

APPROVAL SHEET

Title of Thesis: MY-THESIS-TITLE

Name of Candidate: MY-FULL-NAME
DEGREE-NAME,
GRADUATION-YEAR

Thesis and Abstract Approved: _____
MY-ADVISORS-NAME
MY-ADVISORS-TITLE
Department of Computer Science and
Electrical Engineering

Date Approved: _____

Curriculum Vitae

Name: MY-FULL-NAME.

Permanent Address: MY-FULL-ADDRESS.

Degree and date to be conferred: DEGREE-NAME, GRADUATION-MONTH
GRADUATION-YEAR.

Date of Birth: MY-BIRTHDATE.

Place of Birth: MY-PLACE-OF-BIRTH.

Secondary Education: MY-HIGH-SCHOOL, MY-HIGH-SCHOOLS-CITY,
MY-HIGH-SCHOOLS-STATE.

Collegiate institutions attended:

University of Maryland Baltimore County, DEGREE-NAME MY-MAJOR,
GRADUATION-YEAR.
MY-OTHER-DEGREES.

Major: MY-MAJOR.

Minor: MY-MINOR.

Professional publications:

FULL-CITATION-INFORMATION.
FULL-CITATION-INFORMATION.

Professional positions held:

EMPLOYMENT-INFO. (START-DATE – END-DATE).
EMPLOYMENT-INFO. (START-DATE – END-DATE).

ABSTRACT

Title of Thesis: MY-THESIS-TITLE

MY-FULL-NAME, DEGREE-NAME, GRADUATION-YEAR

Thesis directed by: MY-ADVISORS-NAME, ADVISORS-
TITLE
Department of Computer Science and
Electrical Engineering

Insert the text of your abstract here. This template was created for making it easier to write a UMBC Dissertation or Master's thesis. It was adapted from Vivek Tawde's template (<http://www.csee.umbc.edu/~vivek1/writeThesis.html>) by Eric Eaton in 2005.

DISCLAIMER: The adapter and original author make no guarantees as to this template's accuracy in formatting a thesis or dissertation according to the UMBC Graduate School requirements. You are fully responsible for making certain that your final thesis is formatted correctly; this template's sole purpose is to give you a starting point for your thesis. By using this template, you implicitly agree to these terms and agree to use this template solely at your own risk. It is intended only for academic use – all other use is prohibited.

Be certain to add one final blank page to the thesis when you submit it (according to 2005 UMBC Grad School Formatting Requirements). The template does not do this automatically so that the table of contents won't include a blank entry.

**A Framework for Predicting and Controlling
System-Level Properties of Agent-Based Models**

by
Donald P. Miner

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Computer Science
2010

INSERT-DEDICATION-HERE

ACKNOWLEDGMENTS

Write your acknowledgment here.

TABLE OF CONTENTS

DEDICATION		ii
ACKNOWLEDGMENTS		iii
LIST OF FIGURES		vii
LIST OF TABLES		viii
Chapter 1	INTRODUCTION AND MOTIVATION	1
1.1	Agent-Based Models	1
1.2	Wolves, Sheep and Grass	3
1.3	The Swarm Application Framework	7
1.4	Dissertation Organization	7
Chapter 2	CHAPTER-TITLE	8
2.1	SECTION-TITLE	8
2.2	SECTION-TITLE	8
2.2.1	SECTION-TITLE	8
2.2.2	SECTION-TITLE	8
Chapter 3	CHAPTER-TITLE	9

3.1	SECTION-TITLE	9
3.2	SECTION-TITLE	9
3.2.1	SECTION-TITLE	9
3.2.2	SECTION-TITLE	9
Chapter 4	CHAPTER-TITLE	10
4.1	SECTION-TITLE	10
4.2	SECTION-TITLE	10
4.2.1	SECTION-TITLE	10
4.2.2	SECTION-TITLE	10
Chapter 5	CHAPTER-TITLE	11
5.1	SECTION-TITLE	11
5.2	SECTION-TITLE	11
5.2.1	SECTION-TITLE	11
5.2.2	SECTION-TITLE	11
Chapter 6	CHAPTER-TITLE	12
6.1	SECTION-TITLE	12
6.2	SECTION-TITLE	12
6.2.1	SECTION-TITLE	12
6.2.2	SECTION-TITLE	12
Chapter 7	CHAPTER-TITLE	13
7.1	SECTION-TITLE	13
7.2	SECTION-TITLE	13

7.2.1	SECTION-TITLE	13
7.2.2	SECTION-TITLE	13
Chapter 8	CHAPTER-TITLE	14
8.1	SECTION-TITLE	14
8.2	SECTION-TITLE	14
8.2.1	SECTION-TITLE	14
8.2.2	SECTION-TITLE	14
Chapter 9	CHAPTER-TITLE	15
9.1	SECTION-TITLE	15
9.2	SECTION-TITLE	15
9.2.1	SECTION-TITLE	15
9.2.2	SECTION-TITLE	15
REFERENCES	16

LIST OF FIGURES

1.1	A screenshot of NetLogo's graphical user interface while executing a flocking simulation.	2
1.2	A screen shot from NetLogo's Wolf Sheep Predation model.	4
1.3	The control and monitor interface for the Wolf Sheep Predation model. . . .	5
1.4	Differences in populations based on minor changes of the <i>sheep-gain-from-food</i> parameter.	6

LIST OF TABLES

Chapter 1

INTRODUCTION AND MOTIVATION

The behavior of individual agents in an agent-based model (ABM) is typically well understood because the agent's program directly controls its local behaviors. What is typically not understood is how changing these programs' agent-level control parameters affect the observed system-level behaviors of the ABM. The aim of this dissertation is to provide researchers and users of ABMs insight into how these agent-level parameters affect system-level properties. In this dissertation, I discuss a framework named the Swarm Application Framework(SAF) that I have developed that can be used to predict and control system-level behaviors of agent-based models. With this framework, users can interact with ABMs in terms of intuitive system-level concepts, instead of with agent-level controls that only indirectly affect system-level behaviors.

1.1 Agent-Based Models

Agent-based models are used by scientists to analyze system-level behaviors of complex systems by simulating the system bottom-up. At the bottom of these simulations are individual agents that locally interact with other agents and the environment. All the behavior in an ABM, from agent-level local interactions to system-level behaviors, emerge from these local interactions, which are governed by the individual *agent programs*. ABMs can

be used to understand how changes in individuals' *agent-level parameters* affect *system-level properties*.

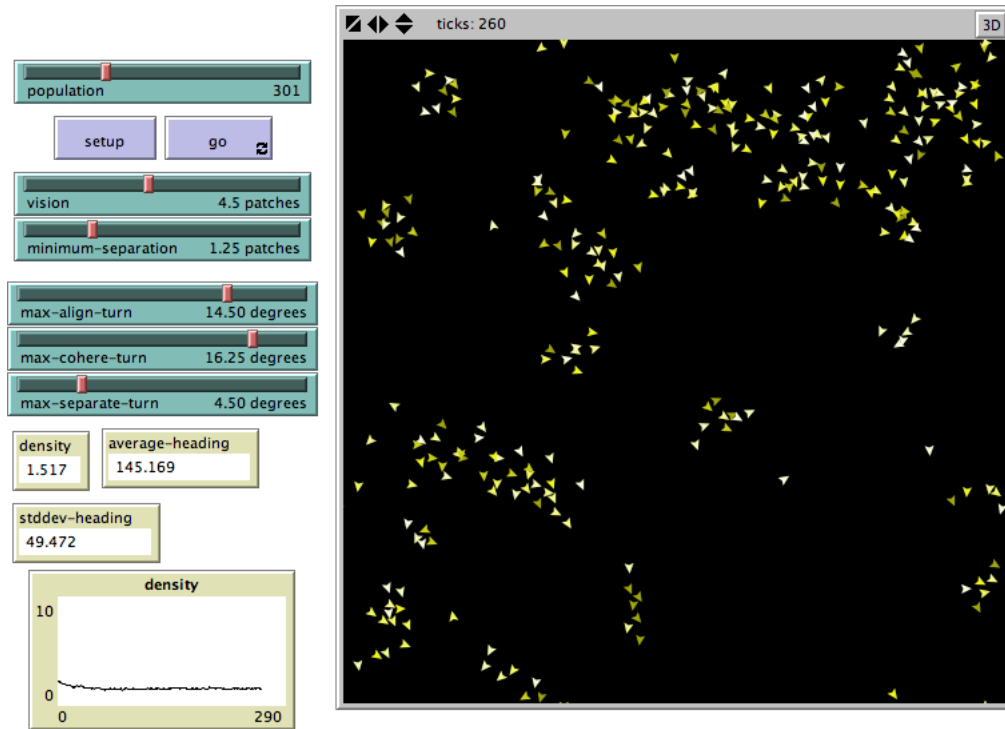


FIG. 1.1. A screenshot of NetLogo's graphical user interface while executing a flocking simulation.

Agent-level control parameters adjust the behaviors of agent-based programs. However, scientists are not typically interested in the local interactions between agents—they are interested in the resulting system-level behaviors that result. For example, researchers that studied agent-based models of lane formations in army ants were interested in the traffic patterns of the lanes, not the individual behaviors of the ants (Couzin & Franks 2003). In other work, researchers that studied locusts were interested in determining at what critical density locusts begin to swarm and destroy crops (Buhl *et al.* 2006). Typically, scientists analyze ABMs by viewing visualizations of the environment or gathering statistical data on the simulation. For instance, NetLogo, an agent-based modeling programming

environment (Tisue & Wilensky 2004), has monitors, plots and visualizations to convey system-level properties to the user. In Figure 1.1, monitors are displaying *density*, *average-heading* and *stddev-heading* statistics for a flocking domain. In addition, a plot of density shows how it has changed over time. These tools are used by a researcher to generate a mental model of how the agent-level control parameters of the flocking domain (the sliders seen in the user interface) affect these system-level properties.

Although using ABMs for researching agent-based systems has proved useful in a number of domains, I suggest there is a glaring conceptual disconnect, from the user’s perspective, between the agent-level controls and the system-level properties. The classical ABM control method of adjusting agent-level properties is unintuitive because they only indirectly affect the system-level properties. With the current methodology, a simulation has to be executed in order to observe what the values of the system-level properties will be. The only way to configure the system to have it exhibit a desired system-level behavior is a time consuming iterative process of guess-and-check. In summary, a determination of what an ABM will do at a system-level, given only the agent-level parameters, is not possible with current software. The main goal of the Swarm Application Framework is to bridge this gap between agent-level parameters and system-level properties.

1.2 Wolves, Sheep and Grass

Throughout this dissertation, I will use NetLogo’s Wolf Sheep Predation model (Wilensky 1997), which is bundled with NetLogo’s standard Model Library,¹ as an example to explain concepts.

A snapshot of its NetLogo visualization is shown in Figure 1.2. This multi-agent model simulates a food chain consisting of wolf agents, sheep agents and grass in a two-

¹<http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation>

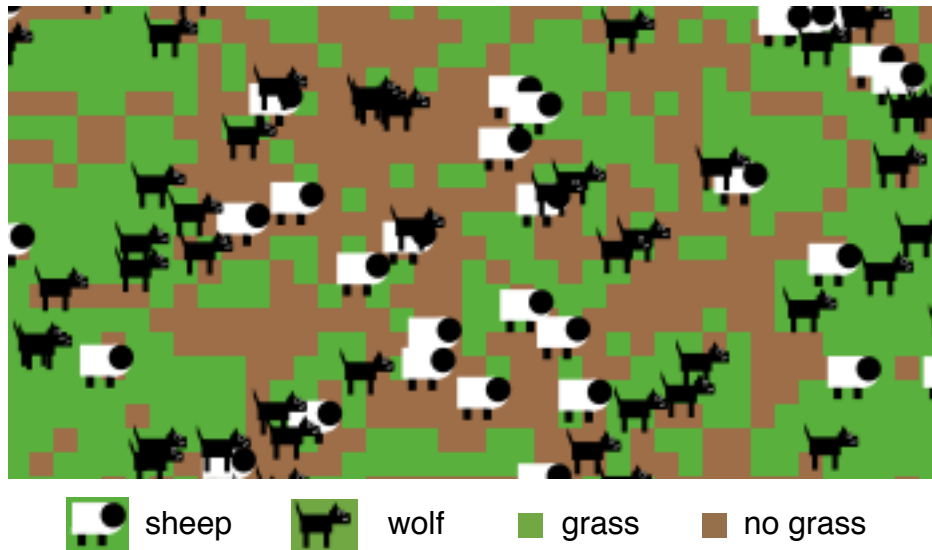


FIG. 1.2. A screen shot from NetLogo's Wolf Sheep Predation model.

dimensional space. The model is controlled by seven agent-level control parameters, and directly affect the following agent behaviors:

- The system is initialized with *initial-number-sheep* sheep and *initial-number-wolves* wolves.
- Wolves and sheep move randomly though the space.
- Wolves and sheep die if they run out of energy.
- Wolves eat sheep if they occupy the same space in the environment. Wolves gain *wolf-gain-from-food* units of energy from eating sheep. The sheep dies.
- Sheep eat grass if they are on a location of the environment that has grass. Sheep gain *sheep-gain-from-food* units of energy from eating grass. The grass dies in that grid location.

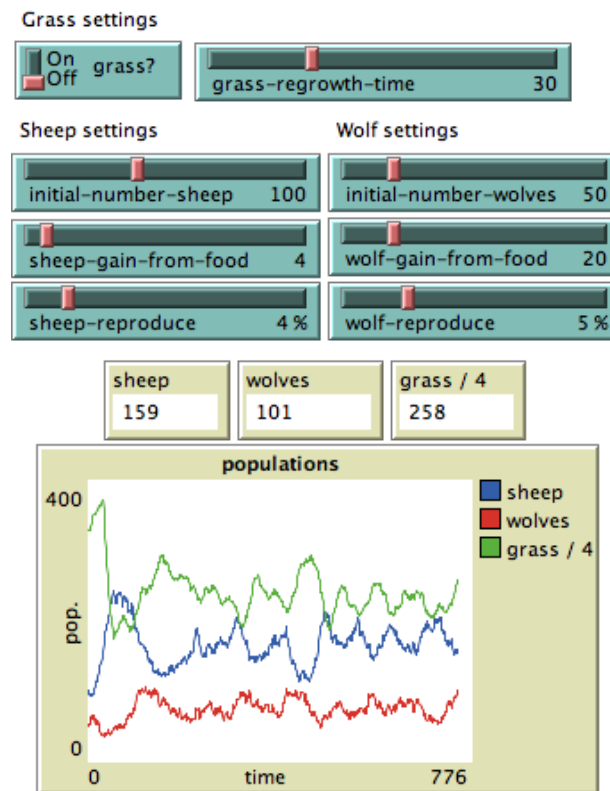


FIG. 1.3. The control and monitor interface for the Wolf Sheep Predation model.

- Every time step, each sheep and each wolf has a chance (*sheep-reproduce* and *wolf-reproduce*) to reproduce asexually. Both the parent and the child split the parent's original energy evenly (i.e., parent's energy divided by two).
- Grass regrows after *grass-regrowth-time* number of time steps.

The system-level concepts we are interested in are the number of sheep, the number of wolves and the number of grid locations containing grass. In NetLogo, these properties are displayed with monitors and a plot, as seen in Figure 1.3. The number of each population of agents may change continuously, but the average number of sheep converges. Another interesting feature is some ecosystems fail: either sheep or both sheep and wolves go extinct.

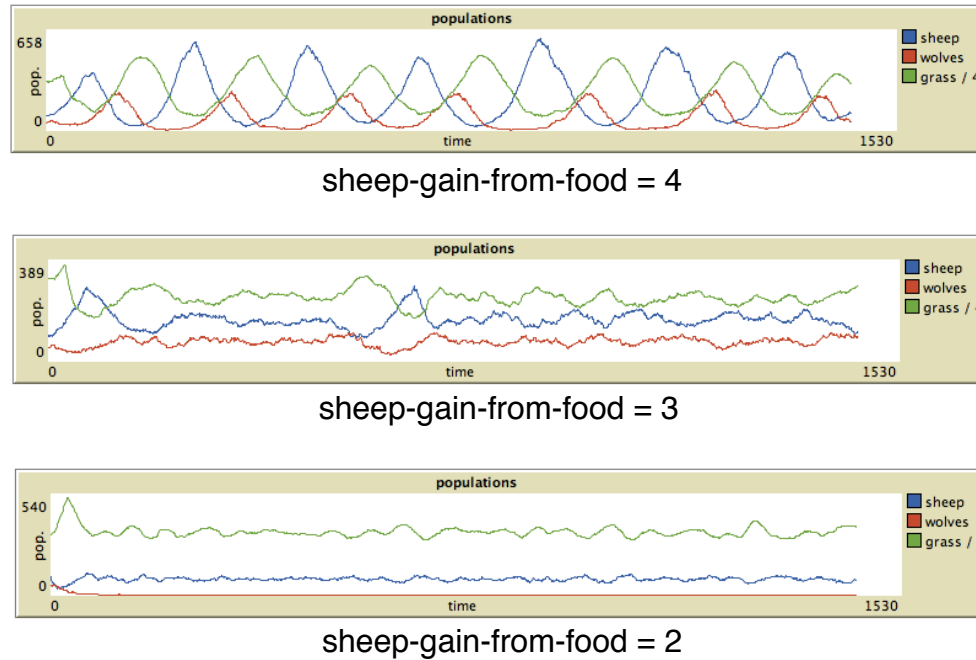


FIG. 1.4. Differences in populations based on minor changes of the *sheep-gain-from-food* parameter.

After working with this ABM for some time, a user will begin to realize that slight changes in the control parameters will yield wildly different behavior. For example, by setting *sheep-gain-from-food* to 2, 3, and then 4, major differences in system-level behavior are apparent by viewing the graphs in Figure 1.4. When the value of *sheep-gain-from-food* is 4, the system rhythmically exhibits major changes in all three agent populations. When the value is 2 or 3, the population remains relatively stable, but the average population values are different. When the value is low enough (e.g., 2) the wolves go extinct.

The Wolf Sheep Predation model is a good example of the intuitive disconnect between agent-level parameters and system-level properties. There is no clear *explicit* relationship between the controls presented in the user interface and the resulting system-level properties. An experienced user may have a qualitative understanding of the correlations,

but would not be able to predict quantitative concepts such as the average number of sheep after 2000 time steps. In the Results Chapter, I will show that this intuitive disconnect in this domain can be solved by SAF.

1.3 The Swarm Application Framework

1.4 Dissertation Organization

Chapter 2

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

2.1 SECTION-TITLE

2.2 SECTION-TITLE

(?)

2.2.1 SECTION-TITLE

2.2.2 SECTION-TITLE

Chapter 3

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

3.1 SECTION-TITLE

3.2 SECTION-TITLE

(?)

3.2.1 SECTION-TITLE

3.2.2 SECTION-TITLE

Chapter 4

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

4.1 SECTION-TITLE

4.2 SECTION-TITLE

(?)

4.2.1 SECTION-TITLE

4.2.2 SECTION-TITLE

Chapter 5

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

5.1 SECTION-TITLE

5.2 SECTION-TITLE

(?)

5.2.1 SECTION-TITLE

5.2.2 SECTION-TITLE

Chapter 6

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

6.1 SECTION-TITLE

6.2 SECTION-TITLE

(?)

6.2.1 SECTION-TITLE

6.2.2 SECTION-TITLE

Chapter 7

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

7.1 SECTION-TITLE

7.2 SECTION-TITLE

(?)

7.2.1 SECTION-TITLE

7.2.2 SECTION-TITLE

Chapter 8

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

8.1 SECTION-TITLE

8.2 SECTION-TITLE

(?)

8.2.1 SECTION-TITLE

8.2.2 SECTION-TITLE

Chapter 9

CHAPTER-TITLE

INSERT-CHAPTER-TEXT-HERE

9.1 SECTION-TITLE

9.2 SECTION-TITLE

(?)

9.2.1 SECTION-TITLE

9.2.2 SECTION-TITLE

REFERENCES

- [1] Buhl, J.; Sumpter, D.; Couzin, I.; Hale, J.; Despland, E.; Miller, E.; and Simpson, S. 2006. From Disorder to Order in Marching Locusts. *Science* 312(5778):1402–1406.
- [2] Couzin, I., and Franks, N. 2003. Self-organized lane formation and optimized traffic flow in army ants. In *Proceedings of the Royal Society of London, Series B*, volume 270, 139–146.
- [3] Tisue, S., and Wilensky, U. 2004. NetLogo: A simple environment for modeling complexity. In *International Conference on Complex Systems*, 16–21.
- [4] Wilensky, U. 1997. NetLogo Wolf Sheep Predation model. Technical report, Northwestern University.