24th Conference on Behavior Representation in Modeling and Simulation (BRiMS)

http://cc.ist.psu.edu/BRIMS2015/

BRiMS was co-located with the International Social Computing, Behavioral Modeling and Prediction (SBP) Conference

at the University of California in Washington, DC, USA

from March 31, 2015 to April 3, 2015
<table>
<thead>
<tr>
<th>Page</th>
<th>Section / Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I. Paper Presentations</td>
</tr>
</tbody>
</table>
| 2    | A Process for Developing Accurate Kinesic Cues in Virtual Environments  
From: Stephanie J. Lackey, Karla A. Badillo-Urquiola, Eric C. Ortiz, & Irwin L. Hudson |
| 10   | Cognitive Leadership Framework using Instance-based Learning  
From: Noam Ben-Asher, Jin-Hee Cho, & Sibel Adalı |
| 18   | Malware Identification Using Cognitively-Inspired Inference  
From: Robert Thomson, Christian Lebiere, Stefano Bennati, Paulo Shakarian, & Eric Nune |
| 26   | Extending Generative Models of Large Scale Networks  
From: Eli Stickgold, Bruce Skarin, Ian Stewart, & Corey Lofdahl |
| 35   | How Could Cyber Analysts Learn Faster and Make Better Decisions?  
From: Fangting Wu & Cleotilde Gonzalez |
| 43   | Semantic Labeling of Objects in a Simulated Environment  
From: Ralph W. Brewer, MaryAnne Fields, Arne J. Suppé, & Dave Wagner |
| 51   | Agents and Decision Trees from Microdata  
From: John B. Nelson, William G. Kennedy, & Ariel M. Greenberg |
| 59   | Long-Term Dementia Care: Modeling the Decision Process  
From: Emily S. Iharaa, William G. Kennedy, Catherine J. Tompkins, & Michael E. Wolf-Branigin |
| 66   | An Emotion and Temperament Model for Cognitive Mobile Robots  
From: Lyle N. Long, Troy D. Kelley, & Eric S. Avery |
II. Poster Presentations

Patterns of Life in the Foreground and Background: Practical Approaches to Enhancing Simulation-Based Interaction Skills Training
Robert Hubal, Jeremiah Folsom-Kovarik, Angela Woods, Randy Jones, & Jon Carbone

Predicting Trust Dynamics and Transfer of Learning in Games of Strategic Interaction as a Function of a Player’s Strategy and Level of Trustworthiness
Michael Collins, Ion Juvina, Gary Douglas, & Kevin Gluck

Crowd Social Network Metrics and Measures for Non-lethal Weapon Effectiveness Testing
Elizabeth Mezzacappa, Gordon Cooke, Kenneth Short, Charles Sheridan, Gladstone Reid, Kevin Tevis, Robert DeMarco, Nasir Jaffery, & John Riedener

Group-Based Transactive Memory to implement Computationally-Plausible Social Cognition in Agents
Geoffrey P. Morgan, Kenneth Joseph, & Kathleen M. Carley

Role of Information Asymmetry in a Public Goods Game for Climate Change
Medha Kumar, Rohit Chouhan, & Varun Dutt

Cognitive and Probabilistic Models of Group Decision Making
Yuqing Tang, Christian Lebiere, Katia Sycara, Don Morrison, & Paul Smart

Individual Differences in Information Processing in Networked Decision Making
David Hughes, Jin-Hee Cho, Sibel Adali, & Jennifer A. Mangels

III. Tutorial

Introduction to Dynamic Network Analysis
Geoffrey P. Morgan, William Frankenstein, & Kathleen M. Carley

IV. Plenary Speaker

Commander Joseph P. Cohn, Office of the Secretary of Defense

BRIMS Conference Committee 2015

Conference Co-Chairs
Daniel N. Cassenti, U.S. Army Research Laboratory
William G. Kennedy, George Mason University

Program Committee Chairs:
David Reitter, The Pennsylvania State University
E. Webb Stacy, Aptima, Inc.
Elizabeth Mezzacappa, Army, Target Behavioral Response Laboratory

Treasurer
Tiffany S. Jastrzembski, Air Force Research Laboratory

Web Manager
Tracy Sanders, University of Central Florida

Executive Committee:
Brad Best, Adaptive Cognitive Systems
Brad Cain, Defence Research and Development Canada
Bruno Emond, National Research Council Canada
Coty Gonzalez, Carnegie Mellon University
Brian Gore, NASA
Kristen Greene, National Institute of Standards and Technology
Jeff Hansberger, Army Research Laboratory
Tiffany Jastrzembski, Air Force Research Laboratory
Randolph M. Jones, SoarTech
Troy Kelley, U.S. Army Research Laboratory
Kenneth Kwok, DSO National Laboratories, Singapore
Christian Lebiere, Carnegie Mellon University
Bharat Patel, Defence Science and Technology Laboratory, UK
Michael Qin, Naval Submarine Medical Research Laboratory
Frank E. Ritter, The Pennsylvania State University
Tracy Sanders, University of Central Florida
Venkat Sastry, University of Cranfield
Barry Silverman, University of Pennsylvania
David Stracuzzi, Sandia National Laboratories
Walter Warwick, Alion Science and Technology
Robert E. Wray, SoarTech
Support

Special thanks to the 2015 International Social Computing, Behavioral Modeling and Prediction Conference for their gracious willingness to co-locate with BRiMS 2015.

Our heartfelt thanks to all the reviewers listed below who offered their time and consideration to the paper submissions. We appreciate your hard work.

Shah Jamal Alam
Dustin Arendt
Stephen Blessing
Nathan Bos
Magdalena Bugajska
Brad Cain
Daniel Cassenti
Gordon Cooke
Andrew Crooks
Bruno Emond
Kristen Greene
Ariel Greenberg
James "JR" Hollister
Randolph Jones
Troy Kelley
Bill Kennedy
Jong Kim
Othalia Larue
Christian Lebiere
Lyle Long
Deryle Lonsdale
Andreas Luedtke

Elizabeth Mezzacappa
Geoffrey Morgan
Jonathan Morgan
Nisheeth Srivastava
Jonathan Pfautz
Nguyen Quoc
David Reitter
Frank Ritter
Venkat Sastry
Barry Silverman
John Sokolowski
Webb Stacy
David Stracuzzi
Robert Thomson
Wayne Thornton
Bela Veksler
Melissa Walwanis
Walter Warwick
Elizabeth Whitaker
Robert Wray
Martin K.-C. Yeh
I.
PAPER PRESENTATIONS
A Process for Developing Accurate Kinesic Cues in Virtual Environments

Stephanie J. Lackey
Karla A. Badillo-Urquiola
Eric C. Ortiz
Institute for Simulation and Training
University of Central Florida
Orlando, FL 32826
slackey@ist.ucf.edu, kbadillo@ist.ucf.edu, eortiz@ist.ucf.edu

Irwin L. Hudson
U.S. Army Research Laboratory
The Human Research and Engineering Directorate
Simulation and Training Technology Center
Orlando, FL 32826
irwin.hudson@us.army.mil

Keywords:
Behavior Cue Analysis, Kinesic Cues, Simulation-Based Training, Computer Animation

ABSTRACT: Computer animations exhibit the illusion of movements or actions of virtual agents and assets within a virtual environment display. Two distinct animation categories exist: two-dimensional (2D) and three-dimensional (3D). 2D animation is typically stylized and used primarily for entertainment-based efforts such as cartoons and low-fidelity games. 3D animation is applied to a wider variety of domains (e.g., entertainment games, serious games, and training simulations). A well-designed 3D computer animation enables a realistic representation of action portraying the true context of movement, particularly human gestures (Badler, Palmer, & Bindiganavale, 1999). All humans convey intent whether purposefully or not via verbal and non-verbal cues (Bavelas, 1990; Givens, 2002). Kinesic cues convey information to an observer through body language and gestures. Emerging research in training human threat detection requires virtual agents exhibiting kinesic cues to provide visual stimuli within Simulation-Based Training (SBT) applications. Thus, guidelines and specifications for system developers are required. This paper presents a process for defining, designing, and animating kinesic cues using a commercially available software application to mimic realistic human behaviors, movements, and gestures. Through this discussion, culturally agnostic kinesic cues are presented, and relevant limitations are identified. The process described and lessons learned represent a logical progression in the formalization of developing advanced visual models for training Warfighters, law enforcement agents, and first responders to detect and classify human threats.

1. Introduction

Reading the human terrain is a critical skill for Warfighters, law enforcement, and first responders. Detection and classification of explicit and implicit behavioral cues relies on observation and perception of visual cues such as gestures, facial expressions, and posture. Furthermore, timely detection and classification of behavioral cues enables Warfighters to proactively identify potential threats within a given environment (Flynn, Sisco, & Ellis, 2012). In order to make proactive tactical decisions, understanding the typical day-to-day operations of an environment, or baseline, is required to efficiently and effectively identify anomalies (see Figure 1.1).

Combat Profiling serves as the foundation for much of the previous behavior cue analysis research, development, and training funded by the U.S. Armed Forces (Gideons, Padilla, & Lethin, 2008; Van Horne, [Figure 1.1 Behavior Cue Detection (Badillo-Urquiola, Lackey, & Ortiz, 2013)](image-url)
The core concepts of Combat Profiling are being extended to enhance training of active duty Warfighters, law enforcement officers, and civilians (Van Horne, 2013), and the associated training methods are evolving, as well. Historically, training such skills occurred in traditional classroom paradigms (Gideons, Padilla, & Lethin, 2008), but recent research and development efforts are moving behavior cue detection into SBT environments (Schatz, Wray, Folsom-Kovarik, & Nicholson, 2012; Wang-Costello, Tarr, & Marraffino, 2013). Improving the effectiveness and distribution of this type of training requires advancing both training methods and technologies. One of the remaining challenges faced by the SBT community is to define requirements for visually representing behavior cues in virtual environments.

This research aims to clearly articulate requirements for 3D animations of behavior cues in order to directly support ongoing research investigating the training efficacy of traditional SBT strategies (e.g., highlighting, massed exposure, etc.) and game-based principals (e.g., leader boards, tokens, etc.). Thus, a Behavior Cue Catalog (BCC) resulted from a multi-year effort to theoretically define, describe, categorize, and prototype 3D models of biometric and kinesic cues. For the 12 kinesic cues presented in this paper, the theoretically based definition, explanation of possible cause(s), and verbal and visual guidance documented in the Behavior Cue Catalogue is summarized. (For a detailed description of the biometric research found in the BCC see Lackey, Badillo-Urquiola, and Ortiz, 2014.)

Before discussing the research approach and results, it is important to provide background information about the field of Kinesics. Kinesics is the interpretation of non-verbal cues such as body language, gestures, postures, and facial expressions displayed by individuals (Birdwhistell, 1970; Leathers, 1997). Four key categories have been identified for classifying kinesic cues: manipulators, illustrators, regulators, and emblems (Ekman, 1992; Ekman, 2004; Leathers, 1997). Manipulators typically indicate an effort to soothe oneself. One part of the body massages, rubs, pinches, scratches, or manipulates another part of the body to relieve stress or discomfort. In some circumstances, manipulators can also indicate a state of relaxation. Illustrators offer additional context to other modes of communication. This type of cue may be confused with emblems; however, illustrators facilitate the listener’s understanding of messages (Ekman, 1992). Illustrators are used to emphasize a word or phrase and when something is difficult to explain; however, they have no meaning separate from context of speech (Harrington, 2009). Behavioral cues that monitor or control the listener of a conversation are known as regulators. Regulators serve as conversational mediators (Ekman, 2004). Finally, emblems represent culturally specific cues understood by every individual in a culture group and do not need words to have meaning (Ekman, 1992). Illustrators may be confused with emblems (i.e., culturally specific cues); however, they facilitate the listener’s understanding of messages (Ekman, 1992). Although illustrators typically emphasize a word or phrase when something is difficult to explain, they have no meaning separate from context of speech (Harrington, 2009). This paper focuses on universal behavior cues (Navarro & Karlins, 2008), thus examples of emblems will not be addressed in detail, but the concept is listed here for completeness.

2. Approach

A systematic, four step approach was used to create the kinesic models depicted in the BCC. The first step was to document design specifications based upon a literature review and Subject Matter Experts (SME) input. Second, a 3D artist created sample models depicting the cues defined. Next, preliminary validity testing confirmed adherence to specifications or identified areas of improvement for each model. Finally, the models and design documentation were modified per the test results. The next section provides relevant details from each phase.

2.1 Define Requirements and Design Specifications

The Non-Verbal Dictionary of Gestures, Signs, and Body Language (Givens, 2002), served as the foundation for understanding kinesic cues, and was augmented by an extensive literature review that referenced a variety of disciplines including: psychology, sociology, communications, criminal justice, political science, health, and medical professions. Historic and contemporary scholarly sources were consulted including books, journal articles, and conference papers, in addition to insight gained from SMEs. Four cue categories emerged: manipulators, regulators, illustrators, and emblems. As previously stated, the first three categories are the focus of the research presented.

Upon completion of the identification and categorization phase, the research team consulted with SMEs to prioritize the cues addressed. Following cue prioritization, the research team defined each cue and developed illustrative descriptions to facilitate 3D modeling efforts. Resulting design specifications include concise definitions culled from the literature, explanations of possible underlying causes, and implications for creating virtual models. The cues included in this paper represent cues that were identified in the literature, prioritized by SMEs, and
could be prototyped using typical commercially available tools. Three cues investigated, but excluded, are described in the limitations section.

2.2 Develop Models

The development phase followed. All 3D models utilized for this effort were purchased and downloaded from a third party digital media reseller. Design parameters (e.g., high-polygon count, high-resolution UV-mapped textures, rigging, and rendering attributes) served as selection criteria. In order to fully understand the animation process employed, a few key terms require explanation.

**High-Polygon Count:** Polygon count refers to the number of polygons used to create a model. Polygons in 3D modeling are simple 2D shapes enclosed and connected to comprise a full model. The more polygons a model contains, the more realistic its appearance in both static and dynamic imagery. Models with high-polygon counts, considered high-fidelity, increase demand on computer processing power in 3D space and Virtual Environments (VE) depending on the number of virtual agents. For the purpose of this effort, the polygon counts for the models utilized averaged between 7,500 and 10,000, and processing power presented no limitations.

**Ultraviolet (UV)-Mapped Textures:** Each model included pre-designed, high-resolution, UV-mapped textures to accurately portray photorealistic human facial features, skins tones, and clothing. Basic wire frames constitute the foundation of all models prior to defining the surface appearance by applying 2D texture maps to UV coordinates. 3D software applications use horizontal and vertical axes called UV coordinates to project 2D images onto the model. Texture maps depict objects such as faces, clothing, or hair. Typically, multiple detailed texture maps shape the appearance of each model. A texture map’s level of detail directly correlates to the precision of a model’s appearance. Photorealistic textured aids in making a model of a human seem more lifelike.

**Rigging:** Rigging fits a solid model with a skeletal system, or “bones,” and manipulators for controlling and posing the model during the animation phase. The number of bones included in a human model varies according to the needs, specifications, and final use of the model. The rigging process defines how the bones shape and move a model. The level of detail required to appropriately represent a particular movement drives the complexity of the rigging process. Model objects called manipulators and controls drive motion and the number of manipulators and controls vary according to the complexity of the movements displayed. Animators use various references such as motion-capture, video footage, or live actors to ensure animations to properly convey realistic actions, timing, and gestures.

**Key Framing:** An industry standard for replicating motion in computer animations, key framing, depicts movement or motion according to start and stop points on a timeline. This process plays a key role in portraying realistic, life-like human movements. Smoothly transitioning from one position/movement to another involves multiple key frame sequences. Each movement possesses its own set of key frames. Each part of the model being animated may have interacting transitions such as raising an arm from a downward position, rotating the wrist, and placing it behind the head. Multiple key frames enable realistic lifelike movements.

**Rendering:** Rendering represents the final step in displaying 3D models and animations on a computer, and refers to the process by which the computer converts models and attributes in a 3D environment into 2D images. Rendering can include individual frames, sequences of frames, or consecutive sequences (i.e., an animation). Different rendering algorithms produce varying results and range from cartoon-like to photorealistic high-fidelity imagery.

**Export:** Following the rendering process, an animation must be exported from the 3D modeling application in order to be imported into the final medium (e.g., game engine or digital movie). Not all 3D software applications use output formats native to the final application. However, third party conversion applications enable streamlined conversions.

**Animation process:** Developing 3D computer animations that accurately portrayed human behaviors, movements, and gestures detailed in the Behavior Cue Catalog required a systematic approach. The following steps summarize the process used by the research team to animate the specified models:

1. Identified individual models possessing the necessary parameters (e.g., high-polygon count, high-resolution UV-mapped textures, rigging, and rendering attributes) and imported models into a commercially available 3D software application (i.e., Autodesk’s 3ds Max).

2. Scaled models in the 3D environment to maintain a standard, uniform size. The 3D software application utilized represented measurements in units. All 3D models were scaled proportionately to a 1 unit = 1 foot ratio. Each male 3D model represented a
human height of 6’0” and each female 3D model represented a human height of 5’7.”

3. Reviewed and test rendered all texture maps to ensure objects associated with each model displayed properly.

4. Observed, at minimum, two males and two females of varying skin tones demonstrating each cue. These visualizations were video recorded and used as references by an artist to animate and replicate each pose on the 3D models.

5. Key framed each model’s motion on a timeline by moving and shifting the rigging manipulators in the 3D space to represent real-world movements. This process was the most time consuming in developing the animations because of the criticality of accurately animating subtle nuances associated with certain gestures (e.g., covering the mouth or hands on hips). The animator replayed each animation several times to ensure each action conveyed the intended message (within the bounds of the technology capabilities and limitations).

6. Rendered each model animation. The rendered output files were produced by the application playing the final animation (e.g., game engine or digital movie). The final renders for this effort were both single frames and animations. The single frames were rendered as high-resolution jpeg images and the animation sequences were Windows movie files. The render sizes or resolution was based upon the final output sizes needed per animation (e.g., whole body or facial features only).

2.3 Preliminary Validity Test and Model Revision

The members of the requirements and specification team analyzed the visual representation of each cue and assessed the accuracy of the behavior exhibited compared to the documented requirements. Discrepancies were noted and shared with the 3D artist to facilitate fine-tuning of each model. For instance, the fingers of the models exhibiting the wring hands cue moved more than expected. The amount of movement for the individual fingers was subsequently minimized as much as possible by the 3D artist. Additionally, the realism of each model was assessed. For example, the hands on the fair skinned female appeared excessively large compared to the relative size of the model’s body, and the fingernails on the medium skin toned male appeared unusually long after the first iteration of modeling. Each feature was adjusted until expected relative sizes were represented.

3. Results

The resulting models presented below provide sample still images of the animations developed. Up to four cues are presented for each cue type (i.e., manipulators, illustrators, and regulators). The results presented focus on the typical visual depiction of each cue as opposed to a specific scenario. In general, the first appearance of a cue or a decreased frequency in the display of a gesture can represent a low level of intensity. Alternatively, a high level of intensity for this effort could be defined as repeated or constant display of a cue. Such decisions depend on scenario details and context; and thus require definition in scenario design documentation.

3.1 Manipulators

Clenched Fists: The hand is closed, with fingers curled and squeezed into palms. This typically signals an emotional state of aggression or anger (Givens, 2002; Lackey, Maraj, Salcedo, Ortiz, & Hudson, 2013). The individual may be unconsciously displaying their anxiety (Givens, 2002). To relieve stress, the individual clenches their fist. See Figure 3.1.

Hand Behind Head: This cue is exhibited by rubbing, scratching, or touching the back of the neck or head with an unclenched hand. Typical causes include a sense of uncertainty or discomfort, because the individual may be in disagreement or have frustrations (Givens, 2002). This gesture typically reflects negative emotions or thoughts (Givens, 2002). To relieve stress, the person touches or rubs the back of their head. Figure 3.2 below demonstrates this cue.
Cover Mouth with Hand: In this case, the individual places their hand in front or on top of the mouth. Someone telling a lie will cover their mouth indicating they are metaphorically trying to hide the lie. This action suggests that the liar is trying to hide themselves from the person they are trying to deceive (Kassin & Fong, 1999). Touching the lips with fingers also induces a sense of relaxation or stress relief (Givens, 2002). See Figure 3.3.

Cover Eyes with Hand: An individual will close or cover their eyes to metaphorically disregard the conversation or ignore the object in front of them. This cue can be considered a protective gesture because the hand is in contact with the face. Avoiding eye contact by closing or covering the eyes prevents anxiety (Blair & Kooi, 2003). See Figure 3.4.

Wring Hands: Squeezing and twisting both hands together suggest a desire to relieve stress and nervousness in an effort to find comfort (Navarro & Karlins, 2008). See Figure 3.5.

Rub Palms: Moving palms back and forth against each other typifies nervousness; rapid rubbing can indicate relaxation or expecting a successful result (Navarro & Karlins, 2008). Refer to Figure 3.6.

3.2 Illustrators

Point Finger: To extend forefinger (also referred to as index finger) in a stabbing motion denotes anger, aggression, hostility, or unfriendly action (Givens, 2002; Navarro & Karlins, 2008). Pointing without a stabbing motion is used to indicate direction or to direct someone’s attention (Givens, 2002). Refer to Figure 3.7.

Slap Forehead: When an individual hits his or her forehead with their hand, it typically indicates that the person is honestly recalling information and telling the truth. This can also be an indicator of uncertainty (Givens, 2002). See Figure 3.8.
3.3 Regulators

*Check Six:* The term Check Six, is short for the phrase “check your six o’clock” (Lackey, Maraj, Salcedo, Ortiz, & Hudson, 2013), may also be referred to as Cut-Off (Givens, 2002). In this case, an individual turns their head to look over their shoulder or turns their entire body around 180° (Lackey, Maraj, Salcedo, Ortiz, & Hudson, 2013). It depicts nervousness, uncertainty, or disagreement. If the individual is displaying a sustained cut-off, it may indicate shyness or disliking (Givens, 2002). See Figure 3.9 for an example.

*Palms Down:* Expression of this cue involves one palm facing downward. Having the palms face downward indicates dominance, confidence, and assertiveness. Palms down can also be accompanied by aggressiveness (Givens, 2002).

*Palms Up:* This cue involves palms facing upward in an open position and indicates thoughtfulness or uncertainty (Givens, 2002).

*Hands on Hips:* Hands on Hips illustrates both phenomena described above. Palms placed onto the hips with elbows flexed outward, away from the body with palms facing downward depicts an assertive attitude (e.g., anger). See Figure 3.10. Alternatively, Figure 3.11 shows palms facing up due to the rotation of the thumbs forward (analogous to facing upward) (Givens, 2002).

4. Discussion

The results presented offer insight into communicating scenario design elements with simulation or model developers. The specifications and visualizations shared offer insight into how to incorporate such requirements into scenario design and test documentation. Ongoing experimentation will determine the efficacy of the kinesic cue specifications reported. However, preliminary experimental results suggest the method and procedure presented positively impact the ability to represent kinesic cues in VEs and SBT. Ultimately, the work presented provides a foundation upon which to build and illustrates the benefits of an interdisciplinary approach to addressing research gaps in the field of behavior cue training development.

5. Limitations

The lack of consensus in the literature concerning the underlying causes of body language and gestures presented a significant challenge. Consulting with SMEs from a relevant task domain, military operations, mitigated this risk. Context plays a critical role in behavior cue diagnosis and analysis. The literature related to culturally agnostic cues and the driving force behind them is limited. However, the research team carefully crafted the definitions presented by balancing tradeoffs between basic and applied research.

Of the cues presented, the Wring Hands cue posed the most difficulty to accurately simulate. Rotation constraints on the wrist and finger joints caused
asynchronous movement of the individual fingers. Minimization of this excessive movement partially addressed this issue but compromised the naturalness of the animation. Three additional cues proved too challenging to sufficiently animate due to technology limitations: Gaze Avoidance, Zygomatic Smile, and Cross Arms. These cues involve multiple, linked attributes or level of detail beyond the capability of standard commercial modeling applications. Research efforts are continuing to confront these complexity challenges.

Finally, the models presented require verification. At the time of writing, SME validation steered animation adjustments. Ongoing human participant experiments using the model specifications presented will serve to remedy this limitation.

6. Conclusion

The purpose of this research was to develop science-driven design requirements and recommendations for accurately modeling kinesic cues in VEs. A natural question that follows is, “Are these models valid?” Empirical validation of the 12 models presented is the first recommendation for future work. Similarly, the question “Are we modeling all of the appropriate cues?” warrants attention. Recommendations for future work include investigating the comprehensive list of cues categorized by domain (e.g., combat, law enforcement, self-defense). Furthermore, the integration of culturally agnostic and culturally specific cues would fuse disparate veins of research: universal behavior cue analysis and human social cultural behavior modeling. In addition to these theoretical research questions, the technical implementation requires attention from the applied research and industry perspective. Advancement of 3D animation capabilities, including detailed modeling of finger, hand, and facial motions, is recommended. Finally, moving this type of behavior representation beyond the contemporary standard in SBT platforms into Virtual Worlds opens up opportunities for large-scale training events. The initial behavior cue design specifications presented herein set the stage for improving Warfighter training in simulation-based environments.

7. Acknowledgements

This research was sponsored by the U.S. Army Research Laboratory – Human Research Engineering Directorate Simulation and Training Center (ARL HRED STTC), in collaboration with the Institute for Simulation and Training at the University of Central Florida. This work is supported in part by ARL HRED STTC contract W91CRB08D0015. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL HRED STTC or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

8. References


**Author Biographies**

**STEPHANIE J. LACKEY** earned her Master’s and Ph.D. degrees in Industrial Engineering and Management Systems with a specialization in Simulation, Modeling, and Analysis at the University of Central Florida (UCF). Her research focused on prediction, allocation, and optimization techniques for digital and analog communications systems. She joined UCF Institute for Simulation and Training’s Applied Cognition and Training in Immersive Virtual Environments Lab in 2008, and assumed the role of Lab Director in 2010. Dr. Lackey leverages her experience in advanced predictive modeling to the field of human performance in order to develop methods for improving human performance in simulation-based training environments and human-robot interfaces.

**KARLA A. BADILLO-URQUIOLA** is a Graduate Research Assistant at the Applied Cognition and Training in Immersive Virtual Environments Lab. In January of 2012, she earned the UCF Undergraduate Researcher of the Month award. Her research interests include: simulation-based training; immersive virtual training environments; physiological & human factors; terrorism and counter-terrorism. As a recipient of the McNair Graduate Fellowship Program, Mrs. Badillo-Urquiola will begin to pursue her Masters of Science in Modeling and Simulation in fall 2014.

**ERIC C. ORTIZ** is a Virtual Learning Environment Developer for the Applied Cognition and Training in Immersive Virtual Environments Lab at the University of Central Florida’s Institute for Simulation and Training. He is currently a student in the Modeling and Simulation Ph.D. program at UCF. Eric is a subject-matter expert in 3D modeling and production pipelines, VE-related software, video production, and interactive technologies.

**IRWIN L. HUDSON** is the Science & Technology Manager responsible for leading the Simulation and Training Technology Center’s Unmanned Ground Systems Research. Mr. Hudson is Contract Officer Representative for several robotic research efforts. He also serves as the Assistant Contract Officer Representative to Dr. Neal Finkelstein for the Research Academic and Operational Support Indefinite Delivery, Indefinite Quantity contract, which supports a large percentage of the research and development budget for STTC’s Blended Simulation Research Branch. He is currently pursuing his Ph.D. in M&S from UCF.
Cognitive Leadership Framework using Instance-based Learning

Noam Ben-Asher  
IBM T.J.Watson Research Center  
Yorktown Heights, NY  
Email: nbenash@us.ibm.com

Jin-Hee Cho  
US Army Research Laboratory  
Adelphi, MD  
Email: jinhee.cho@us.army.mil

Sibel Adalı  
Rensselaer Polytechnic Institute  
Troy, NY  
Email: sibel@cs.rpi.edu

Keywords:  
Cognitive modeling, Trust, Decision Making

ABSTRACT: Leadership has been extensively discussed for a long time in the literature by describing the ideal traits of a leader, emphasizing the positive influence of the good leader on group, team, and organization performance. Although many different types and traits of leadership have been proposed, little work has considered the pivotal fact that the leader has limited capacity and resources (e.g., cognition, time, energy) when dealing with numerous time sensitive tasks in real, dynamic settings. This work is interested in how the leader can provide effective feedback to improve a follower’s team performance for establishing long-term assets in the team, while maximizing the utilization of limited resources. To this end, we adopt the Instance-Based Learning (IBL) cognitive architecture to develop a leadership framework that determines whether to provide feedback to a follower, and if so, how much feedback should be provided depending on the follower’s level of performance readiness. This proposed framework aims to maximize the trust improvement of followers, while minimizing the cost spent in providing feedback to achieve cost-effective decision making. By reflecting the concept of situational leadership, our proposed framework allows the leader to respond adaptively to followers based on their readiness. Our simulation results show that the proposed IBL-based framework provides an appropriate level of adaptive feedback depending on the readiness of a follower given no prior knowledge about the followers, while the leader can make cost-effective decision in feedback provision. Such cognitive leadership framework can be useful as a cognitive inspired decision aid tool for a human leader, or part of an autonomous and adaptive feedback system.

1. Introduction

Leadership is defined as the process of influencing activities of an individual or a group towards goal achievement in a given situation (Hersey, Blanchard, & Johnson, 2001). Leadership has been recognized as the key factor that can lead groups, teams, and organizations to success for centuries. A large body of literature has discussed the characteristics of different leadership styles in order to suggest an ideal leadership to be applied in real management settings (Hersey et al., 2001; Bass 1991; Rowe, Reardon, & Bennis, 1995). The examples include charismatic, transformational, situational, shared, motivational leaderships, among others (Hersey et al., 2001; Bass 1991). However, these concepts are too broad and often do not consider the reality of the leader’s capacity limitations in cognition, time, and energy. Commonly, the leader may face difficulties in dealing with many time-sensitive tasks while being obligated to provide effective feedback to a follower. In such situations, the leader may resort to simple, rigid rule-based decisions that result in suboptimal performance. Further, the leader may not have sufficient prior knowledge about the follower and not be able to make effective and efficient decisions in providing the appropriate feedback based on the follower’s needs and readiness in a specific resource-constrained situation. As such, a successful leader is required to learn from experience and adapt to the dynamics of followers.

In this work, we are interested in how a leader with limited capacity and resources can provide effective and efficient feedback to a follower whom the leader does not have prior knowledge about. To this end, we propose a cognitive leadership framework using the Instance-Based Learning (IBL) architecture to provide adaptive, efficient, and effective feedback to a follower in different stages of performance readiness. The IBL-driven decision making framework aims to provide cost-effective decision making that maximizes the follower’s capacity in terms of trust improvement by the leader’s feedback while minimizing the cost spent by the leader in providing feedback.

Unlike the existing studies on leadership, this work has the following contributions: (1) We propose a cognitive framework for a leader’s decision making in providing feedback to a follower in various stages of performance readiness. IBL is used to model the adaptivity of the decision maker to dynamic settings based on experience. To the best of our knowledge, our work is the first to employ the IBL in modeling the cognitive leadership framework for this type of decision making; (2) Given no prior knowledge of the follower’s performance readiness, our framework enables the leader to dynamically assess trust towards a follower and use this trust assessment to decide whether to provide feedback and how much feedback to provide; (3) We adopt a multidimensional
has been investigated in the area of management and the relationship between leadership and team performance. Inspirational, and supportive attitudes for radical change: commanding, logical, different styles of leaders based on their behaviors and organizational vision. Rowe et al. (1995) derive four types of leaders: transactional leader (e.g., supervisor to control) to a transformational leader (e.g., motivator to share an organizational vision). Bass (1991) considers four leadership styles: telling for followers lacking in the training, confidence, or desire to complete a given task by providing detailed directions; selling for followers with high confidence and willingness but insufficient capability to complete a given task by clarifying directions; participating for followers who are motivated to achieve goals but lack self-confidence by encouraging them to participate in decisions and support efforts; and delegating for followers who are sufficiently capable, confident, and motivated to achieve a task by providing minimum directions. The role of a leader is mainly via two ways of support: relation-oriented feedback (e.g., emotional encouragement) and task-oriented feedback (e.g., clear task instructions). According to the situational leadership theory of Hersey et al. (2001), a leader should be able to use a different leadership style depending on the readiness of a follower.

One of the popular models by Hersey et al. (2001) considers four leadership styles: telling for followers lacking in the training, confidence, or desire to complete a given task by providing detailed directions; selling for followers with high confidence and willingness but insufficient capability to complete a given task by clarifying directions; participating for followers who are motivated to achieve goals but lack self-confidence by encouraging them to participate in decisions and support efforts; and delegating for followers who are sufficiently capable, confident, and motivated to achieve a task by providing minimum directions. The role of a leader is mainly via two ways of support: relation-oriented feedback (e.g., emotional encouragement) and task-oriented feedback (e.g., clear task instructions). According to the situational leadership theory of Hersey et al. (2001), a leader should be able to use a different leadership style depending on the readiness of a follower.

The relationship between leadership and team performance has been investigated in the area of management and information management. Carson and colleagues (2007) examine pre-conditions to enhance shared leadership on team performance. Lee, Lee and Seo (2011) propose a team creativity model by investigating shared leadership, knowledge sharing and cognition-based trust. Ehrlich and Cataldo (2014) report that a team’s productivity and quality increases when team leaders share more information.

Unlike prior work, this work considers a leader’s limited capacity and resources and provides a cognitive leadership framework that delivers efficient and effective decision making strategies.

### 2.2 Cognitive Architectures and Instance-Based Learning

In a dynamic decision making setting, cognitive architectures, such as ACT-R (Anderson & Lebiere, 1998), SOAR (Laird, Newell, & Rosenbloom, 1987) and others, have been commonly used to provide an integrated representation of human cognition. Cognitive models, constructed using these architectures, allow careful examination of cognitive processes that drive human decision making (Gonzalez, Lerch, & Lebiere, 2003). Cognitive models based on IBL theory (IBLT) focus on decision making and learning from experience in dynamic settings (Gonzalez et al., 2003). IBLT emerging from ACT-R, proposes a generic decision-making process that recognizes decision situations, generates instances through the interaction with the decision task, and finishes with reinforcement of the instance leading to desired outcomes. Accordingly, decision making situations are represented as instances stored in memory. An instance is composed of three parts: (1) situation (S) that is a set of attributes representing a situation; (2) decision (D) that is made in the particular situation; and (3) utility (U) that is the experienced outcome following the decision. IBLT decision cycle includes several stages: recognition, judgment, choice, and execution. In Recognition stage, a decision maker identifies relevant attributes for a specific decision situation. Judgment stage determines the relevancy of past experiences (instances) in current decision making situation. The activation of instances in memory may be influenced by the recency and frequency of their occurrences in the past. Memory activation determines the probability that an instance will be retrieved from memory and participated in the next phase. In the absence of previous experiences relevant to the current situation, pre-defined heuristics are triggered for decision making. In the Choice stage, retrieved instances and their retrieval probability are used to calculate expected utility for each of the decision options, then the option with the highest expected utility is chosen. Finally, in the Execution stage, feedback regarding the decision is provided (Gonzalez et al., 2003). We chose IBL to model the decision making to achieve the adaptive, learning human
decision making in dynamic environments as well as the transition between exploration and exploitation (i.e., maximization). For more details on how to formulate IBL models, see (Lejarraga, Dutt, & Gonzalez, 2012).

IBL models have been popularly used in dynamic decision making context where a human decision maker uses memory to retrieve past experience for present decision making. Lejarraga et al. (2012) demonstrate that a single IBL model constructed for a specific repeated binary choice task can be generalized to different variants of repeated tasks requiring a binary decision as well as to probability learning tasks. More specifically, constructed IBL models can reflect human behavior in simple stimulus-response practice and skill acquisition tasks, training paradigms, and mental fatigue (Lejarraga et al., 2012). The experience-based learning process of an IBL model was successfully extended to include descriptive information and biases as risk aversion (Ben-Asher, Dutt, & Gonzalez, 2013). A pair of IBL models successfully consider the dynamics of cooperation in iterated Prisoner’s Dilemma as well as reciprocity and other complex social interactions (Gonzalez, Ben-Asher, Martin, & Dutt, 2014; Gonzalez & Ben-Asher, 2014).

3. Agent Model

3.1 Scenario of Leader-Follower Relationships

We consider a decision making scenario in a military tactical environment with a hierarchical structure of a commander, assistant commanders, and followers by each assistant commander. We concentrate on how a mid-level leader, the assistant commander (A), makes decisions in a situation that requires A to gather and aggregate inputs (e.g., information and opinions) from her/his followers (F), and provides an aggregated opinion to the upper level leader, a commander (L). In this situation, A wants to maximize efficiency of utilizing limited capacity when providing feedback to Fs, while providing useful information for L to make a critical decision. Therefore, both decisions are closely related to mission performance.

To be specific, A needs to make a decision on whether to spend resources in providing feedback to the followers. We classify the follower’s performance in two categories: their citizenship through participation in a task (willingness) and their ability to accomplish specific tasks (competence). As shown in prior literature, a leader can improve the competence of followers through feedback that aims to explain how to accomplish a task. The willingness can be improved through relation oriented feedback such as emotional encouragement. Both types of feedback are known to improve the overall task performance. We model how A can provide feedback to follower F in two steps. We first model each follower’s individual characteristics in the competence and willingness dimensions, and how both characteristics can change as a function feedback from a leader in Section 3.2. As the leader cannot directly access individual’s characteristics, we model the leader’s subjective belief, i.e. trust, for each follower’s performance using the same two dimensions of competence and willingness (reliability), following the universal dimensions of social cognition (Adali, 2013). We show how the leader can form judgments of followers by observing their behavior in the team. Finally, we show how the leader can decide how much feedback to provide based on his own individual characteristics and his trust for a follower.

3.2 Modeling an individual’s characteristics

We consider two aspects of an individual’s specific characteristics: competence and willingness.

**Competence:** Each entity, whether an assistant commander or a follower, has task specific competence such as the ability to discern whether or not the received information (i.e., factoids) is true on a given proposition (i.e., ‘Is there improvised explosive devices (IEDs)’?). The competence $C_k$ represents the probability that entity $k$ will correctly evaluate a piece of evidence. The competence can change based on feedback received from an upper level leader at time $t$, which is modeled by

$$C_k(t) = \min\left(C_k \times (1 + \delta N_{TOF}(t)), 1\right)$$ (1)

where $N_{TOF}(t)$ is the total number of task-oriented feedback (TOF) $k$ received from the upper level leader regarding task execution until time $t$. $\delta$ is a learning rate in the range of $[0, 1]$. Eq. 1 implies that higher competence leads to faster learning.

**Willingness:** Each entity has willingness representing the level of commitment to cooperate by providing or forwarding information to an upper level leader, and achieve task/mission success. We model the willingness $W_t$ of entity $k$ as the probability that an entity will forward a piece of information. The willingness of a follower is affected by the amount of feedback received from the upper level leader at time $t$ and is given by

$$W_k(t) = \min\left(W_k \times (1 + \delta N_{ROF}(t)), 1\right)$$ (2)

where $N_{ROF}(t)$ is the total number of relation-oriented feedback (ROF) such as encouragement $k$ received from the upper level leader until time $t$.

Leader A and followers F each have individual characteristics in both dimensions, which are seeded by an initial value. We only model the changes in these dimensions as a function of direct feedback by the leader to investigate the impact of different strategies for the leader.

3.3 Trust Assessment
As a leader cannot directly access a follower’s competence and willingness, he can only form trust beliefs about these two based on his observations in a task environment. We model these trust beliefs using Bayesian updates of Beta distributions where the mean and variance of the distribution represent the value and the uncertainty regarding the trust belief. We use a separate distribution for each dimension. A more detailed explanation of this formulation can be found in (Chan & Adali, 2012).

The belief $T_{i,j}$ of entity $i$ for entity $j$’s trustworthiness are based on the number of binary positive ($\alpha$) and negative ($\beta$) evidences of $j$’s competence or willingness as observed by $i$. We use the beta-binomial conjugate distributions to capture the dynamic trust evaluations. Each distribution $B(\alpha_0, \beta_0)$ is seeded with initial values of positive and negative evidence. Given the new positive and negative evidence ($r$ and $s$, respectively), the posterior trust distribution is given by $B(\alpha+r, \beta+s)$. The expected (mean) value of trust, $E(T^X)$ for a trust property $X$ (competence or willingness), is given by

$$E(T^X) = \frac{\alpha+r}{\beta+s+r+s}$$  \hspace{1cm} (3)

We omit trustee $j$ (ratee), trustee $i$ (rator), and $t$ (time) notations for simple representation. Based on the above method, the trust $T$ of entity $j$ evaluated by entity $i$ at time $t$ on trust property $X$ (competence or willingness) based on evidence available at time $t$ is denoted by $T^X_{i,j}(t)$.

The belief $T^X_{i,j}$ of entity $i$ for entity $j$’s trustworthiness are based on the number of binary positive ($\alpha$) and negative ($\beta$) evidences of $j$’s competence or willingness as observed by $i$. We use the beta-binomial conjugate distributions to capture the dynamic trust evaluations. Each distribution $B(\alpha_0, \beta_0)$ is seeded with initial values of positive and negative evidence. Given the new positive and negative evidence ($r$ and $s$, respectively), the posterior trust distribution is given by $B(\alpha+r, \beta+s)$. The expected (mean) value of trust, $E(T^X)$ for a trust property $X$ (competence or willingness), is given by

$$E(T^X) = \frac{\alpha+r}{\beta+s+r+s}$$  \hspace{1cm} (3)

We omit trustee $j$ (ratee), trustee $i$ (rator), and $t$ (time) notations for simple representation. Based on the above method, the trust $T$ of entity $j$ evaluated by entity $i$ at time $t$ on trust property $X$ (competence or willingness) based on evidence available at time $t$ is denoted by $T^X_{i,j}(t)$.

### Table 1. Observed trust evidence of $F$ by $A$

<table>
<thead>
<tr>
<th>Competence</th>
<th>No Input</th>
<th>True Input</th>
<th>False Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competence</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Willingness</td>
<td>0</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

While the competence and willingness trust are both modeled by the same type of distribution, the evidence regarding each is collected differently in our model. The willingness evidence is based on whether the leader $A$ (entity $i$) observed the follower $F$ (entity $j$) providing input. In willingness evaluation, the correctness of the input is not considered. The competence evidence is based on whether the input provided by a follower is actually correct. Table 1 summarizes how positive or negative evidence is counted to estimate trust belief of $A$ towards follower $F$. Notice that $A$ does not evaluate $F$’s competence upon no input provided by $F$.

### 3.4 Modeling Leadership and Feedback

We adopt the notion of situational leadership (Hersey et al., 2001) based on the four types of leadership strategies including telling, selling, participating, and delegating. We model the different types of leadership by adjusting the amount of feedback including TOF and ROF, based on the leader’s trust for a follower’s competence and willingness.

TOF is focused on instructions on task execution to improve a follower’s competence while ROF provides emotional support to enhance his/her willingness (Table 2). The amount of TOF ($N^\text{TOF}_{i,j}(t)$) and ROF ($N^\text{ROF}_{i,j}(t)$) provided by leader $i$ to follower $j$ at time $t$ is affected by the two trust attributes (i.e., competence and willingness) and is estimated by

$$N^\text{TOF}_{i,j}(t) = \frac{w_i(t)}{w_{ij}^*(t)} N^\text{ROF}_{i,j}(t) = \frac{w_i(t)}{w_{ij}^*(t)}$$  \hspace{1cm} (4)

Note that a leader’s willingness ($W_i(t)$) affects the amount of feedback provided to the follower, which in turn affects follower’s competence and willingness.

### Table 2. Modeling of leadership and followership styles

<table>
<thead>
<tr>
<th>Leadership styles</th>
<th>Feedback (TOF, ROF)</th>
<th>Followership styles</th>
<th>Follower Trust ($T^w_{i,j}(t), T^p_{i,j}(t)$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telling</td>
<td>(high, high)</td>
<td>Conformist</td>
<td>(low, low)</td>
</tr>
<tr>
<td>Selling</td>
<td>(high, low)</td>
<td>Passive</td>
<td>(low, high)</td>
</tr>
<tr>
<td>Participating</td>
<td>(low, high)</td>
<td>Alienated</td>
<td>(high, low)</td>
</tr>
<tr>
<td>Delegating</td>
<td>(low, low)</td>
<td>Exemplary</td>
<td>(high, high)</td>
</tr>
</tbody>
</table>

### 4. Computational Framework

#### 4.1 IBL-Based Decision Making

We employ an IBL model to provide a decision solution for $A$’s perspective ($i$) for a follower $F$ ($j$) using the Python PyIBL package (http://downloads.ddmlab.com). We describe the unique details of the three components that were constitute the IBL-based feedback mechanism: situation, decision, and utility. For a detailed description and formulation of the general activation, retrieval probability and blending mechanisms of the IBL’s decision making process see Lejarraga et al. (2012).

**Situation ($S_{i,j}$):** a situation that $i$ makes a decision towards $j$ is represented as a vector of attributes including (1) the time left by the deadline, $t_d$, (2) the current trust level of $i$ towards $j$ in terms of willingness and competence ($T^w_{i,j}, T^p_{i,j}$), and (3) a binary indication whether $j$ provided $i$ with information, $a_{i,j}$. This is given by

$$S_{i,j} = \{t_d, T^w_{i,j}, T^p_{i,j}, a_{i,j}\}$$  \hspace{1cm} (5)

All components above are dynamic as discussed earlier: $i$’s estimated trust in $j$ may change as the two entities interact over time and $j$’s capability grows over time as $i$ provides her with feedback. The value of $a_{i,j}$ depends on the action of $j$ in each time point.

**Decision ($D_{i,j}$):** $i$ will provide feedback to $j$ if providing feedback to $j$ in that specific situation resulted in positive outcome (i.e., highest utility) in the past. This decision process involves retrieving relevant instances from $i$’s memory, computing the retrieval probability for each of the instance and, choosing the option that yields the highest expected utility. This process is influenced by the recency and frequency of the experiences, memory decay ($d$) and a
noise parameter for capturing the variability in activation (σ).

In novel situations, where \( i \) does not have any previous experience with a follower, \( i \) will make a rule-based decision where 1 indicates ‘provide feedback’ and 0 is ‘do not provide feedback’ as follows

\[
D_{i,j}(t) = \begin{cases} 
1 & \text{if } (t_d \geq \gamma) \text{ and } c_f(i,j)(t) < \varepsilon \\
0 & \text{otherwise} 
\end{cases} \tag{6}
\]

\( t_d \) indicates the time left by the deadline where each factoid has a respective decision deadline, and \( \gamma \) is a design parameter denoting the minimum time left for the given decision where \( \gamma > 0 \). \( \varepsilon \) represents the maximal amount of resources \( i \) is willing to risk (i.e., spend), when the expected utility from providing feedback is unknown. \( c_f(i,j)(t) \) is the cost for a ‘provide feedback’ decision and is estimated as

\[
c_f(i,j)(t) = \begin{cases} 
\varepsilon & \text{if } N_f(i,j)(t) > 0 \\
0 & \text{otherwise} 
\end{cases} \tag{7}
\]

We assume that each feedback, either TOF or ROF, incurs an equal cost where \( N_f(i,j)(t) = N_f^{ROF}(t) + N_f^{TOF}(t) \) and \( c \) is a constant parameter to convert the number of feedback to the organizational feedback standards. We used the exponential form to calibrate the cost treating the less feedback (i.e., decrease in \( N_f(i,j)(t) \)) incurs high cost similar to a utility function in (Horvitz & Rutledge, 1991).

Utility (\( U_i \)): It is computed based on the outcome after the decision \( D_{i,j} \) is made at time \( t \). The utility is a function of \( i \)’s estimated trust improvement in \( j \), time left until the deadline, \( t_d \), and the cost to provide feedback, \( c_f(i,j)(t) \). \( U_{i,j}(t) \) is computed by

\[
U_{i,j}(t) = \Delta t_{i,j}(t - \Delta t) e^{-\lambda (\frac{1}{2} \sigma^2 + \gamma c_f(i,j)(t))} \tag{8}
\]

Where \( \lambda \) is a constant to adjust the impact of the cost \( c_f(i,j)(t) \) and time factor \( t_d \) on the utility. \( \Delta t_{i,j}(t - \Delta t) \) is \( j \)’s average improved trust in competence and willingness after \( i \) provides \( j \) with feedback at time \( t \) and is computed by

\[
\Delta t_{i,j}(t - \Delta t) = \frac{\Delta t_{i,j}(j - \Delta t) + \Delta t_{i,j}^w(j - \Delta t)}{2} \tag{9}
\]

Similar to the cost function, we used an exponential form following the similar utility function with respect to action and time Horvitz and Rutledge (1991).

4.2 Metrics

To compare the performance of the proposed IBL leadership framework to its counterparts, we introduce the following metrics:

- **Proportion of Feedback** indicates the average ratio of feedback provided to a follower by \( A \).
- **Follower Performance** is the average value of a follower’s competence and willingness, and calculated based on Eqs. 1 and 2 respectively.

- **Cost of Feedback**, \( c_f(i,j)(t) \), is the average cost introduced by \( A \)’s feedback to a follower at time \( t \), estimated based on Eq. 7.
- **Cumulative Utility** is the metric indicating the average accumulated utility from time \( t=0 \) until \( t_m \) the entire mission session time, and computed by

\[
\hat{U}_{i,j}(t) = \int_{t=0}^{t_m} U_{i,j}(t) dt \tag{10}
\]

where \( U_{i,j}(t) \) is the utility obtained by \( i \) by providing feedback to \( j \) at time \( t \) and calculated by Eq. 8.

- **Cumulative Cost** shows the accumulated feedback cost, \( c_f(i,j)(t) \), from time \( t=0 \) until \( t_m \) and estimated by

\[
\hat{C}_{i,j}(t) = \int_{t=0}^{t_m} c_f(i,j)(t) dt \tag{11}
\]

5. Simulation Results and Analysis

5.1 Experimental Setup

To investigate the proposed IBL-based framework, we designed four feedback conditions representing different leader behavior in providing feedback to followers:

- **Constant-feedback**: This condition assumes that \( A \) provides follower feedback all the time inattentive to \( F \)’s action or state.
- **IBL-based feedback**: \( A \)’s feedback decisions are determined based on the proposed IBL model.
- **No-feedback**: \( A \) does not provide any feedback at all.
- **Random-feedback**: \( A \) provides random feedback based on a given probability, \( p=0.5 \).

As shown in Table 2, we experiment under different levels of readiness reflecting the four types of followership: conformist, passive, alienated, and exemplary. We vary the individual characteristics willingness and competence to (.25, .25), (.75, .25), (.25, .75), and (.25, .25) to examine how different types of followers evolve and change over time. Leader’s initial trust towards followers is set at medium level of trust (.5, .5) for both competence and willingness. Thus, we show experimental results with the five different types of followers. In the experiment, followers in the four conditions observed a random sequence of 200 factoids. This was iterated for 100 runs, and for simplicity we set the time left by the deadline with \( t_d=5 \) for all situations. Table 3 summarizes the default values of key design parameters.

### Table 3. Framework parameters and their default values

<table>
<thead>
<tr>
<th>Param.</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Param.</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma ) (IBL noise)</td>
<td>.25</td>
<td>( \varepsilon )</td>
<td>.25</td>
<td>( \lambda )</td>
<td>1</td>
</tr>
<tr>
<td>( t_{w}^A ) (A’s willingness)</td>
<td>.80</td>
<td>( \delta )</td>
<td>.02</td>
<td>( c )</td>
<td>2</td>
</tr>
<tr>
<td>( d ) (IBL decay)</td>
<td>1.5</td>
<td>( \gamma )</td>
<td>2</td>
<td>( \sigma_\beta )</td>
<td>1</td>
</tr>
</tbody>
</table>

5.2 Performance Analysis

In this section, we show our experimental results and analyze the observed general trends based on the
performance comparison of the four different feedback approaches:

**Proportion of Feedback:** Table 4 shows the overall probability that the leader provides *IBL-based feedback* to a follower $F$ across 200 trials. These results indicate that the *IBL-based feedback* is sensitive to the initial willingness and competence of $F$. It is observed that the lower the capability of $F$ is, the higher feedback $A$ provides; and vice-versa. Interestingly, we observe that the competence level of $F$ is the key factor that determines the proportion of feedback provided by $A$, showing $F$'s lower competence triggers high feedback by $A$. The reason is because trust can be maximally improved when the follower provides true input. This leads to increase in both willingness and competence that is reflected in high trust improvement, which is coupled with high utility based on Eq. 8.

**Table 4. Mean and Standard Deviation (SD) for the proportion of feedback provided by the IBL leadership framework**

<table>
<thead>
<tr>
<th>Follower Type</th>
<th>$(w, c)$</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformist</td>
<td>(.25, .25)</td>
<td>.305</td>
<td>.046</td>
</tr>
<tr>
<td>Alienate</td>
<td>(.25, .75)</td>
<td>.199</td>
<td>.024</td>
</tr>
<tr>
<td>Realist</td>
<td>(.50, .50)</td>
<td>.180</td>
<td>.057</td>
</tr>
<tr>
<td>Passive</td>
<td>(.75, .25)</td>
<td>.284</td>
<td>.055</td>
</tr>
<tr>
<td>Exemplary</td>
<td>(.75, .75)</td>
<td>.115</td>
<td>.083</td>
</tr>
</tbody>
</table>

To evaluate the adaptivity of the IBL-based feedback, we investigate how the amount of feedback provided by $A$ for $F$ evolves over time in Figure 1 for the different types of followers over time. Initially, $A$ has no prior knowledge about followers’ performance readiness. Thus, $A$ provides all followers with a similar proportion of feedback that gradually increase. This behavior lasts for about 25 trials where $A$ is able to learn about the capabilities of the followers and their dynamics. Once $A$ acquires some experience with $F$, we see a gradual adjustment of the proportion of feedback to $F$’s needs. For example, the proportion of feedback provided to follower $(w=.75, c=.25)$ and $(w=.75, c=.25)$ received increasing proportions of feedback from $A$. However, we find a gradual decrease in the proportion of feedback as the followers with initial low competence have higher levels of trust. In general, the feedback is provided until the trust improvement is maximized without exceeding the limit of the cost. After about 160 trials, all followers receive similar and relatively low proportions of feedback. Following, the proportions of feedback slightly fluctuate. These fluctuations represent an exploration process of $A$ trying to improve the different followers after they already reached high level of competence, combined with the decay of the old instances.

![Figure 1. Average proportion of IBL-based feedback provided by $A$ for each follower type across trials](image1)

![Figure 2. Effect of feedback on the followers' willingness](image2)

![Figure 2. Effect of feedback on the followers' competence](image3)

**Follower Performance:** Figure 2 shows how $F$’s actual willingness and competence (based on Eqs. 2 and 1) evolves over time under four different feedback conditions, discussed in Section 5.1. *No-feedback* does not change trust while the *Constant-feedback* quickly reaches the maximum trust of 1 in both competence and willingness. *Random-feedback* provides feedback with the probability $p=.5$, thus showing the trust values between *Constant-feedback* and *IBL-based feedback*. Since *IBL-based feedback* only provides feedback under certain conditions of utility and cost, we do not observe the maximal improvement of followers’ performance, rather a consistent increase.

In Figure 2(a), lower initial willingness (i.e., $w=.25$) slows down the learning process while higher initial willingness (i.e., $w\geq .5$) speeds up learning to reach the maximum value of 1. Noticeably, even with *IBL-based feedback*, high initial willingness pushes performance to the maximum value of 1. On the other hand, in Figure 2(b), lower
competence (i.e., \(c=.25\)) slows down the learning process while higher competence (i.e., \(c\geq.5\)) expedites the learning to reach the maximum performance value.

Cost of Feedback: Figure 3 shows the average cost when \(A\) provides feedback to \(F\) based on the feedback cost computed by Eq. 7. No-feedback does not incur any cost while Constant-feedback incurs the highest cost. Random-feedback introduces more cost than IBL-based feedback but less than Constant-feedback. Furthermore, the average costs incurred by Random-feedback are relatively stable overtime, indicating that the amount of feedback (as computed by Eq. 4) is relatively fixed throughout the 200 trials, while IBL-based feedback and Constant-feedback conditions exhibit adjustable amounts of feedback. Out of the three conditions with feedback provided, IBL-based feedback performs best, showing the lowest feedback cost. Except IBL-based feedback, all other feedback approaches maintain the feedback cost over more trials regardless of the \(F\)'s levels of competence and willingness. Furthermore, IBL-based feedback is quite sensitive to a different follower type, showing more adaptive feedback to less capable followers with low willingness and competence. This proves adaptive nature of feedback achieving the situational leadership which provides feedback based on the degree of a follower’s readiness. Similar to the previous observations in Figure 1, low competence introduces more feedback.

Cumulative Utility & Cost: Figure 4(a) shows the cumulative utility calculated based on Eq. 10 under the four difference feedback conditions and with respect to each different follower type. Overall, performance in No-feedback condition is the worst compared to all other conditions. As \(A\) does not provide any feedback, \(F\) does not improve and maintains the initial levels of willingness and competence. When both willingness and competence remain low (\(w=.25, c=.25\)), the interaction yields negative cumulative utility. Only when both competence and willingness are high, \(A\) can benefit from the interaction with \(F\) without providing feedback.

Figure 3. Average cost of providing feedback to a follower across trials under four different feedback conditions

Figure 4. Assistant commander’s utility and cost across trials under four different feedback conditions

Given that \(A\) provides feedback, we find that the cumulative utility is positive in all the conditions and with generally increasing trends. For \(F\) with low competence (e.g., \(c=.25\)), Constant-feedback and Random-feedback perform better than IBL-based feedback. On the other hand, when competence is high (e.g., \(c=.75\)), IBL-based feedback performed the best among all because \(F\)'s high competence can maximize improvement in Eq. 8. In particular, we observe the significant outperformance of IBL-based feedback when \(F\) has high willingness and competence such as (\(w=.75, c=.75\)). This is due to the adaptive nature of the IBL-based feedback that adjusts the amount of feedback based on \(F\)’s capabilities and the decrease of the cost of providing feedback. In parallel, highly competent \(F\) learns faster than low competence \(F\) as trust improvement is a function of \(F\)’s capability (see Eqs. 1 and 2). Faster learning and performance improvements in \(F\) also influence the utility of \(A\). For \(F\) with (\(w=.75, c=.75\)), there is only a small difference between the two extreme feedback conditions, No-feedback and Constant-feedback, while IBL-feedback performs the best. This again highlights the importance of tuning the amount of feedback to \(F\)’s needs, and the ability of our framework to do so.

Figure 4(b) shows the cumulative cost (see Eq. 11) over time for the five types of followers under four different feedback conditions. Well aligned with our previous results in Figure 3, the lowest cumulative cost in IBL-based feedback. This shows the best cost efficiency of IBL-based feedback that achieves trust improvement of followers, while maintaining low operational costs. Thus, compared to all other conditions, the IBL-feedback is most suitable to
operate in time-sensitive, dynamic decision making situations where a leader has limited capacity and resources.

6. Conclusions and Future Work

In this work, we integrated the concept of trust with a cognitive leadership framework in dynamic decision-making situations. We showed that this framework provides adaptive feedback to facilitate the performance improvement of followers.

The novelty of this work lies in that (1) it provides efficient and effective decision making framework for a leader to improve followers’ performance; (2) it assumes no prior knowledge about the followers and incorporates the limited capability of the leader; and (3) it enables the leader to make efficient and effective decisions with dynamic trust assessments to maximize the followers’ performance with minimal cognitive cost to the leader.

Results show that a follower’s competence, rather than willingness, significantly affects triggering of feedback leading to overall performance improvement. In addition, results demonstrate that the IBL-based cognitive leadership framework significantly outperforms other compared schemes in terms of the utility and cost efficiency by improving followers’ performance. The key merit of the IBL-based scheme is the cognitive adaptive learning. More complex strategies with a large set of rules may outperform the IBL-based feedback. However, such a strategy should be crafted for a known set of followers, which has non-adaptive learning and is not applicable in highly dynamic decision making situations.

We plan to extend this work by (1) including comprehensive sensitivity analysis of key design parameters such as the critical thresholds for time deadline and minimum cost; (2) extending the learning mechanism of the followers to consider internal learning; and (3) considering the decision making between the middle-level leaders and a commander to measure the decision making success metric. Further research will also compare the performance of an IBL model with other adaptive models, and eventually evaluate the ability to account for human leader-follower interaction.

Acknowledgements

Research was in part sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

7. References


Ben-Asher, N., Dutt, V., & Gonzalez, C. (2013). Accounting for the integration of descriptive and experiential information in a repeated prisoner’s dilemma using an instance-based learning model. In Proceedings of the 22th behavior representation in modeling & simulation (BRiMS) conference (pp. 11–14).


Malware Identification Using Cognitively-Inspired Inference

Robert Thomson, Christian Lebiere and Stefano Bennati
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
412-268-6028
(thomsonr, cl, sbennati @andrew.cmu.edu)

Paulo Shakarian, Eric Nunes
Arizona State University
699 S. Mill Avenue
Tempe, AZ 85382
(shak, eric.nunes @asu.edu)

Keywords:
Malware Analysis, Inference, Instance-Based Learning, Functional Modeling, Cognitive Architectures

ABSTRACT: Malware reverse-engineering is an important type of analysis in cybersecurity. Rapidly identifying the tasks that a piece of malware is designed to perform is an important part of reverse engineering that is generally manually performed as it relies heavily on human intuition. This paper describes how the use of cognitively-inspired inference can assist in automating some of malware task identification. Computational models derived from human-inspired inference were able to reach relatively higher asymptotic performance faster than traditional machine learning approaches such as decision trees and naïve Bayes classifiers. Using a real-world malware dataset, these cognitive models identified sets of tasks with an unbiased F1 measure of 0.94. Even when trained on historical datasets of malware samples from different families, the cognitive models still maintained the precision of decision tree and Bayes classifiers while providing a significant improvement to recall.

1. Introduction

Malware reverse-engineering is an important task for cyber-security. While large amounts of data can be sorted and filtered using machine learning techniques, identifying the tasks that a piece of malware is designed to perform is manually performed as it relies heavily on human intuition (Sikorski & Honig, 2012). The complexity of this task increases substantially when you consider that malware is constantly evolving, and that how each malware instance is classified may be different based on each cyber-security expert’s own particular background.

Malware classification occurs in two stages: the first is determining whether a given binary is malicious (Tamersoy, Roundy & Horng 2014; Firdausi, Lim, Erwin, & Nugroho, 2010) and then classifying this malware by family (Bayer, Comparette, Hlauschek, et al., 2011; Kinable & Kostakis, 2011; Kong & Yan, 2013). Malware family classification has suffered from two primary drawbacks: (1) disagreement about malware family ground truth as different analysts (e.g. Symantec and MacAfee) cluster malware into families differently; and (2) previous work has shown that some of these approaches mainly succeed in “easy to classify” samples (Perdisci, 2012; Li, Liu, Gai & Reiter, 2010), where “easy to classify” is a family that is agreed upon by multiple malware firms. In this paper, we look to infer the specific tasks a piece of malware was designed to carry out. While we do assign malware to a family, to avoid the two aforementioned issues the family partition is done probabilistically and the ground truth compared to the tasks each malware performed rather than an assignment to a family.

The ability to stably and accurately sort substantial information from the environment is a key element of human cognition. It is generally accurate even with incomplete evidence and limited feedback. It thus seems beneficial to examine features of human cognition that may guide our development of algorithms to sort through the large amounts of data generated by malware analyses.

We argue that malware identification techniques can be improved using cognitively-inspired inference. Cognitive architectures such as ACT-R (Anderson, Bothell, Byrne, et al., 2004) have previously been shown to effectively model human cognition on a variety of decision-making (Lebiere, Gonzalez, & Martin, 2007) and general intelligences tasks (Lebiere, Gonzalez, & Warwick 2009), including complex domains such as intelligence analysis (Lebiere, Pirolli, Thomson, et al., 2013). Further, due to the ability of these models to mimic human cognition, they have been shown to perform well on reasoning tasks where historical knowledge is sparse, limited, or dissimilar to the current context (Taatgen, Lebiere, & Anderson, 2006).

An example of the efficiency gained through cognitive inference is the cognitive model of backgammon that is able to learn to perform at a highly skilled level after playing a few hundred games, as opposed to tens-of-thousands to millions of games for the equivalent machine learning algorithms to reach a comparable performance (Sanner et al., 2000). The key aspect of cognitive inference that is leveraged to achieve this efficiency and capability is the combination of symbolic problem decomposition with statistical learning, made possible by the tight integration of symbolic and subsymbolic representations in cognitive architectures such as ACT-R.

In this paper we leverage the cognitively-inspired inference mechanisms in ACT-R to identify the tasks associated with a piece of malware. Using a real-world malware dataset (Mandiant Corp, 2013), our cognitive models identified sets of tasks with an unbiased F1 measure of 0.94 – significantly out-performing baseline approaches including a decision-tree and naïve Bayes classifier while using only highly scalable online learning.
2. Cognitively-Inspired Inference

Machine learning algorithms like deep learning are massively parallel and can cope with large amounts of data; however they are limited because of their relatively primitive semantics and thus are best suited for the initial filtering and structuring of data. On the other end of the spectrum, human inference has memory and attentional limitations, cognitive processes are powerful, where adaptive heuristic strategies are adopted to accomplish the tasks under strong time constraints using limited means. An advantage of using a cognitively-inspired model to describe inferential processes is that the underlying architecture provides the benefits of human-inspired inference while allowing for more flexibility over constraints such as human working memory.

There is a valid use of cognitive architectures for artificial intelligence that makes use of basic cognitive mechanisms while not necessarily making use of all constraints of the architecture. Reitter & Lebiere (2010) introduced a modeling methodology called accountable modeling that recognizes that not every aspect of a cognitive model is reflected in measurable performance. In that case, it is arguably better to specifically state which aspects of the model are not constrained by data, and rather than mock up those aspects in plausible but impossible to validate manner, simply treat them as unmodeled processes. This approach results in simpler models with a clear link between mechanisms used and results accounted for, rather than being obscured by complex but irrelevant machinery. For instance, while the models described in this paper use activation dynamics well-justified against human behavioral and neural data to account for features such as temporal discounting, we do not directly model working memory constraints to allow for more features of malware and more instances to be present in memory.

3. The ACT-R Cognitive Architecture

We leveraged features of the declarative memory and production system of the ACT-R architecture to complete malware identification. These systems store and retrieve information that correspond to declarative and procedural knowledge, respectively. Declarative information is the kind of knowledge that a person can attend to, reflect upon, and usually articulate in some way. Conversely, procedural knowledge consists of the skills we display in our behavior, generally without conscious awareness. Modules are encapsulated and may process information in parallel within one another. However, there are two serial bottlenecks in processing: only one production may be executed at a time, and the contents of a module can only be accessed through a buffer that can only contain one chunk at a time.

3.1 Declarative Knowledge

Declarative knowledge is represented formally in terms of chunks. Chunks have an explicit type, and consist of an ordered list of slot-value pairs of information. Chunks are retrieved from declarative memory by an activation process: $P_i = \frac{(e^{A_i/s})}{\sum_j e^{A_j/s}}$ where $P_i$ is the probability that chunk $i$ will be recalled, $A_i$ is the activation strength of chunk $i$, $\sum A_j$ is the activation strength of all of eligible chunks $j$, and $s$ is momentary noise inducing stochasticity by simulating background neural activation.

The activation of a given chunk $i$ ($A_i$) is governed by its summed base-level activation ($B_i$) reflecting its recency and frequency of occurrence, spreading activation ($S_i$) reflecting the effects that buffer contents have on the retrieval process, partial matching score ($P_i$) reflecting the degree to which the chunk matches the retrieval request, and finally a noise value ($\varepsilon$) including both transient and permanent noise: $A_i = B_i + S_i + P_i + \varepsilon_i$. Subsymbolic activations approximate Bayesian inference by framing activation as log-likelihoods, with base-level activation ($B_i$) as the prior, the sum of spreading activation and partial matching as the likelihood adjustment factor(s), and the final chunk activation ($A_i$) as the posterior.

A chunk’s base-level activation is computed by summing across the number of presentations $n$ for chunk $i$ the log of the time $t$ since the $j$th presentation discounted by decay rate $d$, with this an optional constant $\beta_j$ added to this value: $B_i = \ln(\sum_{j=1}^{n} t_j^{-d}) + \beta_j$. Base-level activation corresponds to the Bayesian prior of a chunk’s activation. A benefit of base-level activation is that it provides an automated procedure for frequency-based strengthening as well as temporal discounting.

The spread of activation ($S_i$) is computed by the following equation: $S_i = \sum_k \sum_j W_{kj} S_{kj}$ where elements $k$ being summed over are the set of buffers in the model, elements $j$ being summed over are the chunks which are in the slots of the chunk in buffer $k$ (these are referred to as the sources of activation), $W_{kj}$ is the amount of activation from sources $j$ in buffer $k$ weighted by parameter $W$, and $S_{kj}$ is the strength of activation from chunk $j$ to $i$. Strengths of association correspond to the Bayesian log-likelihood of chunk $i$ being relevant given context elements $j$. $S_{kj}$ is therefore defined as $\log(P(i|j)/\log(P(i)))$. These associations are built up from experience, and they reflect how chunks co-occur in cognitive processing. The spread of activation from one cognitive structure to another is determined by weighting values $W$ on the associations among chunks, which determine the rate of activation flow.

Chunks are also compared to the desired retrieval pattern using a partial matching mechanism ($P_i$) that subtracts from the activation of a chunk $i$ its degree of mismatch $M_{ki}$ to the desired pattern $k$, additively for each component and chunk value: $P_i = \sum_k PM_{ki}$. Both the spreading activation and partial matching mechanisms serve to automate efficient contextual priming effects.
While the most active chunk is usually retrieved, a blending process (i.e., a blended retrieval; see Lebiere, 1999; Wallach & Lebiere, 2003) can also be applied that returns a derived output $V$ reflecting the similarity $S_{ij}$ between the values of the content of all chunks $i$ and compromise value $j$, weighted by their retrieval probabilities $p_i$ reflecting their activations and similarity scores: $V = \arg\min \sum_i p_i (1 - S_{ij})^2$. This process enables the generation of continuous values (e.g., probabilities) in a process akin to weighted interpolation.

### 3.2 Procedural Knowledge

*Production rules* are used to represent procedural knowledge. They represent and apply cognitive skill in the current context, including how to access and modify information in buffers and transfer it to other modules. Each production rule is a set of *conditions* and *actions* which are analogous to IF-THEN rules. Conditions specify structures that are matched in buffers, and correspond to information from the external world or other internal modules. Matching production rules effectively means: if the conditions of a given production match the current state then perform the following actions.

### 3.3 Instance-Based Learning

Instanced-based learning (IBL) is the theory that people have a general-purpose mechanism whereby situation-action-outcome observations are stored and retrieved from memory. IBL offers constraints on explanation by grounding implicit learning within the mechanisms of a cognitive architecture. The dynamics of an instance’s sub-symbolic activations (e.g., frequency and recency in the base-level equation) provide a scientifically-justified mechanism for determining which instances are likely to be retrieved for a given situation, and also can explain why they were retrieved and what factors came into play.

These instances are represented with slots containing the conditions (contextual cues), the decision made (an action), and the outcome of the decision (the utility of the decision). Before sufficient task-relevant knowledge is available, alternatives are evaluated using heuristics (e.g., random choice, loss-minimization, maximizing gain). Once sufficient instances are learned, decision-makers retrieve and generalize from these instances to evaluate and make a decision, and execute the task. As for learning, the generalization process is constrained by mechanisms with the cognitive architecture, in this case partial matching and blending.

The process of feedback involves updating the outcome slot of the chunk according to the post-hoc generated utility of the decision. Thus, when decision-makers are confronted with similar situations while performing a task, they gradually abandon general heuristics in favor of improved instance-based decision-making processes (Gonzalez & Lebiere, 2005). IBL methodology has been used in a number of research applications including the AFRL 711 HPW/ RHA’s model of Predator operators. It can also be used to represent individual differences in experience and capacity by providing and parameterizing content from a single individual (e.g., Sanner et al., 2000; Wallach & Lebiere, 2003).

### 4. Malware Identification Task

We created a dataset identified by the popular malware report by Mandiant Inc. (2013). Dynamic malware analysis was performed using the ANUBIS (2014) sandbox. From the ANUBIS data, a total of 1740 malware attributes were identified (see Table 1 for a select sample).

<table>
<thead>
<tr>
<th>ATTRIBUTES</th>
<th>INTUITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasDynAttrib</td>
<td>Has generic attribute in the analysis</td>
</tr>
<tr>
<td>usesDll(X)</td>
<td>Malware uses a library X</td>
</tr>
<tr>
<td>regAct</td>
<td>Conducts an activity in the registry</td>
</tr>
<tr>
<td>fileAct</td>
<td>Conducts an activity on a certain file</td>
</tr>
<tr>
<td>proAct</td>
<td>Malware initiates or terminates a process</td>
</tr>
</tbody>
</table>

We studied all families where there were at least 5 samples successfully processed by ANUBIS, which provided 15 families and 137 samples (see Table 2).

<table>
<thead>
<tr>
<th>FAMILY</th>
<th>NUMBER OF SAMPLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BISCUIT</td>
<td>6</td>
</tr>
<tr>
<td>BOUNCER</td>
<td>5</td>
</tr>
<tr>
<td>COOKIEBAG</td>
<td>6</td>
</tr>
<tr>
<td>GOOGLES</td>
<td>5</td>
</tr>
<tr>
<td>GREENCAT</td>
<td>22</td>
</tr>
<tr>
<td>NEWSREELS</td>
<td>14</td>
</tr>
<tr>
<td>STARSYPOUND</td>
<td>21</td>
</tr>
<tr>
<td>TABMSGSQL</td>
<td>7</td>
</tr>
<tr>
<td>TARSIP-ECLIPSE</td>
<td>7</td>
</tr>
<tr>
<td>TARSIP-MOON</td>
<td>5</td>
</tr>
<tr>
<td>WEBC2-BOLID</td>
<td>5</td>
</tr>
<tr>
<td>WEBC2-CSN</td>
<td>8</td>
</tr>
<tr>
<td>WEBC2-GREENCAT</td>
<td>6</td>
</tr>
<tr>
<td>WEBC2-HEAD</td>
<td>9</td>
</tr>
<tr>
<td>WEBC2-YAHOO</td>
<td>11</td>
</tr>
</tbody>
</table>

Based on malware family description, we associated a set of tasks with each malware family (that each malware in that family was designed to perform). In total, 30 malware tasks were identified for the given malwares (see Table 3). On average, each family performed 9 tasks.

<table>
<thead>
<tr>
<th>TASK</th>
<th>INTUITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>beacon()</td>
<td>beacons back to adversary’s system</td>
</tr>
<tr>
<td>bruteForceSqlLogin()</td>
<td>uses a brute-force technique</td>
</tr>
<tr>
<td>capturesKeystrokes()</td>
<td>Captures keystrokes from the target</td>
</tr>
<tr>
<td>createModifyFiles()</td>
<td>Designed to modify target’s files</td>
</tr>
<tr>
<td>createProc()</td>
<td>Designed to create a new process</td>
</tr>
<tr>
<td>Download()</td>
<td>Download files to the target</td>
</tr>
<tr>
<td>function</td>
<td>description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>encryptedComms()</td>
<td>Uses encrypted communication</td>
</tr>
<tr>
<td>enumFiles()</td>
<td>Enumerate files on the target</td>
</tr>
<tr>
<td>enumUsers()</td>
<td>Enumerate users on the target</td>
</tr>
<tr>
<td>exeArbitCmds()</td>
<td>Execution of arbitrary commands</td>
</tr>
<tr>
<td>gatherSysInfo()</td>
<td>Gathers system information</td>
</tr>
<tr>
<td>maintPersist()</td>
<td>Maintains persistence on the target</td>
</tr>
<tr>
<td>openListenPort()</td>
<td>Opens a listening port on the target</td>
</tr>
<tr>
<td>procEnum()</td>
<td>Enumerates running processes</td>
</tr>
<tr>
<td>procTerm()</td>
<td>Allows termination of processes</td>
</tr>
<tr>
<td>redirNwTraffic()</td>
<td>Re-directs target’s network traffic</td>
</tr>
<tr>
<td>sendPwdInfo()</td>
<td>Sends target password information</td>
</tr>
<tr>
<td>serviceEnum()</td>
<td>Enumerates target’s services</td>
</tr>
<tr>
<td>serviceManip()</td>
<td>Manipulates target’s services</td>
</tr>
<tr>
<td>shell()</td>
<td>Provides adversary remote shell</td>
</tr>
<tr>
<td>smartCardMonitor()</td>
<td>Monitors target for smart card use</td>
</tr>
<tr>
<td>sqlQueryToAttacker()</td>
<td>Uses SQL query</td>
</tr>
<tr>
<td>ssl()</td>
<td>Uses SSL for communication</td>
</tr>
<tr>
<td>sysEnum()</td>
<td>Enumerates systems on a network</td>
</tr>
<tr>
<td>takeScreenShots()</td>
<td>Takes screen shots</td>
</tr>
<tr>
<td>uninstall()</td>
<td>Includes an uninstall routine</td>
</tr>
<tr>
<td>updateMwCfg()</td>
<td>Update the malware’s configuration</td>
</tr>
<tr>
<td>upload()</td>
<td>Designed to upload files from target</td>
</tr>
<tr>
<td>usesHttp()</td>
<td>Uses HTTP for communications</td>
</tr>
<tr>
<td>webC2()</td>
<td>Uses a web-based C&amp;C</td>
</tr>
<tr>
<td>upload</td>
<td>Designed to upload files from target</td>
</tr>
</tbody>
</table>

### 4.1 Decision Tree

For baseline comparison to the cognitive models, we first implemented a decision tree. This hierarchical recursive partitioning algorithm is widely used for classification problems. The decision tree finds the attribute that maximizes information gain at each split. The total entropy is the weighted (fraction of samples in each split) sum of the two entropies. The attribute that minimizes this entropy (in turn maximizing information gain) is the best split attribute. We calculated the entropy for each split to be: $E = -\sum_f p(x) \times \log p(x)$. Each node in the tree is divided into two groups, one having the best split attribute and the other which does not have that attribute. In order to avoid over-fitting, the terminating criteria was set to less than 5% of total samples (i.e., the node with less than 5% of total samples is declared as a leaf node).

During the testing phase, for a new malware sample, we start from the root of the trained tree and for each node we see if the best split attribute is present in the test sample; if yes we assign the sample to Group 1 otherwise we assign it to Group 2. We continue this procedure iteratively until we reach a leaf node. Since labels are not used during training to build the tree, the leafs may or may not be pure, thus generating a probability distribution over the malware families. This family distribution is assigned to the test samples. Tasks are then determined by summing up the probability of the families associated with the task, with a threshold set at 50%.

### 4.2 Naïve Bayes Classifier

Due to its similarity to ACT-R’s activation equation, we decided to use a Naïve Bayes classifier as a secondary baseline approach. Naïve Bayes is a probabilistic classifier that uses Bayes theorem with an independent attribute assumption. During training we compute the conditional probabilities of a given attribute belonging to a particular family. We also compute the prior probabilities for each family i.e., the fraction of the training data belonging to each family. More specifically, given a malware sample $S$ with a set of attributes $(a_1, a_2, \ldots, a_d)$, the probability that the given sample belongs to family $(f)$ is calculated as $P(S | f) = \frac{P(f) \times P(S | f)}{P(S)}$. For a given sample the total probability $P(S)$ doesn’t vary, so we can safely ignore it. Naïve Bayes assumes that the attributes are statistically independent hence the likelihood formula can be written in the simplified form $P(S | f) = P(f) \times \prod_{d} P(a_i | f)$. This generates a distribution over families for a given sample.

During testing, the probability of a malware sample belonging to a family is just the product of the individual attribute belonging to that family and its prior probability. The association of the test sample with each malware family is computed, generating a distribution over malware families and the tasks associated with the sample are determined in a similar way to that of decision tree.

### 5. Cognitive Models

Two distinct models were created that leveraged separate parts of the activation calculus. The models are built using the inferential mechanisms of the ACT-R cognitive architecture and learn to recognize malware samples based upon a limited training schedule similar to the actual experiences of a human analyst. Given a malware sample, the model generates a probability distribution over a set of malware families, then infers a set of likely malware intents based upon that distribution. The models primarily leverage the sub-symbolic mechanisms of the ACT-R architecture, especially the activation calculus underlying retrieval from declarative memory. Each sample is represented by its set of static and dynamic attributes. The model operates in two stages: first by family, then by intent. To assign family, the model generates a probability distribution over the set of possible malware families from the activation in long-term memory of the chunks representing instances of those families. To assign intent in a second pass the model uses a similar process to generate likely intents from a representation linking each malware family to known intents. How they accomplish these two stages, specifically which representations and mechanisms are used, distinguish the two models.
A rule-based model leverages the Bayesian memory activation mechanisms. Its representation is relatively compact, involving a single chunk for each family whose associations abstract the various instances belonging to that category, but whose associations need to be computed and do not involve temporal discounting and other adaptive features (see Thomson & Lebiere, 2013). The instance-based model is based on more direct, incremental learning that accumulates malware instances in memory and leverages neurally-plausible pattern matching processes such as partial matching and blending (Lebiere et al., 2013) but is less parsimonious with storage and thus has potential scalability issues for large data sets.

### 5.1 Rule-Based Model

This ACT-R model is not strictly rule-based because it does not in any way include a rule that determines its judgment, e.g., in the way that the decision tree is a representation of a hierarchical decision procedure that repeatedly partitions the attribute space in subcategories. Rather, this model is called rule-based because each family is represented as a single chunk, and the subsymbolic information associated with that chunk, specifically the base-level and strengths of associations, constitute an implicit definition of belonging to that family. Those parameters can be learned incrementally, or they can be set to reflect the aggregate statistics of an entire training corpus. We followed the latter procedure. Specifically, the base-level associated with family chunk \( f \), representing the a priori probability of a malware sample belonging to that family, is set to the log of the Bayesian prior \( \ln(p(f)) \).

Similarly, the strength of association from malware attribute \( a \) to family \( f \) is set to the log-likelihood ratio \( \ln(p(a|f)/p(a)) \). These strengths of association are multiplied by the attentional focus \( W_a \) associated with each attribute to determine the total activation flowing from the set of attributes associated with the current malware to the family chunk. Together, base-level and spreading activation determine the total activation of the family chunk \( f \), in turn determining the probability associated with that family through the Boltzmann (softmax) distribution. As for the baseline models, intent probabilities are then determined by summing up the probabilities of families associated with that intent, with the same threshold of 50% determining a positive intent identification.

### 5.2 Instance-Based Model

This model follows the instance-based learning theory (IBL; Gonzalez, Lerch, and Lebiere, 2003) that is particularly relevant to modeling naturalistic decision making in complex dynamic situations. The instance-based approach is an iterative learning method that reflects the cognitive process of accumulating experiences and using them to make decisions. In this case a chunk is created for each malware instance rather than each malware family. Thus it is a straightforward instance of online learning where each new experience results in a new memory chunk. Each chunk represents the set of attributes together with the family identification. The base-level activation of each chunk is learned by to the mechanism described in the introduction. The power law decay makes it sensitive to the recency of presentation, allowing both for old malware instances to quickly decay away as well as for new ones to rapidly reach prominence.

If the same instance (i.e., same attributes and family) is presented multiple times, the base-level will also reflect the frequency of presentation. Strengths of associations are not used: rather the effect of context, as represented by the set of attributes of the current malware, will be reflected through the partial matching mechanism. The match score of a chunk to the current context will reflect the similarity between the attribute sets of the current malware sample and each instance in memory, as measured by the dot product between the respective attribute vectors. The retrieval process then extracts from the chunk its family identification. The blending mechanism computes a probability distribution over all possible family values, reflecting the activation of each instance. Thus the same factors as in the rule-based model are reflected in the computation, specifically the frequency of each malware and overlap in attribute space, but using distinct architectural mechanisms.

The second part of the process, going from probability distribution over families to intents, is entirely different. Instead of simply summing up the probabilities of each families associated with a given intent, the instance-based learning approach is also applied for this second step. This time, the instances learned associate the probability distribution over families computed for the given malware with its actual intents. Given a new malware instance, a retrieval process matches its family probability distribution against those of previous instances, and extracts the probability of each intent using the same blending process used for generating the family probabilities. Intents reaching the 50% threshold are again selected. The key aspect of this process is that it is now sensitive to the entire probability distribution over families rather than simply a sum of its values.

### 6. Results

We compared both the instance-based and rule-based models against both the decision tree and naïve Bayes classifiers for both leave-one-out cross-validation and leave-one-family-out cross-validation. In the leave-one-out cross-validation, for each of the 137 malware samples, we trained 136 samples and tested on the remaining one. This procedure was repeated for all samples and the results
were averaged. Similarly, in the leave-one-family-out we trained on 14 families and tested on the 15th.

Pairwise t-tests \((t(136))\) for the leave-one-out cross-validation showed that instance-based \(>\) rule-based decision tree \(>\) naïve Bayes model (all \(p < .01\); see Figure 1). The instance-based model outperforms baseline approaches in detecting 9 of 15 families with an average F1 difference greater than 0.3 with at least 99% confidence \((t(136) = 4, p = .01)\), while neither the decision tree nor naïve Bayes methods significantly outperformed the instance-based model on any family. We argue that instance-based model is superior because it uses the full pattern of the probability distribution over families rather than just a sum (as in the rule-based model).

![Figure 1. Leave one out cross-validation.](image1)

Similarly, pairwise t-tests for the leave one-family out cross-validation found that instance-based \(>\) rule-based decision-tree \(>\) naïve Bayes \((p < .01\); see Figure 2).

![Figure 2. Leave one family out cross-validation.](image2)

The results of these analyses indicate that the cognitively-inspired models performed quantitatively better than the baseline machine learning approaches. Also, in the leave-one-out analyses, the iterative nature of the instance-based cognitive model allowed for it to reach near-asymptotic performance (and its best when compared against baseline approaches) after only 40% of the training instances.

We also investigated which features of families may cause the cognitive models to exhibit superior performance over baseline approaches. Figure 3 presents a similarity matrix between malware families. Similarity was computed by generating the Kullback-Leibler Divergence of intents’ ground truth between families, averaged over all instances of a family. KLD is a non-symmetric measure of the difference between probability distributions. It is important to note that a family is not always maximally similar to itself; each cell contains the average of the similarities between each pair of malware in that family.

![Figure 3. Similarity matrix for all 15 malware families.](image3)

By examining the family-wise performance for leave one out cross validation (see Figure 1) we determined that the decision tree has difficulty predicting malware tasks from BISCUIT, WEBC2-CSN, WEBC2-GREENCAT, TABMSGSQL, COOKIEBAG and BOUNCER. As seen in Figure 3, these families have similarity values to each other, leading to substantial confusion within the decision tree. For instance, BISCUIT (top row) is highly similar to 7 other families. It is thus likely that the decision tree will confuse one of the other families for BISCUIT, and potentially vice versa. In essence, the decision tree is a purely symbolic reasoned and will only work accurately if the categories are linearly separate in the dimensions of its representation (i.e., malware features). It fails in these families because there is no logical condition that can be designed to separate them.

The Naïve Bayes model does not fall prey to the same family confusability as the decision tree as it is an inherently statistical algorithm. Instead, from Figure 1 we determined that Naïve Bayes had difficulty predicting malware tasks from WEB2-YAHOO, NEWSREELS, BOUNCER and STARSPOUND. These families are not as similar (as seen in Figure 3), so these families are in some sense uncorrelated. Naïve Bayes has one core assumption, namely that family attributes are statistically-independent. This assumption is incorrect for the given
malware analyses. The families that the Naïve Bayes algorithm has difficulty judging, while not highly correlated, share a relatively large percentage of their attributes with all families when compared against other families from which it can correctly classify.

7. Discussion

We presented two models using cognitively-inspired inference scaled beyond traditional memory limitations in order to rapidly identify malware samples. When compared against baseline machine learning algorithms such as decision trees and Naïve Bayes, our cognitive models were better able to classify malware samples across all 15 sample families. The baseline machine learning algorithms each exhibited difficulties with several families that our cognitive models did not. This is because each is fundamentally limited to the feature set chosen by the modeler, a problem common to machine learning.

By leveraging the temporal dynamics (e.g., recency and frequency) of the cognitively-inspired base-level equation, we were better able to classify highly-similar families than the decision tree. In addition, while the base-level equation can be explained in Bayesian terms and generally reproduces behavior consistent with Bayesian reasoning, in the case of the instance-based model, it is not limited to assumptions of statistical independence. Instead, the instance-based model acts as a universal approximator that does not depend on any kind of linear separation: the more instances that the model perceives, the more accurately it will interpolate between them in a multidimensional space.

It is interesting that the rule-based model performs substantially better than the Naïve Bayes model because the spreading activation equations has the same independence assumption between sources as does the Naïve Bayes algorithm. We argue that the base-level learning component and weighting of the spreading activation equation that compensates, providing the benefits of both a decision-tree and Naïve Bayes classifier without all the drawbacks. For instance, the weight of the spreading activation is effectively forced to be 1 in the Naïve Bayes classifier. By varying this weight, the ACT-R model is capable of determining the relative importance between the prior and posterior likelihoods.

Furthermore, our models were able to reach peak performance when compared against baseline machine learning models after only 40% of stimuli were presented. This is not to argue that cognitively-inspired models are a panacea for malware identification or a clear improvement over machine learning techniques. While our models were able to learn with fewer samples, the processing overhead of situating our models within a cognitive architecture means that they do not necessarily operate faster than machine learning techniques, especially when scaled to larger datasets. That criticism aside, we have elsewhere argued that cognitive models work best when used to supplement human-in-the-loop operations whereby these models take input initially processed by machine learning algorithms (such as deep learning) to do some dimension reduction and provide the cognitive model with loosely structured data from which it may draw inferences (Thomson, Lebiere, & Bennati, 2014). Also, due to the rapid learning across sparse datasets, the ACT-R model has been used to provide top-down feedback to deep learning algorithms (Vinokurov et al., 2011).

Our work substantially differs from malware family classification as we look to infer the tasks a malware was created to perform directly whereas malware family classification is mainly used to help guide an analyst into identifying tasks by first identifying a family. It is noteworthy that we were able to train our classifiers on data of malware of different families than the malware we are attempting to classify and were still able to obtain a set of tasks with over 60% recall on the best-performing cognitive models. Further, as a side-effect, we created a probability distribution over malware families as part of an intermediate step – though the ultimate inference of malware tasks is independent of how the historical malware families are classified by family. This suggests that at this point machine learning, cognitive models and human experts provide complementary strengths that should be leveraged for the challenging problems facing us in the cyber security domain.

8. Acknowledgement

This work was sponsored by the Intelligence Advanced Research Projects Activity via DOI contract number D10PC20021. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI, or the U.S. Government.

9. References


**Author Biographies**

**ROBERT THOMSON** is research faculty at Carnegie Mellon University in Pittsburgh, PA and works with the Naval Research Laboratory in Washington, DC. His main research interests are learning theories and the structure of memory, particularly their application to computational cognitive architectures and robotics.

**CHRISTIAN LEBIERE** is research faculty in the Psychology Department at Carnegie Mellon University. His main research interests are computational cognitive architectures and their applications to psychology, artificial intelligence, human-computer interaction, decision-making, intelligent agents, and cognitive robotics.

**PAULO SHAKARIAN** is a researcher at Carnegie Mellon University in Pittsburgh, PA. He has written two books, including Elsevier's "Introduction to Cyber-Warfare" and his social network analysis software is used by the Chicago Police.

**STEFANO BENNATI** is a researcher at Carnegie Mellon University in Pittsburgh, PA.

**ERIC NUNES** is a Ph.D. student at Carnegie Mellon University in Tempe, AZ.
Extending Generative Models of Large Scale Networks

Eli Stickgold
Bruce Skarin
Ian Stewart
Corey Lofdahl
Charles River Analytics
625 Mt. Auburn Street
Cambridge, MA 02138
617-491-3474
estickgold@cra.com, bruceskarin@gmail.com, istewart@cra.com, clofdahl@cra.com

Keywords:

social media, social networks, network analysis, homophily, agent-based modeling, generative models, anonymous networks

ABSTRACT: Since the launch of Facebook in 2004 and Twitter in 2006, the amount of publicly available social network data has grown in both scale and complexity. This growth presents significant challenges to conventional network analysis methods that rely primarily on structure. In this paper, we describe a generative model that extends structure-based connection preference methods to include preferences based on agent similarity or homophily. We also discuss novel methods for extracting model parameters from existing large scale networks (e.g. Twitter) to improve model accuracy. We demonstrate the validity of our proposed extensions and parameter extraction methods by comparing model-generated networks with and without the extensions to real-life networks based on metrics for both structure and homophily. Finally we discuss the potential implications for including homophily in models of social networks and information propagation.

ACKNOWLEDGEMENTS: This work was performed under DARPA contract number W31P4Q-12-C-0235. The authors thank Dr. Rand Waltzman for his significant technical support and eager engagement on this project. This work was funded in its entirety by the Information Innovation Office (I20). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We also acknowledge the contribution of Professor Frank Witmer who helped with the initial modeling efforts.

1. Introduction

The proliferation of social media has profoundly changed the information landscape since the early 2000s. The emergence of massive social media services such as Facebook and Twitter (Java, Song, Finin et al., 2007; Viswanath, Mislove, Cha et al., 2009) has led to the proliferation of massive, highly-connected social networks. Understanding modern age information flows requires understanding the way these networks grow, evolve and decay over time (Hughes, Rowe, Batey et al., 2012). By their nature however, large portions of these networks are invisible, either hidden by privacy policies or existing in entirely different media: a strong link may exist between two real-world friends who rarely interact directly on Twitter, but nevertheless have a strong impact on each other. This creates ethical risks and technical challenges for direct study, even though most hidden users are often qualitatively similar to visible users (Madden, 2012).

Early analysis attempts solved the first problem by using anonymized datasets, which left the structural information intact, but removed the personally identifying information. Unfortunately, it was quickly proved that even sparse anonymized networks are vulnerable to de-anonymizing attacks with very little information required (Narayanan & Shmatikov, 2008). Furthermore, completely anonymized networks have the problem of removing vital context regarding the actors and links in the network.

Given these risks and challenges, simulation models, and in particular agent based models, can help provide a better analogue for study and research. More realistic models also have the potential to improve our ability to understand and predict information propagation in modern social networks that can only be partially observed. Furthermore, while the exact meaning of links within a social network may be debated (e.g. a personal
friendship is different from a fan-celebrity relationship), virtually all social networks include information-diffusion aspects that can in turn be modeled.

In this paper, we describe an agent based model that leverages social science theory and existing network modeling methods to produce synthetic directed, weighted social networks. This model is then used in a case study focused on Twitter networks, which provides both model parameters and a ground-truth for conducting network comparisons. Such a model may offer insight to analysts seeking patterns in link-formation between users, particularly in cases of news-related information diffusion.

We begin by explaining the need for features based on social science and how they affect the process of agents forming and updating. In particular, we focus on the principle of homophily (McPherson, Smith-Lovin, Cook et al., 2001), which has been found to have strong implications on the spread and interpretation of information within a network. We then describe our model for reproducing homophily in synthetic networks. We follow the model with a discussion of the novel methods we used for extracting parameters from existing network data. In closing we compare our models with and without the new features to existing Twitter networks and discuss the implications for further research and application. In summary, our findings indicate that in the context of representing social networks, the addition of these new features provides measurable improvements over other network generation methods.

### 2. Structure and Homophily

There already are a variety of methods for generating networks of different structural archetypes like small-world, random, and scale-free. Some common approaches include the Watts-Strogatz, Erdős–Rényi (ER), and Barabási–Albert (Barabási & Albert, 1999) models. These have proven to be accurate analogs for the structure of many real-world networks using metrics such as the degree distribution, diameter, and clustering coefficient.

Yet for some networks, in addition to structural features there are also social and functional features that affect how networks create additional archetypes such as the polarized crowd, community clustered, customer support, and broadcast networks as described in Smith, Rainie, Shneiderman & Himelboim (2014). In cases such as these, a different approach is required to represent the basis for forming social/functional features in network models. In addition, standard statistical models such as the ER random graph generator are often inadequate in replicating the structure of social networks (Mislove, Marcon, Gummadi, Druschel, & Bhattacharjee, 2007), which inspires the search for more accurate models.

Research into more robust methods of modeling social networks includes latent space models, Hidden Markov Models, and actor-oriented models (Snijders, 2011). One advantage of actor-oriented models is the ability to include behavior-based theory instead of relying on a purely statistical approach. Actor-oriented models also enable more natural extensions to the modeling of other network behaviors such as information propagation and network adaptation and evolution.

In focusing on behaviors related to social networks, the principle of homophily provides a useful heuristic for determining the likelihood of link formation between two given actors. Homophily asserts that similar people interact more frequently than dissimilar people. In cases such as language, the assertion of homophily holds strongly (Hale, 2014), while in other cases there can be a wider degree of variation. Homophily has also been found to have powerful implications on the information people receive and the attitudes they form. For these reasons, homophily is an important feature for generating realistic analogues of existing social networks.

### 3. Homophily in Agent-Based Modeling

Agent-based modeling is an ideal formalism for a behavior-based approach to the generation of social networks because of its flexibility in representing diverse attributes and behaviors. In an agent based model, individual actors can incorporate a dynamic range of demographic and social attributes and behaviors for use in determining how links are formed between actors.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Example Entities/Range</th>
<th>Use</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Categorical</td>
<td>English, Spanish, French, etc.</td>
<td>Both</td>
<td>High</td>
</tr>
<tr>
<td>Gender</td>
<td>Categorical</td>
<td>Male, Female</td>
<td>Link formation</td>
<td>Moderate</td>
</tr>
<tr>
<td>Activity</td>
<td>Ordinal</td>
<td>High, Medium, Low</td>
<td>Link formation</td>
<td>Low</td>
</tr>
<tr>
<td>Status</td>
<td>Scalar</td>
<td>Power law distribution</td>
<td>Messaging</td>
<td>Moderate</td>
</tr>
</tbody>
</table>
Demographic types that might be represented include categorical (e.g. gender, language), ordinal (e.g. time zone, ranking), and scalar (e.g. latitude/longitude). Each demographic can be parameterized with a distribution for the number of members, and by ranking their importance to link formation. Table 1 provides several example demographics that were used in our research. To determine if a link is formed between two given agents we used a modified version of the preference function described in Pasta, Jan, Zaidi, & Rozenblat (2013).

After a number of experiments generating networks of different sizes, we also found the best fit when we applied the function to different subgroups at each stage of network formation. For initialization the preference function is applied to all of possible initial agent pairings. During early growth, after the seed network is created, the preference function is then applied only to a selected subgroup (e.g. language, gender, etc.) for a given agent. After an agent has a network larger than a specified degree, then the function is applied only to connections within a given degree distance.

3.1 Modified Preference Function

To recreate the principle of homophily, each agent uses a modified version of the preference function commonly used in structural approaches to network generation. The algorithm used here builds on the work of Pasta, Jan, Zaidi, & Rozenblat (2013) by taking a weighted average of the similarity between two given agents based on the relative importance of each demographic. For each demographic $D_p$, of the type categorical, the similarity for any two agents $i$ and $j$ is assigned as such:

$$D_p(i, j) = \begin{cases} 1 & \text{if } i_p = j_p \\ 0 & \text{if } i_p \neq j_p \end{cases}$$

For ordinal and scalar demographic types, the similarity between two agents is simply the normalized Manhattan distance, where $\delta_p$ represents the difference between the maximum and minimum values of the range that can be assigned to a given demographic $D_p$:

$$D_p(i, j) = 1 - \frac{|i_p - j_p|}{\delta_p}$$

The cumulative similarity $S$ between any two actors $i$ and $j$ is then the weighted arithmetic mean, calculated over $n$ demographics. Where $w$ is the weight assigned to each demographic regarding its importance in link formation:

$$S_{i,j} = \frac{\sum_{p=1}^{n} w_p D_p}{\sum_{p=1}^{n} w_p}$$

For the structural preference functions, we selected triadic closures (aka friend-of-a-friend) and degree preference. The triadic closure preference $T$ between two agents $i$ and $j$ is formulated as

$$T_{i,j} = \frac{|i \cap j|}{\min(i, j)}$$

where $\min(i, j)$ represents the minimum number of edges in either actor’s network. Using the minimum ensures that agents with a smaller number of edges are not penalized against agents with large networks. Degree preference $G$ is then assigned by the following:

$$G_{i,j} = \frac{deg_j}{\max(deg_n)}$$

With this formulation, the degree preference of $i$ is based on the out-degree of $j$ relative to the maximum out-degree within the entire network. If $j$ is the agent with the most followers, it will receive a score of 1.

Similar to the cumulative similarity, the cumulative structural preference $R$ is then:

$$R_{i,j} = \frac{w_p T_{i,j} + w_p G_{i,j}}{\sum_{p=1}^{n} w_p}$$

Finally, the total preference $C$ for agent $i$ to follow actor $j$ is given by the weighted sum of the respective cumulative measures:

$$C_{i,j} = w_b R_{i,j} + w_x S_{i,j}$$

When an agent is selected to form a connection, this preference function is used to create a distribution of measurements that is then randomly sampled for a specified number of edges. As discussed in detail in Section 5, the resulting network is now able to match existing measures of structure (e.g. network diameter and degree distribution) as well as measures related to homophily (e.g. similarity and cohesion).

3.2 Two-stage Selection Process

Even though preliminary assessments found improvements in network similarity, we noticed that for certain demographic categories such as language, the modularity was more significant than for other demographic categories such as language, the modularity was much more significant than could be accounted for in the preference function. This becomes particularly apparent as the size of the network grows and the weight of any given features is diluted.
For this reason after initialization and during early network growth we applied a two-stage process. In the first stage, an agent determines from which demographic category to draw from. Then in the second stage the agent selects a link based on the preference function as applied only to the subgroup. This process is similar to other latent-variable models such as composite-network friendship detection (Zhong, Xiang, Fan, Liu & Yang 2014).

After an agent has a network larger than a given degree, then instead of using all the agents within a given demographic category the pool of possible connections is built from the connected components for a given degree distance. From a behavioral perspective this two stage filtering is more representative of the individual constraints for large social network mediums like Twitter. That is, when deciding who to follow, new users do no look at the whole of Twitter, but rather within certain categories like language and from the referrals within his or her existing network.

3.3 Network Generation

The majority of the parameters used for generating networks are extracted from an existing social network (as described in the following Section 4). The remainder are currently set using manual experimentation. Both agents and links are added to the network over time at a rate based on the average growth rate of the observed network. New agents are created as follows:

Input: Demographics, Parameters
Output: New agent
for each Demographic do
    P=random number between (0,1)
    D=Assign category/value Dn(P)
end for

Links are then formed by the following:

Input: AgentPool
Output: New link
Total=0
for agenti < AgentPool do
    for agentj < AgentPool do
        Total = Total + Ci,j
    end for
end for
P=random number between (0, Total)
Target = AgentPoolC(P)
if initializing
    SourcePool=AgentPool
end if
else if TargetDegree < 2
    SourcePool=DemographicPool
end if
else SourcePool=ConnectedComponent
Total=0
for agenti < SourcePool do
    Total = Total + CTarget,i
end for
P=random number between (0, Total)
Source = SourcePoolC(P)

3.4 Extracting Model Parameters

In order to make use of these extensions, we needed a way to determine appropriate values for each parameter. Because these values depend on the network being modeled, we developed a technique for extracting values from a representative network that could be used to determine accurate values for any of a number of possible use-cases. To test our technique, several social networks were constructed from sets of tweets collected over the course of a week in September on a variety of news topics using Twitter’s API. The corpus for this study was created by filtering the public stream of tweets for keywords related to then-current news topics. From that corpus, we filtered tweets for the keyword “Ukraine” and collected approximately 650,000 tweets to use as our primary dataset. We chose this topic because of Ukraine’s prominence in international news at the time of the study, due to the nation’s public conflict with Russia. The relatively high volume and diversity of the participants involved ensured a reasonable ground-truth dataset.

Following collection, the team converted the raw tweets into a directed network using the @-symbol as a proxy for an edge between users. For example, the tweet “@userA breaking Ukraine news!” from user B would result in a directed edge from B to A. The team chose this tweet-based approach to social network construction over a typical snowball-sampling approach that selects one prominent network actor, collects all of their followers, and repeats the process on these new actors. We made this decision for two main reasons:

1. **Scalability**: it is easier to build a large network from tweets than query Twitter recursively for an ever-growing network.
2. **Information diffusion**: the use of the @-symbol implies diffusion of information from one user to another, which is a key area of interest for this study.

In addition to this basic structural information, we queried Twitter for user-specific data to provide node attributes to the network. In contrast to similar social media studies (De Choudhury, 2011), we only extracted self-reported attributes available directly through the API, rather than attempting to speculate on inaccessible attributes such as gender. This decision limits the overall amount of user
data available but more importantly ensures that the demographics for our model’s agents are based as much as possible in reality and less on prior expectations.

Once the networks were built, a set of static and dynamic metrics were extracted to quantify basic network trends. The key to choosing the metrics was generalizability across potential models, meaning statistics that can differentiate a range of known network types. For instance, the ratio of edges to nodes is higher in small-world networks than in scale-free (hierarchical) and random networks. The following metrics were extracted as initialization parameters for the model:

- **Actor pool size**: the total number of nodes in the network, intended as a cap to the model’s growth.
- **Nodes per hour**: number of new nodes that appear in the network in a given hour.
- **Edges per hour**: number of new edges formed in the network in an hour.
- **Edge adding rate**: the $\mu$ and $\sigma$ of every node’s edge-adding rate, i.e. edges/node/hour.
- **Activity frequency**: the distribution of all users’ frequency of posting statuses, i.e. statuses/day. Through experimentation with k-means clustering, this distribution is discretized as a set of three clusters representing low, medium, and high activity.
- **Language split**: percentages of the languages spoken by users in the network. All languages with percentages below 5% were classified as “Other” to minimize data sparseness.

After extracting these parameters for model initialization, we extracted unweighted in-degree and out-degree as a means of comparing real and synthetic social networks. The in-degree of a node $n$ is equal to the number of edges incident on $n$, while out-degree is equal to the number of edges leaving $n$. Using the overall distributions for in- and out-degree, we were able to establish quantitative goals for the model to achieve over the course of its simulation.

4. Results

Preliminary results demonstrate the success of incorporating homophily into the network generation process. We find that the model accounting for homophily in user demographics produces the most realistic network in terms of in- and out-degree (5.1). In addition, our model yields emergent properties to match the ground-truth network, including homophilic clustering (5.2).

For our preliminary results, we used a set of four networks of nearly identical scale: (1) a slice of the ground-truth (GT) network from tweets; (2) a synthetic scale-free network generated using the B model of the Barabási-Albert (BA) algorithm (Barabási and Albert 1999); (3) a network generated using only structural preference (S); and (4) a network generated using structural preference and homophily (S+H).

We chose to use the BA algorithm rather than similar network-generation algorithms as our baseline because it is more flexible in matching our ground-truth networks. For instance, the Watts-Strogatz model produces a fully connected network, which differs from our fractured ground-truth network. In contrast, the BA algorithm can generate a scale-free network with multiple separate components, similar to the visible Twitter network. Also, the BA algorithm can be easily parameterized to achieve a nearly identical structure, since the B model requires only the final node and edge count as parameters. In contrast, the Watts-Strogatz method requires the final node count, the average degree, and an alpha parameter that requires extensive tuning to generate realistic social networks.

Each of our networks is summarized in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>GT</th>
<th>BA</th>
<th>S</th>
<th>S+H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node count</td>
<td>32,026</td>
<td>32,136</td>
<td>32,026</td>
<td>32,026</td>
</tr>
<tr>
<td>Edge count</td>
<td>43,275</td>
<td>43,000</td>
<td>43,714</td>
<td>43,746</td>
</tr>
</tbody>
</table>

4.1 Degree Distribution Comparisons

To begin, we compared the in-degree distribution of all networks in order to verify the expected scale-free nature of human social networks (Snijders, 2011). Removing all zero-degree nodes from each network, we arrive at the distribution of normalized degree frequencies in Figure 1.

![Figure 1: In-degree distribution](image.png)

All four networks appear to follow the predicted power-law distribution, although with highly variable curves: for instance, the GT network has the sharpest curve.
However, the in-degree frequencies for networks generated by our models are consistently closer to the ground-truth frequencies than the BA network. This trend is quantified in Figure 2, which charts the relative deviation of each model’s degree distribution from the ground-truth distribution.

![Figure 2: Deviation from ground-truth in-degree distribution](image)

Note the consistently lower deviation of both S and S+H networks’ in-degree, particularly in the 1, 6, and 7-degree categories. The high deviations in the BA network imply that the GT network does not follow an ideal power-law distribution in the same way as the BA network. Lastly, all three models perform similarly in the 11-100 category, indicating that the GT network has a heavier tail than predicted. In nearly all other categories, the S and S+H networks achieve a lower deviation than the BA network.

We can further quantify this trend by calculating the mean deviations, as well as the Pearson correlation coefficient and Euclidean distance from the ground-truth distribution to the distributions of the three generated networks. We found the best-fitting model to be the one incorporating homophily, as it maximizes similarity and minimizes distance. The fitness statistics are summarized in Table 3, with the bold numbers highlighting the S+H network’s close fit with the GT network.

<table>
<thead>
<tr>
<th>Fitness statistics</th>
<th>BA</th>
<th>S</th>
<th>S + H</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-degree deviation (μ ± σ)</td>
<td>87 ± 54%</td>
<td>66 ± 35%</td>
<td>62 ± 30%</td>
</tr>
<tr>
<td>Euclidean distance</td>
<td>27.6</td>
<td>20.5</td>
<td>18.6</td>
</tr>
<tr>
<td>R²</td>
<td>0.75</td>
<td>0.84</td>
<td>0.87</td>
</tr>
</tbody>
</table>

A similar trend favoring the homophily model was also found in the out-degree distributions of the four networks. Applying the same procedure to the networks’ out-degree distributions, we arrive at the deviations in Figure 3.

![Figure 3: Deviation from ground-truth out-degree distribution](image)

Again, the deviations favor the homophily model results as well as the structure-only model results, as compared to the BA model results. The results here are better than the in-degree deviations: for instance, the S+H network yielded a mean out-degree deviation of 39% as opposed to the in-degree deviation of 62%. In addition, for out-degree, the S+H network had a much lower deviation in the “11-100” category than the other two models, which is a contrast from the models’ equal performance in the same category with respect to in-degree. These discrepancies suggest a reworking of the model’s edge-formation process to balance out-degree with in-degree. The S+H network shows room for model improvement but was still successful compared to the BA network.

### 4.2 Emergent Network Properties

Moving beyond low-level degree measures, we show that the homophily-dependent model yields emergent properties similar to ground-truth data. We selected three metrics to showcase the similarity of emergent properties: triadic closure, average path length, and giant component size. Triadic closure is calculated with the same formula outlined in Section 3, measured because of its correlation with short-range network cohesion. Average path length is equal to the mean of all shortest paths connecting nodes in the network, and we measured it because of its relation to clique-formation within a connected network. The average path length only includes paths between nodes within the giant component, which is the largest connected subgraph within the network and often includes the majority of nodes and edges. We also measured giant component size, because it provides a summary of large-
scale network cohesion which is key in predicting the breadth of information diffusion.

Table 4 outlines key structural metrics demonstrating the success of our model in replicating the GT network. The bold numbers indicate the closest fit to the GT statistics.

### Table 4: Emergent properties

<table>
<thead>
<tr>
<th>Emergent property</th>
<th>GT</th>
<th>BA</th>
<th>S</th>
<th>S+H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triadic closure (undirected)</td>
<td>0.40</td>
<td>0.00</td>
<td>0.059</td>
<td><strong>0.075</strong></td>
</tr>
<tr>
<td>Average path length (undirected)</td>
<td>5.10</td>
<td>8.96</td>
<td><strong>6.31</strong></td>
<td>8.30</td>
</tr>
<tr>
<td>Giant component size (% of nodes included)</td>
<td>59.8%</td>
<td>91.9%</td>
<td>69.9%</td>
<td><strong>63.3%</strong></td>
</tr>
</tbody>
</table>

These numbers are more abstract to interpret but still demonstrate the success of our model in replicating ground-truth data. First, the GT network has an unexpectedly high rate of triadic closure, which is explained by the apparently high influence of mutual connections on link formation. Triadic closure was still matched best by the S+H model, indicating that homophily provides a boost when combined with our other structural metrics used in network generation.

Secondly, the average path length was best matched by the S network and was lower than anticipated in the GT network. This corroborates the implication that the GT network is denser, which yields a “small-world” effect. Still, the success of the S network suggests that our generation process outlined in Section 3 leads to a topology more realistic than the topology created by BA’s preferential attachment alone. The higher path length in the S+H network is likely related to homophilic clustering (explained later), since the average distance between nodes increases when the nodes exist in distinct clusters.

Thirdly, the giant component size was best matched by the S+H network and is a product of the network’s preferential attachment tendency. For instance, a model only incorporating preferential attachment (BA) tends to yield one core network rather than multiple networks (Newman 2002). This is clear when comparing the giant component size of the S and S+H networks, since the purely structural metrics that led to the S network overemphasized the importance of preferential attachment. Further, the addition of homophily in the S+H network dampened that effect and reduced the giant component to a more realistic size. Overall, these three emergent structural metrics demonstrate the success of our network generation process and particularly the importance of homophily in yielding high-level patterns.

In addition to purely structural metrics, we examined how homophily was correlated with apparent cluster formation in the ground-truth and synthetic networks. Visualizing the Twitter data in Figure 4 using Gephi (Bastian, Heymann and Jacomy 2009), we can infer a close connection between language and cluster structure. Node colors indicate user language and node size is relative to degree, such that nodes with more connections are larger.

We note particularly that non-English users tend to cluster by language. While connections between language clusters do exist, each language appears to cluster more often than not. A similar emergent cluster structure was replicated in our S+H network and can be seen in Figure 5, with identical node-coloring and node-sizing.
While the apparent grouping by homophily is an interesting observation, it does not guarantee that the language-clustering is identical in both networks. For instance, the GT language clusters appear to be denser than the S+H clusters. We verify this apparent pattern of language-clustering by partitioning the graph by language and calculating the density of each partition using

\[ D_G = \frac{|E_G|}{|V_G|(|V_G| - 1)} \]

where \( E_G \) and \( V_G \) represent the sets of edges and vertices in \( G \). We choose density as a metric for basic network coherence that scales predictably to networks of varying sizes. The densities are shown in Figure 6, which shows densities for the overall graph as well as each language’s subgraph (using abbreviations for the same languages as those displayed in Figures 4 and 5).

![Figure 6: Subgraph density (by language)](image)

While not an exact fit, the correlation between the networks is noteworthy. For each language comparison, the S+H network achieves a similar density to the GT network, which can be seen in the relatively low absolute deviation between networks (\( \mu = 0.000829 \)). This is particularly evident in the French language cluster, which has the lowest relative density deviation between networks (13.4%). Based on density similarities, homophily by language is a clear factor in the GT clustered network structure and is emulated well by the S+H model.

Of course, it is important to remember that language is not the only demographic factor affecting edge formation. Future testing will help untangle the connections between partially-dependent demographics such as language and time zone. For instance, it is more likely for speakers of a language such as Ukrainian to represent fewer time zones than speakers of English, a more geographically diffuse language. Such overlaps between demographics could be addressed by a modified node-similarity function that incorporates conditional probabilities, such as \( P(\text{language} = \text{Ukrainian} | \text{time zone} = \text{UTC+2}) \).

5. Conclusions

In this paper, we described novel techniques for improving generative agent-based models of modern social media networks and demonstrated their effectiveness at improving synthetic network quality and realism by comparing produced networks to data about publicly available networks, in this case those found on the social media service Twitter. The continued improvement of this style of generative model, rooted in both network metrics and social science, is necessary for long-term understanding of the propagation of information, especially in partially-visible or wholly dark networks.

Future research in this area will extend and improve these models by incorporating additional social science-based agent-level parameters, including other traits not yet recognized. Further testing could also examine the ability of these models to correctly predict the structure of partially-visible or dark networks by examining the propagation of memetic information through partially-visible networks, such as Twitter, in comparison to the same kind of propagation in synthetic networks. Past studies (Peddinti, Ross, & Cappos, 2014) have proven that private traits across social media users are predictable, and this suggests a similar predictability among private users in general. Thus, future models will likely be able to model “dark” agents in social networks with little extra modification. This sort of testing will provide support for agent-based generative models as a means of uncovering the dynamics of network formation.
6. References


Author Biographies

ELI STICKGOLD is a scientist at Charles River Analytics Inc. whose research interests include machine learning techniques, cognitive models and network modeling.

BRUCE SKARIN is a consultant at Charles River Analytics Inc. whose research interests include system dynamics simulation, agent-based simulation, and network modeling.

IAN STEWART is a software engineer at Charles River Analytics Inc. whose research interests include network analysis, social media analysis, and noisy natural language processing.

COREY LOFDAHL Ph.D. is a senior scientist at Charles River Analytics Inc. whose research interests include modeling and simulation, quantitative analysis, and complex social systems.
How Could Cyber Analysts Learn Faster and Make Better Decisions?

Fangting Wu  
School of Information Science and Technology  
Sun Yat-sen University, Wai-Huan-Dong Road 132, Guangzhou 510006, P. R. China  
wufangt@mail2.sysu.edu.cn

Cleotilde Gonzalez  
Department of Social & Decision Sciences  
Carnegie Mellon University, 208 Porter Hall, Pittsburgh, PA 15213  
coty@cmu.edu

Keywords:  
Cyber SA, IBLT, simulation, tolerance

ABSTRACT: The reliance on humans have been the weakest link and also the most promising power in the design of cyber security systems. To acquire Cyber Situation Awareness (Cyber SA), the ability to comprehend and predict possible cyber threats in a network, defender’s experience is essential. Models of cyber analyst’s learning behavior may serve to measure Cyber SA and how well a cyber analyst maintains and develops this awareness as time progresses. This paper builds a computational model that proposes a way to analyze the cyber analyst’s awareness at both threat level and attack scenario level. In the threat level, analysts define typical threats for higher-level analysis based on similarity and sequentiality. The attack scenario level takes the recency, frequency and weight difference of threats into consideration to identify whether an organized series of cyber events is an attack or not. This model builds on Instance-Based Learning Theory (IBLT) and proposes a way to provide quantitative feedback regarding potential loss of an organization’s property and public image. Following on past research with IBL models we also investigate how the risk tolerance of a cyber analyst influences decision making and learning processes. We provide simulated results with this model. From these results we could conclude that the tolerance to risk is essential for performance. Lower tolerance will learn faster and make correct decisions more steadily with higher hit rate and lower false alarm rate.

1. Background

Cyber Security has attracted more and more attention worldwide since the increasing of the complexity of network environment. Cyber Security is a socio-technical research topic which requires not only the security technology, but also a deep understanding of the inner decision process of cyber analyst (i.e., a defender) and cyber attacker to enhance the security. The reliance on humans, such as the decision processes of the defenders and attackers, have become the most vulnerable and unpredictable part of cyber security system which in turn also means a great potential to support the design of cyber security systems with psychological research. In the studies of Cyber Situation Awareness (Cyber SA), we aim to understand how cyber defenders and attackers obtain the perception of the elements of network environment within a volume of time and network space, the comprehension of their meaning and the projection of their status in the near future (Endsley, 1995).

Within the cyber security community, researches have traditionally been concentrated on attackers, such as the simulation of the real-time attacks with Petri nets (Zakrzewska & Ferragut, 2011) and impact assessment based on malefactors’ behavior (Kotenko & Chechulin, 2013). However, another side of the research topics is the decision process of the cyber analyst who is responsible for protecting the systems; for example research on how different personalities affect the security decisions (Whalen & Gates, 2007).

In the decision process of a cyber analyst, we know that people learn from experience, and efficient cyber analysts tend to pick out the attack events quickly and accurately based on their experience (Dutt, Ahn, and Gonzalez, 2011; Ben-Asher & Gonzalez, 2015). Cognitively it is then interesting to measure to what extent a human cyber analyst is aware of the network situation, i.e., has reached a certain level of cyber situational awareness. And how well he/she manages to maintain and develop this awareness as time progresses. We could integrate this knowledge into the design of the future cyber SA system.

This Cyber SA system combines two important levels. The lower level relies on some cyber security softwares/tools, such as an Intrusion Detection System (IDS), firewalls, anti-virus systems and malware...
detectors. The function of this level is data collection and filtering of information. Firstly, they generate alerts and descriptions of cyber situations, providing the data used for further analysis. These systems also reduce the data dimensions which could take advantage of current high-pace development of artificial intelligence technology. In this field, many researchers are trying to find and improve artificial intelligence algorithms for effective cyber-attack detection, for example, Ahmad et al. (Ahmad, Abdullah & Alghamdi, 2009) detect probing attacks using a supervised neural network that is resilient backpropagation. This approach is primarily based on classifying good traffic from malignant events in the sense of probing of service. Apart from that, more and more Artificial Intelligence theories, such as expert systems, intelligent agents, search, learning and constraint solving (Tyugu & Branch, 2011) are also applied to the design of practical cyber defense (CD) system. However, we could not rely just on these technologies due to its proven high false alarm rate (Banford et al, 2010), that is, the systems tend to overreact on the safe network traffic. Higher level cyber situation-awareness analyses are still done manually by human mental process, which makes it labor-intensive, time-consuming, and error-prone (Barford et al, 2010; Tadda & Salerno, 2010). At the higher level, we present a simulated model based on the Instance-Based Learning Theory (IBLT) (Gonzalez, Lerch & Lebiere, 2003) with loss evaluation and we simulate the process of how cyber analysts may accumulate their knowledge about the network environment and improve their decisions.

The paper is organized as follows. In Section 2, we present a two-level (instance-level and scenario-level) memory structure. In Section 3, we elaborate the construction of this Cyber SA model based on IBLT with the order of instance-level recognition phase, scenario-level recognition phase, judgment phase, choice and execution phase and feedback phase. In Section 4, we analyze how tolerance influences the efficiency of decision making. Finally, we offer concluding remarks in Section 5.

2. Scenarios/Sequences in Memory

In the Cyber SA system, the cyber situation is analyzed by taking an orderly sequence of cyber instances (threat or non-threat) as a unit. It is reasonable since single threat instance could only cause limited damage to the system, and most of the well-organized cyber attack are combined by multiple threat instances, so the sequence of threats could be more directly related to the total loss instead of single threat in the sequence, and from the mental process of security analysts, they always refer to some typical threats which are associated orderly in a sequence and have bigger weights than normal threats to determine if it is an attack or not and what type of attacks it is. When a typical threat B1 occurs and matches well with some instance in the memory, cyber analysts will try to figure out if there another typical threat B2 which in their memory is following B1.

Correspondingly, the analysts’ experience in the memory is stored in the format of sequences or scenarios of instances, as shown in Table 1. Each scenario in the memory contains 3 parts: situation (S), decision (D), utility (U) (Gonzalez, Lerch & Lebiere, 2003). The situation part represents the typical examples of threats which contribute to as evidences to determine the status as an attack or not. Here we apply 4 attributes (target IP, IDS alert, operation success or failure and network load) of the cyber situation which is used in the past researches (Dutt, Ahn & Gonzalez, 2011; Ben-Asher & Gonzalez, 2015) The D part contains the attack decision and estimated loss, in which attack decision has 8 values ranging from 0 to 8, which represent the basic attack types (Reconnaissance, Network mapping, Port scanning, Sniffing, IP address spoofing, Session hijacking, Denial of Service (DoS) and Distributed Denial of Service (DDoS)) of this scenario and 0 represents that it is not an attack. The loss evaluation is to perform the impact analysis of the attack and allow the analysts to figure out the potential loss caused by the current cyber scenarios and make better decisions. Here we apply the multi-attribute analysis method (Butler & Fischbeck, 2002) to evaluate the loss of this scenario, and 4 attributes (represent lost revenue, lost productivity, public embarrassment and regulatory penalty respectively) to calculate the loss of property and public images, whose weights are represented by $X_n$ (n=1, 2, 3, 4) in Table 1. Note that, the 4-attribute loss value will be 0 if this scenario is not an attack. The U part is used to represent the outcome of the decision, which has the similar format with the D part.

Table 1: A new kind of instance using in the Cyber Security Decision Making. Note that, the last row presents the weights of different losses to the organization.

<table>
<thead>
<tr>
<th>Situation (typical examples of threat situations)</th>
<th>target IP</th>
<th>IDS alert</th>
<th>operation success or failure</th>
<th>Network traffic load</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileserver</td>
<td>yes</td>
<td>failure</td>
<td>3.2Mbps</td>
<td></td>
</tr>
<tr>
<td>webserver</td>
<td>no</td>
<td>failure</td>
<td>6.4Mbps</td>
<td></td>
</tr>
<tr>
<td>workstation</td>
<td>yes</td>
<td>Success</td>
<td>10.2Mbps</td>
<td></td>
</tr>
<tr>
<td>workstation</td>
<td>yes</td>
<td>failure</td>
<td>6.7Mbps</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision</th>
<th>Utility (4 attributes for loss evaluation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is an attack of type i (i = 1,2,3,...,8)</td>
<td>It is an attack of type j (j = 1,2,3,...,8)</td>
</tr>
<tr>
<td>$X_1=0.21$ $X_2=0.1$ $X_3=0$ $X_4=0$</td>
<td>$X_1=0.29$ $X_2=0.25$ $X_3=0.12$ $X_4=0.003$</td>
</tr>
</tbody>
</table>

36
3. Cyber SA Model Construction and Evaluation

In the Cyber SA Model based on IBLT, the analysts’ decision process is a sequential combination of recognition phase, judgment phase, choice phase, execution phase and feedback phase, which could be concluded as the flow chart in Figure 1. In the last section, we take a scenario/sequence of cyber instances (threat and non-threat) as the basic unit of cyber situation evaluation as it effectively represents the nature and consequence of the cyber attack and how human brain stores these associated information. Then in the first step, the recognition phase of IBL model, we need to figure out the most similar and therefore likely to be retrieved scenarios stored in memory with the current situation scenario, this process combines 2 hierarchies: instance (threat or non-threat) level and scenario (attack or non-attack) level, which will be discussed as follows.

3.1 Instance-level Recognition Phase

For each of the scenarios stored in memory, we compare them separately with the current situation scenario to define their similarity values for further scenario-level
evaluation. Then how to compare the scenarios in and outside the memory? The process could be described as Figure 2. For every instance of current cyber situation scenario (named Scenario A in Figure 2), we compare it with all the threat instances (the situation part) of scenarios in the memory, such as Scenario B in Figure 2. If Scenario A is actually of the same type of attacks with Scenario B, then A must contain highly similar instances to the sequential typical threat instances of B. This similarity manifests not only in the emergence of highly similar instances, but also in the order in which they appear.

Firstly, to retrieve the most matching threat instance in B for a specific instance of A, such as A1, the similarity values of the 4 attributes of A1 and Bi (i = 1, 2, …, n, where n represents the instance number in Scenario B) are calculated by using Euclidean distance formula to measure the distance between the two instances.

Secondly, we get the highest value from all these similarity values which in potential means that the instance with the highest similarity value is the one that will be retrieved from Scenario B to represent A1. For every instance in the sequence of current situation A, we perform the aforementioned process of instance-level recognition to retrieve an instance from B. As we mentioned before, sequentiality is a significant factor to identify the type of attack. So the activation value calculation happens in order. If instance B2 (the second instance of the sequence A’s situation part) is retrieved for A1, then instance A2 will continue to find the retrieval instance with the highest activation value beginning from B3 (the next one to B2).

Now, we could get the instance sequence of A and the retrieval sequence from B with relative similarity values, as shown in Table 2. The first column is the threat instances of A identified by the aforementioned activation method, and the second column serves as the representative instances in B with the highest similarity values (shown in the third column) corresponding to the first column.

Table 2: Information gained from instance-level recognition of A and B.

<table>
<thead>
<tr>
<th>Instances in A</th>
<th>Corresponding retrieved instances in B</th>
<th>Similarity values</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>B2</td>
<td>87%</td>
</tr>
<tr>
<td>A3</td>
<td>B3</td>
<td>92%</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

3.2 Scenario-level Recognition Phase

Then we apply the weighted sum formula (Formula 1) to calculate the overall similarity $SIM_{AB}$ of Scenario A and Scenario B, which will be further used to determine whether to combine B or not to calculate the utility of current scenario A in the judgment phase.

$$SIM_{AB} = \frac{1}{n} \sum_{i=1}^{n} w_k \cdot Similarity_{AIB_k}$$  \hspace{1cm} (1)

In Formula 1, $n$ refers to the total number of instances in A. $Similarity_{AIB_k}$ refers to the Euclidean distance of the 4 attributes of cyber situation of Instance Ai and Bk, where Bk is the most similar instance to Ai in the Scenario B, corresponding to the second column in Table 2. $w_k$ is weighted value representing how important is the emergence of instance Bk for the decision of B. They could be set to be 1.0 for the convenience of the simulation. The valuable $Similarity_{AIB_k}$ corresponds the third column of Table 2,
which represents the similarity of the instance Bk and instance A1.

Activation is the reflection of the scenario usefulness in the past and the relevance of that scenario to the current instance context (Gonzalez, Lerch & Lebiere, 2003), quantitatively given by the following formula (Dutt, Ahn & Gonzalez, 2011),

$$Activation_B = \ln \left( \sum_{i=1}^{j} (t - t_i)^{-d} \right) - SIM_{AB} + \varepsilon_i$$ (2)

where j refers to the total number of the trials in which scenario is once retrieved and used. \(t_i\) refers to the serial number of trials in which scenario B is used for the \(i^{th}\) time. The first part of the equation sign’s right side is the base-level learning mechanism and reflects both the recency and frequency of use for the \(i^{th}\) instance since the time it is created (Dutt, Ahn & Gonzalez, 2011). The second part is the similarity component and represents the mismatch between a situation’s attributes and the situation (S) part of instances in scenario B of memory. \(\varepsilon_i\) is the noise value that is computed and added to an instance Bi’s activation at the time of its retrieval attempt from memory (Dutt, Ahn & Gonzalez, 2011).

Here, we could make the decision if A is an attack or not by analyzing and combing the highest activation values. We could set a threshold for the activation values to identify if it’s an attack or not. This threshold value could be set as default by the security analyst and represent analysts’ tolerance towards cyber attack. A risk-averse security analyst will have a lower threshold value for activation and be more sensitive to the cyber events. He might take a cyber event to be an attack even the highest activation he/she gets from the aforementioned process is very low. However, a risk-taking analyst with higher tolerance will have a higher activation threshold and will be less possible to identify a specific event to be an attack unless the highest activation value are higher than its threshold. In the simulation, we could set the high threshold as

$$T_{\text{high}} = 90\% \times (\text{Max}_{Activation_{Bi}} - \text{Min}_{Activation_{Bi}}) + \text{Min}_{Activation_{Bi}}$$

and low threshold as

$$T_{\text{low}} = 70\% \times (\text{Max}_{Activation_{Bi}} - \text{Min}_{Activation_{Bi}}) + \text{Min}_{Activation_{Bi}}$$

Applying recurrently the instance-level and scenario-level recognition, we could get the similarities of A and all the scenarios stored in the memory. Then in the judgment phase, we could pick the highest value \(Activation_{\text{highest}}\) of activation values and use a threshold value to test it. Once the \(Activation_{\text{highest}}\) is higher than the threshold value, scenario A is considered to be the same type of cyber events as the scenarios with \(Activation_{\text{highest}}\).

### 3.3 Utility Calculation in Judgment Phase

The utility in this system refers to the actual decision and the loss of a specific successful attack. The loss evaluation part will help the security analyst to evaluate and perceive the potential damage to the organizations before the results become apparent and take effective measures selectively to fulfill the most important security requirements of specific business fields. We apply the concept of “multi-attribute analysis” (Butler & Fischbeck, 2002) to perform the utility calculation of cyber attack.

This method could not only consider the asset valuation but also capture the essence and uncertainty of the underlying loss. For example, it is difficult to quantify the damage that a successful attack does to the corporate image. However, this image loss may be far more important to the organization than the actual loss of revenue caused by the attack or the hours it takes to recover for service (Butler & Fischbeck, 2002). Organizations from different business fields may have different security requirements. When the attack is executed, security analyst needs to make decisions to protect the system based on the security requirements and potential loss amount of different aspects, such as lost revenue, lost productivity, corporate embarrassment and regulatory penalty using the following formula.

$$U(x_1, x_2, x_3, x_4) = \sum_{i=1}^{4} w_i \cdot v(x_i),$$ (3)

Where \(v(x_i)\) is a single-attribute value function defined over levels of \(x_i\), which is given by

$$v(x_i) = \frac{x_i}{x_i^*}$$ (4)

Where \(x_i\) is the loss amount of the \(i^{th}\) attribute and \(x_i^*\) is the maximum value for the attribute provided by the security managers. This function ensures that \(0 \leq v(x_i) \leq 1\) to eliminate computational problems caused by the different units of measure. Note that, for the convenience of simulation, we assume the values of \(v(x_i)\) as a whole as shown in the rightmost 4 columns of Table 1.

The weighted value \(w_i\) is defined by Swing-Weight Method (Butler & Fischbeck, 2002), in which the security manager is asked to discover new types of loss regarding a hypothetical situation, and give the threat results in the worst situation based on their experience.
The security manager is then asked to rank the hypothetical outcomes from the most important to the less important. Then, the decision maker assigns the scores to all of the attributes based on the relative importance to the most important attribute. Finally, the actual weight values we use are defined by dividing each of these values by the sum of all of them. Note that, the weights will vary from different business fields, in the experiment, we assume them to be steady for a specific business field, i.e. for a specific organization system.

3.4 Choice and Execution Phase

Based on the aforementioned phases, if the system has some similar experience for current situations, then we now could retrieve the most possible scenario from memory to represent the current situation and apply the U part and the weighted values for the system to evaluate the potential loss. And immediate reactions to the attack based on specific security requirements and potential loss are required to execute. Otherwise, if the system doesn’t have a similar experience for current situations, i.e. the Activation_highest is found lower than the threshold, the system will turn to heuristic methods instead.

3.5 Feedback Phase

In the IBL model, to improve the cyber analysts’ performance, the actual situation and loss of the current cyber scenario should be updated to the memory and provide knowledge for future decision making. We first decide how accurate the results are by comparing the relative error of the utilities, and update some properties of good scenarios to make it more possible to be retrieved in the future and otherwise to expire the useless scenarios.

The feedback here is a combination of two important categories: creating new knowledge or scenarios of cyber security and strengthening existing memories through use. When the system doesn’t have experience of specific attack, i.e. the highest activation value of current scenario is low, new scenario data with valid decision and utility should be added to the memory. While the system makes the correct decisions (hits and correct rejections), the system should strengthen the decisions based on the utility, i.e. the difference of actual loss and supposed loss. The use frequency of the retrieved scenario in current trial \( t_i \) is updated by

\[
t_i = 1 - \frac{Loss_{utility} - Loss_{decision}}{Loss_{utility}}, \tag{5}
\]

If the relative error of loss is 0, meaning that the analyst gains the correct loss analysis from the memory, the decision is highly correct and should be strengthened greatly; while the relative error amounts to or even bigger than 1, meaning that the analysis of the loss is far from the reality, then the decisions should not be strengthened.

The loss attributes could be set by cyber managers based on specific requirements of the organizations, although we generally set it as 4 attributes with multi-attribute analysis method (Butler & Fischbeck, 2002) as shown in Section 2.

4. Results

In the last section, we have built a complete model for cyber security analyst’s decision learning and making process. Note that we have set a threshold for activation values to represent an analyst’s risk tolerance towards cyber events. In this section, we test how cyber analysts’ risk tolerance influence their cyber situation learning and decision making by analyzing the curves of hit rates representing cyber analysts with different tolerances towards the cyber attack. The results could be concluded in the following Figure 3-6.
Figure 5. Decision learning and making with activation threshold = -1.

Figure 6. The influence of different activation values (risk tolerances) on hit rate of cyber situation awareness.

Figure 7. The influence of different activation values (risk tolerances) on false alarm rate of cyber situation awareness.

In the Figure 3-5, the colorful bars show the changes of hit, false alarm, correct rejection and miss over the simulation of 50 trials. The horizontal axis represents the trial number while the vertical axis represents the hit rate ranging from 0 to 1. The areas of different colors in each trial’s colorful bar amounts to the relative quantity of hit, false alarm, correct rejection and miss.

At the beginning of every simulation, the miss and false alarm dominate the outcomes of cyber analysts’ decisions. However, it is clear that the number of hit and correct rejection increases with the trial number increases while the number of false alarm and miss decreases at the same time. They all reach a stable level but the speed to make it is different due to the differences of risk tolerance levels, i.e. the activation threshold values. Figure 3 represents a relatively slower and worse decision-making process when compared with Figure 5. In Figure 3, the risk-taking cyber analyst with a relatively high tolerance level (high activation threshold) takes 35 trials to reach its stable level (with fluctuation) with the hit rate of 93%. However, Figure 5 presents a decision-making process with faster speed and better performance. In Figure 5, the cyber analyst with a relatively low tolerance level (low activation threshold) takes 25 trials to reach it stable level with 100% hit rate.

Furthermore, Figure 6-7 show the changes of hit rate and false alarm rate over 50 trials. Furthermore, the fluctuation of three learning curves vary a lot with the biggest fluctuation in Figure 3 and the smallest fluctuation in Figure 5. We could conclude that the analyst with lower tolerance will learn faster and make better decisions more steadily with higher hit rate and lower false alarm rate. Clearly, when the ACTIV threshold is smaller, the performance of the model is better. However, when the threshold reaches the border line, the performance could not be improved anymore.

5. Conclusion

Cybersecurity is critical across military and civilian systems, and it is necessary to understand the decision making process of cyber analysts and combine it into the design of future cyber security systems. In this paper, we have presented a computational model with instance and scenario recognition and feedback of loss evaluation based on Instance-Based Learning Theory (IBLT). The simulation also suggests that a cyber analyst with lower tolerance towards the attack may have better performance in the decision making.

For future work, we are going to verify the results by comparing them with the human analyst’s experiments and develop and transfer this modeling method into a toolkit to support the higher-level decision making on cyber situations in the cyber security systems.

6. References


**Author Biographies**

Cleotilde Gonzalez is Associate Research Professor and Director of the Dynamic Decision Making Laboratory, Department of Social and Decision Sciences, Carnegie Mellon University.

Fangting Wu is the visiting student of the Dynamic Decision Making Laboratory from Sun Yat-sen University.
Semantic Labeling of Objects in a Simulated Environment

Ralph W. Brewer  
MaryAnne Fields  
Arne J. Suppé*  
Dave Wagner**  
US Army Research Laboratory  
Aberdeen Proving Ground, MD 21005  
410-306-4244  
*Carnegie Mellon University, The Robotics Institute  
5000 Forbes Ave, Pittsburgh PA 15213  
**General Dynamics Robotic Systems  
1231 Tech Ct. Westminster, MD 21157

ralph.w.brewer.civ@mail.mil, mary.a.fields22.civ@mail.mil, suppe@andrew.cmu.edu, dwagner@gdrs.com

Keywords:  
Perception, Semantic Labels, Simulation, Human-Robot Interaction

ABSTRACT: Situational awareness and trust are essential to promote seamless interactions between a robot and human. Perception of the world by each of these is through different lenses. Robots need a better understanding of the world through classification of the scene and semantic understanding of objects in the local environment. Perception algorithms provide a way for robots to gain knowledge of the environment. Training and testing of these algorithms take significant effort on the part of researchers. With the development of a simulation tool, researchers are now able to systematically validate their perception algorithms in a simulated environment before conducting research in the field.

1. Introduction

The U.S. Army Research Laboratory’s (ARL’s) Robotics Collaborative Technology Alliance’s (RCTA) is a collaboration of industrial, academic and government scientists focused on developing robotics technologies for future autonomous systems (Bornstein, 2012). Research in the program is divided into Perception, Intelligence, Human Robot Interaction (HRI) and Dexterous Manipulation/Unique Mobility (DMUM).

Semantic scene understanding is an important area of research in the field of perception. Semantic classification is a technique for classifying images, or sensor output, based on the meaning of objects or activities in the scene. Its goal is to assign human-understandable labels to the elements of the image. This type of image classification supports human robot interaction enabling soldiers to use natural phrases such as “Go to the back of that building and watch the door in the rear.” (Lennon et al, 2013) This task requires the robot to have the ability to identify personnel, buildings, doors, obstacles, and other objects in the scene. In this case, the robot must identify concrete objects such as door and building and abstract concepts such as “rear” and “back”.

The RCTA has developed a classifier (the Carnegie Mellon University (CMU) semantic classifier) to label pixels in a color digital photograph of an outdoor scene as one of the following: sky, tree, asphalt floor, grass, building, object, concrete floor, or gravel floor (Munoz, Bagnell, and Hebert, 2010). Early live experiments demonstrated that the classifier was able to successfully segment and label most pixels in most images, however, live experiments are expensive undertakings requiring a commitment of time and resources from both the researchers and evaluators (Lennon et al, 2013). In addition, programs such as the RCTA integrate many different research efforts onto a single robotic platform during the live experiments. Integration issues and software bugs may reduce the quality and quantity of data collected during an experiment.

This paper describes our efforts to supplement the live experimentation process in the RCTA with a simulated environment that supports the development of these algorithms. We are able to provide a rich virtual environment, with buildings and objects commonly found in an urban setting, for image collection. Since we have complete knowledge of the environment, we also provide “ground truth” to systematically validate the algorithms. The remainder of this paper is organized into 6 sections. The next section provides some background on commonly used techniques in object recognition and semantic scene understanding. Section 3 provides some general background on our simulation tool. In section 4, we discuss our recent improvements to this tool that support the development of semantic classification algorithms. The experimentation section talks about a
specific experiment in August 2014. This is followed by the results of that experiment. We conclude this report with a discussion of planned future improvements to support the development of semantic perception.

2. Semantic Image Segmentation

An image presents a two dimensional representation of a scene in the form of pixels with red, green, and blue (RGB) attributes. Some sensors, such as the Kinect camera provide a fourth dimension of information - depth. This gives context and spatially separates portions of the scene. Semantic image segmentation attempts to classify each pixel or element in an image based on its color, texture, location or other factors. Early algorithms focused on separating the background of an image from its foreground. Current algorithms identify pixels as members of classes such as sky, tree, or building (Yao et al, 2012). Some of the research in this area attempts to increase the number of classes and to decrease the classification time. Many of the algorithms use supervised machine learning approaches to train a robot to identify instances of the classes. This approach requires a large database containing many examples of the classes. Objects can be particularly challenging. General classes, such as “tree”, have a lot of in-class variation (oaks do not resemble pine trees). Variations in object pose or scene lighting can result in very different appearances of these objects.

There are some datasets and public image repositories that are used in computer vision research, such as Google Images and Flickr, as well as visual recognition benchmarks like Caltech 101 (Fei-Fei, R. Fergus, and P. Perona, 2006), ImageNet (Deng, et. al., 2009), and the University of Washington RGB-D Object Database (Lai, 2013). The University of Washington dataset was collected using a Kinect sensor and organized into 51 categories. Tools such as LabelMe (Russell, et. al, 2008) allow researchers to segment regions of the images by hand and label them by name or category. Even with the publically available databases and the LabelMe annotation tools, developing datasets to support specific computer vision research efforts is laborious and time consuming. Further, objects in a scene from the dataset may only be observed from a few angles which make generalized object recognition more challenging.

The RCTA field environment along with objects in it does not fit well into using one of the public databases of labeled objects. An alternative to using LabelMe to annotate objects from these databases is to classify the objects in a 3D virtual environment. Labeled objects can be viewed from any angle and distance and in open or cluttered environments. Scene properties, such as lighting and the amount of clutter in that scene; and object properties such as texture, color or the meaning of the semantic label can be varied to quickly generate a large dataset of semantically labeled scenes. This approach does require time to build the desired objects in a 3D modeling tool so that they have a secondary semantic layer. However, once the objects are built, researchers can quickly use the data to train and evaluate their semantic perception algorithms. One of our goals is to investigate ways to use simulated data to support the learning process.

3. Simulation Tools

The simulation environment we use for this project is Robotic Interactive Visualization and Experimentation Technology (RIVET) (Gonzales et al, 2009). It is built upon the Torque 3D (T3D) Game Engine designed by Garage Games (Game Development Tools, 2014). We use a distributed client server paradigm with one instance of RIVET as a server - other instances of the engine connect to this server as clients allowing multiple users to participate simultaneously in the simulation. We also use this client/server paradigm to connect the simulated robotic vehicle and sensors to external software components. RIVET has proven an invaluable tool for conducting training, experimentation, testing and debugging of algorithms. Tools like RIVET provide an essential initial evaluation of robotic algorithms and concepts in multiple environments that can be varied systematically to exercise specific algorithm features.

To support our research in the RCTA program, we built a scale replica of the Fort Indiantown Gap Combined Arms Collective Training Facility (CACTF) which supports training activities for Military Operations in Urban Terrain (MOUT) using United States Geological Survey digital terrain map. Each virtual building is a high fidelity, fully functioning model constructed in 3DS Max using blueprints and pictures of the facilities. The interiors are also functional - objects such as tables and

![Figure 1. A comparison between a physical experimentation facility (left) and its virtual simulation (right).](image)
chairs can be placed appropriately and dynamic objects are able to traverse through the halls, stairways, and rooms. Figure 1 shows a comparison between the actual site and its virtual representation.

Figure 2 illustrates some of the differences between the simulated and real worlds. The virtual gas station on the right does not include the sewer drains, or the manhole covers shown in the photograph on the left. Also, some of the building features are missing. This level of detail is sufficient for our current semantic classification algorithms – more detail will be added to the virtual world, as necessary. These high fidelity virtual worlds allow us to conduct matched field and virtual tests.

Our virtual experiments mirror our field tests – the robot drives through the CACTF using its sensors to collect information. The current vehicle used in our experiments is the Husky by Clearpath Robotics. It is represented in the simulation as a multi-component rigid body model using the Nvidia PhysX physics engine. The model is constructed in 3DS Max with the PhysX plugin to define vehicle’s rigid body attributes. The vehicle can be equipped with various simulated sensors such as the Bumblebee stereo camera, Hokuyo Lidar, and Micro Ladar. Users specify the location and orientation of these devices using a graphical user interface we developed. The sensor models can be local or remote clients in the simulation system. The use of the client sensor allows engineers to utilize a separate computer with its own CPU and video card to process the sensor output thereby freeing up computing power on the server and also emulating the hardware configuration of the real robot.

Exporting the physical state of the vehicle supports interaction between the vehicle control software and the virtual environment. There are two simulated versions of the vehicle that differ in the method used to control the platform. The first model has no autonomy and the platform is commanded through a joystick to allow researchers to collect data from the simulated sensors. The second model is autonomous and is designed with a platform mobility software component that allows the robot to receive commands from another computer to maneuver it about the simulation. This model allows the perception and intelligence component of the robot to interact.

To support the RCTA semantic perception research in RIVET, we created a semantic sensor to portray object classification data. This sensor data provides ground truth information for perception researchers as they develop algorithms to semantically label the environment. It can also supply data to reasoning and cognitive algorithms allowing those research efforts to use perception information that is currently unavailable due to algorithm speed or other issues. We created both a semantic classifier sensor, a custom tool to provide labeled data to algorithms outside the simulation, as well as a methodology to set up a semantic representation for the virtual environment. Truly autonomous robotic platforms will need tight integration between cognitive and perception components. This semantic simulation allows each field to make progress in parallel.

4. Adding a semantic sensor

Modifying RIVET to support semantic perception is a two step process. We need to add an extra texture to each of our object models that is visible to our semantic sensor. We also need to create a model for this sensor. In our previous work, we developed a model of the bumblebee stereo camera (Dean et al, 2014). The model provides two images of the same scene that are offset by the distance between the right and left cameras. We modify this model so that the left camera provides a standard RGB image of the scene, while the right camera provides a labeled image of the same scene using a Direct X shader to read a “semantic” texture layer from each of the objects in the virtual world. An example of the output of the semantic sensor is shown in Figure 3. Note that the labels in the right-hand image are shown as colors.
As with many game engines, objects are constructed using an external 3D modeling tool and imported into RIVET’s virtual world. In this discussion, we will use 3D Studio Max (3DS Max) – other 3D modeling tools will also work. Our basic approach is to extend the textured 3D models normally used in a game environment by adding an additional texture layer visible only to the semantic sensor. We will use the church shown in Figure 3 to illustrate our process. In a realistic rendering of the church, it is divided into a number of sub-objects, such as the roof, doorway, or tower, which are textured separately using UVW maps. A UVW mapping is used to project a texture map onto a 3D object while maintaining control of the scale, position and orientation. The church uses 24 distinct texture maps. Often, UVW maps are layered to provide more realistic textures – for instance dirt or graffiti could be added as an additional layer in our church model. To support our work with semantic labeling, we add an additional UVW layer that represents the semantic layer. This additional layer will only be visible using the semantic sensor DirectX shader designed to show the overlay mapped textures.

To allow more flexibility in our virtual experiments, we do not assign specific semantic textures to the models in 3DS. We assign the semantic coloring within the simulation, using the world editor, instead. In general, most objects contain multiple segments, each with a different semantic label. In the case of the church, the 24 visible textures are replaced with 4 textures indicating the building structure, windows, doors and the concrete stairs. Objects do not necessarily need to be segmented. An example of this is in Figure 4 as the gas pump in the right background past the church which has one color signifying it as a single semantic object. Each object class is assigned a specific, but arbitrary color. For the purposes of this research, there were 16 separate classes of objects as outlined in Table 1. The index value is assigned to each pixel based on its classification. The final column shows the color associated with the RGB value.

<table>
<thead>
<tr>
<th>INDEX</th>
<th>LABEL</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>asphalt</td>
<td>26</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>1</td>
<td>concrete</td>
<td>172</td>
<td>172</td>
<td>172</td>
</tr>
<tr>
<td>2</td>
<td>grass</td>
<td>48</td>
<td>71</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>gravel</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>sky</td>
<td>100</td>
<td>161</td>
<td>182</td>
</tr>
<tr>
<td>5</td>
<td>wall</td>
<td>255</td>
<td>172</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>door</td>
<td>255</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>window</td>
<td>0</td>
<td>255</td>
<td>243</td>
</tr>
<tr>
<td>8</td>
<td>vehicle</td>
<td>190</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>pole</td>
<td>192</td>
<td>173</td>
<td>152</td>
</tr>
<tr>
<td>10</td>
<td>tree</td>
<td>114</td>
<td>72</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>hydrant</td>
<td>182</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>object</td>
<td>36</td>
<td>82</td>
<td>88</td>
</tr>
<tr>
<td>13</td>
<td>pump</td>
<td>245</td>
<td>155</td>
<td>214</td>
</tr>
<tr>
<td>14</td>
<td>cone</td>
<td>10</td>
<td>104</td>
<td>95</td>
</tr>
<tr>
<td>15</td>
<td>dirt</td>
<td>248</td>
<td>147</td>
<td>108</td>
</tr>
<tr>
<td>16</td>
<td>stairs</td>
<td>255</td>
<td>0</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 1. Semantic Values

5. Experimentation

As we stated earlier, we have two goals for our simulated semantic sensor work. The first is to provide labeled perception data to RCTA researchers for intelligence, cognitive robotics, or HRI research. We can use our semantic sensor to send labeled images to graphical user interfaces and other external programs to be used in this research. Our tool will be able to supply usable perception data for this research even as our perception research continues. Note that our semantic sensor provides perfect information – the image on the right-
hand side of Figure 3 gives a “ground truth” semantic labeling. At this time, we do not plan to introduce any uncertainty in the simulated semantic labeling.

Our second goal is to support semantic perception research by providing virtual worlds that can be used to train and test semantic labeling algorithms. Ultimately, we want to supplement real world images with images from our virtual environment in both the training and testing phases of the algorithm. However, virtual environments do not capture all the natural variations in structure, lighting, or texture so simulated images can be very different from those in real world. In light of this, it is important to validate the behavior of CMU semantic classifier in our virtual environment. Our hypothesis is that the CMU classifier training can be trained and tested with images from the RIVET virtual world. In our experiments, we use both images from the semantic sensor – the RGB image provides raw input for the classification algorithm and the labeled image can be used to label training images or as ground truth to evaluate the algorithm’s performance on test images.

The semantic classification code named ROSHIM (Robot Operating System Hierarchical Inference Machine) developed at CMU uses a stacked hierarchical labeling framework (Munoz, 2013) for semantic segmentation. Given an input image, the algorithm creates a hierarchy of regions, ranging from almost the entire image at the top of the hierarchy to finer super-pixels at the lowest level. The superpixels are labeled with a scene classifier using Scale-Invariant Feature Transform (SIFT) (Lowe 2004), Local Binary Patterns (LBP) (Ojala, Pietikainen, and Maenpaa 2002), and texture features which allow us to segment and localize labeled objects. At each level of the hierarchy, ROSHIM reports a distribution of labels for each region at that level. The label distribution at one level is passed as features to the classifiers for regions of the lower layer of the segmentation hierarchy (Munoz et al, 2010). This process should work on the virtual images generated from our simulation environment.

Our validation experiments used the virtual CACTF environment. For this study, we changed the field of view of the semantic sensor to 120 degrees and we also reduced the number of classes by merging some of the classes in Table 1. This was done to match the camera parameters and label set used in live experiments. For instance, instead of having a separate class for window, wall, building, and door these were merged into one class – “Wall.” Table 2 lists the reduced set of classes used in our experiments along with the original classes included in each of the reduced classes.

<table>
<thead>
<tr>
<th>Semantic Object</th>
<th>Joined Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>concrete</td>
<td>CONCRETE FLOOR + SIDEWALK</td>
</tr>
<tr>
<td>wall</td>
<td>WINDOW_Opening + WALL + BUILDING + DOOR_Opening</td>
</tr>
<tr>
<td>object</td>
<td>PERSON + UNKNOWN</td>
</tr>
<tr>
<td>cone</td>
<td>TRAFFIC_CONE + TRAFFIC_BARREL</td>
</tr>
<tr>
<td>tree</td>
<td>TREE + SHRUB</td>
</tr>
</tbody>
</table>

Table 2. Reduced semantic classes used in the validation experiments.

To collect the sensor data, we drove the simulated Husky robot through an area of the simulated world and recorded approximately 200 image pairs, RGB and labeled, from the simulated semantic sensor. These images were mirrored to increase the size of the training data set.

The labeled images needed further processing before they could be used as ground truth. For example, not all the labeled classes are used so some classes are merged. Most importantly, the labeled images are generated from a rendering pipeline that adds some lighting effects that cannot be turned off. The sky and trees, in Figure 4, show variations in color which may lead to misleading results in our analysis. Other surfaces in the semantic images have smaller yet still significant variances in color. CMU developed a Matlab script to remove the color variations, add labels to the unlabeled pixels, and

![Figure 4. Comparison of the original semantically labeled image (left) and the processed ground truth image (right)](image-url)
merge the classes to the set shown in Table 2. Note that the shading from the trees has been removed from the left image leaving just the single color outlining the forest in the right. Figure 4 shows a comparison of the original labeled image and the final ground truth image. Note that we store the final ground truth in textual form.

All experimental data was collected using the robot camera within the simulation. By driving the robot around the simulated FTIG, collection of new and unseen data is possible from numerous different angles, ranges, and illumination. The performance of the CMU classifier was evaluated using 5-fold cross validation. That is, 4/5 of the data is used for training a new model, and 1/5 is used for testing. The results for that 1/5 are shown in table 4. The process is repeated 5 times and then the result is used to generate the confusion matrix.

6. Results and Discussion

The classifier in ROSHIM processed each RGB image to produce an image with each pixel labeled with the color codes from Table 1. This image was compared to the corresponding ground truth information. Table 3 shows a color representation of the confusion matrix. Each row represents pixels whose true label is the actual label associated with the row. Each cell in a row gives the percentage of pixels labeled with the predicted label. The blue cells indicate a low probability of mislabeling the object. In general, the percentages range from 0.0% – 0.1%. A notable exception is that 6% of the wall pixels are mislabeled as window pixels. Along the diagonal, the percentage of correctly classified pixels ranges from 99.9% (sky) to 94% for object and window.

Our hypothesis is that simulation images are useful enough to be able to test the validity of the classifier. To us a precision and recall of higher than 90% for each class of objects would prove the robustness of the simulation for semantic object classification. There are four possible outcomes: true positive (t_p), true negative (t_n), false positive (f_p), false negative (f_n). The precision score (Equation 1) is determined using pixel-by-pixel comparison of the ground truth and analyzed images. Precision is also called a positive predictive value. The sensitivity or recall is the true positive rate as given by Equation 2. Table 4 shows the precision and recall values for each of the classes used in our experiment.

![Figure 5. A classified view of the church](image)

The analysis of this test scenario shows the value of simulation in semantic labeling of images. The ROSHIM results shown in the confusion matrix, Table 3, was determined with an average of 5 cross validation folds of 191 dissimilar images coupled with a mirror of each, and using a wide field of view to capture images. In machine learning, the F1 (Equation 3) score is used to measure overall test accuracy with higher scores indicating greater accuracy.

\[
\text{Precision} = \frac{t_p}{t_p + f_p} \quad (1)
\]

\[
\text{Recall} = \frac{t_p}{t_p + f_n} \quad (2)
\]

\[
\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)
\]

The overall micro-averaged F1 score for this test was .992 on a scale from 0 to 1 with 1 being the best possible result. There is room for improvement, but it does show that the algorithm is able to differentiate a small set of
objects in a scene. As the scene becomes more cluttered with additional objects to identify and classify, the algorithm will need to be re-evaluated. Through the use of RIVET, researchers can use the simulated environment to identify anomalies in their algorithms and then strengthen them for the research assessments in the field trials.

While the overall results are promising, there are some differences in the simulation to note. Figure 6 shows the super-pixel segmentation for the simulated image of the church. This segmentation contains some differences from similar images from the field test, particularly in the sky. Normally, one expects the super pixels to have a more compact, regular shape similar to those seen in the bottom half of Figure 6. The long, skinny segments seen in the sky could be caused by a couple of things related to the game engine. The first is the way in which the game engine takes the image used for the sky and creates a 3 dimensional look to it. The second is the camera used to gather the images is set to 120 degree field of view which may cause this fanning out of the sky.

![Super pixel segmentation of the church showing anomalies in the sky segmentation](image)

**Figure 6. Super pixel segmentation of the church showing anomalies in the sky segmentation**

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>asphalt</td>
<td>0.993</td>
<td>0.999</td>
<td>0.996</td>
</tr>
<tr>
<td>concrete</td>
<td>0.994</td>
<td>0.998</td>
<td>0.996</td>
</tr>
<tr>
<td>grass</td>
<td>0.989</td>
<td>0.996</td>
<td>0.992</td>
</tr>
<tr>
<td>gravel</td>
<td>0.987</td>
<td>0.963</td>
<td>0.975</td>
</tr>
<tr>
<td>sky</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>wall</td>
<td>0.989</td>
<td>0.991</td>
<td>0.990</td>
</tr>
<tr>
<td>door</td>
<td>0.972</td>
<td>0.967</td>
<td>0.969</td>
</tr>
<tr>
<td>window</td>
<td>0.939</td>
<td>0.912</td>
<td>0.925</td>
</tr>
<tr>
<td>vehicle</td>
<td>0.974</td>
<td>0.984</td>
<td>0.979</td>
</tr>
<tr>
<td>pole</td>
<td>0.956</td>
<td>0.946</td>
<td>0.951</td>
</tr>
<tr>
<td>tree</td>
<td>0.947</td>
<td>0.929</td>
<td>0.938</td>
</tr>
<tr>
<td>object</td>
<td>0.939</td>
<td>0.937</td>
<td>0.938</td>
</tr>
<tr>
<td>pump</td>
<td>0.967</td>
<td>0.972</td>
<td>0.969</td>
</tr>
<tr>
<td>cone</td>
<td>0.997</td>
<td>0.995</td>
<td>0.996</td>
</tr>
<tr>
<td>stairs</td>
<td>0.956</td>
<td>0.876</td>
<td>0.914</td>
</tr>
</tbody>
</table>

Table 4. Precision and recall values for each class

7. Conclusion

This paper demonstrates that the CMU semantic classifier gives acceptable results when run in the simulated CACTF environment. In the future, we plan to add more semantic objects to RIVET to increase the complexity of the scene. With that we will be able to do a more thorough evaluation of the semantic classifiers in this improved environment. This simulation can also support parallel development without the use of a perception algorithm using only the ground truth data.

8. References


Author Biographies

RALPH W. BREWER is a Modeling and Simulation Research Analyst at the US Army Research Laboratory with 5 years experience. He is also involved in efforts to develop modeling and simulation tool to further the development, analysis and testing of autonomous ground robotic vehicles for ARL’s Robotic Collaborative Technology Alliance.

MARYANNE FIELDS is a robotics researcher at the US Army Research Laboratory with 15 years of experience in robot planning and perception. She has also involved in efforts to develop modeling and simulation tool to further the development, analysis and testing of autonomous ground robotic vehicles.

ARNE J. SUPPÉ is a PhD student at Carnegie Mellon University’s Robotics Institute. His interests are in applied computer vision for field robotics. Prior to this, he worked for almost 10 years as a Senior Research Programmer at the Robotics Institute.

DAVE A. WAGNER has 7 years experience in real time, hardware in the loop, simulation and modeling of multiple platform class and sensor types while working at General Dynamics Robotics System. He also has over 28 years of experience working on real time systems.
Agents and Decision Trees from Microdata

John B. Nelson, William G. Kennedy
George Mason University
4400 University Drive
Fairfax, VA 22030
JBN@pathdependent.com, WKennedy@gmu.edu

Ariel M. Greenberg
Johns Hopkins University / Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
Ariel.Greenberg@jhuapl.edu

Keywords:
Microdata, Probabilistic Graph Models (PGMs), Decision Trees, Agent-Based Modeling (ABM)

ABSTRACT: This paper discusses the development of a model of the household migration behavior of a nation’s population. From information synthesized from across available microdata sources which are each temporally, spatially, or topically inconsistent in coverage, we learned decision trees and instantiated agents in an agent-based model. The generative results of the whole-country simulation of this ABM mimicked the observed macro-level findings, engendering confidence in this method to develop agents and decision trees from microdata.

1. Introduction

The Air Force Research Laboratory’s National Operational Environment Model (NOEM) is an ambitious project to rapidly populate models of any arbitrary country from a wide variety of open source data, and enable in silico experimentation on these models. It integrates several model types. Most modules were first implemented as system dynamic models. A radicalization model, part of NOEM’s behavior module, was implemented as an agent-based model adapted from (Epstein, Steinbruner, & Parker, 2002) and is still expanding to accommodate migration and crime related behaviors. In this paper, we summarize our effort over the last year in developing a data-driven agent-based model (ABM) of migration in the Republic of Colombia providing future behavior module capabilities to advance NOEM’s migration and crime modules.

2. Data Sources

Broadly, we seek two classes of data to feed analysis and modeling: microdata and event data. This fine resolution is necessary if we assume heterogeneous decision making, a hallmark of agent-based modeling. Aggregate statistics are insufficient. We need to have realistic household socio-demographic variables and resource endowments. Here we enumerate those data sources ultimately employed.

1. IPUMS–International (Integrated Public Use Microdata Series, International) is a clearinghouse for microdata samples, which are anonymized but statistically valid samples from census data. For Colombia, the census source is DANE (Departamento Administrativo Nacional de Estadística). This is a very high resolution demographic and socio-economic sample. Roughly 1:10 households, geographically covering all of Colombia, are represented.

2. The Barometer series, Latinobarometer (LB) for Colombia, provides results from cross-sectional surveys that gauge public opinion on topics political, economic, and security-related.

3. Data Synthesis

A perennial challenge to performing social science research on collections from the developing world is data acquisition. Researchers interested in conducting country-specific modeling or statistical studies routinely encounter a dearth of particular microdata – survey or census data consisting of individual records from persons or households within a desired country, time frame, and topic set. On the rare occasion when such a set is available, it is usually from a small-N study.

This document is a product of work sponsored by the Air Force Research Laboratory (AFRL). Work is executed under the provisions of NAVSEA Contract # N00024-13-D-6400, Task Order # 169, Task ID # MC204. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of AFMC, AFRL, or the United States Government.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.
Microdata clearing houses such as the International Household Survey Network (IHSN), nationally-hosted microdata repositories like the Colombia National Statistical Office’s DANE/ANDA, and topic-oriented regional surveys like the Barometer series (Asia, Afro, Latino, and Arab Barometer) provide microdata sets of independently conducted collections, each set distinct in time, topic, or geographic granularity (if not distinct geographic coverage).

Integrated sets in the spirit of the University of Minnesota’s Integrated Public Use Microdata Series (IPUMS), like the NSF’s Terra Populus and the NIH’s Integrated Demographic and Health Series, harmonize within topics by identifying the fuzzy intersection of survey questions across time and space and grouping related questions into variables ranging from detailed to general.

Dataset fragmentation confounds research into interaction between topics covered by separately-sourced data sets, and stymies research into predisposing and precipitating factors that occur prior to data collection of the consequent behavior under study. For example, an exploration into the relationship between the topics of criminal victimization and migration behavior requires spanning barometer and census sets (topic synthesis), and a study into resource-driven migration requires backward imputation in time of the resources available to respondents during the period of their departure (time shifting).

Here we describe a methodology to synthetically unite microdata disparate in time and topic for use in statistical studies, and to seed realistic agents for agent-based models of complex social systems on which to run simulation experiments.

Such a methodology enables the production of a unified data assimilation pipeline (a synthetic repository of repositories) that compiles statistically reasonable synthesized microdata on demand. The keystone task in this vein is verification and validation (V&V) of synthesized data. Toward this end, we propose to first perform a survey of small-N studies and characterize each study by time, space, and topic coverage. This backbone of natively unified sets scopes which disparate large-N microdata sets may be synthetically unified. The sets synthesized from the identified large-N microdata are then necessarily amenable to be verified and validated at the record, aggregate, and results levels against the more inclusive small-N studies.

Time-shifting and topic synthesis operate on surveys collected within a particular country. Whereas one country’s data is not reliably portable even to similar nearby counties, sourcing data from uncovered countries of interest cannot be overcome synthetically.

We introduce the novel idea that computational social science models can also serve as synthesis artifacts. Models require synthesized data; but they also can produce novel syntheses. Specifically, we propose to use probabilistic graph models to draw from diverse datasets and generate synthetic microdata samples, instantiated in agent populations, which encapsulate nonlinear relationships inferred from the source data.

The basic process we used is shown in Figure 1.

### 4. Modeling

In modeling migration, there are two essential questions: what causes a household to migrate, and what criteria do they use to select their destination? We assume that, in making both of these decisions, households use information about their actual and relative circumstances (Hear, 2012). Unfortunately, the data on actual and relative circumstances prior to migration do not exist. To overcome this problem, we use Probabilistic Graphical Models as a means of estimating prior and comparative circumstances. Then, we construct decision trees that incorporate these estimates.

#### 4.1 Probabilistic Graph Model

The rise of cheap and abundant computational power has made the use of probabilistic graphical models (PGMs) feasible. PGMs – and, in particular, Bayesian Networks (BNs) – are attractive in general for three reasons (Koller & Freeman 2009).

![Figure 1 - Pipeline Process from Microdata through Simulation and Validation](image-url)
First, PGMs represent statistical relationships concisely. Specifying a full joint-probability table for many variables quickly becomes intractable owing to combinatorial explosion, even ignoring the issue of statistical significance. By inferring conditionally-independent relationships, PGMs factor the event space into a manageable subspace. At the same time, PGMs are transparent (in contrast to opaque models, like artificial neural networks). Consequently, an expert can interrogate proposed graphical structures for face validity; suggest alterations; and, importantly, impose constraints implied by established theory.

Second, the graphical structure allows for fast inference and sampling, both in general and conditional upon some known evidence. Sampling conditionally upon known evidence (i.e. sampling over the posterior distribution) is particularly useful as a means of joining disparate models.

Third, in the case of BNs, there is an existing and growing collection of open-source and proprietary tools for learning both the structure and parameters of the models. This learning may be purely data-driven, deriving the most likely models from extant data; or, it may be inferred while incorporating expert opinion.

These attributes are especially attractive for social science research, where control and prediction are subordinate to satisfactory explanation. Fidelity of proposed relationships matters, and a tool that is transparent, fast, and amenable to theory building and hypothesis testing is very useful. Recognizing the promise of PGMs for social science research, BNs are experimentally employed for this project in three distinct but related ways.

While IPUMS provides only cross-sectional data, certain survey years have variables indicating the respondent’s previous residence (sub-national regions or both sub-national regions and urban areas). This variable grants a toehold for imputing the most plausible explanation (MPE) for the respondent’s previous characteristics, casting it from cross-sectional to quasi-longitudinal. First, a collection of BNs is learned for each urban area represented in IPUMS, limited only to non-migrant respondents. Then, these responses are partially-translated backwards through time, adjusting deterministic variables when possible (e.g. subtracting five years from their age) and marking other variables as unknown. The known variables – those that are deterministic or immutable (e.g. sex) – are used as evidence when querying the learned, urban-level BNs. This procedure effectively asks: assuming the respondent is not exceptional and that municipalities change slowly (over many years), what values could we expect the missing values to take for migrants, in the statistical sense? Once imputed, a new set of BNs can be learned which infer the statistical structure of observed and imputed variables at the previous and present sites.

This same general procedure – sampling from an existing BN conditional upon some fixed evidence – can be used to join disparate datasets. Generally, when two different datasets share a set of common variables such as the respondent’s socioeconomic status, municipality, gender, age, family size, and number of children, a merged record can be synthesized. A record (sampled from the BN or drawn from survey data) from one dataset is taken as prior information; then, the BN inferred for the second data source is sampled, conditional upon the values of variables in the union of the two sets – the evidence. The result is a statistically plausible merger of two disparate data sources.

In addition to their use in exploratory data generation, the collection of PGMs is used to seed the agent-based model with survey data. The alternative – assuming a non-existent source of longitudinal and cross sectional survey data – would be to initialize the model statically. By using the PGMs, agent initialization is empirically constrained, yet stochastically determined. And, it also allows for redistribution, something typically prohibited with raw data.

<table>
<thead>
<tr>
<th>IPUMS</th>
<th>SEX</th>
<th>AGE</th>
<th>Has AUTO</th>
<th>HOME OWNERSHIP</th>
<th>HAS PHONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB-2000</td>
<td>S1</td>
<td>S2</td>
<td>S10G</td>
<td>S10C</td>
<td>S10F</td>
</tr>
<tr>
<td>LB-2001</td>
<td>S1</td>
<td>S2</td>
<td>S10G</td>
<td>S10C</td>
<td>S10F</td>
</tr>
<tr>
<td>LB-2002</td>
<td>S1</td>
<td>S2</td>
<td>S10G</td>
<td>S10C</td>
<td>S10F</td>
</tr>
<tr>
<td>LB-2003</td>
<td>S1</td>
<td>S2</td>
<td>S10G</td>
<td>S10C</td>
<td>S10F</td>
</tr>
<tr>
<td>LB-2004</td>
<td>S1</td>
<td>S2</td>
<td>S10G</td>
<td>S10C</td>
<td>S10F</td>
</tr>
<tr>
<td>LB-2005</td>
<td>S6</td>
<td>S7</td>
<td>S15H</td>
<td>S15C</td>
<td>S15F</td>
</tr>
</tbody>
</table>

Table 1. Mapping Join Variables to Source Variables

While survey data sets with a large sample size are exceptionally rare, small-N surveys that ask a broad set of theoretically interesting and useful questions are less rarely available. Since the synthesized data can be sampled easily and dynamically, these small surveys can be used for verification and validation of the amalgamated models. Recognizing that the statistical significance of small-N surveys is limited – especially when asking questions about the joint distribution – it is still useful to ask: how well does the synthesized model comport with the small set of observations? If they are statistically similar, this provides evidence of fidelity with the real world (see section on Verification & Validation). Absent demonstrable verisimilitude, it allows the researcher to ask what relationships might have been missing, which assists theory building.
4.2 Synthetic Respondents

In answering the modeling questions, the first task becomes: what did the respondents’ lives look like prior to migration? IPUMS is the largest extant source of harmonized microdata for Colombia. Yet, as mentioned previously, it is purely cross-sectional. However, the existence of columns giving the prior residence for individual respondents provides the aforementioned toehold for generating estimates of prior conditions. From this column, and using PGMs, the desired responses can be synthesized.

First, the set of respondent household heads who did not migrate in the five years prior to survey is collected and partitioned by municipality. As non-migrants, there is no geographic shift for this group. There is only a shift in time, affecting their age and possibly number of children. For example, if it was reported that they had a child less than five years of age at the time of survey, their number of children would need to be decremented when assessing their situation five years prior. It is certainly possible that their resources would have changed during this period. However, for the sake of imputation, it is assumed that their resources were static. Roughly, this translates into the assumption that during this period, their number of children would need to be decremented. When assessing their situation five years prior. It is certainly possible that their resources would have changed during this period. However, for the sake of imputation, it is assumed that their resources were static. Roughly, this translates into the assumption that for non-migrants, little changes in five years; but, since we know they were stationary in geographic space, treating their resources as fixed gives us a means of estimating resource levels by municipality for non-migrants. A set of PGMs—one for each municipality—is trained on the data in this partition.

Then, the set of respondents who did migrate in the five years prior to survey is collected. This group is then shifted back five years in time and placed in the municipality of their previous residence (as previously discussed). However, we cannot assume that their prior resources are static, as in the case of non-migrants. They have shifted geographically, and different municipalities have very different resource profiles. Instead, for each migrant, the geographically specific, non-migrant PGM is queried with six variables as the evidence keys. The first three are:

1. **AGE_GROUP** is a mapping from their recorded age to a categorical value: $[0, 20) \Rightarrow 0$, $[20, 30) \Rightarrow 1$, $[30, 40) \Rightarrow 2$, $[40, 50) \Rightarrow 3$, $[50, X) \Rightarrow 4$. This mapping was necessary to reduce the parameter space for the PGM over age. Age value is important, but if a query were conditioned by a value bounded by 0 and 100 in integer space, there would be too few observations for some events, even when backed by larger data sources. This variable changes, but by a deterministic rule with respect to time.
2. **IS_MALE** is a predicate with *true* signifying the respondent was male. This variable is predominantly constant with respect to time.
3. **HAS_AUTO** is a predicate with *true* signifying the respondent had at least one automobile. This variable was included as a proxy for resource wealth, but it is a portable asset so it is reasonable to assume it follows migratory respondents.

Querying the departure PGM with the evidence variables provided by the extant microdata yields a statistically probable, synthetic observation. That is, the evidence parameters are taken as fixed, and the PGM returns a set of observations for the non-evidence fields that is justifiable given the structure learned on the empirical data. Or, in the context of migration, it yields a set of resources and other variables that were expected for the migrant in the departure site, given their fixed parameters.

After the migrants were mapped to a set of synthetic records at the departure site, a new series of PGM were generated. This series learned on the concatenation of the synthetic migrant records and the observed non-migrant records in the departure municipality. This PGM is subsequently used to seed the agent-based simulation.

In addition to back-imputed PGMs developed from the IPUMS microdata, a series of PGMs were constructed for the Latinobarometer (LB) data. These data have a much smaller sample size, with roughly 600 household head observations per year between 2000 and 2005, inclusive. A PGM was constructed by learning the structure and parameters for each year, separately. These PGMs are used to update the respondent profiles on an annual basis, at a national level.

Ideally, a national-level PGM could be used to express prior probabilities, joined in a Bayesian fashion to produce an estimate for each municipality given the observations. This is necessary, because the geographic area covered -- 153 unique municipalities -- disperses the sample size significantly. With a prior model (an extension discussed in the Prior Models recommendation), the PGM could be specific to each municipality, as in the case of IPUMS. However, since the geographic coverage is neither exhaustive for Colombia, nor as extensive as IPUMS, additional imputation would be required.

To provide this imputation, the data could be geographically filled in using the IPUMS data to find similarities between municipalities. First, the set of municipalities would be partitioned into two sets: one set for those with overlapping coverage between IPUMS and LB (the candidate set) and another with only IPUMS. To fill the latter set, the most similar municipality from the overlap set would be identified; then, the PGM for this municipality would be transplanted.
Selecting the metric for similarity requires more empirical work, yet there is an obvious candidate. First, the candidate set would be filtered so that only those with similar levels of urbanism are retained. Then, from this reduced set, the municipality with the most similar proportions of migration would be selected. The assumption here is that municipalities experiencing similar migratory events were subject to similar processes. This is a strong assumption, but in absence of additional observations, it is a required step.

Although the national-level PGMs were less geographically specific, they should be capable of capturing patterns of victimization, and the relationships between the provided variables. Using the national-level PGMs, Latinobarometer was stochastically joined with the IPUMS to generate rich, synthetic responses. To join the two PGMs, a sample is first drawn from the IPUMS PGM in a specific municipality. Then, five of these variables are used as evidence parameters when querying the Latinobarometer PGM in a specific year: IS_MALE, AGE_GROUP, HAS_AUTO, OWNS_DWELLING, and HAS_PHONE. (See Table 1 for a mapping of LB source variables to the model's variables, by year.) OWNS_DWELLING is a predicate indicating home ownership by the household head (identified by SIDP as a predictive factor for victimization). HAS_PHONE indicates that the respondent claimed to own a phone, a critical variable for establishing respondent communications capability. Although the Latinobarometer PGM was national level only, the synthetic joined record benefits from the localization provided by the IPUMS PGM. Home ownership and telecommunication networks vary greatly by municipality, which is captured by the IPUMS PGM.

The result of this procedure is a set of tools capable of generating empirically plausible sets of synthetic responses that are geographically and temporally localized. These data are otherwise unavailable, with no surveys conducted with similar depth and breadth.

4.3 From Observed Patterns to Decision Trees

The set of IPUMS-based PGMs constructed as a means of temporally and geographically seeding and updating agents in the agent-based model were also used to interrogate the logic behind respondent migratory decisions outside the ABM. The set of non-migrant, raw IPUMS responses were taken as given. Additionally, a set was constructed for each respondent in IPUMS that did migrate. These migrants were moved back in time and space, and joined with resource profiles that were statistically probable for them. Then, the two sets were concatenated, representing a statistically probable survey for all respondents at their departure sites. This concatenation was then used to analyze two broadly identifiable decisions: the decision to leave in general, and the selection of a destination site.

The first question – succinctly, who migrates? – proved to be more difficult to answer. Seemingly, there are many idiosyncrasies in choosing to leave your residence, excepting expulsion. However, some general patterns became apparent. Curiously, the population of migrants appear better educated, wealthier, and more connected than the population of non-migrants, even after accounting for urbanism.

However, non-migrants are not necessarily the least wealthy group. When looking at non-migrants against all migrant types – migration due to violence, work needs, family needs, and health needs, and college – we find a remarkable similarity between migrants fleeing violence and non-migrants. There are several candidate explanations for this observation. First, the inference engine struggles with violent migration, because it is the statistically smallest sample of the group. Second, the inference engine struggles with violent migration, because the stochastic join variables failed to capture some essential discriminating factor. Finally, migrants who are moving for a job, college, health, and family are the socioeconomic elite. Everyone else is part of the lower to lower-middle class (henceforth, the disadvantaged), which should be the largest group with respect to socioeconomic inequality. The disadvantaged are more likely to be victims; but, violence and victimization has a strong spatial element. Therefore, the socioeconomic variables may not weigh heavily when looking at the population from 10,000 feet. It just means they are predisposed to being victims of violence, if violence arrives. Retaining only the set of migrants and partitioning by imputed cause of migration, more patterns become apparent.

In general, the decision to migrate appears to be stochastic. Observed characteristics predisposed some groups to migration, but an agent-centric decision structure remained elusive. However, the second question – of which destination is chosen – yielded more concrete answers. To set up the problem, each known migrant was backwards imputed to his or her origin site. Then, a set of seven candidate destination sites was collected for each respondent. These candidate sites were composed of the two largest nearby municipalities; the two smallest nearby municipalities; two random municipalities; and, the site actually chosen, as given by IPUMS. Here, nearby means within a few hops, where a hop of 1 spans contiguous municipalities. If the chosen site was included in the set of six candidate sites, an additional random site was included. The two largest sites were included given extant literature demonstrating the robustness of gravity models (Peeters, 2012) with population as an attractor. The smallest sites were
included as contrasts. And, the random locations were used to ensure sufficient variance.

For each destination site, the expected value for each variable was generated by sampling over the destination-site PGM. This expected value could be thought of as the agent’s expectation if they were to move to a considered site. That is, given their age, gender, resources as proxies by owning a car, literacy, education, and household structure, what resources (including employment) could they expect to have at the candidate destination? The expectation for each variable was transformed into a rank ordering between the individual respondent’s candidate expectations. For example, the municipality that represented the least likely probability of having a job for them would be ranked 1 and the most likely would be ranked 7. These ranks were computed with respect to each variable independently.

The destination site reported in the raw data would be flagged as their choice amongst the set of candidates. This synthesized choice data was then fed to the CART decision tree learning algorithm (Breiman, et al. 1984). Classification algorithms occasionally produce bad results when there are unbalanced classes; that is, if there is one class that dominates the other in terms of frequency of occurrence (Japkowicz, 2000). Here, there are six "NOT-CHOSEN" and one "CHOSEN" for every respondent, thus qualifying as unbalanced. To alleviate this problem, the "CHOSEN" record was paired with one random record from the respondents six other "NOT-CHOSEN" records. This balanced the classes for presentation to CART. However, the underlying ranks were still computed with respect to the full candidate set, so the rank comparison structure is retained.

The resulting decision tree identified patterns in migrant destination site selection, contingent upon the reason for their migration. Following the specification of the CART algorithm, the tree is constructed from root to leaf in a manner to maximize information gain at every step. Effectively, CART is a tree search algorithm; the decision nodes are selected if they provide the best reduction in entropy, from amongst the set of candidate tests. Or, said more simply, it picks the condition that best separates the classes, then moves on to the next level of the tree. This procedure explains why variables expected to be explanatory are sometimes absent. If there is co-linearity between variables, then the classifier will find little gains to be made once the first variable had been selected. For example, if HAS_ELECTRICITY was selected as a node, then HAS_PHONE may subsequently provide little additional explanatory power, as they are highly correlated.

For all migrant types, the population of the destination was identified as the first node in the decision tree. Effectively, this means that the destination population best partitioned the space of chosen and not chosen candidate sites. This is in agreement with the extant literature on gravity models in migration. The decision trees conditioned by migration type follow.

Figure 2 depicts the truncated decision tree for households reporting migration due to "Family" considerations. The condition is portrayed with a square box. If the condition is true, the left path is followed; if it is false, the right path is followed. The leaf nodes (ovals) show the number of IPUMS households falling into this category (n) and the odds of selecting this destination site.

To assist in theorizing, another set of decision trees was learned on the Latinobarometer data, classifying which respondents experienced victimization. Note, these trees did not require PGM generation or backwards imputation. They were learned on the raw survey data. When the CART algorithm was endowed with all the available variables, corruption was the best classifying node. This is reasonable – people who have experienced corruption seem more likely to have experienced violence. Ignoring corruption, the size of the town, access to drinking water, gender, age, and whether or not the respondent read the newspaper (information provenance), were the algorithmically selected nodes (For space considerations, this set of trees are not depicted). Perhaps, whether or not the respondent read the newspaper affected the perception of crime rates.

Another decision tree was limited to perceptual variables. Again, the size of the town was the most important factor, and reading a newspaper remained important. However, employment, perceptions of future economic conditions, and their perception of whether their children would be would live better
comprise the next layer of nodes. Seemingly, violence and the perception of violence affect prospective evaluations more than they do retrospective ones, even though the event of violence should be attached to the retrospective evaluations, in that they occurred already.

5. Simulation

In the previous section, we described our approaches to producing synthetic respondents, and their decision trees. The products of these approaches furnish the properties and cognition (respectively) of the agents and behaviors to be simulated.

5.1 Homeland Model

HomeLand, our agent-based simulation of migration and victimization in Colombia, is described in detail in Kennedy, et al. (2014) using the ODD+D approach (Muller et al., 2013). It was built in Java using the MASON agent-based modeling environment (Luke, et al. 2005). The nearly 5 million agents represent individual households of the entire population of Colombia and are located in 1 km² parcels based on the population density. Each household’s location is within a municipality within department (the equivalent of US counties and states) and initialized using population data. To support social communication among our agents, we extended MASON by building a message exchange layer (MEL). Each household considers whether to migrate and where they might migrate each step representing a year. The results of that decision process may or may not be communicated as a recommendation to the household’s social network made up of their immediate neighbors, more distant neighbors, and a few households selected at random from the entire country.

5.2 Experiments & Results

Our experimental design is 2 × 2, given in Table 2 with conditions marked with short names. We generate migration behavior under conditions in which household decisions are, or are not, informed by communications from other households, and decide where to move with or without using perceptual data. The social condition tallies recommendations from the household’s social network on whether or not to migrate. The perceptual condition uses annual Latinobarometer survey data to evaluate possible destinations and that data includes perceptions of relevant trends, such as whether households that move there think the economic conditions are better. Simulation output is reported in the NOEM input/output format, the schemas for which are given in the tables below (values available on request). As is, constituent values are produced at the household level, but household members could be retrieved reconstruct the full population.

5.3 Verification & Validation

Verification and validation is conducted here in the spirit of model clamping described above. Input to the simulation is verified by comparing synthesized respondent data to the source respondent data from multiple original sources. We perform this comparison of mortality and employment (from IPUMS) at the municipality level for the year 2005, which each set and the simulation have in common. Comparison of victimization between LB and the synthesized set is challenged by few observations by municipality in LB, and LB’s low statistical representativeness of the population’s demographics.

Output from the simulation (population counts, migration) over the course of the simulated period, 2000-2005, is validated at the department level against the 2005 endpoint captured by IPUMS. The final population counts by department over the five year simulated period for each of the four conditions are in reasonable agreement, excepting the occasional outlier as shown in Figure 3.

6. Discussion and Conclusions

The final population counts by department over the five year simulated period for each of the four conditions are in reasonable agreement, excepting the occasional outlier. Where the baseline and social conditions agree closely with IPUMS counts, the conditions social + perceptual, and perceptual tend to underestimate. Where the baseline and social conditions overestimate or underestimate IPUMS counts, the conditions social + perceptual, and perceptual tend to be closer to the IPUMS counts.

This pattern of compensation between model conditions that take perception into account with those that do not suggests the possibility that the particular cognitive scheme for an agent to employ may itself be selected according to personal or environmental characteristics. For example, if conditions are particularly fearful, stressful, or dreadful, perception may impact decision-making more than under conditions that are bad, but not so extreme.

<table>
<thead>
<tr>
<th>Table 2 - Experimental Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without communications</td>
</tr>
<tr>
<td>Excluding Perceptual Variables</td>
</tr>
<tr>
<td>Including Perceptual Variables</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3 - Data Set Agreement (Regression, R²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesized Set</td>
</tr>
<tr>
<td>IPUMS (mortality)</td>
</tr>
<tr>
<td>IPUMS (employment)</td>
</tr>
</tbody>
</table>
In this paper, we documented the simulated reproduction of migration behavior as collected by the Colombian census. Much of our effort was in the integrating data from small surveys, modeling and analyzing that data, and developing representations of the decision-making of the households’ internal migration. This was then used in a simulation and confirmed to match the available data.

While this project was primarily an effort to conduct advanced development for NOEM's migration and crime modules, two methodological capabilities applicable to both modules as well as the behavior module emerged along the way: microdata set synthesis to seed agents, and the production of decision trees from these synthesized sets to endow the agents with empirically-grounded models of decision making. We suggest these methods are useful in developing data-driven social simulation.

7. References


The authors wish to acknowledge the statistical office that provided the underlying data making this research possible: National Administrative Department of Statistics, Colombia.


Author Biographies

JOHN B. NELSON is a Ph.D. student at George Mason University. He is interested in computationally modeling belief systems, especially with respect to American voting.

WILLIAM G. KENNEDY, Ph.D., Captain (USN, Ret.) is a research assistant professor with the Department of Computational Social Science, part of the Krasnow Institute for Advanced Study, George Mason University.

ARIEL GREENBERG is a Project Manager in the Intelligent Systems Center of JHU/APL. His research spans the domains of computational biology and social science, including topics in psychophysiology and behavioral modeling. Ariel received degrees in Biology and in Electrical Engineering from University of Maryland, College Park.
Long-Term Dementia Care: Modeling the Decision Process

Emily S. Ihara\textsuperscript{a}  
William G. Kennedy\textsuperscript{b}  
Catherine J. Tompkins\textsuperscript{a}  
Michael E. Wolf-Branigin\textsuperscript{a}  

George Mason University  
\textsuperscript{a}Department of Social Work, \textsuperscript{b}Department of Computational Social Science  
4400 University Drive  
Fairfax, VA 22030  
703-993-2023, -9291, -2838, -4229  
{eihara, wkennedy, ctompkin, mwolfbra}@gmu.edu

Keywords: family caregiving, agent-based modeling, systems dynamic modeling, stress and coping theory, Alzheimer’s disease

\textbf{ABSTRACT:} Prior to a crisis situation, family members often state that they will not have a loved one placed in a long-term care facility (nursing home care); however, when the situation arises, what enables some family members to hold true to this statement and others unable to follow through? This paper explores the complex decision-making process that family members may go through while caring for a loved one with dementia. Decisions may fluctuate as the challenges resulting from behavioral changes characteristic of different stages of dementia (specifically Alzheimer’s disease) occur. This paper combines system dynamic modeling and agent-based modeling to represent a notional model of older adults with dementia and their associated caregivers. A caregiving stress and coping paradigm and current policy provisions are used to inform the decision-making process family members may experience while making the decision to become caregivers and maintain community-based caregiving responsibilities. Experimentation of different levels of relief showed that certain levels alleviate caregiver stress. Implications of these findings are discussed.

1. Caregiving for Individuals with Dementia

Alzheimer’s disease and other dementias are debilitating, progressive, and costly, affecting individuals, their families, and the long-term care system. Approximately 5.2 million people were diagnosed with Alzheimer’s disease in 2014 and projections are that these rates may nearly triple to 13.8 million (Alzheimer’s Association, 2014). Informal family caregivers provide the majority of care to frail older adults; this assistance is invaluable and fulfills an important role not only for persons with dementia, but for society as a whole (Robison, Shugrue, Fortinsky, & Gruman, 2014).

Dementia caregiving can be a frustrating and difficult experience depending on the symptoms of the individual with dementia and the environmental supports that are in place to assist the family caregiver. Although families try to keep their loved ones out of institutions as long as possible, the absence of relief for the dementia family caregiver may have deleterious outcomes for both the family caregiver and the individual with dementia. The family caregiver may experience high levels of stress, depression, and illness (Schulz, Boerner, Shear, Zhang, & Gitlin, 2006), leading to poorer quality of life for both the individual with dementia and family caregiver and possible early nursing home placement (Benjamin, Matthias, Kietzman, & Furman, 2008; Gaugler, Kane, Kane, & Newcomer, 2005; Yaffe et al., 2002).

1.1 Stages of Dementia

As Alzheimer’s disease and other dementias progress, behaviors and subsequent caregiving responsibilities change. In the early stage of the disease, an individual may experience mild cognitive difficulties, but is typically able to continue to perform activities of daily living (ADLs) and communicate. Caregiving at this stage is often more supportive, helping the individual cope with memory loss. During the middle stages, damage to the brain may affect a person’s behavior, ability to communicate, and ability to perform basic tasks. Common behaviors as the disease progresses may include wandering, repetitive behavior, physical and verbal outbursts, and sleep changes (Alzheimer’s
Association, n.d.). Caregiving at this stage involves more hands-on assistance with ADLs, such as dressing, bathing, eating, and grooming. Wandering behavior often creates a safety issue for those living in the community, and preventing wandering becomes a prime caregiving challenge. For those living alone, the individual may need to move in with relatives or to a residential care facility. Caregivers who are not able to supervise their loved ones all day must find a way to keep the individual safe, and may turn to options such as adult day health care or a personal companion. The middle stage of the disease typically lasts the longest and may have several crisis points as the level of independence decreases. During the later stages of the disease, an individual may have difficulty eating or swallowing, may need assistance with walking, may need extensive personal care, and may lose the ability to communicate with words. At this point, the needs of the individual may exceed the caregiver’s ability to provide the necessary care at home (Alzheimer’s Association, n.d.).

Studies have shown that behavioral issues rather than cognitive abilities are more highly correlated with caregiver burden and depression, especially behaviors such as aggression, agitation, and wandering at night (Gallicchio, Siddiqi, Langenberg, & Baumgarten, 2002; Gaugler et al., 2005; Gonyea, O’Connor, Carruth, & Boyle, 2005; Rinaldi et al., 2005). Appropriate interventions are necessary to alleviate caregiver burden and maintain individuals with dementia at the most appropriate level of care (Etters, Goodall, & Harrison, 2008).

1.2 Theoretical Framework: Stress and Coping

This study uses Lazarus and Folkman’s (1984) stress and coping paradigm to model the decision-making process for family caregiving of loved ones with dementia across the different stages. The caregiver stress and coping paradigm depicts the adaptational outcomes related to the stressors of caregiving based on the appraisal, coping responses, and social support of the individual caregiver. Stressors experienced by family caregivers of people with Alzheimer’s disease and other dementias include the specific stage of the disease (depicting the severity of cognitive impairment), behavioral problems such as wandering and aggressive behavior, and the inability to perform activities of daily living (Haley, Levine, Brown, & Bartolucci, 1987).

The caregiver’s appraisal of the level of stress he/she is experiencing, the ability to manage the stress appropriately, and the level of social support that is available may determine a caregiver’s decision to move a family member from community-based care to a long-term, institutional caregiving environment such as nursing home placement. In particular, crisis situations may create a sudden increase in stress that is beyond the caregiver’s ability to cope. Interventions that assist the caregiver and prevent inappropriate or unwanted nursing home placement may contribute to sustainable solutions that enhance the quality of life for the individual with dementia and the family caregiver.

There are many complexities that come into play as family members consider the need for increased care for frail family members. Ihara, Horio, and Tompkins (2012) conceptually grouped variables into two domains—motivation and capability—in their study of grandchildren opting to provide care for their grandparents. They defined capability as a family member’s discretionary time and proximity to the frail older family member and motivation as the desire and sense of obligation to provide care after considering the costs and benefits.

1.3 Policy Options

In our model, possible interventions include increasing options that will support family caregivers. Policy options such as increased respite care availability, tax incentives, work place policies, and adult day health services may support aging-in-place (Chen, 2014). Some of these options are currently available through laws such as the Family Medical Leave Act (P.L. 103-3), provisions under Title III, Part E of the Older Americans Act related to the National Family Caregiver Support Program (P.L. 109-365), and the Lifespan Respite Care Act (P.L. 109-442) (Ihara et al., 2012).

Unfortunately, home- and community-based services are often out of reach for near-poor older adults who may not qualify for publicly funded services. Provisions for long-term care under the 2010 Patient Protection and Affordable Care Act have provided several expansions of home- and community-based services (HCBS) under state Medicaid programs, including the Balancing Incentives Program, the Community First Choice state plan option, and the home health state plan option (O’Shaughnessy, 2013). These and other programs such as the Community Innovations for Aging in Place Program help promote aging in place (Greenfield, 2012), but the growing need
for services may not match the availability or ability of state and local communities to meet all of the demand.

In 2009, the National Alliance for Caregiving reported that more than half of caregivers who responded to a survey asking them to rate six potential policies or programs indicated that a $3,000 tax credit would be either their first or second choice. To test this policy option, Ihara et al. (2012) used an agent-based model to explore the likelihood that grandchildren would become a primary caregiver for a frail grandparent. They found that a targeted-policy scenario where high-income families do not get a tax credit, middle-income families receive a $3,000 tax credit, and low-income families receive a higher tax credit had better results for motivating grandchildren to become caregivers than the universal policy of a flat tax credit for all caregivers.

These various policy options potentially underlie the decision-making process of an older adult and his/her family regarding the best living situation including independent living, home-based supportive living, assisted living, or nursing home placement. Further, these options may not necessarily alleviate the burden for all families, pointing to the need to better understand what mix of services and support can enhance the decision for caregivers and care recipients.

2. Simulated Model

To focus on the decision-making involved with this topic, we use a mixed approach. Overall, this is an agent-based model (Gilbert, 2008) with the individual agents built on system dynamics models of their health and stressors. The model is implemented in NetLogo (Wilensky, 1999) and this description of our model is based on the approach described as an ODD (Grimm et al., 2010) and ODD+D (Müller et al., 2013). This paper is not a full description of the model, but focuses on the agents and their behavior.

Our notional model, named Carington, has 100 agents representing older adults and approximately 60 agents for their associated caregivers because approximately 40 of the older adults provide their own care. The caregivers may be family members (spouse, adult daughter, or other kin), professional caregivers, or institutions. Each step of the model represents a year. With each step, the general and mental health of the older adults may decline. If conditions change, the provider of the care may change from self to family, from family to a professional, or from a professional to an institution. Changes in conditions are based on the health of the older adult or the perceived stress of the caregiver. The older adult or the caregiver may also pass away. New older adults are added in each step to keep the population of older adults at 100 agents. The mix of care providers is driven by the health of the associated older adult.

2.1 Agents Representing Older Adults

Agents in the system representing older adults have variables for their age, general health, mental health, and who provides their care. The agents are initialized randomly, but are assigned behavioral characteristics that replicate the population statistics mean and standard deviation as appropriate for the simulated age of the agent. They are also initialized with different levels of physical and mental challenges consistent with the data. Many are initially their own care providers. Over time, their need for care due to their general health and level of dementia rises.

With each step of the model, their general and mental health conditions are changed probabilistically to match the changes in the population statistics reported by the Federal Interagency Forum on Aging Related Statistics (2012). As shown in Figure 1, general and mental health decline is not linear; the plot is based on 100,000 live births and is for the total population. Data is also available broken down by sex and race. We use the data for the total population in this model.

![Figure 1. Surviving Americans by Age for 100,000 Live Births](image)

We model the decline in general health and mortality using the data (shape) of the curve in Figure 1. To model the decline in mental health for our agents, we use data provided by the Centers for Disease Control and Prevention (CDC) on exhibiting signs of Alzheimer’s disease as the model for the general level of dementia in Carington.
Older agents needing care are paired with a caregiver agent. The status of an older adult needing care provides input to the decision-making concerning the source of the needed care.

2.2 Agents Representing Caregivers

Caregivers are also represented as agents in the system. Caregiver agents have characteristics describing their capabilities and motivation. Their motivation is described by their relationship to the older adult, level of difficulty associated with caregiving, and their own needs.

Caregivers are modeled as having a current level of caregiving capacity and a current caregiving load, which is increased by additional stressors (such as an increase in their frail older adult’s needs) by assistance as a systems dynamics model. At each step, an evaluation of the caregiver’s previous stress level increases due to changes in the status of the associated care receiver, other stressors, and support systems. We modeled a change in needed care as an additional stressor for one step (a modeled year). If the level of stress becomes too high, a decision is made to change the arrangement for the assistance the associated older adult needs. Changes in caregiving arrangements can include various coping mechanisms for the caregiver, including changing who provides the needed care.

3. Experiment and Results

The purpose of the experiment is to demonstrate that providing services and support can reduce the stress of caregivers, potentially helping them to continue to care for their loved ones at home for a longer period of time. To model the effects of this, we presumed that the relief would reduce the stress proportional to the amount of the time relief relative to the total time. The stress is caused by taking care of the older adult for M hours a day every day. We use less than 24 hours per day (18 hours per day) to account for time the older adult is asleep and the time the caregiver is asleep. The adult day care hours would then also be included in hours of relief for the caregiver. Although the caregiver may still be responsible for the older adult, we are looking for effective hours of relief for the care provider. Using N for the effective adult day care hours per week, we believe the stress would be reduced by the fraction \((7 \times 18 - N) / (7 \times 18)\). We ran our model with and without relief for the care providers. Relief was provided at different levels – 8, 16, 40, 70, 84, and 98 hours per week.

Experimentation with the model allowed us to examine whether the number of hours of relief per week averaged over one model step of a year, had an effect on caregiver stress. Table 1 presents the results of this experiment and shows the difference between the stress levels with and without the relief. As expected, there was no change in caregiver stress when zero hours of relief were provided. An increase of eight hours per week was also not statistically significant. However, increasing relief to 16 hours per week showed statistically significant differences in the average level of caregiver stress, with continued statistically significant results for higher levels of relief. While the average stress of the family caregivers declines significantly, the average number of caregivers is not consistently significant or insignificant. With our maintenance of 100 older adults needing care throughout the runs, their caregivers seem to also have been maintained.

Table 1. Model Runs and Statistical Significance

<table>
<thead>
<tr>
<th>Relief (hours/week)</th>
<th>Average No. Family Caregivers (SD)</th>
<th>Average Caregiver Stress level (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>34.2(5.15)</td>
<td>1.508(0.272)</td>
</tr>
<tr>
<td>8</td>
<td>33.5(4.80)**</td>
<td>1.478(0.267)</td>
</tr>
<tr>
<td>16</td>
<td>33.8(4.84)</td>
<td>1.452(0.262)**</td>
</tr>
<tr>
<td>40</td>
<td>33.7(5.00)</td>
<td>1.404(0.235)**</td>
</tr>
<tr>
<td>70</td>
<td>33.6(5.06)</td>
<td>1.268(0.183)**</td>
</tr>
<tr>
<td>84</td>
<td>33.9(4.91)</td>
<td>1.164(0.201)**</td>
</tr>
<tr>
<td>98</td>
<td>33.7(4.92)*</td>
<td>0.988(0.178)**</td>
</tr>
</tbody>
</table>

* indicates statistical significance (p<0.05)
** indicates statistical significance (p<0.01)

4. Discussion and Implications

Given the results of our experimentation, there are various ways that existing services could be applied to 16 hours of relief per week, including home health aides, adult day centers, and assisted living. The most feasible of these for a family caregiver would be adult day centers, which are a cost-effective way to provide specialized health and social support services for the individual with dementia and a form of respite for the caregiver. Typically, costs for adult day centers average $72 per day. Compared to the cost of a non-medical home health aide ($168 for an 8-hour day), $43,756 per year cost of assisted living or $83,230 to $92,977 per year for nursing home care (Alzheimer’s Association, 2014), adult day centers are a feasible
alternative for enhancing the quality of life for both care recipients and caregivers.

Beyond the cost savings for the long-term care system and the family, studies have shown that use of adult day centers have beneficial effects for individuals with dementia and their caregivers on the days the individual attended the adult day center. These benefits include fewer behavior problems, better sleep, and decreased caregiver stress, cortisol levels, and depression (Gaugler et al., 2003; Klein et al., 2014; Zarit et al., 2011; Zarit, Kim, Femia, Almeida, & Klein, 2014; Zarit, Stephens, Townsend, Greene, & Femia, 2003).

Further, a study of specialized dementia adult day services shows moderately successful results (Logsdon, Pike, Korte, & Goehring, 2014) and provides some evidence for further testing of the effectiveness of such programs to address the needs of a growing population of individuals and families affected by Alzheimer’s disease and other dementias. As policymakers and service providers continue to tackle the complex issue of dementia caregiving, specialized adult day services may be a feasible alternative that is currently out of reach for many families.

Given the nature and complexity of dementia caregiving, our future work will build on this model to incorporate other aspects of the decision-making process. We plan to include support networks in our Carlington model and interactions among caregivers and among older adults. We also plan to experiment with different combinations of policy options and relief (in the form of services and support) that may contribute to a decrease in the family caregiver’s stress level.

5. References


**Author Biographies**

**EMILY S. IHARA**, Ph.D., M.S.W. is an Associate Professor in the Department of Social Work at George Mason University. Her research foci include the social determinants of health inequities for underserved populations, interventions for older adults with dementia, and systems science methodologies.

**WILLIAM G. KENNEDY**, Ph.D., Captain (USN, Ret.) is a Research Assistant Professor with the Department of Computational Social Science, part of the Krasnow Institute for Advanced Study at George Mason University and studies cognition from individuals to societies.
CATHERINE J. TOMPKINS, Ph.D., M.S.W. is an Associate Professor in the Department of Social Work at George Mason University with research expertise in gerontology, dementia care, and caregiving.

MICHAEL E. WOLF-BRANIGIN, Ph.D., M.S.W. is an Associate Professor and Chair of the Department of Social Work at George Mason University. His research focuses on complex adaptive systems and their application to social work practice particularly in relation to the built environment.
An Emotion and Temperament Model for Cognitive Mobile Robots

Lyle N. Long
The Pennsylvania State University
233 Hammond Building
University Park, PA 16802
814-865-1172
http://www.personal.psu.edu/lnl
lnl@psu.edu

Troy D. Kelley
Human Research and Engineering Directorate
Army Research Laboratory
Aberdeen, MD
410-278-5869
troy.d.kelley6.civ@mail.mil

Eric S. Avery
Human Research and Engineering Directorate
Army Research Laboratory
Aberdeen, MD
410-278-2946
eric.savery.civ@mail.mil

Keywords:
Emotions, Temperament, Cognitive, Robotics

ABSTRACT: This paper describes a model of emotions and temperaments (or personalities), and how to implement it in cognitive mobile robots. Emotions such as fear, anger, sadness, happiness, disgust, and surprise can be modeled theoretically, and vary due to reinforcers such as rewards and punishments. The model incorporates exponential decay of the reinforcement effects, so without continual reinforcers the emotion will return to their steady-state values. It is shown that emotions and temperament are coupled through the theory. The constants used in the model of emotions are related to the temperament of the robot. The main five temperaments discussed include Extrovert/Introvert, Neurotic/Rational, Conscientious/Careless, Agreeable/Disagreeable, and Open/Reticent. The emotion and temperament engine (ETE) developed here has been incorporated into SS-RICS, which is a cognitive architecture developed at the Army Research Laboratory and tested in both mobile robots and in simulators.

1. Introduction

Mobile robots are not currently designed with temperaments or emotions, which is referred to as Affective Computing [Picard (2000)]. Temperament (or personality traits) and emotions are not the same thing. Temperaments are traits that an individual animal possesses that are innate and typically fixed for that animal’s life. Emotions vary continuously, sometimes on small time scales. In animals, temperament and emotion (and variations across groups) are as important to survival as cognition. They are crucial to the animal’s survival, and will make robots more effective also (LeDoux, 2000).

There are five main types of temperament in humans and other animals, often called the Big Five (Digman, 1990): Extrovert vs. Introvert, Neurotic vs. Rational, Conscientious vs. Careless, Agreeable vs. Disagreeable, and Open vs. Reticent

When we design and build autonomous robots we do not generally think of these varying across the group, but a heterogeneous mix of traits in a group will make the group more successful. In addition, unlike in biology where these traits are relatively fixed over the life of the organism, these could be varied in intelligent mobile robots.

While people do not completely agree on a complete list of emotions, Damasio (1994 and 2010) discusses six “universal” emotions: Fear, Anger, Sadness, Happiness, Disgust, and Surprise

Plutchik (2001) discusses the same six emotions, but also includes “trust” and “anticipation.” He also describes how there can be varying levels of each emotion in his emotion wheel. Ekman (1999) describes 15 basic emotions. The six (or eight) emotions are common across animals [Braithwaite et al (2013)] and cultures. The approach used herein could easily use more or fewer emotions. Damasio refers to emotions as automated programs for action that have been created through evolution. Emotions are related to reward, punishment, drives, and motivations. There are typically negative and positive emotions, and they are tied to reinforcers (rewards, punishments, lack of reward, and lack of punishments), see Gray (1990), Rolls (1990), and Rolls (2013).

While some investigators have studied affective computing in robots, there are very few studies which incorporate temperament and emotions into mobile robots. And the ones that do exist, do not properly distinguish temperament from emotions (e.g. Barteneva et al, 2007 and Canamero, 2005). Gray (1990) discusses the connection between emotions and cognition. Groups of mobile robots with a mix of personality and emotions will be more effective and have increased mission success. An interesting anecdote relates to the well-known robot soccer competition. One of the researchers remarked that the robots play in the same manner at the start of the game...
as at the end, whereas a human would play very differently in the last few minutes of the game, especially if they were losing. Another example is group behavior. In nature there are many examples of groups (ants, fish, rats, humans, etc.) that are very effective, and the groups usually include a wide variety of personality types.

This paper discusses how to incorporate emotions and temperament into cognitive architectures such as ACT/R, Soar, and SS-RICS. The emotions are basically state variables, and the robot will behave differently depending on which emotion it is experiencing. The temperaments, as described below, are fixed characteristics of the robot (although, unlike in animals, we could vary them in time).

2. Cognitive Software

The work described here uses the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS), although we have used the SOAR cognitive architecture (Laird, 2008) on mobile robots in the past (Hanford and Long, 2014).

SS-RICS is intended to be a theory of robotic cognition based on human cognition. Additionally, a thrust of SS-RICS has been on the integration of theories within the field of cognitive psychology - primarily theories of knowledge representation and organization. The field of knowledge representation in cognitive psychology has been embattled in a struggle to quantify knowledge structures as either symbolic or subsymbolic (Kelley, 2003). Symbolic knowledge is characterized as static, discrete, and conscious. Language is a symbolic representation of knowledge. Subsymbolic representations of knowledge has been characterized as dynamic, distributed, and unconscious. Typically, perceptual or motor skills are characterized as subsymbolic knowledge. Riding a bicycle can be characterized as subsymbolic knowledge. Within SS-RICS, these two representations of knowledge are not mutually exclusive, but instead, lie on either ends of a cognitive continuum (Kelley, 2003). SS-RICS is a hybrid cognitive system that allows for a continuum of knowledge that includes both symbolic as well as subsymbolic constructs. It is believed that this integrated approach is the best way to represent the complete spectrum of cognition.

The Army Research Laboratory's (ARL) Human Research and Engineering Directorate (HRED) developed the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS) beginning in 2006 (Kelley, 2006). SS-RICS is a rule based production system, robotics control system inspired by the cognitive architecture, the Adaptive Character of Thought - Rational (ACT-R) (Anderson and Lebiere, 1998). Development of the system has leveraged heavily from biological and human capabilities for navigation, action and selection, and memory decay.

Long and Kelley (2010), described how robots could become conscious by building on current approaches to cognitive modeling, emotions and temperament will help in this endeavor. One of the most recent advances in SS-RICS has been the incorporation of a memory system that mimics human-type dreams to consolidate memories (Kelley, 2014).

SS-RICS is organized into three layers all utilizing a basic unit of knowledge, or working memory element, called facts. Facts are composed of a tuple: which is name, type, and slots. For example, a fact might contain information such as: (Location234 color value=red). In this example, Location234 is the name of the fact, the type of the fact is "color" and a slot, which has the name of "value" is defined as "red".

One layer serves as the Long Term Memory (LTM), and in practical terms, it is a relational database or a semantic network. As of this writing we are using a variation of ConceptNet (Liu and Singh, 2004) and have made some progress integrating ConceptNet into SS-RICS. ConceptNet allows SS-RICS to relate similar objects to each other (i.e. dogs and cats are similar) and allows SS-RICS to query the database for information about objects. For example, we can query ConceptNet and find that “Sweet” is a “PropertyOf” “Apple”.

The second layer within SS-RICS is the production system, which serves as the executor and Working Memory (WM) component of the architecture. The production system executes goals by matching rules to facts within working memory (not LTM). We are currently working on processes which pull information from LTM into WM by using spreading activation (Anderson and Lebiere, 1998). For example, if the robot defined a goal of going through a door; the robot would pull goals and facts from LTM associated with going through a door, as well as other goals, for going down halls and maneuvering in rooms. Note that this requires goals and rules to have activation to be stored as memories; this is different from ACT-R which doesn’t assign activation values to productions.

The third layer is the sub-symbolic layer. The sub-symbolic layer serves as the perceptual component of SS-RICS. The key aspect of the perceptual layer is that all of the sub-symbolic processors are running in parallel and can be chained together for serial or hierarchical operations. For example, a processor that receives data from a range sensor generates memories representing geometric line segments which it then passes to a line processor to be merged into a higher level line memory. For resource allocation adjustments.
processors may also be turned on and off at runtime as necessary. This makes SS-RICS relatively scalable in terms of real time processing. The sub-symbolic layer provides information to the production system in a continuous fashion represented as facts. As SS-RICS is developed, newer sub-symbolic systems can be added or removed as necessary. We have tried to make this component as modular as possible.

SS-RICS is currently designed to interact with a robotic platform via an interface which defines the required operations and properties needed for SS-RICS to operate the robotic asset and interact with its sensors. This interface allows SS-RICS to communicate with any robotic system which has an implementation that fulfills the interface. We currently have implementations of the interface for Mobile Robots Pioneer platform of robots, the SRV-1 robot, the iRobot PackBot and Clearpath’s Husky A200. SS-RICS was designed and developed using both managed and unmanaged C++ using Microsoft Visual studio 2003, 2008 and 2010.

A flowchart of the system being used here is shown in Figure 1. The emotion engine was added to SS-RICS as a subism, and the emotions and the currently most active emotion are available as state variables.

### 3. Emotion and Temperament Model

The survival of animals (including humans) depends on emotions and temperament. Robots with emotions and temperaments will also allow better interactions with humans. In addition, teams of animals (including humans) are more effective when the groups have a mix of temperaments. This has been shown true for robots (Eskridge et al, 2014), cockroaches (Planas-Sitja et al), fish (Mittelbach et al, 2012), spiders (Pruitt and Keiser, 2014), humans (Pieterson et al, 2006; Moynihan and Peterson, 2004), sheep (Michelena et al, 2009), and other animals. Also, Eskridge and Schlupp (2014) state:

*The combination of different personalities within a group and the associated roles assumed by different members have been found to improve the overall success of the group (Couzin et al., 2005; Dyer et al., 2009; Modlmeier and Foitzik, 2011; Modlmeier et al., 2012). Studies have shown that these personality differences can be stable and maintained over time (Dal et al., 2004; Oosten et al., 2010).”*

The model used herein builds upon the model for happiness by Rutledge et al (2014): \[
\text{Happiness}(t) = w_o + \sum_{j=1}^{f} \gamma_t^{(i-j)} \left( w_1 CR_j + w_2 EV_j+w_3 RPE_j \right) 
\]

where Happiness ranges from 0 to 100 and:

- \( CR \) = Certain Reward (e.g. 10)
- \( EV \) = Expected Value (e.g. 10)
- \( RPE \) = Reward Prediction Error (e.g. 10)
- \( w_o \) = Steady state value of happiness (e.g. 10)
- \( w_1 \) = Magnitude of change (e.g. 0.52)
- \( w_2 \) = Magnitude of change (e.g. 0.35)
- \( w_3 \) = Magnitude of change (e.g. 0.80)
- \( \gamma \) = Rate of decay (e.g. 0.72)

The model has exponential decay given by \( \gamma \). If the subject chose a certain reward (CR), then EV and RPE were zero. If the subject chose a gamble, then CR was zero. They also state:

*“Conscious emotional feelings, such as momentary happiness, are core to the ebb and flow of human mental experience. Our computational model suggests momentary happiness is a state that reflects not how well things are going but instead whether things are going better than expected.”*

The model used here is guided by the above model, but modified to fit the task of incorporating emotions and temperament into cognitive mobile robots. This task is not gambling oriented, but instead involves the robot experiencing rewards or lack of rewards as it goes about its work.

The model for emotion developed here is:

\[
\text{Emotion}(t) = w_o + \sum_{j=1}^{f} \gamma_t^{(i-j)} \left( w_1 R_{ij}^+ + w_2 R_{ij}^- \right)
\]

where there are now six emotions (Fear, Anger, Sadness, Happiness, Disgust, Surprise), each denoted by the subscript \( i \), and there is exponential decay of rewards also. The emotions change with time. The subscript \( j \) denotes a time instance. \( R^+ \) and \( R^- \) denote positive and negative rewards, respectively. So emotion is a vector of length six, but could easily be smaller or larger to accommodate different sets of emotions. Each emotion varies from 0 to 100, which is analogous to Plutchik’s (2001) color wheel.

To determine the current emotional state of the robot, the emotion with the maximum value is chosen, i.e. a winner take all approach as used in Breazeal and Brooks (2003).

One of the very interesting aspects of this model is that it can also model temperament (i.e. personalities) through the constant terms. Temperaments do not vary in time. A temperament matrix \( T_{ij} \) can be defined as:
Few previous papers have so clearly delimited the difference between modeling emotions and modeling temperament. For animals temperaments are fixed in time [Mendl, (2010)], but for robots we would be able to easily change the robots temperament (or personality) by just changing this matrix. It should be possible to model the big five temperaments presented earlier (Extrovert/Introvert, Neurotic/Rational, Conscientious/Careless, Agreeable/Disagreeable, and Open/Reticent) using the above matrix.

The simplest form of the temperament matrix would have all the rows the same (all the emotions vary in the same way), i.e.

\[ T_{ij} = \begin{bmatrix} w_{01} & w_{11} & w_{21} & \gamma_1 \\ w_{02} & w_{12} & w_{22} & \gamma_2 \\ w_{03} & w_{13} & w_{23} & \gamma_3 \\ w_{04} & w_{14} & w_{24} & \gamma_4 \\ w_{05} & w_{15} & w_{25} & \gamma_5 \\ w_{06} & w_{16} & w_{26} & \gamma_6 \end{bmatrix} \]

where for simplicity representative values from Rutledge et al (2014) have been used. As shown above the rows of this matrix relate to: fear, anger, sadness, happiness, disgust, and surprise; respectively. All 24 values in the temperament matrix can be varied however. For example, if we wanted a robot that was usually angry, did not remain happy very long, and was easily surprised, we might use:

\[ T_{ij} = \begin{bmatrix} 50 & 0.52 & 0.35 & 0.72 \\ 50 & 0.52 & 0.35 & 0.72 \\ 50 & 0.52 & 0.35 & 0.72 \\ 50 & 0.52 & 0.35 & 0.72 \\ 50 & 0.52 & 0.35 & 0.72 \\ 50 & 0.52 & 0.35 & 0.72 \end{bmatrix} \]

The complex relations between the temperament matrix and the robot behavior still need to be investigated in more detail.
As mentioned above, SS-RICS is capable of modeling symbolic and sub-symbolic features. The Emotion and Temperament Engine (ETE) developed here is a sub-symbolic program (a “subsim”) in SS-RICS, and can be called by the production system.

4. Results

Three simulations have been run in SS-RICS using different temperament matrices. The matrices are shown in Figure 2. The first case has a temperament that is on average angry. The second one is typically happy, and the third case is fearful on average. These tests were run on a simple maze shown in Figure 3. All the robots started at the same point and ended when it reached the UltimateGoal, as shown in Figure 3. Within the maze there are items the robot might perceive (food, danger, and gruesome items). The food makes the robot happy and it tends to linger in those areas. When the robot first sees danger it is surprised, and then it becomes fearful. It will also try to avoid danger by backing up and turning away from it. The gruesome items will tend to make it disgusted. In tests 1 and 3 the robot only finds food once, but in test 2 the robot sees food twice. For these tests, happiness and sadness are considered related. So when happiness increases, sadness decreases. Also, in test 1 the robot becomes afraid twice since it sees danger twice. The three graphs in Figure 4 show the paths taken by the robot in the three different tests cases.

The tests performed here were simple proof-of-concept tests, more complex tests are being conducted now. The three tests conducted were for three different temperaments (or personalities). The results show that each temperament required a different number of time steps to complete the mission (Angry: 343 steps, Happy: 507 steps, and Fearful: 531 steps). When the robot is happy or fearful, it’s progress towards the goal will be reduced. So it does make sense that the angry robot will reach the goal faster.

5. Conclusions

We have presented a model for both emotions and temperament that can be incorporated into cognitive mobile robots. The model used was inspired by a recent model of happiness [Rutledge et al, 2014], which includes a decay factor. With this model, robots can have personalities (temperaments) and emotions, and the personalities and emotions are coupled, as they should be. Emotions and personalities are quite different features of animals (including humans) but they are coupled. Emotions vary with time and temperament is essentially fixed. A novel feature of this model is the “temperament matrix,” which allows the personality and emotions of the robot to be summarized with only 24 constants (when using six emotions). More (or fewer) emotions could be modeled easily as well. This model has been integrated into SS-RICS and tested in the SS-RICS robot simulator. Additional tests will be performed using AmigoBot and Pioneer robots. In addition, we will be testing teams of robots with different temperaments.

6. Acknowledgments

This work was funded in part by the U. S. Army Research Laboratory (Contract SSP TCN 14039). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.
Figure 3. Maze used for tests.

Test 1 (Angry)  Test 2 (Happy)  Test 3 (Fearful)

Figure 4. Paths of the robot in the maze for the three tests cases.
7. References


Author Biographies

LYLE N. LONG received a B.M.E. degree from the University of Minnesota, an M.S. degree from Stanford University, and a D.Sc. degree from George Washington University. He is a Distinguished Professor of aerospace engineering, computational science, neuroscience, and mathematics at The Pennsylvania State University. He is the Founder and Director of the Graduate Minor Program in Computational Science. He was the Founding Editor-in-Chief of the AIAA Journal of Aerospace Information Systems. From 2007 to 2008, he was a Moore Distinguished Scholar with the California Institute of Technology. He was previously with Lockheed Aircraft Company and Thinking Machines Corporation. The work described herein was performed while he was on sabbatical at the Army Research Laboratory at Aberdeen, MD. He has authored more than 250 journal and conference papers. His current research interests include neural networks, software engineering, cognitive robotics, high performance computing, and unmanned vehicles. Dr. Long received the Penn State Engineering Society Outstanding Research Award in 1996, the 1993 IEEE Computer Society Gordon Bell Prize for achieving highest performance on a parallel computer, and the Lockheed Aeronautical Systems Company Award for excellence in research and development. He is a Fellow of the American Physical Society and the American Institute of Aeronautics and Astronautics.

TROY D. KELLEY is head of the Cognitive Robotics Team and is a Research Psychologist at U.S. Army Research Laboratory Aberdeen Proving Ground, MD. His primary research interests include visual cognition, object recognition, cognitive robotics and computational models of cognition. He has developed the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS) which is a cognitively inspired robotics control architecture. He has authored or co-authored over twenty-five technical publications, proceedings, and journal articles. He was also served as the co-chair of the Behavior Representation in Modeling and Simulation (BRIMS) conference from 2005 to 2007.

ERIC S. AVERY is a Computer Scientist for the U.S. Army Research Laboratory, Aberdeen Proving Ground, Aberdeen, MD. His research interests include cognitive robotics, computer image processing, human-computer interaction, and application development. Eric received his BA in computer science from Lycoming College, MS in computer science from Towson University and is currently pursuing his D.Sc. in Information Technology at Towson University.
II.

POSTER PRESENTATIONS
Patterns of Life in the Foreground and Background: Practical Approaches to Enhancing Simulation-Based Interaction Skills Training

Robert Hubal, Jeremiah Folsom-Kovarik, Angela Woods, Randy Jones, Jon Carbone
Soar Technology, Inc.
3600 Green Ct., Suite 600
Ann Arbor, MI 48105
734-327-8000
{robert.hubal, angela.woods, jeremiah.folsom-kovarik, rjones, jon.carbone}@soartech.com

Keywords:
Cultural variation, intelligent agents, patterns of life, training simulations

ABSTRACT: Interaction skills training, using simulated environments, demands proper behaviors not only from the learner but also from the agents (here, central and background characters) the learner observes or contacts. In several projects we are developing scalable, flexible, and ultimately easily-authored methods to portray natural activity, influenced by sociocultural context, within situations designed for interaction skills training. To meet differing use cases, we are interested in some cases in a high-level perspective of agent activity (e.g., realistic crowd movements) and in other cases in a ground-level view of individual agents with social networks who demonstrate patterns of life. To do so, we use different tools to achieve agent realism, from fuzzy state models to simple behavior algorithms to more complex cognitive reasoning. This paper describes cultural modeling knowledge structures and methods appropriate to a selection of use cases. Recommendations are provided for incorporating realistic sociocultural material into skills training.

1. Introduction

An ability to manage face-to-face social interactions is critical to achieving successful outcomes such as increased flow of actionable information, de-escalation of conflict, correction of errors and misperceptions, increased mutual perceptions of trust and respect, and enhanced cooperation. Understanding—or misunderstanding—of social and cultural influences on these interactions thus has consequences. Training on social interaction skills can be important in fields as different as clinical practice in medicine, community policing in law enforcement, and civil affairs in the military.

Traditional interaction skills training commonly relies on relatively passive videos of common situations, or on paper-based or live-action role plays where hired actors or other learners play the roles of supposed social partners in varying situations. Offshoots of these approaches include tactical decision games (Schmitt, 1994), cultural assimilation (Fiedler et al., 1970), interactive video (Roy et al., 2006), and authentic social interactions (Szulborski, 2005). These learning experiences have some drawbacks (Hubal & Frank, 2001), including limited variability in situations encountered and expense. Further, successful training requires sufficient fidelity of environments and realistic-enough behavior from the learners’ perspective to assure transfer of training. One approach to managing these limitations is to use simulated environments to train. Though not a solution that satisfies all criteria, “virtual role plays” have some advantages including potential variability of situations portrayed and lower distribution costs (Hubal, 2008).

Of particular interest here is the fidelity of the behavior of synthetic characters acting in virtual role plays. Even more specifically, the focus is not only on the central characters with whom the learner interacts, but also on background characters who populate the virtual worlds in which interaction occurs. An important consideration of learning about conducting successful interactions involves presentation of an environment that is reflective of the one in which the learner will encounter. The point is to strive for realism in characters’ behaviors because an important component of learning is to identify important cues or signals and distinguish them from background “noise”. Interaction skills training, then, should take place in appropriately diverse worlds, rendering a clutter of different ages, genders, personalities, ethnicities, cultures, and accepted practices (Endrass et al., 2010; Johnson, 2010; Kim et al., 2009; Taylor & Sims, 2009). The challenge is how to adequately model characters at different scales so as to train learners to recognize not only culturally appropriate population behaviors but also identify what is normal and what is anomalous.

In a series of projects, the authors and colleagues have integrated and are integrating detailed models of cultural daily activities and patterns of life into simulations of synthetic characters inhabiting virtual environments. The design of realistic cultural models introduces hard problems. We have identified and developed (overlapping) prototypical solutions for three such problems:
Background at Scale. There is a need for simulation of many (thousands) of characters with intelligent profiles, that is, each character having a kind of life story that drives its behavior.

Variation. There is a need for variability in characters’ behaviors based on facets underlying their intelligence such as culture, and a demand to avoid predetermined, scripted actions.

Anomaly and Normalcy. Central characters should have additional intelligence (e.g., to effect specific actions, as needed for training on anomalous activities), yet these characters must be able to blend into the general population, requiring not only models of their own behaviors but recognition of others’ behaviors.

In essence, our approach is the interleaved representation of three classes of synthetic character patterns. First is a fuzzy state machine (FuSM) for low level “clutter” presentation of up to thousands of individuals. Second is algorithmic control over basic movements of background characters using sociocultural parameters to make the characters more realistic for training in specific use cases. Third is the use of sophisticated cognitive agents for emulating high-value individuals, that is, central characters that intermingle seamlessly with clutter (FM 3-60, 2010).

As will be discussed in examples, this work includes interaction at different scales. In some projects the focus is on gross movement, essentially culturally realistic group behavior observable from a thousand-foot perspective. In others the focus is on continuous assessment of a scene for expected and anomalous items and events as a learner moves through the world. In others still the learning includes ground-level interaction with individual characters.

2. Current Approaches and Limitations

Many researchers and developers have investigated techniques to incorporate cultural conditions into interactive characters (Allbeck & Badler, 2004; Huang et al., 2009; Kim et al., 2009; Nazir, Aylett, & Cawsey, 2008). Yet the need to achieve variation without hard-coding hundreds of examples demands a means to take high-level inputs and create a variety of scenarios that fit the patterns. When possible, population models or urban crowd movement models may be used. Such models are increasingly realistic, but involve limitations for both central and clutter characters that are relevant to the three problem areas just described.

Relating to the problem of scale, researchers have studied and impressively portrayed how crowds move (Braun et al., 2003; Kim et al., 2012; Pelechano et al., 2005), but where the individuals have come from and where they are going to does not typically matter; they lack underlying goals. Hence the crowds are not moving purposefully based on facets such as culture, neither during normal routine nor in response to chaotic events (Shendarkar et al., 2008; Zheng, Zhong, & Liu, 2009).

For the problem of variation, while researchers have investigated culture-specific characteristics such as pedestrian movement (Fridman & Kaminka, 2010), political identity (Lustick, 2000), and adversarial intent (Loscos, Marchal, & Meyer, 2003; Silverman et al., 2004), they have not fully considered a myriad of alternative factors that can influence behavior. For instance, in religious Islamic cultures, crowd movement potentially changes five times per day when men are instructed to pray. Meanwhile, commercial activity, weather, special events, even tendencies of denizens toward walking in streets versus on sidewalks can influence movement by affecting what paths are available to individuals, where individuals are tending to go, and how many individuals are out and about. Similarly, just as pedestrian activity is culture-dependent, so is vehicular activity. Aside from obvious cross-cultural differences such as traffic density (e.g., cars per person) or rights of way (e.g., drive on the left vs. right) there are culture-specific influences such as types of vehicles on the road (to include size of cars and trucks, and presence of motorbikes, bicycles, rickshaws, and other non-motorized vehicles), general hurriedness, respect for the law, and time-of-day or day-of-week factors (rush hour characteristics, Sabbath day restrictions) (Hood & Diaz, 2003; Zaidel, 1992).

Current models of pedestrian movement and vehicular traffic do not take generally these cultural influences into account.

Last, regarding anomaly and normalcy, modelers have explored how characters are affected by elements of the situation (Hoey & Schröder, 2015; Prendinger & Ishizuka, 2001), and these approaches can be used to support variation in social behaviors generated by the FuSM and cognitive models. But they have not always accounted for how key individuals within crowds can affect others’ behavior and change their own behavior based on crowd characteristics. For instance, population models have not yet taken full advantage of findings derived from network activity such as the modeling of the emergence and spread of infectious disease through a community. Cultural models that take into account network phenomena have the ability to influence how crowds behave in response to central characters, and how central characters’ behaviors can be influenced by other simulation agents. Similarly, few simulated populations are synthesized using census or geographic data to model households and individuals and account for schools, workplaces, restaurants, and other daily locations (Wheaton et al., 2009), but these data promise increased cultural realism.
3. Culturally Aware, POL-Savvy Synthetic Characters

An emerging discipline in intelligence analysis is the recognition that pattern of life (POL) anomalies are an excellent mechanism to identify potential threats (Schatz et al., 2012). Those who live in an area learn to recognize POL anomalies and react accordingly. For example, a quiet marketplace at a day and time when it is expected to be busy is an indicator to the local populace of potential danger, or at least the need for caution (a “time to watch out”; Batty, 2007). However, recognition of, and vigilance to, patterns of life and revealing deviations are important not only to threat detection but also to effective interactions, because every interaction is situated in a context that provides meaning and information (i.e., social affordances; Zebrowitz, Bronstad, & Montepare, 2011).

Perhaps the most complex behavior comes from those individuals who blend in with normal patterns for most of their activities, but who as central characters to the scene step out as required to execute some action, gesture, or dialog that is important to training. It is not only population-level pattern deviations from the norm (e.g., the empty marketplace on market day) that is of interest, but also individual-level behavior. The complexity of getting synthetic agents to behave both unobtrusively and anomalously depending on any current context suggests that pattern-deviating simulated agents must be “aware” of their surroundings—blending in with the realistic crowd, in the case of a pedestrian, or with other traffic, in the case of a vehicle.

Across several projects, our team has developed reasoning structures to improve models of the general population (and of vehicular clutter). For example, in an effort (Hubal, 2014) intended to model cultural daily activity and patterns of life in a constructive simulation, the output is a population of up to some ten thousand citizens, each built around a relatively simple FuSM. The existing models for this ‘clutter’ generate the logic to support dynamic visualization. We have extended the models by identifying parameters that control clutter agents’ behavior that reflect differences between cultures. The approach scales down; even for applications where there is a need for only tens of background characters rather than thousands, hence the need for higher detail, we have developed a systematic specification of characters’ social lives.

POL behavior specifications combine to drive an individual agent (e.g., a background pedestrian or vehicle) through a realistic yet adaptable routine. Behavior specification for an agent consists of a linear schedule of goals. For each time block, the schedule specifies activity type, location, and additional parameters to support getting to the location and engaging in the activity. We use a FuSM to efficiently plan route details and generate appropriate messages for the visualization engine to render the background agent’s behaviors. The “fuzziness” in the fuzzy state machine refers to the manner in which agents’ actions are not completely specified. For example, one schedule might tell a synthetic character to wake up at home around a certain time, travel to work at a certain place, move to a religious site and carry out prayers, and finally return home within a time range. These behaviors are mainly planned before each scenario begins, however, the system does contain repair mechanisms that let synthetic characters react to unplanned or unexpected inputs. For instance, when one character wants to meet with another, or they chance upon each other during routine activities, they may both rearrange their schedules to accommodate the meeting. Currently we are using a probability-based decision whether or not to change the schedule, and if it is changed the schedule is just shifted for some culturally-appropriate time to indicate the event taking place. The next step will be to make schedule change decisions culturally relevant to the scenario as a whole. Either way, these repairs driven by FuSMs represent an efficient approach that lets behaviors incorporate context-appropriate reactions but remain tractable in very large populations.

A second approach is the modeling of some dozen high-value individuals using a rule-based cognitive architecture (Soar; Laird, 2012) and fitting them within the crowd. It is these ‘central’ characters who are designed to understand patterns of life at the population level and reason about behaviors to fit in appropriately—or disguise their deviations from such patterns. Two such examples of training simulations that model cultural behavior are the Cultural Cognitive Architecture (Taylor et al., 2007) and high fidelity characters for small-unit training in game environments (Stensrud et al., 2012).

For both types of agents—clutter and central—the precise scheduling of their actions is determined dynamically at runtime. In this design, central characters reason about the world, such as determining when and where to engage in interactive behaviors with the learner and how to best commingle with clutter agents. Meanwhile, there are two types of formats for background characters. The high-level representations for the thousands of crowd-forming clutter agents specify probability distribution functions for various types of demographics, as well as partially instantiated schedules for clusters of clutter agents. In contrast, when there are tens of background characters who must be more socially adept (e.g., to be portrayed in a first-person environment rather than viewed from a thousand-foot perspective) there are more detailed parameter values associated with them.
4. Parameters for Cultural Awareness

Culturally aware agents are those that exhibit behaviors that reflect locally meaningful customs or that change based on local preferences or demands. Making them so requires identifying culturally relevant daily activities such as shopping, driving, dining, use of sidewalks, use of cellphones, setting up meetings, and religious observance; defining influences of weather, air quality, time of day, religious holidays, or a city’s age or makeup on population-level activities; and modeling street-level interactions such as greetings and vehicular configurations that influence larger crowd behaviors (Hubal, 2014). For implementation, we have added a set of parameters to the schedule representation, allowing values within the schedules to be replaced by parameters. This method improves the ability to write reusable schedules that can be instantiated to different parameter settings.

We have begun to tap several sources to improve synthetic characters behavior models. One resource is a series of country-specific videoclips that show normal urban behavior—culturally-related crowd phenomena such as the passing side, family formations, and proxemics (Fridman & Kaminka, 2010). Other sources are publicly available movies and YouTube videos, as well as scientific literature that addresses cultural or societal differences in movement (e.g., Hershey & McKeown, 2012; Kaup et al., 2008; Mateo-Babiano & Ieda, 2007).

Table 1 shows a list of representative parameters that are applicable to the FuSM, behavior algorithms, and cognitive models, as they can influence characters’ behavior whether viewed from a high-level or ground-level perspective. To control agent behaviors, we employ rules that reference parameter values and guide the flow of activity. Individual-level behaviors are influenced, as well as how individuals’ behaviors affect group behaviors. Thus in one city a greeting between two persons may normally occur on the sidewalk and cause other people to move around the participants (unless and until they move to the side) but not step off the sidewalk, whereas in another city a greeting can occur anywhere in the street and not affect pedestrian flow in the same way. To handle this cultural variation we implemented a “conversation location” parameter. Similarly, in one city a preponderance of commercial activity might take place in open-air markets where a sudden rainstorm could cause a general clearing of shoppers, whereas in another city most shoppers may enter covered stores. This variation suggests a “marketplace exposure” parameter. We have found that a dozen or so such parameters, while not accounting for all possible cultural characteristics, still considerably change characters’ activity patterns.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air quality</td>
<td>Poor, acceptable</td>
</tr>
<tr>
<td>Conversation location</td>
<td>Anywhere, move-to-side</td>
</tr>
<tr>
<td>Day/night activity</td>
<td>Difference-in-amount-and-kind, steady</td>
</tr>
<tr>
<td>Family life</td>
<td>Cross-generational-and-extended, home-centered, non-centralized</td>
</tr>
<tr>
<td>Marketplace exposure</td>
<td>Outdoor, indoor</td>
</tr>
<tr>
<td>Passing</td>
<td>On-left, on-right</td>
</tr>
<tr>
<td>Pedestrian, vehicular speed</td>
<td>Fast-moving, moderate, slow-moving</td>
</tr>
<tr>
<td>Personality/engagement</td>
<td>Busibody, normal, reserved</td>
</tr>
<tr>
<td>Religious observance</td>
<td>Traditional, modern, secular</td>
</tr>
<tr>
<td>Street crossing</td>
<td>At-designated-crosswalks, between-sidewalks, walk-in-street</td>
</tr>
<tr>
<td>Vehicular traffic</td>
<td>Congested, moderate, light</td>
</tr>
<tr>
<td>Vehicular type</td>
<td>Animal-pulled, man-pulled, small-motorized, large-motorized</td>
</tr>
</tbody>
</table>

4.1 POL and Cultural Awareness within Clutter Agent Models

Additional, statistically-relevant ‘personal’ parameters are associated with background agents and refer to aspects of the agents such as “home”, “workplace”, “father”, “driver”, or “coworker”. Production rules define how these different agents behave in different contexts, taking into account how parameters interact (e.g., how does “traditional religious” mix with “busy-body” to form a unique daily schedule that is reflective of an individual). Further, agents’ schedules are allowed to contain branches with probabilistic selection, and individual actions can have a specified probability of occurring. Basic behaviors include walking from place to place or via a route as a single agent or as part of a group; driving as a single agent or as part of a spontaneously setup carpool; conversations as part of scheduled actions or to set up other scheduled actions (like group movement or carpooling); conducting meetings with other agents; and picking up and dropping off objects. It is also possible to include parameters that define interactions between clutter agents and the environment, such as information flow between agents to indicate activity or events (e.g., transmitting knowledge of other agents’ locations) and reactions to natural (e.g., rain) and manmade (explosions) environmental events.

We have used two techniques to help make clutter characters’ behavior realistic. One method is to create a detailed database of characters and their parameters. For each character, we specify a name, a family, a house at a particular location, an occupation, age, gender, ethnicity, personality type, health, wealth, social network, religion, relatives, friends, and attitudes towards others. To lessen the authoring burden, we developed an initial generation algorithm for background characters, taking into consideration gender, age levels, prevalent cultural groups, occupations, and social networks. For example, it is possible to enable the training system to determine the level of religious observance of individual agents at the family level, as-
signing families at random to nearby religious buildings of the appropriate religious sect, then causing the individuals to attend religious events in accordance.

Another method has been to implement activity algorithms. These are essentially small subroutines relevant to certain characters that guide what (walk, play, work, pray), where (anywhere, residence, workplace, religious building), and with whom activities occur. The ‘when’ is determined in concert with the flexible scheduling and POL constraints defined by other parameters, while the ‘how’ is reliant on whatever animations and visualization methods the simulation environment provides.

We have implemented simple and more complex algorithms. As an example of simpler activity, during a study to evaluate social decision-making (Paschall et al., 2005), we developed two versions of a school hallway scenario, one in which routine background activity took place behind the central synthetic character (Figure 1). Similarly, for studies of vigilance and stress control, we devised a number of potential background distractors involving real-world sights and sounds (Hourani et al., 2011; Hubal et al., 2010). As an example of more complex activity, for a project integrating POL into an Army simulated in-country operational environment, we enabled detailed behaviors such as guiding visits to the marketplace, religious sites, teahouses, and the like, initiating actions such as taking breaks and heading home from work, and directing friends to meet up (youtu.be/CitevFwa3TE; Figure 2).

4.2 POL and Cultural Awareness within Central Agent Models

Culturally-aware foreground agents are those that also exhibit behaviors that reflect locally meaningful customs, but not as a general population member—instead, as an individual who blends with the general population but at times purposefully violate cultural norms to accomplish predefined goals within the simulation.

Central character agents follow schedules similar to clutter agents until specific times when, for reasons of training anomaly and normalcy, goal-based reasoning takes over; then control is handed back (e.g., to the FuSM) when goals are achieved. We have demonstrated high-fidelity central character prototypes that can perform urban POL use cases autonomously based on partially specified behavior definitions, ultimately allowing the generation of a large number of different central character behaviors in a single scenario. As one example, we developed a robust, reusable set of goal-based human behavior models for virtual, small-unit training exercises (Stensrud et al., 2012). In this work, an infantry squad arrives in town, fire teams separate as planned and walk in formation, and soldiers run for cover when a team member is hit by a sniper. In this environment we also modeled local civilian agents who mimic activity within the town, walking around the main road, conversing with other agents, and walking in and out of local markets, and respond to unusual activity such as gunfire or the presence of outside forces.

Central character behaviors need to be adaptive and varied, and this is the advantage of modeling them using a cognitive architecture. For example, central characters might walk faster than clutter agents or take purposefully evasive routes. Thus, an agent may have the goal of implanting an improvised explosive device in a public space. The agent may plan to take a circuitous route to reach the location at a specific time, but may change its route in the presence of fighting forces or abandon the plans altogether if its reasoning suggests the goal will not be achieved.

We augment central character behaviors by incorporating cultural parameters to affect production rules. Hence in the explosive device example, while clutter and central character agents essentially work from the same low-level pool of physical behaviors (they have the same visual representation of movement via animations), a central character might need to walk or drive taking that circuitous path. In essence, central character activities should fall within the distribution of clutter behaviors, except when (e.g., for training purposes) they should be forced into distinguishable behavior patterns. The goal is to create a system where culturally appropriate central character behaviors are not obviously different, while still allowing a training system to
5. Findings

There is reason to suspect that realistic background activity influences central interactions. In the school hallway scenarios, presence or absence of other characters and movement in the background appeared to have a slight but meaningful effect on participants’ performance. When there was background activity there were marginally more ambivalent statements (8% of statements vs. 3%), fewer pieces of information sought (10% vs. 16%), more provocative statements (6% vs. 3%), and instances of acquiescence to accommodate risky decisions (4% vs. 2%). These preliminary data did not reach significance, but are suggestive that participants were paying attention to what was happening in the background. Data that we collected from a stress control study (Hourani et al., 2011) seem to show that having a lot to respond to in a simulated environment led some participants to miss some cues. The same was true of participants interacting with a driving simulation having increasing levels of foreground and background activity (Mills & Hubal, 2001). One consideration these studies bring up is how to introduce the complexity of culturally-relevant behaviors, even whether or not to do so, during training; a systematic approach such as that described in Hubal & Frank (2001) may guide such decisions. Further work is needed to help clarify when and how background activity representing patterns of life influences users’ behaviors, especially during training, as they engage with central agents.

6. Future Work on Character Behavior Modification

The background agents developed for these environments demonstrate considerable range in realistic behaviors and are flexible and dynamic, as was intended for them to portray appropriate POL. The agents appear to add considerably to the realism of the training scenarios, but there are some improvements, modifications, and tests that might further enrich the POL behaviors. These challenges, especially in visually rich simulations, require socioculturally-relevant, manipulable data.

In some ground-level visualization environments, background characters are observed to stand in awkward or unrealistic positions. It is rare, for instance, in the real world, for a person to stand idle alone in the middle of a path for any extended period of time, or for multiple characters to stand idle within a small range. Instead, a person would typically find an alcove in which to wait or side wall on which to lean, or engage in conversation with another person nearby, or vary the idling pattern (in more urban contexts, perhaps window shop or check cellphone messages, or enter a building. Further, we have noticed characters, on occasion, stop in the middle of a route from one location to another, and idle with actions such as taking a smoke break. The characters are supposed to look for a convenient spot as marked on the map by an author. While the appearance of the actions themselves is not unrealistic, it is somewhat implausible for a person to stop in the middle of a path to do so. Instead a person would move to the side, or, more likely, engage in the desired behavior at the destination.

This observed unrealistic behavior can have multiple causes. One reason is better animation, the fidelity of which has been shown to influence how users interpret the scene (Hayes-Roth, 2004; Lane et al., 2013). Another source is a possible delay in our implementation of a plan, or of a goal not including positioning information. To improve characters behavioral realism here would require the coding of additional subgoals or constraints on goals. Indeed, we propose adding and studying culture appropriate proxemics, including the culturally normal space left between persons engaged in discussion or walking past and aware of each other, as well as the distance among disparate groups of people and how that distance affects local behavior.

Relatedly, when passing a duo engaged in conversation on an otherwise empty path, a central character should at least acknowledge the ongoing discussion, by waving, nodding, or simply looking that way, if not joining in. When the character does not acknowledge the discussion, then the background activity may be unnecessary. The same is not necessarily true at a more crowded space, such as a marketplace, where pairs and trios of conversationalists are the norm. We realize there is a whole class of “group interaction” problems that needs further exploration; the logistics of how pairs and trios intersect with individuals walking around or standing in line has implications of politeness, personality, and intensity in achieving goals. Implementing this feature should also involve mainly adding rules to character behavior planning, and parameters to suggest better and worse locations for the actions that lessen the need for authoring specifics.

A potential advance in cultural awareness involves social network characteristics. For instance, the friendship paradox is a theorem that the average number of friends of friends is always greater than the average number of friends of individuals (Feld, 1991). Practically, this paradox implies that certain nodes (friends) in a network contribute disproportionately to an average of activity across the network. There are other networking phenomena of interest, including centrality, mixing, and interdependence (Newman, 2010), Building on computational models of infectious disease and geospatial data for synthesized populations (Coo-
ley et al., 2008), we would like to study how these phenomena have cultural implications and incorporate them as appropriate into the synthetic character models. One consideration is to enhance cultural models based on lessons learned creating synthetic populations, based on models used by health agencies to track the spread of disease and drug-related emergency-room visits in the U.S., and on previously-developed synthetic populations for Mexico, Pakistan, India, and other countries (Wheaton, 2011). A further advance is to incorporate social network structures to model crowd movement (Wakamiya, Lee, & Sumiya, 2012).

7. Conclusions

Using fuzzy state logic, algorithms, and agent-based planning we have devised culturally appropriate pattern-of-life behaviors for both background and central agents to augment interaction skills training environments. Our intent is to address three design considerations for training simulation, background at scale, variation, and anomaly and normalcy. A series of parameters is used to make agents’ behavior reflect specific cultures. The systems described have been implemented in a number of virtual environments for training and other simulation purposes. Early tests of these culturally aware POL capabilities are promising, but more experimentation is warranted to determine their effectiveness in improving simulation training.

Acknowledgements

The authors would like to thank colleagues including David Fitzgerald and Brian Stensrud who were instrumental in this work. This work was supported by the U.S. National Science Foundation (NSF) under award IIS-0534211 and the U.S. Army Program Executive Office for Simulation, Training and Instrumentation under contract W900KK-13-C-0053. The views, opinions and/or findings contained in this report are those of the authors, do not necessarily represent the official position of NSF, and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

References


(Eds.), *The Science of Social Vision* (pp. 3-30). Oxford: Oxford University Press.

**Author Biography**

**ROBERT HUBAL**’s research focuses on using technology intelligently to better train and assess knowledge and skills across many domains. He has applied research results to such everyday domains as clinical assessment of social skills, improved patient communications, law enforcement interactions, survey non-response, and driving skills assessment.

**J.T. FOLSOM-KOVARIK** is part of SoarTech’s Intelligent Training area. His research relates to innovative methods for assessment and diagnosis of learner skills, states and traits, as well as adapting learner interactions with foresight and intelligence.

**ANGELA WOODS** is a software systems engineer at SoarTech with a background as a game developer. She also has experience with military simulations, and was a key contributor to the development of OneSAF versions 3.0-5.0. At SoarTech, she is currently the software lead for teams that create intelligent agent based systems and integrate them into military simulations and serious games.

**RANDY JONES** has been principal investigator for a number of SoarTech’s advanced R&D projects. His general areas of research include computational models of human learning and problem solving, executable psychological models, and automated intelligent entities for training and entertainment systems.

**JON CARBONE** is a systems engineer at SoarTech. He holds a Master of Science degree in Computer Engineering from the University of Central Florida.
Predicting Trust Dynamics and Transfer of Learning in Games of Strategic Interaction as a Function of a Player’s Strategy and Level of Trustworthiness

Michael Collins a, b
Michael.Collins.74.ctr@us.af.mil
937-694-1065
Ion Juvina a
Ion.juvina@wright.edu
Gary Douglas b
Douglas.45@wright.edu
Kevin Gluck b
Kevin.Gluck@us.af.mil

a Department of Psychology, Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH 45435
b Air Force Research Laboratory, 2620 Q Street, Building, 852 Wright-Patterson Air Force Base, OH 45433

Keywords:
Cognitive Modeling, A Priori Model Prediction, Strategic Interaction, Trust Dynamics, and Transfer of Learning

ABSTRACT: Individuals playing a sequence of different games have shown to learn about the other player’s behavior during their initial interaction and apply this knowledge when playing another game with the same individual in the future. Here we use a published computational cognitive model to generate predictions for an upcoming human study. The model plays both Prisoner’s Dilemma and Chicken Game with a confederate agent who uses one of two predetermined strategies and whose level of trustworthiness is manipulated. We go beyond the standard postdictive practice and adopt the increasingly popular practice of using the model to make a priori predictions before the human data will be collected in an upcoming study.

1. Introduction and Background

How people learn to trust one another over time and how they use this information to inform their future decisions is a question relevant to many aspects of human interaction. Trust is defined as “the willingness of a party to be vulnerable to the actions of another party based on the expectations that they will perform a particular action” (Mayer, Davis, & Schoorman, 1995). Trust relationships have been proposed to be self-sustaining once developed, allowing individuals to forgo re-evaluation of a person after they have been determined to be trustworthy (Hardin, 2002). Yamagishi, Kanazawa, Mashima, and Terai (2005) found that when participants played a modified version of the game Prisoner’s Dilemma (PD), where participants could choose the amount of points they could risk during each round, over time participants would gradually increase the number of points they would risk as the individuals began to establish trust for one another. Consistent with these results, Castelfranchi & Falcone (2010) suggest that trust mitigates risk and develops through gradual risk-taking between two individuals.

In order to study how individuals behave in different situations, both economic and psychological research have used games of strategic interaction. A game represents an abstraction of a real-world scenario in which participants can win and lose points based on the behavior of both players. Participants can play either with another human participant (e.g., Juvina, Saleem, Martin, Gonzalez, & Lebiere, 2013) or with a preprogrammed strategy (e.g., Juvina, Lebiere, Martin, & Gonzalez, 2012).

Two different strategies that have been used in place of human participants during these games of strategic interaction are the Tit-for-Tat (T4T) (Axelrod, 1984) and the Pavlov-Tit-for-Tat strategy (PT4T) introduced by Juvina, Lebiere, Gonzalez and Saleem (2012). T4T is a simple strategy, which repeats on round N the same choice that the other player made on round N-1. The PT4T strategy is a combination of two different strategies, T4T and Pavlov. Pavlov is another simple strategy that continues to choose the same choice on round N as long as it earned points with that choice on round N-1, only changing choices on round N when it lost points on round N-1. The PT4T strategy repeats the other player’s move from N-1 on round N, just as the T4T strategy, except for when the strategy and the other player make opposite choices and the strategy earns points on that round. Instead of switching to the other player’s choice as the
T4T strategy would, the PT4T strategy repeats its previous choice, as the Pavlov strategy would. The PT4T strategy was created based on analysis of the repetition propensities (the probability to repeat a move following a certain outcome) of humans in PD and in an attempt to develop a strategy that had similar repetition propensities as humans (Juvina et al., 2012).

Previous research has found that when individuals play games of strategic interaction sequentially, they use the information gained about the other player from a previous game to inform their choices when playing with that person again (Juvina et al., 2013). Different explanations have been offered for why these transfer effects occur, such as a similarity between the games, the expectation of the other player to behave as they did in the past, or a strategy that was used during a simpler game continuing to be used in a more complex game (Knez & Cramer, 2000; Devetag, 2003; Bednar, 2012).

Juvina et al. (2013) found that these explanations failed to account for the transfer effects seen when repeated rounds of the games Prisoner’s Dilemma (PD) and Chicken Game (CG) were played sequentially. As an alternative explanation for why transfer effects occur between these games, Juvina et al. (2013) proposed that it is the increase in reciprocal trust between the two players that results in a transfer of learning occurring between these games, allowing them to find the optimal outcome faster in the second game compared to the first. Juvina, Lebiere, and Gonzalez (2014) implemented this idea of reciprocal trust in a computational cognitive model that replicates the transfer effects seen when the games PD and CG are played sequentially in either order.

The results in Juvina et al. (2014) were obtained by fitting the model post-hoc to the human data from Juvina et al. (2013) by manipulating certain model parameters. However, fitting the model post-hoc to a specific dataset does not ensure its validity and generalizability. In order to fit the human data, the model played against itself, using both the same parameters and learning mechanisms to determine how to play both games. This is problematic when trying to understand real world scenarios where individuals are likely to have different goals and understandings of the current situation. Due to these differences, it has not yet been shown that the model can account for human behavior when playing against an individual who has a different approach and a different level of trustworthiness.

We are attempting to validate the model used in Juvina et al. (2014) by using the model to simulate the results of an upcoming study to be conducted with human participants. The model will play two games sequentially, either PD and CG in varying orders or one of the two games twice with a preprogrammed confederate agent. The confederate agent will use one of two predetermined strategies and will have varying levels of trustworthiness. A comparison of the model’s predictions to the behavior of human participants will allow for an opportunity to examine in what types of situations the model can predict the behavior of human participants. In this article, a brief overview of the model and the experimental design of the simulation is offered, along with a discussion of the model’s predictions for the upcoming study to be conducted with human participants.

### 1.1 The Games

Participants will play repeated rounds of the same two games used in Juvina et al.’s (2013) original study, which are PD and CG. Both PD and CG are mixed motive non-zero sum games and are represented by their own payoff matrix (Fig 1.1). During each round in a game, both Player 1(P1) and Player 2 (P2) choose to either defect (A) or cooperate (B). Based on the choices made by both players during every round, P1 or P2 either win or lose a certain number of points.

![Payoff Matrix](image)

Fig 1.1. The payoff matrix for the game Prisoners dilemma (left) and Chicken Game (right).

When either PD or CG is played continually and both players do not know how long they will play, each game has a different optimal outcome. In PD, the optimal outcome over the course of the game is for both players to choose B (mutual cooperation) in order to earn one point each during each round (Fig 1.1). In CG, the optimal outcome is for both players to asymmetrically alternate between choosing A and B, earning three points every other round (Fig 1.1). However, when playing either CG or PD, attempting to choose the optimal outcome is risky. If only one player understands the benefits of sustaining the mutual cooperation or alternation outcome and is willing to reciprocate, then the player who attempts the optimal strategy will lose points as the other player gains points. To avoid this, players must learn to mutually cooperate with one another by sustaining the optimal outcome throughout the game, which maximizes their payoffs when either PD or CG is played repeatedly (Juvina et al., 2013). Due to the fact that each game has a different optimal outcome, the behavior of both players should change along with the games that are played.
Although PD and CG have different payoff matrices, certain characteristics are similar across both games. There are both surface and deep similarities. The surface similarity between PD and CG is that relevant in this context is that both players during either game can choose B to earn one point during each round. Both games also share a deep similarity that is both players mutually cooperating with each other brings about the optimal outcome when either game is played repeatedly. Players can mutually cooperate by both choosing B in PD and asymmetrically alternating between A and B in CG (Juvina et al., 2013). Juvina et al. (2013) has found that when PD and CG are played sequentially the transfer effects between these games occur along both the surface and deep similarities. In particular, more mutual cooperation was seen in PD when played after CG and more alternation was seen in CG when played after PD.

1.2 The Model

A brief overview of the model used to generate the predictions of the upcoming study is given here; a more detailed description of the model can be found in Juvina et al. (2014). The model was built in ACT-R (Adaptive Control of Thought - Rational), which is both a cognitive architecture and a theory of human cognition (Anderson, 2007). In ACT-R, different modules interact with each other in order to complete a task. In the model used for this study, two memory modules are used in order to play both games; these are the declarative and procedural modules. The declarative module stores information that the model has learned from the environment. The procedural memory allows for action selection reinforced through reward patterns that occur within the environment (Anderson, 2007). Both modules are used together to account for human behavior in the two games when played independently and sequentially.

In order for the model to be able to play either game, it needs to be aware of the interdependence between itself and the other player; to do this the model uses instance-based learning (IBL: Gonzalez, Lerch, & Lebiere, 2003). In IBL, past instances of an event are stored in a model’s declarative memory to be recalled later, and inform future decisions. When the model is in a situation similar to a previous experience, it uses information stored in its declarative memory to make a decision about what to do in its current situation. At each round, the model stores in its declarative memory the previous move of both itself and the other player along with the other player’s move for the current round. Throughout both games, each time the model stores a copy of a previous instance that has already been placed in its declarative memory it increases the probability that that specific instance will be recalled when placed in a similar situation again, as controlled by ACT-R’s activation equations (Anderson, 2007).

To account for the behavior of the human participants in each game, the model uses both IBL and reinforcement learning. During each round, the model attempts to recall a previous instance from memory using both its own and the other player’s previous move as retrieval cues. The stored previous instances in the model’s declarative memory allow it to recall what the other player’s next move was when placed in that situation before. The model predicts that the other player will choose the move that was chosen more frequently when placed in similar situations in the past. The model then chooses to cooperate or defect depending on which choice has the greatest utility given the model’s prediction of the other player’s move. Previous rewards the model has received for cooperating and defecting in similar contexts (i.e., the other players expected next move based on the previous move of the other player and the model) determine the utility or the value of these choices to the model (Juvina et al., 2014).

In order to account for the deep transfer effects seen when PD and CG are played sequentially, two trust accumulators and three different reward functions were added to the model. The two accumulators are called trust and trust-invest. Each accumulator starts at zero at the beginning of the first game and increases or decreases depending on the moves both the model and the other player make after each round. The trust accumulator increases when both players either mutually cooperate or when the model defects and the other player cooperates. It decreases when both players mutually defect or when the model cooperates and when the other player defects. The trust-invest accumulator increases with mutual defections and decreases with unreciprocated cooperation. Throughout either game the current levels of the trust and trust-invest accumulators determine the model’s current reward function.

Three reward functions are used which reinforce the model’s choices differently for each of the four possible outcomes that can occur during a game, in turn affecting the model’s behavior. By alternating between three different reward functions, the model uses the reward function that is most applicable to its current situation. The reward function that is applied to the current round of the game is determined by the level of the trust and trust-invest accumulator. When the trust accumulator is positive, the model is reinforced for increasing the payoff of both players. When only the trust-invest accumulator is positive, the model is reinforced for increasing the payoff of the other player. When both accumulators are at or below zero, the model is reinforced for maximizing its own payoff and minimizing the payoff of the other player.

2. The Experiment

The model predictions presented in this paper were generated by simulating a fully balanced 4 x 2 x 2
strategy that will be conducted with human participants. Participants will play both PD and CG or one of these two games twice. Instead of participants playing games with one another as in Juvina et al.’s (2013), participants will play with a “confederate agent”, implemented as a software agent. The confederate agent will use one of two predetermined strategies and the trustworthiness of the agent will be controlled, while playing both games. The model was run in conditions identical to those that future participants will be placed in.

On Qualtrics.com, the online platform that will be used to run the upcoming experiment, we created sixteen conditions with each possible combination of game order, confederate agent’s strategy, and trustworthiness. In each condition, ten preprogrammed versions of each game were developed to ensure random variability in the behavior of the confederate agent. Once the experiment begins participants will first be randomly assigned to a condition and then randomly assigned to play one of the ten possible versions of each of the two games they will play during the experiment. The experimental protocol for the upcoming study was copied when generating model predictions, simulating fifty human participants in each condition.

2.1 The Confederate Agent

The confederate agent will utilize one of two predetermined strategies throughout both games. The T4T strategy will choose on round N the choice that the other player made on round N-1. The PT4T strategy will reciprocate mutual cooperation and defection, but will not reciprocate unilateral cooperation.

Along with using one of two predetermined strategies, the confederate agent’s trustworthiness will be manipulated and randomness will be added into its behavior. To accomplish this, the confederate agent will either cooperate or defect a certain number of times throughout each game at random times. In the high trustworthiness (HT) condition the confederate agent will cooperate and in the low trustworthiness (LT) conditions the confederate agent will defect. For this experiment, we wanted to create a confederate agent that would generate significant differences in the outcomes that were chosen across all conditions. To accomplish this, multiple model predictions for all conditions were run by varying the number of rounds the confederate agent employed its strategy (reactive strategy – T4T or PT4T) and automatic cooperation or defection (fixed strategy). We found that, because PT4T is inherently less trustworthy than T4T (i.e., more apt to defect), to avoid the model only predicting a high frequency of mutual defection during the PT4T HT conditions, a larger percentage of cooperation was needed to raise the strategies trustworthiness. For this experiment, during the T4T conditions, the confederate agent will employ the T4T strategy randomly during 90% of the game, while randomly employing its fixed strategy during 10% of the game. During the PT4T conditions, the confederate agent will employ the PT4T strategy randomly during 65% of the game and randomly employ its fixed strategy during 35% of the game.

3. Results and Discussion of the Model’s Predictions

We computed the frequency of five relevant outcomes during each round in every condition over the fifty different model runs. In order to determine instances of asymmetrical alternation, rounds where one player chose to defect while the other player cooperated or vice versa on round N and had both chosen the opposite choices on round N-1 were identified. The frequency of alternation during each round across all conditions was computed like all other outcomes. Because of the limitation of space in this paper, we cannot report all of the results. All of the model’s predictions are available for viewing and can be downloaded at (http://psych-scholar.wright.edu/jjuvina/publications). A linear mixed effects analysis (LME) was used to assess the effect of strategy, trustworthiness of the confederate agent, and order in which the games were played on the predicted frequency of the five outcomes. P-values were obtained using a likelihood ratio test comparing a full to a reduced model. The 95% confidence intervals for the effects predicted by the LME are also reported. It should be noted that the confidence intervals that are reported are large, which is expected given the large variability generated by each ACT-R model, the randomness added to the confederate agent, and the multitude of experimental conditions. The model’s predictions will be compared to human data from each condition, once the experiment has been run.

Transfer effects were assessed using a paired t-test, run on the frequency of each outcome during the first game compared to the frequency of that outcome when the same game was played second against a confederate agent of the same strategy and level of trustworthiness. Significant results indicate that the order in which the model played the game affected the frequency that an outcome was chosen during that game.

3.1 Effects of Trustworthiness

One clear difference seen across the high and low trustworthiness conditions in the model’s predictions is the level of the trust accumulator. A t-test run on the round-by-round average of the magnitude of the trust accumulator across the simulated low (M = 66.86, SD = 38.11) and high (M = 62.36, SD = 39.17) trustworthiness conditions was
found to be significant ($\chi^2(49) = 66.87$, $p < .001$). The model’s current level of the trust accumulator affects which current reward function is used and will determine whether the model will attempt to maximize its own payoff or the payoff of both players. The difference in the trust accumulator between the simulation of the high and low trustworthiness conditions indicates that the experimental manipulations of trustworthiness were effective. Based on its level of trust, the model predicts a difference in the frequency that mutual defection will occur in both games, despite differences in the strategy used by the confederate agent and order.

A LME was run with the average predicted frequency of mutual defection as a dependent variable, trustworthiness of the confederate agent as a fixed effect, with strategy, order, and round as random factors. A likelihood ratio test was run and found that the trustworthiness of the confederate agent was found to have a significant effect on the predicted frequency of mutual defection ($\chi^2(1) = 277.3$, $p < .001$), increasing the frequency of mutual defection by $75.07\% \pm 6\%$, 95% CI [52\% , 98\%], during the simulated low trustworthiness conditions compared to $15.4\% \pm 6.5\%$, 95% CI [0 \% , 37.33\%], in the simulated high trustworthiness conditions (Fig. 1.2).

Fig 1.2. The average round-by-round frequency that mutual defection was chosen across all of the simulated high (dashed red line) and low (solid black line) trustworthiness conditions.

The trustworthiness of the confederate agent determines whether it will cooperate (high trustworthiness) or defect (low trustworthiness) for a specific number of times (10\% of the rounds in the T4T and 35\% of the rounds in the PT4T) over the course of the game at random times. The model predicts that participants will be sensitive to the trustworthiness of the confederate agent, responding by defecting more throughout the low trustworthiness conditions and less during the high trustworthiness conditions.

3.2 Effects of Strategy

The two types of strategies used by the confederate agent have different criteria for deciding what choice to choose during each round; these differences limit how quickly the model can change from one outcome to another and the outcomes that can be achieved during a game. For example, continual alternation is an outcome that can only be achieved with the T4T strategy and not with the PT4T strategy. Continual mutual cooperation is also an outcome that is harder to achieve with the PT4T strategy, because it is inherently less trustworthy (i.e., more apt to defect). It is the differences in the behavior of these two strategies used by the confederate agent that affected the predicted frequency in which the optimal outcomes will be chosen despite differences in the trustworthiness of the confederate agent or the order in which the games are played.

A LME was run with the average predicted frequency of mutual cooperation as a dependent variable, strategy as a fixed factor, with trustworthiness, order, and round as mixed effects. A likelihood ratio test was conducted and found that the strategy implemented by the confederate agent significantly affected the predicted frequency of mutual cooperation ($\chi^2(1) = 68.867$, $p < .001$). The T4T strategy had a larger affect on the predicted frequency of mutual cooperation, increasing its predicted frequency by $25.1\% \pm .7\%$, 95% CI [0 \% , 70\%] compared to when the confederate agent used the PT4T strategy, increasing the predicted frequency of mutual cooperation by only $19\% \pm 13.7\%$, 95% CI [0 \% , 62\%] (Fig 1.3). A second LME was run with the average predicted frequency of alternation as a dependent variable, strategy as a fixed factor, with trustworthiness, order, and round as random factors. Similar to mutual cooperation, the strategy used by the confederate agent was found to have a significant effect on the predicted frequency of alternation ($\chi^2(1) = 392.21$, $p < .001$). Conditions where the confederate agent used the T4T strategy had a larger affect on the predicted frequency of alternation, increasing the frequency by $12.9\% \pm 0.4\%$, 95% CI [6\% , 30\%] in conditions where the confederate agent used the T4T strategy compared to only $4\% \pm 6\%$, 95% CI [0 \% , 20\%] when it used the PT4T strategy (Fig 1.3).

The strategy used by the confederate agent was also found to have a significant effect on the predicted frequency of mutual defection, controlling for trustworthiness and order. A LME was run with the average predicted frequency of mutual defection as a dependent variable and strategy as a fixed factor, with trustworthiness, order, and round as random effects. A likelihood ratio test was conducted and found that the strategy used by the confederate agent had a significant effect on the predicted frequency of mutual defection ($\chi^2(1) = 574.02$, $p < .001$). Conditions where the confederate agent used the PT4T
strategy had a larger effect on the predicted frequency of mutual defection, increasing its frequency by 54.1% ± 29%, 95% CI [0%, 100%], compared to when the confederate agent used the T4T strategy increasing the predicted frequency of mutual defection by only 36.31% ± .6%, 95% CI [0% , 100%] (Fig 1.3).

The model predicts that participants will react differently to the two different strategies used by the confederate agent. Alternation and mutual cooperation are both predicted to occur at a higher frequency during all of the T4T conditions compared to the PT4T conditions. A higher predicted frequency of alternation occurring during the T4T conditions would be expected, because the PT4T strategy cannot continually alternate throughout the game like the T4T strategy. However, the T4T and PT4T strategy can both mutually cooperate throughout a game. The difference that the frequency of mutual cooperation is predicted to occur is caused by the strategies’ behavior during the experiment when played with repeatedly, because repeated instances of mutual cooperation are harder to obtain with the PT4T strategy than with the T4T strategy. In addition, as is seen in the model’s predictions, the PT4T condition is predicted to have a higher frequency of mutual defection across all conditions, which would affect the model’s trust in the confederate agent, leading it to cooperate less in conditions where the confederate agent used the PT4T strategy compared to the T4T strategy.

3.3 Effects of Order

The optimal outcomes that are chosen during the experiment depend on the games that are played during each condition. For example, alternation is the optimal outcome in CG, but is not an optimal outcome in PD, because alternating between a payoff of +4 and -4 points per round leads to a net gain of 0 for both players. While playing PD, mutual cooperation is the optimal strategy and though mutual cooperation is a possible outcome in CG, it leads to a sub-optimal outcome compared to alternation, +1 point per round compared to +3 points every other round. Juvina et al. (2013) found that order also affects the frequency of the optimal outcomes during a game. The optimal outcome in either PD or CG occurred more frequently when it was played after the other game compared to when played first. Due to the effects that order has been seen to have on the outcomes that are chosen over the course of both games, the model will predict a significant difference in the frequency of the two optimal outcomes over the course of the two games depending on the order that they are played.

A LME was run with the average predicted frequency of mutual cooperation as a dependent variable, order as a fixed effect, with trustworthiness and strategy of the confederate agent and round as random effects. A likelihood ratio test was conducted and found that the order in which the games were played in a condition significantly affected the frequency of mutual cooperation ($X^2(3) = 712.98, p < .001$), increasing the predicted frequency of mutual cooperation by 36.6% ±1%, 95% CI [0% , 79%] in the simulated conditions when PD was played repeatedly (PDPD order), 28.47% ± 1%, 95% CI [0% , 71%], when PD was played before CG (PDCG order), 13.10% ± 1% , 95% CI [0% , 71%], when CG was played before PD (CGPD order), and 10% ±12.3%, 95% CI [0% , 51%], when CG was played twice (CGCG order).
To test the significance of the effect of order on the predicted frequency of alternation, a LME was run with the average predicted frequency of alternation as a dependent variable, order as a fixed factor, with trustworthiness and strategy of the confederate agent and round as random effects. The order in which the games were played was found to significantly affect the predicted frequency of alternation, opposite that of the predicted frequency of mutual cooperation ($X^2(3) = 712.98, p < .001$). Game order affected the frequency of alternation by $15.5\% \pm 5.9\%, 95\%$ CI [0%, 33%], in simulated conditions with the CGCG order, $11.9\% \pm .5\%, 95\%$ CI [0%, 23%], in the CGPD order, $4.95\% \pm .05\%, 95\%$ CI [0%, 23%], in the PDCG order, and $1.86\% \pm .05\%, 95\%$ CI [0%, 20%], in the PDPD order (Fig 1.4).

The affect that the order games were played had on the predicted frequency of the optimal outcomes show that in conditions where the same game is played repeatedly, such as in the PDPD and CGCG order, the model predicts that the frequency of the optimal outcome for that game will continue to increase throughout the condition. The model also makes an uncharacteristic prediction about the frequency that mutual cooperation and alternation in the conditions simulated with the PDCG and CGPD order. It would be expected based on results from Juvina et al. (2013), that conditions with the PDCG order would have a higher frequency of alternation than the CGPD order, and that the CGPD order would have a higher frequency of mutual cooperation than with the PDCG order. Instead, the model predicts that when PD and CG are played in sequence, the highest frequency of mutual cooperation will be in conditions with the PDCG order and the highest frequency of alternation will occur in conditions with the CGPD order.

3.4 Predicted Transfer Effects

Previous results with human pairs have found that when PD and CG were played in sequence, transfer effects between these two games occur along both their surface and deep similarities (Juvina et al., 2013). The same transfer effects have also been found when cognitive models were paired with one another (Juvina et al., 2014) In contrast, when a cognitive model was paired with a pre-programmed agent as in the current study, no deep transfer effects are predicted; the model only predicts surface transfer effects. Mutual cooperation in the T4T HT condition is predicted to occur at a higher frequency during CG when played after PD compared to when played before PD ($t(49) = -21.8871, p < .001$). The same prediction about the frequency of mutual cooperation is made during the PT4T HT condition. Mutual cooperation is predicted to occur at a higher frequency during CG when played after PD compared to when played before PD ($t(49) = -38.429, p < .001$).

The surface transfer effect of mutual cooperation in the PDCG order during the PT4T HT condition is amplified by the limitations of the confederate agent’s strategy. Because continual alternation cannot be achieved with the PT4T strategy, mutual cooperation, a sub-optimal outcome in CG, is left as the only satisfactory outcome that can be achieved given the behavior of the confederate agent. One possible explanation for the lack of deep transfer effects in the model’s predictions is the difference between the behavior of the confederate agent and an actual human player. The confederate agent is simpler than the model (even with the added randomness) and does not learn from the interaction with the model throughout the game. If confirmed, the prediction of a lack of deep transfer will strengthen the claim made in Juvina et al. (2013, 2014) that joint learning and reciprocal trust are key ingredients for a deep transfer of learning in games of strategic interaction.

4. Conclusion
In summary, we are validating a computational cognitive model that has shown to be able to account for the transfer effects that are observed when the games PD and CG are played repeatedly and in sequence with human participants. In order to validate the model, we have made a priori model predictions about the behavior of human participants when playing against a preprogrammed confederate agent across a variety of conditions. From the model’s predictions we have developed five hypotheses for the upcoming study.

**H1**: We predict that mutual defection will be chosen more across all of the low trustworthiness conditions compared to the high trustworthiness conditions.

**H2**: We predict both optimal outcomes (i.e., mutual cooperation and alternation) will be chosen at a higher frequency in conditions where the confederate agent uses the T4T compared to the PT4T strategy.

**H3**: We predict that the frequency of both optimal outcomes (i.e., mutual cooperation and alternation) will depend on the order that games are played in a condition.

**H4**: We predict that across the sixteen conditions no deep transfer of learning will occur.

**H5**: We predict that across the sixteen conditions surface transfers of learning will only occur with the mutual cooperation outcome in the PDCG PT4T HT and PDCG T4T HT condition.

We expect to run the study in 2015. A subsequent publication will reveal the actual empirical results and degree of model predictive validity.

5. References


Author Biographies

**Michael Collins** is a current undergraduate in the Department of Psychology at Wright State University working as a research assistant at the Air Force Research laboratory. His research interest include computational cognitive modeling, game theory and social decision making.

**Ion Juvina Ph.D** is an Assistant Professor in the Department of Psychology at Wright State University. His research interest span the areas of strategic decision making and executive control of cognition and affect. He teaches cognition and learning, psychological game theory, computational cognitive modeling, and experimental design.

**Gary Douglas** is a current human factors graduate student at Wright State University. His research interest include cognitive modeling, trust, and game theory.

**Kevin Gluck Ph.D.** is a Principal Cognitive Scientist at the Air Force Research laboratory and is currently the Acting Technical Advisor for the Warfighter Readiness Research Division.
ABSTRACT: Experimentation using the Crowd Behavior Testbed demonstrates the feasibility of incorporating social network methods into laboratory research with real crowds of real people. For the last two years, the Target Behavioral Response Laboratory has conducted laboratory research on crowd behavior in response to simulated non-lethal weapons. Data and results from this testing reveal social network analysis methods that are relevant to non-lethal weapon testing. Subjects participated in an experiment investigating crowd behavior and response to a control force. During the entire time that subjects were participating, crowd behavior and interactions were videotaped. Videotape recordings of interactions during engagements with control force and informal interactions between crowd members were coded for inter-member interactions. These social communications and interactions were subjected to social network analysis to identify leaders and other influential crowd members, hubs, isolates, dyads, triads, and clusters of nodes (individuals). Two other sources of data were analyzed using network analysis—1) before the study, subjects identified the individuals they had known before the test, 2) after the main crowd-control force experiment, subjects also identified those they thought acted as leaders or were highly capable of influencing the crowd. Social network analysis was then conducted to identify patterns of pre-existing social bonds as well as to identify informally nominated leaders in the group. Procedures to characterize crowds based on social network analysis methods are proposed.

1. Introduction

1.1 Social Network Theory and Crowd Behavior in Response to Non-lethal Weapons

The work described in this report is preliminary and pilot explorations in the utility of Social Network Analysis methods in controlled laboratory investigations of crowd scenarios. The methods of network science may be useful tools in characterizing and quantifying psychosocial crowd characteristics relevant to non-lethal weapon effectiveness. The origin of the science of network analysis was in the analyses of groups of persons and the relationships among them, that is, social network analysis (Carrington, Scott, & Wasserman, 2005; Wasserman & Faust, 1994). Therefore, since the method was fundamentally developed for understanding groups of people, it is not difficult to imagine that these methods may be useful in studying crowds that the Warfighter might encounter (Mezzacappa, Cooke, & Yagrich, 2008). The TBRL uses social network theory as a method of characterizing social bonds within the groups and sub-groups that make up the larger collective of the crowd. The rationale is based on the realization that the behavior of the crowd is greatly influenced by the social bonds among the crowd members. These connections can either be social (e.g., friends, family, clansmen) driven by homophily (having the same attribute, e.g., co-religionists) or simple proximity (the strangers next to you) driven by propinquity (being at the same place at the same time).

The structure of a network is specified by indicating which nodes are linked to other nodes. In studies of groups of people or crowds, the nodes are people, and the links are their relationships or interactions. Within social network analysis, relationships may be psychosocial (as in cohesive bonds between crowd members) or physical (as in distances between crowd members and control force members). The outcomes of social network analysis are indices of characteristics of crowds (e.g., how much intercommunication occurs, how cohesive) and roles of persons in the crowd (e.g., influential persons, isolates).

Given that part of crowd threat assessment includes listening to communication among the crowd, as well as identification of leaders or instigators, the tools of social network science should be investigated in crowd behavior research. Measures such as closeness, betweenness, radiality, structural cohesion, and many others can be adapted for investigations of crowd
behavior; however, the specifics of these network analysis metrics are beyond the scope of this paper (Carrington, Scott, & Wasserman, 2005; Wasserman & Faust, 1994). However, there are three face-valid measures of crowd social characteristics readily amenable to social network analysis: pre-existing social bonds, ongoing social interactions, and identification of leaders and other influential persons.

1.1.1 Pre-Existing Social Bonds

Within a crowd there are a few types of social bonds that are of interest. The first is pre-existing social bonds. That is, there is evidence that persons in the crowds are known to each other, that they may participate in groups of friends, family, or acquaintances (Kenny, et al., 2001). Behavior of a crowd that consists of families could be reasonably hypothesized to be different from that of a crowd that consists of strangers. Therefore, it is of interest to study the composition of the crowds that might consist of small groups of friends who gathered together for the event. That is, the empirical question can be asked “What effect do the pre-existing social bonds have on the crowd response to control force actions?”

1.1.2 Ongoing Social Interactions

During the crowd gathering and through its existence, crowd members are in proximity with each other and will engage with social interaction with other crowd members to exchange information or share observations or give directions. These communications and interactions (nature and frequency) can be hypothesized to relate to crowd responses to control force tactics. For example, doctrine calls for the removal of instigators, that is, persons within the crowd who exhort others to violence (Department of the Army Headquarters, 2005). That is, crowd members influence each other through communications and other social interactions. Therefore, those communications and social interactions occurring within the crowd are of interest in understanding crowd responses to control force methods.

1.1.3 Identification of Leaders and Other Influential Persons

Social status within the crowd is also an important psychosocial variable to consider within crowds. Crowds differ with respect to presence of leaders - followers within the group, and with respect to emergence of leaders during the crowd’s formation and existence. That is, some crowds have formal recognized leaders that wield some authority and can speak for the crowd. Other crowds may not have a formal leader and followers; rather, circumstances reveal a leader, that is, the leader of the crowd emerges during the engagement with the control force. These influential persons can direct the crowd by command or by action. Because of their influence on crowd behavior, a study of methods of identification of leaders and a study of the distinct overt behaviors of leaders are of interest to researchers.
1.2 The Present Work

The present work is a preliminary investigation of the use of social network analysis to assess pre-existing social bonds, social interaction during the life of the group, and emergent leaders within the group. The work was carried on in the context of crowd behavior experiments. The intent is to investigate the associations between social networks within the crowd, and crowd behavior in interaction with a control force wielding non-lethal weapons and their tactics, techniques, and procedures.

First, a brief description of the crowd testbed will be given, followed by an overview of the general crowd experimental procedures. Then, the gathering of data for social network analyses will be described. Finally, the preliminary results will be discussed within the context of understanding crowd - control force interactions. The work concludes with the way forward in the use of social network analysis in investigations of crowds and non-lethal weapons.

2. METHOD

2.1 The Testbed

The Crowd Behavior Testbed’s primary test area is 50 feet square with overhead trussing and a padded floor. It is fully equipped with a comprehensive suite of experimental controls, computer hardware and software, cabling, devices that are sources for stimuli, and means for data capture. The testbed includes computer control systems, motion capture (MOCAP), and audio/visual equipment. Most importantly for the present work is that the truss system includes several cameras surveying the entire testbed, capturing audio and video recordings of all crowd-control force interactions.

At one side of the testbed was the room in which the crowd rested between experimental runs. In this room was a projector screen that was used to instruct the crowds on the experiment as well as other information subjects needed during the experiment. At the goal end of the testbed was the target that was used in the crowd simulation scenarios. The area of blue mat had sawhorses demarcating the “line in the sand” as well as military vehicles (Commercial Utility Cargo Vehicles, CUCVs) placed there for greater fidelity to real operations. The perimeter around the testbed was outlined by empty plastic jersey barriers and sawhorses.

2.2 Experimental Procedures

2.2.1 Human Research Ethics Approval

All human research methods used in this study were approved by the ARDEC Institutional Review Board, which oversees the ethical conduct of research. Subjects were recruited from the general public and paid $20/hour for participation.

2.2.2 Participants

Participants were all over 18 years old and had no condition that impeded their movement or ability to walk (n=19, 10 women, 9 men). The mean age of a participant was 30.74 years (sd=14.72, range 19-63 years old). The ethnic compositions of the crowds were representative of the local demographics.

2.2.3 Stop Approach Scenario

Subjects performed a task that simulated the tactical construct of a crowd facing an area protected by a control force. The control force members wielded stand-off projectile weapons. The simulated stand-off weapon used was the Nerf® gun (www.hasbro.com). This simulated weapon shoots a foam dart; the magazine holds 10 darts. Pilot testing showed the range of the weapon to be 20-30 feet. The ends of the foam darts were covered in blue chalk to mark impacts on the white tee-shirts worn by all subjects. Each chalk mark resulted in a loss of points for the subject.

2.2.4 Paradigm Overview

Subjects were tasked with throwing “rocks” into a target guarded on each side by two control force members (Fig. 1). Each successful throw was awarded points and dollars for the group. Subjects also were tasked with avoiding being hit with projectiles from the stand-off weapons. Each subject was provided with four simulated rocks, marked with the subject number, to be thrown for points. Subjects could choose to be at any starting point on the blue mat. The subject gained money if he or she successfully threw the rocks into the target. The start of the trial was signaled by “Ready. Set. Throw!” broadcast over the loudspeaker. The crowd then had one minute to approach the target and throw the rocks.

During the time the crowd members were allowed to throw the rocks, they were met with the control force personnel wielding simulated stand-off weapons. Subjects were induced to avoid the control force by a loss of points if they were hit by the foam projectiles shot from the simulated stand-off weapon. For every contact or impact with the foam projectiles, the person individually lost several dollars and points from his or
her score. Subjects were wearing identifying numbers to facilitate recording of data.

### 2.2.5 Control Force

Control force manipulations were carried out by the TBRL research staff. The control force was dressed in gear that simulates control force garb: face shield, uniform, boots, and vest. The communications and actions of the control force were carefully scripted and managed. Subjects were instructed not to make physical contact with the control force or any person in the group. Non-compliance resulted in warnings and ejection (while warnings were necessary, no subject was dismissed for this reason). Control force Rules of Engagement were simply to target anyone forward of the sawhorses (the “line in the sand”).

### 2.3 Data Collection for Social Network Analyses

#### 2.3.1 Assessment of Pre-Existing Social Bonds

After consent was obtained, subjects completed a Participant Personal Information Questionnaire that collected health and demographic information. As part of this information, subjects were to indicate other persons in the group they knew. This information was entered into a matrix that was submitted to social network analysis to assess preexisting social bonds in the group.

#### 2.3.2 Assessment of Social Interactions

In the experiment, the crowd engaged in 22 one-minute engagements with the control force. During the time of all engagements, video cameras recorded the entire interaction (both audio and video). Several cameras were mounted on the trusses above so that the entire testbed area was captured.

Video recordings of the experiment were then coded for interactions between subjects. For these preliminary analyses, three two-minute epochs were scored, for the one minute before the engagement and for the one minute during the engagement. Epochs were scored using a simple matrix with Subjects 1-19 across the top and Subjects 1-19 down the side. The videotapes were coded for presence or absence of social interactions between each pair of the 19 subjects. A social interaction was defined as 1) verbal communication, 2) physical contact ("high-fiving"), 3) gestures toward another member ("thumbs up sign"), and 4) non-verbal auditory signaling (clapping). If at any time during the two-minute epoch two subjects had a social interaction, a “1” was marked in the intersection between the row and column of the subjects. At this point, no distinction was made between the sender and the receiver of the communication. Two raters coded the videotapes independently. Their inter-rater reliability was $r=.94$, indicating excellent concordance in coding. The resulting 19 x 19 adjacency matrix was entered into a networking analysis software package.

#### 2.3.3 Assessment of Leaders and Other Influential Persons

Following the experimental procedures on the testbed, subjects then completed questionnaires about their experiences in the interaction with the control force and other group members. One questionnaire asked them to identify which crowd members, if any, they thought had acted as leaders of the crowd. These data were also entered into a matrix that was submitted to social network analysis to identify emergent leaders in the group. These analyses distinguished between the persons doing the nominating and the person they nominated as leader. After completion of the questionnaires and payment vouchers, the subjects were thanked and released.

### 2.4 Social Network Analysis of Crowd Behavior

Matrices derived from questionnaires of pre-existing social bonds, leadership nominations, and coding of social interactions were submitted to analysis by the software ORA Version 1.9.5.4.3 (Carley, Columbus, DeReno, Reminga, & Moon, 2007). Data were rendered into 2D network figures using the ORA visualization function. These visualization techniques were used to generate insights about crowd behaviors, rather than for formal statistical inference. In addition, the data were used to generate tables of sociometric data (density, number of subgroups, isolates, number of linkages among nodes). In the future, these are crowd sociometrics that will be entered into formal analyses.

### 3. Sociometric Measures

#### 3.1 Visualization of Pre-Existing Social Bonds

Fig. 2 shows the visualization of the social bonds. Each subject is denoted as a red node that is labeled with his or her subject number. There exists a preponderance of isolates, that is, people who know no one else in the group, and two sets of dyads (a pair of siblings and a pair of friends).

#### 3.2 Visualization of Social Interactions

Fig. 3 through 5 show the presence or absence of social interactions within the epochs. Each subject is represented as a node in the network, and presence of a communication is represented as a link. Three epochs were visualized: one early, one in the middle, and one toward the end of the experimental session. As the visualization presents, there were considerable changes...
among the patterns of interactions. The early interaction pattern is characterized by a distinct hub, indicating focal persons with whom the rest of the group interacts. The latter interactions tend toward a decrease in isolates and an increased clustering or subgroups that appear to form.

As indicated by these preliminary visualizations, crowds may initially differ significantly in these group level sociometrics in terms of communication linkages. In addition, these crowd metrics change over time. It is possible that over time, or through shared experiences, certain patterns of network structure may emerge.

3.3 Visualization of Leader Nominations

Fig. 6 shows who nominated whom as an informal leader. The nodes in aqua denote those who received no nominations; that is, no subject voted them as leaders. Those nodes who are in red received at least one leader nomination. The origin of arrows directed at each of the red nodes indicates who gave them that nomination. Nodes that are in gold are those who self-nominated themselves as leader; however, they did not receive nominations from any other subject.

As Fig. 6 indicates, Subject 17 received the most nominations as leader. This observation led to an examination of video recordings of his actions. The intent was to gain some insight into what behaviors may have led to his numerous nominations. More specifically, it was a search for differences in overt behavior that distinguished Subject 17 from the rest of the crowd. A sampling of his behavior can be seen in Fig. 7. From the video recording, one can observe that this subject exhibited highly distinct behaviors - he stood apart in front of the crowd facing the control force. Many times he alone charged at the target approaching control force, while laughter could be heard in the background. At times, others followed; at other times, crowd members declined to follow. But, after returning to the safe zone, he again remained relatively aloof from the crowd, despite the crowd directing verbal communications.

3.4 Objective Sociometric

The visualization techniques are valuable in that they allow researchers to draw insights that lead to greater
exploration of the data. In addition, quantitative crowd-level group metrics of social communications can be derived. Number of nodes, linkages, and density (# linkages/# possible linkages) are quantitative crowd-level metrics that can be submitted to statistical analyses and can be used to draw inferences about variables underlying crowd responses to control force tactics. That is, not only can the visual representations give researchers insight into crowd behavior, but the numerical outputs from network analysis can be submitted to formal statistical analyses as either independent variables (initial crowd state) or dependent variables (crowd response).

4. The Way Forward

There are over a dozen laboratories that conduct modeling and simulation of crowds based on theoretical models (Zhou, et al., 2010; Allbeck, 2010; Loftin, Petty, McKenzie, & Gaskins, 2005; McKenzie, et al., 2008; Silverman, 2004). Any subsequent testing is then performed using simulated agents. To our knowledge, this is the first reporting of empirical social network data on crowd behavior that have been collected and analyzed under controlled laboratory conditions. Empirical, in this case, means collection of data on real crowds of real people. With the development of these crowd sociometrics, a wide variety of applied, practical, and tactically relevant questions can now be explored. For example, “What sociometric structures correlate to what patterns of crowd behavior?” Using these data, the network structure and eventually dynamic functions of crowds can be investigated through analyses of behavioral links among the nodes of people.

Identification of these social network structures and functions has a direct benefit for the Soldier. Of particular interest is the network structure relating to behavior propagation through the crowd. That is, given that one person performs a behavior, how likely is it that others in the crowd will follow the behavior? More specifically, if one person runs away from the control force, will others follow suit? If so, which others? Using
network analysis methods, it may be possible to identify patterns of certain categories of sub-groups likely to behave similarly. That is, the methods may provide the ability to differentiate among the probable behaviors of ad-hoc groups of strangers, versus family and friends, versus organized militants.

In the future, network science visualization techniques can be used in threat assessment. One might speculate that the technology could be developed so that sensors that detect location, locomotion, and communication, together with network analysis algorithms could be developed to provide real-time threat assessment in the field. Real-time social network analyses coupled with crowd dynamic analyses could greatly assist the Soldier with this task.

More specifically, the data gathered assist the study of concrete tasks, for example, target selection. Target selection may be thought of as a task of identification of a node/person or subgroup, which is structurally an influential hub within the crowd network (see the central node that is Subject 17). This hub may be considered a likely target in attempts at suppressing crowd approach. The discovery of these distinct behaviors of influential hubs that can be detected through the use of social network analyses is important information that can be used in operational situations involving control force-crowd interactions.

The specific research question for the TBRL is how should crowd composition affect non-lethal weapon (NLW) selection or tactics, techniques, and procedures (TTPs)? Responses to non-lethal weapons may differ between a crowd made up of mostly isolates, such as crowds of people who have no interrelatedness beyond being at the same place at the same time, and crowds who are made up of mostly subgroups, such as families. Questions such as these may now be asked using the sociometrics derived from network analysis.

Because small groups of fewer than 20 persons are being tested, this might be called a focus on local structure. But, this identification may be useful in predicting crowd behavior in more complex, larger gatherings. That is, data from laboratory crowd experiments could be used as input into crowd modeling and simulation investigations. Use of modeling and simulation techniques in the future may greatly expand upon experimental findings from the laboratory.

This work is a piloting effort to develop these methods for further testing on the relationship between social bonds in the crowd and the crowd response to control forces wielding non-lethal weapons and systems. Social network analysis provides measures of critical variables underlying the behavior of crowds. Therefore, they should be further explored for their utility in understanding and predicting crowd behavior. An understanding of methods and metrics of crowd behavior is needed in the development of effectiveness testing of NLWs as well as the TTPs for their use in crowd situations (Department of the Army Headquarters, 2003; Kenny, et al., 2001; Mezzacappa, 2009; Mezzacappa, et al., 2011). The data gathered and the network analyses undertaken will assist in further developing non-lethal options for force protection in crowd situations.

5. Acknowledgements
This work was supported by funding from the Joint Non-Lethal Weapons Program (JNLWP). We would like to thank Lauren Galonski and Erin Hedderich for their assistance on the research. Also, thanks to Dr. Rebecca Jaworski and Martha DeMarco for their assistance on the briefing and manuscript. We would also like to express our gratitude to Dr. Mary Williams at the JNLWP for her continued support of the research at the TBRL.

6. References


**Author Biographies**

**Gordon Cooke, MEME**, is a Principal Investigator at the TBRL. He was also a Chief Engineer, USMA, 11th Armored Cavalry Regiment stationed at the National Training Center, Fort Irwin, CA and served a one-year tour in Mosul, Iraq as the Deputy Division Plans Officer for Multi-Nation Division North-West.

**Robert Demarco, MSBME**, is a Project Lead Engineer and Certified LabVIEW Associate Developer.

**Nasir Jaffery, MBA**, is a quality engineer with an expertise in safety assurance.

**Elizabeth Mezzacappa, PhD**, is a Principal Investigator at the TBRL with expertise in social psychology, stress, and human behavior modeling and simulation issues.

**Gladstone Reid, MSBME**, is a Project Lead Engineer with expertise in experimental automation design and implementation.

**John Riedener, MSSE**, is the Technical Director of the TBRL. He is a Certified Lean/Six Sigma Black belt with a focus on Design of Experiments and Statistical Analysis of Test Results.

**Charles Sheridan, BA**, is a Human Research Technician who has conducted recruitment, screening, running of hundreds of volunteers through studies.

**Kenneth Short, PhD**, is a Principal Investigator in human research on non-lethal weapon effectiveness with expertise in neuroscience.

**Kevin Tevis, MEME** is a Project Lead Engineer with expertise in test bed construction and motion capture.
Group-Based Transactive Memory to implement Computationally-Plausible Social Cognition in Agents

Geoffrey P. Morgan  
Kenneth Joseph  
Kathleen M. Carley  
Carnegie Mellon University  
School of Computer Science  
Institute for Software Research  
5000 Forbes Ave  
Pittsburgh, PA 15213  
gmorgan@cs.cmu.edu, kjoseph@csc.mu.edu, kathleen.carley@cs.cmu.edu  

Keywords:  
Bounded Rationality, Transactive Memory, Social Agents, Network-Centric Simulation  

ABSTRACT:  
There is a need for cognitively bounded implementations of transactive memory for agents. To do this, we use schema theory and tiered social cognition to implement Mead’s Generalized Other (1925). We then compared our new implementation, Construct-ML, with a prior implementation of the same simulation, Construct-O. We were not able to replicate all of the patterns suggested by Construct-O’s results. However, the pattern validity of Construct-ML improves as agents have more cognitive resources, which is suggestive and interesting.

1. Introduction

Social cognition is the ability to encode and retrieve information about other social entities. Humans, our social agents of interest, frequently find it useful to retain knowledge of other actors (social knowledge) as well as knowledge about the world (general knowledge).

Following in the traditions of the Carnegie School (Cyert & March, 1963; March & Simon, 1958; Simon, 1957), these social agents may not be able to access what they know all the time, they may not know what they know, and what they know may be wrong. Similarly, their understanding of other actors is error-prone and perception-based. Regardless, humans use what they think they know about other actors to inform their behavior. If for example, someone needs medical advice, do they consult a doctor or a carpenter? They consult the doctor, naturally, because they have made an inference that doctors tend to have, as a group, knowledge in the medical field. But the same actor would not always seek out the doctor, or if they did they would be ill-served as their house fell down around their ears for want of competent advice! We intend to take advantage of not only this important capability of humans, but also their inferential mechanisms, in this work.

Construct (Carley, 1990, 1991; Carley, Martin, & Hirshman, 2009) is a network-centric agent-based simulation of knowledge diffusion within groups. Agents communicate information to other agents. Agents may forget knowledge they possess. How information diffuses within a group depends on multiple factors: the preferences of individual agents, the initial knowledge of each agent, and the social ties between those agents.

Agents in Construct, like humans, both have knowledge about the world, represented as a knowledge bit array, and knowledge about what other agents know, called Transactive Memory (Wegner, 1995), represented as a per-ego matrix of alters by knowledge. An alter is a potential interaction partner of each individual. In a fully connected system of five actors, there are five \( n \) individuals and twenty-five \( n \times n \) alters; Construct agents may interact with themselves. In earlier iterations of Construct, all agents had representations of all other possible communication partners.

Although in this work we use a very small population to allow for direct model comparison to prior work. Construct has successfully supported hundreds of agents existing in a simulation environment at a single time. However, even though computer hardware has gotten
faster and ever more capable, the growth of computational expense limits earlier versions of Construct from being useful when considering simulations of large populations.

Construct’s computational expense stems from two factors: the size of the knowledge array each agent may possess, and the size of each agent’s transactive memory. Of these two, transactive memory is the dominant term for computational expense. Further, as the size of the agent population increases, the cognitive power ascribed to each agent becomes less plausible. Providing hard limits on the alter list of each agent is tenable, but tends to limit the applicability of the modeling technology to situations where spontaneous link formation is unlikely.

In this paper, we present a method for bounding the cost of transactive memory within Construct by implementing Mead’s Generalized Other (1925). This change improves agent fidelity while capping the costs of transactive memory, allowing many more agents to exist simultaneously within a Construct environment. We suggest that other agent technologies which could take advantage of transactive memory may find our implementation useful and instructive to their own work. We also believe that this change may allow Construct to model many more phenomena than is currently feasible, but we reserve detailed discussion of those possibilities for other work (Joseph, Morgan, Martin, & Carley, 2014). For purposes of clarity to comparisons with older forms of Construct, we refer to the bounded transactive memory version of Construct as Construct-ML.

2. Prior Work
In this section, we describe the related work that has contributed to our approach towards conserving computational resources while also improving the model fidelity of these agents. We conclude with a summary of the extension’s feature and their implications for our modeled individuals.

2.1 Transactive Memory
Individuals often find it valuable to retain an idea of the state of other individuals. We use Wegner’s (1995) description of a networked file system as our narrative. In a networked file system, individual units are assumed to have finite storage capacity. Consequently, information is spread across many of these units, as capacity and demands allow. Units in such a system need some method of accessing information not stored locally in an efficient manner. One solution is that each unit may have information on what other units are likely to be able to access.

It is important, of course, that the depth of knowledge about other agents, which should improve the access of off-board storage, be balanced with the size constraints of that information. A unit must hold some non-trivial amount of local information as well as a representation of what other units may know.

We can think of Transactive Memory’s representation as a three element tuple. Each agent $i$ has for alter $j$ some understanding of the amount of knowledge that alter $j$ has about information set $s$. Previous work has varied representations of this $ijs$ tuple. Palazzolo and his collaborators (2006), represented transactive memory as a single continuous value representing knowledge of set $s$ (which they call topics) for each agent for each alter. In an alternative approach, Carley and Ren (2001) represented each sub-element of a topic $t$ in each agent’s representation of their alters. Thus, an agent’s perception of an alter’s mastery of a particular topic $t$ is a proportion of the elements the agent believes the alter possesses which makes up $t$.

Each of these representations, considered naively, would prove onerous for a cognitively limited human agent. As the size of the population increases, the amount of information required for maintaining transactive memory rapidly dwarfs the amount of direct information present in each networked file system. The principal contribution of this work is a flexible mechanism that allows agents to make educated assessments of alters without needing to store representations of all alters. Our mechanism takes advantage of tiered social cognition, discussed in the next section.

2.2 Tiered Social Cognition
In our discussion of Transactive Memory, we identified that information processing units in an information-rich world need to either keep explicit state about the state of alters or must have the ability to generate useful predictions about the state of those alters. In this section, we discuss one way humans, our information processors of interest, generate useful predictions about the state of alters without needing to keep extensive state on those
alters. They generate these predictions through group affiliation.

Our inspiration for this mechanism rests on work in ethics and social control by Mead (1925). Mead posited that people can make inferential statements on the nature of ethical behavior within their local context of the form, “people of type X tend to do thing Y”. A person can evaluate their own behavior by determining that they are a person of type X and therefore should consider doing Y. Mead called this aggregate of statements the ‘generalized other’, and allowed that many of these inferential statements could exist concurrently within a person’s mind.

These inferential statements may refer to concepts not only of what alters may be able to do, but also to what these alters may know or believe. As Mead (1925, pg. 275) states, “Social control depends, then, upon the degree to which the individuals in society are able to assume the attitudes of the others who are involved with them in common endeavor.” Just as some information processing units may differ on their chosen representations for alters, humans may have more or less nuanced constructs of other humans, and these constructs may include representations of actions, beliefs, and knowledge.

We focus on the last of these objects, and thus can narrow Mead’s statement, to say that “people of type X tend to have knowledge Y”. But how do we define types? Mead suggested that there exists both a large common group, called in his work “society”, but also, independently and concurrently, inferential statements of all groups of which the individual is aware. Thus, we can again transform the structure of the inferential statement to this form “People who are members of Group X tend to have knowledge Y”.

Mead’s argument for social perception is supported by concepts in schema theory (Rumelhart, 1978, 1980). A schema in schema theory is a data structure for representing the generic concepts stored in memory. In schema theory, each individual has a hierarchy of schema that may be applicable to any of various environmental conditions the individual encounters.

The concept can be clarified through examining a specific scenario: confronted with someone examining our neighbor’s wooden porch, we may ask ourselves, “Who is this person?” They may be a professional carpenter repairing the porch, a city inspector checking code, a burglar investigating a prospective target, or various other possibilities. We examine the person’s actions, their apparent attitude, their appearance, their clothes, and use that information, along with relevant historical knowledge, to make an educated guess to answer our own question. In the process of making that guess, we allow whatever knowledge we have that may be applicable to apply. On a social level, we may apply any of three levels of schema to help us answer the question.

- Personal: We know this specific person.
- Group: We don’t know this person, but we can infer that they are members of one or more relevant groups (to us).
- Global: We know this is a person.

We will use that answer to inform future action. We may confront the individual, we may mention it to our neighbor discreetly, or we may do nothing. Schema that help us understand who other people are and what they are likely to know are called “Social Schema” (Kuehne, 1962). These produced social schema are culturally dependent (Little, 1968), but we do not expect that the generative mechanism for these social schema to be culturally dependent.

In schema theory, the availability of schema is determined by environmental cues. Schemas are available if they are relevant. Irrelevant schema do not occupy the individual’s time. Schema-like representations in cognitive agent systems (Anderson, 1996) have found that it is possible for agents to have many schema (implemented as production rules) simultaneously and exhibit human-like cognition as they learn to perform tasks by activating the appropriate rule-sets for the task at hand. Work by Duong and Reilly (1995) used a hierarchy of neural-networks to implement schema theory and model Mead’s Symbolic Interactionism (Mead, 1922), producing a model of racial bias in hiring.

Anderson and his collaborators (2004) suggest, and give empirical evidence, that chunks of our memory are “activated” when they are used, and that this activation decays over time with non-use. We can thus associate schemas agents have with an “activation score”, allowing us to determine their likelihood of use by the agent.
These activation scores, according to Anderson, determine whether or not we are able to recall a chunk or not. If the agent cannot recall the chunk, they must do without it.

Our work takes advantage of the computational tractability suggested by Anderson’s approach, but changes the granularity of the activated chunk. Rather than each chunk representing a single schema-object, of which there be many for a single alter, each chunk represents an alter or a membership group to which an alter can belong.

Thus, if we consider each interaction with an alter to be an “activation”, alters which are frequently contacted will have high activation scores, as will groups to which those members belong or about which we receive information. Group schemas will tend to be sparser but more durable than individual schema.

3. Implementing Social Cognition via Bounded Transactive Memory

In the previous sections, we have discussed transactive memory, schema theory and memory activation and how these elements all play a part in an agent’s social cognition. Through tiered social cognition, we can implement a computationally efficient and cognitively plausible form of transactive memory. Agents who have this capability must be able to be:

- Form expectations about groups, including the Generalized Other
- Revise these expectations
- Generalize about others based on group membership
- Keep track of specific alters of interest
- Revise their expectations of specific alters

In this section, we will discuss our implementation within Construct (Carley, 1991), a validated simulation of information diffusion. We modify the prior transactive memory implementation from Carley and Ren (2001) in three important ways:

- We add transactive memory elements for groups, of similar form to those for alters.
- Transactive memory elements have an activation score, which changes over time as agents interact
- Transactive memory elements may be lost through disuse
- We track the origination time of schemas, which allows for differential treatment of groups and individuals.

As in previous work (Carley & Ren, 2001), we represent a “schema” as a transactive memory vector – a series of $K$ bits, where $K$ represents the number of knowledge pieces, or “facts”, in the system. Each bit represents the ego’s perception of the knowledge of the associated alter, group or generalized other. In the previous work, there was one transactive memory vector for each alter an agent could potentially interact with. In this work, a transactive memory vector may exist for each alter, but also may exist for each group.

These schemas are, as mentioned, arranged hierarchically. An agent determines what an alter knows by starting at the lowest level of the schema hierarchy – the personal level. If that schema is activated above the threshold, then the agent uses this schema to understand the knowledge of the alter. If not, the agent will “construct” the knowledge of the alter based on the groups he is aware the alter is in. If the alter belongs to no groups, then the agent uses his knowledge of what he expects “everyone” to know, which we refer to as his transactive memory of the “generalized other”.

As suggested, our new model still allows agents to determine a value their belief that each alter holds any knowledge set, and therefore trivially captures all five of the behaviors listed that transactive memory systems introduce into simulations. However, our model adds significant functionality in each of these categories, as described below:

- **Forgetting** - Activation equations provide a way for us to directly promote the concept of forgetting – at some point, we actually forget the knowledge of alters and must later reconstruct it
- **Means of Determining Whom to Interact with** - the new implementation now allows us to compute non-zero likelihoods of interaction for *all* agents, as opposed to only those we have specific perceptions of, by considering them as part of a group or the generalized other
- **Specialization** – Specialization can now extend to groups – a member of the “carpenter” group may have the specialized skills of a carpenter, and only be remembered as such
- **Hardening of Opinions** - In a naïve implementation of social cognition, the hardening of opinion was based solely on the fact that agents may not receive
a certain bit when interacting for a long time. In contrast, our implementation allows for a much more robust and cognitively plausible notion of the hardening of opinions – if I label you as a member of group A, it will be difficult for my mind to change that you do not hold all of the attributes of group A – in this instantiation of the model, until you become one of my strong ties.

- **Bounded rationality** - Agents now have an even more limited perception of the knowledge of others, and hence bounded rationality is only increased.

As mentioned, we implement our new model in place of a previous naïve implementation of agent social cognition in Construct, an empirically validated social simulation tool. This allows us to focus solely on our implementation, and avoids a lengthy discussion on the full details of the tool or the model used. For full details on the tool itself, we refer the reader to a useful technical report (Lanham, Joseph, Morgan, & Carley, 2014).

4. Docking with Construct

In this work, we focus on the question of whether Construct-ML is able to replicate results of prior versions of Construct. To do this, we are using a docking analysis (Axtell, Axelrod, Epstein, & Cohen, 1996) to compare Construct-ML with prior versions of Construct. We are choosing to replicate experiments of a previous work that focused on group behavior, documented in Carley and Hill (2001), and referred to then as Construct-O (O is for organizations). The capabilities Construct-O added were later folded back into Construct. When making comparisons, we will refer to the Carley and Hill (2001) iteration as Construct-O, and continue to refer to our extension as Construct-ML.

4.1 Virtual Experiment Methodology

Carley and Hill (2001) introduced to Construct the idea of a second driver of human interaction, the desire for expertise. Earlier iterations of Construct focused on the homophily preference – where agents preferred to interact with people like themselves. Although the homophily drive is more powerful in many social situations, the addition of an expertise preference, a desire to interact with agents with rare knowledge, broadens the applicability of the simulation to include ones where work requires new knowledge to be gained from interaction.

They also wanted to examine the effect of group size on group performance, so their simulations always had two groups. The two groups may be equal or asymmetric sizes.

In short, the Virtual Experiment could be summarized as so:

| Table 1. The Virtual Experiment Design |
| Parameter | Values | # of Values |
| Population Size (PopSize) | 10, 20, 30 | 3 |
| Expertise Drive Weight* | 0.0, 0.25, 0.5, 0.75, 1.0 | 5 |
| Group Size | Undifferentiated, Differentiated | 2 |
| Individual Memory Threshold** | 0, -1, -2 | 3 |
| Knowledge Size | 2 x PopSize | 1 |
| Knowledge Assignment | Random | 1 |
| Groups | 2 | 1 |
| Group Membership | One group | 1 |

| Total Combinations Original Experiment | 30 |
| Total Combinations This Experiment | 90 |

*In this experiment, Expertise and Homophily preferences sum to 1.
** This parameter was added to allow variation in cognitive resources available to agents.

Population size (PopSize) is the number of actors in the simulation. Expertise Drive Weight indicates the relative weighting of homophily and the expertise drive in agents. Group Size is whether the groups are of equal or unequal sizes. In the Carley and Hill (2001) experiments, the amount of knowledge was always scaled to the population size; there were always two groups; each agent was a member of only one group; and knowledge was randomly assigned.

We have taken this basic experimental structure and also manipulated one parameter related to our new transactive memory implementation, individual memory threshold, how long individuals and groups remain in memory. The range of values provided (0, -1, and -2) suggest a wide range of cognitive resources available to agents.
In this experiment, and sympathetic to what was originally done in Carley and Hill (2001), we paid attention to the following outcome variables over the course of the simulation:

- **Knowledge Diffusion** - Of all available information, how much has been distributed to all agents. Mathematically, the sum of all binarized ties in the Agent x Knowledge matrix divided by the total cells in this matrix.

- **Task Performance** - Each group performs a binary classification task 50 times each turn based on a random sampling of 50 knowledge bits (note that bits may, and often will be, represented multiple times in the classification task). Each member votes and majority rules. Group accuracy is reported and averaged for overall performance.

- **Task Consensus** - Based on the same binary classification task, the number of members that agree with the group’s decision for each task is recorded and averaged.

- **Triad Count** - Given node A, B, and C, a triad exists if the probability of interaction between A & B, B & C, and A & C, *in either direction*, is above the average probability of interaction calculated across all actors.

These values are calculated at each time-point. Carley and Hill (2001) focused on the length of time required for each outcome metric to reach 90% of the achieved maximum, as these values are much less sensitive to random noise than the time to reach Maximum.

Carley and Hill (2001)’s results suggested that in Construct-O:

- Small groups reach benchmarks faster
- That time to reach 90% of maximum tends to follow this pattern for outcomes, 1. Diffusion, 2. Performance, 3. Consensus, and 4. Triad development
- That increasing the weight of expertise in the agent’s drives tends to reduce these times, except when expertise weighting is 100% (which vastly increases these times)

We will use these findings to inform a pattern-level analysis of Construct-ML, described in the next section.

### 4.2 Docking Results

In these results, we are investigating the relational, or pattern, validity of Construct-ML to Construct-O. Thus, although we report the actual metrics, the material at interest is the pattern match for each outcome based on Construct-O’s findings.

We have three findings from Construct-O we wanted to investigate. We first explore the issue of group size, with our time to reach 90% for each of our four outcome variables. Table 2 includes pattern match values on group size and outcome patterns.

**Table 2. Population and Outcome Metrics**

<table>
<thead>
<tr>
<th>Threshold = 0</th>
<th>Pop. Size</th>
<th>Group Size Pattern Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion</td>
<td>94</td>
<td>153</td>
</tr>
<tr>
<td>Performance</td>
<td>40</td>
<td>31</td>
</tr>
<tr>
<td>Consensus</td>
<td>20</td>
<td>72</td>
</tr>
<tr>
<td>Triad</td>
<td>122</td>
<td>113</td>
</tr>
<tr>
<td>Outcome Pattern Match</td>
<td>3/6</td>
<td>3/6</td>
</tr>
</tbody>
</table>

| Threshold = -1 |
|----------------|-----------|---------------------------|
| Diffusion      | 80        | 160                       | 236                       | 3/3 |
| Performance    | 34        | 32                        | 28                        | 0/3 |
| Consensus      | 19        | 80                        | 107                       | 3/3 |
| Triad          | 269       | 137                       | 162                       | 1/3 |
| Outcome Pattern Match | 3/6 | 3/6 | 3/6 | 16/30 |

| Threshold = -2 |
|----------------|-----------|---------------------------|
| Diffusion      | 76        | 170                       | 256                       | 3/3 |
| Performance    | 33        | 39                        | 32                        | 1/3 |
| Consensus      | 19        | 84                        | 126                       | 3/3 |
| Triad          | 273       | 286                       | 326                       | 3/3 |
| Outcome Pattern Match | 3/6 | 4/6 | 4/6 | 21/30 |

We represent this table graphically in Figure 1, next page.
Figure 1. Average Turns to reach 90% of Maximum for each Outcome Metric. Lines colored by Threshold, Group Size along the X-Axis.

Note that Construct-ML agents have more cognitive bounds than Construct-O agents. Pattern validity to Construct-O, in general, improves as the agents’ social space and cognitive resources increases.

Our simulation with the settings as given here matches Construct-O’s outcome pattern in relationship to size for the outcomes of Diffusion and Consensus Formation, but not for Performance and Triadic closure.

For performance, this may be because of the implicit parameters built into the task performance evaluation (50 tasks, with a task of size 50) do not well match the original settings, which are not given in the original work. We will investigate the impact of task size on the performance outcome in future work.

The triad outcome is more interesting in that it does not appear to be obviously arbitrary in relation to settings given. We are still investigating the implications behind this triad stability pattern.

Because knowledge informs the stereotypes developed for each group, it’s possible the random nature of knowledge assignment affects these simulations in ways that the prior implementation did not. If pattern validity improves with group-based knowledge assignment, then that will suggest that the model has, in one sense, improved, as random knowledge assignment is not very realistic!

The final outcome we wanted to compare against Construct-O was the impact of interaction drive.

Table 3. Expertise and Outcome Variables, averaged across threshold settings

<table>
<thead>
<tr>
<th>Expertise</th>
<th>0</th>
<th>0.25</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion</td>
<td>170</td>
<td>165</td>
<td>160</td>
<td>164</td>
<td>123</td>
</tr>
<tr>
<td>Performance</td>
<td>37</td>
<td>36</td>
<td>34</td>
<td>37</td>
<td>18</td>
</tr>
<tr>
<td>Consensus</td>
<td>77</td>
<td>75</td>
<td>75</td>
<td>79</td>
<td>30</td>
</tr>
<tr>
<td>Triad</td>
<td>231</td>
<td>224</td>
<td>199</td>
<td>210</td>
<td>153</td>
</tr>
</tbody>
</table>

Again, we do not see full pattern validity to Construct-O. Although we find in Construct-ML, as we did in Construct-O, that more expertise drive weight tends to decrease times to reach stability, we do not find, as Construct-O did, that agents who are only concerned with expertise perform less effectively, instead there is often a large drop in the amount of time required to get to the 90% benchmark. This may be due to an interaction of drive-weight and the transactive memory of groups that was not previously modeled.

Although these results suggest that Construct-ML will not predict similar group outcomes as Construct-O – it does not say that these new outcomes may not be, in practice, more realistic. Comparison to human small-group data is clearly indicated.

5. Conclusions and Future Work

In this paper we have discussed the need for a cognitively bounded implementation of transactive memory for agents. We have described the theory behind our approach, and have discussed the requirements of such a system. We then compared the findings of our model with an earlier iteration of Construct-O, and we were not able to replicate all of the patterns suggested by Construct-O’s results. However, the pattern validity of Construct-ML improves as agents have more cognitive resources, which is suggestive and interesting.

Further work will involve systematic exploration of the robustness of these findings using other, more realistic, knowledge assignment procedures, and also changing the size of the binary classification task vector.
We are happy to discuss implementation details of the transactive memory system described in this work, but have refrained for reasons of space.

References

Author Biographies
GEOFFREY P MORGAN is a PhD student in Carnegie Mellon’s Computation, Organization, and Society program. Morgan has experience in developing autonomous robotic systems, self-learning systems, cognitive models, and user-assistive systems. He is interested in organizational impacts on individual decision-making and on optimizing organizational performance.

KENNETH JOSEPH is a 5th year Ph. D. student in the Computation, Organization and Society program in the School of Computer Science at Carnegie Mellon University. He is interested in a better understanding of the dynamics and cognitive representations of prejudice and its interrelationships with sociocultural structure. In his work, he utilizes both machine learning and agent-based modeling approaches.

KATHLEEN M CARLEY is a Professor and the director of the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. Her research combines cognitive science, social networks and computer science to address complex social and organizational problems.
Role of Information Asymmetry in a Public Goods Game for Climate Change

Medha Kumar¹, Rohit Chouhan¹ ² and Varun Dutt¹

¹Applied Cognitive Science Lab, Indian Institute of Technology, Mandi, India
²Centre for Converging Technologies, University of Rajasthan, Jaipur, India

medha751@yahoo.in, rc655108@gmail.com, varun@iitmandi.ac.in

Keywords:
Climate Change, Public Goods Game, Public Fund, Private Fund, Information Availability, Game theory

ABSTRACT: Atmospheric carbon-dioxide concentration and global temperatures are increasing at an alarming rate. Global cooperation is much needed; however, little is currently known on how information asymmetry among world players influences emergence of cooperation. In this paper, we study the role of information asymmetry about monetary investments among human players using a modified form of a repeated public goods game for Earth’s climate (called, “climate game”). In the climate game, a group of four human players (representing different world economies) play repeatedly against each other where in each round a player decides how much money to contribute to a green fund (the money not invested in the green fund accrues interest as private investment for a player). In an experiment, 6 groups participated in the game across two between-subjects conditions: Info-NoInfo and NoInfo-Info. In Info-NoInfo condition, 3 groups first played the climate game for 50 rounds where information on opponents’ contribution to green fund was known to all players. This play was followed by a game where information on opponents’ contribution to green fund was not known to players. This information presentation was reversed in the NoInfo-Info condition. Results reveal that contribution to the green fund decreased rapidly when players had information on their opponents’ monetary contributions compared to when players did not possess this information. Also, experiencing information in the first game decreased players’ contributions to the green fund in the subsequent game. We discuss implications of our findings for emergence of cooperation against climate change.

1. INTRODUCTION

Climate change has currently become a worldwide phenomenon with disastrous consequences like melting of ice-caps and sea-level rise (Pachauri, 2007). Each year, all countries meet in the Conference of Parties (COP) meetings to discuss and negotiate monetary investments against climate change (UNFCCC, 2014). Although the COPs are a regular event, these negotiations have not resulted in concrete measures against climate change. For example, at COPs there is always speculation among negotiating parties on how much a country or a block of countries is willing to contribute in public funds to avert climate change (Dutt, 2014; 2015). An important aspect of this speculation is how much information a country has about the binding promises of monetary contribution of other countries to public
funds (Dutt, 2014). According to Dutt (2014), sometimes, countries know about the promises of their opponents in advance and sometimes these promises are speculative and hidden from the public.

Although negotiations are an important part of COPs in the real world, very little research has taken place to understand the resulting negotiation behavior in the controlled laboratory environment. Furthermore, prior research has seldom investigated how information asymmetries about contributions made among players influences cooperation against climate change. In this paper, we use a modified form of the classic public goods game (Axelrod, 1997), called “climate game”, in order to study these two goals.

In the climate game, just like in the public goods game, players play a game with each other by making yearly monetary investments to a private (personal) fund and a public (green) fund. The end point of the game is not known to the players. The money put in green fund is multiplied by a factor and the return generated is equally divided among all players. In contrast, the money not invested in the green fund is put in a private fund and it earns interest at a constant rate of return. The game takes into account four players, which represent four economic blocks in the world namely high income, middle income, upper-middle income and low income economies. The goal of each player in the game is to earn as much money as possible by making investments in the private and green funds.

Information asymmetries can be created in the climate game by manipulating the information available to players about investments made by their opponents in each round. For example, Tavoni et al. (2011) have manipulated information in a public goods game for climate change, where players decided whether to contribute €0, €2, or €4 to a climate account (a form or green fund) in each of the 10 rounds. After each round, a player was provided information on individual contributions made by other players as well as the aggregate contribution of the group. In the information manipulation condition it was found that contributions decline in presence of availability of other player’s contribution as compared to the absence of information.

Similarly, Gonzalez et al. (2014) have reported influence of information asymmetry in decisions making in the Prisoner’s Dilemma (PD) game (Rapoport & Chammah, 1965). In this study, as information about opponent actions increased in a repeated PD game, the amount of cooperation across repetitions of play increased as well.

Although prior research has manipulated information availability in abstract 2x2 games (Gonzalez et al., 2014), research that manipulates information availability in an applied domain (e.g., a climate context) is scarce. In this paper, we overcome this scarcity by evaluating the effects of information manipulation in an applied climate-change context using the climate game. For this purpose, we designed the climate game for two scenarios: One with information about monetary contributions is available to opponents and the other, where such information is not available to opponents. We use these two scenarios in a human experiment reported ahead in this paper.

In what follows, first we illustrate the model that is used for developing the climate game. Next, we report an experiment where we manipulated information asymmetry among human players playing the climate game. We report the results from our experiment and highlight the implications of our findings for negotiation behavior against climate change.

2. A MODEL FOR CLIMATE CHANGE

The climate game contains four players, which represent four economic blocks, namely, high income, middle income, upper-middle income and low income economies. This classification of economic blocks into four categories is based upon World Bank data (World Bank, 2015). The payoff for each of these players is defined by the following equation:

\[
\pi_i = \sum_{t=1}^{10} \frac{1}{4} \left[ k_1 \times C_1 + k_2 \times C_2 + k_3 \times C_3 + k_4 \times C_4 \right] 
\]

where \( i = 1, 2, 3 \), and 4 for the four different economic blocks respectively; \( t \) = rounds from 1 to 50; \( e \), is yearly endowment (income) given in form of
GDP; $C_1$, $C_2$, $C_3$ and $C_4$ are contributions of 4 players to their respective green fund in each round $t$; $k_1$, $k_2$, $k_3$ and $k_4$ are the return on investments on the amounts contributed to green fund by players 1, 2, 3 and 4; $k_i'$ is the investment contributed to private fund by different players; and, $\pi_i$ is the payoff for different players.

Table 1. Values of return on investments on the amounts contributed to the green fund by respective players and investment contributed to private fund by high, upper middle, middle and low income economies respectively.

<table>
<thead>
<tr>
<th>Coefficients for the return amounts contributed to green fund</th>
<th>Values of the return amounts contributed to green fund</th>
<th>Coefficients for the amount contributed to private fund</th>
<th>Values of the amount contributed to private fund</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>2</td>
<td>$k_1'$</td>
<td>1.05</td>
</tr>
<tr>
<td>$k_2$</td>
<td>1</td>
<td>$k_2'$</td>
<td>1.02</td>
</tr>
<tr>
<td>$k_3$</td>
<td>0.8</td>
<td>$k_3'$</td>
<td>1.06</td>
</tr>
<tr>
<td>$k_4$</td>
<td>0.2</td>
<td>$k_4'$</td>
<td>1.03</td>
</tr>
</tbody>
</table>

The different $k$ values contributed to green fund is derived based on social dilemma condition (Hoven, 2013)

$$1 < \frac{\sum_{i=1}^{4} k_i}{k_i} < 4(2)$$

The different $k'$ values are obtained from the saving bank interest rates of four economies is based on World Bank data (World Bank, 2015). Social dilemmas are situations in which an individual profits from selfishness unless everyone chooses the selfish alternative, in which case the whole group loses. It is a challenging situation as acting in one’s immediate self-interest is tempting to everyone involved, even though everybody benefits from acting in the longer-term collective interest.

3. HYPOTHESIS

In the information condition (Info), players are aware of the contributions made by their opponents to the green fund after each round of play. While in the no information condition (NoInfo), players do not know about their opponents’ contributions to the green fund. Logically, hypothesis H1 states that in the Info condition, players’ contribution will be less than that in the NoInfo condition. This happens because in the Info condition, participants can see how much others are contributing to the green fund and this knowledge causes them to stop or reduce their contribution to the green fund, overtime; however, such knowledge is not there for the participants in the NoInfo condition and thus they do not decrease their contribution to green fund as in the information condition do. Also, this expectation is supported by the results by Tavoni et al. (2011). Hypothesis H2 states that a first exposure to the Info condition will decrease participants’ contributions to green fund in a subsequent condition more than a first exposure to the NoInfo condition. This expectation is due to the presence of information in the Info condition and that people show learning effects. In general, Info condition causes people to stop contributing over rounds of play as compared to NoInfo condition. Thus, when people have a prior exposure to NoInfo condition, they tend not to decrease their contribution rapidly; rather, the contribution (though less) remains constant across rounds. Furthermore, when people have prior exposure to the Info condition, they tend to sharply decrease their contributions due to the presence of information.

4. METHOD

Experimental design: In the experiment, 24 participants were divided into 6 groups, where each group consisted of 4 participants. The climate game was played across two between-subjects scenarios: Info-NoInfo and NoInfo-Info. Out of 6 groups, 3 randomly selected groups played Info-NoInfo scenario in which the Info condition was played first and it was followed by play in the NoInfo condition. In the Info condition, contributions to the green fund were known to all players in each round; whereas, in NoInfo condition, contributions to the green fund were not known to players across all rounds. Similarly, three randomly selected groups played NoInfo-Info scenario, where the order of presentation of the Info and NoInfo conditions was reversed. As shown in Figure 1, in all conditions, players received certain yearly income and they had to decide as to
how much of that income players want to invest in the green fund for averting climate change. The amount not invested in the green fund is players’ private income. This private income could be invested in a bank account and this income may accrue certain simple interest.

Figure 1. The game uses a fictitious currency EC. Each player gets yearly income of 10 million EC. All players decide to invest a part of this income (X, X’, X'', or X''' million EC) in a green fund for averting climate change. The remaining money is stored in a bank account and it accrues a simple interest. The money contributed by all players in the green fund is multiplied by certain multipliers and then summed together. This sum is then equally divided among all players. Each player payoff at the end of a round is the sum of the return on investments obtains from the bank and the green fund.

Total payoff at the end of each round is the sum of return on investments from the green fund and that from the money invested in private fund. The goal is to maximize total payoff in the climate game by making investments in the green and private funds. Both Info and NoInfo conditions of climate game in each of the two scenarios were run for 50 rounds.

Participants: Twenty-four graduate and undergraduate students from diverse fields of study participated in this experiment, comprising 14 females and 10 males. Ages of the participants range from 18 to 44 years. In self-reports, 56% of participants indicated having heard of public goods game through television, websites, newspapers, magazines or some other means. Also, 90% of participants reported they either completed or are currently pursuing degrees in science, technology, engineering, and management (STEM). All participants received a base pay of Rs 10. The participants could earn an additional maximum bonus of Rs 20, based on their performance in the climate game. Players received base pay of Rs 10 for participating in the experiment and players total payoff at the end of the game was be converted into real money in the following ratio: 1,000 million EC payoff in the game = 10 INR in real money. At the end of the experiment, players’ total payoff in million EC in both scenarios was converted into INR and this money was paid to players in addition to the base payment.

Procedure: Participants were given instructions before they were made to play the climate game. The instructions were given online before the study began. Each participant filled their consent form, demographics information and then went through the online instructions. As part of the instructions, participants were shown an image of what would happen in climate game (Figure 1) and how they may contribute to the two funds from the endowments they received in each round. Once participants acknowledged that they understood the game and the task requirements, they were allowed to interact with the climate game on computer terminals.

5. RESULTS

5.1 Payoff differences between Information and No-Information Conditions (H1)

First, we tested hypothesis H1. Figure 2 shows average contribution to the green fund across 50 rounds in both versions of game, Info and NoInfo. The lines for each condition in Figure 2 are averaged across 24 participants across both scenarios. In both conditions, contribution to the green fund reduced rapidly. Although the contributions to the green fund were higher in the Info condition compared to the NoInfo condition, this pattern reversed with increasing number of rounds. The average contribution in Info condition was lower than that in the NoInfo condition between rounds 25 and 50 (NoInfo = 2.84 > Info = 2.50). These results support hypothesis H1.
5.2 Learning Effects of Information and No-Information Conditions (H2)

Next, we tested hypothesis H2. For this purpose, we compared average contributions to the green fund in the NoInfo condition played before and after the Info condition (i.e., effect of information; see Figure 3). Also, we compared the average contribution to the green fund in Info condition played before and after the NoInfo condition (i.e., effect of no-information; see Figure 4).

![Figure 3](image3.png)

**Figure 3.** Average contributions to the green fund in no-information condition played before and after information condition across 50 rounds.

As shown in Figure 3, playing Info condition first had an impact on participants’ performances in the subsequent NoInfo condition, namely, a decrease in contributions to the public fund. As shown in figure, contribution to green fund reduced rapidly for the participants who played Info condition before the NoInfo condition. Thus, when people have prior exposure to the Info condition, they tend to decrease their contribution to the green fund sharply in the subsequent NoInfo condition. However, the effect reverses if participants play the NoInfo condition first. As shown in Figure 4, the contribution to the green fund was not reduced rapidly in the subsequent Info condition when participants played NoInfo condition first. Thus, when people have no prior exposure to opponent information, they tend not to decrease their contribution to the green fund across rounds in the information condition. Furthermore, Figure 4 also shows that participants who played NoInfo condition first contribute lesser amounts to the green fund in the subsequent Info condition compared to participants who did not play the NoInfo condition first. This latter effect is attributed to the absence of information about opponent contributions in the NoInfo condition.

6. DISCUSSION AND CONCLUSION

Our study used an experiment with human players to test investment decisions towards averting climate change in the presence and absence of information about opponent contributions. At first we tested average contribution to the green fund across 50 rounds in both versions of game i.e. information and no information. In both conditions, contribution to
the green fund reduced rapidly. Although the contributions to the green fund were higher in the information condition compared to the no information condition, this pattern reversed with increasing number of rounds. Then, we tested average contributions to the green fund in no information condition, which was played before and after the information conditions across 50 rounds (i.e., the effect of information). It was observed that when people have a prior exposure to opponent payoff information, they tend to decrease their contribution rapidly. Finally, we tested the average contributions to the green fund in information condition played before and after the no-information condition across 50 rounds (i.e., the effect of information). Thus, when people have no prior exposure to opponent information, they tend not to decrease their contribution to the green fund sharply in the following information condition.

In the information condition, participants can see how much others are contributing to the green fund and this knowledge causes them to stop or reduce their contribution to the green fund, overtime; however, such knowledge is not there for the participants in the no information condition and thus they do not decrease their contribution to the green fund as in the information condition. As a result, the no-information condition produces higher payoffs to the green fund compared to the information condition. Thus, results, in particular after the 25th round, are as per our expectation in H1. Results before the 25th round could be attributed to participants learning the game dynamics.

Furthermore, according to our hypothesis H2 we find that people show learning effects. In general, information condition causes people to stop contributing over rounds of play and no information condition does not. Thus, when people have no prior exposure to information, they tend not to decrease their contribution rapidly; rather, the contribution (through less) remains constant across rounds. Furthermore, when people have prior exposure to information, they tend to more sharply decrease their contributions due to the presence of information itself.

This study is the start of a large research program that investigates how public contributions for averting climate change is influenced by a number of ecologically valid factors. In the immediate future, we would like to extend this game by including stochastic losses due to climate change as part of a player’s payoff.

7. ACKNOWLEDGEMENT

The authors acknowledge the support of the Applied Cognitive Science Laboratory, Indian Institute of Technology, Mandi. Also, the authors would like to thank the Indian Institute of Technology, Mandi and Department of Science and Technology, Indian Government for providing funding on this project (# IITM/SG/VD/32 and IITM/SG/VD/64).

REFERENCES


**AUTHOR BIOGRAPHIES**

**MEDHA KUMAR** is a PhD. Scholar in the School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi. Her research interests include Image Processing, Game Theory, Social Computing and Climate Change.

**ROHIT CHOUHAN** is M.Tech student at the Centre for Converging Technologies, University of Rajasthan, Jaipur. His research interests include Computer Networks and Game Theory.

**Dr. VARUN DUTT** is an Assistant Professor in the School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi. His research interests include Artificial Intelligence, Human Computer Interaction, Environmental Decision Making, and Judgment and Decision Making.
Cognitive and Probabilistic Models of Group Decision Making

Yuqing Tang\(^1\), Christian Lebiere\(^1\), Katia Sycara\(^1\), Don Morrison\(^1\) and Paul Smart\(^2\)

\(^1\)Carnegie Mellon University, USA
yuqing.tang@cs.cmu.edu, cl@cmu.edu, katia@cs.cmu.edu, dfm2@cmu.edu
\(^2\)University of Southampton, UK
ps02v@ecs.soton.ac.uk

Keywords:
Cognitive Modeling, ACT-R, Argumentation, Group Decision Making

ABSTRACT: We introduce an experiment designed to study trade-offs in collaborative decision making environments such as the ability to accumulate information and its impact on the fluctuation of decisions. Two models of the experiment are presented: a cognitive model using the ACT-R cognitive architecture and a probabilistic argumentation model using Markov Random Fields. Representative results from the experiment are presented and compared to the results of the two models. Implications of the results and avenues for future work are discussed.

1. Introduction

Decision making in distributed environments has become a ubiquitous part of our environment. Collaborative networked environments range from Google Docs to elaborate military command and control centers. The design of such environments is far from trivial: while more information is generally better, too much information can also be detrimental by overwhelming its users. Given various cognitive and attentional bottlenecks, decision makers face a fundamental trade-off in interacting with this type of environment. One could attempt to exchange as much information as possible with partners on the collaborative network but the obvious gains are limited by two factors: the limitations of our perceptual-motor capabilities, i.e., how fast we can enter information into the network (e.g., by typing) and parse information received from the network (e.g., by sorting through chat messages), and the limitations on retaining information gleaned externally (e.g., forgetting messages read earlier). Conversely, one could attempt to focus on one’s own experiences, making the most of them by rehearsing them and performing as many inferences as possible, but at the potential cost of neglecting crucial information available externally. The effectiveness of a focus on internal (i.e., personal experience and memory) vs. external (i.e., group experiences shared through the environment) sources of information fundamentally depends upon the precise quantitative nature of our cognitive architecture, the statistical nature of the external environment, and the organizational structure of the information-sharing tool (Reitter & Lebiere, 2012).

In this paper, we assume some degree of information sharing through the information environment and focus on a related dilemma: what level of information to share between decision makers. One possibility is to share detailed information, making sure that all decision makers have all potentially useful information, at the possible cost of overwhelming them with irrelevant details. The alternative is to share a high-level, refined version of the information available, hoping to maximize the utility of the exchange while minimizing the perceptual-motor and attentional costs. The difficulty of that trade-off is that one does not always (or even most of the time) know \textit{a priori} what is likely to be of interest to another decision maker. The only way to know in general would be to have access to all their information, creating a Catch-22 situation.

In the rest of the paper, we present the design of an experiment intended to address the issue, specifically by allowing decision-makers to share detailed facts or a high-level guess regarding a variety of questions that can be answered using the facts. We then present two computational models of decision-making for that experiment that are intended to quantitatively address the tradeoffs described. We then present an analysis of experimental results as well as preliminary results from the model simulations on the accumulation of information and its impact on the fluctuation of decisions. Finally, we draw some parallels between the two models and discuss some potential extensions of the work.

2. Experiment

The task used simulates, using textual information, an artificial world region with political unrest. It requires four cooperating subjects to discover a variety of details and draw conclusions regarding an impending terrorist attack. The data underlying this task are from ELICIT (Chan & Adali, 2012). Facts\(^1\) from which these conclusions can be drawn are released to the subjects in stages over time. Each fact is given to only one subject, each subject receiving facts

\(^1\)ELICIT calls these statements “factoids”
disjoint from those given to other subjects, and the subjects must decide which facts to forward to which other subjects. For each trial, 68 facts are distributed in three waves. Each wave contains roughly ⅓ of the 68 facts. Between two consecutive waves, the subjects have 5 minutes to process a wave of new facts. At the end of a trial, the subject must submit their best conclusions from the facts collected, 15 minutes after starting. In addition to running experiments with four human subjects, automated subjects (bots) were also implemented, and data were collected for single subjects playing with three bots, though without the human subjects knowing their teammates were automated.

Each of the four subjects is asked to answer a different question about the attack: who, where, what and when. For any group of subjects these four questions are distributed once to each subject on four different trials. The first is a training trial, the results of which are not used, followed by three experimental trials. The answer to the who question is the name of the group expected to conduct the attack; group names are colors, such as the “gold group” or the “violet group.” The answer to the where question is a country name; country names are derived from Greek letter names, such as “Chiland” or “Omegaland.” The answer to the when question is a kind of target, such as “embassy” or “military base.” The answer to the when question has a four-fold structure, consisting of month name, day of the month, hour on a twelve hour clock, and “AM” or “PM.” While not the subject of any of the questions, there are also individuals, who serve as links connecting some of the facts presented to subjects; individuals are named after animals, such as “the Lion” or “the Jackal.”

The facts delivered to the subjects are sentences. Some are simple and immediately useful, such as “The attack will be at 11:00.” Though even this fact is delivered to the “where” subject, and so must be forwarded by that subject to the “when” subject. Others are more complex, and must be combined with other information to be useful; for example, “The Azure and Brown groups prefer to attack at night,” or “The Lion is known to work only with the Azure, Brown, or Violet groups.” Some of the facts delivered are essential for constructing correct answers, others are helpful but not essential, and still others are mere noise, contributing nothing to correct answers.

The four subjects interact with the system and with each other through a web-based user interface, Figure 1, implemented with HTML and JavaScript. This interface is divided into several panes. One, on the right, summarizes the player’s current role (who, where, what or when), describes the names and roles of the other players, and allows access to the instructions for reference.

The most prominent pane of the interface is the inbox, to which new facts are delivered. These may be new facts, delivered by the system; or they may be facts forwarded by another subject. Facts are normally displayed here in a partially obscured form, with only a few keywords, such as “Yellow,” “Magenta” and “Green,” legible, the rest of the text being replaced with ellipses. The user can click on a fact to cause the full text to be presented. When the mouse pointer is moved off the fact it is partially obscured again; by recording the users’ mouse actions insight into the the users’ attention can be gleaned. Below the inbox is a pane multiplexed for three purposes: outbox, mylist and guess-box. When used as the outbox facts can be dragged to it, and forwarded to other subjects, in whose inbox they will appear. When used as mylist, facts can dragged to it for future reference; while users can use this for whatever purpose they choose, it is expected that those who do employ it will use it to consolidate facts they suspect are important for answering their own question. Facts in mylist, as in the inbox, are normally partially obscured, and must be clicked to be read in full. At several points in each round subjects are asked to make their best guess so far at the question they have to answer, along with their confidence in the guess, on a five-point scale. In this way, we can trace human subjects’ behaviors on accumulating facts and its impact on the fluctuation of decisions. These decision traces are then compared with the traces produced by the ACT-R model and the probabilistic argumentation model.

3. Models

Two different computational models of this task were implemented, and their results compared to the human data. The reason for using two different modeling paradigms is to study what each can contribute to understanding group decision making and draw lessons from any parallels or differences between models. (Lebiere, Gonzalez, & Warwick, 2009)

3.1. ACT-R Cognitive Model

The ACT-R model uses the ACT-R cognitive architecture (Anderson & Lebiere, 1998) and follows the instance-based learning (IBL) modeling methodology (Gonzalez, Lerch, & Lebiere, 2003). To provide for finer discrimination in judgment and ensure the ability to gradually accumulate evidence from a stream of individual facts, the basic problem of determining the most likely candidate answer for each question is formulated as a goal to assign a probability to each potential answer. The goal is defined as a chunk of type hypothesis that contains three slots:

- Question: the representation of the question, i.e., who, what, where and when
- Answer: the representation of each possible answer, e.g., various groups for who
• Probability: a probability value assigned to the question-answer pair

This representation follows the general IBL pattern of context (question), decision (answer) and outcome (probability). In keeping with the instance-based methodology, this representation is used both for facts as well as goals. Specifically, most facts are transformed into chunks of this type if they make a strong assertion about a given question. For instance, if the fact rules out a particular group, a hypothesis chunk will be created (or reinforced if it already exists) stating (who, group, 0). Conversely, if it strongly implies a group’s involvement, the chunk (who, group, 100) will be created. If the fact mentions the possible involvement of \( n \) groups, then a separate hypothesis chunk is created for each group with a probability of \( 1/n \), reflecting mutually exclusive participation.

Of course, those assertions are not literally correct—rather the intent is to provide the basis for a rough estimate of relative probabilities based on the information provided. More precise facts (e.g., stating actual probabilities, or using qualifiers such as likely or probably) could be used to create more accurate chunk encodings. When the model is asked to generate a guess to a question, it iterates through all the possible answers (e.g., all the groups for a who question) and generates a probability estimate for each using the blending mechanism used for memory retrievals (Lebiere, 1999). During memory retrievals, each chunk in memory has an activation that reflects factors such as recency, frequency, and degree of match to the requested pattern. Recency is factored through a power law decay from the time that the chunk is created. Frequency reflects a power law of practice of the numbers of times that a chunk is strengthened following rehearsals. For degree of match, we assume for simplicity that each question and answer are distinct and no similarities are defined. Blending retrieval then assigns for a given question-answer pair a probability to each chunk matching that request (in general, there will be several) reflecting a softmax (Boltzmann) distribution of chunk activations given a certain amount of noise. Those probability estimates for each chunk associated with the question-answer pair are then blended according to a weighted average of the chunk probabilities (assuming linear similarities over the probability space (Lebiere et al., 2013)). The probability estimates are not normalized but instead the largest one is selected to generate the guess. All parameters controlling the behavior of the model are left at their default values: the base-level decay rate is 0.5, the mismatch penalty is 2.5, the activation noise is 0.25, and the blending temperature is 0.4.

Note that, as mandated by the ACT-R theory, the hypothesis goals generated to provide the guess become themselves chunks in memory, as are guesses received from other agents. This can give rise to cognitive biases such as confirmation bias, where a strong initial estimate leads to overoptimistic estimates later despite contradictory evidence.
3.2. Probabilistic argumentation model

We developed the Markov Argumentation Random Field (MARF) (Tang, Toniolo, Sycara, & Oren, 2014), which is a combination of formal theory of human reasoning in argumentation and Markov random fields. The formal theory of argumentation (Dung, 1995) formalizes the essentials of human reasoning about inconsistent, uncertain and incomplete information in the course of argumentative dialogues. However, in real world scenarios deviation from the formal theory is unavoidable. MARF is a probabilistic model which carries out real world reasoning after the formal theory of human reasoning while at the same time being flexible to accommodate the deviations from the theory. Unlike the ACT-R model which focuses on revealing the cognitive process of human reasoning, MARF follows the knowledge engineering path aiming at reaching correct reasoning as much as possible.

A Markov random field (Koller & Friedman, 2009) is a graphical model which encodes local Markov properties — a random variable is independent of all other variables given its neighbors — as an undirected graph to establish probabilities of all valuations to the variables. Echoing the local Markov properties, Dung’s argumentation semantics (Dung, 1995) can be recovered by applying a list of acceptability rules based on a graphical model of argument interaction. For example, “A is labeled IN (accepted) if all its attackers are labeled OUT (rejected)” (Caminada & Gabbay, 2009). Such rules, which assign acceptability to an argument given the status of its neighbors, also satisfy local Markov properties. Moreover, the construction of arguments as proof networks (Tang, Cai, McBurney, Sklar, & Parsons, 2011) also admit the local Markov properties — the establishment of a conclusion is independent of all other rules given the premises of the rules for the conclusion. These two observations allow us to construct Markov Argumentation Random Fields (MARF).

MARF compiles the argumentative knowledge and received information into a mathematically rigid Markov Random Field. The resulting MARF is able to track both supporting links and conflicting links (argumentative defeats) among the outcomes, the applied knowledge and the received information. It can compute the most probable argumentation for the outcomes and identify the pieces of knowledge or received information that would render the premises or outcomes unreliable or reverse the outcome dramatically.

For example, the MARF in Figure 2 is compiled from the following facts in the ELICIT tasks:\(^2\) (1) The Lion is involved; (7) The Chartreuse group is not involved; (9) The Purple or Gold group may be involved; (10) All of the members of the Azure group are now in custody; (12) There is a lot of activity involving the Violet group; (13) The Brown group is recruiting locals - intentions unknown; (16) Members of the Purple group have been visiting Omega; (18) The Azure group has a history of attacking embassies; and a domain constraint (S-1) there is only one answer for the who question: either Brown, Violet, Chartreuse, Purple, Gold, or Azure.

In Figure 2, Oval nodes are variable nodes tracking the acceptability status (i.e., accepted, rejected, undecided) of predicates (including equality assertion, e.g. who? := “Brown”). Square nodes are factor nodes modeling how predicates acceptability status interrelate with each other regarding the meaning of facts. For example, fact “10) All of the members of the Azure group are now in custody” relates acceptability of predicates inCustody(“Azure”) and the equality assertion who? := “Azure” (the answer to who is “Azure”). If inCustody(“Azure”) is accepted, then who? := “Azure” is likely to be rejected. Every factor node is associated with a weight to reflect how much such a factor should be taken into account when evaluating the probability of an acceptability assignment to predicates via an exponential family distribution parameterized by the weights of the facts:

\[
\text{Pr}(\vec{x}) = \frac{1}{Z} \prod_{F_j \in \mathcal{F}} \exp \left( \langle \vec{W}_j, \phi_j(\vec{x}_j) \rangle \right)
\]

where \(\langle \vec{W}_j, \phi_j(\vec{x}_j) \rangle\) is the inner product of the weights and the argumentative features \(\phi_j(\vec{x}_j)\) of the acceptability variables vector \(\vec{x}_j\) of a fact \(F_j\). \(Z\) is a normalization constant to ensure that \(\text{Pr}(\vec{x})\) is a probability distribution over all possible acceptability assignments. Argumentative features \(\phi_j(\vec{x}_j)\) reveal elements that are essentials in evaluating the meaning of the fact according to the formal argumentation theory. The higher the validity of an acceptability assignment is, the higher the probability of such an assignment will be; the higher the weight of a fact is, the higher the probability will be for an acceptability assignment that conforms with the meaning of the fact.

With MARF, we can model the interactions of premises, conclusions, inference rules, and argument attacks quantitatively through potential functions. Simple operations on these potentials facilitate the computation of a coherent probabilistic interpretation of the argumentation outcome — the argumentation structure along with the acceptability status assigned to premises, conclusions, inference rules and arguments. In addition, MARF provides a computational framework to learn probabilistic evaluation functions of the premises and outcomes following data revealing human reasoning.

\(^2\)The numbering of the facts are same as it is in the ELICIT fact set coded as 1aGMU.
4. Results

In this section, we will compare the results of human experiments, the ACT-R model, and the MARF model running on the same ELICIT task.

4.1. Human experiment results

Sixty subjects, divided into twelve groups of five, were recruited and finished the task. While they did not know how they were divided, four of each five worked cooperatively together, and the fifth worked separately, with three bots. Among the 60 subjects who participated in our experiments, 15 of them (including the subjects who worked with bots) answered the “who” question for the fact set “1aGMU17”. The results are depicted in Figure 3. Among all these participants, 50% of them reached to the correct answer, “the Violet group”, after seeing the first wave of facts. After seeing the second wave of facts, 100% percentage of the participants reached the correct answer. However, after the third wave, about 40% of the participants were confused by the new facts and changed their answers from the correct one.

4.2. ACT-R Results

Sample results for the “who” question are presented in Figure 4 of fact set “1aGMU17”. Probability estimates for each possible answer (i.e., all groups) are presented for each of three waves of facts. Note that those are the unnormalized (non-exclusive) probability estimates rather than actual (exclusive) forced-choice answers. The initial estimate for the violet group (the correct answer, as it turns out) is the highest following the first and second batch of facts, making it the preferred choice in these two phases as
for the human subjects. However, the estimate for the violet group falls to third-highest after the third batch of facts due to a dilution effect from a number of facts mentioning other possibilities.

Note that these results were generated without reflecting the effect of previous guesses on later phases. This would be a case where confirmation bias could actually lead to a correct final answer by strengthening the correct guess based on the effect of early evidence.

4.3. Probabilistic argumentation results

The results MARF over the same ELICIT task (the fact set “1aGMU17”) is depicted in Figure 5. The three waves of incoming facts are compiled into three MARFs. Figure 2 is the MARF compiled from the first wave of facts. In this first wave MARF, the fact “(1) The Lion is involved” is disconnected from other facts because the meaning of this fact is disconnected from other facts. After the second wave of facts, fact (1) is connected to the majority of facts (as depicted in Figure 6); however, there is a new disconnected fact (15). As more and more facts becomes available, the MARFs are able to consider more connected facts to evaluate the acceptability of the predicates underlying the meanings of these facts. After 3 waves of facts, the MARF is able to evaluate acceptability status of each answer as marginal probabilities, i.e., the probability of accepted, rejected and undecided, considering all available facts. To align with the results of human experiments and the ACT-R model, Figure 5 plots the probabilities of accepting the answers omitting the probabilities of rejected and undecided status of these answers where weights of all the facts are set to 10 and the weight of the domain constraint is set to 100. In the first two waves, MARF decides that the “Purple” group is the answer with probability of 94%. However, after the receiving the third wave, MARF changes its opinion sharply to the “Violet” group. This is the case because different from human and ACT-R model, the MARF is constructed to be decisive separating the accepted answers and the rejected answers as much as possible while following meanings of the available facts as closely as possible.

Figure 5: The probabilistic argumentation result

5. Conclusions and Further Work

We present an experiment and two computational models of group decision making. While both data analysis and model development are preliminary, they highlight interesting emerging effects. Rather than following a linear path, the deductive processes faced with a constant stream of facts induce a fluctuation in beliefs that reflect a potentially rich dynamic. Both computational models capture some aspects of the human data but not others. While both include a representation of the deductive process (e.g. a question activates its relevant facts and answers in the ACT-R model; an MARF factor models a logical deductive rule) and its constituent facts and conclusions, the processes reflect distinct assumptions regarding the parallel vs sequential nature of inference processes, the implicit vs explicit nature of probabilistic information, and whether those processes are fundamentally optimizing or satisficing. Still, those models share many representational assumptions regarding the nature and structure of the problem representation, which will allow us to formally examine the implications of their assumptions.

ACT-R uses IBL to drive decision making; while MARF uses potential factors to relate facts and answers. If the given facts and its symbolic representation truly reflects logical relation among the facts and answers, by design MARF will produce the right answers. Furthermore, as factors in MARF are interrelated through an undirected graphical model, MARF is not sensitive in the order of receiving facts but sensitive in the availability of facts. On the other hand, since the ACT-R model uses IBL activation, it is more sensitive to the order of receiving facts. Therefore the ACT-R model follows human decision making closely while MARF follows closely the logical relationships of information embedded in the facts to estimate the answers.
Figure 6: The compiled Markov Argumentation Random Field after the second wave of facts (light green oval nodes are accepted predicates; grey oval nodes are rejected predicates; light blue oval nodes are undecided predicates; red square nodes model the argumentative conflicting relationships among predicates)
Numerous avenues of work are possible for both data analysis and model development. We will examine whether learning processes can improve decision making with experience, develop models of judgments for information sharing, and analyze various experimental conditions to determine the answer to our initial question as to whether information is best shared at the most detailed level of basic facts or in the form of refined, high-level conclusions.

Acknowledgments

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defense or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References


Author Biographies

Yuqing Tang is postdoctoral research fellow in the Robotics Institute at Carnegie Mellon University. He has worked on symbolic argumentation-based reasoning, probabilistic hierarchical task networks and Markov Decision Processes (MDPs) planning, computational argumentation-based reasoning and dialogues for multi-agent planning, and argumentation-based trust models.

Christian Lebiere is a research faculty in the Psychology Department at Carnegie Mellon University. Since 1991, he has worked on the development of the ACT-R cognitive architecture. His research interests include cognitive architectures and applications to cognitive psychology, artificial intelligence, human-computer interaction, decision-making, intelligent agents, cognitive robotics and neuromorphic engineering.

Katia Sycara Katia Sycara is a Research Professor in the Robotics Institute, School of Computer Science at Carnegie Mellon University, where she is the Director of the Advanced Agent-Robotics Technology Lab. She serves as the Sixth Century Chair in Computing Science at the University of Aberdeen.

Don Morrison is a Senior Research Programmer in the Psychology Department at Carnegie Mellon University, also working in the Department of Social and Decision Sciences. His current work is centered on building tools to facilitate the construction of cognitive models.

Paul Smart is a senior research fellow in the School of Electronics and Computer Science at the University of Southampton. His main area of research is the Semantic Web. His research activities sit at the interface of the network, web and cognitive sciences.
ABSTRACT: In today's networks, individuals frequently face the problem of information overload. The amount of information available for a decision is often much larger than a person can process to make an informed decision. Past research has shown that individuals can differ significantly in how they use information in making decisions. Individuals may differ in their willingness to seek and incorporate more information into their decision making, some relying more on information at hand than simple heuristics. Individuals' desire to reach a closure quickly by making a decision may differ as well, depending on situational factors such as the level of inherent ambiguity or uncertainty in the decision. These factors have not yet been studied deeply in the context of networked information processing in terms of their impact on the timeliness and accuracy of decisions. In this paper, we address this problem by introducing an agent-based model that incorporates four characteristics representing individual differences: competence, engagement, decisiveness and reliance on neighbors' opinions for corroboration. Based on a novel way of modeling the degree of problem difficulty, we investigate the impact of individual differences in networked decision making through comprehensive simulation experiments. Our simulation results show that being more engaged with a task does not always improve team performance and can lead to information overload if it is coupled with high information push activity. Similarly, heuristic decisions as a result of high decisiveness can be useful in various problem settings and can be further improved by a small amount of corroboration.

1. Introduction

In today’s networks, individuals frequently face the problem of information overload: the amount of information available for a decision far exceeds the capacity of individuals to process information. However, individuals also differ in their motivations for seeking and engaging with information. Need-for-cognition (NC) [Cacioppo et al., 1982] refers to an individual’s desire to seek relevant information and integrate evidence to reach a conclusion. On the other hand, need-for-cognitive-closure (NCC) [Webster and Kruglanski, 1994] indicates the desire to arrive at a decision quickly to avoid discomfort caused by ambiguity or uncertainty. It has been shown that the individual differences in NC and NCC play a significant role in information sharing [Henningsen and Henningsen, 2004] and the influence one may have over others in the network [Marsden and Friedkin, 1993]. While agents models have incorporated the notion of bounded rationality [Carley et al., 2009], the impact of individual differences especially in the NC and NCC scales on accuracy and timeliness of decision making has not been studied deeply in the literature.

In this paper, we introduce an agent-based model for decision making in which individuals share facts with each other and make decisions based on information at hand in a networked environment. We manipulate
two significant aspects of the simulation: the characteristics of individual agents based on the NC and NCC scales and the overall problem difficulty. And then we investigate how these factors impact both the timeliness and the accuracy of decision making. The model proposed in this paper expands on our prior agent-based model [Chan et al., 2013], [Chan and Adali, 2012] with three new measures of an individual’s differences: engagement, corroboration threshold and decisiveness. While engagement is related to NC scale, corroboration threshold, and decisiveness are models of the different components of the NCC scale. In addition, we use the agent competence model from our prior work to arrive at four interrelated individual difference parameters.

We conduct an agent-based simulation study in which agents share information with the intention to make an informed decision. We introduce a novel way of modeling problem difficulty in terms of facts that contain arguments in support of making a decision (pro) and arguments against the same decision (con). Furthermore, we consider information that is useless (i.e., noise) but can cause confusion if interpreted as valuable due to lack of expertise. Our simulation scenario allows agents to make heuristic decisions when only little information is available about the given task and more informed decisions can be made upon receiving more of the available information. By changing the distribution of facts along with the two dimensions of type of evidence (i.e., pro vs. con) and benefit (i.e., valuable vs. noise), we manipulate the underlying difficulty of the problem. The more difficult a problem is, the higher the risk of making an incorrect decision with limited information and in the presence of information processing errors.

Using this novel model, we study how individual differences impact the timeliness and accuracy of decision making. We demonstrate that being more engaged with a task by processing and sending out information to the network does not always improve team performance and can lead to information overload. This type of engagement results in noisy information being multiplied in the network faster than the network’s ability to filter it out. Similarly, higher decisiveness modeled as reliance on smaller set of facts can be more robust to this type of information overload in some situations. Interestingly, corroboration coupled with high decisiveness results in best performance overall, by making quick decisions and then reducing the overall noise in the network quickly by routing only information that supports a given decision.

2. Related Work

There is a great deal of work on factors relating to information sharing behavior in terms individual or social motivations. In trust literature, the focus is on understanding how individuals trust others as sources of information for decision making [Fiske et al., 2007]. Often the competence of others and their reliability are prominent factors. In information processing, individuals also concentrate on factors that relate to the properties of the information itself [Hilligoss and Rieh, 2008] that signal whether a piece of information is likely to be true based on heuristic factors such as the presentation of the information and the confidence of the source. In particular, the information consumer integrates these multiple concerns to derive the credibility of the information as well as the trust for the source depending on the decision context [Adali, 2013]. If the information consumer has sufficient cognitive resources and expertise in the problem domain, they are more likely to process information in an effortful manner and rely on their own judgment. In other cases, they are more likely to rely on the surface cues of the information itself or on the trust for the sources. Some agent-based models have incorporated the trust aspect of information processing into networked decision making situations [Chan et al., 2013], [Thunholm et al., 2009]. In particular, the taNdem (A Trust-based Agent framework for Networked DEcision Making) agent simulation system [Chan et al., 2013] is the first to explicitly model the competence and willingness of agents as well as the trust beliefs for the competence and willingness of other agents. However, the taNdem does not consider the individual differences of agents in information processing behavior as well as the underlying difficulty of the problem being solved, which is the focus of this paper. Other work has concentrated on the influence of opinion leaders and beliefs of individuals in settings with agents with bounded processing capacity [Carley et al., 2009]. In this paper, we do not incorporate prior beliefs and social influence to the model, but leave these to our future work.

An additional line of work considers an individual’s information seeking behavior in an information pull scenario. The main problem is to understand how individuals choose to query sources, what keywords they use and which sources they select [Case, 2008]. Some information models study how individuals’ understanding of the problem domain evolve over time based on the information they process [Pioroli and Fu, 2003]. They use a brain activation model that shows how people build a mental model of the problem space. On the other hand, other models aim to understand how the underlying tasks, the awareness of existing information and the outcomes of different information seeking actions predict the future actions of individuals [Leckie, 2005].
In this line of work, the models need to balance situational factors, such as availability of information, with personal factors such as the desire to seek information [Ingwerson and Jarvelin, 2005].

Two types of individual differences play a significant role in these information models: the need for cognition (NC) [Cacioppo et al., 1982] and the need for cognitive closure (NCC) [Webster and Kruglanski, 1994]. An individual with high NC tends to engage more with information, processing more and basing her decisions on the gathered information [Cacioppo et al., 1982]. An individual with low NC tends to take a heuristic approach in information processing and to make decisions based on it such as fluency of information (i.e., presentation of information). The NCC scale [Webster and Kruglanski, 1994] deals with the desire of an individual to quickly reach closure by making a decision. An individual with high NCC tends to make faster decisions while an individual with low NCC can delay decisions. While these two measures are correlated [Kossowska and Bar-Tal, 2013], NCC is more complex integrating five different personality aspects: preference for order, predictability, decisiveness, discomfort with ambiguity and closed-mindedness. Hence, delaying a decision may not necessarily be a result of the desire to process more information as in NC, but for a desire to resolve ambiguity of the underlying decision. Furthermore, it has been shown that the individual components of the NCC scale remain the same across multiple cultures [Mannetti et al., 2002].

Despite the importance of these two factors (NC and NCC) in determining how individuals process information, there is little work in understanding how these factors interact in networked decision making scenarios in which teams often have to deal with information overload. To the best of our knowledge, this paper is the first to investigate their impact on networked teams.

3. Agent-based Model

In the paper, we consider an agent-based model where agents are connected to each other through an undirected network which could model either a communication network connectivity, a social network ties or organizational role based relationships. Individuals communicate with all their neighbors in the network at all simulation steps to accomplish a task. We leave the impact of prior or task specific trust which would lead to changing communication partners throughout the simulation to future work. Agents in the simulation exchange information to help each other make a decision. Each unique piece of information is called a factoid. In the information sharing scenario we consider, a fixed number of factoids are distributed to all the agents’ inbox in the beginning of the simulation, and diffuse to other agents throughout the simulation.

At each step of the simulation, each agent processes some of the factoids from its inbox. For each factoid they process, they make a determination on whether the factoid contains valuable information or not. If the agent thinks the factoid is valuable, it will first put this factoid in its knowledge base and immediately send the factoid to all its neighbors in the network. Agents will not send the same factoid out more than once. However, an agent may receive and process the same factoid multiple times as it arrives from different neighbors as long as it is not yet in its knowledge base. In other words, if the agent did not think a fact was valuable the first time it has seen it, it may do so upon receiving it multiple times. The new factoids received from neighbors will accumulate at the top of the inbox of each agent at the end of each simulation step.

This simulation scenario is very similar to ones proposed in prior work [Chan et al., 2013], [Thunholm et al., 2009] with a few differences. First, the fact that new information accumulates on top of the inbox models the reality of many information push scenarios such as micro-blogs or an inbox sorted by recency. As another new addition, agents will make decisions as soon as they think that they have processed sufficient number of factoids. The decisions are based on the amount of knowledge available in this model, but not a time deadline. This choice allows us to investigate how other simulation factors impact both the timeliness and the correctness of decisions.

Once an agent makes a decision, it continues to share factoids with other neighbors, but now it only shares factoids that it thinks as valuable and supports the current decision. This allows agents to influence other agents’ decisions by selective sharing. Agents may change their decisions as more facts become available in the simulation. In our model, an agent’s behavior is only influenced by the personal (nodal) characteristics, not by edge characteristics like trust (e.g., trust relationships with communication partners). We leave this aspect of investigation for our future work.

3.1 Modeling of Decision Tasks

To enable the agents to make decisions, the factoids corresponding to a task are divided into four types.

The benefit of factoids differs between valuable and noise. A valuable factoid is evidence relevant to decision making. A noisy factoid (or noise) is information that should be disregarded completely in an ideal scenario. However, agents may make errors in detecting usefulness
of the evidence and judge a valuable factoid as noise (i.e., false positives), and a noisy factoid as valuable (i.e., false negatives). These errors lead to using incorrect information in decisions and increased noise in the network.

The evidence of a factoid can be Pro or Con. Pro factoids are arguments supporting a decision, and Con factoids are arguments against the decision. Agents do not make errors regarding the evidential value of a factoid, whether a factoid is for or against a decision. If a fact is Pro, it will always be known correctly as Pro, or vice-versa.

The ground truth of a fact is either Pro Valuable (PV), Pro Noise (PN), Con Valuable (CV) and Con Noise (CN). Each decision task is supported by a number of factoids of different types. We will use the representation \([V : (x_1/x_2), N : (y_1/y_2)]\) to represent the number of facts with ground truth PV \((x_1)\) and CV \((x_2)\), and PN \((y_1)\) and CN \((y_2)\). The correct decision for the agent should be Pro if \(x_1 > x_2\) and Con, otherwise.

As agents are bounded, they do not have access to all the facts and to the ground truth for the facts. Without global view based on perfect knowledge, all the agents make decisions based on their own knowledge base at a given point in time which contains the factoids they perceive valuable. We represent the knowledge base of an agent at some point \(t\) in the simulation with \([([z_1/z_2])\] where \(z_1\) is the number of Pro factoids and \(z_2\) is the number of Con factoids the agent has processed and perceives as valuable at time \(t\) (including those identified in error). An agent will make a Pro decision if \(z_1 > z_2\) and a Con decision otherwise.

The distribution of factoids along the four dimensions represents the inherent difficulty of a decision. Suppose we have \([V : (50/25), N : (10/10)]\). This is an easy decision because there is overwhelming evidence in support of the decision and very little noise. Even if an agent has a small subset of the factoids available, it is very likely that it will have a higher number of Pro factoids than Con factoids, resulting in a correct decision. This is true even if the agent does not correctly identify the value of some subset of factoids. In this case, waiting to process more factoids is not likely to improve the decision accuracy.

A more difficult decision setting could be given by \([V : (50/25), N : (10/100)]\). As there is more noise
than valuable factoids, the agents must not misidentify valuable information and filter it out. Also, they must not make errors regarding the many Con noise factoids and use them as valid evidence against the decision. In short, even a small tendency to make errors can result in incorrect final decisions due to multiplied noise in the network and loss of valuable information.

To summarize, there is a big risk of making a wrong decision if \( x_1 > x_2 \) but \( y_2 \gg y_1 \) and \( y_2 \gg x_1 \). In other words, if there is a lot of Con noise \( (y_2) \) compared to pro valuable facts, then even a small percentage of error can lead to incorrect decisions. Note that if \( y_2 \gg x_1 \) and \( y_2 \approx y_1 \), the risk is reduced because agents now can make equal mistakes for both Con and Pro noise. Even though the decisions are more random, the risk of errors is reduced.

### 3.2 Modeling Agent Characteristics

The behavior of each agent is a function of its personal characteristics modeled by four distinct parameters: competence \((c)\), engagement \((e)\), corroboration factor \((cf)\), and decisiveness \((d)\). We can manipulate the characteristics of all the agents in the simulation, or a subset of them. Each agent is cognitively limited, they can process at most \( C \) factoids in a single simulation step and can only make decisions based on the information available.

- **Competence** \((c)\) models the task specific expertise of an agent (between 0 and 1). An agent with a competence value of \( c \) will correctly identify the benefit of a fact (valuable or noise) with probability \( c \). When \( c = 1 \), the agent will always identify a fact correctly. The evidence type is always correctly processed, regardless of the competence level.

- **Engagement** \((e)\) models the level of engagement of an agent with the decision making task (between 0 and 1). An agent with engagement \( e \) will process \( C \times e \) facts from its inbox at each simulation step, with \( e = 1 \) representing full engagement. Engagement controls how much information is incorporated into decisions and models NC.

- **Corroboration factor** \((cf)\) models an agent’s reliance on the corroboration of facts by others for decision making (integer value 1 or higher). This parameter has an effect both in processing of facts and in decision time. When processing a fact with \( cf > 1 \), the agent will find a factoid valuable if it is sent by at least \( cf \) agents at decision time regardless of its own opinion. When \( cf = 1 \), the agent only relies on its own evaluation of the factoid. High \( cf \) values can be a reliable signal of the benefit of facts when agents have a competence of 0.5 or higher, but it may take many simulation steps for an agent to observe a high \( cf \) value.

### Decisiveness \((d)\)

Models how many facts the agent needs to have seen (regardless of their perceived benefit) before making a decision (between 0 and 1). An agent with decisiveness of \( d \) will need to have seen at least \( (x_1 + x_2) \times (1 - d) \) factoids before making a decision. A decisiveness of 0.2 means that the agent must have seen at least 80% of the \( (x_1 + x_2) \) factoids. Hence, low decisiveness threshold means that agents need to see a lot of facts and will make decisions more slowly.

Agents make a decision after meeting the decisiveness threshold. At this point, they base their decision on the facts in their knowledge base that have passed the corroboration threshold. In essence, agents can change their mind in two ways. First, a factoid they perceived as noise may eventually be put in their knowledge base if it is seen \( cf \) times. Secondly, a factoid that was considered valuable may eventually be disregarded at decision time if it has not met the \( cf \) threshold. Once an agent makes a decision, they send only facts that support their decision. Both corroboration factor and decisiveness model various aspects of the NCC scale.

The details of the agent actions and its dependence on the given agent characteristics are given in Algorithm 1.

### 4. Experimental Setup

Given the model described in Section 3, we run a number of experiments to understand the impact of different factors in team performance. We use the following performance metrics to evaluate our model:

- **Correct decisions**: total number of agents (out of 20) making a correct decision at the end of the simulation.

- **Accuracy**: percentage of decisions that are correct at the end of the simulation with accuracy of 1 representing 100% of correct decisions. In our settings, the PRO decision is always the correct decision without loss of generality.

We create a Watts-Strogatz network with 20 agents, each node connected to 3 neighbors with a 0.2 probability of rewriting edges. Then, we seed all the agents with the factoids from the problem space: \( P = \{(x_1/x_2), (y_1/y_2)\} \) where \( (x_1/x_2) \) are the number of valuable pro/con factoids, and \( (y_1/y_2) \) are the noise pro/con factoids. Each factoid is sent to 3 agents randomly selected in all our experiments. We run each experiment for 10,000 steps and repeat 100 times. In all our tests, the maximum amount of information that can be processed by an agent at a single simulation step is 100 (i.e. \( C = 100 \)).

#### 4.1 Engagement

We first study the impact of engagement. We set the corroboration factor \((cf)\) to 1, forcing agents to
only consider their own judgments of factoids’ benefit. Intuitively, if an agent is engaged with an activity, they are expected to incorporate more information and make better decisions. To study this hypothesis, we construct 4 different experimental problem settings, given the same total number of facts as shown below.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y_1$</th>
<th>$y_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sc. 1</td>
<td>50</td>
<td>40</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Sc. 2</td>
<td>50</td>
<td>40</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Sc. 3</td>
<td>50</td>
<td>40</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Sc. 4</td>
<td>50</td>
<td>40</td>
<td>0</td>
<td>150</td>
</tr>
</tbody>
</table>

We run two sets of experiments with low (0.6) and high (0.7) competence as shown in Figure 1. In many scenarios, higher engagement actually reduces the decision accuracy. The reason is that whenever an agent determines that a factoid is valuable erroneously, it copies the factoid to all its neighbors. Hence, noise is multiplied quickly. As a result, if each agent processes a lot of information at once and ends up sending out a lot of noise, the overall noise in the network is suddenly multiplied. Let us now consider a different extreme case. Suppose a single factoid is processed by each agent at each time step. For a noise factoid to be sent from agent 1 to 2, and then 2 to 3, both agents 1 and 2 have to make consecutive errors (by probability 0.3 each). As long as the overall competence of agents is above 0.5, noise is slowly eliminated in network processing. We have illustrated this effect in our previous work [Adalı et al., ress].

Given these two competing factors, the problem space is crucial in determining which one is going to be more dominant. In scenario 4 with a considerable amount of misleading noise (150 CN factoids), even small errors lead to a large amount of noise being multiplied in the network and agents spend all their time filtering this information out. By increasing engagement, filtering is delayed and the overall team effectiveness is reduced. In scenario 2, we see that agents with high decisiveness are impacted negatively from this problem. As information is passed through the network, the agent needs to wait to make a decision, letting the filtering process take place to reduce noise. However, in scenarios 1 and 3, where the noise tends to provide evidence for the correct decision, high engagement allows good decisions to be made quickly and frequently.

Overall, we can observe increased engagement only helps accuracy in the situations without high quantities of misleading noise. In situations where engagement can cause information overload, lower decisiveness is more beneficial.

Finally, we can verify this finding by looking at the timing of decisions as shown in Figure 2 (a). Each of the four scenarios was run with a competence of 0.7. Higher engagement results in faster but lower accuracy initial decisions. Initial decisions guide the remaining network traffic, resulting in faster convergence to similar decisions for the other nodes. The only case in which this is not true is in Scenario 4 with a lower number of facts and an easier problem scenario.

### 4.2 Corroboration Factor

In this section, we consider the impact of the corroboration factor in final decision accuracy. For these experiments, we set engagement to 0.8 and consider the two scenarios shown below.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>Competence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy</td>
<td>50</td>
<td>40</td>
<td>10</td>
<td>100</td>
<td>0.8</td>
</tr>
<tr>
<td>Difficult</td>
<td>50</td>
<td>40</td>
<td>50</td>
<td>100</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Agents’ increased reliance on corroboration improves accuracy in two ways. First, in low competence cases, the agent’s own opinion alone is too noisy and considering facts corroborated by others helps improve accuracy significantly as it is unlikely for noise to be highly corroborated when competence is above 0.5. Hence in the easy scenario with a competence of 0.8, the improvement due to corroboration is small despite the slightly higher imbalance in the noise. The second effect is due to decisiveness. Higher decisiveness requires fewer factoids, leading to faster decisions. After a decision, agents route only factoids that support their decisions and reduce the overall traffic in the network. High decisiveness with a small amount of corroboration leads to optimal results by increasing accuracy of early decisions and improving overall performance of the network. This quick convergence for high decisiveness (0.8) can be seen in Figure 2 (b). The final decision accuracy is shown in Figure 3. We first note that corroboration improves decision accuracy significantly but there is no significant improvement above a factor of 2 in our problem setting (in which each factor is sent out to 3 agents in the beginning). There is already a significant network effect in filtering noise. Furthermore, the odds of receiving the information 4 times is negligible in our network as each agent is connected to 3 others on average.

5. Conclusions
In this paper, we introduced an agent model for studying the impact of the need for cognition and need for closure individual difference scales on networked decision making. The proposed model models agents with various characteristics: the competence in distinguishing between noise and valuable information, the decisiveness in terms of being able to make decisions based on few factoids, relying on corroboration to reduce ambiguity and engagement to process multiple facts at each time. We modeled the degree of problem difficulty in a novel way that allows us to study the impact of these differences in realistic information sharing scenarios. Our simulation experiments show that when agents are low in competence, the dependence on corroboration is high. High decisiveness is not always desirable as agents may miss out relevant information while making their decision. Reliance on corroboration with high decisiveness results in an optimal scenario, leading to fast heuristic decision making with high accuracy through corroboration. Higher engagement resulting in higher
Fig. 3: The impact of corroboration in easy (top) and hard (bottom) problem scenarios.

information processed and sent to the network is not always desirable, reducing the ability of the network to reduce noise through information dissemination paths. This effect is likely to be more intense in denser communication networks. In our future work, we plan to investigate the effects of different network structures, a subset of agents with different characteristics on overall network performance as well as other aspects of the NC and NCC scales such as the desire to stick with one’s decisions and open-mindedness on network and team performance.

Acknowledgment

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

III.
TUTORIAL
Introduction to Dynamic Network Analysis

Geoffrey P. Morgan  
William Frankenstein  
Kathleen M. Carley  
Carnegie Mellon University  
School of Computer Science  
Institute for Software Research  
5000 Forbes Ave  
Pittsburgh, PA 15213  
gmorgan@cs.cmu.edu, wfranken@cs.cmu.edu, kathleen.carley@cs.cmu.edu

Keywords:  
Network Analysis, Key Entity Identification, Grouping, Network Comparison, Network-Centric Modeling

ABSTRACT: We discuss dynamic network analysis and how it can be useful both for instantiating simulations and for novel simulation outputs. The tutorial focuses on providing attendees grounding in network analysis, allowing attendees to incorporate network data as inputs or as outcomes of their simulation tools. We first introduce key network analysis concepts and statistics including centrality measures and brokerage. To ground these concepts and provide some intuition for thinking about different types of centrality measures, we run a short hands-on analysis of network data in ORA based on intelligence from the Revolutionary War. We then introduce the data structures commonly used in network analysis and help attendees import their own data into ORA. We then introduce attendees to grouping algorithms including clique identification, Concor, and Newman-Girvan. We conclude with additional methodologies to compare networks such as QAP. These methods are particularly useful for comparing simulation inputs and outputs, and inform analysis of simulation outputs. Attendees are encouraged to bring their own data, but sample data will be provided.

1. Introduction

Network analysis, the evaluation of how nodes (including agents, resources, locations, and more) relate to one another, has proven useful in many diverse fields, from ecology, organizational psychology, and cyber-vulnerability analysis. This tutorial is designed to provide attendants with a grounding in network analysis methodology and ideas on how to incorporate networks into simulation inputs and outputs. Networks are useful ways of representing virtual proximity, limits on an actor’s social cognition (Simon, 1991), and interactions between nodes. We present both theoretical information and how to apply these analysis techniques using ORA (Carley, Pfeffer, Reminga, Storrick, & Columbus, 2013).

1.1 Why is this material important?

This material is useful to anyone trying to incorporate large data-sources into their simulation tools. Networks provide a useful and effective way of leveraging large-scale event data that is well-suited to informing agent-based models. Further, network analysis techniques may provide new ways of leveraging and analyzing existing simulation data. Finally, network techniques are becoming more common, and having some familiarity with the area should provide some utility to many attendees.

1.2 Tutorial Purpose

The tutorial intends to introduce attendants to network analysis and offers ideas of how network analysis can be used to both inform simulation and examine simulation outputs. It will present material refined yearly for a Summer Institute in network methodologies held at Carnegie Mellon since 2001.

1.3 Intended Audience

The intended audience of this tutorial includes people first learning about network analysis techniques or those who are interested in evaluating existing data via network analysis. It provides the broader theoretical context on network analysis that may be useful to these attendees while still focusing on applications. People
also interested in learning about network-centric simulation, either to use existing simulations or to incorporate elements into their own tools, will also be well served. Attendees are encouraged to bring their own data (ideally a sub-sample).

2. Contents
In the course of this tutorial, we will discuss how to identify key entities in a network, various methods of grouping entities together within a network, ways of transforming data from multi-mode (agents x locations, for example) into single-mode data, and finally ways of comparing networks.

The tutorial will include lecture and exercise based elements.

2.1 Characterizing a network and Key Entity Identification
We introduce tutorial attendees to some of the key terms and ideas used in network analysis literature, such as nodes, ties, modes and classes, and the concept of dynamic networks. Nodes are analogous to agents; they represent individuals in a network. Ties are analogous to links or edges between nodes in a network. Modes and classes highlight how networks can represent ties between nodes of the same type or class, such as a social network representing ties between agents; networks can also represent ties between different types of classes, such as membership lists of organizations. In doing so, we review standard notation for representing these networks as matrices, as well as statistics used to characterize networks.

We introduce and define network statistics characterized at the level of the network and node. At the network level, these statistics include density, link count, isolate count, component count, reciprocity, characteristic tie length, clustering coefficient, average distance, and diameter. At the node level, these statistics include centrality measures including degree, betweenness, eigenvector, and closeness. By considering these node level statistics and the type of network being analyzed, we can identify network ‘elite’ and key entities (Wasserman, 1994).

In identifying key entities, we consider the type of network being analyzed as well as the interplay of different network statistics – and highlight the role of the meta-network (Carley, 2002; 2005) in helping characterize these networks for organizations. To characterize the behavior of key entities, we consider brokerage behavior, bridges, and structural holes in networks. We utilize different centrality measures to match key entities to these behaviors and network features. Degree centrality, which considers a node’s total number of connections, can be useful for identifying resource-rich individuals. High betweenness centrality indicates an ability to connect disparate groups, but may also limit the actual paths available for action to the node due to politics. Eigenvector centrality, which rewards being connected to highly degree central nodes, indicates agents with strong social capital. Closeness centrality can indicate nodes able to utilize several different resources very quickly.

2.2 Paul Revere, Broker of the American Revolution – A motivating example
We demonstrate the power and intuition of network analysis with a motivating example of British intelligence from immediately before the American Revolutionary War (Han, 2009). In doing so, tutorial attendees have an immediate, concrete example of how to think about network analysis for their own research, as well practical experience with utilizing ORA.

We start with membership lists for five distinct groups that played a crucial role in pre-revolutionary Boston: St. Andrews Lodge, Loyal Nine, North Caucus, Long Room Club, and the Boston Committee. We also have lists of individuals who would be affiliated with the Tea Party and London Enemies – not official groups yet, but members would be placed on intelligence watchlists. This data will be included in tutorial attendee packages for easy import into ORA (Healy, 2013).

An initial analysis of this data does not reveal significant insight. However, by manipulating these membership lists into social and organizational networks, we gain insight into key entity behaviors. We manipulate the membership data by “folding” networks to infer social (agent by agent) networks and organizational (organization by organization) networks – multiplying the membership lists (an agent by organization network) by its matrix transpose.
We gain further insight and intuition into using network and node statistics by contrasting the networks generated when we consider only the networks generated by actual organization membership lists, deleting the intelligence watchlist information of Tea Party and London Enemies. We also gain insight into how centrality measures behave when computing new measures on the smaller dataset.

In using the Paul Revere data, tutorial attendees will gain familiarity with ORA’s range of features and reports, which will prepare them for using their own data for the remainder of the tutorial.

2.3 Network Data Import to ORA
After providing a motivating example of Paul Revere and his connections in colonial Massachusetts, we use this section to help participants understand the forms network data takes and to import and do basic analyses their own data.

We describe the two foundational network data representations, matrices of data (sometimes called the dense representation), and edge or link-list data (sometimes called the sparse representation). Much modern data not explicitly collected for network analysis can be most conveniently represented in link-list form, where rows from existing table-data can be used to create links.

Further, modern network analysis often finds it useful to be able to add attributes or characterizations to nodes. This can be useful in a variety of ways, including partitioning nodes and visual examination of networks colored by specific attributes.

In this section, we will teach participants how to use ORA to import various forms of network data, ideally including a sample of their own data, how to understand the ORA interface, and how to run network reports to get quick assessments of the measures discussed in the earlier section.

2.4 Grouping Algorithms
In this section, we will provide a primer in various ways groups can be identified in network data. We provide a definition of a group, reasons why grouping is useful, a brief taxonomy of grouping methods, and how we evaluate grouping algorithms. We then turn to specific discussion on Clique identification, the Concor algorithm (Breiger, Boorman, & Arabie, 1975), and the Newman-Girvan algorithm (Newman & Girvan, 2004).

Attendees will then be encouraged to perform grouping analysis either on their provided data, or on sample data provided for the purpose.

2.5 Comparing networks
Comparing networks is particularly useful to the modeler, as change in a specific network of interest could be a useful outcome of a simulated intervention. We discuss why standard statistical approaches are not suitable for comparison of network structures, since neither the independence nor identical distributions assumptions are supported.

We discuss Hamming Distance (Hamming, 1950), which is the number of differences between the two networks, when their matrices are transformed into one large vector. Hamming Distance, while useful for understanding overall differences in a network, does not provide a useful structural comparison.

We then present QAP (Hubert & Schultz, 1976) and MRQAP models (Krackhardt, 1988), which allow analysts to assess how likely one network structure is to appear as the result of another network structure. MRQAP can be thought of, essentially, as a small simulation model evaluating many instances rapidly and then providing a confidence bound on the strength of relationship between two networks.

We provide sample data that allows multiple hypotheses to be tested using QAP/MRQAP procedures, and guide attendees in how to perform this test on this sample, or their own data if it is sufficient for testing purposes.

3. Conclusion
In this tutorial, attendees will learn about network representations, common network measures of centrality and importance in network, commonly applied grouping algorithms, and comparison methods for networks. This tutorial will be useful to people interested in networks, or interested in how to leverage large-scale event-data for use in agent-based simulation.

We believe this hands-on tutorial will be useful and interesting to many modelers.
4. References

Author Biographies
GEOFFREY P MORGAN is a PhD student in Carnegie Mellon’s Computation, Organization, and Society program. Morgan has experience in developing autonomous robotic systems, self-learning systems, cognitive models, and user-assistive systems. He is interested in organizational impacts on individual decision-making and on optimizing organizational performance.

WILLIAM FRANKENSTEIN is a PhD candidate in Engineering and Public Policy at Carnegie Mellon University. His research focuses on issues at the intersection of international security and technology policy, including risk communication and risk management and their application to the field of weapons of mass destruction proliferation. He is especially interested in the application of social media analysis, simulation, and network modeling to this field.

KATHLEEN M CARLEY is a Professor and the director of the Center for Computational Analysis of Social and Organizational Systems (CASOS) at Carnegie Mellon University. Her research combines cognitive science, social networks and computer science to address complex social and organizational problems.
IV.

PLENARY PRESENTATION
Experience

Deputy Director, Human Performance Training and BioSystems Directorate, OUSD(AT&L)
August 2013 - Present

Military Deputy; Program Officer “Neuroscience of Warfighter Performance” Office of Naval Research
July 2010 - August 2013

Deputy Director of Research – Science, Technology, Engineering and Mathematics
May 2012- August 2013

President, US Naval Aerospace Experimental Psychology Society
April 2009 – June 2012

Acquisition Professional
October 2008 - Present

Program Manager, Defense Advanced Research Projects Agency
March 2008 – July 2010

Naval Aerospace Experimental Psychologists - Assistant Speciality Leader
November 2006 – September 2009

Office of Chief of Naval Personnel, Branch Head, Strategy & Concepts
November 2006 – March 2008

Office of Chief of Naval Personnel, Lead, Human Systems Integration
August 2006 - November 2006

Potomac Institute for Policy Studies - Lewis and Clark Fellow
August 2005 - August 2006

Office of Naval Research - Deputy PM for Virtual Technologies & Environments
August 2005 – August 2006

Naval Research Laboratory – Head, Training Technology & Evaluation; IRB Chair
March 2002 – August 2005

Naval Air Warfare Center (NAVAIR) Orlando - Lead, Training Effectiveness Evaluation
February 2000-March 2002

Commander Joseph Cohn is an Aerospace Experimental Psychologist (AEP) in the U.S. Navy’s Medical Service Corps currently assigned as Deputy Director to OUSD (AT&L)’s Human Performance Training and BioSystems Directorate, overseeing both the DoD’s Human Research Protection Programs and the DoD’s $3B/yr investment in Human Systems and Medical research. During his previous tour, he served in the Office of Naval Research’s Human and Bioengineered Systems Division, as a Military Deputy and Program Officer. He also served as ONR’s first Deputy Director of Research, for Science, Technology, Engineering and Mathematics (DDoR – STEM). As Military Deputy and Program Officer, he was responsible for developing & implementing all aspects of strategic planning for DoN’s $80M/yr human systems research investments and directly managed an additional $70M research portfolio covering basic and applied biomedical and human systems research to enhance warfighter performance. As DDoR – STEM, he was responsible for all aspects of the DoN’s $90M/yr investment in outreach efforts to develop its future STEM workforce. CDR Cohn’s first assignment was at NAVAIR Orlando, as the Lead, Training Effectiveness Evaluation for Virtual Environments, where he transitioned the Conning Officer Virtual Environment system to the Surface Warfare Officer’s school, reducing qualification time by 50%. He was then assigned to the Naval Research Laboratory, where he established the Warfighter Human Systems Integration Laboratory, which created new technologies that reduced training system costs by as much as 50%, for NAVAIR, the USMC’s Training and Education Command and the Army’s Soldier Battle Labs. CDR Cohn also served as Deputy PM for ONR’s $55M Virtual Technologies & Environments program, leading the development & transition of modeling & simulation - based training technologies to the Navy, Marine Corps and Army. Next, CDR Cohn was Head, Strategy & Concepts Branch at OPNAV N1, responsible for developing the Chief of Naval Personnel’s MPT&E Research Investment Strategy. Following this, he was a Program Manager at the Defense Advanced Research Projects Agency directing over $70M in basic & applied research projects that delivered cutting edge biomedical and information technology products, including: in-theater brain-imaging / TBI diagnosis technologies, advanced brain-system interfaces, technologies that inoculate warfighters against PTSD, and a Digital Tutoring system that reduced by an order of magnitude the time required to train novices to perform at the expert level.

CDR Cohn has co-authored over 80 publications, chaired numerous panels and workshops and been an invited speaker to national and international conferences on human systems research. He co-edited a 3-volume book series focusing on training system development, a book on enhancing human performance in high risk environments and is working on a book entitled “Modeling Sociocultural Influences on Decision Making.” His military decorations include the Defense Meritorious Service Medal, the Joint Meritorious Unit Award (2), the Meritorious Service Medal (4), the Navy Commendation Medal (3), the Army Commendation Medal, and the Navy Achievement Medal (2). He was a co-recipient of the Undersecretary of Defense (AT&L)’s 2014 Award for Excellence, in recognition of his support for the DoD’s Ebola efforts. He was a co-recipient of the 2013 Admiral Jeremy M. Boorda Award for Outstanding Integration of Analysis and Policy-Making. In 2012, he received the USN AEP Society’s Michael G. Lilienthal Leadership Award. In 2009 he received the Association of Medical Service Corps Officers of the Navy’s “Best in Innovation” Award for developing a portable Traumatic Brain Injury diagnosis tool; in 2007 he received that Association’s “Best in Innovation” Award for developing neurocognitive technologies to ensure Warfighter resilience. In 2006 he received the Navy Modeling & Simulation Award, training Category, from ASN (RD&A). From 2006 to 2009 he served as Assistant Specialty Leader for the Aerospace Experimental Psychologists, responsible for recruiting new officers, mentoring over a dozen junior officers and liaising with the Navy’s Bureau of Personnel to meet the administrative needs of 30+ officers. He is a Fellow of the American Psychological Association, and the Society of Military Psychologists, & Associate Fellow of the Aerospace Medical Association. He also Co-chairs the International Cross Cultural Decision Making Conference, & the 2015 National Defense Industrial Association’s Human Systems Division Conference.

Education

Certificate, Effective Writing in the Federal Government
Office of Paracental Management / Management Development Center – September 2007

Certificate, Aviation Psychology, Human Factors and Applied Cognition
George Mason University – May 2006

Postdoctoral Fellow, Center for Complex Systems
Florida Atlantic University - June 1999

Doctor of Philosophy, Neuroscience
Brandeis University – August 1997

Bachelor of Science, Biology
University of Illinois, Urbana-Champaign - May 1992

Certificate, Information Systems Management
The George Washington University – March 1989

Certificate, Crisis Management
University of Colorado, Boulder - August 2002

Certificate, Executive Management
Institute of Executive Management - January 2000

Certificate, Air Traffic Control
National Traffic Control Center - September 1999

Certificate, Brain/Neuroscience
California Institute of Technology/Washington University - August 1992

Certificate, Experimental Psychology
California Institute of Technology/Washington University - August 1992

Certification, Project Management
The George Washington University - June 1999

Certification, Research Ethics
University of Chicago - September 2012
Understanding, Representing & Enhancing Intuitive Decision Making: From Individuals to Societies

Progress, Challenges, and Opportunities for Representing Behavior

Joseph Cohn, PhD
Deputy Director, Human Performance Training and BioSystems,
Office of the Assistant Secretary of Defense (Research & Engineering)

Research in human pattern recognition and decision-making suggest that there is a “sixth sense” through which humans can sense unique patterns without consciously seeing them (Rensink, 2004; Winerman, 2005) and subsequently act on these patterns. Evidence is accumulating that this capability, known as “intuition,” enables the rapid detection of patterns in ambiguous information contexts, that it informs the decision making process and, most importantly, that it may not require domain expertise to be effective (Jung-Beeman et al., 2004). Intuitive decision making processes are different than analytic decision making ones, which require that decision makers either have significant domain expertise (which translates to years of experience) or that decision makers have access to powerful computational and display technologies. Because intuition may help warfighters make difficult decisions in dynamic and uncertain environments, under significant time constraints intuition is a strong candidate for further exploration as the basis for developing a new set of decision support training technologies. The focus of this keynote address is on challenges and solutions to understanding, representing (through models) and enhancing intuitive decision making. In a broader sense, this keynote will use intuitive decision making as an example to also look at fundamental questions regarding how human behaviors are understood and represented at different levels of fidelity – loosely following Newell’s ‘bands’ (Newell, 1990), with a particular emphasis on making the leap from representing an individual’s intuitive decision making behavior to representing a society’s decision making behavior.

References


