Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

AGENT-BASED MODELING: AN INTRODUCTION AND PRIMER

Christopher W. Weimer
J. O. Miller
Raymond R. Hill

Air Force Institute of Technology
Department of Operational Sciences
2950 Hobson Way
Wright-Patterson AFB, OH 45433, USA

ABSTRACT

Agents are self-contained objects within a software model that are capable of autonomously interacting with the environment and with other agents. Basing a model around agents (building an agent-based model, or ABM) allows the user to build complex models from the bottom up by specifying agent behaviors and the environment within which they operate. This is often a more natural perspective than the system-level perspective required of other modeling paradigms, and it allows greater flexibility to use agents in novel applications. This flexibility makes them ideal as virtual laboratories and testbeds, particularly in the social sciences where direct experimentation may be infeasible or unethical. ABMs have been applied successfully in a broad variety of areas, including heuristic search methods, social science models, combat modeling, and supply chains. This tutorial provides an introduction to tools and resources for prospective modelers, and illustrates ABM flexibility with a basic war-gaming example.

1 INTRODUCTION

This tutorial provides background, application context and a how-to-get-started look at the simulation paradigm known as agent-based modeling (ABM). Those with familiarity of the field might note that the term ABM is not the standard term. Other labels for the paradigm we discuss include agent-based simulation, complex adaptive simulation systems, even object-oriented simulation. For this tutorial we use the ABM term throughout but discuss some of the rationale for use of the other terms.

We start the tutorial with some definitions of ABM and provide a view of the background work that has led to the current state of ABM. This background is not intended to be a definitive history of ABM, as once again such a history will have many versions based on the background and fundamental simulation beliefs of the writer of the history. Rather, we recount some of the influences we view as key to the development of the current ABM paradigm.

We then move onto recounting some of the applications of ABM. Simulation is a powerful, general purpose analytical tool, more often than not listed as one of two favored tools among analysts (statistical analysis or regression modeling being the other favored tool). The general applicability of simulation, and the performance of modern computers, means simulation can be used in not only the descriptive role for which it is generally intended, but also in a prescriptive role with the addition of simulation-based optimization modules. Again, the application review is not comprehensive but selective, intended to motivate the wide-ranging applicability of ABM.

Finally, we provide some how-to-get-started information. Creating a simulation from scratch is computationally intensive. Adding ABM capabilities add further computational infrastructure. Avoiding the need to create the home-grown computational environment means using one of the many publically available ABM infrastructures. We review some of these infrastructures and focus the getting starting portion of this tutorial on use of the NetLogo environment, one that is extremely popular and quite useful.

1.1 Background

An early influence on ABM was non-computational; the early theories focusing on human behavior in complex societal systems. Researchers such as Adam Smith and Donald Hobbs were attempting to understand emergent behavior in systems using individual human behavior as the catalyst for the system behavior (Heath and Hill 2010). If we turn this approach around, and use the model to observe the emergent behavior instead of using the model to explain the behavior we have a key characteristic, and benefit, of ABM.

The introduction of the computer revolutionized how to conduct quantitative analysis. The computer could quickly accomplish calculations that might overwhelm the human operator, plus the computer could do them more accurately. The computer also introduced new models of thought. Von Neumann noted that computers could "break the present stalemate created by the failure of the purely analytical approach to nonlinear problems" (Von Neumann 1966) and thereby give the researchers a means to empirically develop new theories. This insight from 50 years ago captures one of the appeals of ABM; the ABM can be used as a computational search machine from which one might derive theories of behavior among system entities, theories that can then be tested and proven using conventional methods.

Arguably, cellular automata (CA) are viewed as the computational precursor to the ABM. An intent of a CA is demonstrate complex behavior and interactions among neighboring entities using simple rules of interaction isolated within each of the simple entities; complexity through simplicity. As discussed in Heath and Hill (2010), an early but very notable CA model was the Game of Life by Conway (as recounted in Langton 1989). The Game of Life is a checkerboard in which the entities (each cell) change color using one of three simple rules based on their interaction with neighboring cells. Despite the simplicity, research soon uncovered that certain starting conditions on the board led to differing patterns of behavior, none of which were programmed into the CA; emergent behavior viewed by Bonabeau (2002) as a crucial aspect of ABM. An interesting offshoot of this early CA research is research efforts to replicate natural systems based on simple rules. Examples include capturing the flocking behavior of birds, the movements of crowds of people, even the behavior of waves in a body of water.

The CA research reinforced a key concept from complexity and complex systems, sensitivity to initial conditions. Lorenz (as recounted in Gleick 1987) was the first to find the condition using his models of weather. Other complexity concepts include the use of simple rules to mimic complex behavior and the concept of a strange attractor, areas of behavior predictable over the long term despite short-term unpredictability. A limiting aspect of the CA for more general analytic use was the game-board environment of the entities. Naturally, an extension was to free the entities from the game-board and allow their movement and interactions in a wider field of play. This can be seen as the step towards the complex adaptive system (CAS) simulation (CASS).

The field of CAS draws more inspiration from the biological world (North and Macal 2007) than do precursor methods. Within the defined playing field for the simulation, entities are allowed to roam based on some purpose and interact with other entities while having some awareness of the state of the field of play. The entities within these CAS models were dubbed "autonomous agents" due to their propensity to react to the other entities and the environmental conditions without the need for some higher-level guidance within the simulation.

Jennings, Sycara, and Wooldridge (1998) define an autonomous agent as:

- situated within some environment from which it receives sensory input;
- autonomous in the sense that human intervention is not required for the agents to proceed;
- flexible in that the agent perceives and reacts to its environment and further exhibits goal-directed behavior; and
- operating on its own thread of execution.

As recounted in Hill and Heath (2010), properties of a CAS include:

- Aggregation – entities can be grouped such that a group shares traits and behaviors and can thus be treated the same;
- Nonlinearity – the property stemming from CAS roots in complexity such that the system behavior is not the sum of the individual entity behavior meaning the behavior observed at the system level is not directly attributable to individual entities; and
- Diversity – the property that the agents are not homogeneous (as likely common in the CA modeling paradigm) and it is this diversity of entities in the model that can bring about the richer suite of interactions leading to the system level behavior.

Arguably, the development and extensive use of CAS models brought about the rapid growth and acceptance of the ABM modeling paradigm. The CAS approach allowed a wide range of applications, from social science (Epstein and Axtel 1996) through personnel modeling (Hill and Gaupp 2006), to combat analysis (Ilachinski 2000). Moreover, the closer alignment of the simulated field of play, as compared to the checkboard field of CA, meant easier acceptance of the abstraction of the real system played in the simulation. Many of the general purpose tools in the ABM realm closely align with the CAS modeling paradigm. A survey of *Winter Simulation Conference* proceedings will find a number of such instances to include an agent-based simulation track.

The CAS modeling paradigm is really the basic component of the general ABM approach. One might consider the ABM to encompass the CA and CAS paradigms but allow for richer modeling scenarios.

2 GENERAL DEFINITIONS AND GUIDELINES

2.1 Defining What We Mean by an ABM

A good starting definition of ABM comes from a *Winter Simulation Conference* tutorial, ABM "is a modeling and computational framework for simulating dynamic processes that involve autonomous agents." (Macal and North 2014). This however is quite generic. Some argue that an ABM is nothing new and is just another discrete event simulation with simulation entities executing more complex rules. This of course is a true statement as well. The dilemma arises in the subtle distinctions between what one considers a simulation and what one considers an agent.

From Macal and North (2014) an ABM has

1. Agents complete with their attributes and behaviors;
2. Agent relationships and methods of interactions to include definitions of whom the agents can interact with; and
3. The agent environment in which the agents exist with the other agents in the system.

The list above from Macal and North helps define the ABM but does not help much in distinguishing the ABM from any other discrete-event simulation. Thus, we would propose the following:

Definition 1 *An ABM is a simulation framework, using primarily the discrete-event scheduling paradigm, where the entities within the simulation have a greater degree of autonomy in movement and decision making than generally found in simulation models.*

We can clarify the definition with examples. Consider a discrete-event, network-based simulation of a shopper within a store. The shopper entity will follow certain rules and visit certain stations upon entering the system and when complete will leave the system. Interactions with other shopper entities might be limited to knowledge of line lengths or number of shoppers in a section of the store. An ABM version of the same simulation may provide the shopper entity with a set of goals to achieve, each of which might have varied levels of performance and the shopper entity might survey the immediate environment to determine subsequent steps and note the behaviors of other shopper entities using that knowledge to modify its own behavior. Thus, the difference between the simulation paradigms is subtle.

Structurally, there are other differences. Typically agents are self-contained, such as objects or very object-like, such that their memory is uniquely their own, that memory is used to fire off their rules of behavior, behaviors whose purpose is to attain some goal internal to that agent. Quite often the ABM is built upon some object-oriented simulation framework as such a framework facilitates the encapsulation of the agent behavior rules and memory.

Heath, Ciarallo, and Hill (2012) point out other aspects that differentiate the ABM as a modeling paradigm. They note that all simulation modeling endeavors require abstracting the real-world system or process into a conceptual form and then into a executable form. The ABM requires a different abstraction process focused on the goals and decisions of the distributed, autonomous entities in the system. The ABM is not necessarily going to realize a network flow of the entities through the system (which is quite a common flow model in discrete-event simulation). Finally, the entities within the ABM are provided some level of internal intelligence.

2.2 When to Consider Using an ABM

There are a variety of rules as to when to use an ABM. These rules, compiled from works such as Macal and North (2014), North and Macal (2007), as well as some of the works of the authors already cited, are the following. Consider using the ABM modeling paradigm when:

- The system or process is representable by distributed, interacting agents;
- The decisions required and the rules by which an entity is to make these decisions, are well defined;
- The agent behavior is a focus of the study and in particular how those behaviors might lead to the system-level emergent behavior;
- Adaptation within the system by entities within that system are a focus of the work;
- When adaptations by entities might affect other entities thereby changing the nature of the system under study is an aspect of the study of interest.

Another useful device for determining when to use an ABM is the taxonomy found in Heath, Cirallo and Hill (2009). The taxonomy was intended to guide the amount of validation required for the ABM. However, it serves well in defining whether or not to consider the ABM paradigm.

Each of the roles in Figure 1 are a function of the level of understanding of the real system. For a Predictor simulation, the real system is quite well known, well understood, and one simulates the system to build useful predictions for the system. For the Moderator, the system is less well understood, the model is incomplete and is executed to gain insight about the system. Finally, in the Generator role, little is known about the true system so the purpose of the simulation is to develop some fundamental theories about how the system behaves. An ABM approach is an excellent option for simulations in the Generator role and likely a preferred option for simulations in the Mediator role. The ABM approach functions well in a Predictor role but other simulation paradigms are likely more efficient and effective.

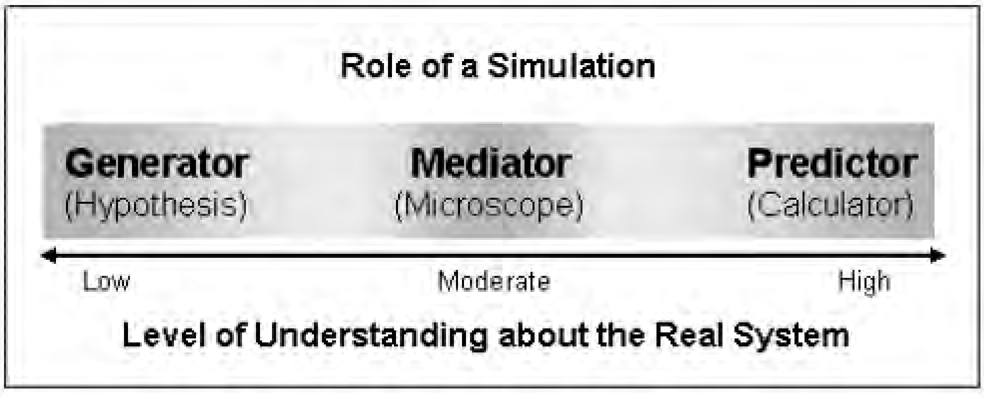

Figure 1: Purpose of the Simulation Taxonomy useful for determining when to use an ABM (Heath, Ciarallo, and Hill 2009).

3 APPLICATIONS OF ABM

There has been a wealth of applications of ABM. Table 1 in Macal and North (2014) list 15 separate areas of application with examples in each. These include agriculture, air traffic control, economics, emergency evacuation, healthcare and social behavior. Heath, Ciarallo, and Hill (2009) surveyed 279 applications of ABM in compiling their survey of ABM modeling practices with a focus on model validation.

A special issue of Mathematics and Computer Modelling focused on agent-based models. Elliston and Cao (2006) used an ABM to examine managerial decisions affecting fishery operations while Soulie and Thebaud (2006) examined the impact of regulatory measures on the same industry. Purnomo and Guizol (2006) examined foresting operations while Bah et al. (2006) examined land and resource use policies near oil drilling areas. Georgoudas, Sirakoulis, and Andreadis (2007) modeled earthquake activity. The *Journal of Simulation* has published two special issues on the subject, one in 2010 and another in 2013. Numerous other journals have similarly had special issues on the use of ABM within their domain.

The next subsections provide a more detailed review in specific application domains of ABM.

3.1 Social Sciences

ABM of sociological phenomena is not new; one of the first ABMs examined racial segregation in housing (Schelling 1971). Advances in computer processing have enabled greater use of this technique in the last two decades. Epstein and Axtell’s (1996) Sugarscape marked the beginning of a research paradigm termed Generative Social Science (GSS). The key desideratum of GSS is the use of the simplest possible set of rules to explain an emergent behavior of interest (Epstein 2006).

GSS has gained popularity as a methodology, and examples of its application can be found in many of the social sciences including economics (Zhang et al. 2010; Roozmand et al. 2011), archaeology (Epstein 2006), and sociology (Gorman et al. 2006; Mäs, Flache, and Helbing 2010). In psychology, Epstein (2006) generated thoughtless application of norms in an ABM and Willer, Macy, and Kuwabara (2009) supported this with laboratory experiments showing support of norms that disagree with personal beliefs. This demonstrates the potential for GSS and traditional experimentation to augment each other.enemy engagements and an extend mission duration.

Agent_Zero (Epstein 2013) presents the most recent generative social science model by Joshua Epstein. Ideally, hypothetical rule or behavior sets should be grounded in experimental results, and with Agent_Zero Epstein took pains to base his agents’ behavior on neurocognitive studies. Agent_Zero is presented in two forms: a detailed look at an individual based on mathematical models generally accepted in the neurocognitive literature and an agent-based model.

The mathematical model (Epstein 2013) is a series of differential equations that describe fear activation under continued stimulus and fear extinction when stimuli cease. This model is based on classical conditioning theory and parameterized through experimental studies of neuronal activation (also known as Pavlovian conditioning). In this model, an unconditioned stimulus (US) such as an electric shock follows presentation of a conditioned stimulus (CS) such as an audio tone. The US naturally evokes an unconditioned response (UR) of fear. Over time, the CS will also evoke a conditioned response (CR): fear. Even in the absence of the US, the CS will continue to evoke the CR, though this will extinguish over time. The ABM implementation of Agent_Zero captures the Affective component described in the mathematical model and adds a Rational and a Social component. In Agent_Zero Epstein (2013) presents five computational parables demonstrating how the various components interact and the resulting emergent behavior, along with a large number of interesting extensions. The Agent_Zero text includes NetLogo (Wilensky, 1999) code and numerous figures throughout graphically depicting various results and trajectories.

3.2 Combat Modeling

For a typical combat simulation the blue and red forces defined as adversaries make up a dynamic, non-linear, complex adaptive system in which the overall system behavior emerges from the aggregate interactions among individual agents (Carres 2002). Therefore an ABM approach makes sense for modeling combat. Looking for some of the earliest uses of ABM within the military modeling community we look back to October 1995, when two scientists working for the Commanding General of the US Marine Corps Combat Development Command embarked on Project Albert (Brandstein, Horne, and Friman 2000). Project Albert used a combination of new models and tools, multidisciplinary teams, and the scientific method to understand how ABM techniques could be correctly applied to represent a broad spectrum of military operations. As part of Project Albert Dr. Andy Illachinski (2000) developed an ABM called the Irreducible Semi-Autonomous Adaptive Combat (ISSAC) model referenced earlier in this paper.

At about the same time the Air Force Space Community (SMC/XR) was releasing their first version of an ABM called the System Effects Analysis Simulation (SEAS) in April 1994 (USAF 2011). SEAS has the ability to model the presence and interaction of a large variety of unique agents within a combat mission scenario. Some examples of the agents that can be represented in SEAS are tanks, Surface to Air Missile (SAM) sites, Unmanned Aerial Vehicles (UAVs), fighters, and satellites. A typical mission scenario from SEAS is illustrated in Figure 2.

Figure 2 shows that SEAS represents not only various combat agents but also their respective sensors and communication devices. SEAS is built around three different logic entities: agents, devices, and environments. Agents interact through use of devices (weapons, sensors, communication systems) with each other and the environment with conflict outcomes emerging from the resulting interactions. Agents represent logical members acting within the combat mission scenario at the individual combatant level or as a larger military unit such as multi-ship fighter flight or Army platoon or company. The environment represents the battlespace consisting of locations, terrain, weather, jamming, and day or night characteristics (USAF 2011).

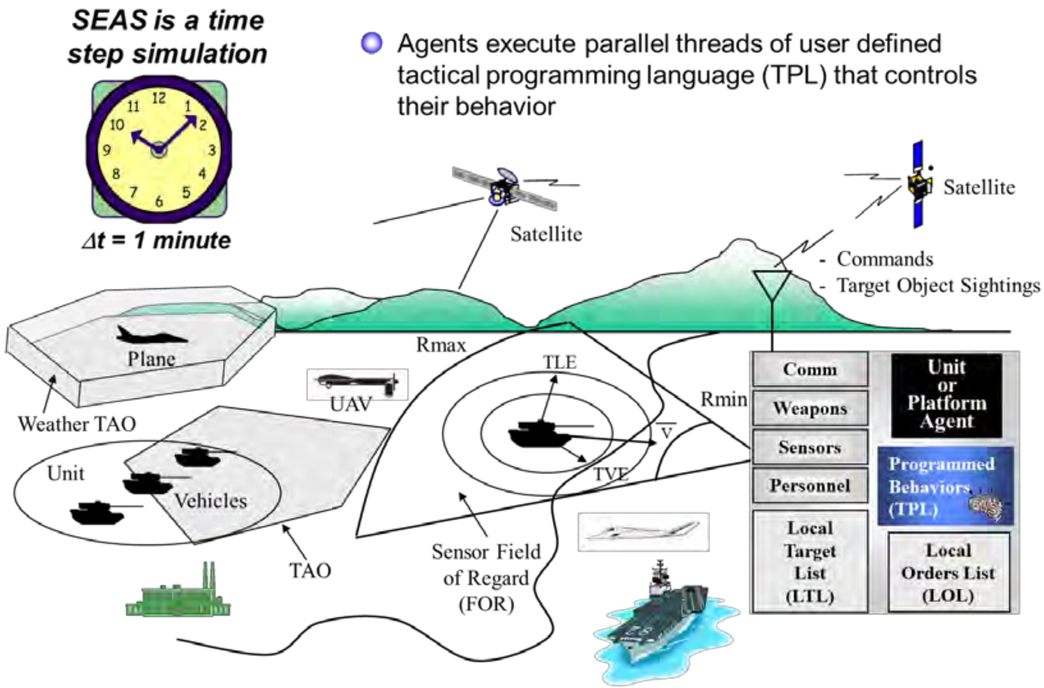

Figure 2: SEAS mission scenario representation (USAF 2011).

A relatively new ABM gaining traction with the Air Force simulation community is the Analytical Framework for Simulation, Integration, and Modeling (AFSIM), originally a tool developed by Boeing that the Air Force Research Lab (AFRL) now manages (Connors 2015). AFSIM consists of a set of tools used to load simulation scenarios, populate different objects within the simulation, and then control the simulation execution. Because AFSIM is object-oriented, objects can be defined, reused, expanded, and manipulated easily. A primary benefit of object-oriented simulation systems such as AFSIM is that libraries of existing platforms, to include weapons, sensors, and vehicles, can be used as a base to define new platforms or systems within the simulation environment. Autonomous agents in AFSIM are, in essence, a combination of different platform, sensor, and weapon objects combined with algorithms to generate behavior (i.e. actions and responses to the environment) that represent realistic decision-making (Connors 2015).

3.3 Supply Chains

When building a simulation to provide some insight on supply chain performance, an ABM approach makes sense due to the large number of interactions among different processes such as production, marketing, shipping, inventory control and participants such as suppliers, manufacturers, wholesalers, and customers. The discussion in this section focuses on the area of inventory control and provides a number of specific examples of research in this area.

Ito and Abadi (2002) develop an ABM for a warehouse system composed of three subsystems; agent-based communication system, agent-based material handling system, and agent-based inventory planning and control system. This warehouse system monitors the fluctuation and uncertainty of demands from customers, and provides just-in-time delivery of materials. The ABM utilizes master agents and subagents including customer, supplier, order, inventory, product, supplier order, and automatic-guided vehicle agents. With additional research the authors plan on incorporating the capability for autonomous setting of parameters to determine the order points or order-up-to-level point of products based on the history of customer orders and supplier lead times.

Li and Li (2008) develop an ABM for a multi-location inventory system with several retailers who share one supplier. The model considers demand lead time, replenishment lead time, and transshipment lead time but without employing a central agency to determine transshipments with retailers making decisions separately. Optimal inventory policies are explored by considering holding, ordering, transshipment, backorder, and transshipment benefit costs.

Jirong et al. (2008) develop a 4-level multi-agent system model for supply chain inventory with a decision making model for every enterprise agent in the supply chain. Their approach is selected due to the dynamic nonlinear complexity of such a supply chain inventory system. Results from their study exploring the influence of lead time and information sharing among the four agent types (retailer, wholesaler, distributor, and manufacturer), confirm that the information sharing strategy effectively decreases the variation amplitudes of inventory for each enterprise in the supply chain (diminishes the bullwhip effect).

Sirivunnabood and Kumara (2009) explore appropriate risk mitigation strategies for a supply chain network under supplier risks. Their ABM includes supplier agents, plant agents, warehouse agents, customer agents, and a controller agent. Supply chain operation is simulated and performance evaluated under randomly generated risk events. Their analysis explores the impact of having a redundant supplier and reserving more inventories as two risk mitigation strategies against four types of risks, defined by frequency and duration.

Krishnamurthy, Khorrami, and Schoenwald (2008) consider a new inventory control technique for large-scale supply chains incorporating stochastic transport delays, manufacturing times, repair times and probabilistic characterization of part repair. Optimization techniques for inventory control of bidirectional stochastic supply chains are computationally intractable, leading the authors to use an ABM. They model an aircraft supply chain involving multiple original equipment manufacturers, depots, bases, squadrons, and planes. Through use of an adaptive feature the simulation can adjust stock levels with the objective of reducing excess inventory and maintaining or increasing mission capability of aircraft. Simulation output can be used to determine the number of parts of each type that each site should order from an associated supplier site, and the number of parts of each type to start manufacturing.

Harper et al. (2011) proposes a framework for designing an agent based simulation to allow for easy aggregation and/or disaggregation of agent characteristics, behaviors, and interactions using a supply chain modeling context. When ABM is to be used with different levels of resolution the key steps that are affected are initial planning, agent and agent rule design, data collection and entry, and model execution. While identifying the purpose of the model and the questions the model is intended to answer, there must be some delineation between the different levels or resolution needed for these questions. This is not a trivial process, but can be eased by systematically analyzing the system under study and determining what data is available. The way agents are designed affect the ease of switching levels of resolution. Since multiple levels of resolution have different data requirements, the data collection and entry process is a key step in ABM for aggregation and disaggregation. More data analysis is necessary to validate the method of data aggregation, so data collection and data analysis generally take more time. However, this is balanced by the ability to model and analyze selected parts of the system at a high level of detail or more of the system at an aggregated level. Finally, the model execution process requires some data input changes to change levels of resolution.

4 HOW TO BUILD AN ABM

4.1 Getting Started

While courses in ABM are becoming increasingly common at universities, it remains the exception rather than the rule, so researchers may find the task of getting started with ABMs daunting. However, the last decade has seen significant maturation of the field in the resources available to prospective agent-based modelers.

There is not, as yet, a standard textbook for the field, but several recent books take on the task of introducing agent-based modeling. North and Macal (2007) present the topic in a business context using spreadsheet models and Repast. Both Railsback and Grimm (2012) and Wilensky and Rand (2015) present the topic in a more topic-agnostic context focusing on the use of NetLogo. Borschev (2013) addresses ABM along with discrete-event simulation and system dynamics modeling focusing on the use of AnyLogic. Depending on your needs, any of these books make excellent resources.

Once comfort with the agent perspective has been built, user forums and tutorials for each of the ABM software tools represent the most up-to-date and useful reference for building expertise in the mechanics of building a model. Most tools also have considerable libraries of models built for research and instructional purposes that are well documented.

4.2 Software Tools

At its root, ABM has arisen from the development of object-oriented programming (OOP) languages. As such, many modelers prefer to build ABMs from scratch using OOP development environments. Common OOP languages include Java, C++, C#, Python, and Ruby. This is the most flexible environment within which to build a model, and in many cases these models will run the fastest. However, many modelers prefer to work in an environment with pre-constructed methods that facilitate ABM development.

The list of ABM development tools is far too long to exhaustively present here, but some common ones include NetLogo, Repast, AnyLogic, and Simio. NetLogo (Wilensky 1999) and Repast (North et al. 2007) are both free and open-source software for Mac OS X, Windows, or Linux. These are very flexible programming tools with limited graphical user interfaces. AnyLogic and Simio are both commercial tools with more robust graphical user interfaces that can be used for ABMs or to embed agents in other types of models.

4.3 An Example ABM

Lanchester (1916) presented a now classic deterministic model of outcomes of ranged combat based on the fighting effectiveness and the troop strengths (i.e., number of soldiers) of two opposing forces. The system of differential equations that came to be known as the Lanchester Square Law is given by

$$\begin{aligned} b'(t) &= -cr(t), \qquad b(0) = b_0, \\ r'(t) &= -kb(t), \qquad r(0) = r_0, \end{aligned}$$

where *c* and *k* are the fighting effectiveness coefficients of the red and blue forces, respectively, and *b* and *r* are the troop strengths of the blue and red forces, respectively.

While the Lanchester Square Law has been widely applied as an attrition model, the lack of high-quality data regarding troop strengths over time in battle has created problems in validating the model. In applications attempting to explain the Battle of Kursk and the Battle of Ardennes, it has proved not to be a good predictor of outcomes (Lucas and Turkes 2004). In another application to battles between fire ant colonies, the Square Law again failed to explain outcomes (Plowes and Adams 2005). Perhaps an agent-based model of battles could provide a better explanatory model.

As a first step, an agent-based model could be built to replicate basic Lanchester's Square Law performance. Let the two forces be equally effective. Then the below NetLogo code is sufficient to replicate Lanchester's Square Law. Note that, in NetLogo, turtles are the pre-built class of agents that are capable of movement.

```
to setup
   clear-all
   spawn-forces
   reset-ticks
end

to spawn-forces
```

```
    create-turtles 2000 [
       set color blue
       setxy random-xcor random-ycor
    ]
    create-turtles 2050 [
       set color red
       setxy random-xcor random-ycor
    ]
end

to go
    ask turtles [ attack ]
    tick
end
to attack
    let target one-of turtles with [color != [color] of myself]
    if target != nobody [ ask target [ die ] ]
end
```

This code has two simple sections: setup and go. To setup, all variables and agents are cleared, and then the square space is filled with blue agent and red agents. To go, in random order, each agent kills exactly one enemy, if an enemy remains to kill. Running many iterations of this simulation, we find that the mean number of agents remaining once one side has been killed is equal to that predicted by Lanchester's Square Law. An example run's outcome is shown in Figure 3.

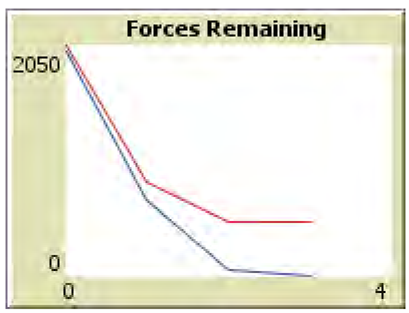

Figure 3: Outcome of basic Lanchester ABM.

Solving Lanchester's equations is much faster than running many iterations of the ABM, with the ABM requiring exponentially more time to compute. However, the ABM has the advantage of flexibility and natural interpretability. In the basic model, the agents are placed randomly, targeting occurs without regard for range, and all agents are homogeneous. It is unknown how each of these changes in assumptions would impact the deterministic Lanchester model, but an ABM can easily be altered to address these questions.

Let us add some basic heterogeneity. Suppose that test of red force weaponry indicate that, at a range of 1 unit, they will kill an enemy with probability 0.9, with that probability decreasing exponentially with range. Meanwhile, blue force weaponry has an equivalent probability of 0.95, also with exponential degradation with range. It is safe to assume, in this scenario, that each soldier will choose the closest target. By adding a few lines to our "attack" method, this is easily implemented.

```
to attack
    let targets turtles with [color != [color] of myself]
    let target min-one-of targets [distance myself]
    if target != nobody [
       let accuracy 0
       ifelse color = blue [ set accuracy 0.95 ] [ set accuracy 0.9 ]
       if random-float 1 < (accuracy ^ (distance target)) [
```

```
            ask target [ die ]
        ]
    ]
end
```

This modification alters the outcome of the battle; the blue forces can overcome the numerically larger red forces with their technological superiority. An example outcome is shown in Figure 4.

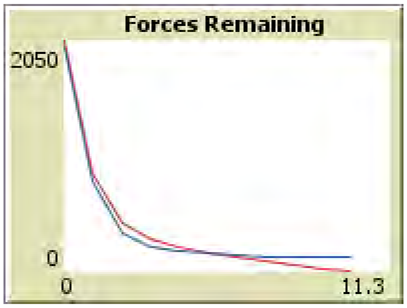

Figure 4: Outcome of a technologically imbalanced battle.

One can imagine many variations of the base model that could easily be programmed; different troop maneuvers, introducing elite forces on either side and intra-force heterogeneity, reinforcement strategies are just a few. The ability to code these departures showcases the strength of the agent-based approach. It is recommended to start with a simple model and incrementally increase the complexity when building in this way.

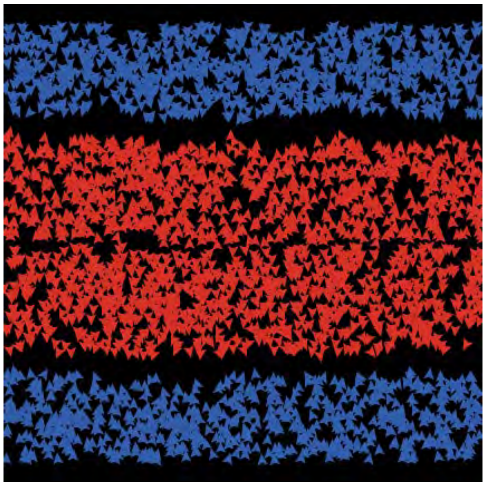

Figure 5: One tick into a battle with red forces surrounded.

For a final deviation from the Lanchester model, consider a red force being attacked from two sides by a blue force with only 90% their troop strength (see Figure 5). Blue force weapons have mean accuracy of 0.62 at 1 unit distance, whereas red force weapons have mean accuracy of 0.60 at the same range, with individuals' accuracy varying according to a normal distribution with a standard deviation of 0.1. Blue forces are short on ammunition, so they have orders to only fire from a maximum range of 5 units. The red forces, meanwhile, can fire at any range. If a soldier fires, they can move only 25% the distance that they could move otherwise. This level of complexity remains simple to implement in an ABM. The code for this scenario is shown below.

```
turtles-own [
```

```
  attack_radius
  accuracy
]

to setup
  clear-all
  spawn-blue-forces
  spawn-red-forces
  reset-ticks
end

to spawn-blue-forces
  create-turtles 1800 [
    set color blue
    set attack_radius 5
    set accuracy random-normal 0.62 0.10
    set accuracy min list 1 accuracy
    set accuracy max list 0 accuracy
    set xcor random-xcor
    set ycor (max-pycor / 2) + random-float (max-pycor / 2)
    if random 2 = 1 [ set ycor (- ycor) ]
  ]
end

to spawn-red-forces
  create-turtles 2000 [
    set color red
    set attack_radius 32
    set accuracy random-normal 0.60 0.10
    set accuracy min list 1 accuracy
    set accuracy max list 0 accuracy
    set xcor random-xcor
    set ycor (random-ycor / 2)
  ]
end

to go
  ask turtles [ attack ]
  tick
  if count turtles with [color = blue] = 0 or count turtles with [color = red] = 0
[stop]
end

to attack
  let targets turtles in-radius attack_radius with [color != [color] of myself]
  let target min-one-of targets [distance myself]
  ifelse target != nobody [
    face target
    if random-float 1 < ( accuracy ^ (distance target) ) [
      ask target [ die ]
    ]
    move 0.25
  ] [
    move 1
  ]
end

to move [ steps ]
  if any? turtles with [color != [color] of myself] [
    face min-one-of (turtles with [color != [color] of myself]) [distance myself]
  ]
  forward steps
end
```

This more complex model yields a slight advantage to the blue forces; over 100 replicates, 68 result in a blue victory. This result is statistically significant ($p < 0.001$). More importantly, the ABM gives the flexibility to ask the question in the first place.

ACKNOWLEDGMENTS

The views expressed in this article are those of the authors and do not reflect the official policy of the United States Air Force, Department of Defense, or the US Government. The authors would like to thank Dr. Stewart Robinson for the opportunity to present this tutorial.

REFERENCES

Bah, A., I. Toure, C. Le Page, A. Ickowicz, and A. T. Diop. 2006. "An Agent-Based Model to Understand the Multiple Uses of Land and Resources Around Drillings in Sahel." *Mathematical and Computer Modeling* 44:513-524.

Bonabeau, E. 2002. "Agent-based Modeling: Methods and Techniques for Simulating Human Systems." *Proceedings of the National Academy of Sciences* 99:7280-7287.

Borschev, A. 2013. *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6.* AnyLogic North America.

Brandstein, A., G. Horne, and H. Friman. 2000. "Project Albert + ROLF 2010 = Red Orm." Swedish National Defense College Paper.

Cares, J. R. 2002. "The Use of Agent-Based Models in Military Concept Development." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucesan, C. H. Chen, J. L. Sowndon, and J. M. Charnes, 935-939. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Connors, C.D., J. O. Miller, and B. Lunday. 2015. "Using Agent-Based Modeling and a Designed Experiment to Simulate and Analyze a New Air-to-Air Missile." *The Journal of Defense Modeling and Simulation*: *Applications, Methodology, Technology* 13(3):321-330.

Elliston, L. and L. Cao. 2006. "An Agent-Based Bioeconomic Model of a Fishery with Input Controls." *Mathematical and Computer Modeling* 44:565-575.

Epstein, J. M. and R. Axtell. 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. Brookings Institute Press, Washington, DC.

Epstein, J. M. 2006. *Generative Social Science: Studies in Agent-Based Computation Modeling.* Princeton University Press, Princeton, NJ.

Epstein, J. M. 2013. *Agent_Zero: Toward Neurocognitive Foundations for Generative Social Science*. Princeton University Press, Princeton, NJ.

Georgoudas, I. G., C. Ch. Sirakoulis and I. Andreadis. 2007. "Modelling Earthquake Activity Features using Cellular Automata." *Mathematical and Computer Modeling* 46:124-137.

Gleick, J. 1987. *Chaos: Making a New Science*. Viking, New York, NY.

Gorman, D. M., J. Mezic, I. Mezic, and P. J. Gruenewald. 2006. "Agent-Based Modeling of Drinking Behavior: A Preliminary Model and Potential Applications to Theory and Practice." *American Journal of Public Health* 96(11):2055-2060.

Harper, T. J., J. O. Miller, R. R. Hill, and J. R. Wirthlin. "Agent Based Simulation Design for Aggregation and Dissaggregation." In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and J. Fu, 259-270. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Heath, B. L., F. Ciarallo, and R. R. Hill. 2009. "A Survey of Agent-Based Modeling Practices (January 1998 to July 2008)." *Journal of Artificial Societies and Social Simulation* 12(4).

Heath, B. L. and R. R. Hill. September 2010. "Some Insights into the Emergence of Agent-Based Modeling." *Journal of Simulation* 4(3):163-169.

Heath, B., F. Ciarallo, and R. R. Hill. 2012. “Validation in the Agent-Based Modelling Paradigm: Problems and a Solution.” *International Journal of Simulation and Process Modeling* 7(4):229-239.

Heath, B. L., F. W. Ciarallo, and R. R. Hill. July 2013. “An Agent-Based Approach to Analyze the Impact of Warehouse Congestion on Cost and Performance.” *International Journal of Advanced Manufacturing Technology* 67(1-4):563-574.

Hill, R. R. and M. P. Gaupp. 2006. “Visualization Tools for Prescriptive Analysis of a Complex Adaptive System Model of Pilot Retention.” *International Journal of Modeling and Simulation* 6(4):378-384.

Ilachinski, A. 2000. “Irreducible Semi-autonomous Adaptive Combat (ISAAC): An Artificial Life Approach to Land Warfare.” *Military Operations Research* 5(3):29-47.

Ito, T. and S. M. M. J. Abadi. 2002. “Agent-Based Material Handling and Inventory Planning in Warehouse.” *Journal of Intelligent Manufacturing* 13:201-210.

Jennings, N. R., K. Sycara, and M. Wooldridge, 1998. “A Roadmap of Agent Research and Development.” *Autonomous Agents and Multi-Agent Systems* 1:7-38.

Jirong, W., L. Jun, Z. Yunhong, and H. Zongwu. 2008. “Simulation Study on Influences of Information Sharing to Supply Chain Inventory System Based on Multi-agent System.” *IEEE International Conference on Automation and Logistics*. 1001-1004. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Krishnamurthy, P., F. Khorrami, and D. Schoenwald. 2008. “Decentralized Inventory Control for Large-Scale Reverse Supply Chains: A Computationally Tractable Approach.” *IEEE Transactions on systems, man, and cybernetics* 38(4):551-561.

Lanchester, F. W. 1916. *Aircraft in Warfare: The Dawn of the Fourth Arm.* London: Constable and Company Ltd.

Langton, C. G. 1989. *Artificial Life*, in Artificial Life, Addson-Wesley Publishing Company, Redwood City, CA.

Li, W., and C. Li. 2008. “Transshipment Policy Research of Multi-Location Inventory System Based on Multi-Agent System.” *IEEE International Conference on Automation and Logistics*, 1344-1351. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lucas, T. W., and T. Turkes. 2004. “Fitting Lanchester Equations to the Battles of Kursk and Ardennes.” *Naval Research Logistics* 51:95-116.

Macal, C. and M. North. 2014. “Introductory Tutorial: Agent-Based Modeling and Simulation.” In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley and J. A. Miller, 6-20. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mäs M., A. Flache, D. Helbing. 2010. “Individualization as Driving Force of Clustering Phenomena in Humans.*” PLOS Computational Biology* 6(10). DOI:10.1371/1000959.

North, M. J. and C. M. Macal. 2007. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford University Press, Oxford, UK.

North, M. J., N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. 2013. “Complex Adaptive Systems Modeling with Repast Simphony.” *Complex Adaptive Systems Modeling* 1(3). http://www.casmodeling.com/content/1/1/3 [Accessed April 24, 2016].

Plowes, N. J. R., and E. S. Adams. 2005. “An Empirical Test of Lanchester's Square Law: Mortality During Battles of the Fire Ant Solenopsis Invicta.” *Proceedings of the Royal Society of London B: Biological Sciences* 272(1574): 1809-1814.

Purnomo, H. and P. Guizol. 2006. “Simulating Forest Plantation Co-Management with a Multi-Agent System.” *Mathematical and Computer Modeling* 44: 535-552.

Railsback, S. F., and V. Grimm. 2012. *Agent-Based and Individual-Based Modeling: A Practical Introduction.* Princeton University Press, Princeton, NJ.

Roozmand, O., N. Ghasem-Aghaee, G. J. Hofstede, M. A. Nematbakhsh, A. Baraani, and T. Verwaart. 2011. “Agent-Based Modeling of Consumer Decision Making Process Based on Power Distance and Personality.” *Knowledge-Based Systems* 24(7):1075-1095.

Schelling, T. C. 1971. “Dynamic Models of Segregation.” *Journal of Mathematical Sociology* 1(2):143-186.

Sirivunnabood, S. and S. Kumara. 2009. “Comparison of Mitigation Strategies for Supplier Risks: a Multi Agent-Based Simulation Approach.” *IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, 388-393. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Soulie, J-C. and O. Thebaud. 2006. “Modeling Fleet Response in Regulated Fisheries: An Agent-Based Approach.” *Mathematical and Computer Modeling* 44: 553-564.

USAF. 2011. *SEAS User Manual*. SMC/XR, Los Angeles, CA.

Wilensky, U. 1999. *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern U.: Evanston, IL USA. http://ccl.northwestern.edu/netlogo/ [Accessed April 24, 2016].

Wilensky, U., and W. Rand. 2015. *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo.* Cambridge, MA: MIT Press.

Willer, R., Macy, M. W., & Kuwabara, K. 2009. “The False Enforcemenet of Unpopular Norms.” *American Journal of Sociology* 115(2):451-490.

von Neumann, John. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana and London, 1966.

Zhang, W., G. Li, X. Xiong, and Y. J. Zhang. 2010. “Trader Species with Different Decision Strategies and Price Dynamics in Financial Markets: An Agent-Based Modeling Perspective.” *International Journal of Information Technology & Decision Making* 9(2):327-344.

AUTHOR BIOGRAPHIES

CHRISTOPHER W. WEIMER is a Major in the United States Air Force and a doctoral student in the Department of Operational Sciences of the Air Force Institute of Technology. His email address is christopher.weimer@afit.edu.

J. O. MILLER is an Associate Professor in the Operational Research Department of the Air Force Institute of Technology. He holds a Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research interests include simulation and combat modeling. His email address is john.miller@afit.edu.

RAYMOND R. HILL is a Professor in the Operational Research Department of the Air Force Institute of Technology. He holds a Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research interests include simulation, applied statistics and heuristic optimization. He is the co-chair for the Military, Homeland Security and Emergency Response track for 2016 and 2017. His email address is rayrhill@gmail.com.