AUTOMATIC GENERATION OF BELIEVABLE AGENT BEHAVIORS

by

Bridgette Parsons

Ph.D. Proposal in

Computer Science and Engineering

The College of Engineering and Computing

University of South Carolina

2013

Dr. Jose M. Vidal, Major Professor

Dr. Michael Huhns, Committee Member

Dr. John Rose, Committee Member

Dr. Marco Valtorta, Committee Member

Dr. Abbas Tavakoli, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

CHAPTER 1

INTRODUCTION

Agent-based modeling has become a commonly-used method of simulation in a wide variety of fields. It has been used to simulate traffic patterns, markets, supply chains, wildlife ecology, and networking. It is used in both the "hard" sciences and in the social sciences, due to its ability to show emergent behaviors, or behaviors that arise from the interaction between the different agents. However, the creation of an agent model can be a difficult task for non-programmers, and the model that is created may need alteration if it does not fit the real-world data as precisely as needed, making the task even more difficult.

Because simulations need to be accurate, the decision processes for agent-based models are frequently derived from sequential observations of behavior. However, the current methods of calculation for an agent's decision process require a great deal of time, and the calculations can be very complex to do by hand. Those with expert knowledge in the area that is being simulated have to work together with programmers to provide the information that is necessary to filter the data and calculate the appropriate probabilities for the various actions. This required cooperation can create further difficulties, as those that understand the area being simulated do not necessarily understand the limitations imposed by a computational model, and the programmers may not have the expert knowledge needed to correctly interpret the data.

We propose to develop an algorithm that will calculate and output one or more formal representations of an agent's behavior, similar to a Markov decision process, but with support for cycles, as needed, from the collected information, as well as a tool that will easily allow the alteration of each decision process to meet the requirements of the modeler. This tool will then produce a file in a standard format that can be efficiently read and used in a variety of applications, in order to create initial decision processes for believable behaviors.

CHAPTER 2

Related Work

Agent-based modeling is a useful method to simulate systems in which individual behaviors are varied and complex. It has the ability to show phenomena that arise from the interaction of individual agents that would otherwise be difficult to predict. It is also a flexible system of simulation, allowing the programmer to more easily change the behaviors of individual agents, or to add more agents to the system. This makes it ideal for the simulation of complex, dynamic systems, particularly those that involve human behavior (Bonabeau, 2002).

Because simulation has become such a commonly used tool, it is important that it is both reliable and valid. A simulation can be said to be reliable if it produces the same results when given the same data. While it can be relatively easy to determine if there are consistent results that are obtained with a particular model, determining the validity of that model can be more difficult (Garson, 2009). The validity of a simulation can be defined in more than one way. If a simulation appears to observers to respond in the same way as what is being simulated, it can be said to have “face validity.” This is the easiest type of validity to determine, but it can also be inaccurate. “Event validity”, obtained by providing the simulation with known data and checking the results, is a more accurate method of determining the validity of the simulation. By using known data, the correlation between the results of the simulation and the known results can be determined, giving a good sense of the simulation’s accuracy (Stanislaw, 1986).

2.1 MODELING HUMAN BEHAVIOR WITH AGENT-BASED MODELS

There have been many different decision-making methods used in the attempt to create believable behavior in agent-based models. Some researchers, such as Konolige, have argued in favor of a deductive, first-order logic system of decision-making, basing the agents' knowledge on a core set of beliefs and the knowledge that can be derived from it (1986). In this type of model, the agent's behavior is rule-based, using only the knowledge that the agent has about the world and what can be logically determined by using that knowledge. This, however, does not appear to be the way that humans reason, and seems more suitable to a knowledge base than an agent-based model.

To make the model less restrictive, some researchers have used Hintikka's "possible worlds" approach, allowing the agent a certain amount of belief in any state of the current world that could be possible, based on the facts the agent is currently aware of (1967). Ginsberg and Smith argued for this approach, because the number of changes in the world between time steps is relatively small, so that keeping track of changes in the world, and therefore changes in the agent's belief state, should be relatively simple (1988). However, when using their approach, if information contradictory to an agent's current belief state is discovered, the entire knowledge base needs to be reconstructed. It then becomes extremely computationally expensive. Furthermore, scalability becomes an issue, as beliefs about every possible fact in the given world have to be stored and updated for each agent individually.

The BDI model, or Belief, Desire, and Intentions model, is a less computationally complex version of the possible worlds model. It stores the agent's current beliefs in the state of the world based upon past events of which the agent is aware. Desires are the agent's current goals, and Intentions are the agent's plans to achieve those goals. This model is based upon ideas from cognitive psychology about the nature of human thought (Georgeff et al., 1999). By extending intentions to include multi-agent planning, Cohen and Levesque described a method of using BDI models in multi-agent systems (1990). Their agents also include a degree of commitment to their goals, allowing an agent to drop its goals in favor of a new one when interacting with another agent.

2.2 SEQUENTIAL OBSERVATIONS VERSUS COGNITIVE MODELING

A problem with modeling human behavior based upon theories of cognition is that cognitive models are difficult to quantify. Humans do not seem to make decisions based entirely upon mathematical formulae or deductive logic. In contrast to standard machine-learning methodology, humans actually make poorer decisions when given too much information. Instead, humans use heuristics, and these heuristics are not necessarily consistent between each individual, types of decisions, or even each time the same decision is made. They can be derived from individual preference, generalizations, culture, experience, or even learned from others (Gigerenzer and Gaissmeier, 2011).

Emotions also play a large part in human decision-making, as shown in neurological studies related to damage of the prefrontal cortex (Bechara et al., 2000) and studies on the amygdala and Pavlovian responses (Seymour and Dolan, 2008). There have been many attempts to incorporate emotion into agent decision-making, such as the Cathexis model (Velásquez, 1997), EBDI (Jiang et al., 2007), and a Neuro-Fuzzy agent with emotional intelligence (Sharada and Ramanaiah, 2010). However, as shown by Martínez-Miranda and Aldea, while emotion is important in human decision-making, emotional models tend to only perform well in the specific environment for which they were designed (2005). The concept of “emotion” is not well-defined in psychology, making it also difficult to quantify (Sloman, 2001). This makes a standard cognitive-based model for even a simple scenario almost impossible to accurately define and implement.

Time-sequence agent-based models, or agent-based models derived from sequences of observations over time, however, have been implemented with a great deal of success in a variety of areas other than human behavior. For instance, OptorSim uses predictive modeling to optimize the allocation of resources for file sharing, replication, and job execution in a Data Grid based on the sequence of jobs executed over time (Bell et al., 2003). A model of Chilean agriculture designed by Berger predicted economic changes based upon irrigation and adoption of new farming technology (2001). LUCITA, or Land Use Changes In The Amazon, was developed to determine the effects of local farming on Amazonian deforestation (Deadman et al., 2004). Time-sequence traffic data for densely populated regions has been used to predict traffic forecasts in Germany, allowing travelers to select the most appropriate route to their destinations (Wahle and Schreckenberg, 2001)

Likewise, single agents have been successful in learning patterns of human behavior from sequential observations. In an attempt to learn human behavior based on emotion in a smart-home setting, Leon et al. created the iSpace and iDorm, test facilities that used sensors to detect physical changes that are associated with certain emotions (2010). They used an analysis of sequential behavioral patterns along with Autoassociative Neural Networks that use physiological responses to predict the emotional state and likely behavior of the smart-home occupant. This allowed the agent to automate some of the smart-home's systems.

ILSA, an agent-based smart-home system designed by Guralnik and Haigh to assist the elderly, uses sensor readings to determine sequences of the occupant's behaviors (2002). It uses the sequential patterns of which sensors fire to determine which times certain activities take place, such as what time a person wakes up, and what time they go to sleep. The researchers that designed ILSA concluded that the order of sensors firing over a time interval was important to learn the behavior patterns of the person living in the home.

MavHome, a smart-home designed by Cook et al., also uses sequential sensor readings for behavior prediction (2003). The goal of MavHome was to adapt to the behavior of its inhabitants by automating processes such as turning up the heat in the morning, or turning on the light and coffee maker after the bedroom alarm goes off. Instead of just using sensors for doors opening and closing, MavHome uses a wide variety of sensors, such as temperature sensors and sensors to monitor the lawn moisture level. Behavior patterns are learned online, and prediction algorithms are used to match patterns in order to determine which devices to operate within the home. Because the learning is online rather than offline, it uses a string compression algorithm, Active LeZi, to compress the behavior sequence and increase the agent's online learning speed. Active LeZi uses a variable order Markov model to predict the probability of the next behavior in the sequence, reducing computational time (Gopalratnam and Cook, 2004).

2.3 SEQUENTIALLY-BASED MODELS OF HUMAN BEHAVIOR IN MULTI-AGENT SYSTEMS

Because of the relative ease of creating more believable results in video games, the behavior models of human NPC's, or non-player-character agents, is already frequently based upon sequential observations. In many First-Person Shooters, as well as in games like *Forza Motorsport* and *Black and White*, opposing agent actions mimic the actions of users playing the game. This provides the player with a challenge more suitable to his ability level and style of play, making it more enjoyable. It also makes the opposing agent's behavior seem more realistic without being too computationally difficult to calculate (van Hoorn et. al, 2009).

More recently, there has been some success in building believable agents in academic competitions. A framework has been developed for the RoboCup simulation league using a combination of observed player behavior and case-based reasoning that makes training the robotic soccer player much more simple (Lam et al., 2006). It learns behaviors from logs of human players' actions. It then turns these into cases to be used in case-based reasoning. These cases are then weighted automatically using a k-nearest neighbor classifier. This algorithm is run until the agent responds to a situation in a simulation in the same way as the previously stored, "test" agents. After the agent has learned a sufficient amount of behaviors from humans, it can then train other agents (Floyd et al., 2008).

BotPrize is a competition using the game *Unreal Tournament 2004* that uses the Gamebots system (Adobbati et al., 2001). It is held every year, and is a DeathMatch First-Person Shooter type game. Human judges play the game along with the bots, and try to distinguish between human and agent players. The UT^2 bot from the Neural Networks Group at the University of Texas, Austin, originally used recorded human behavior sequences to navigate when the bot became stuck. This bot placed second in 2010 (Karpov et al., 2012). After the 2010 competition, they increased the role of human behavior imitation in the evolution of combat behaviors and in navigation (Schrum et al., 2011). Subsequently, they fooled more than 50% of the judges in the 2012 competition, tying for first place with another bot that also mirrored human behavior.

As agents based upon sequential observations become more believably human, it can clearly be seen that imitation is a valid way of representing human behavior. While it would be computationally difficult to model all possible behaviors in a simulation that is very open-ended, many simulations have a limited number of possible actions available to the agent, making imitation a good way of providing a believable initial decision process (Umarov et al., 2012). Automation of the actual calculation of transitional probabilities between behaviors would allow more time for the modeler to make adjustments as needed, making modeling easier and providing a more accurate, believable end-product.

CHAPTER 3

PROPOSED RESEARCH

THESIS GOAL:

As the behavior process that we are attempting to reproduce is comprised of a finite sequence of behaviors, we cannot use a standard MDP. Instead, we will use a Sequential Compressed Markov Decision Process, or SCMDP, which is an MDP with start and end states, and with the ability to account for behavior cycles. Our goal is to design an algorithm that will, given a text file containing sequential observations of behaviors, automatically produce a standardized-format text file of an SCMDP that can generate that behavior process. We will first check for reliability of the model by making sure that it can derive all of the behaviors used in its creation. We will then create a NetLogo tool that will take the text file as input, and allow the modeler to alter the SCMDP and rewrite the text file with the new information. To show that the SCMDP is easily parsed due to its standardized format, we will create example simulations in both NetLogo and Unity that use the same SCMDP text file. These simulations will be used to show the face validity of the simulations, or that the behavior generated by the simulations is believable. To test the face validity of the simulation, we will show it to approximately 10 people in a blind test, asking them to differentiate between one of the original, real-world behavior sequences and the behavior produced by the SCMDP. We will then expand the algorithm to include the possibility of producing more than one SCMDP using classification based on compressed workflows.

Plan Of Work:

To reach our goal of creating one or more believable agents based on the automated analysis of sequential behavior, we have the following projects:

The MAGIC (Models Automatically Generated from Information Collected) Algorithm:

The data we are using to create the decision process for each agent is sequential data. Therefore, a variation of the Markov Decision Process, which we will refer to as a Sequential Compressed Markov Decision Process, or SCMDP, appears to be the best method, as the MDP is inherently sequential, as shown by Papadimitriou and Tsitsiklis (1987). To create the SCMDP, the collected data must be analyzed statistically. Calculating this by hand can be difficult and time-consuming, particularly if the expert that requires the model is not the model's creator. The MAGIC algorithm automates sequential data analysis and creates an SCMDP for the agent, allowing the modeler to focus on the implementation of the behaviors themselves rather than the creation of the actual decision process.

In the MAGIC algorithm, the sequential data is read from a text file as strings of behaviors, or sequences of behavior observations, as shown in Figure 1 below, taken from the nursing administration simulation:

room	726	standard	enter_room		
room	726	standard	clean_hands2		
room	726	standard	login_to_mobile_computer		
room	726	standard	scan_patient_id		
room	726	standard	review_patient_computer_record		
room	726	standard	review_patient_med_box		
room	726	standard	prepare_meds_for_admin		
room	726	standard	scan_patient_meds		
room	726	standard	scan_patient_meds		
room	726	standard	perform_preadmin_assesments		
room	726	standard	dispose_of_trash		
room	726	standard	prepare_meds_for_admin		
room	726	standard	explain_meds		
room	726	standard	administer_meds		
room	726	standard	dispose_of_trash		
room	726	standard	document_med_admin		
room	726	standard	review_patient_computer_record		
room	726	standard	assess_patient_for_other_needs		
room	726	standard	logoff_computer		
room	726	standard	move_computer_out		
room	726	standard	clean_hands2		
room	726	standard	bid_farewell_patient		
room	726	standard	move_to_next_room		
room	726	interrupti	inquiry_patient		
hallway	726	standard	document_post_administration		

Figure 1: Data Used in the Nursing Administration Simulation

These behaviors were observed by students following nurses during their rounds over a period of three months and recorded using an iPad app. The results were combined into one comma delimited text file. The start and end of each sequence is determined by the associated room number. The behavior string is made up of an ordered list of the behaviors in a given sequence.

Because the behavior strings are of finite length, and some behaviors are unlikely to occur first or last, start and end states are added to each behavior string. Groups (or cycles) of behaviors that occur together repeatedly are regarded as one behavior state, and the string is changed accordingly to avoid miscalculation of transitional probabilities. Each behavior is regarded as a state, and the transitional probabilities from one state to the next are calculated using the Markov assumption of conditional independence. Cycle checking and transitional probability calculations are done using a dynamic programming approach, and a file is then output with the SCMDP states and transitional probabilities written in a standardized format that we will refer to as ABML, or Agent-Based Modeling Language. The specifics of ABML have yet to be determined and form part of the work proposed for this thesis. The current version of the algorithm is in Figure 2, below:

MAGIC (File filename, Int maxTaskRepeat)

```
Open File filename
Create List of taskNames
Read File filename
ForEach task in File filename:
        If task is not in List of taskNames, add task to List of taskNames
Create List of behaviorStrings
While not end of file:
        Read task
        If task is first in behavior string, add "start" to behavior String
        Add task to behaviorString
        If task is last in behaviorString, add "end" to behaviorString
Create list of cycles
Foreach behaviorString in behaviorStrings:
        Foreach cycle in cycles:
        If behaviorString contains cycle:
                Replace cycle in behaviorString with "CYCLE" + cycle number
        substringLength ←2
        While substringLength <= maxTaskRepeat:
                i←0
                while i<length(behaviorString)-substringLength-1
                        substring ← behaviorString[i:substringLength-1]
                        if substring contains "CYCLE":
                                i←i+1
                                continue
                        count ← number of times behaviorString contains substring
                        if count > 2:
                                add substring to List of cycles
                                replace substring in behaviorString with "CYCLE" + cycle number
```

```
                    i←i+1
               substringLength←substringLength + 1
Create SCMDP[state][transitionState] where the number of states and transitionStates are the
number of tasks in the taskList + the number of cycles in the cycleList
Foreach behaviorString in behaviorStrings:
     Foreach behavior in behaviorString:
          Add 1 to the corresponding transitionState
Foreach state in SCMDP:
     total ← 0
     Foreach transitionState in state:
          total←total + transitionState
     Foreach transitionState in state:
          transitionState←transitionState / total
Print SCMDP to text file
```

Figure 2: MAGIC Algorithm

The current output of the AMBL file follows the format:

[“State name” “State to transition to” Transitional probability “State to transition to” Transitional probability …]

The following in Figure 3 is an example of the current AMBL text output using the MAGIC algorithm and the data from the Nursing Simulation:

["other_care" "other_care" 0.048387 "close_med_box" 0.123656 "document_post_administration" 0.166667 "document_med_admin" 0.193548 "Cycle0" 0.198925 "Cycle3" 0.220430 "Cycle5" 0.258065 "login_to_mobile_computer" 0.279570 "move_computer_out" 0.301075 "Cycle19" 0.322581 "Cycle10" 0.338710 "Cycle16" 0.344086 "Cycle15" 0.365591 "scan_patient_meds" 0.370968 "scan_patient_id" 0.430108 "perform_assessment" 0.526882 "obtain_meds_pyxis" 0.537634 "review_patient_computer_record" 0.580645 "explain_meds" 0.586022 "dispose_of_trash" 0.596774 "review_patient_med_box" 0.688172 "move_to_next_room" 0.704301 "review_information" 0.709677 "setup_for_med_admin" 0.720430 "Cycle25" 0.725806 "bid_farewell_patient" 0.806452 "Cycle24" 0.822581 "clean_equipment" 0.827957 "assess_patient_for_other_needs"

0.865591 "administer_meds" 0.892473 "clean_hands" 0.967742
"prepare_meds_for_admin" 1.000000]
["close_med_box" "other_care" 0.064935 "close_med_box" 0.069264
"obtain_meds_medroom" 0.082251 "document_post_administration" 0.134199
"document_med_admin" 0.242424 "prepare_meds_for_admin" 0.264069 "Cycle3"
0.268398 "Cycle5" 0.311688 "login_to_mobile_computer" 0.316017
"move_computer_out" 0.341991 "Cycle19" 0.354978 "Cycle12" 0.359307 "Cycle14"
0.367965 "Cycle15" 0.385281 "scan_patient_meds" 0.402597 "other" 0.419913
"scan_patient_id" 0.428571 "perform_assessment" 0.571429 "obtain_meds_pyxis"
0.601732 "review_patient_computer_record" 0.645022 "explain_meds" 0.658009
"logoff_computer" 0.662338 "review_patient_med_box" 0.692641 "review_information"
0.696970 "put_on_gloves" 0.714286 "obtain_meds_outside" 0.718615 "Cycle22"
0.731602 "Cycle25" 0.740260 "bid_farewell_patient" 0.787879 "clean_equipment"
0.805195 "access_computer" 0.809524 "assess_patient_for_other_needs" 0.848485
"administer_meds" 0.926407 "clean_hands" 0.987013 "setup_for_med_admin"
1.000000]
["scan_nurse_badge" "review_patient_computer_record" 0.090909
"perform_assessment" 0.181818 "perform_preadmin_assesments" 0.272727 "Cycle6"
0.363636 "scan_patient_id" 0.454545 "enter_room" 0.545455
"login_to_mobile_computer" 1.000000]

Figure 3: Data Produced Using the MAGIC Algorithm

THE MAGIC BAG (MAGIC BEHAVIOR ADJUSTMENT GRAPH) TOOL:

The MAGIC BAG tool will be created in NetLogo, which provides a simple interface that is easy for end-users to understand. The MAGIC BAG loads and parses a text file written in ABML and turns it into a graph, allowing the SCMDP's behavior states and transitional probabilities to be adjusted more easily by those who are less familiar with programming by using buttons and simple drag-and-drop commands. It then outputs a file in ABML to be used in an actual simulation. A screenshot of the MAGIC BAG tool in its current state is shown in Figure 4 below:

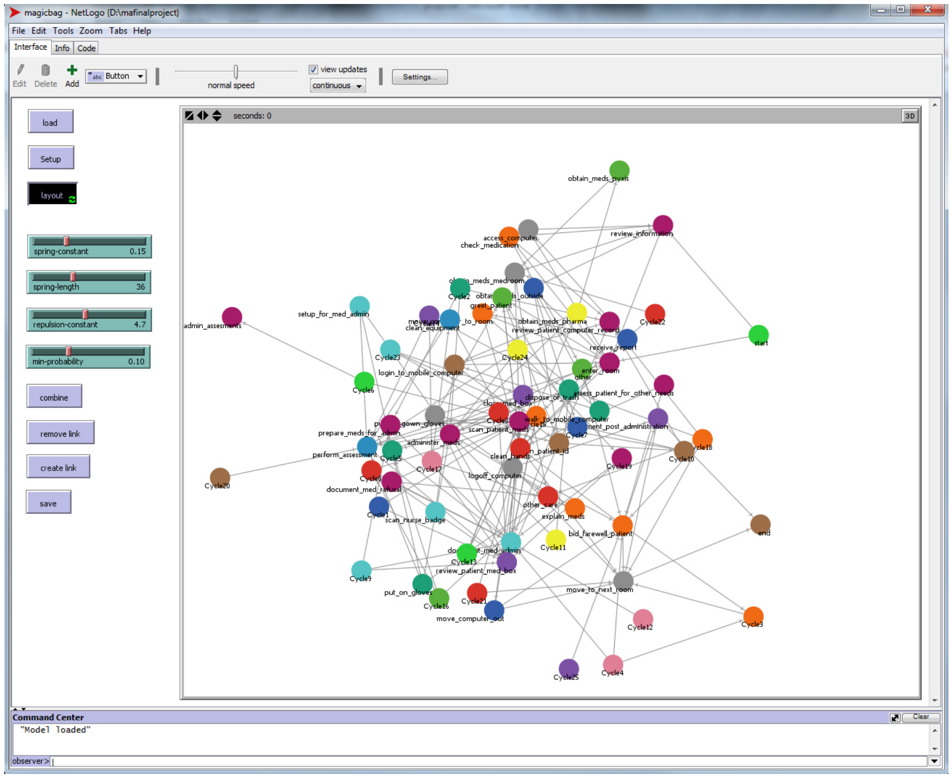

Figure 4: The MAGICBAG Tool

This tool currently allows end-users to adjust the minimum probability for a transition to be allowed, eliminating any transitions under the minimum. It also allows the creation of new transitions between states, direct elimination of transitions, and the combination of two states into one. The resulting graph can then be saved as a text file.

THE MULTI-MAGIC ALGORITHM:

There are times when one collection of behavior sequence data incorporates the behaviors of individuals with different behavior styles or roles, and it would be good to have more than one MDP+ in order to distinguish between those roles. The Multi-

MAGIC+ algorithm allows the user to specify the number of SCMDP's to be produced by the algorithm. It uses compression on the behavior strings, removing behaviors that are repeated in sequence in order to create the most compact version of the workflow. It then creates a dummy workflow and uses edit distance (Liu et al., 2007) with the k-nearest neighbor method in order to cluster the behavior sequences. Each cluster is then processed as a separate SCMDP.

TEST PHASE ONE:

A series of approximately 100 random finite strings of characters from 5 to 15 characters in length will be generated and saved as a text file, as shown in Figure 5 below.

```
f,a,c,c,b,b
a,a,d,d,e,b,a,f,a,a,c,b
c,b,f,b,f,f,b,d,a,c
e,a,a,b,d,b,a,c,d,a,b,d,b
b,a,f,a,e,a,a,e,b
e,b,e,c,b,f,a
c,e,e,d,e,e,d,f,a,b,d,e,c,a,c
c,d,e,a,f,c,c,a
e,b,e,c,d,e,b,b,f,d
b,d,f,f,c,f,e
d,e,c,d,f,f,a,a,e,c,d,f,e,e,a
f,a,f,c,b,f,e,d,d
b,e,c,f,d,d,a,c,f,c,e,c
d,d,f,c,e
```

Figure 5: Randomly generated character strings for Test One

These characters will be used to represent observed behaviors. The MAGIC algorithm will then be used to create an SCMDP from this text file. The resulting

SCMDP will be checked against the initial strings of characters to see if they can be reproduced using that SCMDP. The number of strings that can be reproduced will be divided by the total number of strings to give a reliability percentage for the model.

Test Phase Two:

Simulation One: Nursing Administration

The nursing administration simulation focuses on nursing behavior in a hospital setting. Data is obtained by the observation of nurses during their rounds. The simulation is a 3D version of an actual hospital floor, built using the hospital's floor map. Each room contains only one patient. The patient has an attached script with variables for the number of medications that the patient requires, whether or not the patient requires special medications, and whether or not the patient requires some other type of care. The nurse agent uses an SCMDP logic controller to determine the most appropriate behavior given the available data.

Simulation Two: Canine Social Interaction

The canine social interaction simulation will focus on interaction between domestic dogs in a dog park setting. Data will be obtained from observations made by animal behavioral scientists. These observations will be analyzed using the Multi-MAGIC algorithm to create different styles of canine behavior. The resulting canine agent interaction will then be observed and tested for believability.

Simulation Three: Gamer Behavior

A simple shooter-style game will be created where all player actions are logged in a text file. A group of approximately 10 people will play the game, and their actions will be analyzed using the MAGIC algorithm. The resulting SCMDP will be used to create a bot that will imitate the play style of the human players. The human and bot games will both be recorded, and a different group of approximately 10 people will be asked to determine which is the player and which is the bot.

APPROXIMATE TIMELINE

- Design MAGIC Algorithm: mostly completed (completion by Dec 2013)
- Check internal validity: partially completed (completion by Dec 2013)
- Create MAGIC BAG NetLogo tool: partially completed (completion by Dec 2013)
- Create example simulations: one created but not fully implemented, plans for 2 more (completion by May 2015)
- Design of Multi-MAGIC Algorithm (completion by May 2015)

REFERENCES

Adobbati, Rogelio, Andrew N. Marshall, Andrew Scholer, Sheila Tejada, Gal A. Kaminka, Steven Schaffer, and Chris Sollitto. "Gamebots: A 3d virtual world test-bed for multi-agent research." In *Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS*, vol. 5. Montreal, Canada, 2001.

Bechara, Antoine, Hanna Damasio, and Antonio R. Damasio. "Emotion, decision making and the orbitofrontal cortex." *Cerebral cortex* 10, no. 3 (2000): 295-307.

Bell, William H., David G. Cameron, Ruben Carvajal-Schiaffino, A. Paul Millar, Kurt Stockinger, and Floriano Zini. "Evaluation of an economy-based file replication strategy for a data grid." In *Cluster Computing and the Grid, 2003. Proceedings. CCGrid 2003. 3rd IEEE/ACM International Symposium on*, pp. 661-668. IEEE, 2003.

Berger, Thomas. "Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis." *Agricultural economics* 25, no. 2-3 (2001): 245-260.

Bonabeau, Eric. "Agent-based modeling: Methods and techniques for simulating human systems." *Proceedings of the National Academy of Sciences of the United States of America 99*, no. Suppl 3 (2002): 7280-7287.

Cohen, Philip R., and Hector J. Levesque. "Intention is choice with commitment." *Artificial Intelligence* 42, no. 2 (1990): 213-261.

Cook, Diane J., Michael Youngblood, Edwin O. Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. "MavHome: An agent-based smart home." In *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*, pp. 521-524. IEEE, 2003.

Deadman, Peter, Derek Robinson, Emilio Moran, and Eduardo Brondizio. "Colonist household decisionmaking and land-use change in the Amazon Rainforest: an agent-based simulation." *Environment and Planning B* 31 (2004): 693-710.

Duffy, John. "Agent-based models and human subject experiments." *Handbook of computational economics* 2 (2006): 949-1011.

Garson, G. David. "Computerized Simulation in the Social Sciences A Survey and Evaluation." *Simulation and Gaming* 40, no. 2 (2009): 267-279.

Georgeff, Michael, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. "The belief-desire-intention model of agency." In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pp. 1-10. Springer Berlin Heidelberg, 1999.

Gigerenzer, Gerd, and Wolfgang Gaissmaier. "Heuristic decision making." *Annual review of psychology* 62 (2011): 451-482.

Ginsberg, Matthew L., and David E. Smith. "Reasoning about action I: A possible worlds approach." *Artificial intelligence* 35, no. 2 (1988): 165-195.

Gopalratnam, Karthik, and Diane J. Cook. "Active LeZi: An incremental parsing algorithm for sequential prediction." *International Journal on Artificial Intelligence Tools* 13, no. 04 (2004): 917-929.

Guralnik, Valerie, and Karen Zita Haigh. "Learning models of human behaviour with sequential patterns." In *Proceedings of the AAAI-02 workshop "Automation as Caregiver*, pp. 24-30. 2002.

Floyd, Michael W., Babak Esfandiari, and Kevin Lam. "A case-based reasoning approach to imitating RoboCup players." In *21st International Florida Artificial Intelligence Research Society Conference*, pp. 251-256. 2008.

Hintikka, Jaakko. "Individuals, possible worlds, and epistemic logic." *Nous* 1, no. 1 (1967): 33-62.

Jiang, Hong, Jose M. Vidal, and Michael N. Huhns. "EBDI: an architecture for emotional agents." In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, p. 11. ACM, 2007.

Karpov, Igor V., Jacob Schrum, and Risto Miikkulainen. "Believable Bot Navigation via Playback of Human Traces." In *Believable Bots*, pp. 151-170. Springer Berlin Heidelberg, 2012.

Konolige, Kurt. "What awareness isn't: A sentential view of implicit and explicit belief." In *Proceedings of the 1986 conference on Theoretical aspects of reasoning about knowledge*, pp. 241-250. Morgan Kaufmann Publishers Inc., 1986.

Lam, Kevin, Babak Esfandiari, and David Tudino. "A scene-based imitation framework for robocup clients." *MOO-Modeling Other Agents from Observations*(2006).

Leon, Enrique, Graham Clarke, Victor Callaghan, and Faiyaz Doctor. "Affect-aware behaviour modelling and control inside an intelligent environment." *Pervasive and Mobile Computing* 6, no. 5 (2010): 559-574.

Liu, Jia Jie, G. S. Huang, Y. L. Wang, and R. C. T. Lee. "Edit distance for a run-length-encoded string and an uncompressed string." *Information Processing Letters* 105, no. 1 (2007): 12-16.

Martınez-Miranda, Juan, and Arantza Aldea. "Emotions in human and artificial intelligence." *Computers in Human Behavior* 21, no. 2 (2005): 323-341.

Papadimitriou, Christos H., and John N. Tsitsiklis. "The complexity of Markov decision processes." *Mathematics of operations research* 12, no. 3 (1987): 441-450.

Priesterjahn, Steffen. *Online imitation and adaptation in modern computer games*. Südwestdeutcher Verlag Für Hochschulschriften, 2008.

Railsback, Steven F., Steven L. Lytinen, and Stephen K. Jackson. "Agent-based simulation platforms: Review and development recommendations." *Simulation* 82, no. 9 (2006): 609-623.

Schrum, Jacob, Igor V. Karpov, and Risto Miikkulainen. "UTˆ 2: Human-like behavior via neuroevolution of combat behavior and replay of human traces." In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pp. 329-336. IEEE, 2011.

Schrum, Jacob, and Risto Miikkulainen. "Evolving multi-modal behavior in NPCs." In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pp. 325-332. IEEE, 2009.

Seymour, Ben, and Ray Dolan. "Emotion, decision making, and the amygdala." *Neuron* 58, no. 5 (2008): 662-671.

Sharada, G., and O. B. V. Ramanaiah. "An Artificial Intelligence Based Neuro-Fuzzy System with Emotional Intelligence." *International Journal of Computer Applications* 1, no. 13 (2010): 74-79.

Sloman, Aaron. "Beyond shallow models of emotion." *Cognitive Processing: International Quarterly of Cognitive Science* 2, no. 1 (2001): 177-198.

Stanislaw, Harold. "Tests of Computer Simulation Validity: What Do They Measure?." *Simulation and Gaming* 17, no. 2 (1986): 173-91.

Umarov, Iskander, and Maxim Mozgovoy. "Believable and effective AI agents in virtual worlds: Current state and future perspectives." *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)* 4, no. 2 (2012): 37-59.

Van Hoorn, Niels, Julian Togelius, Daan Wierstra, and Jürgen Schmidhuber. "Robust player imitation using multiobjective evolution." In *Evolutionary Computation*, 2009. CEC'09. IEEE Congress on, pp. 652-659. IEEE, 2009.

Velásquez, Juan D. "Modeling emotions and other motivations in synthetic agents." In *Proceedings of the National Conference on Artificial Intelligence*, pp. 10-15. John Wiley & Sons LTD, 1997.

Wahle, Joachim, and Michael Schreckenberg. "A multi-agent system for on-line simulations based on real-world traffic data." In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pp. 9-pp. IEEE, 2001.