
```

Name	Туре	Purpose
PROBN	MATLAB Function	Computes the probability of
		finding a particular state

Convention:PN = PROBN(N,OC,STATELIST)Description:used to compute the probability of the occurrence of state PN. OC is the<br/>occurrence vector corresponding the states in STATELISTReturn Values:returns the probability of finding state N(: , : , n) in the set of N .

```
function pn = probn(N,OC,statelist)
```

```
% PN = PROBN(N,OC,STATELIST)
% Description: used to compute the probability of the occurrence
of state PN.
% OC is the state occurrence vector corresponding the states in
STATELIST
% Return Values: returns the probability of finding state N(: , :
, n) in
\% the set of N .
웅
[rl,cl,dl]=size(statelist);
[rs,cs,ds]=size(N);
if (rl \sim = rs) or (cl \sim = cs)
    display ('Error: the rows and columns of the sample state
must match that of the state list')
    return
end
pn = zeros(1,ds); % preallocate the array
for n = 1:ds
    pn(n) = OC(findstate(N(:,:,n),statelist))/ds;
end
end
```

The following are Graph Theoretic related functions

Name	Туре	Purpose
MAKE_ADJACENCY_LISTS	MATLAB Function	Extracts adjacency list
		from a data file

Convention: [ADJLIST]=MAKE\_ADJACENCY\_LIST(DATAFILE, RANGE) Description: Reads a DATAFILE of the type produced by NetLogo EBS simulations.

Return Values: ADJLISTS, a cell array (TICK x ENTITY x RUN)

Related functions: depends on GET\_NETLOGO\_ADJLIST

```
function [AdjLists]=make adjacency lists(Datafile, Range)
% [ADJLISTS] = MAKE ADJACENCY LISTS(DATAFILE, RANGE)
% Datafile = string corresponding to a filename
% like Datafile = 'C:\WORK\Emergent Intelligent
Behavior\Dissertation\Experiments\Particle
System\Particles 1007301 CE-Random-table.csv'
% Range = the range of ticks to extract within Datafile
% Returns a cell array ADJLISTS(TICK x ENTITY x RUN)
% Datafile is a csv table of text Datafile of the type produced
by NetLogo's BehaviorSpace
% tool. Where the data is in the form of some header information
% "rnum", "Factor 1", "Factor 2",... "Factor n", "step", "reporter
data", "link-list"
% where "link-list" is a string of the form [[0 ...] [1 ...] ---
[m ...]]
è
% Related Functions: depends on GET NETLOGO ADJLIST
옹
8
응
[AL]=get netlogo adjlist(Datafile,1);
[r,c,enum]=size(AL); %don't care about enum here.
enum = Range; %This is the range of the enumerations. Set enum
to 1:enum to get all ticks
%clear AdjList
AdjList = cell(r,c,length(enum)); %Preallocate the cell array
            %Get each adjacency list in the range
ndx = 1;
for n=enum
              %Read the adjacency lists for the range of
enumerations
    [AdjLists(:,:, ndx)]=get netlogo adjlist(Datafile,n);
   ndx = ndx+1;
end
```

Name	Туре	Purpose
FINDSTRUCT	MATLAB Function	Finds structures for a given
		time-step

Convention: STRUCTURES=FINDSTRUCT(ADJLIST)

Description: Identifies all structures in a CO at a given observation (tick). ADJLIST is a 1xN cell array that is the adjacency list of the system at an observation, e.g., AdjList(3000,:) is the adjacency list for the system at observation 3000.

Return Values: SCOUNT is the number of structures in the system STRUCTURES is a cell array of adjacency lists for each structure at an observation.

Related functions: GETADJLIST, LIST2MAT

```
function [scount, Structures] = findstruct(AdjList)
%AdjList is a 1xN cell adjancency list where N is the total number of
nodes
%A structure is any set of nodes that makes a complete graph.
%Structures is a cell array of adjacency lists
% ----->> To Fix: structures should be 2 or more in relationship
% ----->> a single node should not be considered a structure.
% ----->> Fix this to ignore if AdjList only has a single entry.
8
                        %A temporary cell array
NxS{1} = [];
%alist = AdjList;
Structures{1}=[];
                        %Start with an empty set
scount = 1;
                        %The search node initial index - always start
ndex = 1;
with the first node.
nodelist = [];
                           %List of all searched nodes.
tlist = [];
N=length(AdjList);
                         %Number of all nodes to be searched.
                        %This is the master list of candidate nodes.
CN = 1:N;
Initially this will be all
                        %nodes in the system, i.e. the length of AList.
  while ~isempty(CN)
      tlist = getslist(AdjList,CN(1));
      if length(tlist) > 1
      for nn = tlist
```

```
Structures{scount,ndex} = AdjList{nn};
ndex=ndex+1;
```

```
end
```

scount=scount+1;

```
ndex = 1;
      end
     CN = setdiff(CN,tlist);
  end
  scount=scount-1;
end
function st_list = getslist(alist,ndx)
adjlist=alist{ndx};
slist = []; %templist = [];
checklist = adjlist;
while ~isequal(slist,checklist)
   templist = setdiff(checklist,slist);
    slist = checklist;
    for n = templist
        templist = union(templist,alist{n});
     end
    checklist = union(templist, checklist);
 end
 st_list = checklist;
end
```

Name	Туре	Purpose
FINDALLSTRUCT	MATLAB Function	Finds all structures for all
		time-steps in an experiment

# Convention: ALLSTRUCTURES=FINDALLSTRUCT(ADJLIST)

Description: Identifies all structures in a CO for all observations (ticks). ADJLIST is an MxN cell adjacency list where N is the total number of nodes and M is the number of runs in the sample set.

A structure is any set of nodes that makes a complete graph..

Return Values: ALLSTRUCTURES, an MxP cell array where each P is the set of structures in the Mth sample, i.e., cell array of adjacency lists for each structure at each tick in an observation set.

```
Usage: >> structs=findallstruct(AL1);
```

```
>> structs{1,1}
ans =
```

```
[1] [1x2 double] [1x5 double] [1x2 double] [11]
```

```
>> structs \{1,5\}\{:\}
ans =
1
ans =
2 9
ans =
3 4 7 8 10
ans =
5 6
ans =
11
```

Related functions: FINDSTRUCT is used in FINDALLSTRUCT and is called repetitively.
Name	Туре	Purpose
CLUSCO	MATLAB Function Compute the clust	
		coefficient

#### Convention: [CG,CN]=CLUSCO(SAM)

Description: Computes the clustering coefficient for each vertex (CN) and the global clustering coefficient (CG) for a graph represented by the simple adjacency matrix (SAM).

```
function [cq, cn] = clusco(sam)
[CG, CN] = CLUSCO(SAM)
%SAM is an adjacency matrix. If it is a digraph CLUSCO will convert it
to a
%simple adjacency matrix.
%CG is the graph clustering coefficient
%CN is the clutering coefficient at each node
% The clustering coefficient for each node in a simple graph is given
by
% the ratio of the number of triangles that can be formed on that node
and
% the number of triads in which that node is central.
% Given an adjacency matrix A, its square will give the number of paths
of
% length 2 and raising it to the third power will give the number of
paths
% of length 3. We can find the number of triangles with a node n in the
% center by looking at the diagonal elements of A^3 as these will be
the
\$ paths of length 3 that start and stop on node n. We have to divide
that
% result by 2 since the triangles are counted both directions.
 The number of triads with n central is given by d(n) \star (d(n)-1)/2
% where d(n) is the degree of n.
% The clustering coefficient for the graph is the average of the
clustering
% coefficients for those nodes which have a clustering coefficient,
i.e.,
% those with d(n) > 1.
sam =(sam +sam') > 0; %make sure that SAM is simple, i.e., undirected
and
sam = sam - diag(diag(sam));
                                                   %no self loops
                %vector containing degree of each node
dn=sum(sam);
ddn=dn.*(dn-1); %possible number of links of neighbors x 2
sam3=diag(sam^3)'; %vector that is the number of triangles for each
node x 2
cn=sam3 ./ ddn;
% tmp = ~isnan(cn); %toss out the NaNs
% tcn=cn(tmp>0);
tmp = isnan(cn);
cn(tmp)=0;
tcn = cn;
cg = sum(tcn)/numel(tcn);
```

Name	Туре	Purpose	
CHARPATH	MATLAB Function	Compute characteristic path	

Convention: [LAMBDA, EFFICIENCY, ECC, RADIUS, DIAMETER] = CHARPATH(D) Description:

```
function [lambda,efficiency,ecc,radius,diameter] = charpath(D)
%CHARPATH Characteristic path length, global efficiency and related
statistics
ş
ş
    lambda = charpath(D);
8
    [lambda.efficiency] = charpath(D);
8
    [lambda,ecc,radius,diameter] = charpath(D);
8
જ
    The characteristic path length is the average shortest path length
in
    the network. The global efficiency is the average inverse shortest
ક
path
    length in the network.
ક્ર
ş
ક
    Input:
                D,
                                 distance matrix
ક
ક
    Outputs:
                                 characteristic path length
                lambda,
                efficiency,
ક્ર
                                 global efficiency
                ecc,
                                 eccentricity (for each vertex)
8
ક્ર
                                 radius of graph
                radius,
ક્ર
                diameter,
                                 diameter of graph
응
    Note: Characteristic path length is calculated as the global mean
ક્ર
of
    the distance matrix D, excludings any 'Infs' but including
ક
distances on
옹
    the main diagonal.
8
જ
% Mean of finite entries of D(G)
lambda = sum(sum(D(D~=Inf)))/length(nonzeros(D~=Inf));
if isnan (lambda) %this is the condition for unconnected vertices
    lambda = 0;
end
% Eccentricity for each vertex (note: ignore 'Inf')
ecc = max(D.*(D~=Inf),[],2);
% Radius of graph
radius = min(ecc); % but what about zeros?
% Diameter of graph
diameter = max(ecc);
% Efficiency: mean of inverse entries of D(G)
```

Name	Туре	Purpose
LIST2MAT	MATLAB Function	

Convention: ADJMAT=LIST2MAT(ADJLIST)

Description: Creates the adjacency matrix ADJMAT from the cell array ADJLIST which must be a cell array of the type created by GETADJLIST.

Return Values: ADJMAT, a matrix of adjacencies of the form

N <sub>1</sub>	N <sub>2</sub>		Nn
N <sub>2</sub>	0	1	1
:	1	0	1
N <sub>n</sub>	1	1	0

Related functions: GETADJLIST

```
function AdjMat = list2mat(AdjList)
% AdjMat = list2mat(InputList)
% Creates the adjacency matrix AdjMat from the cell array
AdjList.
% AdjList must be a cell array of the type created by
getadjlist.m.
% See also get netlogo adjlist.m
[NumofRec NumofVert]=size(AdjList);
                                           % NumofRec is the
number of rows (records)
                                            % NumofVert is the
the number of vertices
AdjMat=zeros(NumofVert, NumofVert, NumofRec);% preallocate the
matrix
for k=1:NumofRec
    for i=1:NumofVert
        [numrows, numcols]=size(AdjList{k,i}); %numrows should
always be 1
        for j=1:numcols %numcols will be the number of items in
the list
            indx = AdjList\{k,i\}(1,j); %indx is the index to the
vertex
            if indx ~= i
                            %diagonals are 0
               AdjMat(i,indx,k)=1; %
               %AdjMat(k,i,indx)=1; %
```

```
%AdjMat(AdjList{k,i}(1,j),i,k)=1;
end
```

end

end

 $\operatorname{end}$ 

Name	Туре	Purpose
SGAM	MATLAB Function	Used on individual
		structures.

## Convention: SAM=SGAM(SAL)

Description: Creates a unique (starting with 1) adjacency matrix SAM from the cell array for a specified structure SAL which must be a cell array of the type created by using FINDSTRUCT. SGAM renumbers the vertices in SAL to create an adjacency matrix corresponding to the simple graph that defines the structure.

Return Values: SAM is a 1 initialized adjacency matrix.

Related Functions: SGAM uses LIST2MAT internally. SAM is used as input to CLUSCO which computes the clustering coefficient for a structure.

Example Usage: Assume an adjacency list AdjList(M,N) where M is the number of observations and N is the number of vertices. Then,  $>> AdjList\{m,:\}$  will return the adjacency list of the mth observation. Let's say that >> AdjList{900,:} ans = 1 ans =2 14 ans =3 4 ans =4 3 ans =7 15 5 6 ans =5 15 6 ans = 7 5 15 ans = 10 12 13 8 11 ans =9 11 ans =10 8 11 12 13 ans =11 8 9 10 ans =13 12 8 10 ans =

FINDSTRUCT is then used to identify the structures that exist at the mth observation.

```
>> slist = findstruct(AdjList(900,:))
slist =
[1] [1x2 double] [1x2 double] [1x4 double] [1x6 double]
```

We can also see that there are numel(slist), i.e., 5, structures here. Note that technically a single entity is not a structure since it is in no relationship to another.

Next, we extract just the adjacency list for a particular structure.

```
>> sal={AdjList{900,slist{2}}}
sal =
[1x2 double] [1x2 double]
```

And now we use SGAM to return the 1-indexed adjacency matrix for that structure:

```
>> sam=sgam(sal)
sam =
0 1
1 0
```

In this case we see that structure-2 at observation 900 is simply two entities.

We can see in slist that structure-5 consists of 6 entities. Let's extract the adjacency matrix just for structure-5.

```
>> sal={AdjList{900,slist{5}}}
sal =
  [1x5 double] [1x2 double] [1x5 double] [1x4 double] [1x4 double] [1x4
double]
>> sam=sgam(sal)
sam =
  0
      0
          1
               1
                   1
                       1
  0
      0
          0
              1
                   0
                       0
   1
      0
          0
              1
                   1
                       1
   1
      1
          1
               0
                   0
                       0
      0
           1
               0
                   0
   1
                       1
      0
          1
                   1
   1
               0
                       0
```

function sam = sgam(sal)% Convention: SAM=SGAM(SAL) % Description: Creates a unique (starting with 1) adjacency matrix SAM % from the cell array for a specified structure SAL which must be a cell % array of the type created by using FINDSTRUCT. SGAM renumbers the % vertices in SAL to create an adjacency matrix corresponding to the simple % graph that defines the structure. % Return Values: SAM is a 1 initialized adjacency matrix. % Related Functions: SGAM uses LIST2MAT internally. SAM is used as input to % CLUSCO which computes the clustering coefficient for a structure. %SGAM simple graph adjacency matrix from adjacency list %This helper function takes a structure list and reassigns the numbering %so that the simple graph that represents the structure starts from 1. %Use the result of this e.g., %saj2={AdjList{948,structures{5}}} which will return the adjlist for %a particular structure %test2 = sgam(saj2) %which will start the new adjlist at 1 and compute the %corresponding adjacency matrix using list2mat %Use this to produce the adjmat just for a single structure nnum = 0;tsal{1}=[]; for m=1:numel(sal) %clean up empties if ~isempty(sal{m}); tsal{m}=sal{m};nnum=nnum+1; end end sal = tsal; nnum = numel(sal); %the # of nodes, this is the # of rows in sal %[nnum,~] = size(sal);

```
for n = 1:nnum %look at each row
    x=sal{n}(1); %get first label in row, we will be looking
for it in the list
    for nn = 1:nnum %again look at each row
```

```
mnum = numel(sal{nn}); %the # of nodes in the row
ind=find(sal{nn}==x); %any in row that are x?
sal{nn}(ind)=x;
for m = 1:mnum %the # of nodes in the row
    test = sal{nn}(m);%retrieve a node
    if test == x %is it the label we are checking?
        sal{nn}(m) = n;%if so, replace it with n
    end
end
end
end
```

```
sam = list2mat(sal);
```

Sample Scipts for Reading Data and Computing Metrics					

- Get the adjacency list for each run
  - o >> CP1ALists = make\_adjacency\_lists(ALfilename,1:100);
- Compute the state vector for each run

   >> for n=1:100;SV(n,:) = EBSSV(CP1ALists,1,n);end;
- Compute the clustering coefficient for each run

   >> for n=1:100;CC(n)= clusco(list2mat(CP1ALists(1,:,n)));end;
- Computer the characteristic path length for each run
  - >> for n=1:100;CPL(n)=
     charpath(pathlength(list2mat(CP1ALists(1,:,n))));end;

Computing the Characteristic Path Length for the Minimally Constrained Particle System

for n=1:1000;MCPcpl(n)= charpath(pathlength(list2mat(MCPALs(1,:,n))));end;

Computing the Clustering Coefficient for the Minimally Constrained Particle System

for n=1:1000;MCPcc(n)= clusco(list2mat(MCPALs(1,:,n)));end;

Retrieve the Adjacency Lists for rest-length = 1 in the Inherent Constraint Particle System

TestIC1= make\_adjacency\_lists(TestIC,1:100);

# APPENDIX C: A HEURISTIC MEASURE OF ENTITY SOPHISTICATION IN EMERGENT BEHAVIOR SYSTEMS

In the following discussion, ideas derived from the characteristics used to describe agents are applied to the EBS concept of *entity*. Keeping in mind the distinction between entities and agents; "agent" as viewed here is a descriptive architecture, or implementation method as opposed to a class of entity that may be represented by any coding paradigm. This distinction is reasonable when one considers that objects as inanimate as a rock or as elaborate as a human can be represented by an agent with the appropriate characteristics. In the EBS context, a rock might indeed be an entity that is part of an EBS such as an avalanche just as much so as a human who is part of a terrorist organization. Given such consideration, either the avalanche or the terrorist organization is an EBS comprised (mainly) of either rock entities or human entities, both of which can be modeled using agent-based methods and specified with the appropriate properties. The entities of interest are the *actors* in the system.

This excursion into entity sophistication came about during the course of the dissertation study. It is included here in an appendix since, although it is not critical to the development, analysis, and results of the dissertation, it is an area of relative interest and should prove useful if further developed. That being said, the distinction this effort revealed regarding the distinction between "complexity" and "sophistication" is valuable to the modeler and the dissertation does make use of that distinction.

## **Extending the Agent Taxonomy to Characterize Entity Sophistication**

Synthesizing the predominant thinking in agent research, Moya and Tolk proposed a taxonomy for classifying agents according to three general characteristics: 1) reasoning, 2) perception, and 3) action. These three characteristics are described by the ten properties as shown in Figure C-1.



Figure C-1. The Agent Taxonomy of Moya and Tolk

Since the exploration of synthetic EBS is made using agent-based methods, it is reasonable to examine agent-based models of EBS entities through the application of some form of the Moya-Tolk agent taxonomy. By assigning certain relative values to these characteristics, a heuristic for measuring the sophistication of an entity represented by an agent can be asserted. (At this time these relative values are specified as linearly monotonic although future work might suggest other functional relationships.)

Following is a discussion on the concepts in the agent taxonomy and how these are extended to EBS entities in general.

#### The Reasoning Characteristic

The Reasoning characteristic addresses an agent's ability to react to its environment, make decisions, seek to satisfy goals, and whether or not it has a belief structure. This characteristic is directly applicable to actors in an EBS and so we seek to assign value to; the type of architecture governing the cognitive structure of the actor, the internal representation or beliefs the actor can maintain, the volition the actor can assert in order to act to satisfy goals, and the extent to which the actor can react to events.

## Architecture

The *Architecture* property refers to the cognitive architecture of the entity and speaks to why an entity behaves the way it does. Can it form abstractions, learn, and so forth, or does it simply react to information (stimulus)? On one end of the scale are entities that are purely reactive and do not suggest the ability to contemplate a goal, the actions of which are governed by a memory-less rule-based decision process. Such entities can be described as merely reacting without thought to external stimulation and are said to be *tropistic*. Contrary to the attribute of tropism is that of volition. Entities that exhibit volition are deliberate in their actions. The Architecture property does not deal so much with what an entity does but rather why it does what it does. Is it because it is just simply reacting to external forces, or is it contemplating its action with regard to attaining some objective? The measure of the Architecture characteristic is an indication of an entity's ability to exercise *volition*, that is, a cognitive process by which it decides on and commits to a course of action. Entities that are highly volitional may reason according to symbolic representation and manipulation. In one manner this can be

thought of as the entity's ability to choose to expend energy to change its state with hope of a worthwhile reward. In practice, it is unlikely that an actor will be either purely tropistic or purely volitional; instead, the laws of physics at the very least remain inviolable so there is always some degree of tropism. In our assigning of a value to the Architecture characteristic we will consider those actors that are limited to a first order response to external stimuli (more tropistic) as the less sophisticated actors and those that are capable of seeking a goal (more volitional) to be the more sophisticated. Figure C-2 depicts the specification for the Architecture property.



Figure C-2. Architecture Property of the Reasoning Characteristic

Beliefs

The ability of an entity to form and maintain abstractions about its environment, itself, or other entities is indicated by the *Beliefs* property. Such abstraction allows the entity to project future states, i.e., form expectations, and factor these expectations into its decision processes. An entity need not have a high Architecture measure to exhibit this property to some degree; indeed a simple state machine can make decisions based on local observations where the degree of expectation might be related to its sensorium and intricacy of its state machine. A simple particle, e.g., the rock in the avalanche, does not consider its environment but is merely subject to it; although it has no ability to ponder whether it should fall or not, the physics which govern its behavior causes it to move or remain at rest. This leads directly to the next property of Goals.



Figure C-3. Beliefs Property of the Reasoning Characteristic

Goals

The rock of the avalanche can be thought of as seeking the lowest potential energy level it can based on the constraints of its environment and interactions with other rocks. This is an example of an *implicit goal*. There is no volition on the part of the rock and it reacts essentially instantly to the forces acting upon it. On the other hand, a mountain goat standing in the path of the avalanche may have the *explicit goal* of reaching the top of the mountain. Its Beliefs give it pause for contemplation and it must decide if it is safer to step aside and continue its climb after the avalanche, or to try to run down the mountain to escape. In either event the goat's survival will depend on its ability to react appropriately, considering both immediate and long-term goals, all in a timely fashion.



Figure C-4. Goals Property of the Reasoning Characteristic

252

Reactivity

The ability of the entity to react appropriately in a timely fashion is its *Reactivity*. Moya and Tolk observe that reactivity tends to be more immediate for tropistic agents as they tend to be driven by implicit goals and less so for more deliberative agents. As a measure related to entity sophistication it is reasonable to assume that the more sophisticated agent representing an entity will exhibit a greater cogitation of a stimulus. Reactions that are non-volitional and strongly constrained by the environment will tend to be nearly immediate and indicative of the implicit goals associated with the less sophisticated agent. (Implicit goals need little logic on the part of the entity.) A trivial case is an entity such as a wall that does not change state but merely serves as a boundary such as for a particle. In such a trivial case there is no reaction on the part of the wall.



Figure C-5. Reactivity Property of the Reasoning Characteristic

## The Perception Characteristic

Primarily affecting the beliefs property of an agent's architecture, the perception characteristic describes an agent's ability to be aware of its environment and other agents, its inherent errors in its awareness, and its cognizance of past causes and effects.

## Access

The scope of an agent's ability to sense and access its environment, including other agents, is a measure of the potential influence or Access that agent can have. Russell and Norvig [Rusell and Norvig] describe the concept of rational agents and describe them in terms of sensors and effectors. They translate those abilities into measures of *percepts* and *actions* which are to be considered in forming performance measures of agent success. Russel and Norvig stated, "Obviously, there is not one fixed measure suitable for all agents. We could ask the agent for a subjective opinion of how happy it is with its own performance, but some agents would be unable to answer, and others would delude themselves. (Human agents in particular are notorious for "sour grapes"—saying they did not really want something after they are unsuccessful at getting it.) Therefore, we will insist on an objective performance measure imposed by some authority. In other words, we as outside observers establish a standard of what it means to be successful in an environment and use it to measure the performance of agents." The approach used here is similar. Until such time as truly objective measures of agency can be formulated, the values of these properties will necessarily be assigned subjectively. This thinking is extended to our interest in agents used to represent entities in a model of an EBS where the extent of an agent's access can range from none (a seemingly trivial case) to *Objects* defining local environment or other meso-scale *Structures*, to complete

access to the macro-scale environment (in EBS terms the Context of Observation).



Figure C-6. Access Property of the Perception Characteristic

## Accuracy

Whereas the Access property addresses an agent's scope of sense and influence, the *Accuracy* property deals with how precise the agent can be with its inherent ability to sense and influence. If its sensorium or effectors are of poor resolution or produces uncertain results then the agent will have a low accuracy of either sensing or influencing anything within its scope. Similarly, if the detail and precision of the agent is such as to allow it to produce very certain results, then it can be said to have a high degree of accuracy. For example, an agent that can look or move only in the direction of the four cardinal points would be less accurate compared to one that could consider eight. For some *Objects* in an EBS, such as a bounding wall, Accuracy is not applicable in the sense it would be for the *Actors* in the EBS so it might be indicated by null or zero Accuracy, an admitted trivial case.



Figure C-7. Accuracy Property of the Perception Characteristic

## Memory

Moya and Tolk define memory in an agent as its ability to consider past states, actions, and results to support decision-making. Clearly, the existence of memory directly affects the Goals and Reactivity properties of the Reasoning characteristic. The Moya-Tolk perspective of agent memory is particularly well suited to assessing entity sophistication. Although emergent behavior can be exhibited by ensembles of memoryless entities, many easily recognized EBS are comprised of entities with memory. These memory structures can be categorized as:

Environment (I remember the last time I saw that door.),

Action Self (I remember the last time I saw that door I opened it.),

- Action Others (I remember the last time I saw that door I opened it and zombies came out.),
- *Effects Self* (I remember the last time I saw that door I opened it, zombies came out and I fought them.), and
- *Effects Others* (I remember the last time I saw that door I opened it, zombies came out, I fought them and killed them.)

Each step indicates a more sophisticated memory relationship thus suggesting a more sophisticated agent.



Figure C-8. Memory Property of the Perception Characteristic

## The Action Characteristic

The Action characteristic describes what manner of interaction an entity can have with other entities and its environment. Moya and Tolk recognized this characteristic as being dependent on many factors in a specific agent definition but identified communication and negotiation as the main capabilities. In this regard, the application of their agent taxonomy to entity sophistication deviates only in the specific properties identified. Whereas Moya and Tolk identify the properties of Communication, Protocol, and Negotiation, here we consider only Communication and Protocol for agents representing entities in an EBS.

## Communication

Moya's and Tolk's taxonomy examines Communication from the agent perspective of Magedanz, Rothermel, and Krause (Magedanz, et al., 1996) who emphasize a telecommunication environment and information services. Magedanz, et al proposed a taxonomy for intelligent agents that emphasizes agents whose purpose is to communicate information and in particular communication amongst networked computer systems. Since the concept of relationship is foundational to the study of EBS and relationship can be considered a form of communication between entities, the Moya-Tolk agent taxonomy is applied to entities in an EBS with some modification motivated by the ideas of perception and action. For the interest of EBS, we consider the following properties for the agent characteristic.



Figure C-9. Communication Property of the Action Characteristic

Protocol

The Protocol property describes the manner in which an entity can communicate with others. Simpler entities that typically only interact with others by direct contact are *obtrusive*. If their interaction is by means of effectors that afford them some degree of freedom or volition with their interaction, then there is a more sophisticated value of the Protocol property. If the entity is able to communicate with others by *establishing and maintaining paths* for information transfers then the Protocol property has the greater sophistication value of *dialog*.



Figure C-10. Protocol Property of the Action Characteristic

## The Entity Sophistication Heuristic

Figure E-11 summarizes the previous discussion and shows the characteristics of entities comprising an EBS as they might be implemented by agent-based models. Figure E-12 presents in a relational method to aggregate the entity characteristics' measures to form a measure of entity sophistication.



Figure C-11. Entity Sophistication Derived from the Agent Characteristics of Moya and Tolk



Figure C-12: A Graph View of the Entity Sophistication Heuristic

The heuristic in table form
-----------------------------

	Archi	tecture	Tropistic	Deliberative	Hybrid	A = T  or D or H	
ß	Belief	ŝ	Environment	Next Goal Self	Next Goal Others	B = E+Ns+N o	R = A + B + G + Ry
j.	Goals		Implicit	Explicit		G = I + E	1
Reaso	Reactivity		Immediate	Delayed	Optimized	Ry=I or D or O	
			••••••••••••••••••••••••••••••••••••••	1	·····		
	Acces	S	Partial	Complete		As = P  or C	
ption	Accur	acy	Inaccurate	Accurate		Ay=I or A	P = As + Ay + M
Perce	Memory		Environment	Action	Effects	M = En+A+Ef	
Action	Communication		Local	Network	Broadcast	C= L+N+B	
	Protocols		Interference	Effectors	Dialog	P = I+E+D	A = C + P
				]			

## Applying the Measure

Applying the measure requires each entity to be assessed according to the properties within each characteristic of the heuristic.

## Spring-Particle System

The spring-particle is the actor in this EBS. The properties of the spring-particle are: 1) spring-constant, which is from Hooke's Law, i.e., F = -kx where the force of the spring is related to the displacement of the spring x by the spring constant k, 2) damping, which is a loss factor that tends to decrease the velocity of the particles, 3) rest-length, which is simply the length of the spring where its displacement is 0 and 4) shear-point, which is the distance between any particle pair that if exceeded will remove their connecting spring, i.e., the spring breaks. Each of these spring-particle properties can be interpreted with regard to the sophistication heuristic.

## Reasoning in the Spring-Particle System

Architecture: The spring-particle actors manifest no intelligence but are purely reactive. As such these are wholly tropistic. None of the spring-particle properties affect the Architecture property.

Beliefs: The spring-particle considers its environment only in that if it collides with another entity its spring is compressed from the force of the collision. Its own next goal is prescribed to seek equilibrium as the spring seeks its rest-length. This equilibrium will be static if the spring-particle is damped and will be dynamic if not damped. If the shear-point is set to 0 then the spring-particle does not react based on what others are doing but only in response to forces exerted directly on it; this is the minimally constrained case. Its Beliefs property spans Environment and Next Goal Self. The Reactivity of the spring-particle appears at first to be immediate, but is actually delayed according to response time of the spring.

_						
	Architecture	Tropistic	Deliberative	Hybrid	A = 1 (T)	
ß	Beliefs	Environment	Next Goal Self	Next Goal Others	B = 1 + 2 (E+Ns)	R = 1 + 2
ini	Goals	Implicit	Explicit		G = 1 (Im)	D = 7
Reaso	Reactivity	Immediate	Delayed	Optimized	Ry = 1 + 2 (I + D)	K-/

Perception: The spring-particle has very limited scope of sense and influence as it can only consider forces directly acting on it and they must be from direct entity contact. Therefore, its Access is Partial. However, what it is able to sense is accurate and precise so it demonstrates high Accuracy. On the other hand, since the spring-particle is purely reactionary and maintains no history of either its environment or interaction with others it is truly memoryless.

	Access	Partial	Complete		As = 1 (P)	
ption	Accuracy	Inaccurate	Accurate		Ay = 2 (A)	P = 1 + 2 P = 3
Perce	Memory	Environment	Action	Effects	M = 0	L J

Action: The spring-particle entities have the potential to form fully connected networks and so can achieve broadcast level of sophistication in communication. Particles can interact through collision locally and can form networks. The entities in essence interfere with each other during contact and can communicate throughout a network when relationships (springs) are established. They do not achieve dialog since there is no expectation of return information.

E	Communication	Local	Network	Broadcast	C=1+2+3	$\Lambda = 6 \pm 3$
Actio	Protocols	Interference	Effectors	Dialog	P = 1 + 2	A = 9

## Potential Problems with the Heuristic (Subjects for Further Research)

The entity sophistication heuristic provides a means to distinguish amongst different types of entities where the entities are readily decomposable and the properties can be easily specified. However, are the property values correct? For example, the Communication property of the Action characteristic assumes that Network is of greater sophistication than Local, and that Broadcast is of greater sophistication than Network. However, one can argue that broadcasting is a simpler communication method than establishing a network; therefore, Network should be the greater sophistication. One can also argue that all are a form of network, but the emphasis in the Communication property is the extent that a single entity can communicate with others; only with others which are adjacent (locally), to others through intermediary entities (networked), or to all others at once (broadcast). Additional research is needed to determine this.

The Heuristic is built bottom-up, so the tendency is to build agents with the taxonomy in mind, which would produced an obvious measure of sophistication (pick your level of sophistication.) It would be very desirable to infer the properties of the characteristics from observed data. This could in essence provide a dynamic measure of sophistication with observations of entities over time. It is unknown at this time what inference or data-mining methods apply, but this would be a valuable investigation.

### References

- 1. Vicsek, Tamas. The bigger picture. *Nature*, VOL 418, 11 July 2002.
- Moya, Lisa. Andreas Tolk. Towards A Taxonomy of Agents and Multi-Agent Systems. Spring Simulation Multiconference 2007 (SpringSim'07).
- 3. Russell and Norvig. Ch 2 Intelligent Agents. *Artificial Intelligence: A Modern Approach* by Stuart Russell and Peter Norvig, c 1995 Prentice-Hall, Inc.

O. Thomas Holland is a senior engineer at the Naval Surface Warfare Center in Dahlgren Virginia assigned to the Missile Systems Engineering and Integration Branch as the Senior Strategist for Modeling and Simulation. His current organizational responsibilities include the development of strategies for increased M&S technology development and skills across the branch, division and department. He is the lead for the NSWCDD M&S Community of Practice, heads the NAVSE M&S Forum, is the NAVSEA representative to the ASN/RDA M&S Leadership Council, and the Marine Corps Systems Command's Technical Authority Expert (TAE) for Modeling & Simulation.

# **Education:**

Ph.D. Modeling and Simulation, Batten College of Engineering and Technology, Old Dominion University

M.S., Electrical Engineering, Virginia Polytechnic and State University, 1996 Thesis Title: Fractal Based Texture Features for Classification of Cast Ductile Iron Surface Roughness Images

B.S., Electrical Engineering, Tennessee Technological University, 1983

# **Professional Experience:**

2012 - Present, NAVSEA M&S Forum Lead

2011 - Present, USMC SIAT M&S Technical Authority Expert (TAE)

2009 - Present, G25 Senior Strategist for Modeling & Simulation

2008 - Present, NSWCDD M&S Community of Practice Lead

2005 - 2009, NSWCDD TEAMS Director

2004-2005, Consultant to MCSC (MCB Quantico)

1998 – 2004, TEAMS Technology Coordinator

1998-2000, Marine Corps Modeling and Simulation Working Group

1997-1998, Established Testing, Evaluation, Assessment, Modeling, and Simulation (TEAMS) Facility

1991 - 1998, Advanced Processors for Weapons Sensor Fusion Section Leader

1988 – 1991, Independent Exploratory Development Lead (Artificial Neural Systems)

1983 - Present, NSWCDD Electronics Engineer

# **Selected Awards**

- 4 Patents
- NSWCDD Leadership Award for Employee Development
- MCSC Team Award for M&S applications to the Marine Expeditionary Rifle Squad
- NSWCDD Human Awareness Award
- Office of Chief of Naval Research Certificate of Recognition
- Technology to Sea Excellence Award