BRiMS 2012

21st Conference

Behavior Representation In Modeling & Simulation

Table of Contents

Page	Section / Paper

BRIMS Conference Committee 2012

Conference Chair

Tiffany S. Jastrzembski	Air Force Research Laboratory

Program Committee C-Chairs:

Bradley J. Best	Adaptive Cognitive Systems
William G. Kennedy	George Mason University
Robert St. Amant	North Carolina State University

Executive Committee:

Joe Armstrong, CAE

Brad Cain, Defence Research and Development Canada

Bruno Emond, National Research Council Canada

Coty Gonzalez, Carnegie Mellon University

Brian Gore, NASA

Jeff Hansberger, Army Research Laboratory

Kenneth Kwok, DSO National Laboratories, Singapore

John Laird, University of Michigan

Christian Lebiere, Carnegie Mellon University

Christopher Myers, Air Force Research Laboratory

Bharat Patel, Defence Science and Technology Laboratory, UK

Sylvain Pronovost, Carleton University & CAE

Frank Ritter, Penn State University

Venkat Sastry, University of Cranfield

Barry Silverman, University of Pennsylvania

Neil Smith, QinetiQ

LtCol David Sonntag, Asian Office of Aerospace Research and Development

Webb Stacy, Aptima

Mike van Lent, SoarTech

Walter Warwick, Alion Science and Technology

Jason Wong, Naval Undersea Warfare Center

Patrick Xavier, Sandia National Laboratories

Poster & Exhibits Program – Jeanne Eury/Lodestar Group

Sponsors

The BRIMS Conference thanks the following sponsors for their continued support for this year's Conference.

Air Force Research Laboratory (AFRL)

Air Force Office of Scientific Research (AFOSR)

Army Research Laboratory (ARL)

Natick Soldier Research, Development, & Engineering Center (NSC)

National Aeronautics and Space Administration (NASA)

UK Ministry of Defence (MoD)

BRIMS 2011 Best Paper & Best Student Paper Awards

Best Paper Awards:*

★ Towards Adding a Physiological Substrate to ACT-R
Christopher Dancy, Frank Ritter, & Keith Berry (The Pennsylvania State Universityhart

★ Modeling Social Transmitted Affordances: A Computational Model of Behavioral Adoption Tested Against Archival Data from the Stanford Prison Experiment
Benjamin Nye (University of Pennsylvania)

Recommended Reads:

◆ A Comparison of Rule-Based versus Exemplar-Based Categorization Using the ACT-R Architecture
Matthew Rutledge-Taylor, Christian Lebiere, Robert Thomson, James Staszewski, & John Anderson (Carnegie Mellon University) (12-BRIMS-010)

◆ Socio-cognitive Networks: Modeling the Effects of Space and Memory on Generative Social Structures
Changkun Zhao[1], Ryan Kaulakis[1], Jonathan Morgan[1], Jeremiah Hiam[1], Frank Ritter[1], & Geoffrey Morgan[2] ([1]The Pennsylvania State University & [2]Carnegie Mellon University) (12-BRIMS-007)

◆ Evaluation of Two Intelligent Tutoring System Authoring Tool Paradigms: Graphical User Interface-Based and Text-Based
Shrenik Devasani[1], Stephen Gilbert[1], & Stephen Blessing[2] ([1]Iowa State University, [2]University of Tampa) (12-BRIMS-011)

◆ The Organizational Dynamics of Submarine Piloting and Navigation Teams
Ronald Stevens[1] & Trysha Galloway[2] ([1]UCLA, [2]The Learning Chameleon, Inc.) (12-BRIMS-008)

◆ Understanding Sensemaking Using Functional Architectures
Robert Thomson, Christian Lebiere, Matthew Rutledge-Taylor, James Staszewski, & John Anderson (Carnegie Mellon University) (12-BRIMS-013)

Best Student Paper Award:*

★ Towards Adding a Physiological Substrate to ACT-R
Christopher Dancy, Frank Ritter, & Keith Berry (The Pennsylvania State University)

BRIMS
BEHAVIOR REPRESENTATION IN MODELING & SIMULATION

I. PLENARY SPEAKERS

Papers are in order as they appear on the agenda.

Dylan Schmorrow, Ph.D.

Captain, Medical Service Corps, US Navy

Experience

Office of the Secretary of Defense, Deputy Director, Human Performance, Training and BioSystems
October 2008 – Present

Office of the Secretary of Defense, Acting Director, Human Performance, Training and BioSystems
April 2010 – January 2011

Office of Naval Research, Executive Assistant to the Chief of Naval Research
July 2005 – October 2008

Defense Advanced Research Projects Agency, Program Manager
December 2000 – October 2005

Naval Research Laboratory, Technology Integration Chief Scientist
October 1998 – October 2001

Naval Postgraduate School, Assistant Professor
June 1996 – October 1998

USS Dwight D. Eisenhower, Flight Deck Research Officer
December 1994 – February 1995

Naval Air Warfare Center, Branch Head and Project Officer
May 1994 – June 1996

Naval Aerospace Medical Institute, Flight Student
October 1993 – April 1994

Captain Schmorrow is a U.S. Naval Officer in the Navy's Medical Service Corps and has been appointed by the Navy Surgeon General as the Specialty Leader of the Aerospace Experimental Psychologist Community. He is also an Acquisition Professional in the Naval Acquisition Corps and is currently serving in the Office of the Assistant Secretary of Defense (Research and Engineering) as the Deputy Director for Human Performance, Training and BioSystems with purview over the defense technology areas of human performance, medical, man-machine systems, training, civil engineering, environmental quality, and chemical and biological defense. Responsibilities include providing technical leadership, management oversight, policy guidance, and coordination for over $3 billion in research and engineering programs in the DoD to ensure that these areas are focused, relevant, and eminently capable of satisfying current and anticipated defense needs. He provides executive and supervisory leadership and authoritative scientific and technical advice to afford future forces the requisite knowledge, science, and technology for critical warfighting capabilities. In this role, he has established collaborations with the National Science Foundation, the National Institutes of Health, the Department of Homeland Security, the Defense Advanced Research Projects Agency, and the DoD Services to directly include government, academic and industry researchers in advancing these efforts. Additionally, he serves as the OSD Human Social, Culture, Behavior Modeling Program Manager and the Executive Secretary for the Defense Science Board Study on Autonomy. He also leads international efforts to promote and conduct cooperative scientific research and exchange of technical information through the NATO Research and Technology Organisation and is the U.S. National Representative of The Technical Cooperation Program's (TTCP) Human Resources and Performance (HUM) Group.

Education

Jenkins Postdoctoral Fellow, **Applied Cognitive Research**
Naval Postgraduate School – September 1998

Doctor of Philosophy, **Experimental Psychology**
Western Michigan University – June 1993

Doctoral Fellowship, **Experimental Psychology**
Georgetown University – August 1992

Master of Science, **Modeling, Virtual Environments and Simulation**
Naval Postgraduate School – September 1998

Master of Science, **Operations Research**
Naval Postgraduate School – September 1998

Master of Arts, **Philosophy**
Western Michigan University – June 1993

Master of Arts, **Experimental Psychology**
Western Michigan University – August 1990

Bachelor of Science, **Economics and Psychology**
Western Michigan University – June 1989

Dr. Schmorrow received a commission in the U.S. Navy in 1993, completing naval flight training in April 1994. He initially served at the Naval Air Warfare Center Aircraft Division as both a branch head of the biomedical support branch and as lead scientist on an acceleration research project officer in the crew systems department. During this time he deployed on the USS Dwight D. Eisenhower in support of "Deny Flight" and "Provide Promise" Operations in the Former Yugoslavia. He then served as both an Assistant Professor and the John G. Jenkins Postdoctoral Fellow at the Naval Postgraduate School and subsequently as the Chief Scientist for Human-Technology Integration at the Naval Research Laboratory (NRL). While at NRL he was selected as a Program Manager at DARPA responsible for creating and fostering imaginative, innovative, and high-risk research ideas yielding revolutionary technological advances in biomedical and information science and technology in support of the U.S. military. He then was the Executive Assistant to the Chief of Naval Research where he coordinated actions between the Office of Naval Research (ONR) and tenant commands, the Secretary of the Navy, and the Chief of Naval Operations, as well as intergovernmental agencies and international S&T organizations. From 1999 through 2008 he concurrently served as an ONR Program Officer leading medical and human performance S&T programs that transformed promising technologies into operational capabilities; he successfully transitioned numerous prototypes to Navy and Marine Corps acquisition programs. Since 2008 he has had DoD-wide oversight responsibilities for DoD Human Systems Technology areas of personnel selection, training, leadership, cognitive sciences, interface design, personnel protection, combat feeding, human systems integration, and human performance and also served as an Acting Director in the ASD(R&E) Research Directorate from 2010 to 2011 during senior executive transitions. He has authored over fifty scientific publications, lectured internationally in fifteen countries, and edited a dozen professional journals and books. He is a recipient of the Navy's Top Scientists and Engineers Award, as well as both the Society of U.S. Naval Flight Surgeons' Sonny Carter Memorial Award for his contributions to improve the health, safety and welfare of military operational forces and the Human Factors and Ergonomics Society's Leland S. Kollmorgen Spirit of Innovation Award for his contributions to operational neuroscience that led to the founding of the field of Augmented Cognition. His military decorations include the Defense Superior Service Medal, Legion of Merit, Meritorious Service Medal (3 awards), Navy Commendation Medal, Navy Achievement Medal, Armed Forces Service Medal, and NATO Medal.

dylan.schmorrow@osd.mil

Time to Get Real in Sociocultural Behavioral Research and Engineering: The Rubber is Hitting the Road Across the DOD

U.S. Armed Forces have long understood the operational value of understanding the mindset of opposing forces and securing the cooperation and support of local populations. However, the U.S. is now expected to engage foreign populations more routinely, at all operational phases, and across a broader range of mission types than ever before. Success requires being able to anticipate how factors such as culture, society, group identity, religion, and ideology influence the behavior of foes and others in foreign populations. This new reality demands a broader, deeper capability, realized at tactical, operational, and strategic levels, and founded on the social and behavioral sciences:

Since 2008, the HSCB Modeling Program has helped support realization of this vision by funding innovative and rigorous applied research, advanced technology development, and prototypes. That research has helped build the sociocultural behavior science base, but has also been applied to current operations. The Program has also provided thought leadership and helped to bring greater coherence to the many programs and initiatives underway, in part through its *Focus* series of national conferences, which have involved over 600 participants each of the last two years.

Some of the Program's most exciting work is where the rubber hits the road in support of the nation's Combatant Commands (COCOMs). The Program now has technology transition agreements in process or completed with all appropriate geographic commands and the US Special Operations Command (USSOCOM). These agreements will enable transition of research designed to address core US challenges, including strengthening international and regional security, monitoring instability, and countering violent extremism. A prime example of this applied research is the World-Integrated Crisis Early Warning System. W-ICEWS has been developed to provide COCOMs and other operational entities with the ability to better anticipate where instability is imminent, in time to take effective action as appropriate.

Significant progress has been made toward building DoD capability for understanding sociocultural behavior, and there has been delivery of solutions to military end users. However, there is much work to be done. To ensure that R&E investments reflect the U.S. military's strategic priorities and leading operational challenges, the HSCB Program developed the Sociocultural Behavior Capability Areas framework. It calls for building warfighter capabilities to ***Understand*** sociocultural behavior, ***Detect*** relevant sociocultural signals, ***Forecast*** through persistent sensing of the environment, and ***Mitigate*** with measurable COA grounded in the social and behavioral sciences. The framework and recommendations regarding R&E priorities are presented in the document *Sociocultural Behavior Research and Engineering in the Department of Defense Context*. http://www.dtic.mil/dtic/tr/fulltext/u2/a549230.pdf

Ultimately, the test of the knowledge products, technologies, and models produced through DoD sociocultural behavior R&E will be how they contribute to development of the future force, giving analysts, warfighters, and leaders more time and opportunity to do what they do best: out-think and out-innovate adversaries by bringing all instruments of power to bear.

Frank E. Ritter

Professor of Information Sciences and Technology, of Psychology, and of Computer Science and Engineering

http://acs.ist.psu.edu

Frank Ritter's current research is in the development, application, and methodology of cognitive models, particularly as applied to interface design, predicting the effect of behavioral moderators, and understanding learning. With Martin Yeh, he has an iPhone app, caffeinezone, for predicting the time course and effects of caffeine, and his lab is building tutors for the Marine Corps on shooting and combat lifesaving skills. He has helped write and edit several books. A report on applying cognitive models in synthetic environments, was published by the Human Systems Information Analysis Center (HSIAC) as a State of the Art Report (2003), a book on order effects on learning was published in 2007 by Oxford, and he contributed to a National Research Council report (Pew & Mavor, eds., 2007) on how to use cognitive models to improve human-system design.

He is about to publish a book of practical advice on how to run studies given as a tutorial here at BRIMS 2012 (Sage) and the ABCS of what psychology do systems designers need to know (Springer).

Three of his papers on modeling users have won awards, one on high level languages with St. Amant was selected for the "Siegel-Wolf Award for best applied modeling paper" at the International Conference on Cognitive Modeling, and two were selected for recommended reading lists by the BRIMS conference, one on interfaces for models and one on how rules of engagement can be influenced by moderators. The ABCS book has repeatedly won awards at the HCI Consortium annual meeting.

He has served as an external examiner in England for degree programs in cognitive science and knowledge management systems. He currently edits the Oxford series on cognitive models and architectures for Oxford University Press. With Drs. William Kennedy and Bradley Best, he has co-chaired the BRIMS conference and edited special issues of the best papers for Computational and Mathematical Organizational Theory. His work has been funded by ARL, Darpa, DMSO, Dstl (UK), DSTO (Australia), DTRA, and ONR. He spent the Fall 2005 semester as a Senior Fulbright Scholar at TU/Chemnitz in Germany.

Using Behavior Representation Models in Risk-Driven Design

Frank E. Ritter, PhD

Applied Cognitive Science Lab, College of IST, Pennsylvania State University

A report by the National Research Council (Human-system integration in the system development process: A new look. Pew & Mavor, eds., 2007, available free with registration;

http://www.nap.edu/openbook.php?record_id=11893) noted a new way to include human factors in developing systems of systems, the risk-driven spiral model. In this talk I introduce the theory in this report and note a few extensions based on thinking since the report. The report argues that most systems are developed with a mind to what are the riskier aspects of the design and implementation. The report notes how different perspectives have different views of what is risky, and that risk-driven spiral model can be used to organize methods in HCI. (This theory can apply to developing our own models as well.) The report calls for using user models as a shared representation between designers and design stages. The use of models as shared representations in design offers a new outlet and use for user models, but also raised new challenges and repeats old ones, like the ability to build models easily and make them explainable to others. Knowing this report will help modelers understand their own work, find find their audience, and apply their models more effectively.

P. Jeffrey Brantingham

P. Jeffrey Brantingham is Associate Professor of Anthropology at the University of California Los Angeles. He received his PhD in Anthropology from the University of Arizona and held a postdoctoral position at the Santa Fe Institute before joining UCLA. Brantingham was trained as a Paleolithic archaeologist and has conducted extensive fieldwork in Mongolia and China focused on the study of hunter-gatherer adaptations to harsh desert and high altitude environments. It is this interest in the foraging behaviors of hunter-gatherers which he has translated into the study of criminal foraging behavior in complex urban environments, teaming up with mathematicians at UCLA and the Los Angeles Police Department to build mathematical and computational models of crime. Brantingham's work on crime has been funded by the National Science Foundation, NGA, ARO, and ONR. He currently is PI on a AFOSR MURI project *Inferring Structure and Forecasting Dynamics on Evolving Networks*, which is a collaboration among social scientists and mathematicians aimed at building models of social networks that are simultaneously data driven, behaviorally-grounded and mathematically rigorous.

Contact Information:

P. Jeffrey Brantingham
Associate Professor and Vice Chair
Department of Anthropology
UCLA
Tel: 310-267-4251
Fax: 310-206-7833
Email: branting@ucla.edu
http://paleo.sscnet.ucla.edu

Predicting Crime Patterns with Behaviorally-grounded Mathematical Models

It has long been know that crime is non-uniformly distributed in space and is highly dynamic in time. This talk will review the behavioral foundations of mathematical models of crime pattern formation developed at UCLA. Critical to these models are behavioral observations that criminal offenders search locally for their targets. Diffusion limitation, as understood mathematically, therefore plays a key role in crime pattern formation. At a theoretical level, the models lead to unique predictions about how policing interventions may alter, displace or disrupt crime hotspots. A more fundamental question remains, however: Is it possible to accurately predict crime in space and time using such models? The presentation will review preliminary results of randomized controlled field experiments, conducted in collaboration with the Santa Cruz and Los Angeles Police Departments, which show that Predictive Policing mathematical models outperform the best-practices of police crime analysts in forecasting crime.

Peter Pirolli

Peter Pirolli is a Research Fellow in the Augmented Social Cognition Area at the PARC, where he has been pursuing studies of human information interaction since 1991. Prior to joining PARC, he was an Associate Professor in the School of Education at UC Berkeley. Pirolli received his doctorate in cognitive psychology from Carnegie Mellon University in 1985. He is an elected Fellow of the American Association for the Advancement of Science, the Association for Psychological Science, the American Psychological Association, the National Academy of Education, and the Association for Computing Machinery Computer-Human Interaction Academy.

Making Sense of Sensemaking: From Neural to Social Networks

Sensemaking is a natural kind of human activity in which large amounts of information about a situation or topic are collected and deliberated upon to form an understanding that becomes the basis for problem solving and action. It goes beyond simply finding information. It is involved in learning about new domains, solving ill-structured problems, acquiring situation awareness, and participating in social exchanges of knowledge. Examples of sensemaking tasks include understanding a health problem to make a medical decision, understanding the weather to make a forecast, intelligence analysis to identify strategic threats, and the collaborative collection and understanding of an emergency by first responders. The opportunity (and challenges) are enormous for developing a scientific foundation to better support complex sensemaking in the digital world, whether by individuals or online groups and communities. As Allen Newell suggested, human behavior can be viewed as a hierarchically organized set of systems rooted in physics and biology at one end of the spectrum and large-scale social and cultural phenomena at the other end. Different layers of phenomena dominate at different levels of this organization: neural, psychological, economic, and social--just to name the more familiar. In this presentation I will sample theories and models addressing different layers of sensemaking phenomena ranging from individual cognition to online social communities. I argue that integrated multi-scale theories are needed to understand and predict how microscale factors at the level of the individual percolate upwards to yield macroscale emergent phenomena at the social level and how macroscale social factors percolate downward to shape individual sensemaking.

II. SYMPOSIUM

Papers are in order as they appear on the agenda.

Accelerating the Evolution of Cognitive Architectures

Organizer
Kevin Gluck
Air Force Research Laboratory
711 HPW/RHAC – Cognitive Models and Agents Branch
2620 Q Street, Bldg 852
Wright-Patterson AFB, OH 45433-7955
937-938-3552
kevin.gluck@wpafb.af.mil

The idea that the mind should be rigorously studied in modeling and simulation traces its intellectual roots to the landmark Newell, Shaw, and Simon (1958) paper, in which they proposed information processing models implemented in computer code as explanations of human problem solving capabilities. Fifteen years later Newell (1973) adopted the stronger position that these information processing models must be developed as unified theories of cognition in order to achieve the desired goal of understanding the human mind. In the nearly four decades since, this idea has motivated dozens of new research programs intending to develop integrating, unifying theories, sometimes called *cognitive architectures* – see Byrne (2003), Gray (2008), Langley, Laird, and Rogers (2009), Taatgen and Anderson (2009), and Gluck (2010) for introductions and overviews on this topic. Cognitive architectures are broad, domain-general theories of the mechanisms and structures that enable mind and intelligent behavior. Often overlooked by those not working within cognitive architectural theory is the fact that they are also *evolving*, as functional capabilities expand and as various explanatory mechanisms are evaluated and subsequently incorporated, adapted, or discarded (Cooper, 2007). The breadth and depth of the accomplishments in cognitive architectural theory to date comprise an impressive collection of scientific contributions. While acknowledging and celebrating these achievements, we must also admit to some frustration and concern regarding the slow pace of progress in these integrative systems. This symposium is motivated by the idea that the time is right for us to reconsider the formalisms, methods, and technologies we use to develop and evaluate cognitive architectures, with the goal of accelerating their evolution. Each presenter will explicitly address some of the factors that hinder progress in this area and propose changes in thinking or approach that will overcome those hindrances.

This symposium should be of interest to a broad cross-section of the BRIMS community for at least two reasons. First, there is the fact that the requirement for formal computational implementation and the use of modeling and simulation aligns cognitive architectural theory and progress with the core mission of the BRIMS conference, as described in the call for papers. Second, the development of cognitive architectures is often motivated by an interest in *application*. That is, an objective of cognitive architects is often that the architectures have some applied utility. This may not be universally true, but the evidence for this is clear in some cases. For example, Anderson (1976) actually ends his book introducing the ACT theory with a statement of the importance of application for his research program by saying, “I would like to conclude this chapter with a remark about one of the ultimate goals I have set for my research efforts . . . that is, that it produce a theory capable of practical applications.” (p. 535). Newell’s (1990) position was that, “Applications are an important part of the frontier of any theory. . . . A unified theory of cognition is the key to successful applied cognitive science.” (p. 498). A third example is found in the EPIC architecture, for which some of the earliest publications (Kieras & Meyer, 1997 ; Kieras, Wood, & Meyer, 1997) make it clear that applications in system design served an important motivational role in its creation. The fact that this motivation persists is clear on the EPIC website, which states that EPIC is, “. . . for constructing models of human-system interaction that are accurate and detailed enough to be useful for practical design purposes.” The emphasis among cognitive architects on application opportunities aligns this symposium with the BRIMS community’s more applied researchers and technologists.

How Can We Accelerate the Evolution of Cognitive Architectures?

Richard L. Lewis
University of Michigan
rickl@umich.edu

If we take the start of contemporary cognitive science to be the mid 1950s, it is sobering to note that cognitive architectures have been pursued for about half of the field’s existence: it has been nearly 30 years since the publication of Anderson’s *The Architecture of Cognition*, and over 20 years since Newell’s *Unified*

Theories of Cognition. It seems potentially useful then to reflect about the state and rate of progress, and ask whether our current approaches to advancing cognitive architecture theory and practice are serving the field well, or in need of substantial modification. Newell considered a very similar question (what is required to move unified theories of cognition forward?) at the end of his 1990 book, and I have found it useful to revisit his injunctions, because they remain relevant:

N1. There should be many unified theories, at least for a while.
N2. We should develop consortia (it takes relatively large communities to work on a cognitive architecture).
N3. Be synthetic—incorporate local theories.
N4. Be prepared to modify existing theories—even strongly and in radical ways.
N5. Make UTCs easy to use.
N6. Acquire domains of application.

The substantive points I wish to make are related to these injunctions and concern diagnoses of problems and conjectures for remedies. I will advance the following specific claims about problems:

P1. When it comes to exploring architectural theory, practice in the community has been too conservative and slow (N4), in part because it has not sufficiently exploited and kept pace with advances in local theory (N3).
P2. When it comes to making UTCs easy to use (N5), the community has tended to focus on improving software artifacts and tools (e.g. to make it easier to program architectures), but this focus on software has been misplaced. It does not address the key barriers to the use and programming of cognitive architectures, which are fundamentally theoretical, and not issues of software engineering. A major theoretical barrier is the perceived increase in degrees of freedom in accounting for data that is introduced by the architectural separation of strategy from fixed structure.
P3. When it comes to acquiring domains of application (N6), standard practice has been to build relatively large models of relatively complex task domains, but this practice is slow and has not led to significant cumulative scientific or applied benefit.

I will make the following conjectures about possible remedies for these problems:

R1. Exploit specific advances in other areas of computational and mathematical cognitive science and machine learning—especially control theory and decision theory perspectives—which will go some way toward addressing (P1) and (P2).
R2. Adopt a "lighter-weight" approach to the construction of cognitive architecture software artifacts that places greater emphasis on theoretical transparency (in part enabled by advances exploited in R1), a clear (even formal) specification of the adaptive/behavioral problem of interest, and rapid architecture exploration.

I will illustrate these remedies briefly with modeling results in one or two "basic science" domains and one "applied" domain. (Candidate domains include eye-movement control in reading, interference in short-term memory, "fast-and-frugal" decision making heuristics, and control of attention in piloted aircraft.)

Guidelines for the Design of New Cognitive Architectures

Sashank Varma
University of Minnesota
sashank@umn.edu

Cognitive architectures are critical for the design of veridical models of behavior (Anderson, 1983; Newell, 1990). Therefore, the relatively small size and limited diversity of the population of architectures in cognitive science have been limiting factors on the evolution of fitter architectures. The question, then, is how to increase the number and diversity of available architectures (Varma, 2011)? This talk presents guidelines for the construction of new architectures. These guidelines are drawn from the design community, and supported by events in the history of cognitive architecture and computational modeling.

1. First, the architect invents a new style of cognitive information processing. This is a creative or imaginative act.
2. Next, the architect embodies the new style in a new architecture – expresses the desired function in a new form.
3. Finally, the architecture is made available to the cognitive science community. Those members who are transformed by its new view on cognitive information processing – who "actively receive it" – have their research programs reshaped and redirected.

These guidelines have been part of the "black art" of architectural design for the past 30 years, known to only a handful of researchers. This is perhaps one reason why the population of architectures has remained small in size and lacking in the diversity required for evolutionary progress. Public dissemination and critical discussion of these guidelines will enable researchers located outside "architectural hotspots" such as

Pittsburgh and Ann Arbor to contribute to the population of architectures. Increasing its size and diversity promises to accelerate progress towards better computational models of behavior.

Towards Functionally Elegant, Grand Unified Architectures

Paul S. Rosenbloom
University of Southern California
rosenbloom@usc.edu

When developing cognitive architectures, the ultimate goal is typically a unified theory of intelligent behavior, with the working focus then being on integrating across the capabilities implicated within central cognition, and the result being a unified architecture for cognition. What can be called a *grand unified architecture* sets the bar higher, striving to also include the key non-cognitive aspects of intelligent behavior, such as perception, motor control, personality, motivation and affect. Such architectures can further be considered *functionally elegant* if they provide the requisite breadth of functionality in a simple and theoretically elegant manner, yielding a form of cognitive Newton's laws that provides broad coverage from interactions among a small set of general principles/mechanisms. The pursuit of functionally elegant, grand unified architectures provides a challenging research path, yet one that points the way towards rapid progress beyond today's state of the art, even within the more traditional cognitive focus; and which should also support both deep science and useful systems.

I am currently approaching this goal by rethinking architectures from the ground up, leveraging the interactions between a pair of very general mechanisms – graphical models (factor graphs, in particular) and piecewise continuous multivariate functions – to yield a parameterized space of state-of-the-art capabilities over the processing of symbols, probabilities and signals (Rosenbloom, 2011a-b). The availability of this broad parameterized space promises to accelerate the evolution of cognitive architectures by facilitating the exploration of a wider range of the requisite capabilities and their variations; and without the need to explicitly implement a whole new module for each. Work to date – much of which will be summarized here – demonstrates that within the resulting space can be found: standard flavors of long-term memory, such as a procedural rule-based memory and declarative semantic and episodic memories, plus other variations and blends (Rosenbloom, 2010); forms of knowledge-based (Rosenbloom, 2011c), decision-theoretic (Chen *et al.*, 2011) and social problem solving; perception (Chen *et al.*, 2011) and mental imagery (Rosenbloom, 2011d); and key bits of language processing. Much more is still required on many of these topics, and additional capabilities must also be added, but the already proven applicability of graphical models to many of these problems shines a bright light on the path towards their rapid incorporation into such a grand synthesis. It also may help understand other topics – such as personality, motivation and affect – that have not previously been investigated via these kinds of techniques. For some capabilities – such as learning – more principles/mechanisms will likely be required, but functional elegance still looks to be within reach, with the inclusion of only a small number of additional general principles/mechanisms.

Acknowledgments

The participants in this symposium thank the following organizations for the financial support that enabled participation in this symposium and the substantive research on cognitive architecture described therein: the Air Force Research Laboratory's Human Effectiveness Directorate and Office of Scientific Research, the United States Army and the Army Research Laboratory, the Defense Advanced Research Projects Agency, the Federal Aviation Administration, the National Aeronautics and Space Administration, the National Science Foundation, and the Office of Naval Research.

References

Anderson, J. R. (1976). *Language, memory, and thought*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1983). *The architecture of cognition*. Mahwah, N.J.: Lawrence Erlbaum.

Byrne, M. D. (2003). Cognitive architecture. In J. Jacko & A. Sears (Eds.) *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications* (pp. 97–117). Mahwah, NJ: Lawrence Erlbaum .

Chen, J., Demski, A., Han, T., Morency, L-P., Pynadath, D., Rafidi, N. & Rosenbloom, P. S. (2011). Fusing symbolic and decision-theoretic problem solving + perception in a graphical cognitive architecture. *Proceedings of the 2nd International Conference on Biologically Inspired Cognitive Architectures* (pp. 64-72). Arlington: VA: IOS Press.

Cooper, R. P. (2007). The role of falsification in the development of cognitive architectures: Insights from a Lakatosian analysis. *Cognitive Science, 31*(3), 509-533.

Gluck, K. A. (2010). Cognitive architectures for human factors in aviation. In E. Salas & D. Maurino (Eds.) *Human Factors in Aviation, 2nd Edition* (pp. 375-400). New York, NY: Elsevier.

Gray, W. D. (2008). Cognitive architectures: Choreographing the dance of mental operations with the task environment. *Human Factors*, *50* (3), 497–505.

Kieras, D., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction*, *12*, 391–438.

Kieras, D. E., Wood, S. D., & Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *ACM Transactions on Computer-Human Interaction*, *4*, 230–275.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*, 141-160.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual Information Processing* (pp. 283-308). New York: Academic Press.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA.

Newell, A., Shaw, J. C., & Simon, H. A. (1958). Elements of a theory of human problem solving. *Psychological Review*, *65*, 151–166.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Rosenbloom, P. S. (2010). Combining procedural and declarative knowledge in a graphical architecture. *Proceedings of the 10th International Conference on Cognitive Modeling* (ICCM 2010).

Rosenbloom, P. S. (2011a). Rethinking cognitive architecture via graphical models. *Cognitive Systems Research*, 12(2), 198-209.

Rosenbloom, P. S. (2011b). Bridging dichotomies in cognitive architectures for virtual humans. *Proceedings of the AAAI Fall Symposium on Advances in Cognitive Systems*.

Rosenbloom, P. S. (2011c). From memory to problem solving: Mechanism reuse in a graphical cognitive architecture. *Proceedings of the 4th Conference on Artificial General Intelligence* (pp. 143-152). Mountain View. CA: Springer.

Rosenbloom, P. S. (2011d). Mental imagery in a graphical cognitive architecture. *Proceedings of the 2nd International Conference on Biologically Inspired Cognitive Architectures* (pp. 314-323). Arlington, VA: IOS Press.

Taatgen, N. A. & Anderson, J. R. (2009). The past, present, and future of cognitive architectures. *topiCS in Cognitive Science*, 1-12.

Varma, S. (2011). The design and evaluation of cognitive architectures. *Cognitive Science*, *35*, 1329-1351.

III. PAPERS

Papers are in order as they appear on the agenda.

Modeling Explicit and Implicit Ties within an Organization: A Multiple Model Integration

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

Keywords:
Socio-Cultural Modeling, Model Comparison, Performance Prediction

ABSTRACT: *We present a simulation designed to capture the impact of both explicit authority ties and implicit socialization ties on the performance of an organization adapting to a turbulent world. We present a summary of three key models which informed our approach. We then outline and describe the operation of our resulting simulation. Using an experiment which manipulated both the authority network structure and the stress each organization placed on socialization, we show that socialization has a non-linear impact on peak organizational performance and on the performance of top management. We also demonstrate that the authority structure has some impact on the performance of both the organization in toto, as well as on top management.*

1. Introduction

An organization is a group of people assembled in order to complete one or more tasks. Organizations must adapt to dynamic conditions to survive. One method organizations use to adapt over time is turnover. March (1991) demonstrates that this strategy should be effective in dynamic environments, arguing that an organization should attempt to ignore its own understanding of the environment when hiring new actors because new actors bring new information. Morgan and Carley (2011) examined what effect adding a selective hiring function to March's Mutual Learning Model might have by integrating a model of social reflexivity. Drawing from Morgan, Morgan, and Ritter's (2010) work on social reflexivity in combat units, Morgan and Carley (2011) confirmed that the more random the selection method – the more effective the turnover strategy is in confronting a dynamic environment.

These models (March, 1991 and Morgan & Carley, 2011) do not address, however, how organizational hierarchies or structures influence performance. Many organizations exhibit multiple levels of hierarchy. Further, one can argue that hierarchy, itself, is a response to a dynamic environment. Sub-units pay attention to key issues relevant to their tasks and report to management as and when necessary.

The Garbage Can Model (Cohen, March, and Olsen, 1972) examines organizational decision-making in terms of the flows of information, people, and problems. Carley (1986a, 1986b) extended the Garbage Can Model with hierarchical ties, using it to model naval operations.

There is a rich body of work that considers the role of both formal and informal ties (Selznick, 1948; Oh, Chung, and LaBianca, 2004). Although it is possible to consider representing explicitly both formal and informal ties via multiplex networks (with multiple ties of authority, friendship, and acquaintance, among others, represented at once), modeling even a reasonably large organization via such explicit ties quickly becomes infeasible. Further, how formal and informal ties operate to transfer information remains an open research question. Instead, we see the Hierarchical Garbage Can Model (adapted from Carley, 1986a) as a method of representing formal ties, and the Mutual Learning Model (March, 1991) as a method of representing informal ties. Combining these models provides us a tractable method of representing an organization with both kinds of ties.

2. The Simulations

In this section, we will review the assumptions, constraints, and products of each of the pre-existing models. We will briefly review each to illustrate the processes used in the integrated model. The three models being integrated are March's Mutual Learning Model (March, 1991), the Hierarchical Garbage Can (Carley, 1986), and the Participation Model (Morgan, Morgan, and Ritter, 2010). In previous work, Morgan and Carley (2011) integrated the Mutual Learning Model and the Participation Model but we will outline the original model.

2.1 The Mutual Learning Model - Modeling Implicit Connections

The Mutual Learning Model (March, 1991) is an intellective agent-based model from the organizational literature. It posits that there is an external environment, represented as a *c*-tuple of values either 1 or -1 that organizations must adapt to in order to perform well. March suggests that organizations with a more accurate understanding of the external environment will perform better than organizations that do not. Organizations do not directly perceive the environment, but instead infer the characteristics of the environment from their members, learning from high-performing members to construct what is referred to as the organizational code. The organization begins with a blank organizational code (all 0s), and thus has no inherent bias. Each member of the organization is an agent and has their own *c*-tuple of values – which represents their views on the environment. Once the organization has inferred a particular bit attribute of the environment, it socializes its members to agree with its stances. Since the organization may infer from its members incorrectly, this socialization can inhibit the performance of individuals, but the socialization mechanism is necessary to develop the distinct views of high performers.

Without turnover, the Mutual Learning Model quickly reaches equilibrium – where all agents and the organization have identical values in their respective *c*-tuples – and thus no further learning or socialization will occur. If the environment continues to change after equilibrium is achieved, the organization will be unable to adapt to these changes and the organization's performance will degrade.

March presented turnover as a mechanism for combating this degradation of performance, showing that even modest turnover allows the organization to maintain performance as the environment continues to change. March cautions, however, that turnover is only effective when the organization selects individuals essentially at random from all aspects of the environment that can feasibly change. We add the caution 'that can feasibly change' because, for example, tax laws may change but all accountants are likely to need basic math skills.

Although this model is insightful and influential in the organizational literature, it presents an organization as a collection of individuals without formal ties. The organization has no inherent structure, and organizations are structured, more or less effectively, in order to cope with a changing environment. Further, organizations are composed of individuals, which often find it difficult to hire individuals at random as opposed to picking actors much like themselves, phenomena often described as homophily, or preference for same. For a discussion of homophily, please see McPherson and Smith-Lovin, 1987.

2.2 The Hierarchical Garbage Can - Modeling Explicit Authority Connections

The Hierarchical Garbage Can (Carley, 1986a) is an emulative agent-based model based on the theoretical work of Padgett (1980) and implemented in a tool called GARCORG (Carley, 1986b). The model posits that an organization has multiple tiers with its lowest tier, team members, dedicated to detecting change in the environment. Each higher tier has access to the insights of the lower tiers, although access to the findings of lower tiers may be blocked either due to access constraints or a perceived lack of salience from the higher tiers.

Each team member is responsible for paying attention to a particular issue, but the organization may not be able to effectively access a team member's findings or discount the importance of the issues to which a team member is assigned. Because the Hierarchical Garbage Can does not characterize an external environment, these assessments are made based on the flow of information higher levels are able to perceive from the team member, and can result in the removal even of a high performing worker due to structural flaws in the organization. Areas where team members are routinely replaced are labeled as problem spots, and indicate an area of the organization that deserves attention.

This model explicitly represents formal authority ties and presents a useful account of structural flaws in organizations. However, the role of implicit ties in organizations can moderate the organization's performance through the actions of boundary spanners (Oh, Chung, and LaBianca, 2004). Nevertheless, implicit ties within the organization can be difficult or impossible to discern, and the information transference capabilities of the differing kinds of informal ties are not well known. Thus, the Mutual Learning model presents a method of representing these informal ties in aggregate. Individuals do not have specific characteristics in the Hierarchical Garbage Can Model (other than the spot they fill in the organization) and thus the hiring or transfer of individuals was an aspect of the process not considered in the original model.

2.3 The Participation Model – Introducing Actor Bias

The Participation Model (Morgan, Morgan, and Ritter, 2010) is a mathematical model designed as an overlay. This model is focused on the behavior of individuals and how their behavior changes due to influence of fellow team members, team leaders, and the objects of their behavior. The work focused on the highly variable performance of small combat teams, and drew on a survey of relevant sociological and psychological literature to suggest a reflexive mechanism to moderate behavior of individuals across many contexts. These contexts may involve both positive and negative actions towards objects. Their results showed a pattern consistent with documented historical records, with the reflexive mechanism decreasing overall combat performance while reliably increasing the variation in results.

Morgan and Carley (2011) applied this work to the Mutual Learning Model, using the bias mechanism to influence hiring of individuals. Whereas the original work focused on the physical distance between actors, this work focused on the social distance between actors based on their shared perceptions of the environment. As March (1991) predicted, the social reflexivity mechanism tends to decrease the efficacy of turnover as a mechanism to confront changes in the environment. Further work has revealed that the more introspection that is applied to the hiring process, the less effective the turnover mechanism.

3. The Integrated Simulation, the Unified Hierarchical Model

In this section, we describe the integrated simulation, which takes aspects of each of these models. This model is an intellective agent-based simulation. Since both the Mutual Learning and the Hierarchical Garbage Can model the operation of organizations and present quite different pictures of the workings of an organization, the integration was not entirely straightforward. A larger aim of this work is to develop a method for integrating multiple models related to similar phenomena, even models that concern the same phenomena at different granularities. Critical to this integration approach is to identify key assumptions of each model and allow the operation of the other models to inform these assumptions. All assumptions of each model must be either assumed by the final integrated model or explained by the inter-operation of those models. We will summarize the assumptions and processes of each model at the conclusion of this section in Table 3.1.

Like the Mutual Learning Model, this model supposes both an environment with multiple aspects and an organization attempting to optimize its performance within that environment. Each aspect, as in the Hierarchical Garbage Can, is matched to a team member. Above team members, there are team leaders, group leaders, and a single CEO actor. As in the Hierarchical Garbage Can, we have four tiers, although four tiers do not represent a strong commitment of the model but instead is intended to present a sufficiently deep organization to allow substantial structural variation. Every member of the organization is a staff member. All staff members have their own perception of environment. Every staff member must have at least one tie to the next higher level of the organization (except the CEO), but may have additional authority links (i.e., a team leader may report to multiple group leaders, and a team member may report to multiple team leaders).

As environment bits change, team members in charge of that issue have an opportunity to perceive the change in the environment. If the team member becomes aware of the change, higher levels of the management hierarchy also become immediately aware of the change, subject to their ability to access their subordinates work or their interest in that work. Thus, with this mechanism and the one that follows, changes in the environment tend to cause changes in the organizational code.

After the previous process concludes, the Mutual Learning Model mechanisms are used to develop an organizational understanding of the environment and to socialize actors. Staff members are evaluated based on the portion of the organizational code they are responsible for – and are replaced if that portion of the organization's code is incorrect for more than a consecutive number of turns defined by a grace period. New staff members will be hired by one or more team leaders using the work developed in Morgan and Carley (2011), including at least one team leader with oversight for that position.

Based on this description, we can characterize both the original models and the new model, which we do in Table 3.1.

Table 3.1 Shared model characteristics between the Mutual Learning Model (MLM), the Hierarchical Garbage Can (HGC), the Participation Model (Par), and the new simulation, the Unified Hierarchical Model (UHC)

Characteristic	MLM	HGC	Par	UHC
Org in an environment	♦			♦
Environment changes over time	♦			♦
Org learns from agents	♦			♦
Org socializes agents	♦			♦
Agents leave Org at random	♦			
Org replaces agents at random	♦	♦		
Org has explicit authority ties		♦		♦
Team members generate info		♦		♦
Information travels along ties		♦		♦
Information transfer has error		♦		♦
Org removes under-performers		♦		♦
Org can have structural flaws		♦		♦
Explicit access constraints to info		♦		
Context moderates agent action			♦	♦
Dyad distance moderates action			♦	♦
Spatial distance moderates action			♦	
Social distance moderates action				♦
Agents implement homophily bias				♦
Committee makes hiring choice				♦
Org accuracy measured over time	♦			♦
CEO accuracy measured over time				♦
Structural Flaws tracked over time		♦		♦

In the remaining sub-sections, we will describe in more algorithmic detail the initialization, operation, and outcomes of the new integrated model. Throughout these sub-sections, we will define the usage and purpose of each of the following variables, summarized in Table 3.2.

Table 3.2 Summary of key variables in the Unified Hierarchical Model

	Purpose
c	Environmental complexity – and also indicates the number of team members the organization possesses, $c > 0$
m	Team leaders in the organization, $c \geq m > 0$
g	Group leaders in the organization, $m \geq g > 0$
r	The probability of having multiple authority ties, $1 > r > 0$
s	The length of the simulation in turns, $s > 0$
l	Grace period (in turns) before an organization terminates a team member that is underperforming, $s \geq l > 0$
t	Probability that environment bits flip value from turn to turn.$1 > t > 0$
p	Probability that a responsible team member will perceive a change in the environment, $1 > p > 0$
u	Supervisor capacity for updates per turn, $c > u > 0$
a	Informs the probability of the organization learning from high performers, $1 > a > 0$
o	Probability that staff members change their bits to match the organizational code, per bit, $1 > o > 0$

3.1 Initialization

To initialize the simulation, the modeler must make several decisions. They must determine how complex the environment is – that is, how many aspects of the environment are likely to change over time. We label the quantity of these aspects as c.

In this work, we assume that each aspect of the environment has a corresponding team member that is responsible for tracking that aspect, so c represents both the complexity of the environment and the number of team members the organization will possess.

The modeler must also decide on the number of group managers (g) and the number of team leaders (m). There must be at least one group manager and one team leader. It is possible for there to be any number of team leaders and group managers, but we will assume an upper bound of the quantity of staff members of the tier below. So m should not exceed c, and g should not exceed m.

Finally, some staff members report to multiple people. This possibility is informed by the quantity r, which should range between 1 (inclusive) and 0 (exclusive). When determining whether an agent should have an additional authority tie, assuming that the agent does not already report to every actor in the higher tier, the probability of a new authority tie (i.e., of an additional "reports-to" relationship) is equal to r raised to the power of the number of current authority ties the actor has. Thus, every staff member (except the CEO) will form at least one authority tie, but team members and team leaders may form additional authority ties. If $r = 1$, then every team member will report to every team leader and every team leader will report to every group leader. Figure 3.1 compares an organization with a very small r value to an organization with $r = 1$.

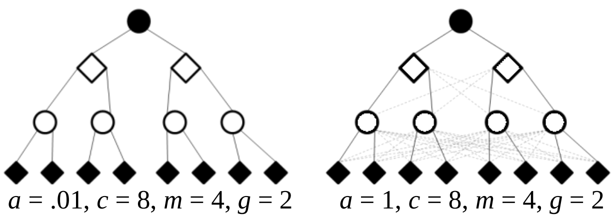


Figure 3.1 The redundancy variable indicates the probability that an actor will have multiple "report-to" relationships in the organization

3.2 Operation

Once initialized, the simulation must proceed through a number of turns, defined by the quantity s. Each turn is composed of several phases:

1. Environment Changes
2. Formal Authority Information Transfer
3. Organizational Inference
4. Implicit Socialization
5. Turnover

Phase 1, *environment changes*, addresses changes in the external environment. Each environmental bit has a probability, t, of flipping in value. This probability should be between 0 and 1. Conventionally, t should be set towards the lower range, indicating that any particular aspect of the environment is more likely to maintain its current state rather than change.

Phase 2, *formal authority information transfer*, handles the direct transfer of information among staff members along the explicit authority ties. For each environment bit that changed in Phase 1, the team member responsible for that aspect has a probability, p, of recognizing that the environment has changed. If the team member observes the change, he reports it to all of his team leaders. Each manager (team leaders, group leaders, and the CEO) has a capacity, u, for receiving updates. If that capacity is exceeded, the update is ignored. Otherwise, the manager updates their own understanding of the world and passes along the update to their superiors. Figure 3.2 illustrates this mechanism.

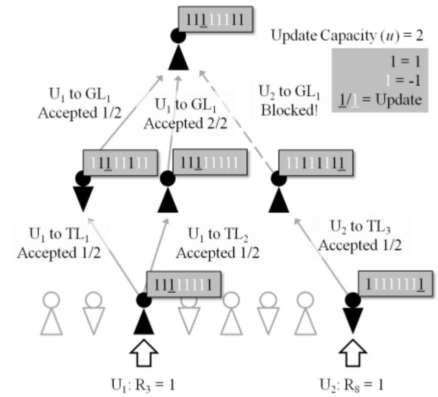


Figure 3.2 Supervisors have limited capacity for updates. Staff Member 8's update was blocked at the Group Leader level.

In Phase 3, *organizational inference*, the organization refines its organizational code based on the individual perceptions of its staff members. The organization identifies all high-performing staff and generates a probability, per environment bit, that the organization will infer that the staff's majority opinion is correct. This probability is informed both by the quantity, a, and the level of consensus for the correct setting of that bit among high performers. Higher values of a indicate an organization willing to take more risks, whereas lower values of a indicate a more conservative firm profile.

In Phase 4, *implicit socialization*, the organization socializes actors to agree with its stance on each bit. This stance may be at odds with the environment and may even defy the information they previously transmitted in Phase 2! The probability of socialization per aspect is defined by the quantity o. Higher values of o indicate an organization that stresses socialization, focusing on making sure that new hires rapidly transition into useful members of the staff. Lower values of o indicate an organization with a longer view, allowing individuals more time to mature into the organization.

In Phase 5, *turnover*, the organization replaces under-performing staff. Team members are replaced if the aspect of the organizational code for which they are responsible is incorrect for longer, consecutively, than a defined grace period. This grace period is captured by the quantity l, and should be shorter than the total length of the simulation, s. Staff members are replaced, using the method described in Morgan and Carley (2011), by a committee of three team leaders who have an implicit bias towards new hires similar to themselves. The committee is first populated by team leaders which have management authority over the position. If there are not enough team leaders to form a committee of three, then the committee includes other group leaders and, if necessary, the CEO.

3.3 Outcomes

In this simulation tool, there are several outcomes of interest. We retain the Mutual Learning Model's organizational performance metric, measuring the accuracy of the organizational code. We also track and report the CEO's performance, measured as the accuracy of their personal code against the environment as a related but distinct measure. Referencing the Hierarchical Garbage Can, we will keep track of the organization's minimum, maximum, and average number of rehires across all team member positions per turn. Because we are also interested in the effect of an organizational hierarchy on hiring, we track the average number of individuals examined per opening per turn.

4. Virtual Experiment

In this section, we will discuss a virtual experiment that examines the tradeoffs between two different authority structures with three levels of implicit socialization.

In this experiment, we keep the number of mid-level managers (team leaders and group leaders) constant, but vary their proportions. We define two structures, represented here as ACME and ZENO. ACME has ten team leaders but only two group leaders. ZENO has six team leaders and six group leaders. We expect information to travel differently through these structures.

In ACME, environment changes will usually be successfully communicated to team leaders. Group leaders may, by contrast, be overwhelmed by the updates they receive from their team leaders. Updates to the CEO will often be successfully transmitted.

In ZENO, more updates from team members will escape the capacity of their leaders. However, nearly all updates received by team leaders will also be successfully related to group leaders. The CEO's understanding may lag, as this actor has too many group leaders to pay attention to them all. ZENO is likely to be a less efficient organization because the group leader level seems likely to have a great deal of spare capacity, and may even have a few "empty-suits", group leaders without any team leaders reporting to them.

Some research (Oh, Chung, and LaBianca, 2004) suggests that inter-group socialization, in moderation, improves performance, but that socialization can be over-emphasized and thus harm performance. Consequently, we examine the interplay of the socialization and formal authority information transfer mechanisms to determine if we could replicate this finding. We used three distinct levels of *o* for this experiment.

All other factors were held constant and are listed for completeness in Table 4.1. For each combination of experiments, we ran 200 replications of the simulation. In total, we ran 1200 separate instances of the simulation.

Table 4.1 Virtual Experiment Variables and Constants

Variable	**Values**	**#**
Structure	ACME: $m = 10$, $g = 2$; ZENO: $m = 6$, $g = 6$	2
Socialization (o)	0, .05, .9	3
Constants		
Complexity (c)	50	1
Redundancy (r)	.3	1
Simulation Length (s)	100	1
Grace Period (l)	5	1
Turbulence (t)	.05	1
Perception Acc (p)	.9	1
Update Capacity (u)	3	1
Staff Agreement (a)	.5	1
	Total Combinations	6

5. Results

In this section, we examine the results of the virtual experiment defined in the previous section and summarized in Table 4.1.

As expected and can be seen in Figure 5.1, ZENO tends to underperform ACME over time when socialization is held constant. Both firms reach a peak performance level and then begin to degrade. This degradation, despite turnover, is in large part because turnover is only applied to team members, as opposed to all staff. New individuals enter the organization, but the managerial class, which outperforms these new hires, suppresses their ideas to the organization's detriment.

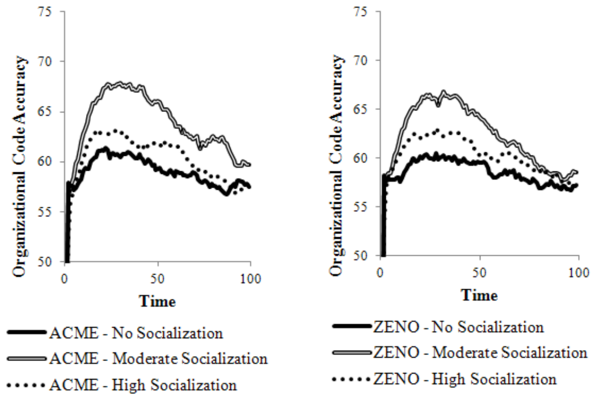


Figure 5.1 ACME and ZENO have similar organizational performance characteristics, although ZENO tends towards lower performance than ACME over time.

Socialization has a large and non-linear effect on performance, as can be seen in Figure 5.2. The chart shows the (averaged) peak performance score each firm achieved for all six conditions. Although explicit authority structure may have some small impact on peak performance, the implicit socialization probability has a large and non-linear impact.

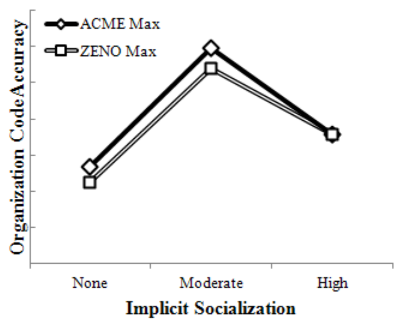


Figure 5.2 As suggested by Oh, Chung, and LaBianca (2004, p. 869) - an organization which supports cross-team socialization will perform better than one that either forbids it or mandates it

The impact on the CEO also suggests that a moderate amount of socialization across authority ties is useful. In Figure 5.3, we can see the averaged accuracy of the CEO's perceptions. Although the CEO in the highly socialized environment achieves their peak performance more rapidly than in the moderate case, their peak is lower. The graph, for both firms, also underlines the value of socialization. A CEO forced to rely only on the information relayed by their direct subordinates reaches their peak much later and that peak is much lower. The ACME CEO appears to retain relevance longer than their ZENO counterpart, possibly due to ZENO's less efficient group leader corps.

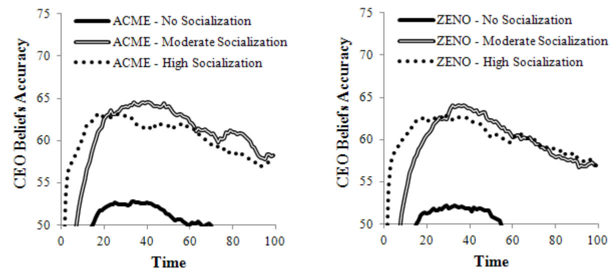


Figure 5.3 CEOs tend towards much better performance when they have more information than their direct authority ties alone can relate. The moderate socialization case peaks later, but higher, than the high socialization case.

The other metrics we tracked showed results closely correlated to the performance of the firm. Firms that are higher performing tend to need to replace fewer team members. Applicant review numbers followed closely the pattern found previously in Morgan and Carley (2011), where organizations that stressed socialization review many more candidates for open positions than organizations that did not.

6. Discussion

In this work, we integrated multiple models of interest related to predicting and understanding organizational performance. We presented two alternative visions for how organizations confront and adapt to change as summarized in the Mutual Learning (March, 1991) and Hierarchical Garbage Can (Carley, 1986a) models. We unified these two approaches to produce a new model that represents both explicit authority ties, as well as implicit relationships such as joint social activities or group co-membership. Using this new model, we replicated a result suggested by the business literature (Oh, Chung, and LaBianca, 2004) on how socialization can both help and harm an organization.

We also demonstrated that our model can suggest impacts of structural choices on organization outcomes. However, our organizational structures are noisy and highly variable because we took a stochastic approach to defining the authority structure. By contrast, GARCORG (Carley, 1986b) presented users with fully defined explicit structures based on naval operations. Our stochastic approach may also have resulted in noisy structures that were not sufficiently different (across the 200 averaged instances) when viewed through our average, maximum, and minimum metrics. It is also possible that those metrics are simply too coarse to identify prevalence towards hot-spots in either ACME or ZENO. We do not believe this work well characterizes the relative impact of structure versus socialization, but suggests that such a characterization may be possible in the future.

The Participation Model was used in the model as defined in Morgan and Carley (2011) to inform the hiring process, but the Participation Model also offers many insights that could affect managerial performance. One purpose of leadership is to force subordinates to confront the environment and perform their tasks despite of it. Morgan, Morgan, and Ritter (2010) present the work in an analogous, albeit combat-related, context. Thus, the variable *p*, rather than remaining constant for all staff, should relate to the quality and quantity of leadership.

In the current simulation, turnover applies only to under-performing team members. In practice, people at all levels of the firm leave those organizations for many reasons. Future iterations of this model will apply turnover to all staff members. Nevertheless, most organizations approach the hiring of top level executives differently from hiring at lower tiers of the organization - if only because no higher tier exists to make hiring choices.

This successful multi-model integration suggests that our larger approach towards multi-modeling and multi-level modeling may be of some use to the larger community.

7. References

Cohen, R.M., March, J.G., and Olsen, J.P. (1972). A garbage can model of organizational choice. *Administrative Science Quarterly*, 17, 1-25

Carley, K.M. (1986a). Measuring Efficiency in a Garbage Can Hierarchy. *Ambiguity and Command: Organizational Perspectives on Military Decision Making*, 165-194. Baylor, Boston, MA: Pitman.

Carley, K. M. (1986b). Measuring efficiency in a garbage can: Implications for crisis management. *Ambiguity and Command: Organizational Perspectives on Military Decision Making*, 195-231. Baylor, Boston, MA: Pitman.

March, J.G. (1991). Exploration and exploitation in organizational learning. *Organization Science* 2(1), 71-87

McPherson, J. M., Smith-Lovin, L. (1987). Homophily in Voluntary Organizations: Status Distance and the Composition of Face-to-Face Groups. *American Sociological Review.* 52, 370-379.

Morgan, G.P. and Carley, K.M. (2011). Exploring the impact of a stochastic hiring function in dynamic organizations: In *Proceedings of the 20th Conference on Behavior Representation in Modeling and Simulation*, Sundance: UT

Morgan, J.H., Morgan, G.P., and Ritter, F.E. (2010). A preliminary model of participation for small groups. *Computational, Mathematical, and Organizational Theory*. 16. 246-270.

Oh, H., Chung, M., and LaBianca, G. (2004). Group social capital and group social effectiveness: The role of informal socializing ties. *Academy of Management Journal*, 47(6), 860-875.

Padgett, J.F. (1980). Managing garbage can hierarchies. *Administrative Science Quarterly*, 25, 583-604.

Selznick, P. (1948). Foundations of the theory of organizations. American Sociological Review. 13:1. 25-35

Acknowledgements

This work was supported by the Office of Naval Research (ONR MURI N00014-08-1-1186). We thank Jonathan H. Morgan for his incisive comments.

Author Biographies

GEOFFREY P MORGAN is a PhD student in Carnegie Mellon's Computation, Organization, and Society program. He is interested in organizations, individuals, and how the two influence each other.

KATHLEEN M CARLEY is 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.

Socio-cognitive Networks: Modeling the Effects of Space and Memory on Generative Social Structures

Changkun Zhao, Ryan Kaulakis, Jonathan H. Morgan, Jeremiah W. Hiam, Joseph P. Sanford, Frank E. Ritter
The College of Information Sciences and Technology
The Pennsylvania State University

Geoffrey P. Morgan
School of Computer Science
Carnegie Mellon University

Keywords: Cognitive modeling, socio-cognitive network, memory

ABSTRACT: This paper examines the relationships between environmental and cognitive factors as they influenced the formation of social networks through modeling how these factors affect tie formation between pairs of agents in a simulated world. We modeled worlds consisting of 20, 40, and 60 ACT-R agents and examined the influence of population size, run time, map configuration, and navigation strategies, comparing the density and clustering of the resulting networks. We found that all these factors affect tie formation for agents with perfect memories, with population size having the greatest effect. We also examined the effect of these exogenous factors on the ties' strength in the agents' memories by combining and analyzing egonets. We found that changes to each of these exogenous factors affected the network's average memory activation value of each tie, with population size having a negative effect and run time having a positive effect. Map configuration and navigation strategy both influenced network structure. Further, we found that using the agents' activation values as a threshold for network inclusion was a useful way for identifying core groups and subgroups within the network. These findings provide further insights into the cognitive dimensions underlying networks and their structures, as reflected by Dunbar's (1998) number and Simmelian numbers. These results also show that these factors need to be reported when describing network simulations.

1. Introduction

We examine here two socio-cognitive processes that influence the formation of social networks: spatial reasoning and memory retrieval threshold. This work is motivated by a desire to better understand how socio-cognitive processes influence the development of persistent patterns of relations, represented in this paper as network topologies. By socio-cognitive processes, we refer to both those cognitive resources and mechanisms necessary to create and sustain social ties, as well as those group-level factors known to moderate human decision-making (Morgan, Morgan, & Ritter, 2010; Morgan & Carley, 2011). We focus on spatial reasoning and retention in this paper because these two processes seem foundational to understanding the emergence of social networks in a variety of contexts. We hope to deepen our understanding of network formation by modeling the relationship between cognitive and environmental factors, as it pertains to tie formation between agent dyads.

To explore this relationship, we introduce a set of agent-based models and experiments that test the influence of: (a) population size, (b) run time, (c) map configuration, and (d) agent navigation strategies. The outputs of this model are interaction networks (whole networks representing the total number of agent interactions that occurred within a single run) and ego-nets (declarative representations of the agent's friends network). For any one run, there is 1 interaction network and as many ego-nets (networks from the agent's egocentric point of view) as there are agents in the experiment. We compare the number of ties and total degree of 54 interaction networks in 11 different conditions. We also merged the individual ego-nets to examine how the structure of the socio-cognitive network changed as the semantics of the tie were tuned.

Our model is unusual in that we model social processes using a cognitive architecture (ACT-R) that is primarily associated with cognitive science. To our knowledge, Carley (1991, 1992) and Newell (1994) were the first to implement a model based on a cognitive architecture (Plural-Soar) to study organizations. More recently, Gonzalez, Lerch, and Lebiere (2003), Lebiere, Gonzalez, Dutt, and Warwick (2009), Reitter and Lebiere (2010b), and Juvina, Lebiere, Martin, and Gonzalez (2011) have used cognitive architectures to model human decision making in collaborative tasks. Barrett, Eubank, and Marathe (2006) have developed a large simulation with millions of non-cognitive light agents. While our work builds upon these efforts, our interest in network formation poses some unique challenges. We review these challenge in light of the current literature and our solutions to them in the next two sections.

2. Computational Social Models

Researchers have developed agent-based models to explore a variety of questions. We briefly examine two major modeling approaches: cognitive and social modeling. These approaches are not necessarily mutually exclusive; however, combined socio-cognitive models are relatively rare because they are generally expensive to create and run.

Social simulation models frequently, but not always, use bounded rational agents. Bounded agents are bounded both cognitively and socially (Simon, 1991). These agents generally engage in some kind of goal-driven time-constrained decision cycle dependent on local information. In addition, agents are usually adaptive, though options are generally conditioned upon previous action. Often, these simulations demonstrate aggregate behaviors that are emergent. These complex system-level behaviors arise out of the agents' discrete interactions, but cannot be explained entirely in reference to them. Further, at different levels of observation, different kinds of emergent behavior can be seen. It is often these kinds of traits that simulations are uniquely able to capture.

We treat the topology of interaction networks and their associated characteristics as an emergent property of the social system, defined in terms of interaction opportunities and memory constraints. Kraut et al. (2002) found that actor proximity fundamentally influences the evolution of network topologies by determining the interaction frequencies of actors across the network, while Allen (1977) demonstrated that the probability of two people communicating in an environment could be represented with a decreasing hyperbolic function of their distance. After a certain distance, the probability that two people will communicate decreases rapidly, making tie formation unlikely. We, thus, chose to focus on factors that are known to directly affect agent proximity or inter-agent distance: population size, run duration, and map configuration.

Cognitive models have historically focused on modeling human cognition at the symbolic and sub-symbolic level (Newell, 1990; Anderson, 2007). Providing models of perception and memory, we can use cognitive architectures like ACT-R to simulate the formation of social ties in declarative memory. We believe that over time memory constraints fundamentally influence a social network's topology and capabilities by constraining the network's ability to process information, identify important changes in state, and respond to those changes.

Here, we look at the processing of social information by exploring the concept of nodal carrying capacity, the number of agents an agent can retain in memory. To that end, we examine how environmental factors contribute to the consolidation and retention of social ties in memory.

This concept is similar Dunbar's (1998), where limitations associated with the neocortex limit the number bi-directional ties any one person can retain in memory. Dunbar argues that maintaining stable relationships requires repeated memory activations to identify not only one-on-one relationships but also third party relationships (i.e., the knowledge that my friend is also friends with other actors who I monitor). Further, he claims that the cognitive load associated with maintaining this set of relationships in memory rises exponentially as group size increases (Dunbar, 1998, p. 63). Based on retrospective empirical studies, Dunbar (1998, pp. 65-78) argues that this ratio between cognitive load and group size underlies the small-world effect observed by Milgram, Simmel, and others.

Therefore, we expect that larger populations acting over longer time periods in fully connected environments will result in the most connected declarative network structures. We also expect that less connected layouts will result in interaction networks that consist of more fragmented networks, leading to smaller ego-nets. We also expect that map configurations characterized by nexus points will exhibit behaviors similar to the water-cooler effect (DiFonzo, 2008). We, however, are less certain where we might see thresholds in network formation, where for instance population growth no longer has an effect or run time is no longer relevant.

3. Nodal Carrying Capacity: The Effect of Agents' Memory and Space

Having summarized our model's exogenous and cognitive factors, we provide both a definition and a prediction as to how that factor will influence network formation.

3.1 Interaction Frequency

We model three factors that influence interaction frequency: population size, run time, and map configuration.

Population density: We predict population density will have the greatest impact on social interaction frequency. Here, we model such shifts by changing the population size, holding the environment size constant.

Length of simulation (run time): We predict longer run times will lead to more ties and denser networks. Consequently, determining the run time lengths necessary for a network to reach a stable state under a given set of conditions (e.g., memory decay of ties) is important for accurately representing the formation of a group of interest.

Environment configuration: We predict the configuration of the environment will influence the structure of the simulated social network. We measure the relative connectivity of our three map configurations by defining its *grid ratio*. The *grid ratio* is the ratio of the number of edges over the total number of edges possible for a rectangular grid containing the same number of rooms.

We tested three map configurations (Figure 1). The first configuration (1a) is a full 5x5 grid with *grid ratio* 1.0. We expect this environment will result in relatively high connectivity. The second configuration (1b) has a central area with *grid ratio* 0.75. We believe this central meeting point will lead to network densities and clustering that are less pronounced than those associated with the 5x5 map but more than those associated with the hallway map. The third configuration (1c) is a two-hallway configuration with *grid ratio* 0.6. This configuration should lead to low connectivity due to the large distances between agents.

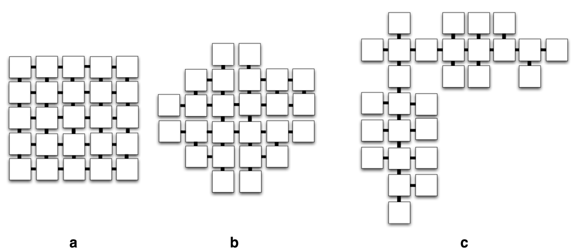


Figure 1. (a) 5x5 grid, (b) Central Area map, (c) and Hallway map

3.2 Cognitive factors

To better simulate the construction of social networks, it is necessary to consider the behavior patterns of agents at the cognitive level. In this paper, we particularly focus on memory decay and navigation strategies.

Memory: We examine memory effects on tie formation by using Anderson's activation theory (2004) to model the construction of social knowledge in declarative memory. In our model, the number of friends depends on the number and size of active long-term memory chunks representing the agent's social relationships. The number of active memory chunks is influenced by several factors, including initial memory activation, retrieval threshold, memory decay rate, time of retrievals, and practice time.

Navigation strategies: In a social network, the agents' movement patterns will influence the social network's topology by influencing any one agent's interaction opportunities. For example, a policeman walking a beat will have more acquaintances than a person who spends most of their time at home, if only because the policeman has more opportunities to meet people.

To replicate human navigation behavior, we implemented two navigation strategies: random-walk and fixed-path.

1) *The Random-walk strategy* replicates navigation patterns without a specific goal. It randomly selects an available direction to move.
2) *The Fixed-path strategy* follows a set path in a small area. This strategy simulates routine navigation behavior, such as going to work or shopping.

4. Experiment Environment

To model multi-agent social behavior using cognitive architectures, we constructed a simulation environment, VIPER. All of our experiments were conducted on a 2GHz eight-core server with 8GB of RAM. The server runs Linux 2.6.31 under Ubuntu 11.04, with SBCL 1.0.52 Lisp, and ACT-R 6 (Anderson et al., 2004).

4.1 The VIPER Server

VIPER models the constraints associated with embodiment on social networks. It supports multi-agent simulations to study network science. It is lightweight in that it is text-based, but is extendable and records agent behaviors. VIPER is designed to be a part of a distributed model that resolves events in either real or accelerated time. The network's speed and frequency of communication are determined by its component agents, with no queue of events being enforced. VIPER is designed so variations in performance originate from the agents participating in the environment, versus being a function of the environment. The VIPER server is based on NakedMUD, an open-source environment. It communicates with client programs using the Telnet Protocol.

Within the environment provided by the server, agents or human subjects are situated on maps of interconnected rooms. The agents can see and communicate within each room. Agents can walk between the rooms, and can interact with objects in the rooms.

To connect ACT-R to VIPER, we implemented the Telnet Agent Wrapper for ACT-R (TAWA) in Common Lisp. It supports logging in, waiting for synchronization, logging, halting, and writing results to CSV files. It also exports a number of functions that provide ways to examine the environment, speak, listen, move, and otherwise control a virtual body in VIPER.

When an ACT-R model is wrapped by TAWA, executions of model code are delayed until a privileged administrator agent signals for synchronization. Error conditions are also caught by TAWA and standard UNIX error codes are returned instead of dropping into the more standard debugger. For example, a successful run returns 0 to the parent process, while any error (e.g., network errors like the server being unreachable) returns a non-0 value. Returning error codes like this allows automated error checking in large-scale experiments.

4.2 Synchronization

Because memory decay and networks are strongly temporal, we paid special attention to time. To synchronize the agents, the administrator agent (which does not take part in the experiment) waits for all of the TAWA wrapped agents to finish loading and logging in. It then signals to TAWA to begin the simulation. Because TAWA delays the evaluation of the model code until synchronization, no agent experiences time before the synchronization signal. Further, all ACT-R models are set to run in real-time and for the same amount of "real time", so they all halt after the same perceived period. Thus, the total time experienced is the same for all agents.

4.3 Scalability

Early benchmarks showed that ACT-R processes took up about 80MB per process. We would only have been able to run about 100 processes on a single 8GB machine before swapping would occur. To reduce the per-process footprint, a number of optimizations were implemented. Basic space reductions were achieved by using the DECLARE Lisp construct, as well as by pre-compiling the components, removing the debugger, and saving the whole system (sans the ACT-R agent model) as a system image. This reduced our per-process memory footprint somewhat, but they were not the biggest contributions towards memory usage reduction.

In SBCL Lisp 1.0.52, the "--merge-core-pages" flag was recently added. This flag enables Kernel SamePage Merging (Arcangeli, 2009) under recent versions of Linux. This optimization flags shared areas of memory as being able to be merged unless modified. Because a significant percentage of our agents were replicated, we found that we could reduce the per-process memory footprint as low as 8MB per process (with one shared copy of the merged pages excepted). Thus, the only activities that increase the size of this footprint are changes within individual agent models. This lets drastically larger number of agents to be run, whether on single processors or HPC.

5. Experiment and Results

We now discuss our experiment's method and results.

5.1 Experiment parameters

We used 54 runs of our simulation to test three environmental factors: population density, running time, and map configuration, which are shown in Table 1.

Table 1. Experiment parameters.

Variable	Values	#
Population	20, 40, 60	3
Run time (s)	125, 250, 500	3
Map	Full-Grid, Central, Hall (100%, 75%, 60%)	3
Navigation strategy	Random, Fixed-Path	2
Total possible combinations		18

We tested the effect of memory retention by analyzing the activation values associated with each agent's friend chunks, represented as an ego-net log file. Each file contains the names and the last meeting locations of that agent's friends. In ACT-R, the activation value represents the memory strength of an object or an event. With the activation value of each relation chunk, we can easily convert the friend weight into a meaningful idea of tie strength.

5.2 Results

Our simulation generates two types of network data: (a) log data extracted from Viper directly, and (b) egocentric data stored in each agent's declarative memory. We examine them and some related network measures.

Log network. Figure 2 shows a sample network. The nodes in the figure are the agents in the simulation (N=20); the ties in the figure are un-weighted and represent the co-occurrence of two agents in the same room at any point in the run. Thus, the log network represents the ground truth of each agent's opportunities to meet with other agents, and the memory of agents with perfect memory.

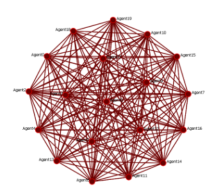


Figure 2. An example ground-truth network from log data (20 agents, 125 s running time, Hallway configuration)

Table 2 compares eight runs on the ground-truth networks. We found that each factor influences the co-occurrence network, reflected in the network's density and tendency towards clustering. Density is the percentage of all possible ties in the network that are found in that network. Population size tends to decrease the network's density and has some effect on clustering. Run time tends to increase the network's density and decreases clustering. The grid ratio increases both density and clustering. Navigation strategy increases density and decreases clustering.

Merged Ego networks. The other type of data collected in the simulation is not ground-truth, but instead the individual ego-networks for each actor stored in that agent's declarative memory. Figure 3 shows an example.

Figure 3. An example ego network.

Each agent represents actors it has met as a working memory element (WME) chunk. The activation of each WME can be used to derive the amount of time a human would require to recall the actor. The semantics of each dyadic tie is important in interpreting a network. Thus, we consider the density of the merged ego-network across various activation levels. An activation value of '-3' indicates that the actor will need as much as 0.2 seconds to recall the chunk, whereas an activation of '3' indicates that the actor will need less than 5 ms to recall the chunk (but perhaps longer to report it).

Figure 4 offers an example of how a single merged ego-network can show how structure changes as the criteria for tie formation increases. Increasing the activation threshold provides a way for identifying core groups and subgroups within the network.

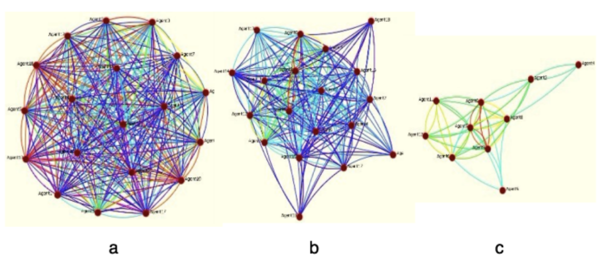


Figure 4. The same ego network at various memory thresholds, (a) -3.5, (b) 0.0, and (c) 1.0.

We considered two rules for determining whether the tie should exist: bi-directional (where threshold, t, must be met by <u>both</u> A_{ij} and A_{ji}) and directional (where threshold, t, must be met by <u>either</u> A_{ij} or A_{ji}). Figure 5 shows the density of the merged ego-networks as various activation levels are sampled—from this analysis it is clear that map topology influences the creation of the merged ego-network.

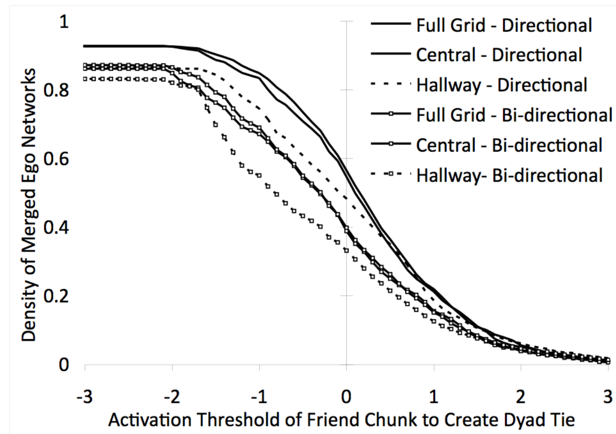


Figure 5. Map configuration and selection criteria affect the generated ego network.

5.3 The Effect of Nodal Capacity

We also examine the relation between influential factors and activation values of each configuration. Table 2 lists the population sizes, run times, navigation strategies, and average activation values for agents across eight runs.

Table 2 shows that these factors influence the activation values between agents. The population size has a negative influence on the average activation because larger populations decrease the average activation value significantly (from 0.304 to -1.117). Running time has a positive influence on the average activation value—it increases from 0.304 to 1.607 as we increase the running time from 125 s to 500 s.

Table 2. Results comparing eight networks to investigate trends in the ground-truth co-occurence networks, the cognitively limited merged ego-centric networks, and the average chunk activations for friends.

				Ground Truth		*Merged Ego-Centric*		Average
Pop	**Run Time**	**Grid Ratio**	**Navigation**	**Density**	**Clustering**	**Density**	**Clustering**	**Activation**
20	125	0.60	Random	0.905	1.000	0.810	0.855	0.304
40	125	0.60	Random	0.880	0.924	0.220	0.467	-0.678
60	125	0.60	Random	0.859	0.961	0.051	0.086	-1.117
20	**125**	0.60	Random	0.905	1.000	0.810	0.855	0.304
20	**250**	0.60	Random	0.947	0.944	0.805	0.852	0.931
20	**500**	0.60	Random	0.947	0.944	0.855	0.897	1.607
20	125	**0.60**	Random	0.905	1.000	0.810	0.855	0.304
20	125	**0.75**	Random	0.947	0.944	0.600	0.696	0.126
20	125	**1.00**	Random	0.950	0.947	0.660	0.724	0.440
20	125	0.60	**Random**	0.905	1.000	0.810	0.855	0.304
20	125	0.60	**Fixed Path**	0.947	0.944	0.855	0.897	1.992

Table 2 shows that these factors influence the activation values between agents. The population size has a negative influence on the average activation because larger populations decrease the average activation value significantly (from 0.304 to -1.117). Running time has a positive influence on the average activation value—it increases from 0.304 to 1.607 as we increase the running time from 125 s to 500 s.

We also found that the map configuration influences the average activation values, but that this influence does not correspond to the grid ratio. It shows that the "Central" map with a grid ratio of 75%, has the lowest average activation values (0.126); the "Hallway" map with a 60% grid ratio has a higher value (0.304); and the 5-by-5 map with 100% grid ratio has the highest average activation value (0.440). The map configuration's effect on average activation values may be due to some chunks never being formed (and thus never being averaged) in the hall-way map configuration, as indicated in Figure 5.

The results show that the fixed path strategy has a positive influence on the average activation values. Because agents using this strategy walk around a small area, the agents tend to have fewer ties but higher activation values per tie, simulating a neighborhood effect.

5.4 Operationalizing Dunbar's Number

We also preliminarily examine the influence of these factors of Dunbar's number by applying a fixed activation threshold to simulate limits of cognition. For this analysis, we consider networks with an activation threshold of 0.0, ACT-R's RT parameter was set to -3.5, thus the time to recall a chunk is 0.011 seconds based on ACT-R's memory equations.

In Table 2, we use two measures to evaluate the thresh-olded ego-networks, density and clustering. As the net-work's size increases, the network's density decreases and clustering coefficient decreases. As the simulation's length increases, the density and clustering coefficient increases. As the environment becomes more intercon-nected, there is some evidence to suggest that the network density increases and the clustering coefficient increases, although this evidence is mixed. The Fixed Path agent, which traverses the space differently, retains more of its edges and shows more clustering.

In our results we highlight density because it illustrates the effect of nodal carrying capacity (defined in this case by an activation threshold) on a simulated social network. Table 2 suggests that population size has the highest influence on the merged ego-net, which is not surprising, because a network maintaining a constant density as new actors are added would require more and more ties from the marginal actor. Also, the agent's attention is limited, and attending to more agents may require more resources than the agent can bring to bear, which suggests one mechanism for how Dunbar's number may moderate social activities. The other three factors also have some influence on the found density and structuration of the network. In real social contexts, we suspect these three factors with smaller effects would interact, and perhaps magnify the effect of population size.

5.5 Interaction Density on Locations

Additionally, environment configurations can create loci of interaction or activity spaces (Brantingham & Brantingham, 1993). These locations are where the majority of all interactions occur. Brantingham and Brantingham use this concept to study crime densities, but this idea can be expanded to other activities, such as co-occurrence or socialization. When traveling to or between these spaces, people tend to take routine paths. Costanzo et al. (1986) demonstrated that people near one another tend to travel along the same paths to activity hotspots. Therefore, we expect that agents will also tend to take high frequency paths to common locations because they are constrained by the world's geometry.

These high activity spaces for one of our environments are shown in Figures 6a and 6b. Figure 6a shows the connectivity between all agents and the rooms in which they have interacted, while Figure 6b shows a heat map of room activity. Given the concentration and degree of these spaces, we show that agents who traveled between activity spaces tended to travel along the same path. This result is similar to the water-cooler effect, which suggests that interaction happens naturally in shared public locations.

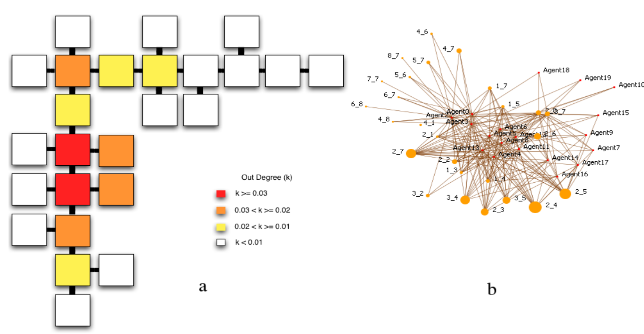


Figure 6. (a) Hallway map's heatmap; (b) Agents-by-location network.

6. Discussions and Future Work

In this study, we created a multi agent social network simulation that provides a flexible platform to examine several influential factors in social networks. Based on the existing literature, we hypothesized two basic types of influential factors, exogenous and cognitive factors. Our exogenous factors included population size, run time, and map configuration, while our cognitive factors included navigation strategies and memory activation parameters.

Our results show how cognitive and environmental factors can influence network growth and shape. From the simulation results, we find that all three factors influenced the ground-truth co-occurrence networks, as well as the merged ego-networks. As expected, the co-occurrence networks are less affected than the cognitively limited merged ego-networks. The effect of running time is not as significant as we expected, and shows plateauing after 250s run for these configurations. The large running time also weakens the effect of map configuration in our ground-truth networks because it provides agents enough time to travel around the whole map. By examining interaction density on locations, we also find that the shared public locations have higher interaction densities, in a manner similar to the water-cooler effect.

Taking advantage of the ACT-R memory mechanism, we are able to create an egocentric view of the social network by looking into the chunks in the declarative memory for individuals and across a whole network. We found the structure and density of the merged egocentric network to depend heavily on the criteria for tie formation, with the most generous criteria producing a network very similar to that suggested by the ground-truth networks.

By examining the activation values between agents, we also found that the four factors examined influence the activation values of ties between agents. The result of agent activation logs show that the population size has a negative influence on the average activation (smaller groups have stronger ties); that running time has a positive influence on the average activation value; and that map configuration has some influence on the average activation but that the change of value does not correspond to changes in grid ratio. This suggests that grid ratio is not a sufficient measure of map configuration at least with these maps, and we need to find a more accurate measure in the future. We also found that navigation strategies do influence activation values, with the Fixed Path strategy resulting in a neighborhood effect (strong localized ties).

From this preliminary study we found that the exogenous influential factors have impact on both measures of the network with a threshold. The population size has the highest influence on the merged ego-network's density, suggestive of the implications of the effect of Dunbar's number on the real social activities.

Finally, we conducted a preliminary study of the effect of these factors on Dunbar's number and of applying a cognitive limit on each agent. We used an activation threshold to implement a cognitive limit on tie strength and to suggest some meaning for the ties. We measured the cognitively limited network's density and inherent structuration.

Thus, reports about simulated networks need to report these factors and similar factors when describing their simulations. Knowing the values of these factors will be necessary for duplicating results because these factors have strong and interacting effects.

Future avenues of work will build upon some of the more interesting issues. First, we would look at analysis of normalized thresholds to see if there are regularities in their effects on network topology. Second, we would run more agents and more runs (Ritter, Schoelles, Quigley, & Klein, 2011), because the system to demonstrate these effects was kept deliberately small. Finally, we would extend our analysis on the effects of cognition on network measures analogous to Dunbar's Number.

7. Acknowledgements

This work was supported by a grant from DTRA (HDTRA1-09-1-0054). We thank Jaeyhon Paik for help on the agent design and Jeremy Lothian for assistance with VIPER.

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psych. Rev*., 111(4), 1036-1060.

Barrett, C., Eubank, S., & Marathe, M. (2006). Modeling and simulation of large biological, information and socio-technical systems: An interaction based approach. In D. Goldin, S. A. Smoka, & P. Wegner (Eds.), *Interactive computation: The new paradigm* (pp. 353-394). Berlin Heidelberg: Springer.

Brantingham, P. L., & Brantingham, P. J. (1993). Nodes, paths and edges: Considerations on the complexity of crime and the physical environment. *Journal of Environmental Psychology, 13*(1), 3-18.

Carley, K. M. (1991). A theory of group stability. *American Sociological Review, 56*(3), 331-354.

Carley, K. M. (1992). Organizational learning and personnel turnover. *Organizational Science, 3*(1), 20-46.

Carley, K. M., & Newell, A. (1994). The nature of the social agent. *Journal of Mathematical Sociology, 19*(4), 221-262.

Carley, K. M., & Prietula, M. J. (Eds.). (1994). *Computational organization theory*. Hillsdale, NJ: Erlbaum.

Costanzo, C. M., Halperin, W. C. & Gale, N. (1986). Criminal mobility and the directional component in journeys to crime. In R. Figlio, S. Hakim & G.

Rengert, (Eds.) *Metropolitan Crime Patterns*. 73-96. Monsey, NY: Criminal Justice Press.

DiFonzo, N. (2008). *The watercooler effect: A psychologist explores the extraordinary power of rumors*. New York: Avery.

Dunbar, R. I. M. (1998). *Grooming, gossip, and the evolution of language*. Cambridge, MA: Harvard.

Gonzalez, C., Lerch, J. F., Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science, 27*(4), 591-635.

Juvina, I., Lebiere, C., Martin, J. M., & Gonzalez, C. (2011). Intergroup prisoner's dilemma with intragroup power dynamics. *Games and economic behavior, 2*(1), 21-51.

Kaulakis, R., Zhao, C., Morgan, J. H., Hiam, J. W., Sanford, J. P., & Ritter, F. E. (in press). Defining factors of interest for large-scale socio-cognitive simulations. In *Proceedings of ICCM 2012*, Berlin.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence, 33*(1), 1-64.

Lebiere, C., Gonzalez, C., Dutt, V., & Warwick, W. (2009). Predicting cognitive performance in open-ended dynamic tasks: A modeling comparison challenge. In *Proceedings of the 9th ICCM* 12-13. Manchester, UK.

McCarty, C., Killworth, P. D., Bernard, H. R. Johnsen, E. C., & Shelley, G. A. (2001). Comparing two methods for estimating network size. *Human Organization 60*(1), 28-29.

Morgan, J. H., Morgan, G. P., & Ritter, F. E. (2010). A preliminary model of participation for small groups. *Computational & Mathematical Organization* Theory, *16*(3), 246-270.

Morgan, G. P., & Carley, K. M. (2011). Exploring the impact of a stochastic hiring function in dynamic organizations. In *Proceedings of BRIMS 2011*. 106-113. Sundance, UT: The BRIMS Society.

Reitter, D., & Lebiere, C. (2010). A cognitive model of spatial path planning. *Computational and Mathematical Organization Theory, 16*(3), 220-245.

Reitter, D., & Lebiere, C. (2011). How groups develop a specialized domain vocabulary: A cognitive multi-agent model. *Cognitive Systems Research, 12*(2), 175-185.

Ritter, F. E., Schoelles, M. J., Quigley, K. S., & Klein, L. C. (2011). Determining the number of model runs: Treating cognitive models as theories by not sampling their behavior. In L. Rothrock & S. Narayanan (Eds.), *Human-in-the-loop simulations: Methods and practice* (pp. 97-116). London: Springer-Verlag.

Simon, H. A. (1991). Bounded rationality and organizational learning. *Organizational Science, 2*(1). 125-134.

Zhao, C., Paik, J., Morgan, J. H., & Ritter, F. E. (2011). A multi-strategy spatial navigation model in a text-based environment. In *Proceedings of BRIMS 2011,* 251-258. 11-BRIMS-036: Sundance, UT.

Author Biographies

CHANGKUN ZHAO is a PhD candidate at the College of Information Sciences and Technology's Applied Cognitive Science Lab. He is interested in spatial memory modeling, information processing theory.

RYAN KAULAKIS is a PhD candidate at the Applied Cognitive Science Laboratory in the College of IST. His research interests are in model fitting in cognitive architectures, and soft-computing.

JONATHAN H. MORGAN is a researcher at the College of IST's Applied Cognitive Science Lab with expertise in modeling social interactions and social moderators, including the moderating effect that relative spatial positions have on group performance.

JEREMIAH W. HIAM is a research assistant at the College of IST's Applied Cognitive Science Lab. Game design, AI in games, and psychology of gaming are his fields of interest

JOSEPH P. SANFORD is an undergraduate research assistant at the College of IST's Applied Cognitive Science Lab. Memory, spatial reasoning, and emotion are his fields of interest.

FRANK E. RITTER is on the faculty of the College of IST, an interdisciplinary academic unit at Penn State to study how people process information using technology. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Cog. Systems Res., Human Factors,* and *IEEE SMC Part A: Systems and Humans*.

GEOFFREY P. MORGAN is a PhD candidate in Carnegie Mellon's Computation, Organizations, and Society program. His adviser is Kathleen M. Carley. He is interested in organizations, individuals, and how the two influence each other.

The Organizational Dynamics of Submarine Piloting and Navigation Teams

Ronald Stevens, Trysha Galloway, Peter Wang
UCLA IMMEX Project
5601 W. Slauson Ave, #272
Culver City, CA 90230
310-498-5700
Immex_ron@hotmail.com, trysha@teamneurodynamics.com, peter@teamneurodynamics.com

Keywords:
Teamwork, Entropy, Neurodynamics, EEG, Wavelets

ABSTRACT: *Our objective was to apply ideas from complexity theory to derive expanded neurodynamic models of Submarine Piloting and Navigation (SPAN) showing how teams cognitively organize around task changes. The cognitive metric highlighted was an EEG-derived measure of engagement (termed NS_E) that was modeled into collective team variables showing the engagement of each of six team members as well as that of the team as a whole. We modeled the cognitive organization of teams using the information content of the NS_E data streams derived from calculations of their Shannon entropy. We show that the periods of team cognitive reorganization 1) occurred as a natural product of SPAN teamwork particularly around periods of stress, 2) appeared structured around episodes of communication, and 3) occurred following deliberate external perturbation to team function. These periods of reorganization have a fractal structure and can be lengthy lasting up to 10 minutes. As overall NS_E entropy levels are significantly higher for expert teams, the entropy of neurophysiologic data streams may be a useful candidate for modeling teamwork over scales from seconds to weeks.*

1. Introduction

Teams have been described as complex dynamic systems that exist in a context, develop as team members interact over time, and evolve and adapt as situation demands unfold (Kozlowski & Ilgen (2006). From the perspective of complexity science, they can be thought of as self-organized flows of information spanning biological processes and broader societal activities. As team members interact there are flows of information, often turbulent, that organize periodically around a common goal only to change form again as the task and environment evolve.

In teams, members continually modify their actions in response to the changing actions of other resulting in dynamic synchronizations of information that can be observed across different systems and subsystems, including verbal (Drew, 2005), gestural (Ashenfelter, 2007), postural (Shockley et al., 2003) and more recently neurophysiologic (Stephens et al, 2010; Dumas et al., 2011). Most of these studies have consisted of two-person teams performing coordination tasks or tasks in controlled settings.

Our goal is to expand these ideas to larger real-world teams where the information flows are longer and expertise develops at multiple scales. In a typical training sequence neural events spanning seconds give rise to communication events of tens of seconds and longer team coordination events that contribute to the performance of a task. When aggregated across training sessions, these tasks provide the framework for the formal training. The training sequence depicted in Figure 1.1 spans nearly 10^7 orders of magnitude of seconds for a 10 week course; yet here are few descriptions of the linkages across these subsystems and time scales. Such integrated models of the organization of teams at short (second / minutes) as well as longer (days or weeks) time scales of training could better inform why some teams function better than others; Are they cognitively more flexible and able to more rapidly enter and exit organized neurophysiologic states? Can these abilities be taught, and if so, how? Longitudinal extensions of these models could be capable of both predicting teamwork breakdowns as well as suggesting routes for them to regain their rhythm once it was lost.

Teams, like many complex systems, are thought to operate at a level of self-organized criticality between random and highly organized states (Bak *et al*., 1987). That tenuous but significant state has also been called the edge of chaos, a feature that allows teams to adapt to both momentary disruptions, such as environmental perturbations, and more permanent alterations, such as changes in task requirements. At the 'sweet spot' of organization the team demonstrates both stability and flexibility in the form of supportive co-regulation.

In order to detect these areas of organization we have been developing models of team neurodynamics using symbolic representations of the EEG-derived levels of Engagement that are termed Neurophysiologic Synchronies (NS). In prior studies we have shown that the NS data streams contain information regarding the current and past cognitive states of the team and this information can be quantified by the entropy in the data stream. The idea of entropy is derived from information science and is a measure of the level of uncertainty or "amount of mix" in a symbol stream (Shannon & Weaver, 1949). As teams entered and exited periods of organization, the entropy would fluctuate with low entropy indicating more organization and high entropy more organization.

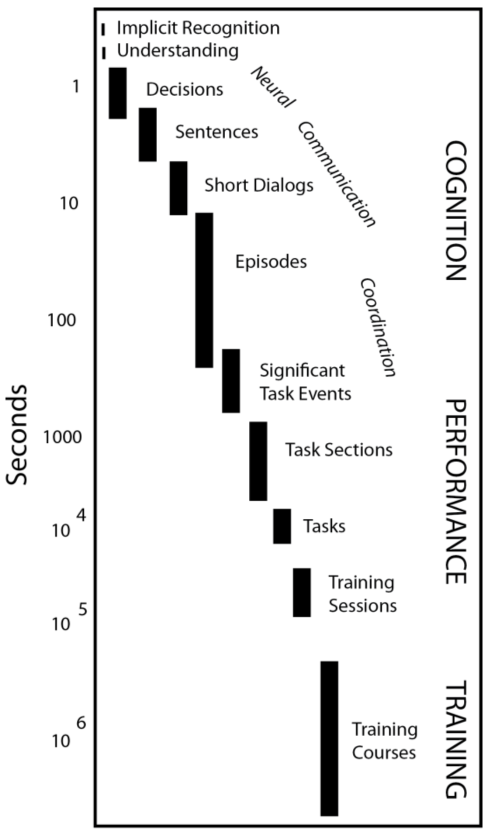


Figure 1.1. Time Scales of Team Training

In this study we describe the cognitive organization of SPAN teams in terms of these entropy fluctuations and begin to link them with team communication and natural and external perturbations to the task.

2. Tasks

These studies were conducted with navigation training tasks that are integral components of the Submarine Officer Advanced Course at the US Navy Submarine School, Groton, CT. The task is a high fidelity Submarine Piloting and Navigation (SPAN) simulation that contains dynamically programmed situation events which are crafted to serve as the foundation of the adaptive team training. Events in SPAN include encounters with approaching ship traffic, the need to avoid shoals, changing weather conditions, and instrument failure. There are task-oriented cues to guide the mission, team-member cues that provide information on how other members of the team are performing / communicating, and adaptive behaviors that help the team adjust to the unfolding situation.

Each SPAN session contains three segments beginning with a Briefing where the overall goals of the mission are presented. The Scenario is a dynamically evolving task containing both easily identified and less well-defined processes of teamwork. One regularly occurring process is the periodic updating of the ship's position termed 'Rounds'. This process occurs every three minutes with a countdown from the 1 minute mark; the regularity / completeness of this process serves as an internal performance metric (Stevens et al., 2012). The Debrief is the most structured part of the training with team members reporting on their performance. Within this reporting structure there are often sequential episodes with overlapping or underlying nested structures as specific events within the Scenario are discussed.

For the studies reported, six members of the teams were fitted with the EEG headsets. These were the Quartermaster on Watch (QMOW), Navigator (NAV), Officer on Deck (OOD), Assistant Navigator (ANAV), Contact Coordinator (CC), and Radar (RAD).

3. Methods

3.1 EEG

The ABM, B-Alert system contains an easily-applied wireless EEG system that includes intelligent software designed to identify and eliminate multiple sources of biological and environmental contamination and allow real-time classification of cognitive state changes even in challenging environments. The 9-channel wireless headset includes sensor site locations: F3, F4, C3, C4, P3, P4, Fz, Cz, POz in a monopolar configuration referenced to linked mastoids. ABM B-Alert® software acquires the data and quantifies alertness, engagement and mental workload in real-time using proprietary software (Berka et al, 2004).

The data processing begins with the eye-blink decontaminated EEG files containing second-by-second calculations of the probabilities of High EEG-Engagement (EEG-E), Distraction and High EEG-Workload (EEG-WL). Simple baseline tasks are used to fit the EEG classification algorithms to the individual so that the cognitive state models can then be applied to increasingly complex task environments, providing a highly sensitive and specific technique for identifying an individual's neural signatures of cognition in both real-time and offline analysis.

The EEG-E metric is an approximation of the multiple ways the term Cognitive Engagement has been reported in the literature. For instance, it has been used to describe the amount of cognitive processing a learner applies to a subject (Howard, 1996) or as something that has to be broken during a task so that a learner can reflect on his / her actions, (Roberts &Young 2008). It shares similarities with alertness or attention and can be visual and / or auditory. It is analogous to the EEG-rhythm-based attention measures that are often associated with alpha power dynamics (Jung *et al*., 1997; Kelly *et al*., 2003; Huang *et al.*, 2007). Operationally, our premise is that precise cognitive terms will be difficult to associate with EEG-derived measures of cognition in the context of teamwork and that functional associations will need to be derived empirically.

3.2 Neurophysiologic Synchronies

Neurophysiologic methods can extend the use of speech for modeling team dynamics by providing "in the head" measures of team dynamics (Warner et al, 2005). As members of a team perform their duties, each would be expected to exhibit varying degrees of cognitive states such as attention, workload, or engagement and the levels of these components at any one time would reflect aspects of team cognition.

Rather than focusing on neurophysiologic markers such as P300 or N400 that rapidly appear and disappear in response to many stimuli we have used EEG-derived measures of engagement (EEG-E) which may persist longer across team members.

Neurophysiologic synchrony models are developed by aggregating EEG-derived measures of Engagement into a vector representing the levels being expressed by each team member (Figure 3.1a). Using artificial neural network (ANN) technologies these vectors are modeled into collective team variables termed neurophysiologic synchronies of engagement (NS_E) that show the engagement of each of 6 team members as well as of the team as a whole (Stevens et al, 2012).

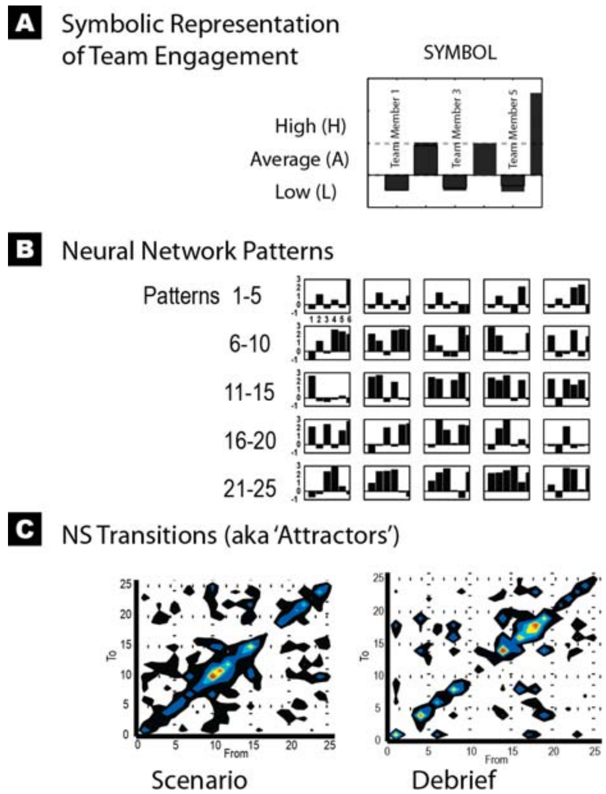


Figure 3.1. Data Flow for Creating Team Neurodynamics Models.

ANN classification of these second-by-second vectors creates a symbolic state space showing the possible combinations of Engagement across members of the team (Figure 3.1b). For ANN training we chose to use a linear architecture of nodes on the assumption that most second-by-second state transitions would be local changes among individual team members and that large team shifts would be indicative of a changing team organization. The linear architecture ensured that the most similar states were proximal and that differences were more distal. This configuration should result in a diagonal line in a second-by-second transition matrix if most transitions were local and in a distributed map if they were more random. Transition matrices plot the NS at time *t* against time *t+1* (Figure 3.1c). Primarily local changes are indicated by the diagonal.

4. Results

4.1 Transformation of Neurophysiologic Measures into Data Flows

The team neurodynamic models outlined above are expanded in Figure 4.1 for one SPAN team session. This figure illustrates the process for transforming the expression of neurophysiologic measures into information flows. The top (Figure 4.1a) plots the second-by-second expression of the 25 NS_E symbols resulting from the ANN training.

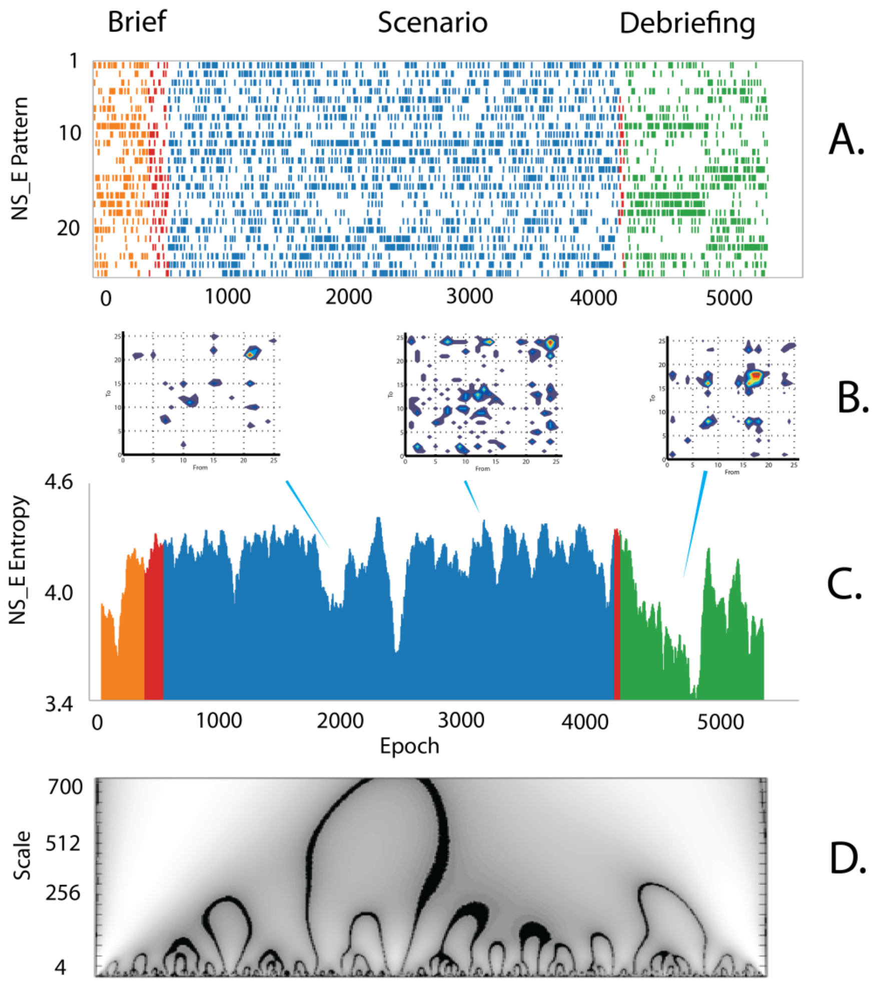


Figure 4.1. Multiple Representations of NS_E Neurodynamics. A) The second-by-second expression of individual NS_E symbols. B, C) The transition matrices for NS_E show the NS_E symbols being expressed at the regions indicated in the entropy profile. D) Continuous Wavelet Transformation of the NS_E entropy stream.

As with most SPAN performances, the expressions were not uniform, but showed qualitative changes over time, particularly at the Scenario / Debriefing junction. For instance NS_E symbols 13-18 were poorly expressed during the Scenario but dominated during the Debriefing. Qualitative changes also occurred dynamically during the Scenario, such as the increased expression of NS_E symbols 21 & 22 between epochs 1800 and 2500, but these are generally less obvious than those at task junctions.

While a symbolic representation of the state of the team is very useful for <u>characterizing</u> team neurodynamics, it is not the best representation for <u>quantifying</u> team neurodynamics. Although there are methods for the quantitative representation of symbols (Daw *et al.*, 2003), we chose to perform a moving average window approach to derive numeric estimates of the entropy (which we illustrate with Shannon's entropy) of the NS symbol stream.

Calculated entropy is expressed in terms of bits; the maximum entropy for 25 randomly-distributed NS symbols would be log2 (25) or 4.64. For comparison, an entropy value of 3.60 would result if roughly half (12) of the NS symbols were randomly expressed. To develop an entropy profile over a SPAN session, the NS Shannon entropy was calculated at each epoch using a sliding window of the values from the prior 100 seconds.

As teams entered and exited periods of organization, the entropy should fluctuate as a function of the number of NS_E symbols being expressed by the team during a block of time (Stevens & Gorman, 2011). As shown in Figure 4.1c, the fluctuations in the NS_E entropy levels were complex, with rapid changes over seconds and longer fluctuations covering minutes.

Entropy is a quantity, the value of which is determined by the state of the system, in our case with regard to the engagement of the team members. By itself, it says nothing about the state of the system or how the system got to that state; this information comes from the NS symbols and transition matrices.

During periods of low entropy (~epochs 1900 & 2400) very few of the 625 potential (i.e. from 25 symbols to 25 symbols) NS symbol transitions were used by the team during a 120 second window. During periods of higher entropy the diversity of transitions increased 2-3-fold reflecting a relaxation of this organization. During the Debriefing Segment (~epochs 4200-4500) the team underwent different cognitive organizations as topics were discussed.

An alternative way of viewing the structure of the data in the entropy of the NS_E data stream is by continuous wavelet transformation (CWT) which transforms a signal into a function with two variables, *scale* and *time*. In this process a one-dimensional wavelet is compared to a section of the time signal and a wavelet coefficient is calculated which indicates how similar the wavelet is to the selected signal section; higher coefficients indicate a closer match. The signal section is then shifted and the comparison re-made, and so on. The scale of the wavelet is then lengthened (which is akin to stretching it out to fit a larger window) and the comparisons along each segment of the signal are again performed.

The process is repeated for all scales (which are from 4 to 712 in Figure 4.1d). This transformation reveals the structure of both the rapidly changing fluctuations (small scales) as well as the longer fluctuations (larger scales). For instance, the clearly demarcated segment between epochs 1750 and 2450 represented a period where the entropy rapidly dropped. This coincided with the team's decision to turn north in front of a merchant ship and in the process approached the bounds of the safe operating envelope; a stressful situation. At the end of that loop the entropy rapidly increased and the audio logs indicated that the team was exiting this period of stress. These areas of decreased entropy and segmented regions of organization indicated that the team was required to change a previous strategy to deal with these particular events in a markedly new way. Following resolution, that region of organization is capped and 'business as usual' followed.

4.3 Entropy Fluctuations are Associated with Conversation Episodes

The areas of NS_E entropy fluctuation can be considerably longer than the time to utter a question or sentence suggesting if there is an association between NS expression and speech, it will be at a higher level of organization, perhaps more like the episodes as described by Salem (2011). Episodes consist of mutually constructed sequences of behavior. When conversation is described as an episode it is around the premise that individuals initially construct messages to be consistent with their individual perceptions, and they evolve these messages in ways that are linked to those of others. The episode may evolve until it is mutually satisfactory to all, or it might continue into another episode. It can be thought of as a discussion around a central theme or topic.

Detailed NS entropy mapping of episode shifting during episodes in the Debriefing segment is shown in Figure 4.4 for one SPAN performance. There were two major discussion topics indicated by the green and orange; the first one was a discussion why the submarine deviated around a merchant and in the second the OOD asked the team if they understood his overall plan. During the first topic the NS_E entropy steadily dropped until closure was reached. The entropy rapidly increased and again slowly declined as closure on the second major topic was reached. Nested within each of these fluctuations are other shorter fluctuations, i.e. the entropy streams appear fractal.

As shown in the lower half of Figure 4.4, a CWT was performed on the data stream, and the audio was analyzed near the different regions of topical organization. There were regions at scales of 50-60 seconds that from speech appeared to be points where the discussion changed. Our hypothesis is that the fractal nature of NS entropy may be linked to the nested structure of the discussion.

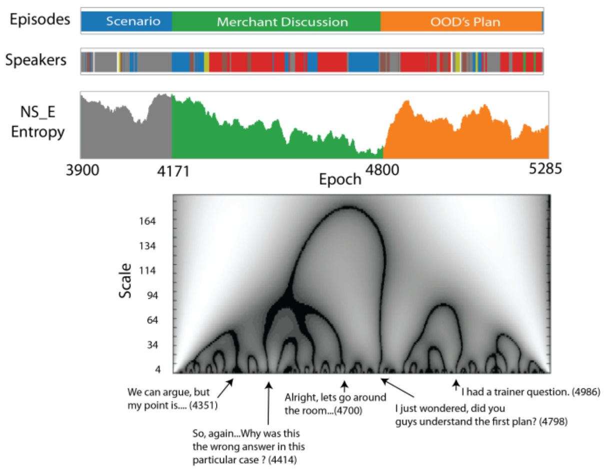


Figure 4.2 Nesting of Episodes During the Debriefing. NS entropy organizes around conversational topics (top); CWT reveals distinctive scaling regions of NS entropy indicating nesting of NS within topics. The beginnings of conversations at the significant scaling junctions are indicated by arrows

There was no clear association between these fluctuations and the person speaking as the primary discussants the CAPT (red), NAV (blue), OOD (purple), and the QMOW (gray), participated throughout the Debriefing.

4.4 NS_E Entropy Fluctuations Occur Around Perturbations to the Task.

We next explored whether NS entropy fluctuations occurred with deliberate perturbations to the team. During two SPAN Scenarios the simulation was paused while the Captain addressed the crew with recommendations. The NS_E profile for one of these events is shown in Figure 4.3. Coincident with the pause there was a drop in NS_E entropy which persisted until the simulation was restarted.

4. Discussion

The studies presented have shown that the periods of team cognitive reorganization that are identified by entropy fluctuations A) occurred as a natural product of SPAN teamwork (Figures 4.1), B) appeared structured around episodes of communication (Figures 4.2), and C) were associated with external perturbations (Figure 4.3). These organizations were lengthy lasting up to 10 minutes and appeared more associated with communication episodes rather than shorter 'units of thought', sentences, or who was speaking.

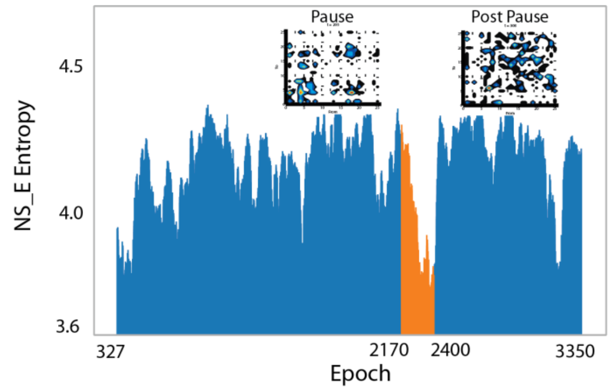


Figure 4.3. Perturbation of the SPAN Task Induces Team Reorganization. During the period highlighted in orange, the simulation was in pause and the attractors were more organized than after the pause.

A second interesting property of NS entropy streams is that they may distinguish experienced SPAN teams from SOAC teams (Stevens & Gorman, 2011). In the mean, experienced teams have significantly higher overall NS entropy than novice teams, suggesting that either there are fewer periods of decreased NS entropy or the decreases have a shorter duration or amplitude, the meanings of which are currently unknown.

These findings have been developed into a model that links the NS_E entropy and transitions with the development of expertise (Figure 5.1). The cognitive organization axis reflects the overall entropy levels and the diversity of transitions in the transition maps. The regularities of the Rounds countdown, along with possible deviations, were obtained from the speech of the Recorder and provided an internal performance metric (Y-axis).

A highly organized team (lower right), as typified by a SPAN team under stress, is shown by tightly organized transitions and low entropy levels, equivalent to the random usage of only 9 of the 25 NS_E symbols. NS transitions pooled from the Scenario segments of six SOAC teams still show restricted transitions, but the mean entropy has increased. As teams progress after their training and develop more experience (SUB Teams), the entropy levels and the diversity of the transitions further increase; from the performance metric this stage would approximate the 'sweet spot' of team function. The data from zero history student teams who had not worked together (lower left), and were unfamiliar with both the task and domain showed the highest entropy, levels that were nearly equivalent to randomized NS_E data streams. As discussed in the Background, this hypothesized structure is consistent with the idea that teams, like many complex systems, are thought to operate at an organization level between random and highly-organized, the so-called edge of chaos or self-organized criticality.

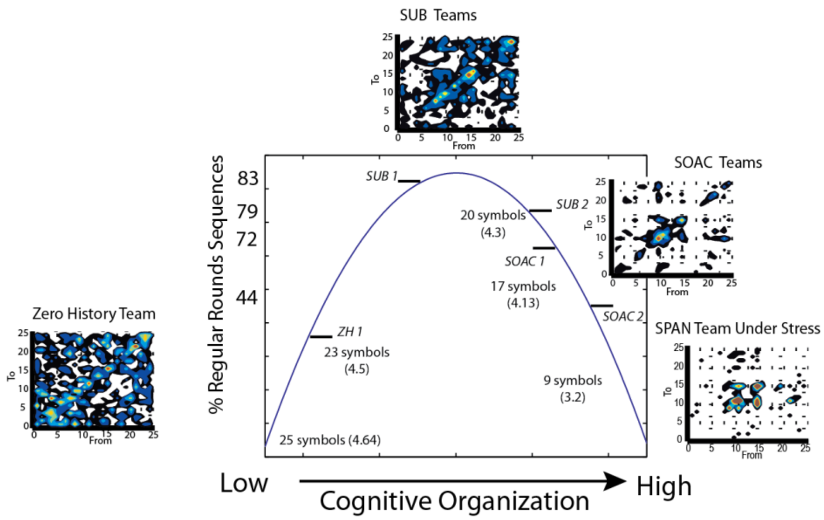


Figure 5.1 Expertise and the Cognitive Organization of SPAN Teams. The text in italics positions the overall NS_E entropy for two SOAC teams (1 & 2) and two SUB teams (1 & 2) along with one zero history team (ZH-1). The numbers inside the parabola are the NS_E entropy values for those teams and the number of NS symbols they represent.

The entropy fluctuations appeared fractal as illustrated by the areas of high similarity across scales in Figures 4.2, 4.4 and 4.5, indicating there is structure in the team's activity across time, both for short-lived and longer-lasting events. We expect to see fractal patterns when nested processes (e.g. neurophysiological, cognitive, behavioral) are interconnected and mutually influential. In fact, recent studies suggest that the entropy fluctuations are likely multifractal (Likens, Amazeen, Gorman, Stevens, in preparation).

The multifractal nature of stock market volatility may provide a relevant metaphor (Sornette, 1998) for teamwork. A normally functioning market is chaotic and the local variability of the process is heterogeneous (i.e.multifractal) in its sources and flows of information. But a market in crisis shows increased coordinated behavior of a large number of agents in the market and a decrease in the diversity of financial multifractality; this additional structure and order in the system process leads to a 'crash'.

Similarly, experienced navigation teams may function closer to the 'sweet spot' of organization where the team demonstrates both stability and flexibility in the form of supportive co-regulation. Multifractal analysis may allow for both the detection of this sweet spot and drift away from it.

The significance of the dynamics being described by neurophysiologic measures is beginning to accumulate, but it is important to couch their dynamics into the broader context of teamwork through the window of team communication. In the Debriefing, there are intriguing associations between NS_E entropy and episodes of conversation that need to be further explored. A working hypothesis is that the fractal nature of NS entropy may be linked to the nested structure of the discussion, and that the nested structure of NS pattern is related to the level of team expertise.

The concept of semantic similarity in Latent Semantic Analysis may provide a relatively objective baseline with which to compare NS patterns of topic shifting, where the cosines across segments of discourse *within* episodes could be compared to cosines across segments of discourse *between* episodes.

Lastly, the earlier finding of higher entropy levels being expressed by experienced teams is especially interesting as it may suggest a higher hierarchical level of NS involvement. If so, then NS may be a useful measure for modeling teamwork over scales from seconds to weeks.

5. References

Ashenfelter, (2007) Simultaneous analysis of verbal and nonverbal data during conversation :symmetry and turn-taking. Unpublished thesis, University of Notre Dame.

Bak, P.,Tang, C.and Wiesenfeld, K. (1987). Self-organized criticality: an explanation of 1 / *f* noise.*Physical Review Letters* **59** (4): 381–384.

Berka, C., Levendowski, D. J., Cvetinovic, M. M., Petrovic, M. M., Davis G. et al. (2004). Real-Time analysis of EEG indexes of alertness, cognition, and memory acquired with a wireless EEG headset. *International Journal of Human-Computer Interaction.* Lawrence Erlbaum Associates, Inc. **17**(2), 151-170.

Daw, C. S., Finney, C. E. A., & Tracy, E. R. (2003). A review of symbolic analysis of experimental data. *Review of Scientific Instruments.* 74, 915.

Drew, P. (2005). Conversation analysis. In Fitch & Sanders (Eds.). Handbook of language and social interaction. Mahwah, N.J. Lawrence Erlbaum.

Dumas, G., Nadel, J., Soussignan, R., Martinerie, J., and Garnero, L. Inter-brain synchronization during social interaction. PlosOne 5, 8 e12166.

Kozolowski, J S., & Ilgen, D. R. (2006). Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest*, 7, 77-124.

Salem, P. (2011). The sweet spots in human communication. *Nonlinear Dynamics, Psychology and Life Sciences*, 15, 3, 389-406.

Shockley, K., Santana, M.-V., & Fowler, C. A. (2003). Mutual interpersonal postural constraints are involved in cooperative conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 29 (2), 326–332.

Shannon, C., & Weaver, W. (1949). *The mathematical theory of communication.* Urbana: University of Illinois Press.

Sornette, D., (2003). Critical market crashes. *Physics Reports*, 78, 1-98.

Stephens, G., Silbert, L., and Hasson, U. Speaker-listener neural coupling underlies successful communication. *Proc. Nat. Acad. Sci*. www.pnas.org/cgi/doi/10.1073/pnas.1008662107.

Stevens, R., Galloway, T., Berka, C., & Behneman, A. Wholgemuth, T., Lamb, J. & Buckles, R (2011a). Linking models of team neurophysiologic synchronies for engagement and workload with measures of team communication. In *Proceedings of the 20th Conference on Behavior Representation in Modeling and Simulation.*

Stevens, R., Galloway, T., Wang, P., Berka, C., & Behneman, A. (2011b). Developing systems for the rapid modeling of team neurodynamics. *Interservice / Industry Training Simulation and Education Conference (I/ITSEC) 2011*. Paper No. 11097.

Stevens, R., Gorman, G. (2011) Mapping Cognitive Attractors onto the Dynamic Landscapes of Teamwork *Proceedings of the 14th International Conference on Human-Computer Interaction*, July 9-14, Orlando, FL.

Stevens, R.H., Galloway, T., Wang, P., and Berka, C. (2012). Cognitive neurophysiologic synchronies: What can they contribute to the study of teamwork? **Human Factors**, in press.

Warner, N., Letsky, M., and Cowen, M. (2005). Cognitive model of team collaboration: macro-cognitive focus. In *Proceedings of the 49th Human Factors and Ergonomics Society Annual Meeting*, September 26-30, 2005. Orlando, FL.

6. Acknowledgments

This work was supported in part by The Defense Advanced Research Projects Agency under contract number(s) NBCHC070101, NBCHC090054, and NSF SBIR grant IIP 0822020. The views, opinions, and/or findings contained are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. Special thanks to Chris Berka Adrienne Behneman, and Veasna Tan for technical support and data collection, and to Drs. Nia Amazeen and Jamie Gorman for helpful discussions.

.

Simulating the Emergence of Conventions in Small-World Networks

Roland Mühlenbernd
Seminar für Sprachwissenschaft
University of Tübingen, Germany
roland.muehlenbernd@uni-tuebingen.de

Michael Franke
Institute for Logic, Language and Information
Universiteit van Amsterdam, The Netherlands
m.franke@uva.nl

ABSTRACT: *Lewis (1969) invented signaling games to show that meaning convention can arise simply from regularities in communicative behavior. This paper contributes to the question how the formation of signaling conventions depends on the social structure of a population. Our results not only show that different language conventions can coexist, but also where to expect uniformity and language contact. We found that place and time of convention formation can be traced well to particular clusters of high/low values of suitable notions from formal network theory. Against prior expectations, we found that agent rationality is less important than network role in deciding how and when an agent adopts a convention.*

1. Introduction

Lewisean signaling games have become a standard model for the pragmatic evolution of semantic meaning (cf. Steels, 1997; Nowak & Krakauer, 1999; Skyrms, 2010). In order to understand the applicability and conceptual adequacy of signaling game models, the most important theoretical question that needs to be addressed is under which circumstances stable signaling conventions can arise. Following a general trend in evolutionary game theory, recent studies have started to probe into the simplifying assumption underlying classical evolutionary dynamics that populations of agents are homogeneous, i.e., barring of social structure. Dispensing with this artificial assumption, Zollman (2005), for instance, has demonstrated for a so-called imitate-the-best dynamic how coexistent language conventions can evolve if the population of language users is arranged on a lattice. Wagner (2009) studied the same dynamic on so-called β-graphs (defined below) which exhibit more realistic small-world properties, namely a high clustering coefficient, paired with a low characteristic path length (Watts & Strogatz, 1998). Wagner's simulations showed that (i) the higher the clustering coefficients the larger the fractions of players that acquire a unique signaling convention, and (ii) the lower the characteristic path length the smaller the number of connected regions of agents that use the same signaling convention.

This paper probes deeper into the relation between social structure and language evolution in order to further our knowledge of synthetic evolutionary processes in structured populations and thereby to pave the way for a more thorough understanding of the sociological factors of linguistic variability. While previous related work has focused on studying which global network structures are especially conducive to innovation and its spread (Ke, Gong, & Wang, 2008; Fagyal, Swarup, Escobar, Gasser, & Lakkaraju, 2010), the present paper investigates more closely the local network properties associated with (regions of) agents that have successfully learned a language or not. In distinction to previous studies, we also focus not on imitation, but on usage-based learning dynamics from evolutionary game theory. To study the effect of agent rationality on language evolutions, we considered best-response dynamics and reinforcement learning.

Our most striking results, in a nutshell, were these.

Firstly, we found that languages form preferentially on locally highly connected subgraphs; borders between languages fall preferentially on regions "in between" highly connected subregions. Secondly, conventionalization depended crucially on local network properties, while the learning dynamics and the amount of agent rationality had hardly any noticeable effect. Thirdly, we compared the local properties of agents who had learned a language early with those who had learned late, and of those who lived at the borders of language regions with those who lived in the interior. To characterize the differences we found, we made a distinction between family men and globetrotters, which we characterized by relative values of suitable clusters of properties from formal network theory. We found that early learners and interior agents tend to be family men with tight local connections, while late learners and border agents tend to be globetrotters with wide-ranging global connections. Finally, we found evidence that the first ones to adopt and stabilize a convention were highly connected family men.

2. Signaling games

A signaling game is a game played between a sender S and a receiver R. Initially, nature selects a state $t \in T$ with prior probability $\Pr(t) \in \Delta(T)$, which the sender observes, but the receiver doesn't. S then selects a message $m \in M$, and R responds with a choice of action $a \in A$. For each round of play, players receive utilities depending on (in the cheap-talk case we consider here) the actual state t and the response action a. We will here be concerned only with a simple variant of this game, which we call *Lewis game*: there are only two states that are equiprobable, two messages and two actions that correspond one-to-one with the states, indicated by the same index. Players share an interest in successful communication, expressed by utility function $U(t_i, a_j) = 1$ if $i = j$ and 0 otherwise.

Although messages are initially meaningless in this game, meaningfulness arises from regularities in behavior. Behavior is defined in terms of strategies. A *behavioral sender strategy* is a function $\sigma : T \to \Delta(M)$, and a *behavioral receiver strategy* is a function $\rho : M \to \Delta(A)$. A behavioral strategy can be interpreted as a single agent's probabilistic choice or as a population average. For a Lewis game, exactly two isomorphic strategy profiles constitute evolutionary stable states (Huttegger, 2007). In these, strategies are pure (i.e., action choices have probabilities 1 or 0) and messages associate states and actions uniquely, like so:

$$L_1\colon \begin{array}{ccccc} t_1 & \longrightarrow & m_1 & \longrightarrow & a_1 \\ t_2 & \longrightarrow & m_2 & \longrightarrow & a_2 \end{array} \qquad L_2\colon \begin{array}{ccccc} t_1 & & m_1 & & a_1 \\ & \times & & \times & \\ t_2 & & m_2 & & a_2 \end{array}$$

3. Learning dynamics

Classical evolutionary game theory assumes a homogenous population of agents and studies evolutionary processes on the aggregate population level. In this paper we focus instead on more fine-grained agent-based evolutionary dynamics. Agents repeatedly play a Lewis game with those agents they are connected with in their social network, and adapt their behavioral strategies based on learning from previous interactions. We consider two kinds of *learning dynamics* that differ with respect to how rational the learning agents are assumed to be: more rational best-response dynamics (BR) and less rational reinforcement learning (RL).

The idea of BR-dynamics is simple: agents remember the past plays that they have been engaged in and derive from their memory a belief about their opponents' behavior; it is to that belief that they play a *rational* best response. We assume here that agents form a belief about the collective behavior of all of their neighbors, not keeping track of each agent separately. More concretely, a given agent's belief about his neighborhood's receiver (sender) behavior $B_r(a|m)$ ($B_s(t|m)$) is just a behavioral receiver (sender) strategy derived by keeping track of *all* of the agent's past interactions. The sender's expected utility for sending m in state t is $EU_s(m|t) = \sum_{a \in A} B_r(a|m) \times U(t, a)$. Accordingly, the receiver's expected utility is $EU_r(a|m) = \sum_{t \in T} B_s(t|m) \times U(t, a)$. A *best response* is an action choice that maximizes expected utility. A sender's set of best response messages for a given state t is then defined as $BR(t) = \arg\max_m EU_s(m|t)$. Accordingly a receiver's set of best response actions for a given message m is defined as $BR(m) = \arg\max_a EU_r(a|m)$. This gives rise to the following *response rules for BR-dynamics*:

$$\sigma(m|t) = \begin{cases} \frac{1}{|BR(t)|} & \text{if } m \in BR(t) \\ 0 & \text{else} \end{cases} \tag{1}$$

$$\rho(a|m) = \begin{cases} \frac{1}{|BR(m)|} & \text{if } a \in BR(m) \\ 0 & \text{else} \end{cases} \tag{2}$$

The second dynamic RL can be captured by a simple model based on urns, also known as *Pólya urns* (cf. Roth & Erev, 1995; Skyrms, 2010). An urn models a behavioral strategy, in the sense that the probability of making a particular decision is proportional to the number of balls in the urn that correspond to that action choice. By adding or removing balls from an urn after each encounter, an agent's behavior is gradually adjusted. For signaling games, the sender has an urn Ω_t for each state $t \in T$, which contains balls for different messages $m \in M$. The number of balls of type m in urn Ω_t designated with $m(\Omega_t)$, the overall number of balls in urn Ω_t with $|\Omega_t|$. If the sender is faced with a state t she draws a ball from urn Ω_t and sends message m, if the ball is of type m. The same holds *mutatis mutandis* for the receiver. The resulting *response rules for RL-dynamics* are:

$$\sigma(m|t) = \frac{m(\Omega_t)}{|\Omega_t|} \quad (3) \qquad \rho(a|m) = \frac{a(\Omega_m)}{|\Omega_m|} \quad (4)$$

The learning dynamics are realized by changing the urn content dependent on the communicative success. In detail: if communication via t, m and a is successful, the number of balls in urn Ω_t is increased by $\alpha \in \mathbb{N}$ balls of type m and reduced by $\gamma \in \mathbb{N}$ balls of type $m' \neq m$. Similarly, for the receiver. In this way successful communicative behavior is more probable to reappear in subsequent rounds. In our experiments, all urns were initially filled with 100 balls and we set $\alpha = 10$ and $\gamma = 4$. From previous work (Mühlenbernd, 2010) we knew that in order to match the plasticity of different learning dynamics, we should consider BR-learners with *unbounded* memory but RL-learners with *bounded* memory. For that reason, an RL-learners' urns only reflected the impact of the last 300 interactions (irrespective of role) that the agent was engaged in. With an initially empty memory, BR-agents initially played entirely at random, just like their RL-cousins.

4. Network games: design, basic notions & simulation results

We modeled a structured population as a β-graph. A β-graph is obtained by first considering a ring of nodes where each node is connected to its k nearest neighbors and subsequently, for each node, rewiring its $k/2$ left neighbors to a random vertex n with probability β (Watts & Strogatz, 1998). For our analysis, we created 10 such β-graphs with 300 nodes, $k = 6$ and $\beta \in \{.08, .09, .1\}$. These parameter choices ensured the small-worldliness of our networks that we had to keep small for obtaining enough data points at manageable computation costs. For each network, we started 20 silulation runs each with either only BR- or only RL-agents. Agents played the standard Lewis game. Communication happened randomly between neighbors on the network, and each agent's behavior was updated separately after each round of communication the agent was involved in. We recorded strategies of agents in suitably chosen regular intervals. Each silulation run ran until at least 90% of agents had acquired a language, or each network connection had been used 3000 times in either direction. The latter was to ensure a compromise between a short running time and sufficient time for learning, but also because we were interested in the results of learning after a realistic time-span, not in limit behavior. An example for a possible resulting network is shown in Figure 1.

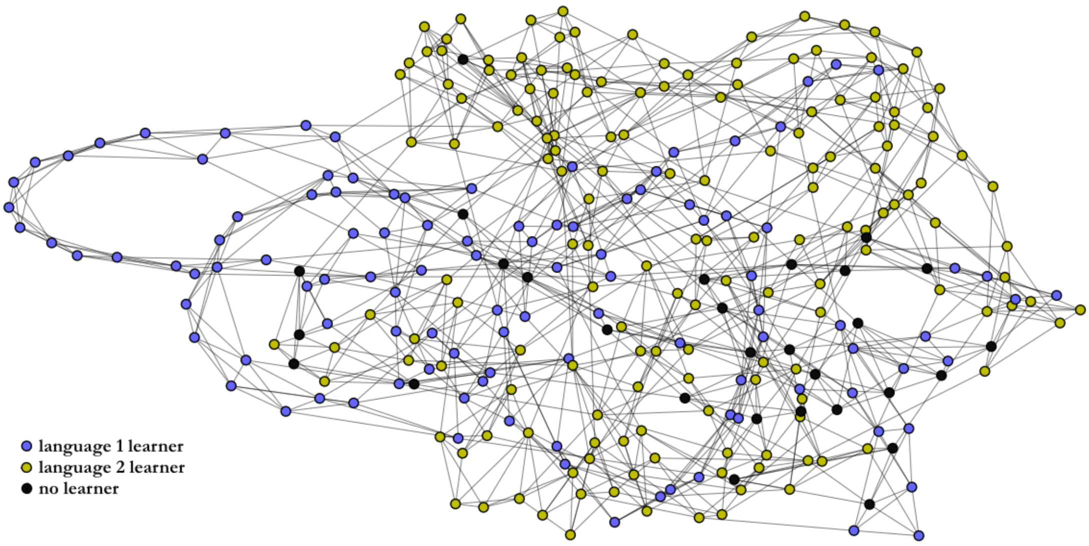


Figure 1: Small-world network after a simulation with 90% learners and 10% non-learners.

4.1 Result 1: Global properties

In order to determine which local network properties best characterize where, on average, learning would be most likely successful, we looked at what we will call *language regions*. A language region is a maximal subset of agents that have acquired the same language that forms a connected subgraphs. Despite the different learning dynamics, our data confirmed Wagner's (Wagner, 2009) results that in small world networks like ours the number of language regions is small while the size of language regions is relatively big. Most of the time, two big language regions formed, one for each signaling convention. BR-dynamics, due to its slightly higher flexibility, was prone to produce a little more regional variability. Figure 2 depicts the *Hinton diagrams* of distributions of the number of language regions for both learning dynamics.

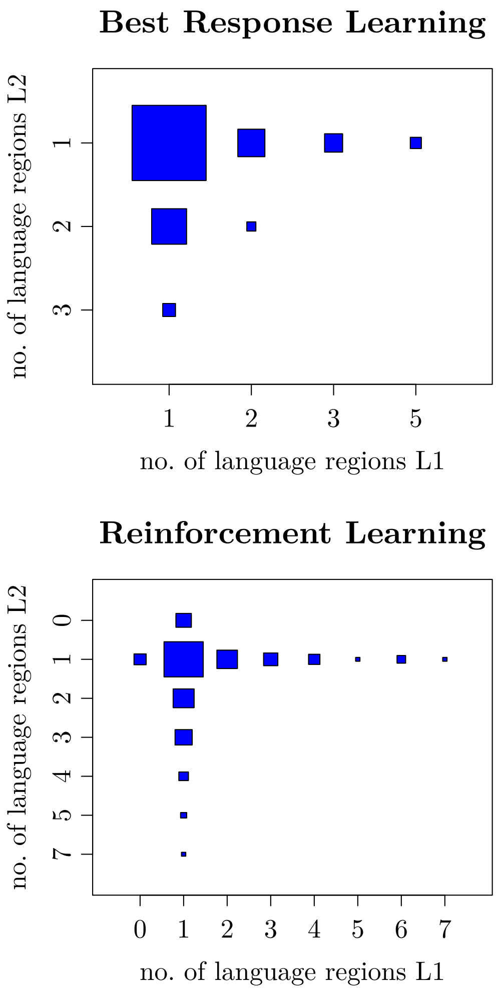


Figure 2: Hinton diagrams for the distributions of the number of regions of both languages; each for one of both dynamics.

On top of that, we also found that each connected language region of a given type had always a higher *average clustering* and *transitivity* value than the expected average value for a connected subgraph with the same size n (= number of nodes), whereas the *density* value didn't exhibit such a divergence (see Figure 3).[1] We may conclude from this that local cliquishness supports the evolution of a local language, whereas density doesn't.

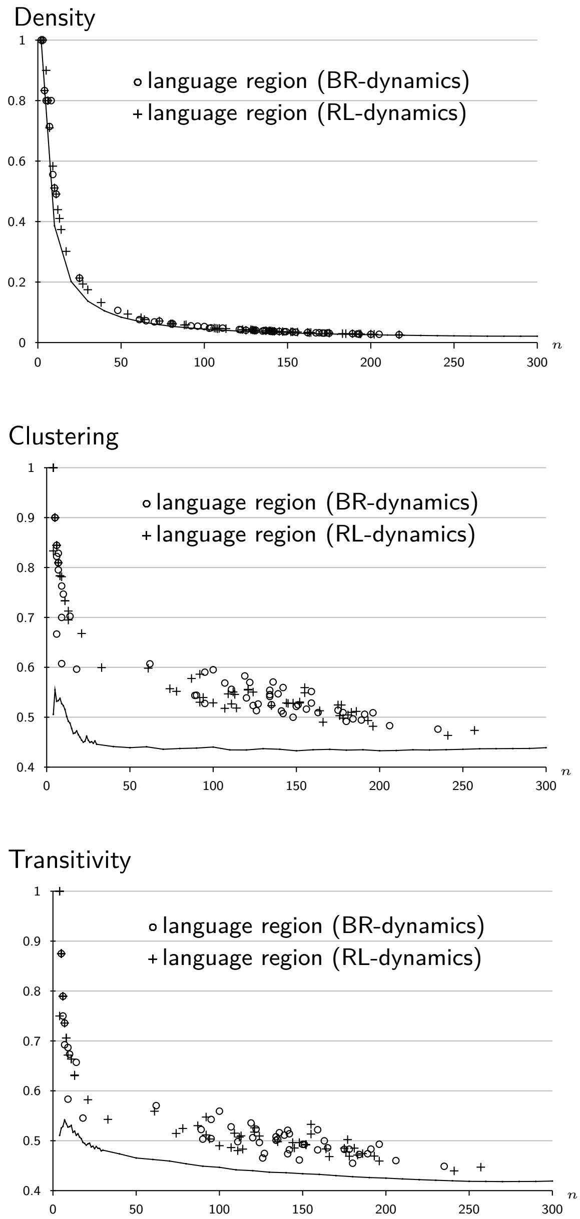


Figure 3: Comparing observed density, clustering and transitivity of language regions with expected values from randomly chosen subgraphs (solid lines, subgraph size along the x-axis).

[1] For the definitions of *average clustering*, *transitivity* and *density* see Appendix A.

4.2 Result 2: Global properties

A further goal was to investigate the relationship between meaning evolution and social network structure. The theoretical challenge here lies in adequately characterizing local network roles in terms of formal notions of network connectivity, which can never be crisp, but must necessarily be of a probabilistic nature. For our present purposes, however, a rather straightforward cross-classification based on whether an agent is globally and/or locally well-connected turned out to have high explanatory value. Using suggestive terminology, we will be mainly concerned with two types of agents, *family men* and *globetrotters*. The former have tight local connections, with less global connections; the latter show the opposite pattern plus a high degree of connectivity.

In order to capture these notions more adequately, we look at suitable notions from social network theory (Jackson, 2008): *betweenness centrality* (BC), *closeness centrality* (CC), *degree centrality* (DC) and *individual clustering* (CL).[2] High values for BC and CC characterize agents that are globally at a central position in the network. So, for a measure of *global connectedness* we looked at relative values of these properties. On the other hand, a high value for CL should be considered a measure for the agent's *local connectedness*. A high value for DC depicts a high degree of connectivity. Family men and globetrotters are thus characterized as follows:

	BC	CC	CL	DC
family man	low	low	high	-
globetrotter	high	high	low	high

In a next examination we were interested in the relationship between agents' local network properties and their language region-dependent position. Based on their learning success and network position, we classified agents into (i) *early learners* vs. *late learners* and (ii) *border agents* vs. *interior agents*. A learner is an agent who, by the end of a simulation run has acquired the same signaling convention in both her sender and receiver role (for RL-agents this meant getting close enough to the pure strategy in question). Early learners are the first 20%, who learned a convention, late learners are the last 20%. Interior agents have only neighbors who learned the same language as they themselves, while border agents are agents whose neighborhood is not uniformly behaving in the same way that they do.

Our results were by and large the same for both learning dynamics: early learners and interior agents tend to be family men, border agents tend to be globetrotters (see Figure 4).

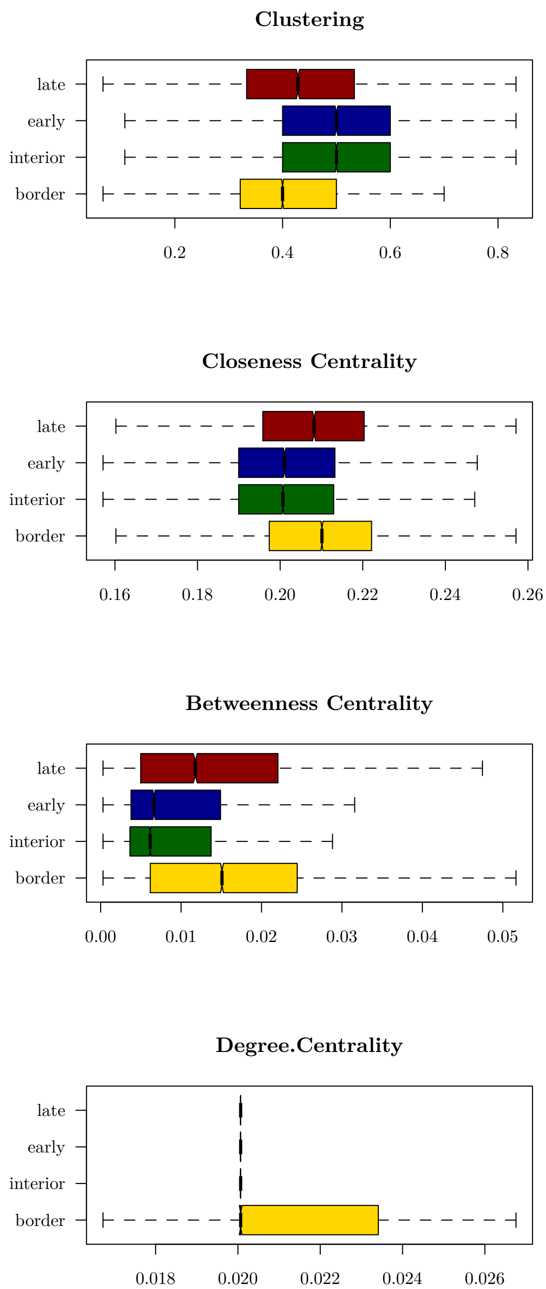


Figure 4: Box plots of local network properties for early vs. late learners, and for border vs. interior agents. Results were roughly the same for both learning dynamics.

[2]For the definitions of these network properties see Appendix A.

Intuitively speaking, this means that in order to quickly learn a language in a social network an agent would have to be well embedded in a dense *local* structure. Globally well-connected agents, on the other hand, have difficulties learning a language in a heterogeneous network very early, because they might be torn between different locally firmly established conventions. (Naturally, the difference between interior and border agents also showed in the time course of learning: interior agents acquired their language significantly faster than border agents.)

4.3 Result 3: Temporal development

A certainly surprising result of our experiments was that the learning dynamics did not have much impact on the local network properties that characterize regional learning success. Phrased more strikingly, we could conclude that an agent's location in the network was more influential to his behavioral adaptation than his rationality. Still, there were, of course, notable differences between learning dynamics. The most obvious difference is that BR-learners settle into conventions much faster than RL-learners (see Figure 5).

The slower RL-dynamics moreover showed a very interesting connection between the temporal development of meaning formation and network structure (see Figure 5, bottom picture): there seem to be three phases of conventionalization which affect different network roles. In phase 1 (ca. 0-50) the first agents to adopt a convention, called *founding fathers*, have a much higher degree of connectivity (DC) as the agents of phase 2 (ca. 50-100), called *stabilizers*, who stabilize the language region around founding fathers. By comparing both groups, stabilizers are classical family men, whereas founding fathers are high-connected family men with more global influence. The last agents to adopt a convention, (after ca. 100 rounds) show more and more the mark of globetrotters. This suggests the interpretation that a convention is usually sparked by influential family men, while it takes a locally well-connected set of *real* family men to fix a meaning convention, so that it can also affect the globetrotters.

5. Conclusion

Our results showed that in small-world networks (realized by β-graphs) multiple language regions emerged and stabilized in each simulation run. Whereby the rationality of agents modeled by the appropriate learning dynamics influences the speed of learning, it rarely affects where conventions emerge and stabilize. We were able to show that instead global and local network properties as well have a deep impact of the particular realization of language regions. The main results are: i) cliquishness supports the emergence of language regions, ii) interior agents learn quickly and have family man properties, whereas border agents on average learn late and have the mark of globetrotters and iii) language regions are initiated by high-connected family men. Recent results revealed that the third point owes to the structure of β-graphs, because a new line of our experiments with another network type called *scale-free networks* showed that here initiators are not family men, but super-influential globetrotters.

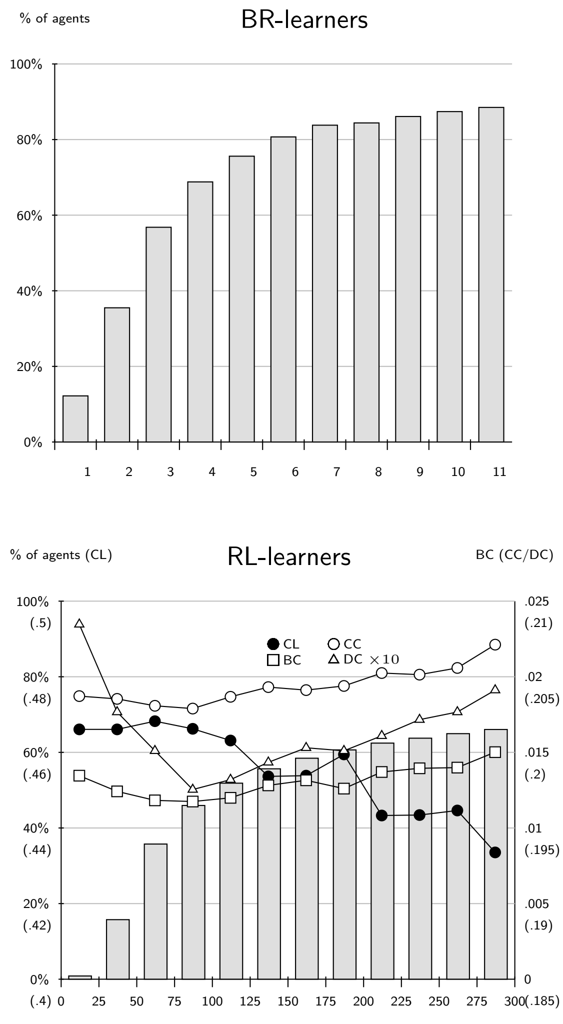


Figure 5: Temporal development of the proportion of agents having settled into their final language for both dynamics (number of simulation steps along the x-axis). The bottom picture also plots the average values for CL, BC, CC and DC for those RL-learners who have settled into their final language during the specified interval of rounds.

A Definitions of network properties

Local properties for a node j in a graph G:

BETWEENNESS CENTRALITY: $BC(j, G)$ is node j's betweenness centrality in network G, defined as the fraction: number of shortest paths between each pair of nodes in G that pass through j, divided by number of shortest paths between each pair of nodes in G;

CLOSENESS CENTRALITY: $CC(j, G)$ is node j's closeness centrality in network G, defined as the fraction: 1, divided by the average distance of j to all other nodes;

CLUSTERING: $CL(j, G)$ is node j's individual clustering value in network G, defined as the fraction: number of connections between neighbors of j, divided by the number of all possible connections between all neighbors of j;

DEGREE CENTRALITY: $DC(j, G)$ is node j's degree centrality in network G, defined as the fraction: number of neighbors of node j, divided by the number of nodes in G;

Global properties for a (sub-)graph G:

AVERAGE CLUSTERING: the *average clustering* of network G is defined as the fraction: sum of *individual clustering* values of all nodes $j \in G$, divided by the number of nodes in G;

TRANSITIVITY: the *transitivity* of network G is defined as the fraction: number of actual triangles in G, divided by number of triads in G; here an actual triangle is a ring of three nodes, a triad consists of two edges with a shared node

DENSITY: the *density* of network G is defined as the fraction: number of edges in G, divided by number of all possible edges in G (= number of edges, if G would be completely connected)

References

Fagyal, Z., Swarup, S., Escobar, A. M., Gasser, L., & Lakkaraju, K. (2010). Centers and peripheries: Network roles in language change. *Lingua, 120*, 2061–2079.

Huttegger, S. M. (2007). Evolution and the explanation of meaning. *Philosophy of Science, 74*, 1–27.

Jackson, M. O. (2008). *Social and economic networks.* Princeton: Princeton University Press.

Ke, J., Gong, T., & Wang,W. S.-Y. (2008). Language change and social networks. *Communications in Computational Physics, 3(4)*, 935–949.

Lewis, D. (1969). *Convention.* Cambridge: Harvard University Press.

Mühlenbernd, R. (2011). Learning with neighbours. *Synthese, 183*, 87-109.

Nowak, M. A., & Krakauer, D. (1999). The evolution of language. *PNAS, 96(14)*, 8028–8033.

Roth, A., & Erev, I. (1995). Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. *Games and Economic Behaviour, 8*, 164–212.

Skyrms, B. (2010). *Signals: Evolution, learning & information.* Oxford: Oxford University Press.

Steels, L. (1997). The synthetic modeling of language origins. *Evolution of Communication, 1*, 1–34.

Wagner, E. (2009). Communication and structured correlation. *Erkenntnis, 71*, 377–393.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature, 393*, 440–442.

Zollman, K. J. S. (2005). Talking to neighbors: The evolution of regional meaning. *Philosophy of Science, 72(1)*, 69–85.

Author Biographies

ROLAND MÜHLENBERND is a PhD student in Linguistics at the University of Tübingen, Germany. With a bachelor in Computer Science and a master in Media Science he has an interdisciplinary background, which is also displayed by multiple interests, especially in Game Theory, Social Sciences & Networks, Artificial Intelligence, Language Evolution and Philosophy.

MICHAEL FRANKE is a post-doc at the Institute for Logic, Language and Information in Amsterdam. His fields of interests are (i) Formal Semantics and Pragmatics, (ii) Philosophy of Language, Mind and Action, (iii) Evolution of Language, Rules and Conventions and (iv) Logic, Decision and Game Theory.

A Comparison of Rule-Based versus Exemplar-Based Categorization Using the ACT-R Architecture

Matthew Rutledge-Taylor, Christian Lebiere, Robert Thomson, James Staszewski, and John R. Anderson
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15218
412-268-6025
mfrt@andrew.cmu.edu, cl@andrew.cmu.edu, thomsonr@andrew.cmu.edu, jjs@cmu.edu, ja@cmu.edu

Keywords:
Modeling, Categorization, Learning

ABSTRACT: *A rule-based approach to categorization is compared with an exemplar-based approach. Both models were developed using the ACT-R architecture. Both approaches yield similar accuracy and are relatively impervious to varying model parameters. Implications for the nature of implicit and explicit knowledge and learning are discussed.*

1. Overview

The research in this paper pertains to the general issue of the relationship between existing cognitive theories of categorization. The use of cognitive models that implement descriptive theories allows for quantitative comparisons of the theories. ACT-R has been used previously to compare and contrast exemplar-based and rule-based approaches to categorization (Anderson & Betz, 2001). This paper describes two groups of ACT-R models of a common task. One set is exemplar-based while the other is rule-based. The results of this paper support the notion that exemplar-based theories and rule-based theories are largely commensurable.

Two groups of ACT-R models of a categorization task are compared. The task was to categorize four types of facilities depicted in simulated satellite images. Each facility type corresponded to one category; the terms facility type and facility category will be used interchangeably, hereafter. Each facility type was defined in terms of the probabilities of the presence or absence of various facility features, none of which were unique to any facility. Human participants were trained to learn the facility categories by studying multiple examples of each. One group of ACT-R models learned the categories by storing the (declaratively represented) examples in memory (i.e., exemplar models). The models in the other group were provided with explicit rules that were supplied to the models a priori.

Within each group of models, there were versions of the model that attended to different features in the examples; in the case of the exemplar models, these versions corresponded to different ACT-R memory retrieval mechanisms (partial matching and spreading activation). The purpose of this modeling effort was to compare the performance of rule-based ACT-R models of categorization to exemplar-based ACT-R models of the task described. The fact that this effort was successful has interesting implications beyond this specific project as it provides evidence for the general commensurability of exemplar and rule-based theories of categorization.

1.1 Category Learning

There are many distinct theories of category learning. Most fall into three main groups: rule-based theories (Goodman, Tenenbaum, Feldman & Griffiths, 2008), prototype theories (Rosch, 1973), and exemplar theories (Nosofsky, 1986).

Rule-based theories are committed to the ability of categorizers to identify the category of an object (or an abstract concept) by testing it against one or more rules. Rules typically take an if/then form whereby the object is deemed to be a member of a category (or, is ruled out) if it satisfies the 'if' conditions of one or more rules. Rule-based theories, such as RULEX, can include the possibility of exceptions (Nosofsky & Palmeri, 1995; Nosofsky, Palmeri & McKinley, 1994; Palmeri & Nosofsky, 1998) and/or probabilistic category assignment (Goodman et al., 2008). The rule-based ACT-R model discussed below employs rules to determine the probabilities that an unlabeled facility is a member of each of four possible categories.

Prototype theory postulates that learned categories are represented, mentally, by prototypes. The membership of an instance to a category is determined by the agreement between the properties of the prototype and the properties of the instance. Multiple-prototype theories allow for multiple prototypes for each category to accommodate non-linearly separable categories.

Exemplar theories postulate that category instances (i.e., exemplars) are memorized individually. Category assignment decisions are made by comparing a new instance to existing exemplars. For example, the judged category can be the one belonging to the exemplar nearest (most similar) to the new instance (i.e., winner-take-all); alternately, the judged category can be based on a function of the combined distances from the new instance to each of the exemplars (e.g., least mean squared distance for each set of category exemplars). Standard declarative memory retrieval in ACT-R would support a winner-take-all categorization process. However, the exemplar-based ACT-R model discussed below makes use of a mechanism called blending, which allows all exemplars to contribute to categorization decisions.

1.2 ACT-R Architecture

ACT-R is a computational implementation of a unified theory of cognition (Anderson et al., 2004; Anderson & Lebiere, 1998). It accounts for information processing in the mind via a set of task-invariant mechanisms, which are constrained by biological limitations of the brain. It consists, primarily, of a set of modules, such as the declarative memory system (DM), and a production system. Each module exposes a buffer, which contains a single chunk, to the rest of the system. Each chunk is a member of a chunk type, and consists of a set of type-defined slots with instance specific values.

Information is processed in ACT-R by the production system, which operates on the contents of the buffers. Each production consists of an if-then condition-action pair. Conditions are typically criteria for buffer matches, while the actions are typically changes to the contents of buffers or actions that trigger operations in the associated modules (e.g., the recalling of a memory). The normal process sequence for a model is to loop through probabilistically selecting an eligible production to fire and executing its effects on the system until no production matches the state of the system, causing the model to stop. The production with the highest net utility (after the effects of noise are factored in) is selected to fire from among the eligible productions.

When a retrieval request is made to declarative memory, the single most active matching chunk is returned. Chunk activation is computed as the sum of base-level activation, spreading activation, mismatch penalty and stochastic noise (see figure 1). Spreading activation is a mechanism that propagates activation from the contents of buffers to declarative memory proportionally to their strength of association. The consequence of this is that chunks in DM that share content with chunks in buffers will have an increased probability of being recalled irrespective of degree of match. Partial matching is a mechanism that allows for chunks in memory that do not perfectly match a retrieval request to be recalled if their activation overcomes a similarity-based mismatch penalty.

$$A_i = B_i + S_i + P_i + \mathcal{E}_i$$

Figure 1. The chunk activation formula in ACT-R. A_i is the net activation, B_i is the base-level activation, S_i is the effect of spreading activation, P_i is the effect of the mismatch penalty, and $\mathcal{E}_i$ is magnitude of stochastic noise.

An advanced memory retrieval mechanism, called blending, differs from standard retrieval in that all chunks in DM that match the retrieval request specification are blended together to create a new chunk, which is retrieved (Lebiere, 1999). This mechanism allows for exemplar categorization models similar to those described in Shi et al. (2010), to be created in ACT-R. These models obey the Luce choice axiom (see figure 2; Luce, 1959), where the weight of each exemplar is based on a similarity metric. The default similarity metric is to compare chunk slot values. In the case of the models discussed in this paper, this amounts to comparing the occurrences of facility features (discussed below).

$$P(i) = \frac{w_i}{\sum_j w_j}$$

Figure 2. Luce choice axiom. The probability that option i is selected is relative to the weighted sum of the pool of options j.

2 Facility Identification Task

Experimental participants were trained to identify four kinds of facilities in simulated geospatial images. Each image is of a single facility (e.g., factory complex) that is composed of a collection of discrete features (e.g., buildings) drawn, probabilistically, from three distinct categories. The three categories of features were: IMINT (image intelligence), representing buildings and other terrain features such as roads and rivers; MASINT (measurement and signature intelligence), representing signals such as chemical concentrations or radiation etc.; and, SIGINT (signals intelligence), representing communication transmissions. There were nine unique IMINT features, seven that represented buildings, and two that represented water features. In contrast, there were only two kinds of MASINT features, while the SIGINT features were entirely homogeneous. Each IMINT could appear at most one time in each image, whereas multiple instances of SIGINT and each MASINT could occur in each image. Additionally, each building (IMINT) could have one piece of rooftop hardware attached to it, or not.

The four facilities were defined by different probabilities for the occurrences of each of the possible features. In the case of the IMINT features, these probabilities simple defined the likelihood of the feature occurring in an instance of the given facility. In the case of MASINTs, SIGINTs, and rooftop hardware the probabilities defined the likelihoods of few or many instances of the feature.

The experiment was divided into two main phases: a training phase and a testing phase. In the training phase the participants were presented with 48 annotated examples of each facility (192 total examples), 16 at a time (in a four by four grid). Participants were not limited in how long they could study the images. Training time ranged from 8 minutes to 73 minutes (mean 24 minutes). In the testing phase the participants were presented with single unlabeled images, one at a time. For each image, the participant was required to report a probability distribution over the four possible facilities indicating the likelihood that the image contained each of the facilities.

3 ACT-R Models

This paper is devoted to comparing and contrasting the performance of ACT-R models of the facility identification task. The main comparison is between models that instantiate an exemplar theory account of category learning and models that instantiate a rule-based account. In common to all the models discussed below are the following details.

The testing phase in the simulations consisted of the presentation and categorization of 300 simulated images of unknown facilities. For each presentation, the ACT-R model holds an instance of a facility frame representing the current facility under examination in the imaginal buffer, which corresponds to the parietal lobe of the brain (Anderson, 2007). The facility chunk-type defines a slot for the facility type, a slot for the total number of IMINTs in the image, a slot for the total SIGINTs, one slot each for the totals of two kinds of MASINT, a slot for the total number of pieces of rooftop hardware on buildings in the image, and one slot for each of the nine kinds of IMINT. Each IMINT slot stores a chunk representing the presence of that IMINT or is left empty. There were an average of 5.05 IMINT features per facility instance.

Three related concepts are used almost interchangeably in this paper: facility chunk, facility frame, and facility exemplar. A facility chunk is an ACT-R representation of a facility instance. Facility exemplars are represented as chunks in declarative memory. The term facility frame is used to refer to schematic structure of a facility and is used in the context of maintaining a facility in an ACT-R buffer.

3.1 Exemplar Models

During the training phase the annotated images are imported into the declarative memory of the model one at a time. For each image, the model temporarily holds a facility frame in working memory by populating the imaginal buffer with appropriate chunk representations of the features present in the image. Once filled, the imaginal buffer is cleared and the facility chunk is committed to memory (DM).

During the testing phase, images are presented to the model one at a time. The model performs a blended retrieval request of DM for a facility frame chunk based on some information available in the images. The facility slot value of the blended chunk is used as the model's answer to the identification question. The model is able to assign probabilities to each category by converting the activations of all exemplar chunks of each category to an aggregate probability using the ACT-R Probability Retrieval (Boltzmann) Equation (see figure 3). Note that the Boltzman equation below is equivalent to the Luce choice axiom (shown above in figure 2), where w = $e^{A/s}$.

$$P_i = \frac{e^{A_i/s}}{\sum_j e^{A_j/s}}$$

Figure 3. Boltzman equation controlling the probability (P_i) that chunk *i* with activation A_i is retrieved relative to the activation levels of all of eligible chunks (*j*). The *s* parameter reflects chunk activation noise.

Two distinct retrieval mechanisms could apply to the recall of facility frames. They are partial matching and spreading activation.

3.1.1 Partial Matching Model

The partial matching version of the ACT-R model uses only the slots that store the counts of the various feature types (and hardware) as part of the retrieval request. This model represents a participant who is not attentive to the particular buildings that are present in a test image. When classifying a facility image, the model compares the feature counts in the image to the counts in facility chunks in DM (see figure 4).

The effect of partial matching is that the model is able to make similarity-based inferences in making facility discriminations. By limiting the model to representing only the numbers of each feature type, this similarity-based inference mechanism is fruitful only if there is a statistically significant difference in the distribution of feature totals for the different facilities. Our results show that such a statistical relationship exists.

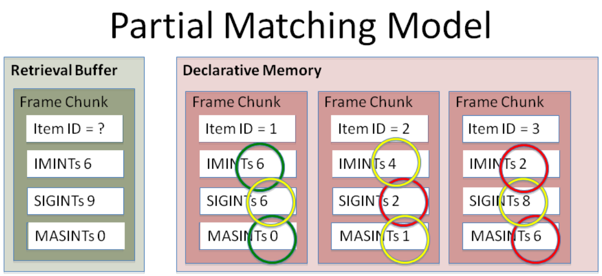


Figure 4. Green circles indicated perfect matching slots values; yellow circles indicated good matches; and, red circles indicate poor matches.

The ACT-R model making use of partial matching only was able to correctly identify the facility in each test sector 46.2% of the time, on a cross-validated 80%/20% training/testing split of the 300 sample scenes. The confusion matrix (see Table 1) listing the probability of classifying an instance of a given facility type as any of the four facility type options shows a pattern dominated by confusion between facilities A and C, and B and D.

Table 1
Confusion matrix for partial matching model

Facility	*A*	*B*	*C*	*D*
A	**.559**	.090	.274	.077
B	.077	**.490**	.116	.316
C	.356	.124	**.375**	.145
D	.108	.288	.180	**.424**

3.1.2 Spreading Activation Model

The spreading activation version of the model ignores the feature counts; instead, the IMINT features in the imaginal buffer form the context of retrieval. The model assembles a facility frame in the imaginal buffer using chunks representing the IMINT features present in the image. Each feature spreads activation to facility chunks in DM that include the feature to a degree inversely proportional to the logarithm of the number of chunks including that feature (see figure 5), a phenomenon known as the fan effect (Anderson, 1974; Rutledge-Taylor & West, 2008; West et al. 2010). This model represents a participant who is solely focused on the particular buildings in the image. When the request for a facility chunk from DM is made, chunks that share IMINT features in common with the image will get a boost in activation, increasing the probability that they will be retrieved.

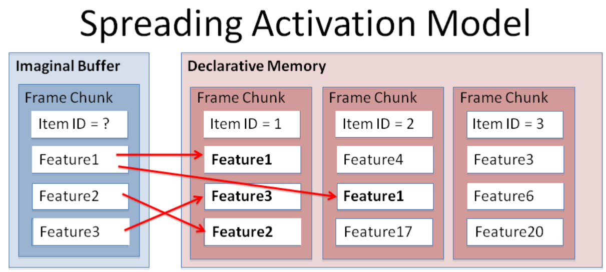


Figure 5. Frames that share features in common with the context in the imaginal buffer receive a boost in activation.

Performance of the spreading activation version of the model was better than the partial matching version. It was able to correctly identify the facility 65.5% of the time, on a cross-validated 80%/20% training/testing split of the data. The pattern of individual confusion probabilities (see Table 2) reflects the overlap between the features likely to belong to each facility, as well as the fact that the number of features increases from facility A to B, C and D, leading to more spreading activation for the latter.

Table 2
Confusion matrix for spreading activation model

Facility	*A*	*B*	*C*	*D*
A	**.585**	.017	.247	.151
B	.006	**.635**	.061	.297
C	.065	.062	**.585**	.287
D	.025	.108	.054	**.813**

3.1.3 Combined Model

A third version of the model, referred to as the combined model, which uses both partial matching over feature counts and spreading activation from IMINT features, was created. In this model, frames include slots for both the feature totals, and for each IMINT feature individually. The count slots are used as retrieval cues (partial matching), while the IMINT features are used as retrieval context (spreading activation). Performance on this was somewhat better than for either model alone. It was able to correctly identify the facility 72.0% of the time, on a cross-validated 80%/20% training/testing split of the data. The confusion matrix is presented in table 3 and displays uniformly good performance.

Table 3
Confusion matrix for combined model

Facility	*A*	*B*	*C*	*D*
A	**.719**	.013	.177	.090
B	.006	**.716**	.074	.203
C	.065	.058	**.691**	.185
D	.035	.133	.079	**.753**

3.1.4 Blending

By default in ACT-R, a retrieval request to declarative memory produces the single chunk representing the frame with the greatest net activation. An exemplar-based model using standard retrieval would be a winner-take-all categorization model. However, we hypothesize that the experimental participant is not making the facility identification judgment based on the single exemplar in memory that best matches the set of features in the target sector. Rather, every exemplar in memory should contribute to the categorization decision. The relative contribution of a chunk is a function of its base-level activation, partial matching and spreading activation. The blending mechanism creates a new chunk of the requested chunk-type that is an aggregate of all the exemplars in memory. The value for each of the new chunk's slots is that which is the best compromise value amongst all the values occurring in all the exemplars, weighted by the activation strength of the exemplars. The blended facility category slot value corresponds to the model's categorization decision.

The outcome of blending is somewhat different than generating a prototype in that the relative contribution of each exemplar is based on its activation, which is affected by the specific retrieval cues (i.e., the specific features present in the facility to be identified). Additionally, there are no persistent prototypes in DM. This is why we consider these models to implement exemplar-based categorization. However, a case can be made that this could be considered a kind of dynamic multiple-prototype learning, if the persistence of prototypes is not necessary.

3.2 Rule-based Models

The ACT-R models of rule-based category learning presented in this paper were created as a proof of concept that a particular choice of rule representation would be effective in producing categorization accuracy approximately equal to, or better than, the exemplar-based ACT-R models. As such, the models do not learn the categorization rules. Rather, optimal rules were assigned to the models. Each rule specified a layer type, the facility to which the rule applied, a multiplicative likelihood factor, and either a single IMINT feature or in the case of rules about countable features (e.g., SIGINTs), a matching quantity. The condition for the rule match is either the presence of a single IMINT feature, or a quantity of countable INTs. The multiplicative factors for the IMINT features were based on a statistical information gain measure for each feature. The count rule factors were estimated, and parameterizable. An additional parameter of the model was the degree of permissible mismatch between the number of count features specified in a rule and the count in the facility to be identified. For example, the rule chunk, (s1 isa rule layer sigint category a value 4.8 factor 3), encodes the rule that the posterior probability of the unknown facility being of category A is three times greater than the prior probability if the facility is within a threshold difference of 4.8 SIGINT features. The threshold and factors were manipulated experimentally. However, a broad range of thresholds and factors results in near ceiling performance in model accuracy.

The models maintained probabilities for each facility category in the goal buffer, and adjusted these probabilities according to the multiplicative factors encoded in all the rules matching the contents of the sector under examination. For each sector to identify, the model applied all of the applicable rules to produce a final probability distribution. The facility assigned the highest probability was interpreted as the model's forced choice response for accuracy evaluation.

4 Results and Discussion

The performance profiles of the rule-following models were largely parameter-invariant. As such, the results for the various combinations of parameters will not be reported; rather, the results presented below reflect the single set of parameters that best matched the accuracy results of the exemplar-based ACT-R model. The count mismatch threshold was 30% (over or under the value specified in the rule), the IMINT rule multiplicative factor was 1.2, the SIGINT factor was 3.0, the MASINT factor was 3.0, and the hardware count factor was 1.2. Table 4 summarizes the relative accuracies of the exemplar-based and rule-based ACT-R models.

Table 4
Comparison of Rule-Based and Exemplar Models

	Rule-Based	*Exemplar*
PM	.476	.462
SA	.657	.655
Both	.755	.720

The version of the rule-following model analogous to the partial matching version of the statistical learning model only applied rules that pertained to object counts. It scored a 47.6% accuracy rate in categorizing unseen sectors, compared to the 46.2% rate of the exemplar model. Another version of the rule-following model analogous to the spreading activation model only applied rules about the presence of specific IMINT features. It scored an accuracy rate of 65.7% compared to 65.5% for the exemplar model. The rule-following model that applied all rules scored 75.5%, while the exemplar model that attended to counts as well as specific IMINTs scored 72.0%.

The agreement between the exemplar learning ACT-R model and the rule-following ACT-R model support the hypothesis that the rules created for the rule following model captured the same information learned by the exemplar learning model. We hypothesize that this is the case because both the exemplar models and rule-based models maximally exploit the information that can be extracted from the data given the parallel limitations imposed of the models.

Human experimental participants scored a mean accuracy of 53.5%. In post-experiment interviews it was revealed that some participants were explicitly aware of the relationships between feature counts and facility categories. However, Given that they outperformed the partial matching models, it is likely that they were also able to detect correlations between specific IMINTs and facility categories. These correlations would have to be applied in an incomplete or imperfect manner as the human participants scored less than the predicted 72% (or better) accuracy of the combined model. An intriguing possibility is that the participants employed a strategy of augmenting an exemplar-based IMINT feature representation with the application of explicit feature count rules.

Experimental evidence suggests that exemplar theories and rule-based theories can make similar predictions, with each accounting for different phases of concept learning (Rouder & Ratcliff, 2006). Exemplars are be relied upon initially; rules are then inferred from the data; and finally, some exemplars are retained to account for rule exceptions. See Anderson & Betz (2001) for an account of how exemplar-based and rule-based models can be combined to produce classification behavior.

The performance equivalence between the two groups of models establishes that functional Bayesian inference can be accomplished in ACT-R either through explicit, rule application or through the implicit, subsymbolic processes of the activation calculus, that support the exemplar model. This should be expected as the semantics and learning mechanisms of the subsymbolic system in ACT-R is fundamentally Bayesian in nature (Anderson, 1990; 1993). Elsewhere it is argued that exemplar models, interpreted as performing importance sampling, provide a plausible mechanism for implementing Bayesian inference (Shi, et al., 2010). This further supports the notion that the blending mechanism of ACT-R, necessary for the current exemplar model, is a cognitively sound alternative to standard memory retrieval in ACT-R.

The comparison of the sets of functionally equivalent models described in this paper is significant as it provides a principled quantitative comparison between two theoretically distinct accounts of categorization. When implemented in a cognitive architecture that obeys a variety of meaningful constraints the two theories, exemplar-based and rule-based, produced equivalent results. We do not interpret this outcome as supporting one theory over another. Rather, we take it to be a lesson when building models that incorporate a categorization component. Specifically, we let the task dictate whether a rule-based or exemplar-based account is most appropriate, rather than a preconceived notion about how categorization ought to be done cognitively.

5 Acknowledgement

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of the Interior (DOI) contract number D10PC20021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained hereon 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.

6 References

Anderson, J. R. (1974). Retrieval of prepositional information from long-term memory. *Cognitive Psychology, 6*, 451-474.

Anderson, J. R. (1990). *The adaptive character of thought*. Erlbaum, Hillside, NJ.

Anderson, J. R. (1993). *Rules of the mind*. Erlbaum, Hillside, NJ.

Anderson, J. R. (2007) *How Can the Human Mind Occur in the Physical Universe?* New York: Oxford University Press.

Anderson, J.R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review, 8*, 629-647

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. (2004) An integrated theory of Mind. *Psychological Review, 111*, 1036-1060.

Anderson, J. R., and Lebiere, C. (1998), *The atomic components of thought*, Erlbaum, Mahwah, NJ, 1998

Goodman, N. D., Tenenbaum, J. B., Feldman, J. & Griffiths, T. L., (2008). A Rational Analysis of Rule-Based Concept Learning. *Cognitive Science,* 32, 108-154.

Lebiere, C. (1999). The dynamics of cognitive arithmetic. *Kognitionswissenschaft* [Journal of the German Cognitive Science Society] Special issue on cognitive modelling and cognitive architectures, D. Wallach & H. A. Simon (eds.)., 8 (1), 5-19.Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis*. Wiley, New York.

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, 115, 39-57.

Nosofsky, R. M. & Palmeri, T. J. (1998). A rule-plus-exception model for classifying objects in continuous-dimension spaces. *Psychonomic Bulletin & Review 5*(3), 345-369.

Nosofsky, R. M., Palmeri, T. J., and McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, 101(1): 53–79.

Palmeri, T. J. & Nosofsky, R. M. (1995). Recognition memory for exceptions to the category rule. *Journal of Experimental Psychology: Learning, Memory & Cognition, 21*(3), 548-568.

Rouder, J. N., & Ratcliff, R. (2006). Comparing exemplar- and rule-based theories of categorization. *Current Directions in Psychological Science*, *15*, 9-13.

Rosch, E. H. (1973). Natural categories. *Cognitive Psychology*, 4, 328-350.

Rutledge-Taylor, M. F. & West, R. L. (2008) Modeling The fan effect using dynamically structured holographic memory. In B. C. Love, K. McRae, & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society*. 385-390. Washington, DC: Cognitive Science Society.

Shi, L., Griffiths, T. L., Feldman, N. H., and Sanborn, A. N. (2010). Exemplar models as a mechanism for performing Bayesian inference. *Psychonomic Bulletin & Review*, 17(4), 443-464.

West, R. L., Pyke, A. A., Rutledge-Taylor, M. F. & Lang, H. (2010) Interference and ACT-R: New evidence from the fan effect. *In Proceedings of the Tenth International Conference on Cognitive Modeling*. Philadelphia, PA: Drexel University.

Author Biographies

MATTHEW RUTLEDGE-TAYLOR is a Post-Doctoral Researcher at Carnegie Mellon University in Pittsburgh, PA. His research interests include modeling theory as well as model building using ACT-R and Holographic Reduced Representations.

CHRISTIAN LEBIERE is a Research Psychologist at Carnegie Mellon University in Pittsburgh, PA. His main research interests are computational cognitive architectures and their applications to psychology, artificial intelligence, human-computer interaction, decision-making, intelligent agents, cognitive robotics and neuromorphic engineering.

ROBERT THOMSON is a Post-Doctoral Researcher at Carnegie Mellon University in Pittsburgh, PA. He received his PhD in Cognitive Science from Carleton University (Ottawa, Canada) in 2011.

JAMES STASZEWSKI is a Professor of Psychology at Carnegie Mellon University in Pittsburgh, PA.

JOHN R. ANDERSON is a Professor of Psychology at Carnegie Mellon University in Pittsburgh, PA.

Evaluation of Two Intelligent Tutoring System Authoring Tool Paradigms: Graphical User Interface-Based and Text-Based

Shrenik Devasani
Stephen B. Gilbert
Virtual Reality Applications Center
Iowa State University
1620 Howe Hall, Ames, IA 50011.
515-294-3092
shrenik@iastate.edu, gilbert@iastate.edu

Stephen B. Blessing
University of Tampa
401 W. Kennedy Blvd.
Tampa, FL 33606.
813-257-3461
sblessing@ut.edu

Keywords:
Intelligent tutoring system, authoring tool, graphical user interface

ABSTRACT: *We describe an evaluation of two intelligent tutoring system authoring tool paradigms, graphical user interface-based and text-based by taking the examples of CTAT and xPST respectively, in two domains, statistics and geometry. We conducted a study with 16 tutor-authors divided into 2 groups (programmers and non-programmers). Our results showed that the GUI-based approach, aided by the visualization of problem-solving strategies, provides a much lower bar for entry when compared to the text-based approach. However, the difference in tutor-authoring time between the two approaches decreases as the tutor-authors gain experience using the respective authoring tools. We also do a theoretical comparison of the two paradigms by applying the cognitive dimensions framework. This research contributes design guidance for architects of tutor authoring systems.*

1. Introduction

Authoring an intelligent tutoring system from scratch is a challenging task since it requires expertise in several fields including cognitive science, computer science and pedagogy. An authoring tool tries to lower the skill threshold required for developing ITSs and also enable their rapid development (Murray, 1999). Past estimates for how long it takes to create an ITS have ranged as high as 200 hours of development time for 1 hour of instruction (Woolf & Cunningham, 1987). Recent ITS authoring tools hope to decrease that by as much as an order of magnitude.

When an authoring tool is being designed, there are several design trade-offs involved, since many of the design decisions that lead to an authoring tool result from conflicting trade-offs. For example, increasing the power of an authoring tool might come at the cost of its ease of use due to the increasing complexity. In this paper, we try to evaluate the effects of the trade-offs involved in the design of GUI-based and text-based authoring tools.

Studies have been conducted to identify the advantages and disadvantages of visual programming, a common technique found in GUI-based authoring tools (Whitley, 1997). Visual representations have shown to be beneficial when the size or complexity of the problem grows (Day, 1988; Polich & Schwartz, 1974). On the other hand, visual representations do not support enough visual elements on a screen at one time, causing low screen density, when compared to textual representations. Therefore, they may not be practical for large problems (Whitley, 1997). We suggest that as tutor complexity increases, the benefits of a GUI will decrease.

For the experiment, we chose CTAT (Koedinger, Aleven, & Heffernan, 2003) and the Extensible Problem Specific Tutor, or xPST (Blessing, Gilbert, Blankenship, & Sanghvi, 2009) as examples of GUI-based and text-based authoring tools respectively. Both CTAT and xPST can be used to develop example tracing tutors (Aleven, McLaren, Sewall, & Koedinger, 2009), for both single strategy problems and multiple strategy problems. Example-tracing tutors are ones in which there is no explicit cognitive model. Therefore, the instruction contained within the tutor is applicable to only a single problem or a small class of problems. This approach trades off wider applicability with easier authoring. However, non-programmers and non-cognitive scientists (e.g., course instructors) have used both authoring tools to create effective ITSs (Aleven, et al., 2009; Maass & Blessing, 2011).

We chose two problem domains of varying complexity from an authoring point of view – statistics and geometry. Statistics problems are sequential in nature and generally have a single solution path. Problems in geometry could have multiple strategies and therefore multiple solution paths.

1.1 Cognitive Tutor Authoring Tools (CTAT)

CTAT supports the development of example-tracing tutors through a technique called "programming by demonstration" (Koedinger, Aleven, Heffernan, McLaren, & Hockenberry, 2004; Nevill-Manning, 1993). Once the interface for a problem has been built, the tutor-author proceeds by demonstrating all possible solution paths to the problem. The interface is typically built using predefined Flash or Java widgets, with some possibility to use existing interfaces. The demonstration of the solution automatically creates a directed, acyclic graph called the "behavior graph", which represents the acceptable ways of solving a problem (Figure 1). The feedback associated with the individual states in the problem (e.g., the hint messages displayed to the student when requested) is added to the tutor by the author through CTAT's GUI.

Though CTAT is capable of supporting the development of both cognitive tutors and example-tracing tutors, we consider only the example-tracing tutor development feature of CTAT in this study.

1.2 Extensible Problem Specific Tutor (xPST)

xPST is an ITS authoring tool that helps rapidly develop example-tracing tutors on existing interfaces such as webpages. An example-tracing tutor is built using xPST by writing a text file that describes the sequences in which individual steps in the problem can be completed by the learner and the answers and feedback associated with each step.

Although text-based, xPST's syntax (Figure 2) was designed such that it was simple enough for non-programmers to use, and also powerful enough for experienced users to build tutors rapidly. The code is entered using a web-based authoring tool that offers syntax-checking.

2. Methods

The experiment involved two independent variables – authoring tool paradigm (text-based or GUI-based) and problem domain (statistics or geometry). Programming level (programmer or non-programmer) was a moderating variable. Dependent variables included a rubric-based quality score for each tutor, the time to develop the tutor, and qualitative data from an exit questionnaire.

2.1 Participants

Participants were recruited through an email advertisement, fliers on campus and word-of-mouth. Sixteen participants completed the study successfully. Each participant took a pre-survey that asked questions about his or her experience with computer programming. A participant was classified as a "programmer" if he or she had taken two or more programming courses and as a "non-programmer" otherwise. The participants were divided into eight groups formed from all possible combinations of programming level (programmer or non-programmer), authoring tool paradigm (text-based or GUI-based) and problem domain (statistics or geometry), with two in each group.

2.2 Materials

The experiment used a between-subjects design: each participant was given the task of building three tutors using a specific authoring tool (CTAT or xPST) for a specific domain (statistics or geometry). All three problems were of the same complexity, as measured by their solutions having six subgoals or steps.

The entire study was conducted online. The participants were allowed to complete their tasks over multiple sessions at their own pace, over a two-week period. The participants were provided with the link to the study webpage that had all the resources and material required in order to complete their tasks. The study webpage contained a brief introduction the field of intelligent tutoring systems. The training material on the webpage included a video tutorial that gave a demo of the step-by-step procedure to be followed while creating a tutor for a sample problem as well as a text tutorial. We estimate the total training time to be about 1-2 hours, though we did not record the exact amount of time individual authors spent with the training materials.

CTAT tutor-authors used Remote Desktop Connection to log in remotely to a Microsoft Windows computer that had CTAT v2.10.0 and Adobe Flash Player 10 pre-installed. xPST tutor-authors logged into the xPST Web-based Authoring Tool (Gilbert, Devasani, Kodavali, & Blessing, 2011) using Mozilla Firefox 3 or higher. The problems for which the tutors were to be

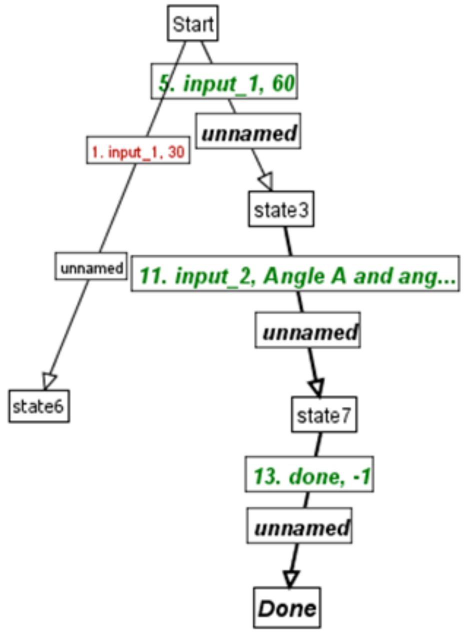


Figure 1: Partial behavior graph of a CTAT tutor

```
sequence
{
        ( (stepA then stepB) or (stepC then stepD) ) then All-Done;
}

feedback
{
        stepA
        {
                answer: 60;
                Hint: “Angle A and angle B are corresponding
                angles.”;
        }
}
```

Figure 2: Partial code of an xPST tutor

built were predefined and the interfaces for the problems were provided to the participants.

CTAT tutor-authors could access the problem interfaces (.swf files) on the remote machine provided to them. xPST tutor-authors were provided with the links to the webpages that contained the problem interfaces. We conducted the study in this way to ensure we were truly investigating just the time to use the ITS authoring system and not the time to set up the system (which was more time consuming for CTAT than xPST) nor the time to set up the problem interface, which is more of an interface or webpage design task.

2.3 Procedures

Each participant was asked to create the three tutors as if he or she was a teacher for that subject preparing homework problems for his or her students. Instructions were provided for each problem that included the problems’ solutions and the types of feedback the tutor must give for each problem, e.g., "If the student enters the median instead of the mean at step 4, display a message that says 'The mean and the median are not necessarily the same.' ". This approach was used to minimize the time spent by participants on pedagogical design. The tutors created by the participant were meant to monitor each step in the corresponding problem. For each problem step, the tutor was to provide exactly one hint. Also, for each tutor overall, there was to be one message that gave

Authoring Tool	Problem Domain	Programmer / Non-programmer	Solution Path Score	Error-Specific Feedback Score
xPST	Statistics	Programmer	6	6
		Non-Programmer	6	6
	Geometry	Programmer	6	6
		Non-Programmer	4.5	5
CTAT	Statistics	Programmer	6	6
		Non-Programmer	6	6
	Geometry	Programmer	6	6
		Non-Programmer	4	6

Table 1: Cumulative scores of created tutors (maximum possible score is 6)

feedback for a specific error that a student might make, e.g. the message mentioned above.

After successful completion of their tasks, the participants received a compensation of $40 in cash and a chance to win $149 in cash through a lottery.

3. Results

We had a total of eight groups with two participants each. Each participant built a total of three tutors, leading to the creation of 48 tutors in all.

3.1 Model Analysis

Each tutor was scored on two criteria – "Solution Path" and "Error-Specific Feedback." A tutor was given a score of 1 under "Solution Path" if it correctly provided tutoring for all possible solution paths in the problem (including providing hints on each step), 0.5 if it correctly provided tutoring for one of the possible solution paths and 0 if it provided completely incorrect tutoring. A tutor received a score of 1 under "Error-Specific Feedback" if it correctly provided the required error-specific feedback and 0, otherwise.

The cumulative scores have been shown in Table 1. Since a group had two participants who built three tutors each, the maximum score possible is six. The model analysis shows that all the tutor-authors who were classified as programmers built tutors that provided accurate tutoring. Tutor-authors who were classified as non-programmers built tutors that displayed accurate tutoring behavior for statistics problems, but slightly less accurate behavior for geometry problems, which were the problems that allowed for multiple solution paths.

3.2 Timing Data

The total time spent in creating each tutor was logged by the respective authoring tools. The web-based authoring tool used by xPST participants calculated the time spent developing a tutor as the sum of the time spent in editing the code and the time spent testing it. The CTAT logger creates a log file that records the time and date, each time the tutor-author interacts with the GUI, from which the total time developing a tutor was calculated. This total time measure for both CTAT and xPST includes the time spent in authoring the tutor as well as testing it.

Figure 3 shows the histogram of the time spent in minutes by the xPST tutor-authors, in building tutors for all three problems.

Figure 4 shows the histogram of the time spent in minutes by the CTAT tutor-authors, in building tutors for all three problems. The log file for the second tutor created by one of the participants (P11) was unavailable.

Figure 5 shows the learning curve for the tutor-authors. It is interesting to see that after the tutor-authors gained experience building three tutors, the average time required in creating a tutor by the groups xPST–Statistics (19 min), CTAT–Statistics (18.75 min) and CTAT–Geometry (18 min) were almost equal. However, the average time required in creating a tutor for the third geometry problem using xPST (52 min) was much higher than the average time required using CTAT (18 min). The geometry problems that we chose involved multiple solution strategies. The results suggest that subtle ordering of steps in multiple-strategy problems is more convenient in CTAT because of CTAT's visual representation of the strategies on the behavior graph.

3.3 Exit Questionnaire Data

All 16 participants answered a short questionnaire after completing their tasks. The questionnaire asked them for their feedback about the authoring tool they had used to create tutors. They were asked to rate the ease of use and power of the authoring tool on a Likert scale of 1 to 5, and to answer open-ended questions about the tool's strengths, weaknesses, and suggestions for improvement. One common theme that emerged from the open-ended questions was that both xPST and CTAT are easy to author once understanding how they

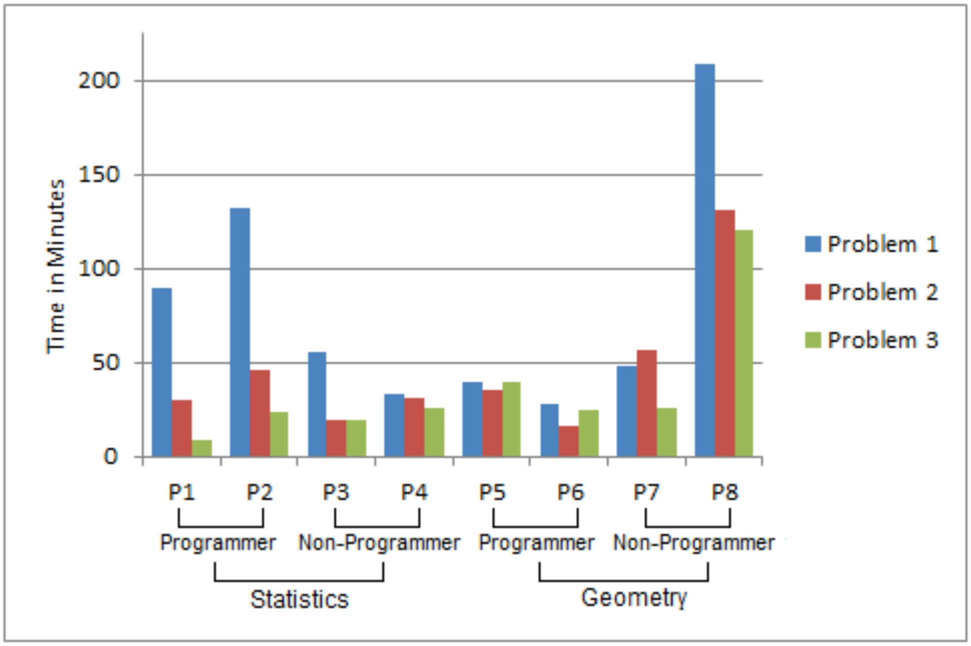


Figure 3: Time spent by xPST tutor-authors

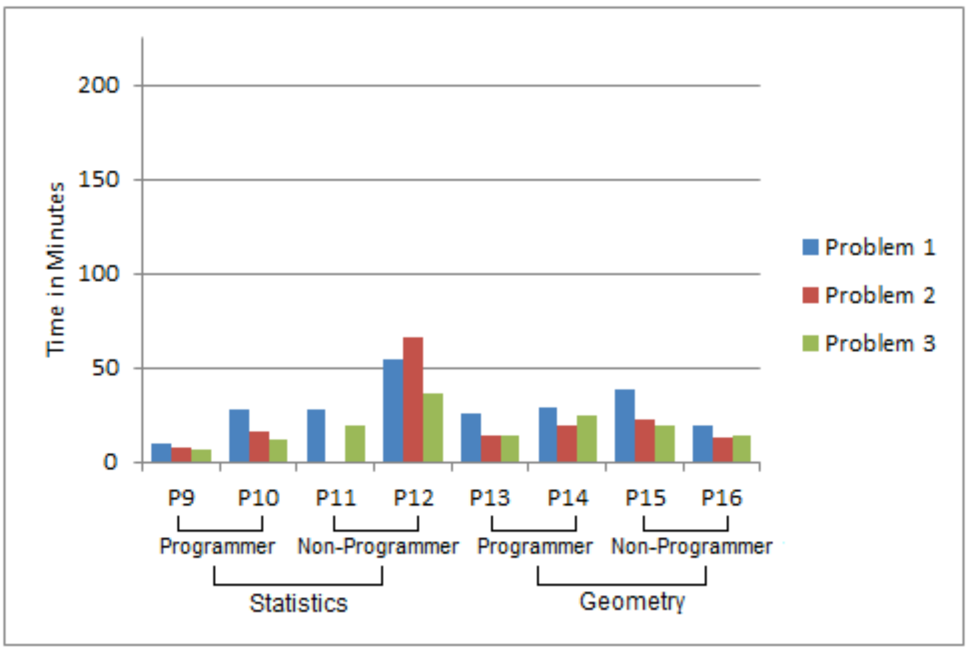


Figure 4: Time spent by CTAT tutor-authors

work. Some quotations are included below that illustrate these:

“Very easy to use once you get a feel for the syntax.” – P7 (xPST – Non-Programmer)

“Once we [sic] understand how to create the tutor then the tool is very simple to use.” – P14 (CTAT – Programmer)

The average ratings from the exit questionnaire have been summarized in Table 2. Both xPST and CTAT were rated slightly higher by programmers for their ease of use when compared to non-programmers.

4. Discussion and Future Work

We used a between-subjects experimental design to prevent order effects and contamination as the participants moved between authoring systems and

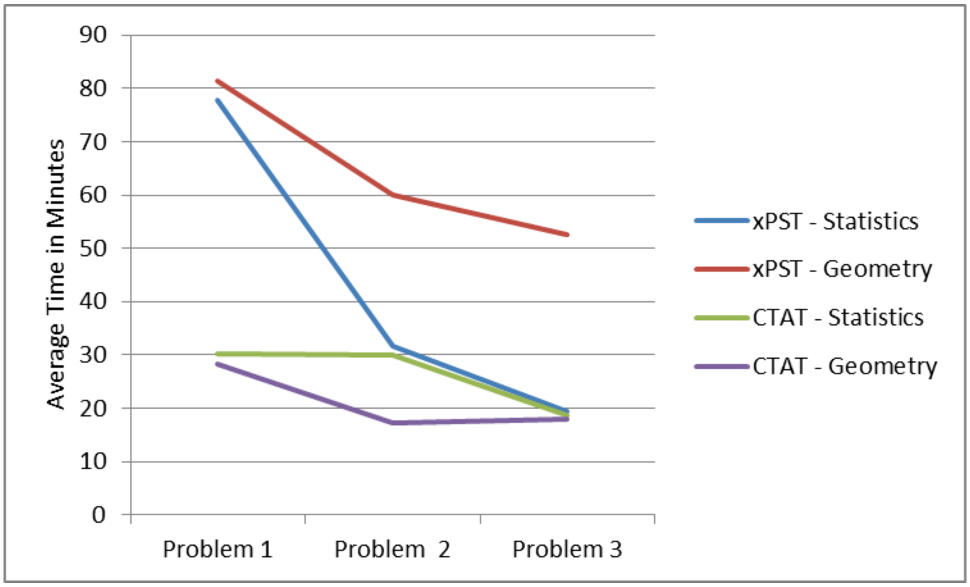


Figure 5: Average time versus problem number (n=4 per group)

Authoring Tool	Programmer / Non-programmer	Ease of Use	Power
xPST	Programmer	4.25	3.50
	Non-Programmer	3.50	3.25
CTAT	Programmer	4.75	4.00
	Non-Programmer	3.50	4.00

Table 2: Average ratings by participants on a Likert scale of 1 to 5 (n=4 per group)

domains. We felt that the comparisons between levels of the independent variables would be compromised if the variables were to be within-subject. As a result, we cannot claim that our results are statistically significant, partially because of the small sample size.

4.1 Applying the Cognitive Dimensions Framework

While the experimental data from this pilot study offers information about learning curves, it is also useful to apply elements of Green & Petre's cognitive dimensions framework (1996), which is helpful for usability analysis generally and was originally used to compare visual programming languages with BASIC. While the framework contains 13 cognitive dimensions, we choose a subset of those for the most applicable comparison. It would be interesting to do an empirical evaluation study involving the comparison of the following cognitive dimensions.

Closeness of mapping is the degree to which elements and operations within the programming language can be mapped directly onto objects and actions within the problem domain. A more direct mapping is preferred. Typically, a visual programming language like CTAT has a closer mapping than a textual language like xPST, especially since textual languages have syntax constraints that do not relate to the problem domain at all. However, xPST has been designed with close mapping in mind, in that steps through the problem correspond to components of code. Mappings in visual programming languages can suffer when looping or branching needs to occur based on calculations of variables, but in the simpler geometry and statistics problems of this study, that issue does not arise. For closeness of mapping, CTAT likely has some advantages over xPST.

Diffuseness/Terseness relates to the number of symbols or lexemes needed to express a concept, in this case, a tutor's structure, and is related to the amount of representation that is viewable simultaneously on screen. In this case, visual programming languages typically suffer once the program reaches a reasonable degree of complexity. In the simpler tutor problems we used, screen real estate was not an issue, but we suggest that with more branches or complex paths to a goal, CTAT would be more diffuse than xPST.

Premature Commitment refers to the extent to which the authoring tool forces the user to make decisions before complete information is available, which often happens when components have significant interdependencies and there are no order constraints. In text-based languages, programmers typically insert stubs to address this issue and return to complete them later. This is the case with xPST. Visual programming languages often force a developer to commit to a certain layout of objects and a certain set of connections before the final layout and connections are finalized. If it is difficult to change these after the system is built, the language has high premature commitment. While CTAT's layout and connections between states are relatively easy to adjust, with higher complexity, xPST likely has some advantages in this dimension.

Secondary Notation refers to the tool's ability to convey meaning about the structure of the program with methods beyond the elements of the program itself. In xPST and other text-based tools, for example, indentation, commenting, and blank lines help illustrate the coherent chunks of the tutor. Some visual programming languages allow boxes or regions to highlight a grouping of related elements, but not all, and often developers with these visual tools spend considerable time not doing the "programming" itself but simply rearranging elements to group or align them appropriately. CTAT supports naming of states and grouping of related states, which aids in visualizing the solution paths. Also, it uses different colors to differentiate between correct and incorrect actions in the behavior graph.

Error-proneness refers to the extent to which the authoring tool induces careless mistakes from the author. The syntax of textual programming languages might be more error-prone, compared to visual programming languages. Moreover, errors might be harder to detect in textual programming languages, especially if the errors are not syntactic in nature. Though xPST was designed such that its syntax is simple to use, it still requires a semicolon as a separator, which could cause errors by non-programmers. Since CTAT uses a graphical user interface to develop a tutor, such careless mistakes occur less often.

Visibility and Juxtiposibility refer to the number of steps required to make desired information visible (the former) and the ability to see separate portions of the system at the same time (the latter). Systems in which visual components hide nested scripts or calculations typically have low visibility and juxtiposibility, requiring the user to right-click to reveal information one component at a time. While CTAT graph nodes contain several pieces of information in their default display, the user is required to dig in to edit information within a node (lower visibility). It is difficult to see the contents of two nodes simultaneously (lower juxtiposibility). Text-based systems like xPST typically have higher visibility (because all the code is present) and high juxtiposibility (via multiple windows).

While we have not addressed all of the cognitive dimensions, we have touched on those that are likely of greatest interest in the context of comparing xPST and CTAT. According to this theory, each tool has some advantages.

4.2 Choose the Best of Both Tools

Generally, a graphical user interface-based approach allows for easier learning initially and a lower bar for entry, as confirmed by the collected data. However, as noted by the theory of cognitive dimensions, once a complex tutor is built, it can be time consuming to edit because of the need to change parameters in multiple locations within the visual structure that are hidden to ensure good visual presentation. One of the advantages of the text-based approach is that debugging and editing an existing tutor may be easier since the entire code is available to the tutor-author at one glance. We conclude by proposing that the best way forward for authoring intelligent tutoring systems and behavior models is a hybrid authoring tool that exploits the synergy between the graphical user interface-based paradigm and the text-based paradigm. Much like common integrated development environments (IDEs) such as Adobe Dreamweaver and Microsoft Visual Studio, the ideal tutor authoring tool would have a “design” tab, which affords tutor-creation through a GUI and a “source” tab that supports the editing of the tutor directly through code. We expect such a tool would cater to programmers and non-programmers, experienced and non-experienced.

5. References

Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*, 105-154.

Blessing, S. B., Gilbert, S., Blankenship, L., & Sanghvi, B. (2009). *From SDK to xPST: A new way to overlay a tutor on existing software*.

Day, R. S. (1988). Alternative representations. *The Psychology of Learning and Motivation: Advances in Research and Theory, 22*, 261-305.

Green, T. R. G., & Petre, M. (1996). Usability Analysis of Visual Programming environments: A ‘cognitive dimensions’ framework. *Journal of Visual Languages and Computing, 7*, 131-174.

Gilbert, S. B., Devasani, S., Kodavali, S., & Blessing, S. B. (2011). *Easy authoring of intelligent tutoring systems for synthetic environments.* Paper presented at the Proceedings of the Twentieth Conference on Behavior Representation in Modeling and Simulation.

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). *Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration.* Paper presented at the Proceedings of Seventh International Conference on Intelligent Tutoring Systems, Berlin.

Koedinger, K. R., Aleven, V. A., & Heffernan, N. (2003). Toward a rapid development environment for cognitive tutors. *Artificial intelligence in education: Shaping the Future of Learning Through Intelligent Technologies, 97*, 455.

Maass, J., & Blessing, S. (2011). *xSTAT: An intelligent homework helper for students.* Paper presented at the Georgia Undergraduate Research in Psychology Conference.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*(1), 98-129.

Nevill-Manning, C. G. (1993). Programming by demonstration. *New Zealand Journal of Computing, 4*(2), 15-24.

Polich, J. M., & Schwartz, S. H. (1974). The effect of problem size on representation in deductive problem solving. *Memory & Cognition, 2*(4), 683-686.

Whitley, K. N. (1997). Visual programming languages and the empirical evidence for and against. *Journal of Visual Languages and Computing, 8*(1), 109-142.

Woolf, B. P., & Cunningham, P. (1987). *Building a community memory for intelligent tutoring systems*.

Author Biographies

SHRENIK DEVASANI recently graduated with his MS in Human Computer Interaction and Computer Science from Iowa State University and is now a research assistant at the Palo Alto Research Center's Intelligent Systems Laboratory. His work involves the building of tools in support of work-process optimization.

STEPHEN GILBERT is Associate Director of the Virtual Reality Applications Center at Iowa State University. His background includes cognitive science and industrial engineering, and he helps direct Iowa State's graduate program in Human Computer Interaction. He is PI on a Live, Virtual and Constructive training contract for the U.S. Army RDECOM STTC.

STEPHEN BLESSING is an Associate Professor of Psychology at the University of Tampa. His main research interest is in authoring tools for intelligent tutoring systems, and he has co-edited a book on the topic.

Implementing Spatial Awareness in an Environment-Agnostic Agent

Simon Gay
Olivier Georgeon
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
simon.gay@liris.cnrs.fr, olivier.georgeon@liris.cnrs.fr

Jong Wook Kim
University of Central Florida
Orlando, FL 32816
Jong.Kim@ucf.edu

Keywords:
autonomous learning, cognitive development, spatial awareness, object persistence.

ABSTRACT: *We designed an autonomous agent that discovers, learns, and exploits basic spatial regularities of interaction with its environment. To do so, we propose implementing a persistence memory system that records bundles of "possibilities of interaction" afforded by objects in the environment, coupled with a local space memory system that represents the agent's surrounding local space (inspired by the vertebrate's tectum). An experiment in a simple simulated environment demonstrates how the agent performs multimodal integration of sensory stimuli, and allocates the origin of such stimuli to "phenomena" located in the external spatial environment. Such mechanisms open the way to implementing agents with minimal preconception of their environment, and to modeling intrinsic motivation in autonomous agents.*

1. Introduction

We address the question of implementing agents with minimal initial preconception of their environment. We define such agents as environmentally agnostic. An environmentally agnostic agent has to autonomously learn to extract relevant information about the environment, and simultaneously organize such information in exploitable knowledge (Georgeon & Ritter, 2011). Environment-agnostic agents are useful to facilitate the development of agent-based models and simulations by reducing the amount of knowledge that must be encoded in the agent. More broadly, studying such agents opens the way to modeling the emergence of new behaviors in autonomous agents.

In previous studies, we started to address this question by implementing an agent that learned hierarchical sequences of behaviors in a bottom-up fashion. To do so, we developed a novel algorithm that we called the intrinsically motivated schema mechanism (Georgeon, Ritter, & Haynes, 2009; Georgeon & Ritter, 2011). With this algorithm, the agent was able to autonomously capture and exploit hierarchical sequential regularities afforded by the environment. This mechanism implemented intrinsic motivation in that the agent's behavior was driven by predefined low-level behavioral proclivities that gave rise to higher-level behavior. This approach stands in contrast from goal or task-directed navigation algorithms (e.g., Batalin, Sukhatme, & Hattig, 2004; Frommberger, 2008). It also differs from classical reinforcement learning techniques (e.g., Sutton & Barto, 1998) in that it addresses the question of developmental learning (i.e., fast learning during the agent's development) (e.g., Lungarella, Metta, Pfeifer, & Sandini, 2003) rather than learning over many trials (often thousands in classical reinforcement learning). In particular, our agent received no predefined reward when a final goal was achieved and we did not implement backward propagation of a reward value.

A subsequent study (Georgeon, Cohen, & Cordier, 2011) showed that an agent equipped with such a sequential learning mechanism was able to acquire basic navigation skills in an open space environment. This study, however, also showed the limits of this purely sequential approach when applied to spatial regularity learning. For example, the agent was unable to notice that two different sequences of movement may lead to the same point in space. Moreover, the agent was unable to discover the persistence of objects. The agent stopped pursuing a target of interest when the target was lost by the sensors (hidden or out of span). To overcome these kinds of limits and to move on toward higher-level learning, we now address the question of the autonomous discovery of spatial regularities. We refer to this issue as implementing mechanisms of spatial awareness in an environmentally agnostic agent.

Our mechanism, a spatial awareness mechanism in an environment-agnostic agent, was inspired by the brain structure most natural organisms have, whose activation maintains some geometrical correspondence with the animal's local surrounding environment. We refer to the mushroom body in the case of insects, and the tectum in the case of vertebrates, also called the colliculus in the case of mammals (e.g., Cotterill, 2001).

In this study, we advocate implementing two initial mechanisms: the *persistence memory* and the *local space memory*. The persistence memory is a long-term memory that memorizes associations of interactions and stimuli based on their co-occurrence. We name such associations by the term *bundle*. This term refers to pragmatic epistemology (e.g., Hume, 1739) that postulates that the knowledge of objects is constructed through usage rather than given a priori. In this framework, Hume proposed the bundle theory of objects. This theory posits that objects consist only of the collection of their properties observed through interaction. Accordingly, we expect our agent's bundles to represent objects in the environment in the form of possibilities of interaction. The second mechanism, the local space memory, is inspired by the tectum in the vertebrate's brain, and consists in an internal geometrical counterpart of the surrounding environment. These two mechanisms constitute the agent's *spatial system*. The spatial system has two objectives: it allows the agent to perform a spatially-organized multimodal integration of sensory stimuli, and it makes the agent able to project the consequences of its actions in an egocentric referential, possibly beyond the range of the agent's perception.

We propose a design methodology that begins by indulging some hard-coded preconceptions to get the spatial system running. In this first step, we setup and demonstrate the coupling between the intrinsically motivated sequential system and the spatial system. The second step consists of progressively removing the preconceptions from the spatial system in order to move toward an agent as much agnostic as possible. Following this approach, we organized the paper in two parts. The first part (Section 2) presents the initial experiment made with the hard-coded spatial system. This experiment illustrates how the agent works. From the lessons learned in this initial experiment, we list the infringements of the principle of agnosticism that need to be addressed. The second part (Sections 3 and 4) reports our algorithms that start addressing these infringements. Finally, the paper discusses our results and draws recommendations for future work.

2. Initial experiment

We implemented an autonomous agent in the environment shown in Figure 1. The agent is represented as a shark. Both the agent mechanism and the environment are implemented in Java. We use the *grid unit* as distance unit. A grid unit correspond to the length of a side of an elementary block object. So far, the environment is static: the agent is the only thing that moves. The agent has four primitive *possibilities of action*: (a) move forward (approximately one grid unit), (b) turn approximately $\pi/4$ to the left, (c) turn approximately $\pi/4$ to the right, and (d) eat fish. The agent moves freely in a continuous space and gets uneven from the grid. The agent has a visual system of 12 pixels covering a total span of π radian. Each pixel reports the dominant color seen in its $\pi/12$ corresponding span. The filled cells and the surrounding perimeter represent walls where the agent would bump if it tries to move through them. The agent cannot see through objects (wall, alga or fish). The agent also has a "9-pixels" tactile system (a 3x3 matrix) that detects adjacent objects or objects below the agent (alga or fish). Fish and alga feel *soft*. Walls feel *hard*.

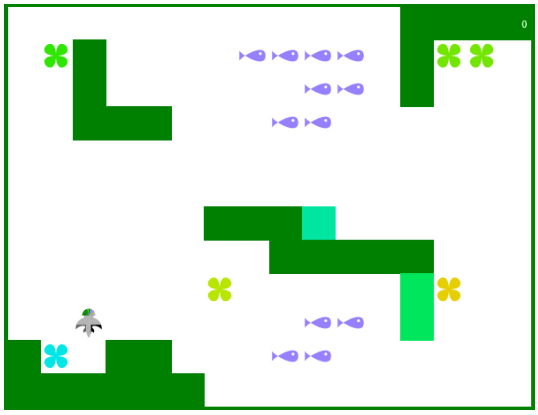


Figure 1: The agent in the environment.

The agent's behavior is generated by the intrinsically motivated sequence learning mechanism described in our previous work (Georgeon & Ritter, 2011). We implemented the local space memory with a radius of 2 grid units. When a co-occurrence of two or more stimuli is detected, the bundle formed by these stimuli is constructed in persistence memory and a pointer is placed in the local space memory to follow the relative displacement of this bundle when the agent moves.

2.1 Analysis of an example run

A representative run can be seen online (Georgeon, 2011). The first two hundred steps of this run are represented in Figure 2, using a technique of activity trace representation developed in a previous study (Georgeon, Mille, Bellet, Mathern & Ritter, 2011). The various tapes show the sensory and internal state of the agent at each step, as described next. A step is a cycle of interaction between the agent and the environment.

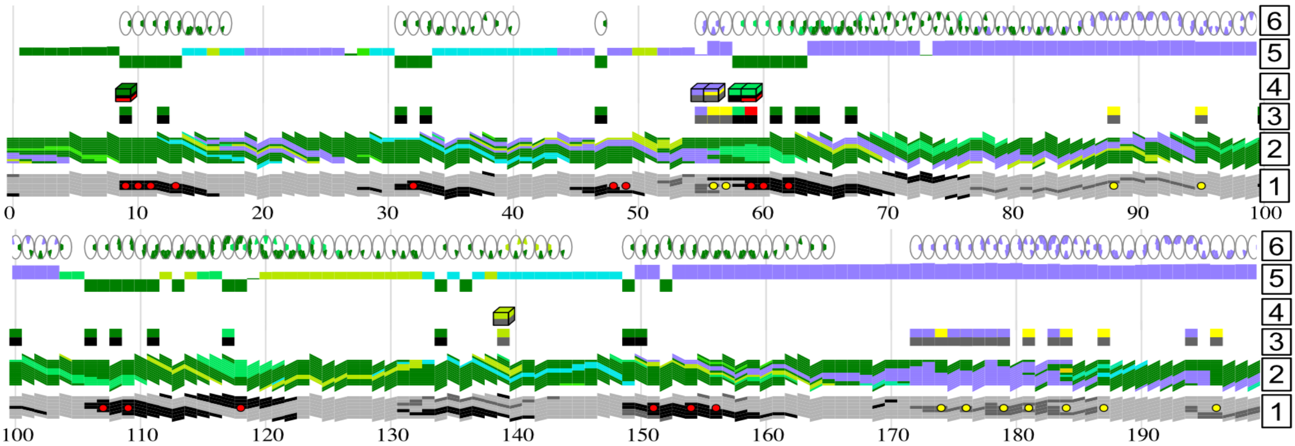


Figure 2: Example activity trace (upper part: steps 0-100, lower part: steps 100-200).

In Figure 2, Tape 1 represents the agent's tactile system (light gray: touching nothing, intermediary gray: touching *soft*, black: touching *hard*). The touching in front of the agent is represented in the center of the tape. The touching to the agent's left side in the upper part, and to the right side in the lower part. The touchings below and behind the agent are not represented. Additionally, red circles represent bumping walls, and yellow circles represent eating fish. The agent experiments bumping on dark green walls on steps 9 through 13, and then significantly reduces bumping on such walls. Similarly, the agent learns bumping on light green walls on steps 59 and 60.

Tape 2 represents the agent's visual perception. The twelve visual pixels are represented vertically as rectangles when the agent moves forward, and as trapezoids when the agent turns. This tape shows colored object traversing the visual field as the agent moves and turns.

Tape 3 represents the co-occurrences of interactions from different sensory modalities. For example, on step 55, the gray square associated with the blue square represents the co-occurrence of touching and seeing a fish. On step 56, the gray square associated with the yellow square represents the co-occurrence of touching a fish and eating a fish. Co-occurrences trigger the construction of bundles, or either confirm or infirm existing bundles.

Tape 4 represents the construction of bundles from co-occurrences. For example, on step 9, the agent constructs a bundle made of the association of touching, seeing, and bumping a wall (green, black, and red cube). On step 55, the agent constructs the bundle of seeing and touching a fish (gray and blue cube). On step 56, the interaction of eating is added to this bundle (gray, blue, and yellow cube).

Tape 5 shows a bar-graph whose color represents the focus of the agent's current *attention*, and whose value represents the associated attractiveness (positive or negative). For example, during steps 1 through 8, the agent focuses on the dark green color just because it is the preeminent color in its visual field. This colors has a mild attractiveness because the agent has not yet learned how to interact with it. On step 9, the agent associates this color with bumping, which makes this color become repulsive. Conversely, on step 56, the blue color becomes highly attractive when it gets associated with eating a fish. At any point in time, the most attractive or most repulsive bundle in the surrounding space wins the agent's attention. The agent has a proclivity to move toward the object of current attention if it has a positive attractiveness, or to turn away from the object of current attention if it has a negative attractiveness.

Tape 6 represents the agent's local space memory, that is, the memory of bundles surrounding the agent. The agent's surrounding space is represented as an ellipse, with the front of the agent being on the right. For example, on step 9, the green area in the ellipse represents the agent being *aware* of the wall in front of it. On steps 188 and 189, the trace shows that the agent does not see nor touch any fish, but it is still aware of a fish on its rear-right side (blue area in the ellipse). This awareness causes the agent to turn back towards that fish and eat it on step 196.

In summary, this experiment illustrates how we envision implementing spatial awareness in an intrinsically motivated agent. The agent was able to gradually learn the associations of possibilities of interaction afforded by persistent objects in the environment. Such associations were memorized internally in the form of bundles. Bundles have a value (attractiveness) related to the possibilities of interaction that they afford. The agent maintains a memory of the position of bundles in local space memory. This memory, in turn, impacts the agent's sequential behavior.

2.2 Infringements of the principle of agnosticism

When implementing this agent, we had to hard code some presuppositions about the coupling between the agent and the environment. We hard coded what co-occurrences were possibly interesting:

a.1 Bumping while touching something with the frontal tactile sensor.
a.2 Eating while touching something with the tactile sensor below the agent.
a.3 Touching something with the frontal tactile sensor while seeing a specific color within the two central visual pixels.
a.4 Eating while seeing a specific color within the two central visual pixels.

We hard coded the agent's knowledge of its basic geometry:

b.1 The position of the tactile sensors in the egocentric reference.
b.2 The maximum radius of the local space memory (2 grid units).

We hard coded the consequences of the agent's actions in the local space memory:

c.1 The move forward action generates a translation of one grid unit in the local space memory.
c.2 The turn action generates a rotation of $\pi/4$ in the local space memory.

To move towards agnostic agents, such hard coded presuppositions should be replaced with autonomous learning mechanisms. To address this concern, we implemented the autonomous learning algorithms described in the next section.

3. Learning correspondence between sensors, actuators, and space

By implementing an algorithm that autonomously learns the correspondence between the values returned by sensors, the agent's actions, and the surrounding local space, we aim at developing a general space-aware system as independent as possible from its sensory and motor configuration. To do so, we first address presupposition b.1 by implementing an algorithm to learn the correspondence between sensors and local space (Section 3.1). Then, we address presuppositions c.1 and c.2 by implementing an algorithm to learn the geometrical transformations that apply to the local space memory depending on each possible action (Section 3.2).

3.1 Correspondence between sensors and local space

This first step consists in learning the structure of the sensory system. This point has a paramount importance as it allows the agent to determine the provenance of stimuli in the surrounding space. The agent can then generate an internal image of the environment that it can manipulate. As our agent is supposed to be agnostic, the algorithm described next is designed to use uninterpreted values returned by sensors.

We called this algorithm the *sensor mapping algorithm* that learns the correspondence between the values returned by sensors and the agent's surrounding local space. This algorithm relates to existing algorithms that allow robots to exploit uninterpreted sensors (e.g., Pierce & Kuipers, 1997). Its specificity, however, is that it constructs a representation of how the sensors cover the surrounding space. We call this representation the *sensory space structure*.

The sensor mapping algorithm uses sensors for which each returned value can be related to the presence of a certain property, for example, an object, on a unique point of the surrounding local space. Specifically, it is intended to work with sensors that return rough information on the distance of the first detected object. Examples of such sensory systems are a stereoscopic visual system that returns approximate distance and color for each pixel, a sonar system that returns distance and echoic property, and a whiskers system (vibrissa) that returns approximate distance and tactile property. We formalize such a requirement as follows:

Each sensory *modality* consists of a set of directional *probes* (e.g., a single whisker, or a "light cone" generating a "pixel" in a visual system). The positions and directions of probes are initially unknown. The probes may not be straight but they must be fixed with regard to the agent (if not, the algorithm must be run for each configuration of the probes). Each probe returns two numerical values: *A* (abscissa) and *S* (stimulus). The value *A* reflects the position of the first object detected along the probe (the object's abscissa along the probe). The only condition on this abscissa is of being a monotonic function of the distance of the object from the agent. This condition needs to consider the fact that objects may mask other objects behind them. The metrics of the abscissa is, however, unknown and may not be linear. These metrics may not be consistent across probes and modalities. The value *S* reflects a physical property of the detected object (e.g., the color for vision, the tactile feeling for touch), or absence of object (e.g., touch nothing).

This set of assumptions indicates that each tuple [*probe*, *abscissa*] corresponds to a single *Point of detection* (P_d) in the agent's surrounding space. Each P_d in the environment is represented by a *Point of sensation* (P_s) in the sensory space structure. A point of sensation is said active if the value *A* returned by the corresponding *probe* is greater than the point's *abscissa*. This means that every sensor used by the sensor mapping algorithm is considered as an array of binary sensors, represented by a set of points of sensation. The sensor mapping algorithm gradually adjusts the positions of P_ss in the sensory space structure to reflect the actual positions of P_ds in the environment, starting from any arbitrary configuration (random or implementing an inborn assumption). It relies upon the assumption that the distance between two points is proportional to the average delay between changes of activity or *S* value of the corresponding probe at each of these two points. The P_ss are placed to optimize the consistency between the delays in the changes of values and P_ss' distance in the sensory space structure.

Once the sensory space structure is learned, the agent can localize a place in the environment as the *origin* of the stimulus. Because the metrics of the sensory space structure does not rely on the metrics of the abscissa of the probes, the localization of the origin is consistent across modalities. Therefore, the sensor mapping algorithm supports a spatially-organized multimodal integration of stimuli. For example, the agent can determine that the origin of a specific tactile stimulus *soft* and the origin of a specific visual stimulus *green* are located at the same place in the environment. Allocating an origin to stimuli implies assuming that stimuli have a *cause* in the environment. Such cause can be called a *phenomenon*, typically defined as any observable occurrence. To external observers, these phenomena correspond to physical objects in the environment (e.g., walls, alga, fish). The agent, however, does not see these objects as we see them, nor does it allocate them the same utility as we do, which is why we refer to the objects as phenomena from the agent's viewpoint.

The agent can construct an *origin map* that represents the location of the phenomena in the environment. Such an origin map is, however, not enough to have an operational representation of the environment. Additionally, the agent needs to learn the relation between its motor actions and the origin map. This question is addressed next.

3.2 Correspondence between actions and local space

This algorithm is called the *motion mapping algorithm*, which learns the correspondence between the actions of the agent and the geometrical transformations in the agent's origin map. This algorithm addresses presuppositions c.1 and c.2 introduced in Section 2.2.

The motion mapping algorithm consists first of computing a vector field that describes the relative movements of the *origins* in the origin map when the agent moves. This vector field can be thought of as an "optic flow" (Figure 4) in the image made of the points of sensation of the sensory space structure. Because the resolution of this image may be low, we used an algorithm inspired by the *insect eye algorithm* developed by Franceschini, Pichon and Blane (1992). This algorithm estimates the movement by measuring the time between variations of values in a point of sensation and its neighbors. Note that the goal of the motion mapping algorithm is not to anticipate complex consequences of actions in the environment (such as the trajectory of objects in motion) nor to allow complex navigation and localization in space (e.g., Mataric, Meyer, & Wilson, 2009; Meyer, Guillot, Khamassi, Pirim, & Berthoz, 2005), but only to learn the relation between primitive actions and the local space.

We assume that the agent cannot move its body parts but can only move as a whole block in a two-dimensional environment. With this assumption, the agent's actions can be expressed as the sum of a translation and a rotation. Consequently, the resulting geometrical transformations in the local space memory consist of the sum of a translation and a rotation in the opposite direction. The algorithm computes the value of this translation and this rotation by measuring the average translation and rotation in the vector field.

Once the translation and rotation values are known, the agent can apply them to the origin map to follow up the relative positions of phenomena when the agent moves. This approach is related to map learning algorithms based on occupancy grids (Elfes, 1989). For example, if a wall is on the agent's left side, and the agent makes a rotation step to right, then the agent knows that the wall moved behind, even though the agent cannot see the wall anymore. Moreover, the origin map is now related to the agent's possibilities of actions. For example, the agent can estimate the distance of phenomena in terms of the actions needed to reach them.

3.3 Bundle construction

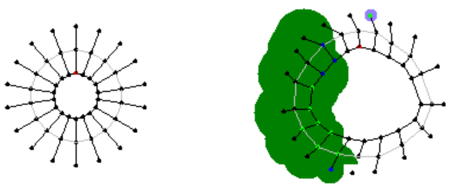
As noted in Section 3.1, origin maps are consistent across sensory modalities because they are based on delays during movements rather than on the metrics of sensors. Therefore, the agent can infer that different stimuli from different sensory modalities are "caused" by the same phenomenon when the origins of such stimuli overlap. The agent creates a bundle to represent the set of the different interactions afforded by this phenomenon. Once learned, the bundles are memorized in persistence memory, and can be subsequently recognized. For example, the agent creates the bundle of touching *hard* from the tactile map, and seeing *dark green*, from the visual map to represent the phenomenon "wall". This mechanism eliminates presupposition a.3. The agent can then subsequently enrich the "wall bundle" by adding the interaction "bump". Then, the sequence learning mechanism will cause the agent to avoid walls when it recognizes them, as reported in Section 2.

4. Second experiment

We implemented these algorithms in a similar agent as presented in Section 2, in the same environment as in Figure 1. We, however, modified the visual and the tactile system to provide more precise input to the algorithms. The visual system now has a resolution of 5° over a total span of 180° (36 "pixels"). Each pixel returns the color and the distance of the first detected object, with a maximum range of 20 grid units. The tactile system is composed of 18 whiskers distributed all around the agent. Each whiskers return the distance and a tactile property of the closest object, with a maximum range of 1.5 grid unit.

4.1 Sensor mapping

The sensor mapping algorithm was tested with the tactile system, with a resolution of 3 points of detection on each whisker at a distance of 0.5, 1, and 1.5 grid units from the center of the agent. Figure 3 allows a comparison of the actual points of detection (Figure 3.a) with the learned points of sensation in the tactile sensory space structure (Figure 3.b). This result was obtained after 1000 steps starting from an initial condition where the individual points of sensation were placed randomly (independently from the whisker to which they belong). This result shows that the agent was able to approximately learn the configuration of its whiskers. The whiskers, however, appear "shrinked". We believe that the precision on the whiskers' length could be improved by considering the movement of the agent learned from the motion mapping algorithm. This would involve interweaving the sensor mapping algorithm with the motion mapping algorithm in the future developments.



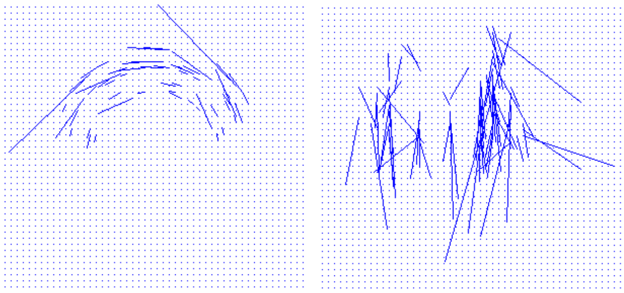
(a) (b)

Figure 3: the real tactile system (a) and the sensory space structure (b) given after 1000 steps. Black radial lines represent whiskers and the black circle line represents the whiskers' basis.

Figure 3b also provides a representation of an instance of the tactile origin map when the agent is sensing a "wall phenomenon" on its left side (dark green) and a "fish phenomenon" on its front (blue).

4.2 Motion mapping

Figure 4 reports examples of vector fields computed by the motion mapping algorithm applied to the visual system. To obtain these results, we, however, hard coded the visual sensory space structure rather than learning it with the sensor mapping algorithm. As noted in Section 4.1, merging these two algorithms remains a challenge that we plan to address in future studies.



(a) (b)

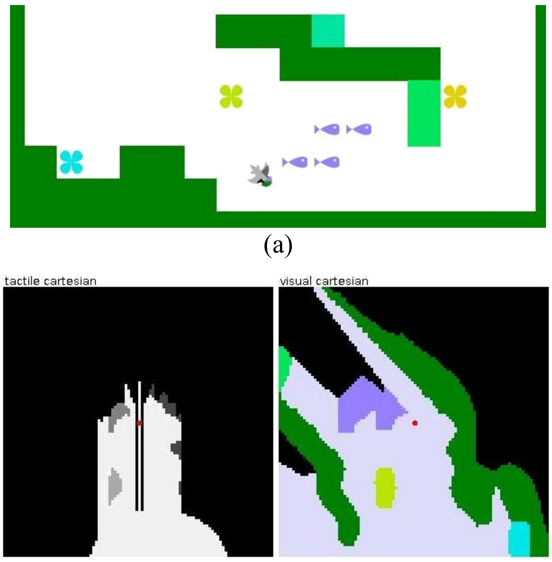
Figure 4: average movement flow given by the visual system, for a rotation (a) and a translation (b).

The average translation and rotation vectors are then computed for each action. The Table 1 summarizes the coefficients measured after 100 steps. Coefficients are the ratio between the linear or angular speed and the distance (in grid unit) or angle (in radius) covered in one simulation step. Even though there is a non negligible error, translation and rotation actions are recognizable.

Table 1 : real and measured translation (Tx and Ty) and rotation (Rz) coefficients.

action	Real coefficients	Measured coefficients
Move forward	Tx = 0 Ty = 0.333 Rz = 0	Tx = 0.022 Ty = 0.376 Rz = $1.19 . 10^{-4}$
Turn right	Tx = 0 Ty = 0 Rz = $1.75 . 10^{-3}$	Tx = 0.058 Ty = 0.017 Rz = $1.53 . 10^{-3}$
Turn left	Tx = 0 Ty = 0 Rz =$-1.75 . 10^{-3}$	Tx = 0.061 Ty =-0.015 Rz =$-1.51 . 10^{-3}$

Figure 5.b shows an instance of the origin map for the tactile system, using whiskers with 15 points of detection each. In this figure, colored areas represent the origin of tactile stimuli: touching soft (light gray), touching edible (middle gray), touching hard (dark gray), empty (white), and black areas indicate untouched areas.



(a) (b) (c)

Figure 5: figure (a) shows the actual position of the agent in the environment. (b) map represents the tactile origin map, (c) the visual origin map. The red point shows the position of the agent in its own egocentric reference, the front of the agent is on the top.

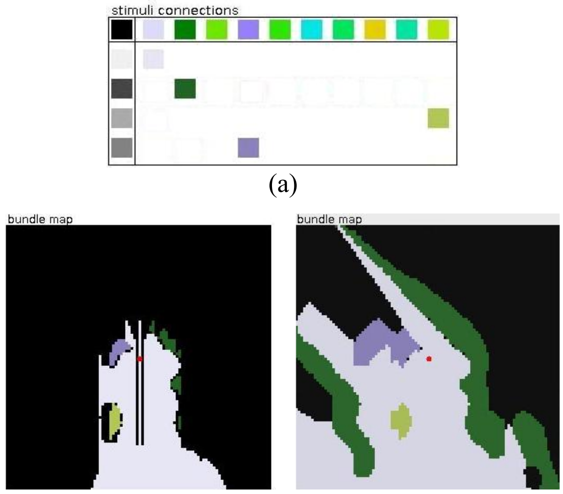
Figure 5.c shows the instance of the origin map for the visual system in the same situation. In both of these figures, the agent's location is represented by a red point and the agent's front is oriented upwards. More precisely, the agent keeps track of the probabilities of different phenomena at each location but the colors in the figures only represent the most likely phenomenon at each location. Figure 5.a represents the corresponding situation of the agent in the environment. Figure 5 shows that this mechanism provides the agent with a sense of persistence of phenomena: in the lower part of the visual map, the yellow and blue phenomena are still present in the visual origin map while being outside of the agent's visual span.

4.3 Bundle construction

As introduced in Section 3.3, bundles are constructed when different sensory stimuli have overlapping origins. Figure 6 illustrates this mechanism. Figure 6.a summarizes the bundles constructed in this instance by associating visual stimuli (x axis) with tactile stimuli (y axis): empty, hard, soft, and edible.

Figure 6.b shows the agent's local space memory in the same instance as presented in Figure 5. In Figure 6.b, bundles are represented by their colors but actually are multimodal representations of phenomena that are spatially localized in the agent's surrounding local space.

These bundles are also used to recognize and localize phenomena according to partial perceptions, by completing missing sensory modalities. For example, in the instance shown in Figure 5.a, the agent can see green object in front of it, but cannot touch them. The agent can determine the missing tactile stimulus according to the learned bundles. In this case, there is one bundle which include the visual green stimulus: the "wall" bundle. The phenomena corresponding to green walls are then added to the local space memory. Figure 6.c shows the local space memory completed by such a recognition system.



(a) (b) (c)

Figure 6: Bundle construction. (a) composition of bundles, y axis: empty, hard, soft, edible, x axis: color. (b) local space memory containing bundles represented by their color. (c) local space memory completed with most probable bundles according to partial perception.

5. Discussion and Conclusion

We propose the implementation of a spatial system to enable an autonomous agent to keep track of objects in its environment. Such system improves the agent's ability to construct increasingly elaborated behaviors. This implementation is part of an ongoing study of how an intrinsically motivated agent becomes aware of the world in which it exists. We believe that the algorithms presented here shed some light on this question by illustrating the relations between the capacity of an agent to orient itself in space and its capacity to allocate a "cause" to its perceptions in the world (phenomena). In particular, this work confirms the importance of time, delays, and sequences in a cognitive system, as many recent studies tend to show (e.g., Nicolelis, 2011). All our algorithms involve time: the sensor mapping algorithm constructs spatial dependencies from temporal dependencies, the motion mapping algorithm learns relations between actions and space, and the agent's decision process is based on sequence learning. We argue for a methodology relying on techniques of activity trace analysis to study temporal dependencies in a cognitive system.

More practically, this work opens the way to modeling agent's behavior without having to program specific behavioral rules and predefined sensors. This will facilitate agent modeling in the future, and will facilitate studies on the emergence of complex behaviors.

Our current implementation, however, still has limitations. One limitation is that the sensor mapping algorithm and the motion mapping algorithm remain to be merged together. The agent should simultaneously learn the consequences of its actions and the structure of its sensory system. We believe that the separation of these algorithms causes imprecision in the whole process that still prevented us from being able to set up a comprehensive experiment to demonstrate the overall improvement of the agent's behavior. Another limitation is that some hard-coded presuppositions still remain. Specifically, presuppositions a.1 and a.2 require creating bundles by associating tactile or visual stimulations with active interactions such as bumping or eating. Addressing this limitation requires taking vision and touch as active processes and merging the control of these processes with the intrinsically motivated sequence learning mechanism. We plan on addressing these questions in future work.

6. Acknowledgment

This work was supported by the Agence Nationale de la Recherche (ANR) contract ANR-10-PDOC-007-01.

7. References

Batalin, M. A., Sukhatme, G. S., & Hattig, M. (2004). Mobile robot navigation using a sensor network . In *Proceedings of the Conference IEEE International Conference on Robotics and Automation*, 636-642.

Cotterill, R. M. (2001). Cooperation of the basal ganglia, cerebellum, sensory cerebrum and hippocampus: possible implications for cognition, consciousness, intelligence and creativity. *Progress in Neurobiology, 64 (1)*, 1-33.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22 (6), 46-57.

Franceschini, N., Pichon, J. M., & Blanes C. (1992). From insect vision to robot vision. *Philosophical Transactions: Biological Sciences, 337 (1281)*, 283-294.

Frommberger, L. (2008). Learning to behave in space: a qualitative spatial representation for robot navigation with reinforcement learning. *International Journal on Artificial Intelligence Tools*, 17 (3) , 465.

Georgeon, O. (2011). Autonomous learning through experience. Retrieved from the web November 10th, 2011. http://e-ernest.blogspot.com/2011/11/ernest-104.html

Georgeon, O.L., Cohen M., & Cordier, A. (2011). A model and simulation of early-stage vision as a developmental sensorimotor process. In *Proceedings of Artificial Intelligence, Applications and Innovations (AIAI 2011)*, 15-18.

Georgeon, O. L., Mille, A., Bellet, T., Mathern, B., & Ritter, F. E. (2011). Supporting activity modelling from activity traces. *Expert Systems*, in press.

Georgeon, O.L. & Ritter, F.E. (2011). An intrinsically-motivated schema mechanism to model and simulate emergent cognition. *Cognitive Systems Research,* in press.

Georgeon, O.L., Ritter, F.E., & Haynes, S.R. (2009). Modeling bottom-up learning from activity in soar. *In Proceedings of the 18th Annual Conference on Behavior Representation in Modeling and Simulation (BRIMS),* 09-BRIMS-016, 65-72.

Hume, D. (1739, ed. 2000). *Treatise of Human Nature.* Oxford: David Fate Norton & Mary J. Norton.

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: a survey. *Connection Science, 15,* 151-190.

Mataric, M. J., Meyer, J., & Wilson, S. (2009). Navigating with a rat brain: a neurobiology-inspired model for robot spacial representation. In *Proceedings of the 1st Conference on From Animals to Animats*, 169-175.

Meyer, J.A., Guillot, A., Khamassi, M., Pirim, P., & Berthoz, A. (2005). The Psikharpax project: towards building an artificial rat. *Robotics and Autonomous Systems, 50 (4)*, 211-223.

Nicolelis M. (2011). *Beyond boundaries: The new neuroscience of connecting brains with machines — And how it will change our lives*, New York: Times Books.

Pierce, D., & Kuipers, B. (1997). Map learning with uninterpreted sensor and effectors. *Artificial Intelligence, 92*, 169-229.

Sutton, R.S., & Barto, A.G. (1998). *Reinforcement learning: An introduction*, Cambridge, MA.

Understanding Sensemaking Using Functional Architectures

Robert Thomson, Christian Lebiere, Matthew Rutledge-Taylor, James Staszewski, and John R. Anderson
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15218
412-268-6284
thomsonr@andrew.cmu.edu, cl@andrew.cmu.edu, mattrt@andrew.cmu.edu, jjs@cmu.edu, ja+@cmu.edu

Keywords:
Sensemaking, Biases, Functional Modeling, Cognitive Architectures

ABSTRACT: *This paper outlines similarities between sensemaking theory and the ACT-R cognitive architecture. We analyze a functional model that interprets geospatial imagery data implemented in the ACT-R cognitive architecture. We also discuss how the various cognitive mechanisms of the functional model fit within sensemaking theory, and finally how an analysis of these mechanisms may give rise to cognitive biases.*

1. Introduction

When people make decisions, they must gather, elaborate, distill, and process (potentially) incomplete, incorrect, or contradictory information from the environment into actionable decisions. Sensemaking is a qualitative description of how information is gathered, structured, and used to generate and revise hypotheses (see Figure 1.1). Information flows through two interconnected processing loops: the foraging loop and the sensemaking loop (Pirolli & Card, 2005).

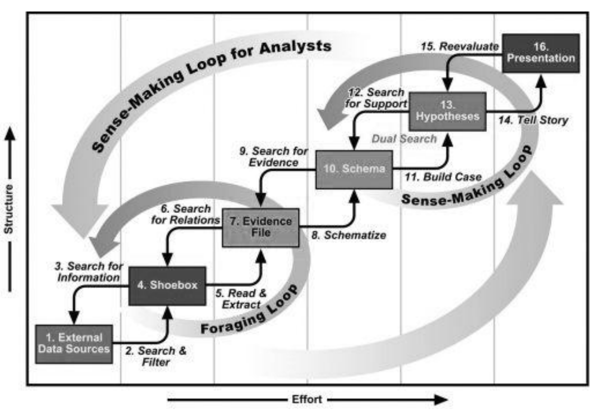


Figure 1.1. An overview of sensemaking. Reproduced from Interactive Automation. Retrieved January 31, 2012, from http://dydan.rutgers.edu/PDDALab/dev/images/flow.png.

Foraging describes how data is gathered, filtered, and aggregated into structured evidence. It is a form of (generally) bottom-up data collection. The stages in the foraging loop can be abstracted to a data gathering and implicit learning process that filters and assimilates information (Kahneman & Treisman, 1984).

The sensemaking loop describes how hypotheses are generated, evaluated, and either revised or discarded. Hypotheses drive top-down processes such as guiding attention to relevant information through the application and interpretation of schemas. Schemas are knowledge structures that both organize data and shape how this data is interpreted. For instance, the schema for a house fire is different whether you are the homeowner, the firefighter, or the arson investigator (Klein, Moon, & Hoffman, 2006). The concept of a frame, used by Klein, Phillips, Rall & Peluso (2006), is essentially equivalent to the sensemaking concept of a schema. In accordance with the fuller body of sensemaking literature, we prefer to use the term frame.

In the sensemaking loop, hypotheses are either revised or discarded when conflicting data enters the system (e.g., when evidence contrary to a hypothesis is encountered). This generally results from a misclassification of data within a frame. The hypothesis (and frame) needs to be revised to fit the new data, or a new hypothesis has to be adopted. Cognitive processes such as insight learning and analogical reasoning are generally incorporated into explanations of the sensemaking loop.

From a cognitive perspective, sensemaking can be broken into six main processes: learning a frame, recalling a frame, assessing the current frame, generating hypotheses, acquiring additional data, and reframing based on this evidence. We will focus on these six component sensemaking processes and how they can be mapped to mechanisms within the ACT-R cognitive architecture.

ACT-R 6 is a computational implementation of a unified theory of cognition. It accounts for information processing in the mind via task-invariant mechanisms constrained by the biological limitations of the brain. While sensemaking theory abstracts away from brain processes, it makes commitments to the control and flow of information that are commensurable with ACT-R's functional perspective. For example, the processing loops in sensemaking can be instantiated in the production rules controlling the follow of control and information in ACT-R. Furthermore, ACT-R is committed to localization of neural architecture, allowing for functional models to guide the development of neurally-inspired models.

To describe the mapping of sensemaking onto ACT-R, we will describe two models of very different tasks from the IARPA-funded ICArUS-MINDS project. Its goal is to create a neurally plausible model of sensemaking that accounts for cognitive biases in the context of intelligence analysis.

1.1 The ACT-R 6 Architecture

ACT-R is a functional cognitive architecture used to model diverse cognitive phenomena. The ACT-R architecture includes long-term declarative memory, procedural memory, and perceptual-motor modules connected through limited-capacity buffers. When a retrieval request is made to declarative memory (DM), the most active matching chunk is returned, where activation is computed as the sum of base-level activation, spreading activation, mismatch penalty and stochastic noise.

Spreading activation is a mechanism that propagates activation from the contents of buffers to declarative memory proportionally to their strength of association. Partial matching is a mechanism that allows for chunks in memory that do not perfectly match a retrieval request to be recalled if their activation overcomes a similarity-based mismatch penalty. Blending is a mechanism similar to partial matching that allows for a memory retrieval that results in a new chunk being created that reflects the consensus of all chunks in memory proportional to their activation instead of the retrieval of an existing chunk.

The flow of information is controlled in ACT-R by a production system, which operates on the contents of the buffers. Each production consists of if-then condition-action pairs. Conditions are typically criteria for buffer matches, while the actions are typically changes to the contents of buffers that might trigger operations in the associated modules. The production with the highest utility is selected to fire from among the eligible productions. Please see Anderson and Lebiere (1998) and Anderson et al. (2004) for a more complete account of the mechanisms implemented in the ACT-R architecture.

2. Sensemaking and ACT-R

ACT-R has previously been used to model several basic components of sensemaking such as evidence marshaling (Pirolli, Fu, Reeder & Card, 2002), and categorization (Anderson & Betz, 2001). The perceptual-motor modules and imaginal buffer in ACT-R are architectural analogues of the foraging processes in sensemaking theory. The visual buffer is used to store a low-level representation of visual stimuli in the environment. The imaginal buffer can be used for the creation of new chunks which are then stored into declarative memory.

In sensemaking, the foraging loop is a process of perceiving information from an external data source, placing it into a catch-all ‘shoebox’, and then organizing this ‘raw’ information into a series of structured evidence files. ACT-R’s visual module and imaginal buffer have functionality that can be used to mirror these information-gathering (i.e., foraging) steps in sensemaking theory. Specifically, the visual buffer ‘perceives’ spatial and object information (from the visicon; ACT-R’s representation of an external data source). This information is then harvested in a ‘raw’ form into declarative memory, which is analogous to the ‘shoebox’. Productions, however, can also aggregate this raw data and place it in the imaginal buffer, which then creates a new organized chunk in declarative memory for later retrieval (i.e., an evidence file). Based on the representational complexity of the task it may be necessary to aggregate perceptual information using the imaginal buffer. For instance, if the raw perceptual representation contains more information than is required for the task, a representation that focuses on the relevant features can be created. Doing so would enhance architectural mechanisms such as spreading activation and improve the efficiency of the learning process.

As previously discussed, the sensemaking loop contains six main cognitive processes: learning, recalling, and assessing a frame, generating hypotheses, acquiring additional data, and reframing. While these processes are represented as separate steps in the sensemaking loop, the cognitive processes subsuming their function are not necessarily distinct. In ACT-R, both frames and hypotheses can be represented as chunks. In general, the difference between frames and hypotheses is the kind of information stored in the chunks, which buffer holds the chunk (e.g., the goal buffer holding hypotheses, and the retrieval or imaginal buffers holding frames), and how productions manipulate the chunk structure.

Framing (learning, recalling and assessing a frame) involves recalling information that was encoded in the foraging loop and then applying an organizational perspective to it. In ACT-R, a frame can be represented by a chunk or multiple related chunks holding rule-like information for the organizing and interpreting of data-chunks (i.e., evidence files) into testable hypotheses. Based on task complexity, it may not be necessary to represent a frame as a series of related (generally hierarchically-organized) chunks if the expected output can be captured in a simple rule-like structure.

In many sensemaking tasks, hypotheses take the form of either an estimate of a forced-choice response or the generation of likelihoods of the presence or absence of a given state of the environment. In ACT-R, a hypothesis can be represented as a chunk that contains the representation of a potential response. An initial hypothesis allows for the model to test against either an actual or theoretical outcome and guides the gathering of additional evidence that leads to reframing.

Gathering more data and reframing occur through feedback on the accuracy of hypotheses and by intuiting regularities in new data gathered by the foraging loop. Top-down feedback occurs by comparing the current hypothesis against a normative (i.e., externally-driven) solution, and then either revising or discarding the current hypothesis and/or reframing the data. This reframing can occur by modifying the current frame. For instance, the model could change values (e.g., weights) associated with a given rule-like representation. An example would be increasing the likelihood of a given outcome based on the presence of a given feature. Reframing can also occur through utility learning, by reinforcing certain productions firing over others. For instance, penalizing productions that yield errors and reinforcing productions that test features which are diagnostic to the task. Reframing can apply to changing the hypothesis for the current data set as well as producing better hypotheses for future data sets.

While early evidence initially shapes the adoption of a frame, this frame can then shape how future evidence is recalled through the base-level and spreading activation mechanisms in the ACT-R architecture. In base-level activation, chunks that have been recalled in the past (which also spreads to related chunks through spreading activation) have higher activation, which make the recalling of similar data-chunks more likely in the future.

Before getting into more specifics regarding the functional and architectural analogues between the ACT-R cognitive architecture and sensemaking theory, we describe two tasks that instantiate the process.

3. The Tasks

The following tasks are designed to study the role of cognitive biases in sensemaking in the context of intelligence analysis. A facility identification task examined the ability of human participants to learn to analyze simulated geospatial images and correctly discriminate facilities in unlabeled images. Six group identification tasks tested the ability of human participants to correctly identify which group was responsible for an attack based on evidence from layers of data in geospatial images, and the application of probabilistic rules associated with the data interpretations.

The facility identification task was compared to both Bayesian normative solutions and human performance. The models of the group identification tasks were developed prior to gathering human data and serve as predictions of the kinds of biases humans may exhibit.

3.1 The Facility Identification Task

Participants were trained to identify four kinds of facilities in simulated geospatial images. Each image depicts a single facility (e.g., factory complex) composed of a set of discrete features (e.g., buildings). The three categories of features were: IMINT (image intelligence), representing buildings and other terrain such as roads and rivers; MASINT (measurement and signature intelligence), representing signals of radiation or chemical concentrations; and SIGINT (signals intelligence), representing communication transmissions.

The statistical breakdown of features was not even: there were nine unique IMINT features, seven that represented buildings, and two that represented water features. In contrast, there were only two kinds of MASINT features, while the SIGINT features were entirely homogeneous. In addition, each IMINT could appear at most one time in each image, whereas multiple instances of SIGINT and each MASINT could occur in each image. Additionally, each building (IMINT) could have attached to it zero or one piece of rooftop hardware. Each of the four facilities had different base rates for the occurrence of each of the possible features.

The experiment was divided into two phases: a training phase and a testing phase. In the training phase participants were presented with 48 annotated examples of each facility (192 total examples), 16 at a time (in a four-by-four grid). In the testing phase the participants were presented with single unlabeled images sequentially. For each image, participants were required to report a probability distribution over the four possible facilities indicating the likelihood that the image contained each of the facilities. For more details on the facility identification task and comparisons with human data, see Rutledge-Taylor et al., (2011; forthcoming).

3.2 The Group Identification Tasks

The group identification tasks were a series of six tasks in which the participants' were to predict which of four groups was responsible for an attack. Each task was presented spatially in a 100 x 100 grid (representing 30 square miles) on a computer screen. The critical feature in Tasks 1 to 3 was signals of activity (SIGACTs), which represented previous attacks by the groups. The pattern of SIGACTs for each group was defined by a group center of activity and a dispersion value. SIGACTs were produced probabilistically according to these definitions.

Task 1 consisted of 10 blocks of 10 trials. A trial consisted of a single SIGACT, represented as a group letter, appearing on the display. On the 10^{th} trial of each block the group responsible was hidden, with the SIGACT represented as an empty square. Participants were required to assess the probabilities that each of the two groups was responsible for the attack, and then were asked to produce a forced choice response.

Task 2 consisted of 10 blocks of 20 trials, similar to those in task 1. The difference was that there were four groups instead of two. In addition, participants were not required to produce a forced choice response after giving their probability estimates. They were instead required to draw a circle for each group that defined the two-to-one ratio of the likelihood of an attack by the given group occurring inside versus outside their circle (e.g., their sphere of influence).

Tasks 3 to 6 (see Figure 2.1) added the complexity of calculating distance along road networks. In task 3, participants were still required to find group centers from a series of SIGACTs, but distance between SIGACTs was now to be judged "as the cow walks" along a road network. As such, tasks 3 to 6 involved visual problem solving (e.g., path planning and curve tracing; Lefevre, Dell'Acqua, Roelfsema, & Jolicoeur, 2011). Task 3 consisted of 10 blocks of 20 trials. Participants were required to produce a probability distribution for each group's likelihood of being responsible for the SIGACT on the last trial of each block.

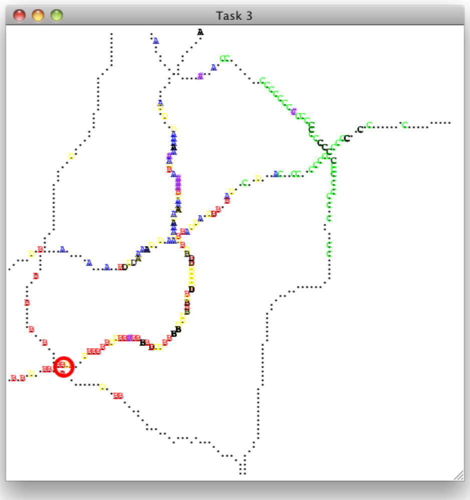


Figure 2.1. A sample screenshot of the group identification Task 3 model. The string of dots represents the road network, the letters represent individual SIGACTs, and the circle represents the model's focus of attention.

In tasks 4, 5 and 6, intelligence data was presented in layers: HUMINT gave the center of activity for each group (the participants were not required to judge this from individual SIGACTs in tasks 4 to 6); IMINT showed which of the roads in the network were the major roads and which were the minor roads; MOVINT showed which roads had dense traffic versus sparse traffic; SIGINT revealed information about whether a group was producing chatter or not; and SOCINT showed the territorial boundaries for the groups on the map.

Tasks 5 and 6 were very similar. In both tasks a road network with an anonymous SIGACT and the four group locations are presented. The participants' task is to update the probability distribution over the four possible groups responsible for the SIGACT after each layer is revealed. The basis for adjusting the probabilities is a set of rules that are provided to the participant. Each rule specifies changes in the relative likelihoods that the groups are responsible based on a piece of layer data. For example, if the SIGACT occurs on a major road, groups A and C are four times as likely to be responsible. In both tasks the HUMINT layer is provided first, and so the initial probability distribution is based on the relative distances between the group centers and the SIGACT. In task 5, the remaining layer are revealed, on at a time, in a random order. In task 6, the participant chooses, one at a time, three of the remaining four layers.

ACT-R models were produced for tasks 1, 2, 3, 5 and 6, prior to any human data being made available. Task 4 was not modeled as it was essentially a subset of task 5 and will be omitted from further discussion. As the models were generated prior to the collection of human data, they are predictive of human performance and provide an opportunity to examine and predict the influence of cognitive biases and provide possible solutions to reduce their impact on human judgments.

As of this submission, human data is still unavailable for the group identification tasks. The ACT-R model currently generates output probabilities for Tasks 2, 3, and 5 such that the highest group (of 4) is given a M = 49.8% probability (with the three other groups equally distributed, M = 16.7%). This is approximately 30% lower than a fully-rational Bayesian model (M = 81.2%) and reflects uncertainty due to conservatism and anchoring effects in the generation of group centers and probability judgments, and stochastic elements within the distance perception and path planning functions.

4. Sensemaking in the Identification Tasks

In the facility and group identification tasks, ACT-R modeled sensemaking processes at different levels of abstraction due to the increased task complexity in the group identification tasks. For the facility identification task, an ACT-R model learned which facility features were the most diagnostic to be attended to in order to correctly classify the facility. The model oscillates between the foraging and sensemaking loops as it acquires evidence, changes frames, and updates its facility identity hypothesis.

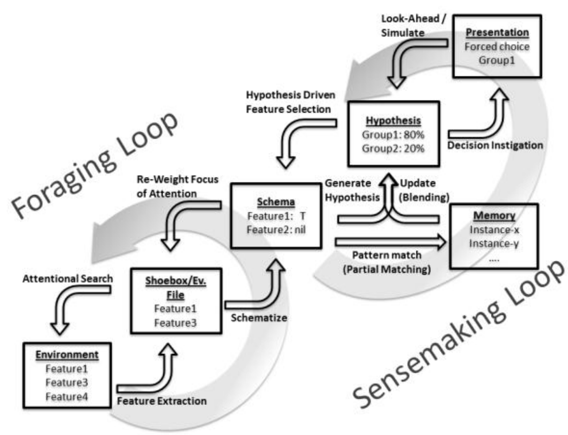


Figure 4.1. The ACT-R analogue to sensemaking.

As described in Figure 4.1, the foraging loop is the sequence of productions (represented as the arrows) which select new features from the environment (e.g., sensorimotor modules) and organize them in the imaginal buffer. In ACT-R, there is little distinction between the shoebox and evidence file, as similar productions determine which features are attended and harvested into declarative memory. The shoebox holds an initial selection of features extracted from the sensorimotor modules, which are then harvested into declarative memory via the imaginal module. Evidence files include a similar feature extraction from either the shoebox (memory) or environment (sensorimotor modules), but also have productions which may re-encode features from the shoebox and extract higher-order feature relations based on the currently-held frame/schema.

The sensemaking loop is the sequence of productions that retrieves a schema (e.g., a facility frame chunk) and generates a hypothesis for what facility is present (or group responsible) in a given task. In the facility identification task, the model oscillates between the two loops, seeking out more evidence and updating its hypothesis until a threshold for the expected utility (e.g., information gain) of making a decision is surpassed.

The facility identification task is an example of a case when a single chunk is sufficient for storing all relevant evidence in a frame. The frame was a single-chunk representation of the set of features to be attended to. Knowledge of the probabilities of the various features being present in a given instance was implicit in the set of chunks and their activations. The facility identification task is thus best described as a category-learning task as each frame can be interpreted as a category exemplar.

In the group identification tasks, the model encodes visual information in the foraging loop hierarchically. For instance, a chunk representing a road segment included a slot for road identification and two slots that each hold intersection-chunks, which in turn had slots for co-ordinate pair chunks containing x and y coordinates as slots. In addition, in tasks 5 and 6 there were several rule-like chunks (each corresponding to a layer) that specified how new evidence impacted the probability of a given hypothesis. Thus the current frame held by the model was represented by the current INT layer in the retrieval buffer, which would be used to update the current hypothesis (i.e., probability distribution).

4.1 Sensemaking in the Facility Identification Task

The ACT-R model of the facility identification tasks implements an adaptive foraging loop. The feature selection process is composed of two distinct phases. In the training phase the model studies a set of images for the purpose of learning which features are associated with each facility. In the learning phase the model must identify the facilities in images, and with feedback learn the optimal utilities for the various feature selector and decision instigation productions.

Learning frames was accomplished during the task's training phase. ACT-R modeled variants instantiating both rule-based and exemplar-based category learning. In this phase the model acquires examples of facilities and commits them to memory. In this case a frame is an abstraction of the accumulated exemplars that is realized during category assignment. The frame for a particular facility is thus the implicit knowledge that the model possesses about the probabilities of the various features being present in an instance. This is functionally similar to a rule with the implicit probabilities for each feature representing a set of conjunctions updating the likelihood of a given facility based on the presence or absence of a given feature.[1] The currently-held hypothesis is the probability that the current image depicts a given facility (and is provided to the model in the training phase).

Due to the feedback received during the training phase, the ACT-R model is constantly reframing by determining the utility of which features are the most diagnostic of the facilities. Feature selection is the process of deciding which features present in an image ought to be attended to, and which should not. Part of sensemaking theory is the ability to aggregate and distill information to maximize the availability of information within the context of working memory limitations. Feature selection, in part, addresses the issue of the working memory capacity for information.

It is presumed that the participants are unlikely to be able to attend to every available feature in every image due to various cognitive constraints. The normative probability that a feature should be selected is based on its utility in facility identification. The ACT-R model uses utility learning to develop implicit preferences for attending to some features over others. Utility learning provides rewards (or penalties) to productions based on their outcome. A positive reward is instigated after a facility is correctly identified, while a negative reward is instigated after an incorrect identification.

The productions of interest during utility learning are divided into two categories: feature selector productions and decision instigation productions. Each feature selector production is specific to a single IMINT feature and a specific intermediate hypothesis about what facility is represented in the given sector. This allows for the utilities of selecting features to be hypothesis-specific and represents the interplay between frames and hypotheses in the sensemaking loop.

Each decision instigation production is eligible to fire after a specific number of features have been selected. Once a decision instigation production has fired, a facility identification event occurs. If the identification is correct, the decision instigation production and all the feature selection productions that lead to the decision are rewarded. If the identification is incorrect, the same productions are penalized.

In the learning phase, the model alternates between updating the model's current hypothesis of what facility is present and selecting a new feature (or electing to stop encoding features). When selecting a feature, all of the

[1] In another paper (Rutledge-Taylor, Lebiere, Thomson, Staszewski & Anderson, forthcoming) we discuss similarities in performance between rule and exemplar-based models on this task.

feature selector productions that are eligible to fire compete. The production with the highest utility will fire, and the feature associated with the production will be added to the selected features stored in the imaginal buffer. This is analogous to acquiring additional data.

The decision to stop encoding features is governed by a set of decision instigation productions that compete against the feature selection productions. The decision instigation productions receive utility rewards and punishments, as do the feature selection productions (similar to the training phase). The model stops encoding features when the relevant decision instigation production fires instead of any of the eligible feature selection productions. This evolving production competition allows the model to learn the rational number of features to encode and represents the model's reframing based on whether the hypothesis was supported or contradicted.

When updating the current hypothesis (stored in the goal buffer), the facility is identified by recalling the facility frame from declarative memory that best matches the features selected so far. The value for the facility ID in the recalled frame is used to update the hypothesis maintained in the goal buffer. When the model stops encoding new features, the current hypothesis is output as the model's final categorization decision for the image. It is possible that, based on new features (reframing), the hypothesis may be updated (new probabilities) or rejected (by choosing a new facility type).

In summary, the process of feature selection in the facility identification task mirrors the sensemaking process. The foraging loop is analogous to the sequence of productions that selects new features from the available pool and organizes them in the imaginal buffer. The sensemaking loop is analogous to the sequence of productions that retrieves a facility frame chunk from DM and generates a hypothesis for what facility is present in the given image. The model oscillates between the two loops, seeking out more evidence and updating its hypothesis until the expected utility of making a decision is greater than that of collecting more data.

4.2 Sensemaking in the Group Identification Tasks

Unlike the facility identification task, the group identification tasks do not have a training phase and the rules for adjusting probabilities are explicitly provided to participants (and the model). The model thus has less opportunity for learning due to feedback (based on revising hypotheses). Instead, the group identification tasks require more general spatial judgments (such as path planning) and the application of multiple rules that do not fit the traditional definition of frame as a singular structure or representation in the sensemaking literature.

In the facility identification task it was also practical (and fit within the spirit of working memory limitations) to represent the relatively small set of features in a single chunk. In the group identification tasks it was neither practical nor cognitively plausible to represent the full set of spatial information within a single chunk in DM. Instead, chunks of spatial information are represented hierarchically. For instance, a group center (in task 3) is located on a road segment, which is made up of a location (co-ordinate pair) along a road segment. Road segments are defined by their endpoints (intersections) and general length and shape, which are also linked to locations.

The basic unit of evidence in tasks 1 to 3 is a SIGACT, which corresponds most closely with evidence files in the foraging loop. SIGACTs are perceived by the visual module, their location and group identity placed in the imaginal buffer, and at the end of each block, an estimate of each group center (and two-to-one boundary in task 2) is performed. This group center and boundary estimate is a kind of spatial frame (insofar as it predominantly contains spatial information). In the model, the group center is calculated for each group in a separate chunk, thus the current frame of the model incorporates four chunks (one for each group center). Using these spatial frames, a hypothesis (i.e., the set of probabilities) is generated and compared against the provided feedback. The model has only a limited ability to reframe because it only receives feedback (i.e., ground truth, not probability distribution) at the end of each block. As such, reframing occurs when the model updates the group center estimate (in the subsequent block) with the identity and location of the target SIGACT from the previous block.

In tasks 3 to 6, distance (for the purpose of generating group centers from SIGACTs and between each group center and a target SIGACT) was not calculated using a 'crow-flies' estimate, but instead by estimating the length of the path along the road network. Due to the complexity of the road network, more than one path could be chosen. As such, the path-planning processes within the model could be seen as their own self-contained sensemaking process (implemented as a non-deterministic hill-climber). Foraging involved the perception of the possible paths at each intersection, framing involved the storage of the path, hypothesis testing involved mentally traversing a candidate road segment, and reframing occurred when the model needed to backtrack to a previous segment (due to hitting a dead-end or detecting that it had gone in a loop). The hypothesis included the currently-held distance estimate, and was revised when a new candidate road intersection was added to the path.

Tasks 5 and 6 use a more general model of hypothesis testing and reframing due to application of multiple layers of INTs. The spatial frames (from task 3) mapping group centers to a target SIGACT now represent a single layer (the HUMINT layer). There is little foraging to be done because the rules and group centers are provided as input to the model. Sensemaking is even more prevalent in task 6 because participants are able to choose their next INT layer (in task 5, three layers are provided in a random order). The model uses utility learning (similar to facility identification) to reward the model when it chooses a layer that leads to a correct probability distribution.

A frame is generated when the first layer of information (HUMINT) is applied to existing group centers. Using this frame, an initial hypothesis is generated (i.e., the initial probabilities for each group). Reframing occurs when an additional layer (SOCINT, SIGINT, MOVINT, or IMINT) is applied, which then revises the current hypothesis. Acquiring additional information occurs externally to the participant (and model) in task 5 because the layers are presented randomly; however the participant (and model) may reframe based on this additional evidence. In task 6, however, the model reframes by choosing a layer based on the current hypothesis. For instance, the model might select SOCINT when the probability distribution is flat, MOVINT or IMINT when trying to dissociate two alternatives, and SIGINT when the distribution is steep. The current hypothesis is revised when the next INT layer is applied (i.e., when the rule is applied which in turn revises the probabilities), and the layers that lead to correct classifications are rewarded.

In summary, the closest equivalent to a single frame in the group identification tasks is an organized representation of the input (i.e., the group centers in Tasks 1 to 3; the HUMINT, SOCINT, SIGINT, MOVINT and IMINT layers in Tasks 4 to 6). This definition of a frame preserves the meaning of the sensemaking processes described in the facility identification task.

More specifically, learning a frame corresponds to the accumulation of evidence supporting the hypothesis of a group being responsible for a SIGACT. This involves the accumulation of group centers estimates (in tasks 1 to 3), the dispersion of attacks (task 2), and generating initial probabilities based on rules (Tasks 4 to 6). Importantly, the model assumes that some spatial and mathematical mapping chunks already exist in memory, and reflect general experience/competencies.

Generating a hypothesis corresponds to the initial probability distribution assigned to each group. The probabilities are an evaluation of how probable it is that a particular group was responsible for a target SIGACT, based on how well the features of the SIGACT (e.g., location) match the characteristics of each group's frame. Reframing occurs whenever feedback occurs and when a new INT layer is applied.

4.3 Modeling Biases in Sensemaking

In sensemaking, biases are usually identified as heuristics in data gathering within the foraging loop and in frame (re-)encoding in the sensemaking loop (e.g., availability heuristic; Klein, Moon, & Hoffman, 2006). Under this interpretation, biases result mainly from architectural constraints (e.g., working memory, attention). Biases, however, may also occur in the sensemaking process due to the nature of the task and how the limitations of human memory and attention degrade performance in systematic ways. When modeling human performance, how the computational model is constructed will influence how biases are reflected in the model. For instance, the bias may be an emergent property of the architecture, or it may be due to the specific strategies of the model itself.

Thus, in cognitive modeling biases may arise from a combination of task demands, architectural limitations (e.g., one item per buffer in ACT-R), and modeler preferences. For instance, confirmation bias (an overly certain belief in the leading hypothesis) may be due to an explicit strategy coded by the model to always focus on the group with the highest probability (e.g., choose the SIGINT rule for that group) or it may be implicit in the architecture of the model (e.g., spreading activation from contextual cues increasing the probability of some features being recalled). This was a factor in the feature selection model of the facility identification task. By maintaining a chunk representing the current facility hypothesis in a slot of the goal buffer, the model was biased towards maintaining that hypothesis because exemplars in DM of the same facility would receive a boost in activation via spreading activation and thus make a greater contribution to updating the current hypothesis. It may also be possible to equate model-level biases with biases resulting from explicit human strategies (Gigerenzer & Gaissmaier, 2011).

Base-rate neglect is another example of a bias which can be a result from either the task, implicit (i.e., architectural), or strategic levels. At the task level, higher levels of complexity can cause base-rate neglect due to the sheer volume of stimuli to store and subsequently recall. To reduce both memory load and processing time, some features need to be abstracted. Similarly, at the implicit level, working memory is generally seen to have a capacity of 7 +/- 2 chunks of information available (Miller, 1956). Once this memory capacity is exceeded, some information needs to be either abstracted or discarded. Finally, at the strategic level, there are coding choices that may be made based on an analysis of task difficulty and an awareness of architectural constraints. For instance, when grouping many dots presented sequentially on a display it may be an explicit strategy to ignore base-rate and focus on grouping elements together.

The goal of cognitive modeling should primarily be to account for biases due to the interplay between the task and architectural levels. Those biases should be considered emergent properties of the model. Explicit strategies would be represented in the model as design choices in terms of the specific productions and chunk types utilized. Explicit reasoning strategies may influence performance (and often reflect architectural limitations), but should not be relied upon as the mechanism to instantiate biases (especially biases considered to be due to automatic processes).

5. Discussion

The advantage of standardizing sensemaking processes with ACT-R is that it provides a common framework for model comparison in extant sensemaking tasks and possible model re-use between tasks. A limitation of cognitive modeling is that models can rarely be generalized or re-used to adapt to a different task. By standardizing the elements of sensemaking with ACT-R, we are providing a roadmap to generate a general cognitive model of sensemaking, capable of making predictions regarding human performance (as opposed to fitting to extant human data). A generalized model of sensemaking is under development, which is based on the principles described in this document.

In summary, the present paper has provided a possible framework and some suggestions for how to encode stimuli and represent hypotheses in ACT-R that would instantiate sensemaking processes. While it was beyond the purview of this discussion to provide a more in-depth link between the individual sensemaking elements and the ACT-R architecture, it does provide a framework for a general cognitive model of sensemaking.

6. Acknowledgement

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of the Interior (DOI) contract number D10PC20021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained hereon 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.

7. References

Anderson, J. R., & Betz, J. (2001). A hybrid model of categorization. *Psychonomic Bulletin & Review, 8*, 629-647.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. (2004) An integrated theory of Mind. *Psychological Review, 111*, 1036-1060.

Anderson, J. R., and Lebiere, C. (1998), *The atomic components of thought*, Erlbaum, Mahwah, NJ, 1998.

Gigerenzer, G., & Gaissmaier, W. (2011). Heuristic decision making. *Annual Review of Psychology,* 62, 451–482.

Kahneman, D., & Treisman, A. (1984). Changing views of attention and automaticity. In R. Parasuraman, D.R. Davies & J. Beatty (Eds.), *Variants of attention* (pp. 29-61). New York: Academic Press.

Klein, G., Phillips, J. K., Rall, E., & Peluso, D. A. (2006). A data/frame theory of sensemaking. In R. R. Hoffman (Ed.), *Expertise out of context: Proceedings of the 6th International Conference on Naturalistic Decision Making*. Mahwah, NJ: Lawrence Erlbaum & Associates.

Klein, G., Moon, B., & Hoffman, R. (2006). Making Sense of Sensemaking 1: Alternative Perspectives. *Intelligent Systems*, 21 (4), 71.

Lefebvre, C., Dell'Acqua, R., Roelfsema, P., & Jolicœur, P. (2011). Surfing the attentional waves during visual curve tracing: Evidence from the sustained posterior negativity. *Psychophysiology*, 48, 1-7.

Marr, D. (1982), *Vision: A Computational Approach*, San Francisco, Freeman.

Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, *63*, 81-97.

Pirolli, P. & Card, S. K. (2005). The sensemaking process and leverage points for analyst technology. In the *Proceedings of the 2005 International Conference on Intelligence Analysis*. McLean, VA: Office of the Assistant Director of Central Intelligence for Analysis and Production.

Pirolli, P., Fu, W.T., Reeder, R. and Card, S. K. (2002). A User-Tracing Architecture for Modeling Interaction with the World Wide Web. *Advanced Visual Interfaces*, Trento, Italy.

Rutledge-Taylor, M. F, Lebiere, C., Vinokurov, Y., Stazewski, J., & Anderson, J. R. (2011). Bridging the gap: A neurally plausible functional model of sensemaking. In Samsonovich, A. V., Johannsdottir, K. R., Chella, A., & Goertzel, B. (Eds.). *Proceedings of the Second Annual Meeting of the BICA Society.* Fairfax: IOC Press.

Rutledge-Taylor, M. F, Lebiere, C., Thomson, R., Stazewski, J., & Anderson, J. R. (forthcominng). A Comparison of Rule-Based versus Exemplar-Based Categorization Using the ACT-R Architecture. *The 21st Behavior Representation in Modeling & Simulation Conference.*

Author Biographies

ROBERT THOMSON is a Post-Doctoral Researcher at Carnegie Mellon University in Pittsburgh, PA. His main research interests are distance perception and spatial organization, serial memory, and biases in reasoning.

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, cognitive robotics and neuromorphic engineering.

MATTHEW RUTLEDGE-TAYLOR is a Post-Doctoral Researcher at Carnegie Mellon University in Pittsburgh, PA.

JAMES STASZEWSKI is a Professor of Psychology at Carnegie Mellon University in Pittsburgh, PA.

JOHN R. ANDERSON is a Professor of Psychology at Carnegie Mellon University in Pittsburgh, PA.

Towards Adding a Physiological Substrate to ACT-R

Christopher L. Dancy
Frank E. Ritter
The College of Information Sciences and Technology, The Pennsylvania State University
University Park, PA 16802
cld5070@psu.edu
frank.ritter@psu.edu

Keith Berry
The Bioengineering Department, the Pennsylvania State University and Medical Corps, US Army Reserves
Hershey, PA 17033
keith.andrew.berry@us.army.mil

Keywords: ACT-R, HumMod, Physiology, Moderators, Stress

ABSTRACT: *Connecting a physiological model to a cognitive architecture presents an attractive option to better simulate a wide range of human behavior. This connection should facilitate both the effects of physiology on cognition (e.g. hunger and decision-making), and the effects of cognition on physiology (e.g. autonomic responses to memory featuring particularly aversive stimuli). To add physiology to a cognitive architecture, it should be represented as a separate module or substrate. We present ACT-RΦ (ACT-R Phi), a connection of the physiology simulation system HumMod (Hester et al., 2011) and the cognitive architecture ACT-R (Anderson, 2007) using an newly created ACT-R module. A model of the startle response and its consequent effects on cognition and physiology is presented to demonstrate an example use of the new substrate. This extended version of ACT-R allows a user to computationally realize theories involving cognition, physiology, and their interaction. This architecture has potential applications to training simulations.*

1 Introduction

The connection of a physiological model to a cognitive architecture provides an opportunity to simulate a wide range of human behavior, cognitive, and physiological. Previously, architectures have been developed to implement different forms of moderators of cognition but they often lack a unified approach.

Based on reviewing several previous systems we note some possible connections and design suggestions for connecting a model of human physiology and a cognitive architecture (represented as ACT-R). We then present ACT-RΦ, a modification of ACT-R 6 (Anderson, 2007) that connects it a model of human physiology, the HumMod simulation system (Hester et al., 2011), to ACT-R. HumMod is a simulation system that provides a top-down representation of human physiology. ACT-RΦ allows the user to enact changes to the underlying physiological substrate via buffers (within a new "physio" module) accessible by productions. Change in physiological variables can consequently drive change of cognitive parameters if a connection has been established. We also provide an example use of the system with an example model derived from a previous ACT-R model.

2 Past Implementations of Cognitive Moderators

Here we provide an abbreviated overview of several previous implementations of cognitive moderators in architectures. They provide lessons for creating an architecture with moderators, or modifying existing architectures to accommodate moderation.

2.1 CoJACK

CoJACK is an adaptation of the Beliefs, Desires and Intentions (BDI) architecture Java Agent Construction Kit (JACK). JACK consists of events, plans, belief-sets, and intentions. CoJACK extends JACK by adding moderators and errors(Evertsz et al., 2009);it adds the ability to include noise (modeled after the noise representation in ACT-R) in the belief-set choice process. This introduces the possibility of retrieval of beliefs which are incorrect or only partially match to the conditions presented. The moderators represented in the modified architecture are fear and caffeine. These moderators can be dynamically set, thus they can be modified during a simulation. They are represented as overlays to the existing architecture, this is similar to previous work on defining theories of stress using the idea of overlaying the existing architecture by Ritter et al. (2007).

Work with CoJACK shows that computational physiology is possible and would lead to interesting changes in behavior. It explores some mechanisms involved in physiology moderating cognition (and vice versa) but the work required implementing a full model of physiology to continue. Additionally, combining multiple overlays to represent multiple effects ultimately will become intractable and lead to creating a representation of physiology to resolve the conflicts in how overlay effects combine and affect one another. Models of physiology that provide systematic representations of gross anatomy potentially provide a modeler with an opportunity to combine and represent multiple overlays within one tractable system.

2.2 MicroPSI

MicroPSI (Bach, 2009) is a hybrid architecture with both symbolic and subsymbolic (neural network) representations based on the Principle of Synthetic Intelligence (PSI) theory. The architecture has urges, motives (an urge with a corresponding goal), and demands underlying the motivation schema. MicroPSI has three physiological based motives, two cognitive based motives, and one social motive. The physiological motives (fuel, water, and “intactness”) provide higher-level representation of homeostatic-based (fuel and water) and somatosensory-based (intactness) motivations.

Bach (2009) admits the mechanisms in MicroPSI fail to represent many of the complexities of human cognition; however it does provides mechanistic representations important for realistic agent autonomy. MicroPSI provides an example of a middle ground between an agent representing human-like cognitive abilities and underlying modulating mechanisms that interact. This architecture displays some of the important aspects of having an underlying motivational system to modulate behavior and cognition. Providing a systematic representation of human physiology to connect to architectures that represent higher-level concepts of the human mind allows one to traverse levels of representation to provide the facilities for users to model the moderating effects of physiology at an appropriate level.

2.3 Gunzelmann model

Gunzelmann and colleagues developed a model that simulates the effects of fatigue (arising from sleep-deprivation) and circadian rhythms by altering ACT-R module parameters (Gross et al., 2006; Gunzelmann et al., 2009). The approach they use is an alteration of the utility of production rules based on alertness or arousal. The authors simulate fatigue and alertness using mathematical models of alertness, the output from these models is used to drive ACT-R module parameters, predominantly those in the procedural system, i.e. those related to utility.

Their model of fatigue illustrates the opportunities afforded by tying a mathematical model representation of a physiological component to a cognitive architecture. With this model they have been able to model a deterioration in driving performance based on fatigue (Gunzelmann et al., 2011). While they were able to simulate some of the effects of fatigue, others like a spike in alertness due to a startle/defensive response proved elusive. A more full representation of physiology at macroscopic levels would potentially allow one to simulate these responses as well as fatigue based on other aspects of physiology (e.g. energy need or thirst).

3 HumMod

HumMod (Hester et al., 2011) is a simulation system that simulates human physiology via a model specified in XML schema. The HumMod system is an extension of physiology research of Dr. Guyton who applied engineering systems analysis to the cardiovascular system under normal and pathologically significant physiologic states. His work continues to serve as the basis of contemporary medical knowledge regarding cardiovascular pathophysiology (Guyton et al., 1972; Montani & Van Vliet, 2009). The current HumMod model is a derivative of the original Guyton model (Guyton et al., 1972) that represents an instance of human cardiovascular physiology using a top-down schema that is defined through hundreds of linear and non-linear state equations over 5000 state variables. HumMod has provisions for simulating normal and abnormal physiology in multiple time scales. Additionally, the model provides several points of access to the nervous system through both the endocrine and nervous systems.

There are two ways to change the values attached to variables in HumMod, changing the underlying XML-based model or changing the set values after the model has been loaded into the simulator system. An alteration of the base-model allows the changing of initial variables, derivations, and connections between variables. Changing the set-values has perhaps less systematic power than a change to the actual model, but allows one to work within the given model and quickly view the data arising from these changes.

Tracking changes in data can become cumbersome in the HumMod system. If one chooses to use the full-system with the built-in user-interface, a user will need time to explore the interface to determine what variables should hold their attention. One can also communicate directly with the model solver (the portion that actually digests the XML model) via message passing through commands written to files. If one chooses to simply use the model solver portion of the system, processing the output data becomes difficult as portions of the 5000+ variables are predominantly used to implicitly affect other variables.

How to best represent these variables and corresponding values remains an open problem.

4 Initial Design for Adding Physiology to a Cognitive Architecture

We note a preliminary design here to tie ACT-R to HumMod, and then demonstrate part of this design. There are several ways one can represent the connections between physiology and cognition depending on the contextual factors in which one is interested. As an example, we discuss appetitive mechanisms and the hypothalamic-pituitary-adrenocortical (HPA) axis. Homeostatic-Appetitive motivation can cause a modification of behavior and cognition to reduce the particular motivation. The HPA axis plays a major role in reactions (cognitive and physiological) to stressful stimuli (Tsigos & Chrousos, 2002).

4.1 Appetitive motivation

Basic bodily motivations deserve important attention when discussing the physiological moderation of cognition. Homeostatic-Appetitive motivations can begin to take priority over attention resources and affect human behavior and cognition depending on the physiological imbalance being sensed by a person. Innate necessities like hunger and thirst homeostasis ultimately can change memory recall and attention (e.g., (Aarts et al., 2001; Mogg et al., 1998)). The need to void and its effect on cognition has also been previously studied including how it affects both working memory (Lewis et al., 2011) and decision-making (Tuk et al., 2011)

In ACT-R one can begin to implement appetitive motivation mechanisms by tying rule and memory parameters to the current states of receptors represented by HumMod. Hunger can be represented with the food portion of the “GILumen” variable. HumMod also provides a representation of the effect of osmoreceptors, which measure changes in extracellular body fluid and are known to be responsible for thirst (McMorris, 2009). These variables can be tied to procedural utility (e.g., :iu and :nu) and declarative memory (e.g., :ans, :bll, and :mp) to have results of physiology control the probability of rule selection and memory is retrieved during activation of these motivations. This change should enact attentional biases. With these motivations and change in utility one also has a mechanism to represent a type of “pleasure” based on whether or not (and to what extent) an action satisfies a motivation. Procedural partial matching also presents an opportunity to represent effects of hunger or thirst. Mogg et al. (1998) determined that there is a selection attention bias related to a hungry state, i.e. participants were found to have a bias in selection of words related to food while in a hungry state.

4.2 The HPA axis

The relation between stress, endocrine responses, and cognition is a good connection to explore for the modeling of effects of physiology on cognition (or vice versa). Stress affects the hypothalamic-pituitary-adrenocortical (HPA) axis and this axis dictates the release of hormones that can subsequently affect cognitive processing. Stimulation of this axis can cause a systematic release of cortisol from the adrenal cortex as well as a release of epinephrine and norepinephrine from the adrenal medulla. The systematic release of cortisol is known to have latent effects on cognition, however it can also have a more rapid effect on stress-related information consolidation and appraisal of new situations (Groeneweg et al., 2011). Cortisol also affects the HPA axis itself by reducing future activation leading to further release of glucocorticoids.

One can use previous studies of stress representation in ACT-R to guide parameter modulation via a more complex physiological system. Ritter et al. (2009) chose to change memory retrieval and vocal module parameters (activation noise, base level constant, and seconds-per-syllable) to better match human experimental data in a serial subtraction task being completed in conjunction with the Trier Social Stress Test (TSST) (Kirschbaum et al., 1993). Additional potential parameters exist in the rule utility facilities represented in ACT-R, e.g. initial utility of a user-defined function (:iu) or initial utility for a learned function (:nu). Upon completion of a stressful task, model production utility can be assessed not only based on reward for task-completion, but also for a return to a homeostatic state. Thus, for this reward we could monitor the delta between physiological variables in HumMod related to a stress response, e.g. adrenal hormones, heart rate, and blood pressure. An increased attention on the contextual stressor could potentially dictate a change in the rule retrieval values for rules relating to shifting attention away from the context. The initial declarative memory activation values during the stressor may also change depending on the input module that contains information on the stressful stimuli, i.e. if the model’s stress is triggered by predominantly visual stimuli then assign a higher initial activation value to that declarative memory when the memory is harvested.

4.3 Summary of discussed modulating effects of physiology

Homeostatic-based appetitive motivation can potentially affect procedural rule choice and memory encoding/retrieval. As motivations are failed to be satisfied, effect on cognition intensifies further causing more extreme changes to cognition and behavior. Stress causes activation in the hypothalamus, eventually triggering sympathetic system activation through mechanisms like cortisol release, which acts as a moderator on activation of neuronal activity important to cognitive function. With innervations of the adrenal medulla by efferent fibers, hormonal release of epinephrine and norepinephrine is initiated as a mechanism for short-term adaptation to the stress; this also has an effect on other aspects of human physiology like heart-rate and blood-pressure. These effects can cause a change in cognitive mechanisms like memory retrieval and rule choice.

4.4 Module vs. Substrate

The question of whether one should use a module or an underlying substrate arises when realizing physiology in a cognitive architecture. Due to a possible difference in answers partially depending on the cognitive architecture, we ground our answer to this question in applying physiology to a modular architecture. We use ACT-R for specific examples.

Modules in a cognitive architecture are utilized to represent functional centers for different aspects of human behavior and cognition. A motor module, for example, would represent certain aspects of human motor behavior like the cognition behind the motor behavior or the outcome of this cognition. Modules in cognitive architectures are useful because they allow the separation of functionality and thus minimize interdependency among aspects of the cognition, essentially allowing the modeler to choose which features of cognition they wish to manipulate.

4.5 Using both concepts together in a cognitive architecture

If one chooses to follow a theory like that instantiated in ACT-R, modules and an underlying substrate should be used to add physiology to a cognitive architecture. The underlying physiological substrate should be used to represent functional aspects human physiology. This substrate should provide a fairly robust simulation of the processes occurring in the body during cognitive actions; at the very least, homeostatic processes should be available to represent primitive motivations like energy need and body temperature regulation.

Anderson (2007) proposes the encapsulation of the (normal) cognitive/behavioral function of neurological structures within architecture modules with ACT-R. Adding a module to represent the receiving and sending of signals to the body, in addition to the underlying substrate, follows an idea of module representation presented in ACT-R. It is less clear if more than one module should be used for this function; encapsulating this functionality within one module is feasible, given that one module avoids the heavy fixed interdependency issues one may hit by using two modules. However, it is also likely that encapsulating functionality in two modules may improve longevity and feasibility of expansions to architecture.

We instantiate the ideas presented in the connection of ACT-R and HumMod by representing the HumMod system as a substrate running in conjunction with the ACT-R architecture. To realize this substrate within the ACT-R architecture, we created a module that represents a physiological substrate and applies its effects to other modules. The physiology portion of the module is provided by the HumMod simulation system.

4.6 ACT-RΦ: ACT-R with physiology

ACT-RΦ is an extension of the cognitive architecture ACT-R that provides an additional representation of human physiology and allows for corresponding bidirectional connection between cognition and physiology. This allows one to model both the environmental and internal effects on central and peripheral physiology and the corresponding effects on cognitive parameters. We accomplish this physiological representation through the use of a system that presents a dynamic model of human physiology, HumMod. HumMod gives a user the option to use the included model of integrative model of physiology or alter that model via XML. The schema for the modified architecture is represented below in Figure 4.1.

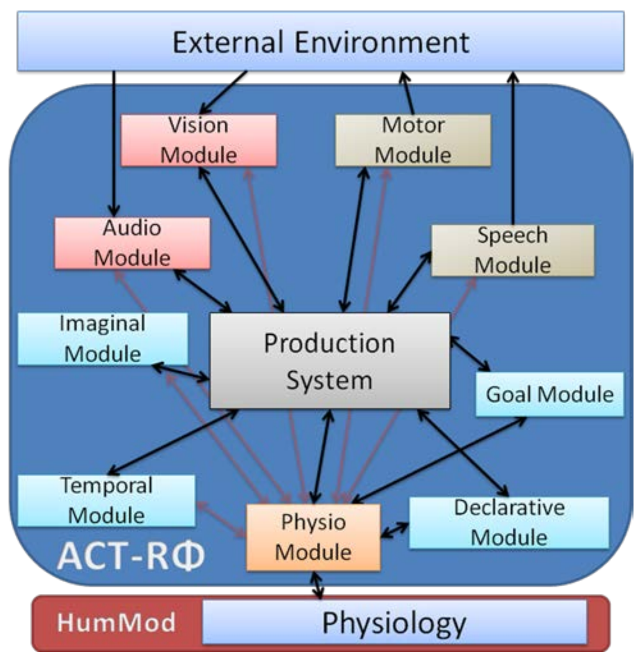


Figure 4.1: An overall schema of ACT-RΦ. The light links represent potential connections between within the extended architecture.

4.7 The physio module

We have developed a module for the representation of physiology and connection of this representation to other ACT-R modules. This module represents an underlying physiological substrate that dynamically obtains data for physiological variables from the HumMod simulation. In this case, a module is used as software construct not as another type of working memory buffer, although it has some memory-like aspects of remembering how recently the body has eaten, for example.

The "physio-substrate" buffer is used to request the module to begin retrieving physiological variable data from the HumMod simulation. One can accomplish this task by putting a chunk of type *phys-var* into the physio-substrate buffer. One can also explicitly request the value of any physiological variable by sending a request to the "phys-substrate" buffer. Physiological variables in HumMod can be explicitly set to a certain value by adding a chunk to the "efferent" buffer. A request can be sent in the form similar to buffers of existing modules. Currently, users must enter the exact variable name representation from HumMod to enact changes via the efferent buffer. This can prove cumbersome and requires a fairly low-level understanding of the model in HumMod to actually use the buffer. Alternative ways to provide a powerful, but more usable change to values is currently being explored.

These functionalities allow the user to model and simulate both cognitive effects on physiology and conversely effects of physiological variables on aspects (e.g., module parameters in ACT-R) of cognition and behavior. Currently, only implicit representations of connections between the physiological variables and ACT-R exist, accessible by function calls; examples of these connections are represented in table 4.1. A modeler may also write their own functions and simply use the module to get and set variable values.

Table 4.1: Example connections built into ACT-RΦ

Concept	**ACT-R**	**HumMod**
Startle Response	:ans	SympNS, Epi
Hunger	:mp, goal-focus, :ppm	GI-Lumen
Thirst	:mp, goal-focus, :ppm	Osmo-Rec, BodyH20.vol

5 A Extension of an Existing ACT-R 6 Model

We have developed a modified version of the ACT-R 6.0 subtraction model developed by Ritter et al. (2009). This model in ACT-RΦ includes provides a representation of the CNS-PNS loop afforded by the addition of the physio module, i.e. an action affects the central nervous system that consequently affects the peripheral nervous system, and this result feeds back to affect the CNS. With the extended model, we assume a fluctuation of sympathetic nervous system activity due to a scheduled sound that causes a form of a startle reflex. This startle reflex causes sympathetic nervous system activation in HumMod and affected variables feedback to affect the noise in retrieving declarative knowledge.

5.1 Connections made between ACT-R and HumMod

With this modified subtraction model, there exists a connection between production rules for fast-response to sudden sounds and modifications to physiological variables. The concept behind these new productions is related to those discussed by Kennedy and Bugajska (2010), who have modeled inhibition of fast-responses by immediately enacting activity in other buffers. In this model, the response to the sudden sound is not an inhibition response, but a physiological response. The model responds to a sudden sound with a change in attention and underlying physiology, i.e. a partial activation of the sympathetic nervous system. The sound also consequently leads to a change in memory. This creates partial simulated sympathetic nervous system response, e.g. increased heart rate, blood pressure, and epinephrine (adrenaline) levels that affect memory retrieval noise in ACT-R.

5.2 Modifications made to original ACT-R 6 Model

Production rules are added to handle the fast processing of the sudden sound stimulus. After the sensing of the loud noise in the aural-location buffer, the model clamps (sets to 1) the central nervous system autonomic nerve integration variable (via the efferent buffer in the physio module) that positively affects the adrenal nerve activation, thus simulating a feature of sympathetic system activation. The epinephrine HumMod variable is tied to Equation 1. The *ansMultiplier* variable was determined by solving for the equation when the :ans parameter was equal to the value found in the non-caffeine parameter set found by Ritter et. al. (2009), and the (current level) epinephrine value is equal to that which is the result of HumMod adrenal nerve activity leading to sympathetic activity (e.g., heart-rate) similar to that found in the original TSST study (Kirschbaum et al., 1993).

$$\frac{\left((Current\ Value)_{Epi} - Baseline_{Epi}\right) * ansMultiplier}{Max_{Epi}} \quad (1)$$

Physiological change is accomplished by sending a query to the efferent buffer in the physio module that specifies the specific variable to be changed and corresponding values; we change *SympsCNS.ClampSwitch* to value 1 and *SympsCNS.ClampLevel* to value 2.9. The sensing of the sound also results in a relatively short processing of the specific sound. The startle response and corresponding physiological change is relatively short-lived, and rules dealing with the subtraction task continue firing shortly after the encounter with the noise.

Table 5.1: Task Performance and number of attempts (avg.) by the original models (Ritter et al., 2009) and modified model.

Parameter Set	Performance (%)	Attempts
09 Average	81.4	48.1
09 Threatened	77.4	40.3
Modified	78.6	35.98

The modified model run 75 times produced output results (table 5.1b & figure 5.1) similar to the original subtraction model (threat condition), and consequently were lower than overall performance of the original model (signified as *09 Average* by Ritter et al., 2009). Though the average performance is similar between variables we do notice an expected drop in performance after the sound has been reduced and subsequent memory noise increases due to an increase in SNS activity. On average, the modified model also outputs less subtraction attempts than the original models in Ritter et al. (2009) (table 5.1). With this altered model we have shown that one can reproduce similar performance of a model with statically set parameters using a model that has parameters dynamically set via an underlying physiological substrate.

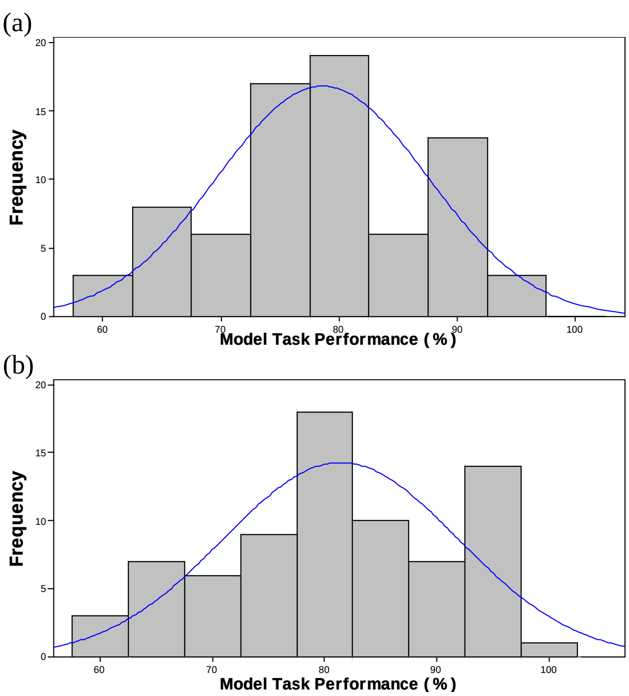


Figure 5.1: The task performance frequency distribution of the *09 average* model (a) and the *modified* model (b) N=75.

6 Discussion and Conclusions

We have introduced ACT-RΦ, a modified version of the ACT-R (6.0) architecture that utilizes a new module to send information to and receive information from HumMod, a system that simulates human physiology using a top-down level of human physiology. Currently, hard-wired connections in the system are allowed to be turned off to give the modeler more flexibility using the provided mechanisms or developing their own. This should aid in allowing users to computationally realize conflicting theories involving human cognition and physiology, but ultimately may force the modeler to self-code even more simple physiological based moderation of cognition. Reasonable hard-wired connections are currently being explored for implementation in the near future.

We have demoed a way in which one may use the ACT-RΦ system. The open-source nature and maneuverability of ACT-R (and consequently ACT-RΦ) allows for a user with experience to enact relatively fast changes in the ACT-R system without necessarily having to reload the entire system. This control, however, does come at the price of usability and physiological connections will likely be better served to have a higher-level representation.

As one continues to consider physiological ties to cognition and the consequent representation of these ties for modeling of human behavior, one is likely to continuously be confronted with issues and concerns arising from considering behavior and cognition on separate levels. We offer non-exhaustive points of discussion related to insights, implications, and possible concerns arising from opportunities afforded by ACT-RΦ.

6.1 Cognitive-physiological timing

HumMod allows one to continuously simulate physiology over time. Timing is now not only important on a subsecond scale, but on minute and even hourly scale. Effects can occur hours after original stimulus and can induce unforeseen effects in the model, providing variability. Homeostatic-appetitive motivation (e.g., energy needs) should be represented over a longer time-scale to better simulate continuous homeostatic mechanisms. This allowance also has implications for the simulation of effects of different causes of fatigue and effects of biological rhythms.

6.2 Visceral perception conflict

With an abundance of data from the HumMod simulation, one is likely to run into conflicts of differing sensations affecting cognitive parameters. An example of a possible conflict that has been studied in visceral sensory research is the idea of pain vs. hunger. It has been shown that there is a lower activation of pain processing regions of the brain if the source of the pain is directly related to the

fulfillment of nutrient deficiency, i.e. hunger (Coen, 2011). Integrating multiple lines of visceral sensory data into perceptual data within the system needs further exploration.
How should a simultaneous sensation and resulting motivations of (e.g., pain and hunger) be resolved within this system? Others have suggested a model that involves a higher priority placed to the more biologically relevant (e.g., Gregory et. al., 2003) but a definitive answer has yet to be reached. The PSI (Bach, 2009) theory perhaps offers some guidance on the matter as is provides separate motives to represent hunger (fuel) and pain (intactness) and reconciles concurrent activation of these representations by a MicroPSI agent in its environment.

6.3 Visualization and ease of use of ACT-RΦ

ACT-RΦ will have additional variables and be a more complex system. Presentation of the system's state will be important. A new visualization system will need to be added to better communicate what is going on in the combined architecture. It will remain difficult to sustain a good balance between showing the maximum amount of information without overloading the modeler with data. This is especially apparent with the integration of models of human physiology as one must not only content with the representation of human physiology, in the case of HumMod 5000+ variables, but also the existing cognitive architecture data and the corresponding connections between them.

As the physiological system continues to evolve a higher-level representation of knowledge, physiological variables, and their connections should be explored. The Herbal (Cohen et al., 2010) language is a good example of such a high-level language with graphic displays that might be extended to support this work. Herbal allows users to represent agents in its own XML-based language and can compile the model code into Soar, ACT-R, and JESS agents. Alternate representations also should be explored to allow for better validation of the models. At the present juncture, it remains difficult for independent validation of representations due to the sheer size, availability of data, and complexity of the system; high level representations may also aid with validation.

6.4 Developing models for military and game simulation

With a physiological connection, one is afforded more representations to simulate environmental effects on human behavior and cognition. This is important in developing more human-like agents for simulation environments like those in military or gaming. Increased autonomy and variability in models will allow for more realistic environments in military-based simulations like VBS2; Evertsz et al. (2009) also discuss the opportunity afforded by including models with moderators, albeit with a different level of representation.

From a military perspective one can now also begin to simulate more behaviors related to fear and anxiety and cognition (e.g., PTSD). Representing a constant anxiety state could potentially be accomplished by changing baseline parameters in the physiological parameters to represent the constant physiological activation (both central and peripheral) present in those in an anxiety/fear state and ensuring a hard-wired connection to cognitive parameters. These states could potentially be elicited due to internal factors (e.g., PTSD), or environmental factors (e.g. battle conditions).

6.5 Final remarks

Minds need brains to support them and brains need bodies to support them. As we become interested in more accurate models of cognition we will need to add a physiological representation of the substrates that on multiple levels support and implement cognition. ACT-RΦ is a step in this direction, and has provided some insights already.

Acknowledgements

This work was funded by the Office of Naval Research (EasyCog SBIR Phase II N091-086/P10008).

References

Aarts, H., Dijksterhuis, A., & de Vries, P. (2001). On the psychology of drinking: Being thirsty and perceptually ready. *British Journal of Psychology, 92*(4), 631-642.

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: OUP

Bach, J. (2009). *Principles of synthetic intelligence: PSI: an architecture of motivated cognition*. New York, NY: OUP

Coen, S. J. (2011). When pain and hunger collide; psychological influences on differences in brain activity during physiological and non-physiological gastric distension. *Neurogastroenterology & Motility, 23*(6), 485-489.

Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2010). Applying software engineering to agent development. *AI Magazine, 31*, 25-44.

Evertsz, R., Pedrotti, M., Busetta, P., Acar, H., & Ritter, F. E. (2009). Populating VBS2 with Realistic Virtual Actors. *In Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation*, pp. 1-8, 09-BRIMS-004. Sundance, UT.

Gregory, L. J., Yágüez, L., Williams, S. C. R., Altmann, C., Coen, S. J., Ng, V., et al. (2003). Cognitive modulation of the cerebral processing of human oesophageal sensation using functional magnetic resonance imaging. *Gut, 52*(12), 1671-1677.

Groeneweg, F. L., Karst, H., de Kloet, E. R., & Joëls, M. (2011). Rapid non-genomic effects of corticosteroids and their role in the central stress response. *Journal of Endocrinology, 209*(2), 153-167.

Gross, J. B., Gunzelmann, G., Gluck, K. A., Van Dongen, H. P. A., & Dinges, D. F. (2006). Computational modeling of the combined effects of circadian rhythm and sleep deprivation. *In Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 297-302. Vancouver, BC, Canada.

Gunzelmann, G., Gross, J. B., Gluck, K. A., & Dinges, D. F. (2009). Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling. *Cognitive Science, 33*(5), 880-910.

Gunzelmann, G., Moore Jr, R. L., Salvucci, D. D., & Gluck, K. A. (2011). Sleep loss and driver performance: Quantitative predictions with zero free parameters. *Cognitive Systems Research, 12*(2), 154-163.

Guyton, A. C., Coleman, T. G., & Granger, H. J. (1972). Circulation: Overall regulation. *Annual review of physiology, 34*(1), 13-44.

Hester, R. L., Brown, A. J., Husband, L., Iliescu, R., Pruett, D., Summers, R., et al. (2011). HumMod: A modeling environment for the simulation of integrative human physiology. *Frontiers in physiology, 2*(12).

Kennedy, W. G., & Bugajska, M. D. (2010). Integrating Fast and Slow Cognitive Processes. *In Proceedings of the Tenth International Conference on Cognitive Modeling*, 121-126. Philadelphia, PA.

Kirschbaum, C., Pirke, K.-M., & Hellhammer, D. H. (1993). The "Trier Social Stress Test": A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology, 28*(1-2), 76-81.

Lewis, M. S., Snyder, P. J., Pietrzak, R. H., Darby, D., Feldman, R. A., & Maruff, P. (2011). The effect of acute increase in urge to void on cognitive function in healthy adults. *Neurourology and Urodynamics, 30*(1), 183-187.

McMorris, T. (2009). Exercise, dehydration and, cognitive function *Exercise and Cognitive Function* (pp. 115-134): John Wiley & Sons, Ltd.

Mogg, K., Bradley, B. P., Hyare, H., & Lee, S. (1998). Selective attention to food-related stimuli in hunger: Are attentional biases specific to emotional and psychopathological states, or are they also found in normal drive states? *Behaviour Research and Therapy, 36*(2), 227-237.

Montani, J.-P., & Van Vliet, B. N. (2009). Understanding the contribution of Guyton's large circulatory model to long-term control of arterial pressure. *Experimental Physiology, 94*(4), 382-388.

Ritter, F. E., Dancy, C., & Berry, K. (2011). The case for including physiology in cognitive modeling. *In Proceedings of the Biologically Inspired Cognitive Architectures*, 303. Washington, D.C.

Ritter, F. E., Kase, S. E., Klein, L. C., Bennett, J., & Schoelles, M. (2009). Fitting a model to behavior tells us what changes cognitively when under stress and with caffeine. *In Proceedings of the Proceedings of the Biologically Inspired Cognitive Architectures Symposium at the AAAI Fall Symposium. Keynote presentation*, 109-115.

Ritter, F. E., Reifers, A. L., Klein, L. C., & Schoelles, M. J. (2007). Lessons from defining theories of stress. In W. D. Gray (Ed.), *Integrated Models of Cognitive Systems* (pp. 254-262). New York, NY: OUP.

Tsigos, C., & Chrousos, G. P. (2002). Hypothalamic–pituitary–adrenal axis, neuroendocrine factors and stress. *Journal of Psychosomatic Research, 53*(4), 865-871.

Tuk, M. A., Trampe, D., & Warlop, L. (2011). Inhibitory Spillover. *Psychological Science, 22*(5), 627-633.

Author Biographies

CHRISTOPHER DANCY is a doctoral student in the College of IST at the Penn State. He is interested in the investigation and modeling of connections between physiology, emotion, and cognition.

FRANK RITTER is on the faculty of the College of IST, an interdisciplinary academic unit at Penn State to study how people process information using technology. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Human Factors*, *Cognitive Systems Research*, and *IEEE SMC IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans.*

KEITH BERRY is a US Army Reserve Flight Surgeon and a doctoral student in the Bioengineering department at the Pennsylvania State University. His interests are focused on the application of soft computing, physiologic modeling, and ensemble methods to clinical decision support to improve the quality, consistency, and efficiency of medical care. He is a former research fellow at the US Army Institute of Surgical Research.

Predicting and Classifying Pedestrian Behavior Using an Integrated Cognitive Architecture

Unmesh Kurup, Christian Lebiere, Anthony Stentz, Martial Hebert

Unmesh Kurup
Christian Lebiere
Department of Psychology, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh PA 15213
unmeshk@andrew.cmu.edu, cl@andrew.cmu.edu

Anthony Stentz
Martial Hebert
Robotics Institute, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh PA 15213
tony@andrew.cmu.edu, hebert@andrew.cmu.edu

Keywords:
Behavior Classification, Behavior Prediction, Integrated Cognitive Architecture

ABSTRACT: *We present an integrated system that combines a state-of-the-art object detection algorithm with the ACT-R cognitive architecture. This system represents the first step towards a more complete integration of perception and cognition for solving real-world tasks. We use this system to detect pedestrians on a sidewalk and classify and predict their behavior. Perception provides a bounding box for each pedestrian detected at each time-step. ACT-R uses this information to track the pedestrians across the scene. Simultaneously, ACT-R checks the tracks against a set of spatial features. During the learning phase, the model learns to associate sequences of these features with the appropriate behavior. During classification, the detected feature sequence is used to retrieve the appropriate behavior from memory. During prediction, partial detected feature sequences are used to retrieve the associated behavior from memory. We provide results of classification and prediction for single and multiple behavior sets, and discuss future work.*

1. Introduction

With some notable exceptions (Trafton et al., 2009) (Avery & Kelley, 2005) (Hanford et. al., 2009), high-level cognitive systems are rarely integrated with low-level perceptual processes. In many situations such integration is unnecessary given the differences in goals (interacting with the environment for perception vs. reasoning or planning about abstract scenarios for cognition), nature of the representations involved (quantitative vs. qualitative) and constraints of the problems themselves (dynamic vs. static). Even in robotics, normal tele-operated robots have or need very little intelligence. Even autonomous robots can perform highly constrained tasks without the benefit of any cognitive help. Similarly, high-level planners or information fusion systems rarely need to interact with or even sense the external world. The real challenge occurs when an autonomous robot needs to do a variety of tasks (or a single task across a variety of environments) thereby requiring the benefits of both robust perception and cognition.

In previous work, Kurup et al. (2011) have demonstrated the use of a high-level cognitive architecture for refining the output of perception. In that work, the output of perception was simulated and an ACT-R model was used to perform tracking and error-correction on the generated data. Here, we expand on their approach by replacing the simulation with a state-of-the-art object detection algorithm and use it to detect pedestrians. The resulting detections are used by an ACT-R model to track pedestrians, and classify and predict their behavior.

2. An Integrated Cognitive Architecture

ACT-R is a well-known cognitive architecture for behavior generation and cognitive modeling (Anderson, 2007) (Anderson et al., 2004). However, ACT-R is primarily a theory of high-level cognition and is agnostic with respect to the particulars of low-level perception. Instead, ACT-R provides an interface (via its perceptual buffers) that is used to interact with perception and the external world. In our work, we use a state-of-the-art object detection algorithm to serve as a perceptual module for ACT-R.

2.1 Discriminatively Trained Deformable Part Models

The Felzenswalb et al. (2010) algorithm for object detection is a "complete learning-based system for detecting and localizing objects in images". The algorithm represents various objects using part models that are deformable allowing for detection of previously unseen configurations or poses. These models are discriminatively trained using only a bounding box for each object in the image. This training allows the algorithm to detect a wide variety of objects including people at a variety of resolutions. Figure 1 shows the result of applying the algorithm to one image from real-world data. The output of the algorithm is a set of bounding boxes that contain the objects detected. Each bounding box also has an associated value that is a measure of the algorithm's confidence that the box contains an object of the specified type (people in our case). Depending on the parameters, the algorithm's performance can vary but in general a low threshold for detection allows the algorithm to maximize the number of objects found. The drawback is that this low threshold can also lead to false-positives as can be seen in Figure 1 (the tree and the reflection of a person in the window of the store are both detected by the algorithm as people). We use an ACT-R model to combine object detections across multiple time-steps to track each object. This process also eliminates false positives.

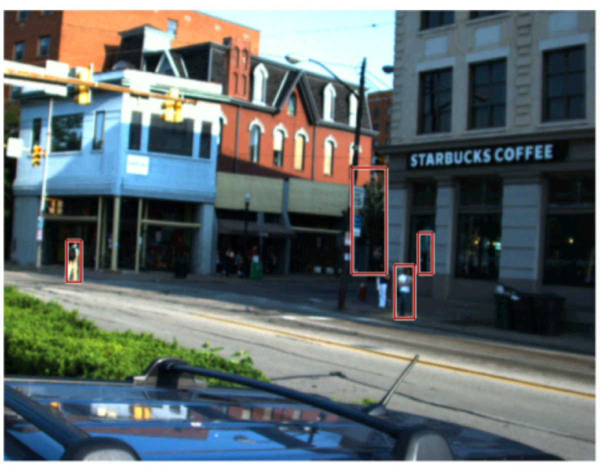


Figure 1. The result of running the Felzenswalb object detection algorithm with a threshold of -0.5.

2.2 ACT-R

The ACT-R cognitive architecture is a modular, neurally-plausible theory of human cognition. The ACT-R architecture describes cognition at two levels – the symbolic and the sub-symbolic. At the symbolic level, ACT-R consists of a number of modules each interacting with a central control system (Procedural module) via capacity-limited buffers. Modules represent functional units with the most common ones being the Declarative module for storing declarative pieces of knowledge, the Goal module for storing goal-related information, the Imaginal module which supports storing the current problem state, and the Perceptual (Visual and Aural) and Motor modules that support interaction with the environment. The only way to control a module and access the results of its processing is through that module's buffer. Modules can operate asynchronously, with the flow of information between modules synchronized by the central procedural module.

Declarative memory stores factual information in structures called chunks. Chunks are typed units similar to schemas or frames that include named slots (slot-value pairs). Productions are condition-action rules, where the conditions check for the existence of certain chunks in one or more buffers. If these checks are true, the production is said to match and can be fired (executed). Only one production can fire at a time. In its action part, a production can make changes to existing chunks in buffers or make requests for new chunks. ACT-R also has an underlying sub-symbolic (numerical) layer that associates values (similar to neural activations) to chunks and productions. These activation (utility in the case of productions) values play a crucial role in deciding which productions are selected to fire and which chunks are retrieved from memory. ACT-R also has a set of learning mechanisms that allow an ACT-R model to learn new declarative facts and production rules, and to modify existing sub-symbolic values to reflect the statistics of the environment.

There are two mechanisms critical (for the purposes of the current work) to the process of retrieval from declarative memory – partial matching and blending. A request for retrieval of a chunk from declarative memory is made by specifying a set of constraints on a subset of the slot values of chunks. If there are multiple chunks that exactly match the specified constraints, the chunk with the highest activation value is retrieved. Normally, the retrieval mechanism only considers those chunks that exactly match the request criteria. However, when partial matching is enabled, the retrieval mechanism can retrieve the chunk that matches the request criteria to the greatest degree. This means that the retrieval mechanism can retrieve chunks that are closely related to the specified constraints when there are no chunks that match the constraints exactly. Partial matching is particularly useful in continuous or dynamic situations such as tracking pedestrians.

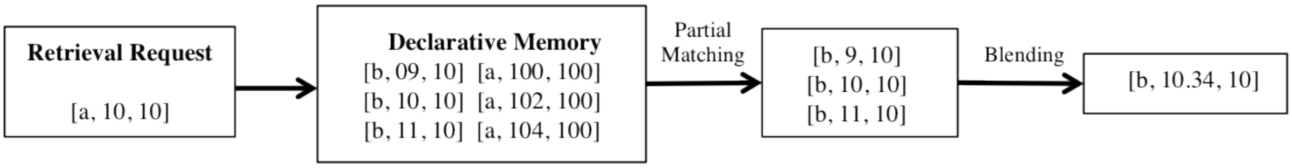


Figure 2. An example of how partial matching and blending affect retrieval from declarative memory.

Blending is a form of generalization where, instead of retrieving an existing chunk that best matches the request, blending produces a new chunk by combining the relevant chunks. The values of the slots of this blended chunk are the average values for the slots of the relevant chunks weighted by their respective activations, where the average is defined in terms of the similarities between values. For discrete chunk values without similarities, this results in a kind of voting process where chunks proposing the same value pool their strengths in a manner similar to machine learning techniques such as k-nearest neighbors (Shakhnarovish & Indyk, 2005). For continuous values such as numbers, a straightforward averaging process is used. For discrete chunk values between which similarities (as used in partial matching) have been defined, a compromise value that minimizes the weighted sum of squared dissimilarities is returned. Figure 2 gives an example of the result of partial matching and blending during retrieval from declarative memory. A more detailed explanation of blending and partial matching can be found in (Lebiere C., 1999) (Gonzalez & Lebiere, 2005) (Kurup, Lebiere & Stentz, 2011).

2.3 Integrated System

Figure 3 shows the complete integrated system. The images of pedestrians walking are first passed through the object detector that has been previously trained to detect humans. The results of the person detector are passed to the ACT-R model that combines these results with its expectations of the locations of pedestrians to track these pedestrians across the scene. Simultaneously, these tracks are also used by the model to classify and predict pedestrian behavior. The following section expands on how the model achieves these goals. The entire system operates in a batch mode where all persons are detected across all images and this data is then fed into the ACT-R model. This step is necessary due to the fact that the person detector takes approximately 30 seconds for each image, rendering real-time detection impossible. There is ongoing research on improving the real-time capability of this algorithm and we have recently integrated this model with another real-time detector and successfully used the system to perform classifications in real-time.

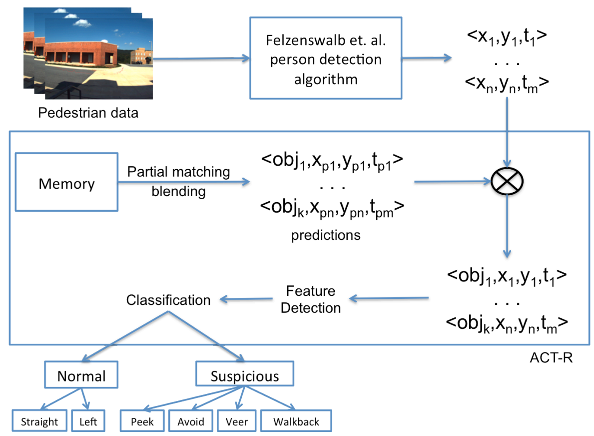


Figure 3. The integrated system for pedestrian behavior classification and prediction.

3. Classifying and Predicting Pedestrian Behavior

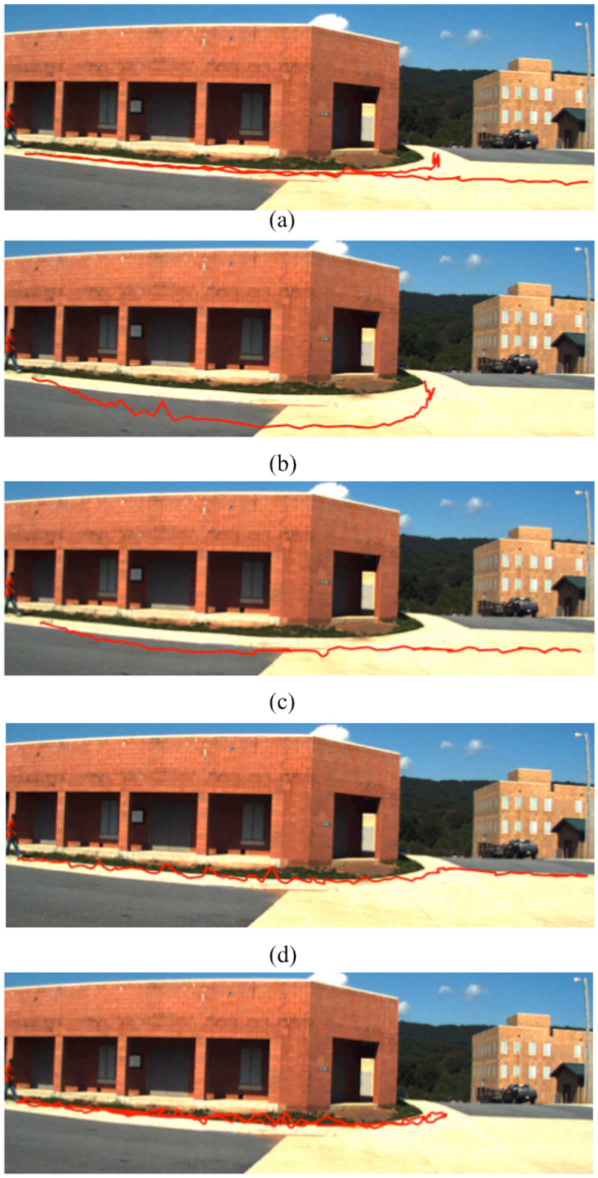
3.1 Checkpoint Scenario

The checkpoint scenario consists of classifying and predicting pedestrian behavior at an intersection with a checkpoint. A pedestrian enters the scene from the left (a convenience for expository purposes) and walks towards the right on the sidewalk. Each pedestrian can perform one of six behaviors as detailed below.

1. Normal Behaviors - At the intersection, the pedestrian can either go straight or turn left. If there is a checkpoint on the path to the left, the pedestrian walks through it. These two behaviors – walking straight or turning left – constitute the two normal behaviors in the scenario. An example of the tracks produced by pedestrians exhibiting these behaviors is shown in Fig 4(a).
2. Peek Behavior - In a peek behavior (Fig 4(b)), the pedestrian steps out on to the street to look for a checkpoint around the building and, finding that it is not present, proceeds to turn left at the intersection.
3. Detour Behavior - In a detour behavior (Fig 4(c)), the pedestrian steps out on to the street to look for a checkpoint (like in the peek behavior) and finding the checkpoint present proceeds to walk straight rather than turn left.
4. Veer Behavior - In a veer behavior (Fig 4(d)), the pedestrian turns left at the intersection, sees the

checkpoint, veers off and proceeds to walk straight on instead of continuing along to the left.

5. Walkback Behavior - Finally, in the walkback behavior (Fig 4(e)), the pedestrian turns left at the intersection, spots the checkpoint, reverses course and proceeds to walk back to the starting location.



(a) (b) (c) (d) (e)

Figure 4. Examples of behaviors Normal (straight & left), Peek, Detour, Veer and Walkback respectively

3.2 Tracking Pedestrians

The input from perception is in the form of bounding boxes for each person in the image together with a confidence value. The ACT-R model converts this bounding box into a single x,y (egocentric) coordinate that is the mid-point along the base of the bounding box. The model represents this information together with the rate of change (delta) in a chunk in declarative memory. The model tracks pedestrians as follows – at each timestep, the model generates an expected location for each person that it knows about (using the partially matched and blended location and delta values from declarative memory chunks associated with that person). It then assigns each location from perception to a person by picking the person closest to that location. If a location is left over (i.e., all known persons have been assigned to locations), a new person is identified at that location. ACT-R's partial matching and blending ensure that the expected location generated by the model places greater emphasis on the recent history of an object in generalizing over noise in the perceptual data, while smoothing out random variations in projecting moving directions. This process can be seen as a spatial version of the general modeling pattern of projecting future system states in dynamic environments (e.g., Lebiere, Gonzalez & Warwick, 2009).

3.3 Features

As the model tracks a pedestrian, it also checks to see if their path encounters certain regions in the scene. These regions, called features, are pre-determined by the modeler. The current model has six features – straight1, straight2, straight3, detour, left and veer (shown in Figure 5). If the path intersects a region, it is said to have the corresponding feature. As an example, the peek behavior would exhibit the feature sequence – straight1, detour, left. During the learning phase, the model learns a chunk that associates the detected sequence of features with the appropriate behavior.

For classification, the model uses the feature sequence to retrieve the appropriate behavior chunk from memory. It performs this retrieval only after the model has finished tracking the pedestrian and has all the features in the sequence. For prediction, the model tries to retrieve a behavior chunk from memory based on partial feature sequences. This allows the model to predict the behavior before a pedestrian has completed his or her path across the scene. This model can be seen as a spatial specialization of a more abstract model that recognizes and anticipates behavior by matching action patterns, illustrating the versatility of the cognitive architecture across both symbolic and metric domains (Oltramari & Lebiere, 2011).

4. Results

We collected data using live actors at the Mobile Operations for Urban Terrain site in Fort Indiantown Gap, Pennsylvania, USA. There were 4 examples of each of the 6 behaviors for a total of 24 behaviors. We also collected data with multiple behaviors. The multiple behavior data is a continuous sequence with

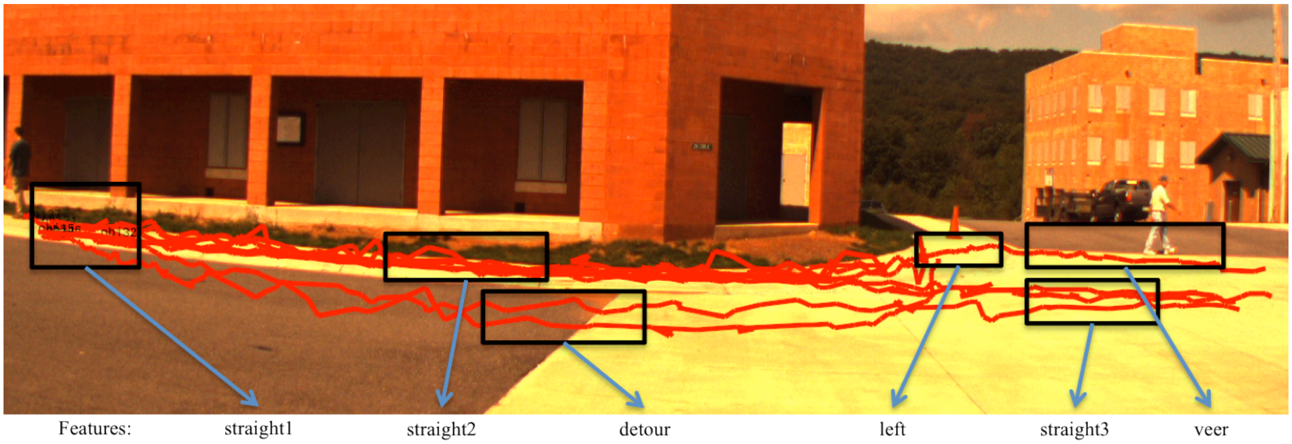


Figure 5. Each black box represents a feature. A pedestrian's track is classified depending on the sequence of features the track intersects.

Classification (Single Behavior Set)		Classification (Multiple Behavior Set)	
Made	99.3%	Made	46.5%
Correct	99.15%	Correct	30.2%
Incorrect	0.15%	Incorrect	16.3%

(a)

Prediction (Single Behavior Set, 2 features)		Prediction (Single Behavior Set, 3 features)	
Made	100%	Made	99.67%
Correct	66.3%	Correct	82.67%
Incorrect	33.7%	Incorrect	17%

(b)

Figure 6. The results of behavior classification and behavior prediction

multiple actors performing a variety of behaviors. There were up to 3 actors in the scene at any one time and on occasion, their paths intersected. This made the process of tracking and classifying their behavior challenging.

We ran the Felzenswalb algorithm with a low threshold value of -0.5 (there is no specified value but the usual range is -0.5 to 0.5) to make sure that all person-like objects were detected. The low threshold meant a number of false-positives but we used two heuristics to eliminate most of them. The first heuristic was a plane of interest that captured the fact that people are most likely found on the street or the sidewalk and not on the side of buildings or in the sky. The second heuristic we used was that a new person enters into the scene from one of the ends rather than appearing in the middle.

We ran monte-carlo simulations (1000 iterations) of classification on both single and multiple behavior examples. In both cases, we randomly selected 3 examples of each behavior (18 examples in total) for the learning set. For single behavior classification, we used the remaining example of each behavior for the testing set (6 examples in total). For multiple behavior classification, the learning set remained the same while the testing set was the multiple behavior set. Fig 6(a) shows the results of classification on both sets. The results are very good for the single behavior case, with the model making a classification in 99.3% of the cases. The model correctly classified 99.15% of the total behaviors and incorrectly classified 0.15% of the behaviors. The results of classification in the multiple behavior case are poorer, with the model only making a classification 46.5% of the time. It correctly classified 30.2% of the behaviors and incorrectly classified 16.3% of the behaviors.

It should be noted that the number of correct classifications in the multiple behavior case is a lower bound. This is due to the way in which the correctness or incorrectness of a classification is counted. The multiple-behavior set has a total of 10 behaviors and the assumption of the monte-carlo simulator is that the model outputs its classifications in the order in which they happen. One drawback to this method is that if the model fails to produce a classification for a pedestrian, the remaining classifications are off by one and results in an incorrect count. For example, let the ground truth be pedestrians 1 through 4 with behaviors normal-straight, walkback, veer and peek respectively. If the model fails to correctly track pedestrian 2, it produces no classification for that pedestrian resulting in the following classification sequence – normal-straight, veer and peek. This translates to one correct classification, one non-classification and 2 incorrect classifications when the reality is 3 correct classifications and 1 non-classification. It should be emphazied that this is not a shortcoming of the model or of ACT-R, but a drawback of automatically checking results. We have tried several methods to bridge this gap but, short of manually checking the results, there is no automated way to determine how well the model is actually doing.

For behavior prediction, we tested the model for two conditions based on when the prediction was made. In the first case, we made the prediction after the first two features were found while in the second condition we made the prediction only after the first three features were found. We tested predictions only on the single behavior case due to problem described earlier of correctly analyzing the results produced by the model in the multiple behavior set. Figure 6 (b) shows the results for these two conditions. As expected, the results are average for the two-feature condition while the extra feature improves the performance from 66.3% to 82.67%.

5. Conclusion & Future Work

Intelligent robots require both high-level cognition and low-level perception. Cognitive architectures have been effective for the former but are usually used in isolation from the latter. We presented a system that integrated a state of the art object detector with the ACT-R cognitive architecture to perform pedestrian tracking and behavior classification. Our initial results have been promising and have indicated at least two avenues for future research.

An immediate need is for automatic recognition of the spatial features used to classify behaviors. In the current paper, these features are specified by the modeler but ongoing work has shown promise in automatically detecting such features. In addition, we are currently using a stationary camera and egocentric coordinates to capture images. Our goal is to move to tracking and classification in situations where the camera is in motion.

In the longer term, our goal is to have better top-down integration between cognition and perception, allowing cognition to provide context and attention resulting in improved performance for perception. Both directions of research emphasize the need to deeply integrate perception, cognition and action in cognitive robotics rather than view each function in isolation.

6. Acknowledgements

This work was conducted through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016.

7. References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe. New York, NY:* Oxford University Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 111 (4), 1036-1060.

Avery, E., Kelley, T. (2005). Using cognitive architectures to improve robot control. In *Proceedings of the 15th conference on Behavior Representation in Modeling and simulation*. 15-18.

Fauconnier, G., & Turner, M. (1998). Conceptual integration networks. *Cognitive Science*, 22(2), 133-187.

Felzenswalb, P., Girshick, R., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (9)*.

Gonzalez, C., & Lebiere, C. (2005). Instance-based cognitive models of decision-making. In D. Zizzo, & A. Courakis (Eds.), *Transfer of Knowledge in Economic Decision Making*. New York: Palgrave McMillan.

Hanford, S., Janrathitikarn, O. & Long, L.. N. (2009). Control of mobile robots using the soar cognitive architecture. *Journal of Aerospace Computing, Information and Communication*. 6. 69-91.

Kurup, U., Lebiere, C., & Stentz, A. (2011). Integrating perception and cognition for AGI. In *Proceedings of the fourth international conference on Artificial General Intelligence*.

Lebiere, C. (1999). The dynamics of cognitive arithmetic. *Kognitionswissenschaft, 8 (1),* 5-19.

Lebiere, C., Gonzalez, C., & Warwick, W. (2009). Convergence and constraints revealed in a qualitative model comparison. *Journal of Cognitive Engineering and Decision Making* 3(2), 131-155.

Oltramari, A., & Lebiere, C. (2011). Extending Cognitive Architectures with Semantic Resources. In *Proceedings of the Fourth Conference on Artificial General Intelligence*.

O'Reilly, R.C., & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. Cambridge, MA: MIT Press.

Rumelhart, D.E., McClelland, J. L. and the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: MIT Press.

Shakhnarovish, D., & Indyk, P. (2005). *Nearest-Neighbor Methods in Learning and Vision*. Cambridge, MA: MIT Press.

Trafton, J. G., Harrison, A. M., Fransen, B. & Bugajska, M. (2009). An embodied model of infant gaze-following. In *Proceedings of the International Conference of Cognitive Modeling*.

Author Biographies

UNMESH KURUP is a post-doctoral researcher in the Psychology Department at Carnegie Mellon University. His areas of interest include Knowledge Representation, Cognitive Architectures and Diagrammatic and Spatial Reasoning.

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 hybrid cognitive architecture and was co-author with John R. Anderson of the 1998 book The Atomic Components of Thought.

ANTHONY STENTZ is a Research Professor in Carnegie Mellon University's Robotics Institute and Director of the National Robotics Engineering Center. He conducts research in autonomous vehicles, dynamic planning, and multi-robot coordination.

MARTIAL HEBERT is Professor at the Robotics Institute, Carnegie Mellon University. His interests include object recognition and scene analysis in images, video interpretation, interpreting 3D point clouds from range sensors. Current focus includes automatic discovery of object models form large image collections and use of geometric reasoning in image interpretation. He is investigating applications of computer vision to mobile robotics, e.g., unmanned ground systems, manipulation, and to assistive systems.

Learning & Prediction in Relational Time Series: A Survey

Terence K. Tan
Christian J. Darken
The MOVES Institute
Naval Postgraduate School
700 Dyer Road, Watkins Ext. 265
Monterey, CA 93943-5001
ktan@nps.edu, cjdarken@nps.edu

Keywords:
Learning, Prediction, Relational Time Series, Sense making

ABSTRACT: *Making sense out of a stream of incoming percepts is the first step in any agent's cognition process. The purpose of sense-making is usually to facilitate sound decision making, often by making predictions of future events or actions. In the case that the percepts are relational, the technologies available for this task are mainly based on production systems or statistical graphical model inferencing processes such as Bayesian networks. To apply these approaches, it is necessary that domain knowledge be known or that examples are available to a supervised learning process. Darken (2005) proposed a situation learning (SL) approach to learn a string of percept sequence into a set of overlapping situations. This approach has much potential for learning and predicting in domains that are characterized by high variability and great number of predicates and terms that become known only at runtime, and which feature a trending or moving context environment. In this paper, we attempt to define relational time series (RTS) and its characteristics for evaluating current learning approaches for learning and prediction of RTS. We also report the prediction accuracies of various prediction techniques based on SL in a benchmark environment.*

1. Introduction

Making sense out of a stream of incoming percepts is the first step in any agent's cognition process. This stream of percepts can be described as a relational time series, which is a time series of percepts in first order logic representation. Relational representation is a natural way to express the relations among the constants in the virtual environment. We can use such a time series to learn the behavior of other agents. Furthermore, relational representation allows inference of additional knowledge from the structural properties afforded by the relations among the constants. In particular, such structural properties can help to predict even atoms that we have not seen before.

In this paper, we first describe the relevance of event prediction to behavior representation in modeling and simulation (BRIMS). Next, we frame the problem of event prediction in relational time series and describe its challenging characteristics of learning and prediction. We then qualitatively evaluate Markovian and Non-Markovian approaches against the problem characteristics. Finally, we present an overview of a situation learning (SL) approach to learning, and present a few prediction techniques in conjunction with the SL and experimental results.

2. Relevance of Prediction Task to BRIMS

Prediction capability is important in many applications. In Modeling and Simulation, Kunde and Darken (2006) showed that predictive ability enhanced the realism of the behavior of a simulated military commander. Human beings do not make decisions based only on the current situation, but also on the predicted development (Kurby & Zacks, 2008). Klein (1999) describes the process of prediction as "mental simulation" while Fauconnier and Turner (2002) describe it as "running the blend". The ability to predict future events and to act based on the predicted states can enhance the fidelity of an agent behavior model.

There are several possible ways of using the predictive information. In production system, where decisions are made based on rules, we can use the future states in conjunction with the current state in the precondition of the rules. A rule that delays a call for fire can be described in Figure 1, using CLIPS syntax. The interpretation of this rule is: if the number of enemy sighted is one, and the future number of enemy sighted is one, and then call for fire. If the number of enemy sighted is one, and the future number of enemy sighted is greater than one, then, issue the wait command. The rationale for these rules is that the agent cannot handle more than one enemy. In reinforcement learning, given a situation, the agent chooses an action from its policy that maximizes some goodness measure. In exploitation mode, the agent will simply choose the action with the highest cumulative reinforcement. In exploration mode, the agent will randomly choose other actions. We envision that in exploitation mode, the agent can generate a prediction of a sequence of future events based on each possible action in the policy, and choose the action that has the desirable state in the predicted sequence. From these examples, we can see that predictive capabilities can be useful to enhance the performance of an agent.

```
(defrule rule1
    (NumberOfEnemySighted 1)
    (Future (NumberOfEnemySighted 1))
     =>
    (assert (Command Fire))
)
(defrule rule2
    (NumberOfEnemySighted 1)
    (Future (NumberOfEnemySighted>1))
     =>
    (assert (Command Wait))
)
```

Figure 1: A CLIPS rule that uses predicted state

3. Relational Time Series

We now define the relational time series (RTS) and the prediction problem, and discuss its characteristics.

3.1 Definitions

We define a relational time series (RTS) as a sequence of relational percepts. Each percept is a ground atom defined as $p_i = r(c_1, c_2, \dots, c_m)$, where r is the predicate and $c_{j\in(1..m)}$ are constants that represent objects. An example of a RTS is given in Figure 2. There are two types of percept: point and interval. The point percept exists or is active for a point in time and immediately ceases to exist. For example, a percept that describes "a ball hitting the wall" becomes obsolete immediately after it occurred. An interval percept occurred and remains true until something happens that change its state. For example, a percept that describes "a ball is in the box" is true until the ball is removed. The interval percept has a '+' indicator in the predicate as shown in Figure 2. A percept that is true is said to be active. The interval percept becomes inactive when a special type of point percept arrives, indicated by '-' in the predicate.

P_i	Time	RTS	Semantics
P_1	1	(loc+ Ed road)	Ed is at location road
P_2	2	(loc + Fox1 road)	Fox1 is at location road
P_3	3	(goE Fox1 east)	Fox1 is going east
P_4	3	(loc- Fox1 road)	Fox1 is NOT at location road
P_5	10	(loc + Fox2 road)	Fox2 is at location road
P_6	11	(goE Fox2 east)	Fox2 is going east
P_7	11	(loc- Fox2 road)	Fox2 is NOT at location road

Figure 2: An example of RTS

The prediction problem can then be defined as follows. Let $\{p_1 p_2 \dots p_n\}$ be the sequence of percepts from the time the agent started learning till the present time, where i in p_i refers to the running index of each incoming percept. A one-step prediction problem is then $\{p_1 p_2 \dots p_n\} \vdash p_p$ where $\vdash$ is an operator that weakly implies that p_p is the next most likely percept. A two-step prediction problem is defined as $\{p_1 p_2 \dots p_n p_{p1}\} \vdash p_{p2}$, given that $\{p_1 p_2 \dots p_n\} \vdash p_{p1}$. This means that the percept predicted by a one-step predictor is used for the second step prediction. The two-step prediction problem can be generalized to a multiple-step prediction problem.

3.2 Problem Characteristics

Learning and prediction in RTS from unknown environments is a hard problem because of a set of challenging characteristics. (1) Since there is no knowledge of the environment, there can be no predefined statistical graphical model or structure for knowing what kinds of atom that will arrive next. This leads to the second characteristic, which is (2) arbitrarily many constants and relations of arbitrary arity. This results in a large state space. To make the matter worse, the sequence of percepts can be (3) chaotic, and a function of a moving context, with different percept subsequences occurring in different contexts. While each atom can be treated as a proposition, ignoring the (4) relational structural properties can miss out opportunities to predict atoms that have not been seen before.

The above characteristics of RTS present many challenges and opportunities for sense-making. We have not seen any research effort that directly addresses the RTS problem. Research areas such as statistical relational learning or operator observable model are the most relevant. However, they do not directly address all RTS characteristics. Sun & Giles (2001) provide a nice introduction and review of approaches for sequence learning. Their review appears to address a sequence of proposition (versus atom), and do not directly address all the characteristics of RTS. In the next section, we will review current possible approaches to RTS and evaluate them against the characteristics of RTS.

4. Current Approaches for RTS

In order to succeed in learning and prediction in RTS from unknown environments, the algorithms must demonstrate online structural flexibility in its learned knowledge base, and flexibility in using the knowledge base to make predictions. Here, we discuss possible learning approaches by organizing them into Markovian and non-Markovian learning approaches.

4.1 Markovian

Markovian approaches refer to approaches that assume Markov properties. These approaches are variations of Markov chain or Hidden Markov Model. In Markov state machine (MSM), each state with the same input can transit probabilistically to different states. Markov state machine is sometimes called Markov Chain (Luger, 2008, Section 9.3.5). If the transition is defined based on the current state, it is termed first order

Markov model. If the transition is based on n previous states, it is termed n-order Markov Model. The main limitation of the Markov lies in its limited potential to generalize to novel situations due to its strict order. A new situation may simply have the order of two states switched, or have extra trivial percepts in between the states, but the Markov model will not detect such a switch and treat it as a new sequence, resulting in over-fitting. Furthermore, these approaches treat each relational atom as propositional, and does not leverage on the relational structure to make inferences.

The observable operator model (Jaeger, 2000) is described to be a generalization of the hidden Markov model. It models a stochastic process in order to compute the probability distribution over all possible future sequences, given that a sequence of observation has been observed. The probability of observing a future sequence is:

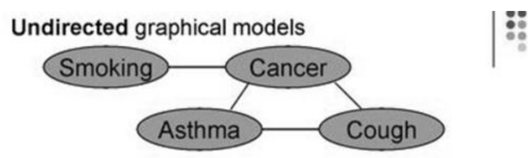
$$P\left(Y_0 = a_{i_0}, Y_1 = a_{i_1}, \ldots, Y_k = a_{i_k}\right) = \mathbf{1} T_{a_{i_k}} T_{a_{i_{k-1}}} \ldots T_{a_{i_0}} w_0$$

Where

- Y_0, Y_1, …, Y_k are random variables in the sequence
- a_{i_0}, a_{i_1}, … a_{i_k} are the observables corresponds to Y_0, Y_1, …, Y_k and i refers to different types of observable.
- **1** is an identity vector that attempts to sum the column vector to form the probability value
- $T_{a_{i_k}}$ is the operator corresponds to an observable at position k in the sequence where $T_a=M^TO_a$ where M^T is the transpose of the state transition matrix and O_a is a diagonal matrix that express the conditional distribution of each observation given each state.
- w_0 is the initial distribution of the hidden states.

The learning process requires prior manual estimation of a dimension (number of feature) and the set of features. This is a potential limitation in an unknown environment. Spanczer (2007) identified that learning in OOM, though Simple, is a partially solved problem. He also highlighted the difficulty of choosing characteristics and indicative events in order to have an efficient algorithm. In addition, OOM uses proposition representation, which does not leverage the structural properties to make prediction.

Statistical Relational Learning (SRL) is not strictly a Markovian approaches, but can be implemented with either Markovian or Non Markovian techniques. SRL attempts to combine first order logic with statistical learning (Getoor & Taskar, 2007). The relational learning addresses the relational structure that better represents the world while the statistical learning addresses the uncertainty of the data by relaxing the hard constraint in the relational domain. SRL are usually modeled using graphical model such as Markov Network (MN) or Bayesian Network (BN). While BN models causality, MN models association between two random variables, in the form of an undirected graph. The nodes in the MN are organized into cliques. A potential function is defined for each clique, which are non-negative real values for each state in each clique. The equation and an example for calculating a joint distribution is given in Figure 3. The example shows four random variables. Smoking and caner nodes form one clique while cancer, asthma and cough nodes form another clique. Suppose that we have Φ(Cancer=true, Asthma=true, Cough=true)=5.0, Φ(Smoking=true, Cancer=true, Asthma=true, Cough=true) = (4.5 * 5.0) / Z where Z is a normalizing factor that sum over all possible states. SRL has seen many applications such as relational classification (Jensen et al, 2004), Link based clustering of web search (Wang et al, 2001), link prediction in relational data (Taskar et al, 2004).



Potential functions defined over cliques

$$P(x) = \frac{1}{Z}\prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

Figure 3: An example of Markov Network (Domingos, 2008)

Khosravi & Bina (2010) identified several limitations of SRL. The biggest limitation is the complexity of inferencing because the size of the graph grows exponentially with the number of attributes and objects. Most inferencing methods are based on the standard Bayesian or MN inferencing approaches. MN's inferencing approach requires the computation of the partition function Z, which make the inferencing process NP-Complete. Most of the current researches are focusing on making the inferencing process more efficient. SRL appears to better suit a domain with low variability such as those that has many instant of data that are usually arranged in a relational database. This is due to the great challenge of structural learning in SRL. In RTS where we expect mostly unknown, large and chaotic state spaces, SLR is unsuitable for RTS.

4.2 Non Markovian (NM)

Non Markovian learning approaches do not regard or may relax the sequential order requirement of the RTS, and invalidate Markov assumption. There are many NM techniques that are capable of learning and making prediction on RTS, with varying learning capabilities. Approaches such as production system (PS), finite state machine (FSM) have no or limited learning capabilities after they are trained and deployed. Many simple event prediction approaches can be implemented based on these approaches that encapsulate domain knowledge. PSs model the domain knowledge as a set of if-then rules. We can define a rule with preconditions that describes the situation to be matched. The consequence of the rule provides the predicted atom. FSMs are similar in that given a state, it can predict the next input/events and the resultant state. These approaches rely heavily on domain knowledge, which can only be created for known environments. Even if the developers have good anticipation capabilities or foresightedness, encoding the large state space is usually prohibitive. Furthermore, the nature of chaos can only be encoded through statistical learning.

Bayesian Network (BN) is another type of NM that encapsulates domain knowledge, usually in the form of causation structure. BNs model the casual relation among the random variables in the form of a directed graph. BN are often used to interpret percept sequences, to derive at possible adversarial goals and actions by computing the posterior probabilities of goals, states, and plans, given the percept sequences (Kott & McEneaney 2006). Given a likely goal or state, other BNs can be used to compute the posterior probability of future actions. Many BNs require the structures to be predetermined and trained offline. Furthermore, when no training example is available, the conditional probability tables are based on human subjective judgments. Hence, BN can only be used if domain knowledge is available. With the large state space and chaotic nature, BN structural learning is unsuitable.

Genetic algorithms (GA) have been used to generate possible scenarios / plans based on perceived goals and situations (Kott & McEneaney 2006). For example, given the current situation and assumed goals, GA can generate the possible future events to serve as predictions. Each GA requires some fitness evaluation functions, which can be heuristics, or simulators. These evaluation functions can limit the nature of scenarios to be evaluated. Furthermore, these functions are developed for known domains. The other limitation is the assumed adversarial goals, which can be inferred using a BN, or based on subjective expert judgment. Here, GA can only be used if domain knowledge is available. While GA can search a large state space efficiently, the state space cannot be predefined for unknown and chaotic environments.

There are NM approaches that are able to continue to learn after they are deployed. These approaches may not require domain knowledge, and are able to learn from unknown situations. Inductive learning is one such approach. In Inductive learning, an agent learns a general function or a set of rules from specific input-output pairs (Russel & Norvig, 2010, Section 19.5). Inductive Logic Programming is a type of inductive learning that induces first order logic theories from examples in relational form. For example, if we have the following atoms: Father(john, caleb), Father(caleb, timothy), grandfather(john, Sheryl), we can induce a rule: $\forall x \forall y \forall z$, Father(x,y), Father(y,z) $\Leftrightarrow$ GrandFather(x, z). The main limitation is on the strict logic constraints. A rule will not be learnt if there is just one counter example. For example, the Grandfather rule is generally true. However, if there is just one case of abnormal relation in the family that contradicts the rule, that rule will be violated, and will not be induced, even though it may be true statistically. Such contradictory phenomenon is common in a chaotic world. While probabilistic inductive logic programming may seem to solve the problem, the entire ILP algorithm must be rerun for each arriving percepts. This poses a great problem because ILP is exponential in the number of predicates and constants. Hence, ILS is unsuitable for online learning in RTS.

Reinforcement-learning (Luger, 2008, Section10.7) can also be regard as a NM approach that supports online learning. In reinforcement-learning, an agent learns a set of policy for actions selection. The policy contains a set of state-action pairs with a value that describes the historical goodness of applying that action in that state. The goodness value is accumulated based on a reward or penalty function known as “reinforcement”. Reinforcement learning is not the same as RTS learning mainly because its main focus is to learn a set of policy, which involves actions taken by the agent, while RTS learning needs to predict environmental states even though they are irrelevant to the reinforcement calculation.

4.3 Discussions

Many current approaches for situation reasoning assume that domain knowledge is known. While these approaches have work well in many applications, they will fail in unknown environments. Unknown environments require agents to be robust and flexible, as well as to be able to learn and to adapt in new environments. While a learning agent is able to improve its performance, the structures of the knowledge representation are usually fixed. Structural or rule learning are usually limited and done offline due to the exponential complexity. We need structural flexibility, or multiple structures to account for the chaotic nature of the RTS.

Methods such as ILP or MSM are either logic constrained, have strict sequence requirement, or based on propositional representation. While Markov model and its variances have found many success stories, its strict sequence requirement prevent it to be used for unknown situations. Likewise, strict logic constraint does not allow ILP to predict atoms that have not been seen before. Reinforcement learning is not designed for RTS learning and prediction. Furthermore, many methods assume propositional data representation even though the relational formalism is a more natural way of representing the world of objects. While SRL may allow structural and statistical inferencing, their largely constraint topological network structures prevent them for uses in unknown and chaotic environments. Hence, these methods are hard to generalize to predict atoms that have not been seen before.

A summary of the evaluation of the current methods for RTS learning and prediction is given in Figure 4. The scores at the last column provide an indication on how suitable each method is for RTS. The higher the score, the higher it may be used for RTS. Nevertheless, since none of them achieve a full score, we need a new learning and inferencing method for RTS. We will introduce a new situation learning (SL) approach for learning a RTS in unknown environment that features structural agility in its learned knowledgebase, and allows flexible use of the knowledgebase to make predictions for unseen states.

Method	Relational Data	Unknown Environment	Huge State Space	Probabilistic Data	Online Structura l learning	Noisy Inference	Score
PS	✓	×	✓	×	×	×	2/6
FSM	×	×	✓	×	×	×	1/6
BN	×	×	×	✓	×	✓	2/6
GA	×	×	✓	×	×	×	1/6
ILP	✓	✓	✓	×	✓	×	4/6
MSM	×	✓	✓	✓	✓	×	4/6
RL	✓	✓	×	✓	×	×	3/6
OOM	×	✓	✓	✓	×	×	3/6
SRL	✓	×	×	✓	×	✓	3/6

Figure 4: A Summary of evaluating current approaches for RTS learning and prediction

5. A Situation learning Approach to RTS Learning and Prediction

The set of RTS characteristics is challenging. Many current learning methods are not designed to directly address these challenges. In this section, we describe a possible solution that shares some properties as the recent event segmentation theory (EST, Kurby & Zacks, 2008), which decompose the RTS into a set of situations. Unlike EST, the boundary of each situation has no semantic correspondence to the real world event, but is based on a temporal function. An advantage of the situation learning approach is that, it does not have event boundary, and hence avoids the high transient error rate at the event boundaries.

5.1 learning

The situation learning (Darken 2005) approach learns a RTS into a set of situations (not to be confused with the related notion of situation in situation calculus). The approach appears to use a sliding time window to identify sets of percepts called "situations". When a new percept arrives, this new percept serves as a reference and forms a situation that contains older percepts that were received and are still active within the time window from the time stamp of this new percept. This new percept becomes the predictive target atom of the situation that has just been formed. If the situation already exists in the knowledge base, the number of occurrence of this situation is incremented. Otherwise, this situation will be added into the knowledge base. Note that this is not a Markovian approach. An active atom can be received long time ago and still persist even though other later atoms have become inactive. Hence, the sequential order is lost. In fact, the sequential order may be relaxed to achieve better prediction accuracy.

Instead of learning the entire RTS with one graphical model such as a BN or MN, the approach effectively generates multiple simple networks of two layers as the time window slides through the RTS. Given a relational time series as shown in Figure 2, the agent starts with zero knowledge and forms the situations as soon as the first percept arrives as shown in Figure 5.

{}	1	(loc+ Ed road)	1
{[loc+ Ed road]}	2	(loc+ Fox1 road) (loc+ Fox2 road)	1 1
{[loc+ Ed road] [loc+ Fox1 road]}	1	(goE Fox1 east)	1
{[loc+ Ed road] [loc+ Fox1 road] [goE Fox1 east]}	1	(loc- Fox1 road)	1
{[loc+ Ed road] [loc+ Fox2 road]}	1	(goE Fox2 east)	1
{[loc+ Ed road] [loc+ Fox2 road] [goE Fox2 east]}	1	(loc- Fox2 road)	1

Figure 5: A collection of situations (left column) and their associated prediction (right column)

When the learning process starts, there is no percept. The current situation is an empty set. When a percept (loc+ Ed road) arrives, it becomes a reference point and a time window is cast in retrospect to determine which percepts are currently active in the window. Since there is no active percept, the situation that predict (loc+ Ed road) is an empty set (first row). When the second percept (loc+ Fox1 road) arrives, it becomes the next reference. Assuming we have a 5sec time window, we have a situation {[loc+ Ed road]} that predicts (loc+ Fox1 road). When the 3rd percept arrives, we have a situation of two active percepts {[loc+ Ed road] [loc+ Fox1 road]} that predicts (goE Fox1 east). When the 4th percept arrives, we have a situation of {[loc+ Ed road] [loc+ Fox1 road] [goE Fox1 east]} that predicts (loc- Fox1 road). Note that the percept (loc- Fox1 road) deactivates the percept (loc+ Fox1 road). When the 5th percept (loc + Fox2 road) arrives, the current active percepts are only{[loc+ Ed road]}, which is the same as our second situation. Hence, (loc+ Fox2 road) is added as another possible predicted percept for the situation {[loc+ Ed road]}. When the 6^{th} percept (goE Fox2 east) arrives, we have a current situation of {[loc+ Ed road] [loc+ Fox2 road]} that predict (goE Fox2 east). When the 7th percept (loc- Fox2 road) arrives, 3 active percepts form the situation {[loc+ Ed road] [loc+ Fox2 road] [goE Fox2 east]}

Formally, the situation learning approach processes the percept sequence $\{p_1 p_2 \ldots p_n\}$ into smaller disjoint set of percepts (called a situation) $\{s_i\}, i = 1..S$ where S is an integer that defines the number of situation. Let $\tau_a(p_i)$ refers to the time in which the atom p_i is active, $\max_\tau[\tau_a(p_i)]$ refers to the latest time in which p_i is active, τ_c refers to the current time when a new atom p_{new} is received and τ_w refers to the time window duration. $p_i \in s_i$ if $\max_\tau[\tau_a(p_i)] + \tau_w \geq \tau_c$. Let p_j refers to the percepts encountered after s_j. We write a consequence c_j as a tuple, $c_j = (s_j, p_j)$, such that p_j follows s_j.

5.2 Prediction

The one step prediction task is then: given a set of consequences $\{c_j\}$ and the current situation $s_c = \{p_{c+1} p_{c+2} \ldots p_{c+z}$, generate a percept p_p that represents the prediction for next future percept p_{c+z+1} . Prediction is correct if $p_p = p_{c+z+1}$. Let $a = \{0,1,2,\ldots,i\}$ be a counter that records the number of correct prediction. Prediction accuracy is $\frac{a}{n}$.

5.3 Prediction Techniques

Darken (2005) provides two simple techniques of prediction. The two techniques are Statistical Look-up Table (SLT) and Variable Matching (VM). SLT searches the situation table to look for a situation that exactly matches the current situation. If a match is found, the percept that follows the matched situation with the greatest frequency will be the predicted percept. VM replaces all constants in the atom with variables. Multiple instances of a constant use the same variable. The matching of situations becomes the problem of variable matching with substitution. A substitution is a list of variable bindings, e.g. θ={?a/?b} where variable ?a from one situation is bound to variable ?b in another situation. SUBST(θ,α) denotes the result of applying substitution θ to situation α. A match is then defined as a bijection of variables between the current situation and a match situation. Finding matches is a graph isomorphism problem. An example of the variable representation is shown in Figure 6. In both techniques, there is no prediction when there is no matched between the current situation and any situation in the situation table.

Constant	Variable
[loc+ Ed road]	[loc+ ?x ?y]
[loc+ Ed grass]	[loc+ ?x ?z]

Figure 6: Constant versus Variables Representation

The above two techniques offer some insights into other possible techniques of prediction. Given a set of consequences and a current situation, the predictive target atom can be derived by simple common inferencing techniques such as pattern matching, Bayesian network or Markov chain in conjunction with the SL. We can interpret each consequence as a network (See Figure 7). Note that the SL-Markov approach assumes that the atoms in a situation are in sequential order, even though the order is lost. These techniques provide means to generate the predictive atom given s_c and $\{c_j\}$. Without the SL approach, these three techniques will face exponential complexity in the learning and inferencing process.

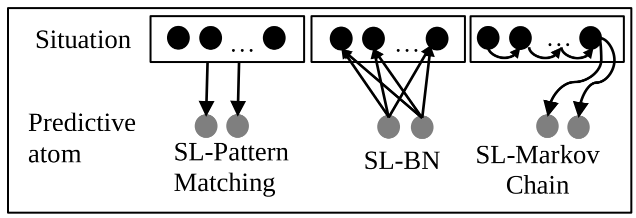


Figure 7: Possible problem formulations for prediction

These multiple simple networks avoid the current challenges in the statistical relational learning (structural learning and exponential inferencing process) by turning the problem into a situation matching and simple inferencing process. The SL approach addresses all challenging characteristics of RTS. Firstly, SL stores the relational data and allows prediction techniques to use the structure of a relational framework. For each unknown situation, SL creates a new situation-prediction tuple, and immediately uses it to predict the next atom. Each situation can accommodate any combination of atoms, regardless of how large the state space is. It manages probabilistic data by having multiple predictive target atoms. Chaos is managed by a simple creation of additional networks for new situations. It can handle noisy inferencing by allowing partial order matching.

We developed additional prediction techniques based on Variable Order Markov Models (VOMM), Multiple Simple Bayesian (MSB) network, and Simple Bayesian Mixture (SBM) in conjunction with SL. When the RTS is decomposed into a set of situations, we can build one simple Bayesian network for each situation with the predictive target atom as the parent node, effectively forming multiple simple Bayesian networks. Since MSB cannot learn certain functions such as Exclusive-OR, we implemented Simple Bayesian Mixtures. SBM contains probability mixture densities, constructed by normalizing a linear combination of two or more Simple Bayesian Networks probability densities having the same domain and range. SBM is implemented using the Estimate & Maximize (EM) algorithm. VOMM is an extension to the Markov chain models in which a variable order is used in place of a fixed order. We implemented a VOMM model using context trees (Buhlmann & Wyner 1999).

6. Experiments and Results

We compared the prediction performance in a benchmark environment that is used in Darken (2005) in which, an agent wanders around and performs actions randomly. Actions include "go eastward", "pick up weapon", "equipped weapon", "hit", and many more. There are other agents (monsters) in the environment such as goblins, trolls and dragons. There are three types of weapon: pitchfork, dagger and sword. Each weapon may be more effective against each type of monsters. Each time a monster is killed, it will leave behind a weapon. Each monster, weapon, agent and location has a unique name constant. The sequence of percepts describes what the agent sees, such as its location, weapons, and monsters. Our prediction task is to predict the next percept the agent may see, given the past percept sequence. Darken (2005) tested the prediction performance by running the algorithms through more than 250,000 percepts. In this study, we want to know how the SL-prediction techniques work in harsh and new environments. We clear off the memory after 100 percepts have been processed and examine the results after 40 batches of 100 percepts are processed. To simulate noisy environment, we randomly swapped the order of two atoms in the current situation. All experiments were run on a Dell XPS Laptop i7 1.87Ghz 16GB RAM with Windows 7.

The prediction accuracies are given in Figure 8. Each bar in the chart represents the mean prediction accuracy with its associated standard error of a predictor. From the standard error indicators, we can see that the differences are significant for at least at alpha $\alpha = 0.05$ for a statistical student-T test with degree of freedom df=39. There is no significant difference between the SLT and VM, and both techniques are significantly worse off than the SL-Bayesian and SL-VOMM techniques. This is due to the strict requirement of exact matching. When the environment is unknown and noisy, the current situation can hardly match the learned situations in the memory. Figure 9 shows the mean number of no-match for 40 batches of 100 percepts. No-match occurs when the algorithm is unable to find a reasonable situation. SL-VOMM handles the no-match problem by varying the order of Markov Model. The SL-Bayesian techniques handle the problem using Laplace distribution.

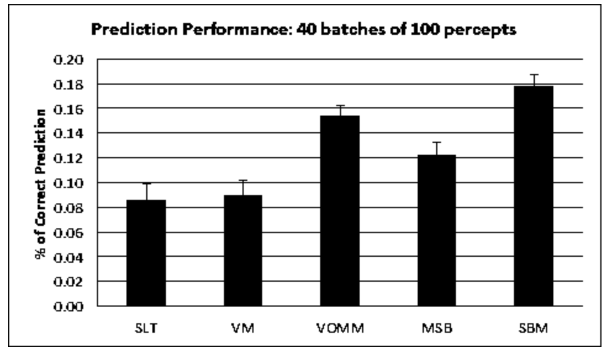


Figure 8 Comparison of Prediction Accuracy for prediction techniques in conjunction with situation learning.

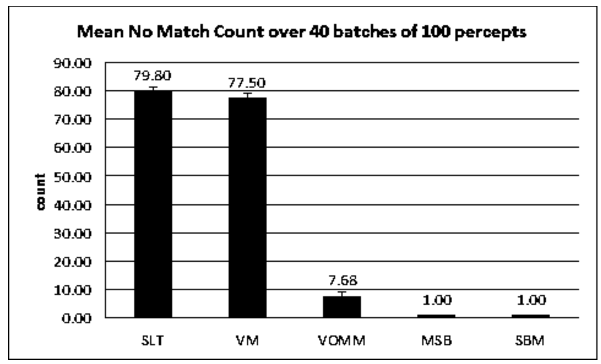


Figure 9 Comparisons of No-Match for prediction techniques in conjunction with situation learning.

The VOMM is a popular approach in sequential and online learning, and handles novel situations better than SL-SLT and SL-VM. While the SL-VOMM does not require exact atom to atom matching, and even allow partial matching, it requires exact sequential adjacency ordering. For example, the sequence of words [The Blue Fish is eating] will not match the sequence [The Fish is eating]. In addition, SL-VOMM treats each atom as a proposition. The multiple simple Bayesian network is able to handle novel situations with the Laplace method of assigning probabilities to newly encountered atoms. Its performance is limited for several reasons. Firstly, there are too many novel percepts. The prior probability for each percept can be very low. The Laplace method assigns a probability that can be unfairly large to new atoms. Secondly, Bayesian network cannot handle exclusive-OR relation. There are atoms that are mutually exclusive. Thirdly, atoms in the sequence are not independent and identically distributed. The SL-SBM performs better than the SL-MSB. However, it also suffers some of the limitations found in SL-MSB. Nevertheless, the purpose of this experiment is to demonstrate the robustness of the SL approach. After a RTS has been decomposed into a set of situation, we can apply different kind of prediction techniques in the inferencing process. One surprising finding in this study is that, non Markovian techniques can perform better than the Markovian one, even though the Markovian techniques are the popular techniques for sequence learning and prediction.

7. Conclusion

Prediction tasks play an important role in agent cognition processes such as planning and decision-making. However, many current prediction approaches assume that we have domain knowledge. To improve the predictive power in unknown domain, this paper suggests a situation learning approach to learn a RTS. This approach makes possible the use of pattern matching, Bayesian network and VOMM as techniques for prediction. Initial implementations have produced encouraging results. With improved predictive power, it may be possible to develop multi-step event prediction to look for events of interest and to quantify their likelihoods. The challenging benchmark environment consists of too many novel situations for SL-LUT, SL-VM, SL-VOMM, SL-MSB and SL-SBM. Any algorithm that attempts to excel in novel situation prediction may have to possess properties of human creativity. At this point, SL-SBM appears to be the best performer.

For future work, we will explore the theory of Cognitive Integration, also known as Conceptual Blending (Fauconnier and Turner 2002). This theory explains the human creative process, which may help to improve the prediction accuracy in unknown and chaotic environments.

8. Acknowledgement

This research effort is partially funded by the Temasek Defence Systems Institute, a strategic alliance between National University of Singapore and US Naval Postgraduate School. http://www.tdsi.nus.edu.sg/

9. References

Darken, C. (2005). Towards Learned Anticipation in Complex Stochastic Environments. In Proc. Artificial Intelligence for Interactive Digital Entertainment 2005, 27-32

Domingos, P. (2008). Markov Logic: A Unifying Language for Information & Knowledge Management. Retrieved from the web: Nov. 19, 2011, http://videolectures.net/cikm08_domingos_mlmaul/

Fauconnier, G., and Turner, M. (2004). The Way We Think: Conceptual Blending and The Mind's Hidden Complexities. New York: Basic Books

Getoor, L., and Taskar, B. (2007). Introduction to Statistical Relational Learning. MIT Press

Jaeger, H., Zhao, M., and Kolling, A. (2005). Efficient estimation of OOMs. In Proc. of NIPS.

Jensen, D., Neville, J., and Gallagher, B. (2004). Why collective inference improves relational classification. In Proc of the 10th ACM SIGKDD International Conf on Knowledge Discovery and Data Mining, pp. 593–598

Khosrav, H., and Bina, B. (2010). A Survey on Statistical Relational Learning. In Proc of Canadian Conference on AI'2010. pp.256~268

Klein, G. (1999). Sources of Power: How People Make Decisions. Cambridge MA, MIT Press.

Kott A., and McEneaney, W. (2006). Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind. Chapman and Hall/CRC

Kunde, D. and Darken, C. (2006). A Mental Simulation-based Decision-Making Architecture Applied to Ground Combat. In Proc of BRIMS 2006.

Kurby, C. A. and Zacks, J. M. (2008). Segmentation in the perception and memory of events. Trends in Cognitive Sciences Vol.12 No.2

Luger, G., (2008). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th Edition), Addison Wesley,

Spanczer, I. (2007). Observable Operator Models. Austrian Journal of Statistics Volume 36 (2007), Number 1, 41–52

Sun, R. and Giles, C.L., (2001). Sequence Learning: From Recognition and Prediction to Sequential Decision. Making, IEEE intelligent systems

Taskar, B., Wong, M., Abbeel, P., and Koller, D. (2004). Link prediction in relational data. Proc of Neural Information Processing Systems, 659-666

Wang, Y., Wang, Y., and Kitsuregawa, M. (2001). Link based clustering of web search results. LNCS, pp. 225–236. Springer, Heidelberg

Author Biographies

TERENCE KIAN-MOH TAN is a PhD candidate at the MOVES Institute, Naval Postgraduate School. He is also a senior member of technical staff at DSO National Laboratories in Singapore.

CHRISTIAN DARKEN is an Associate Professor of Computer Science at the NPS, and an academic faculty member of the MOVES Institute. His research focus is on human behavior simulations. Previously he was Project Manager at Siemens Corporate Research. He was also the AI programmer on the team that built Meridian 59, the first 3D massively-multiplayer game. He received his Ph.D. from Yale University in 1993.

MODELING SOCIALLY TRANSMITTED AFFORDANCES: A COMPUTATIONAL MODEL OF BEHAVIORAL ADOPTION TESTED AGAINST ARCHIVAL DATA FROM THE STANFORD PRISON EXPERIMENT

Benjamin D. Nye, Ph.D.
University of Pennsylvania
Ackoff Center for Advancement of Systems Approaches
120B Hayden Hall
3320 Smith Walk
Philadelphia, PA 19104
benjamid@seas.upenn.edu

ABSTRACT: *Social learning and adoption of new affordances govern the rise of new a variety of behaviors: from actions as mundane as dance steps to those as dangerous as new ways to make IED detonators. Traditional diffusion models and social network structures fail to adequately explain who would be likely to imitate new behavior and why some agents adopt the behavior while others do not. To address this gap, a cognitive model was designed that represents well-known socio-cognitive factors of attention, social influence, and motivation that influence learning and adoption of new behavior. This model was implemented in the PMFServ agent-based cognitive architecture, enabling the creation of simulations where affordances spread memetically through cognitive mechanisms. To examine the effectiveness of this model, its performance was tested against data from the Stanford Prison Experiment collected from the Archives of the History of American Psychology.*

Keywords:
Social learning, affordances, cognitive modeling, agent-based modeling, social networks, social influence, attention, imitation, memes

1 Introduction

Social transmission of affordances is an important mechanism for cultural shifts. In perceptual psychology, an affordance represents the potential for an action (Gibson, 1979). As such, affordance learning directs a significant amount of human experience. For example, learning about the opportunity to buy the latest gadget is a common experience in Western culture. However, not every person learns about every gadget and clearly not every person uses every gadget. Attention, retention, motivation, and action are all limited resources that limit the spread of new behavior (Bandura, 1986).

Despite the discovery of well-established cognitive factors that moderate the spread of new behavior, these factors are not represented in traditional models of adoption of behavior. The most prevalent models for the spread of new behavior are diffusion models and social network models, which each have significant limitations. Systems-dynamics diffusion models capture the *rate* of adoption, but lack detail on *where* the adoption spread (Rogers, 1995).

Social network simulations model *where* adoption spreads through nodes, but often only represent the structure of network connections with no inherent agency of nodes (e.g. no individual differences). Even among network simulations that employ cognitive mechanisms, the network agents often have minimal depth. As such, these models are poorly suited to modeling networks where individual differences moderate the adoption of new behavior. This means that social network simulations model *where*, but cannot model *who* adopted or did not adopt a new behavior.

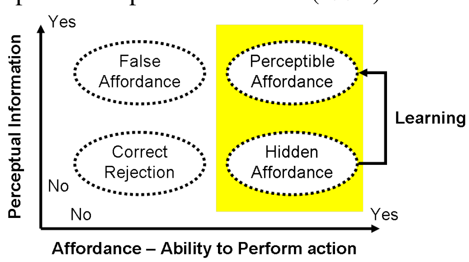
To examine *who* learns and adopts new behavior requires cognitive agents connected through one or more social network layers. Who learns and adopts certain behaviors can be critically important: adopters are often of unequal importance and unequal influence on later adoption. For example, safe sex interventions have a greater impact if they are adopted by more promiscuous individuals. Who will imitate an adopter is also important. Network theory approaches this issue by considering metrics such as centrality (Vilpponen, Winter, & Sundqvist, 2006). However, without cognitive agents, an individual with high betweenness could just as easily be an outcast as a gatekeeper.

To examine this deeper level of analysis, a cognitive model was developed that implements mechanisms that correspond to each process of Observational Learning from Social Cognitive Theory (Bandura, 1986). These cognitive mechanisms were implemented as models in the PMFServ socio-cognitive agent-based framework (Silverman, Johns, Cornwell, & O'Brien, 2006). A cognitive model was implemented which allowed PMFServ agents to socially learn behavior and adopt new behaviors if sufficiently motivated. To examine the effectiveness of the model's ability to represent who learns and adopts behavior, a PMFServ scenario was created that modeled the Stanford Prison Experiment.

2 Modeling Affordance Transmission

The goal of this cognitive model is to represent socially transmitted affordances. The ecological approach to perception posits that the environment is perceived in terms of the affordances that it offers, referred to as direct perception. Affordances always exist: they represent the potential for action (Gibson, 1986). For example, a human has the affordance to swing a hammer. A goldfish does not have this affordance, as it has no hands.

Figure 1: Relationship Between Affordances and Perception. Adapted from Gaver (1991)



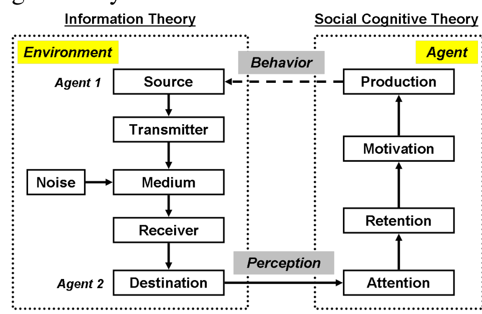
Affordances are not always known, however. As shown in Fig. 1, Gaver (1991) framed this issue using two orthogonal aspects: 1. Is an affordance available? and 2. Is the affordance perceptible? By learning an affordance, an agent moves from having a hidden affordance to having a perceptible affordance (known affordance). In this way, an agent becomes aware of a new action opportunity. Social learning of affordances is important because the space of possible actions for human interaction is vast. Learning by observation greatly reduces this space, exposing an agent to the affordance by demonstration.

3 Affordance Transmission as a Meme

For this model, affordance transmission is framed as a type of meme. A meme is a unit of cultural information that spreads by repeated reproduction from one agent to another (Dennett, 1995). A model for meme transmission was synthesized from Bandura's Social Learning Theory and Shannon's Information Theory as shown in Fig. 2 (Bandura, 1986; Shannon, 1948). These theories provide complementary processes for examining the flow of information between and within individuals, respectively.

The Social Cognitive Theory establishes the necessary stages for an agent to repeat socially learned behavior: attention to the behavior, retention of the affordance, motivation to repeat the behavior, and physical production of the behavior (Bandura, 1986). However, Social Cognitive Theory offers little insight for the transmission of information through the environment. Information Theory addresses transmission through an environment explicitly, where a source transmits through a medium to a receiver to reach a destination.

Figure 2: Systems Model for Meme Transmission



This framework offers a comprehensive view of meme transmission in terms of agents sharing a common environment. It is particularly well-suited to modeling the spread of socially learned affordances, as the information of an affordance directly corresponds to behavior.

4 Cognitive Model Design

Based on this systems model for affordance transmission, a cognitive model was created using the PMFServ cognitive architecture. PMFServ is a socio-cognitive agent-based architecture. This architecture implements cognition using a model-of-models approach: integrating best-of-breed social science models and performance moderator functions (PMFs) to form a cognitive model (Silverman et al., 2012). PMFServ has a long track record for modeling decision-making and has been used for crowd simulation (Silverman et al., 2006), leader simulation (Silverman & Bharathy, 2005), and country stability simulations that had an accuracy of over 85% (O'Brien, 2010).

An attractive feature of the PMFServ framework is that agents employ affordance-based perception (Silverman et al., 2006). A PMFServ agent parses objects in its environment for the available affordances and evaluates these actions based on a set of "activations" in the given context. The agent's cognitive model weighs these activations based on a weighted tree of Goals, Standards, and Preferences (GSP Tree) to estimate the emotions engendered by taking that action. These emotions are then used to calculate the expected subjective emotional utility

(SEU) for a possible decision. This allows the PMFServ cognitive model to determine their level of motivation to perform an action, even for a new affordance they encounter.

However, PMFServ's standard agent perceives all of the affordances of its environment and lacks any cognitive mechanisms for managing attention and retention of new affordances. To simulate affordance transmission, significant additions to the PMFServ model base were required. The following section discusses the theories implemented as models, how these theories interact with existing PMFServ models, and how these models help model affordance transmission.

4.1 Attention Cues

An attention model was designed to selectively filter action events that an agent observes. Any action involves an actor (source), behavior (action), and some outcomes (results). The cognitive mechanisms influencing attention were chosen for their ability to capture cues about *who* (source) and *what* (action) is salient. Preference was also given to mechanisms with high validity and support in literature. Theories of attention and persuasion both indicate that attentional salience is influenced by central and peripheral cues (Treisman & Gelade, 1980; Petty & Cacioppo, 1986). For an affordance, central information includes direct information about the associated behavior. These include whether an agent can perform the observed action or if the action resulted in appealing outcomes. These influences are known as transferability (Bandura, 1986) and motivated attention (Fazio, Roskos-Ewoldsen, & Powell, 1994), respectively.

However, peripheral cues can be equally or more important for directing attention. From a social network view, social influence must be considered important peripheral cues. Social influence is commonly implemented for social network simulation, but is often represented as some intrinsic agent property. The problem with this approach is that social influence is a multi-faceted, relational construct. To address this issue, social influence was represented by implementing multiple established theories of social influence. This section discusses the factors modeled as cues for attention and concludes with the attention model that integrates these cues.

4.1.1 Novelty (Central)

The three central cues modeled were novelty, motivated attention to outcomes, and transferability. Novelty indicates how "new" a stimulus appears (James, 1890). Novelty decreases with respect to the number of prior exposures stored (Johnston, Hawley, Plewe, Elliott, & DeWitt, 1990). To model this, novelty is calculated as a function of an agent's familiarity each action and agent present in an event. The novelty model calculates this based on familiarity levels from memory model, which will be described later in Section 4.3.

For any given event, the novelty is calculated as the root-mean-squared of the familiarity values of the actor of the event and the action of the event. The novelty calculation for an event is shown in Eqn. 1, where f_{Actor} is the familiarity of the event's actor and f_{Action} is the familiarity of the event's action according to the memory model.

$$Novelty(Event) = \sqrt{0.5((1-f_{Actor})^2 + (1-f_{Action})^2)} \quad (1)$$

This representation was chosen because it allows a high degree of novelty if either component is novel. This dynamic was chosen because it allows representation of processes such as dishabituation, where adding an additional stimulus can restore responding to a habituated (familiar) stimulus. In this context, the response of interest is active attention. This implementation allows a return to novelty when a highly familiar person suddenly engages in a totally new action. Conversely, if a straight average was used, then a completely familiar person could be at most 50% novel. Alternatively, taking the maximum novelty component would give no extra credit for a new person taking a new action. A root mean square parsimoniously represents these important dynamics within the simulation.

4.1.2 Motivated Attention (Central)

Motivated attention refers to the tendency of humans to pay more attention to objects or events that are relevant to their goals or needs (Fazio et al., 1994). For example, a hungry person is more likely to notice someone eating. Motivational cues are handled by allowing agents to analyze the outcomes of events that occur in their presence.

PMFServ's core cognitive models evaluate their potential actions based upon "activations" that determine the attractiveness of that action, as mediated by their values and beliefs (Silverman et al., 2006). To calculate a factor for motivated attention, an agent processes an event that results from some other agent's action. In processing this event, the agent calculates their own subjective emotional utility (SEU) as if had they been the actor in that event and the outcomes were the same.

Eqn. 2 displays the central motivated attention calculation for an agent observing a given event

(Note: the 'sgn' symbol represents the sign function, producing -1 for negative values and 1 otherwise). SEU_{Event} represents the subjective expected utility of activations that the perceiving agent would receive had they been the actor in that event and the outcomes were the same. An adjustment to the raw utility rescales the value from utility's range of [-1,1] to [0,1]. The second rescaling factor takes the fourth root of the absolute SEU value. This factor was introduced during model calibration to adjust for the small range over which SEU realistically operates within PMFServ.

$$MotivatedAttention(Event) = 0.5 * (1 + sgn(SEU_{Event}) * (|SEU_{Event}|^{0.25})) \tag{2}$$

4.1.3 Transferability (Central)

The third central cue modeled was transferability. Transferability influence refers to the additional influence conferred by an agent who has similar capabilities and does actions that one could imitate. Often, this trait is studied in children at different developmental stages. Children have a preference to attend to and imitate those of similar ability level on tasks (Bandura, 1986).

The transferability influence model allows agents to process an observed event and determine if they could do the same action at the current time. This determination is only based upon the agent's current affordances at the particular moment, not any past or potential affordances. This implementation has the advantage of easily classifying events into those which they could imitate (Transferability=1) and those that they could not (Transferability=0).

4.1.4 Authority (Peripheral)

Six peripheral cues were also incorporated into the model, representing social cues. The authority influence model represents the additional influence conferred by a position of authority. The effects of authority on behavior have been well documented by Milgram (2004) and Mantell (1971). PMFServ represents the authority of agents within their respective groups (Silverman et al., 2006). Since this factor is already represented, the authority submodel wraps this factor for use as a social cue.

4.1.5 Conformity (Peripheral)

The conformity model has its theoretical roots in the seminal work done by Asch (1955). Later work by Tanford and Penrod (1984) proposed the Social Information Model (SIM), a probabilistic conformity influence function. The Tanford and Penrod (1984) analysis produced a curve as stated in Eqn. 3, where S is the number of conforming sources and T is the total number of non-conforming targets.

$$ConformityInfluence(S,T) = e^{-4*e^{\frac{-S^{1.75}}{T}}} \tag{3}$$

The implemented conformity model uses this equation verbatim. However, the context of its usage is slightly different than that of the original SIM model. While that model assumed a set of confederates, these models assume agents act based upon their own opinions but still exert influence. As such, any set of agents engaged in a particular activity forms a group of influence sources (S). The remaining agents involved in other activities are the target group (T). As such, agents can calculate the conformity influence of any activity in the simulation for any given action occurring at the time.

4.1.6 Similarity (Peripheral)

The similarity model calculates a social influence factor based upon how much an agent feels it has in common with another agent. The influence of similarity on attention and influence has been an influential topic in the domains of social psychology and social network analysis (Platow et al., 2005). PMFServ contains a model that estimates a proxy for similarity, known as GSP congruence (Silverman et al., 2006). GSP congruence is calculated by transforming agents' GSP trees vectors of normalized linear weights and calculating the distance between these vectors. The standard GSP congruence function is shown in Eqn. 4, where $\overrightarrow{w}$ is the perceiving agent's GSP vector, $\overrightarrow{w^*}$ is the observed agent's GSP vector, and N is the number of elements in $\overrightarrow{w}$.

$$GSPCongruence(\overrightarrow{w}, \overrightarrow{w^*}) = \frac{\sum_{i=1}^{N}(\overrightarrow{w_i} - \overrightarrow{w_i^*})^2}{\sum_{i=1}^{N}(\overrightarrow{w_i})^2 + (\overrightarrow{w_i^*})^2} \tag{4}$$

4.1.7 Valence (Peripheral)

Valence influence is caused by general like or dislike of another person. This is related to the "halo effect," whereby an attractive person appears more competent (Kelley, 1955). Experiments such as Hilmert, Kulik, and Christenfeld (2006) have experimentally shown that valence affects social influence. Since PMFServ already has a model for maintaining valence, valence is exposed as a cue for attention. Since valence ranges from [-1,1] in PMFServ and all cues are fitted into a range of [0,1], a small transform is applied to valence values to rescale and shift it into the appropriate range.

4.1.8 In-Group (Peripheral)

The in-group influence model represents the social influence based on membership in a mutual group or clique (Tajfel, 1982). PMFServ has a structure for representing group membership, which allows members to be part of a group. This cue determines if agents share a common group (ingroup=1) or share no common groups (ingroup=0).

4.1.9 Reference Group (Peripheral)

Reference group influence represents the influence based on an agent belonging to a group against which an agent compares themself, such as a desirable group (Kameda, Ohtsubo, & Takezawa, 1997). PMFServ has an analogous factor within its model set that is an agent's "internal membership" with a group (Silverman et al., 2006). Internal membership measures how much an agent desires to participate and support a group. As this measure is explained in Silverman, Bharathy, Nye, and Eidelson (2007), it will not be covered in detail here.

Reference group influence uses a variant of PMFServ internal membership that has been scaled to fit into a range of [0,1]. This model can report back the desire to belong in any given agent's group (if they belong to a group). This value can be independent of in-group influence, since people are not always a member of their preferred group.

4.1.10 Selective Attention

Selective attention is a construct that refers to the additional probability of perceiving events performed on an object that an agent actively perceives, as opposed to other peripheral events (Simons & Chabris, 1999). Selective attention is implemented by having agents keep a record of the objects and agents they are actively attending to at the current time. PMFServ agents are able to actively take actions on other agents, including actions of active perception (watching).

$$SelectiveAttention(x) = \begin{cases} \frac{1}{N} & \text{if } x \in X_{Targeted} \\ 0 & \text{if } x \notin X_{Targeted} \end{cases} \tag{5}$$

As such, the selective attention model records all entities that an agent is currently engaged in action upon. This means that selective attention is focused on any targets being watched or acted upon by an agent. Selective attention is spread uniformly across these targets as noted in Eqn. 5. This allows agents to choose who will be the target of their selective attention, as is observed in the cocktail party effect (Cherry, 1953).

4.2 Attention Mechanism

Based on these cues for attention, an attention model was developed. This model corresponds to a series of winner-take-all competitions for attention between simultaneous events, a process which has some support in neurological research (Lee, Itti, Koch, & Braun, 1999). Attentional salience determines the probability that an agent will attend to an event. This is accomplished by first calculating a salience for each event occurring during a time step. An additional salience term exists to represent inattention salience: the salience of background events not simulated that might be attended to instead of the simulated events. This vector of saliences is normalized to form a probability vector, from which a finite number of events are chosen. Each event is chosen without replacement, except for inattention, which always remains an option.

The algorithm for drawing the set of attended events is displayed as Fig. 3, where N is the maximum simultaneous events attended, E is the set of all current observable events, E_{Att} is the set of currently attended events, and $X(E, E_{Att})$ is a random variable returning at most one unattended event from the set $E \backslash E_{Att}$. The output of this algorithm is E_{Att}, the total set of attended events. If $X(E, E_{Att})$ returns no event, this represents inattention and one less total event will be attended.

Figure 3: Attention Algorithm

E_{Att} = { }
for $i = 1$ to N **do**
  ATTENDED_EVENT = X(E, E_{Att})
  if ATTENDED_EVENT != No Event Attended **then**
    E_{Att} = $E_{Att} \cup$ { ATTENDED_EVENT}
  end if
end for

The probability that an event (e) receives enough attention to be processed cognitively is determined by the distribution of $X(E, E_{Att})$ and will be referred to as $P(e, E, E_{Att})$. The probability distribution for choosing an event to attend is shown in Eqn. 6, where E is the set of all simultaneously observable events, E_{Att} is the set of events already attended to, s_e is the salience of an individual event e, and s_I is the inattention salience. Events with higher salience are more likely to be selected, as they fill a greater fraction of the probability vector.

$$P(e, E, E_{Att}) = \begin{cases} \frac{s_e}{s_I + \sum_{e \in E \setminus E_{Att}} s_e} & \text{if } e \in (E \setminus E_{Att}) \\ \frac{s_I}{s_I + \sum_{e \in E} s_e} & \text{No Event Attended} \\ 0 & \text{if } e \in E_{Att} \end{cases} \tag{6}$$

Attentional salience is calculated as a function of attentional cues previously defined. Each parameter is combined using a linear weighted sum, where the weight of a cue determines its contribution to the event salience. Since the relative strengths of these factors are not well-studied, "best guess" weights were calculated from their observed effect on either attention, perception, or retention. A linear sum was chosen based on the KISS principle, as it was the simplest way to combine cues into a total salience (Axelrod, 1997). While there are good reasons to believe that some of these factors interact, psychology literature has not yet produced the studies that demonstrate how these factors interact.

4.3 Retention Mechanisms

Since this cognitive model was primarily intended to address the issue of "who" learns and adopts new affordances, the memory model was kept as simple as possible. Many affordances of interest are relatively simple and memory effects are not the main barrier to adoption. As such, memory was implemented as a simple associative structure. Associative memory works by strengthening connections between elements, stimuli, or constructs due to repeated pairing (Mackintosh, 1983).

This information is used for two purposes. Firstly, this memory model supports affordance learning. Once an action in stored in the agent's memory, the affordance for that action becomes known. As such, attending to an event with a new behavior will let the agent learn this behavior. Secondly, the model is used to calculate familiarity because this is needed to determine the novelty of events.

$$Familiarity(Entity) = 1 - e^{-r_f * N_E} \tag{7}$$

The familiarity equation is stated in Eqn. 7. The input to the equation, $Entity$, is an action, agent, or other entity contained within a learned pattern. N_E is the number of exposures to that entity and r_f is a familiarity rate that determines the steepness of the curve. Within the current implementation, r_f was set to 0.2 as this allows familiarity to reach 95% after 15 exposures. Empirical research indicates that the exposure effect hits its maximum after between 10 and 20 exposures, so this seemed to be a reasonable familiarity rate (Bornstein, 1989).

4.4 Motivation Mechanisms

Motivation to perform an action is handled using PMFServ's decision model. As PMFServ's decision model has undergone over ten years of development, fully understanding these processes requires careful reading of a number of prior papers (Silverman, Bharathy, Johns, et al., 2007). From the standpoint of affordance adoption, the most important model is the Goals, Standards, and Preferences (GSP) model that stores an agent's personality. This model captures individual differences between agents and determines the outcomes they prefer. These outcomes are represented as activations on GSP nodes. For example, gaining money will create positive activations for a "materialism" preference. An action that results in pain for the agent will give negative activations for a "safety" goal. When agents are modeled using different GSP weights, they tend toward different types of behavior.

To model realistic scenarios, GSP weights are estimated through a knowledge engineering process. In prior experiments, these GSP models have been used to represent leader personalities and examine the impact of different types of leaders on government opposition (Silverman & Bharathy, 2005). As such, these models are capable of representing tendencies down to the individual level rather than simple archetypes. In addition to the GSP model, PMFServ also contains a Physiology model representing hunger and fatigue, whose levels can impact motivation to take certain actions by linking those levels to activations (Silverman, 2004).

4.5 Production Mechanisms

Production mechanisms in PMFServ are represented by the actions associated with affordances. These actions depend on the specific scenario and generate observable events when they occur. The ability to perform an action requires a valid affordance for that action in the environment. As such, the ability to produce an action is atomic – an agent is either able or unable to perform an action. Due to how the memory model is implemented, agents are unable to perform an action unless they are aware of its affordance. This makes intuitive sense, as an agent cannot initiate an action without recognizing the possibility of performing that action (i.e. the affordance).

5 Stanford Prison Scenario

The model was tested on a simulation based on the Stanford Prison Experiment. The Stanford Prison Experiment (SPE) was chosen due to the breadth of data collected, which facilitated the development and validation of the agent-based model. Data to develop this model was collected from the Archives of the History of American Psychology (AHAP).

5.1 Data Sources

The Stanford Prison Experiment was conducted in 1971 and was intended to explore of the impact that assigned roles had on behavior inside a simulated prison environment (Haney, Banks, & Zimbardo, 1973a). In the experiment, 24 subjects were selected and randomly assigned to be prisoners or guards. The experiment, intended to last two weeks, lasted only 6 days due to the growing abusiveness of the guards and signs of distress among the prisoners.

Table 1: SPE Archival Data

Data Source	Use For Simulation
Comrey Personality Inventory	8 factor personality inventory
F-Scale	Measure of authoritarianism
Mach Test	Measure of machiavellianism
Mood Adjective Checklist	Measure of positive and negative affect
Action Frequencies	Frequencies of actions occurring (coded video)
Hour By Hour Logs	List of recorded events, with approximate times

The data extracted from the archives included qualitative and quantitative information. Table 1 displays the types of data available from the Stanford Prison experiment. Further information on this data is contained in Nye (2011). Personality trait information is available through the Comrey Personality Inventory (Comrey, 2008), the F-Scale (Adomo, Frenkel-Brunswik, Levinson, & Sanford, 1950), and the Mach test (Christie & Geis, 1970). The Comrey inventory measures traits: Trustworthiness, Orderliness, Conformity, Activity, Stability, Extroversion, Masculinity, Belonging, and Empathy. Emotional trends were captured using a Mood Adjective Checklist (Haney et al., 1973a). Action incidence data was available as frequencies coded from video tapes and through hour by hour logs listing notable activities (e.g. prisoner resistance). In addition to these data sources within the AHAP holdings, the published results from Haney et al. (1973a), Haney, Banks, and Zimbardo (1973b), and Zimbardo (2007) based on the experiment were examined closely.

5.2 Scenario Design

The Stanford Prison Experiment simulation included 9 prisoner agents and 10 guard agents. The other subjects were alternates that were either not used or were only temporarily on-site and whose activities were not well-documented. Given that their short duration and late addition, their impact on the experiment was likely be minimal.

Agents were simulated using a PMFServ cognitive model as described earlier. Prisoners were added as members to one group, while guards composed a second group. All agents started the simulation with neutral valence toward each other, equal physiological states (low hunger, low fatigue), and equal authority within their respective group (0 for prisoners, 0.25 for guards). This meant that agents initially differed entirely as a result of their personalities, their group assignment, the time they entered the experiment, and their shift (for guards).

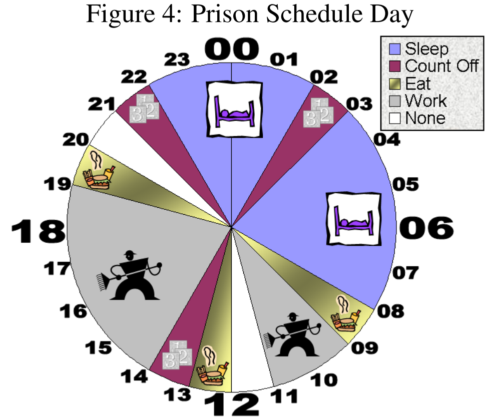
5.2.1 Personality Modeling

GSP personality models were initialized based upon the personality trait data from the Comrey Personality traits, the F-Scale, and the Mach test. While the PMFServ GSP personality model can be set up to use these factors directly, a previously validated GSP tree structure was used instead. This existing personality structure is described in Silverman and Bharathy (2005) and utilizes personality traits intended to correlate with behavior. The measured personality traits from the Stanford Prison Experiment data were used to estimate these GSP weights. This process ensured that each cognitive agent had a personality based on measures administered to a specific study participant prior to the study. Due to space considerations, the full mapping process is not described here but is available in Nye (2011).

5.2.2 Actions Modeled

Two types of actions were modeled in the Stanford Prison Experiment: interpersonal and baseline. Interpersonal actions were the most important to model because the Stanford Prison Experiment recorded action frequencies for these types of actions. The interpersonal actions with measured frequencies were: Command (order from a guard), Help, Information, Insult, Resist, Threaten, Use of Instruments (brandish baton). Interpersonal actions noted in hourly logs included Resist (prisoner resistance) and Throw in Hole (guard putting prisoner in a storage room).

Figure 4: Prison Schedule Day



Baseline activities corresponded with the scheduled activities in the Stanford Prison Experiment: eating, sleeping, counting off, and working (Zimbardo, 2007). These provided a backdrop for interpersonal actions. If prisoners are performing these activities during their assigned periods, guards have less incentive to harass them. Conversely, if prisoners dislike a particular activity they will be more likely to perform other actions such as resisting the guards. This captures a contextual impact on behavioral spread. The appropriate baseline activity at any given time was determined by the prison schedule, as shown in Fig. 4.

5.2.3 Experimental Conditions

Three experimental conditions were simulated to represent alternate hypotheses for the causes of abuses within the prison. Fromm (1973) and others have suggested that the since the guards were not uniformly cruel, individual factors were still a major driving force for abuses. This hypothesis posits that everyone knew abusive behavior (Full Knowledge). It has also been suggested that the orientation given to guards encouraged prisoner abuses (Reicher & Haslam, 2006). This is the Authority hypothesis, that the experimenters made subjects aware of certain affordances. A third hypothesis is that some guards were innovators of abuse and were imitated by other guards (Meme Hypothesis). For example, a document in the archives entitled “Remarks” asks, “Why did S_20 imitate John Wayne rather than S_15?” (names de-identified).

These conditions differed due to the distribution of guards who knew the Throw In Hole behavior and prisoners who knew the Resist behavior initially. The Throw In Hole action occurred when a guard used a supply closet as “the hole” to put a prisoner in isolation. Resist occurred when a prisoner openly defied guard authority and orders. In the Full Knowledge condition, all agents knew their group’s special action at the start of the experiment. In the Authority condition, the simulated agents were primed with a single exposure to the behavior from an “Experimenter” agent with high authority as a cue for attention. In the Meme Hypothesis condition, particular agents were selected as innovators based on reports from the study. These potential innovators were why these behaviors were studied.

Throw In Hole was chosen because it showed evidence of a clear early adopter who was referred to as John Wayne due to his high level of insults and cruelty. Despite John Wayne not arriving until the second shift in the experiment, the supply closet was not used as “the hole” until his arrival. John Wayne (S_13) was used as the innovator for the Meme Hypothesis condition.

Resistance was chosen because it was studied explicitly within the experiment and S_05 was a clear resistance leader to start the experiment. This resulted in a general outbreak of resistance, which was eventually subdued. S_00, a late arrival, appeared to independently have an awareness of passive resistance strategies also. As such, S_05 and S_00 were used as innovators for the Meme Hypothesis condition.

6 Stanford Prison Simulation

The model was first validated using a train and test paradigm. Training was performed only once, but three separate tests were conducted using this base simulation. For testing, three experimental conditions were simulated: Full Knowledge, Authority, and Meme Hypothesis. Each condition was simulated for 30 separate runs, where each run was 693 steps (6 days broken into ten-minute intervals). Every agent was able to take exactly one action during a ten minute step.

6.1 Train: Activation Calibration

Training was required to establish the values for the activations of actions. Training was conducted under the Full Knowledge condition, as not to give any insight or advantage to the model for simulating affordance learning. Ideally training would be automated, but action frequencies were the only data used for training the simulation. Since this consists of only a handful of data points, the data was too sparse and activations had to be calibrated by hand.

Activations were calibrated by simulating the first 20 hours of the six day experiment repeatedly and calculating the frequency that actions occurred during each scheduled activity. The intention of this calibration phase was to ensure that actions occurred with the appropriate relative frequencies with respect to each other. The tuning script generated a report listing the frequency that each action occurred. This report was compared against the expected relative frequency for each action with associated frequency data. Since activations are universal across all agents, this calibration adjusts the relative frequency of actions without significant insight into which agents tend toward which actions.

6.2 Test: Validity Measures

Three types of validation measures were applied to each condition: action ordering, action frequency, and emotional states. Action ordering validation examines the first time that each agent performs an action for the first time. These orderings are compared against orderings taken from the Hour by Hour Logs from the archives. This offers a useful proxy for measuring social learning and adoption of new behavior. Action frequency validation examines how well the actions over the full simulation correspond to the Action Frequencies from the

Stanford Prison study archival data. Emotional state validation refers to the correspondence of the PMFServ agents' emotions against the Mood Adjective Checklist findings presented in Haney et al. (1973a). Additional information on the Stanford Prison Experiment ground-truth data is available in (Nye, 2011).

6.2.1 Action Ordering Validity

The action ordering is the most important external validity test. The order of first expression the Throw In Hole and Resist behaviors was inferred from the original data sources. Action orderings were compared against the ground truth ordering by using an inversion count algorithm that adjusts for ties and right censored elements (Nye, 2011). In this context, a right censored element occurs when an agent does not take the action during a simulation run, making their ordinal position "to the right" of the observed ordering.

Inversion count algorithms determine the minimum number of single-element swaps that are necessary to turn one ordering into another ordering. The inversion number of a random permutation follows a distribution somewhat similar to a normal distribution (Margolius, 2001). Each condition consisted of 30 simulation runs, so each agent's median action ordering was calculated as the median of their positions on individual runs.

Table 2: Action Ordering Correspondence

	Full Knowledge	Authority	Meme
ThrowInHole	0.82	0.67	0.85
ThrowInHole (No Innovators)	0.77	0.58	0.81
Resist	0.71	0.69	0.79
Resist (No Innovators)	0.84	0.80	0.61

Table 2 shows the correspondence of the median sequences to the ground truth sequences, with and without innovators. Correspondence values are determined by a formula of 1-(Inversions)/(Maximum Inversions), where 0.5 represents the inversions expected by random chance. Excluding innovators is necessary for comparison of conditions, since otherwise the Meme condition would have an unfair advantage over other conditions.

In all conditions, the simulation performs much better than chance for predicting the order that agents first adopt each behavior. Additionally, the Full Knowledge and Meme conditions appear superior to the Authority condition. Analysis of the individual runs confirms these trends. These conditions were also examined at a qualitative level. In the Full Knowledge condition, agents consider all their options from the start so the ordering was driven by motivational factors, particularly personality. Orderings in the Authority condition were similar to Full Knowledge, but with greater variation due to the randomness in who learns initially and limited social learning during runs. The Meme condition was strongly driven by both social learning and by personality factors. Regression and correlation analysis indicated that the most stable influences on social learning were valence, selective attention, and in-group membership in this condition.

The Meme condition's strong performance on median-orderings supports it as a plausible mechanism in the Stanford Prison for the spread of the Throw In Hole behavior. However, the Meme condition performs worse on the Resist action than the Full Knowledge sequences. This indicates that the spread of Resist actions was due to personal tendencies and situations, rather than social learning. These findings validate that the simulated Stanford Prison Experiment models the order that agents adopt these behaviors better than chance. Performance on Throw In Hole also indicates that the agents' cognitive models offer added value for examining the spread of affordances through social learning.

6.2.2 Action Frequency Validation

This analysis looked at the relative action frequencies in the simulation, among those that were coded from the Stanford Prison Experiment videos. The Stanford Prison experiment data had counts of commands, helping, information, insults, resistance, and use of instruments. The raw count of each action is normalized by the total count of all these actions–generating the fraction of record actions that fall into each category. For the simulations, which had many runs, these fractions were averaged across all runs for a condition. Table 3 shows the relative proportions of each type of action. Between conditions, the relative frequencies were fairly stable.

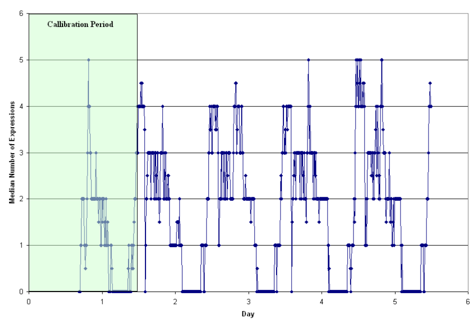
Table 3: Relative Action Frequencies

Action	Ground Truth	Full Knowl.	Meme	Authority
Command	0.38	0.27	0.29	0.29
Help	0.002	0.05	0.05	0.04
Information	0.18	0.16	0.15	0.16
Insult	0.18	0.05	0.03	0.03
Resist	0.09	0.29	0.30	0.30
Threat	0.09	0.02	0.01	0.01
Use Of Instruments	0.08	0.16	0.17	0.18

Despite the calibration over the initial portion of the experiment, the full simulation runs showed a few deviations from the expected action frequencies.

Commands, Help, and Information each fall into their expected ranks– with Commands being a very common action, Information being somewhat common, and Help being uncommon. Commands, while still the most common guard action, were less common than in the actual experiment. Helping was slightly more common, but still very uncommon. Information showed almost an exact match. Insults were significantly less common in the simulation, as were threats. Instead, the use of instruments became a more popular action. Since Use of Instruments, Threats, and Insults are functionally similar, this is notable but not particularly interesting.

Figure 5: Median Prisoner Resistance Over Time



Resistance was significantly more common in the simulation than the actual experiment. It is possible that tuning based upon the first day made resistance more attractive than intended, since the original experiment showed little resistance on the first day. This may have caused resistance to be more common in the later portions of the experiment. Fig. 5 supports this interpretation. Over the first 20 hours of the experiment that were used for calibration, the median resistance occurred at 1.31 resistance actions per time step. Over the full experiment, this value averaged 1.73 actions per time step, a 33% increase. Despite this irregularity, the action frequencies overall were fairly consistent with the ground truth frequencies.

6.2.3 Emotional State Validation

Emotional state validation compared the simulated agents' emotions against those from the empirical experiment. For each simulation run, the average was calculated for agents in the Prisoner group and for agents in the Guard group. Since PMFServ agents utilize 8 primary emotions, each agent's emotions were aggregated according to the Eqn. 8. This was calculated to compare against the original Stanford Prison Experiment, which reported that prisoners were on average three times less happy than guards.

$$\begin{aligned} AggregatedEmotion = \frac{1}{4}(\quad & (Joy - Distress) + \\ & (Pride - Shame) + \\ & (Liking - Disliking) + \\ & (Gratification - Remorse)) \end{aligned} \tag{8}$$

$$\overrightarrow{Emotions_r(Group)[t]} = \frac{1}{N} \sum_{x \in Group}^{N} Emotion_r(Agent_x)[t] \tag{9}$$

To determine the emotional trends of each group, the emotions of the members had to be combined into a representative set of time series for the group. This was done by calculating the mean value of emotions for the group at each time point, for each run. This generated a vector of average emotions for a group for each run. Each element of the vector for any run r at a given time step t follows Eqn. 9 and ranged between -1 and 1. Table 4 shows the mean, median, and standard deviation for these values for each of the experimental conditions. It is evident in looking at the table that both guards and prisoners were somewhat unhappy in the experiment, on average. This matches the ground truth findings.

Table 4: Group Average Emotion Values

Group	Mean	Median	Std Dev
Guards (Full Knowl.)	-0.03	-0.05	0.05
Guards (Authority)	-0.05	-0.05	0.01
Guards (Meme)	-0.05	-0.05	0.03
Prisoners (Full Knowl.)	-0.11	-0.13	0.05
Prisoners (Authority)	-0.13	-0.13	0.02
Prisoners (Meme)	-0.12	-0.12	0.03

T-Tests were run on emotion data used to generate Table 4 to test if the guard emotions were higher than prisoner emotions, for each of the simulation conditions. In all conditions, the probability of the null hypothesis was $p < 1 \times 10^{-6}$. This strongly indicates that guards were happier than prisoners in the simulation. Comparing the means, prisoners were between 2.3 and 3.4 times less happy than the guards in the simulation. This corresponds well with the Stanford findings, which estimated prisoners as being about 3 times less happy than the guards.

7 Conclusions and Future Directions

This implementation offers three advantages over existing computational models of behavioral adoption: unintentional learning, multi-layered social and environmental attention cues, and contextual adoption. Unintentional learning is learning that occurs through normal interaction, rather than directed conversation which occurs in frameworks such as Construct from CMU (Schreiber & Carley, 2007).

Multi-layered attention cues support this process, including six social cues, three informational cues, and a selective attention mechanism. To this author's knowledge, no computational model represents this breadth of factors impacting social learning or behavioral adoption. Since attention is a competitive process, these mechanisms are important to realistically model who adopts new behavior.

Finally, the model supports contextual adoption of behavior. The cognitive model treats new affordances no different than other known affordances, so adopting a new behavior has an opportunity cost of not performing other available behaviors. Since the available affordances change over time, agents may adopt certain behavior in some contexts and not others. Since the model supports modeling individual personalities, individual differences also play an important role in the adoption of behavior.

Moving forward, open questions currently limit the cognitive model used in this study. As noted in Section 4.2, the interaction between attentional cues have not been well-explored. Experimental studies on these interactions would improve understanding of these effects. Secondly, studies on these cues do not explicitly disentangle attentional and motivational impacts on adopting of new behavior. Further exploration of these topics would support improved cognitive modeling of behavioral adoption patterns by detailing how these factors interact.

8 Acknowledgments

Thank you to the Air Force Office of Scientific Research, whose basic research support made this work possible. Also, my sincere thanks to Professor Zimbardo, who was exceptionally responsive and helpful in arranging my access to the archival Stanford Prison Experiment data. Finally, I would like to give a special thanks to the Archives of the History of American Psychology which graciously allowed me to collect data on-site for many days.

References

Adomo, T. W., Frenkel-Brunswik, E., Levinson, D. J., & Sanford, R. N. (1950). *The authoritarian personality*. New York, NY: Harpers and Bros.

Asch, S. (1955). Opinions and social pressure. *Scientific American*, *193*(5), 31–35.

Axelrod, R. (1997). Advancing the art of simulation in the social sciences. *Complexity*, *3*(2), 16–22.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: PrenticeHall.

Bornstein, R. (1989). Exposure and affect: Overview and metaanalysis of research, 1968-1987. *Psychological Bulletin*, *106*(2), 265–289.

Cherry, C. E. (1953). Some experiments on the recognition of speech, with one and with two ears. *The Journal of the Acoustical Society of America*, *25*, 975–979.

Christie, R., & Geis, F. (1970). *Studies in machiavellianism*. Academic Press.

Comrey, A. (2008). The Comrey personality scales. In *The SAGE handbook of personality theory and assessment: Personality measurement and testing*. Thousand Oaks, CA: Sage Publications Ltd.

Dennett, D. C. (1995). *Darwin's dangerous idea: Evolution and the meanings of life*. New York, NY: Simon & Schuster.

Fazio, R. H., Roskos-Ewoldsen, D. R., & Powell, M. C. (1994). Attitudes, perception, and attention. In P. M. Niedenthal & S. Kitayama (Eds.), *The heart's eye: Emotional influences in perception and attention* (pp. 197–216). New York, NY: Academic Press.

Fromm, E. (1973). *The anatomy of human destructiveness*. New York, NY: Henry Holt.

Gaver, W. (1991). Technology affordances. In *Proceedings of the SIGCHI conference on human factors in computing systems: Reaching through technology* (pp. 79–84).

Gibson, J. J. (1979). *The ecological approach to perception*. Boston, MA: Haughton Mifflin.

Gibson, J. J. (1986). *The ecological approach to visual perception*. Mahwah, NJ: Lawrence Erlbaum Associates.

Haney, C., Banks, W. C., & Zimbardo, P. G. (1973a). Interpersonal dynamics in a simulated prison. *International Journal of Criminology and Penology*, *1*, 69–97.

Haney, C., Banks, W. C., & Zimbardo, P. G. (1973b). Study of prisoners and guards in a simulated prison. *Naval Research Reviews*, *9*, 1–17.

Hilmert, C., Kulik, J., & Christenfeld, N. (2006). Positive and negative opinion modeling: The influence of another's similarity and dissimilarity. *Journal of Personality and Social Psychology*, *90*(3), 440–452.

James, W. (1890). *The principles of psychology*. Cambridge, MA: Harvard University Press.

Johnston, W. A., Hawley, K. J., Plewe, S. H., Elliott, J. M. G., & DeWitt, M. J. (1990). Attention capture by novel stimuli. *Journal of Experimental Psychology: General*, *119*(4), 397–411.

Kameda, T., Ohtsubo, Y., & Takezawa, M. (1997). Centrality in sociocognitive networks and social influence: An illustration in a group decision-making context. *Journal of Personality and Social Psychology*, *73*(2), 296–309.

Kelley, G. A. (1955). *The psychology of personal* *constructs*. New York, NY: WW Norton.

Lee, D., Itti, L., Koch, C., & Braun, J. (1999). Attention activates winner-take-all competition among visual filters. *Nature Neuroscience*, *2*, 375–381.

Mackintosh, N. (1983). *Conditioning and associative learning*. New York, NY: Oxford University Press.

Mantell, D. (1971). The potential for violence in Germany. *Journal of Social Issues*, *27*(4), 101–112.

Margolius, B. H. (2001). Permutations with inversions. *Journal of Integer Sequences*, *4*(Article 01.2.4).

Milgram, S. (2004). Behavioral study of obedience. In N. Scheper-Hughes, S. P. Bourgois, & P. I. Bourgois (Eds.), *Violence in war and peace.* Boston, MA: Blackwell Publishers.

Nye, B. D. (2011). *Modeling memes: A memetic view of affordance learning*. Doctoral dissertation, University of Pennsylvania.

O'Brien, S. (2010). Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review*, *12*(1), 87–104.

Petty, R. E., & Cacioppo, J. T. (1986). The elaboration likelihood model of persuasion. *Advances in experimental social psychology*, *19*, 123–205.

Platow, M. J., Haslamb, S. A., Botha, A., Chewa, I., Cuddona, M., Goharpeya, N., et al. (2005). "It's not funny if they're laughing": Self-categorization, social influence, and responses to canned laughter. *Journal of Experimental Social Psychology*, *41*(5), 542–550.

Reicher, S., & Haslam, S. (2006). Rethinking the psychology of tyranny: The BBC prison study. *British Journal of Social Psychology*, *45*(1), 1–40.

Rogers, E. (1995). *Diffusion of innovations*. Free Press.

Schreiber, C., & Carley, K. M. (2007). Agent interactions in construct: An empirical validation using calibrated grounding. In *2007 conference on behavior representation in modeling and simulation (BRIMS).* Norfolk, VA: SISO.

Shannon, C. E. (1948). A mathematical theory of communication. *Key Papers in the Development of Information Theory*. Available from `http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf`

Silverman, B. G. (2004). Toward realism in human performance simulation. In J. W. Ness, V. Tepe, & D. R. Ritzer (Eds.), *The science and simulation of human performance* (Vol. 5, pp. 469–498). JAI Press.

Silverman, B. G., & Bharathy, G. K. (2005). Modeling the personality & cognition of leaders. In *2005 conference on behavior representation in modeling and simulation (BRIMS).* Norfolk, VA: SISO.

Silverman, B. G., Bharathy, G. K., Johns, M., Eidelson, R. J., Smith, T. E., & Nye, B. D. (2007). Sociocultural games for training and analysis. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, *37*(6), 1113–1130.

Silverman, B. G., Bharathy, G. K., Nye, B. D., & Eidelson, R. J. (2007). Modeling factions for "Effects based operations": Part I Leader and follower behaviors. *Journal of Computational and Mathematical Organization Theory*, *13*(4), 379–406.

Silverman, B. G., Johns, M., Cornwell, J. B., & O'Brien, K. (2006). Human behavior models for agents in simulators and games: Part I: Enabling science with PMFserv. *Presence: Teleoperators & Virtual Environments*, *15*(2), 139–162.

Silverman, B. G., Pietrocola, D., Nye, B. D., Weyer, N., Osin, O., Johnson, D., et al. (2012). Rich socio-cognitive agents for immersive training environments – case of NonKin Village. *Autonomous Agents and Multi-Agent Systems*, *24*(2), 312-343.

Simons, D. J., & Chabris, C. F. (1999). Gorillas in our midst: sustained inattentional blindness for dynamic events. *Perception*, *28*, 1059–1074.

Tajfel, H. (1982). Social psychology of intergroup relations. *Annual Reviews in Psychology*, *33*(1), 1–39.

Tanford, S., & Penrod, S. (1984). Social influence model: A formal integration of research on majority and minority influence processes. *Psychological Bulletin*, *95*(2), 189–225.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, *12*(1), 97–136.

Vilpponen, A., Winter, S., & Sundqvist, S. (2006). Electronic word-of-mouth in online environments: Exploring referral network structure and adoption behavior. *Journal of Interactive Advertising*, *6*(2), 71–86.

Zimbardo, P. (2007). *The Lucifer effect: How good people turn evil*. London: Rider.

Author Biographies

BENJAMIN D. NYE is a post-doctoral fellow at the University of Pennsylvania. He performed his doctoral work at the Ackoff Center for Advancement of Systems Approaches, focusing on social simulation, cognitive modeling, and model-based architectures.

Conquest, Contact, and Convention: Simulating the Norman Invasion's Impact on Linguistic Usage

Jason Quinley
Seminar für Sprachwissenschaft
University of Tübingen, Germany
jason.quinley@student.uni-tuebingen.de

Roland Mühlenbernd
Seminar für Sprachwissenschaft
University of Tübingen, Germany
roland.muehlenbernd@uni-tuebingen.de

ABSTRACT: *Here we simulate the impact of the Norman conquest of 1066 on modern communication strategies. The simulations inject a population of "Normans" into a population of "Anglo-Saxons" situated on a scale-free network and incorporate signaling games with a best-response learning dynamic. Various trials accounted for the assertion by modern historians that the salient systematic division of prestige seen in words of French versus Germanic origin is no accident but rather results from social conditions. The two main veins of exploration account for social context and social structure (i.e. network topology). They also illuminate that the conventions seen in modern English could have gone the other way without appropriate social conditions. In particular, we draw attention to the broad range of applicability of our results and methods to situations of invasive, stable populations integrated into a larger one.*

1. Introduction

English shares cognates with many other languages, notably German and French. The Norman conquest of 1066 radically changed both England's history and its language, infusing new words into the lexicon and creating a divergence in register and context. Consider the differentiation between words for animals and meat; e.g. pork (Fr. *porc*) and swine (Ger. *schwein*). Historians point to social conditions like the lifestyles of the Anglo-Saxon laborers and the French-speaking Norman nobility as the cause of the divergence. But could the results have been different? And what can game-theoretic models or network-centric approaches offer to understand this?

Language contact inevitably means contact between speakers. It not only introduces new words into a lexicon but also can trigger shifts in usage and interpretation, as seen in Table 1. These shifts may release speakers from the computationally burdensome disambiguation process involved with *polysemous* words; their meaning can only be resolved by context. E.g. for the sentences "I fed the rabbit" and "I ate rabbit" the meaning of "rabbit" in the first sentence (the animal itself) and in the second (its meat) is derived by the context.[1] This kind of polysemy is especially apparent for words like *rabbit, goose, shark, fish,* etc., but not for animals like *swine, cow,* and *sheep*, which in general cannot be used for the animal's meat. This is called *partial blocking* (see e.g. Blutner, 2000): the option to construe these words with the animal's meat is blocked by specialized words *pork, beef* and *mutton.* Interestingly, *swine, cow* and *sheep* are of Germanic origin and their today's German counterparts *Schwein, Kuh,* and *Schaf* bear the above-mentioned polysemy.[2]

[1] Obviously the usage as an uncountable or mass noun conveys strong evidence for the concrete meaning (*conceptual grinding* effect), but i) it is not the only evidence and ii) can be seen as a contextual feature.
[2] In fact *Kuh* is an exception, blocked by the German word *Rind*.

Further, *pork, beef* and *mutton* are of French origin; their today's French counterparts are *porc, boeuf* and *mouton.* This suggests the assumption that i) *swine, cow* and *sheep* once bore the animal/meat polysemy and were used in a context dependent way and ii) lost this polysemy by adopting alternative words of French origin, causing the partial blocking. We call this process *emancipation from context dependence.*

All in all, in any case of such a partial blocking instance words of French origin designate the animal's meat, words of Germanic origin the animal itself. This follows a salient trend in the English language: vocabulary of French origin tends to the prestigious, whereas words of Germanic stock often fall into working class topics; i.e. the laboring class has to work (Ger. *Werk*), the upper class gets to play (from Fr. *plaisir* = pleasure). Table 1 shows two different types of systematic division of meaning space between words of Germanic and French origin, next to the animal/meat division a distribution of personal and abstract concepts.

personal	abstract
freedom	liberty
knowledge	science
belief	faith
brotherly	fraternal

animal	meat
swine	pork
cow	beef
sheep	mutton
deer	venison

Table 1: Systematic division of meaning space between words of Germanic and French origin, possibly a result of lifestyle and education differences. The left table shows the division between personal (Germanic origin) and abstract (French origin) concepts. The right table shows the division betweens words for animals and their meat, possibly a result of the French speaking nobility eating the meat of the animals raised by the English speaking workers.

In this article we simulate language contact and its impact on language use by applying signaling games (Lewis, 1969) to social networks. We want to i) simulate the emancipation from context dependence and ii) investigate the social parameters responsible for the salient systematic division of meaning space between words of Germanic and French origin. We proceed by outlining our modifications to the standard signaling game, describing the impact of social context and structure on the adoption of the expected convention, and remarking on the model itself and future directions.

2. Context and Signaling Games

How do conventions arise? Lewis (1969) adressed this in his work *Convention* via *signaling games,* a mathematical model of communication where a sender sends a message to a receiver who then interprets it. When we say conventions, we mean by that a system of coordinated behavior pairing information states with actions; this is typically common knowledge among a group of participants. Examples include driving on the left side of the road in the U.K. or nodding one's head to signal *Yes* in many Western countries. A trip to U.S. will evince the arbitrariness of the driving convention, as will a trip to Bulgaria the nodding one. The nodding example cuts to the heart of our discussion because it reveals a situation comprising coordination over permutations of information, signals, and action. To examine this, we first illustrate a standard signaling game.

2.1 Signaling Games

Consider someone who wants to affirm or negate a proposition. He can nod or shake his head towards his colleague, and his colleague will interpret the gesture with an action. If the interpretation matches the intent of the signal, we can say that the communication was successful. Formally, we can think of this interaction as $G = \langle \{S, R\}, T, M, A \rangle$. S is the sender, R the receiver of the game. T is the set of *topics* the speaker wants to communicate; here $T = \{\text{yes, no}\}$. M is the set of *messages*; $M = \{\text{nod, shake}\}$. Last,$A$ is the set of *actions*; $A = \{\text{yes, no}\}$. Observe that T and A need not be the same, but for simplicity's sake we adhere to that here. There are two properties of the game now to consider: i) the overall structure of the decision process seen in Figure 1 and ii) the potential strategies used by agents in the game, examples of which are seen in Figure 2.

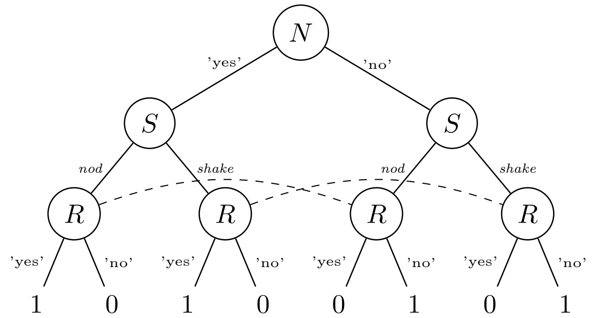


Figure 1: Extensive form game for the standard signaling game's example *nod or shake.*

We assume a base level of rationality in that agents will not choose a strategy with a lower payoff if one with a higher payoff is available. From this, we say the game reaches a *Nash Equilibrium* if agents have no incentive under a given strategy profile to deviate unilaterally from their current strategy. Lewis stated that conventions are such equilibria. In Figure 2, we see that communicative success is more important than the particular convention itself, a principle to keep in mind for the next section. As a general rule, conventions do not merely exist between dyads, but on a societal level. To address this, we delve further into augmenting signaling games with social structure and context.

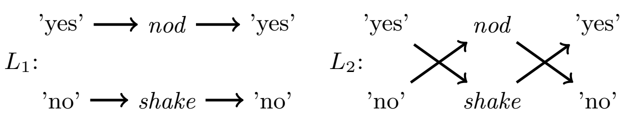


Figure 2: Strategies depicting perfect signaling systems for the standard signaling game's example *nod or shake.*

2.2 Context Signaling Games

Language use is modelled by the *context signaling game* (CSG), a Lewisean signaling game extended by a set of contexts. Such a game is given as $CSG = \langle \{S, R\}, T, L, C, P_T, P_C \rangle$. S is the speaker, R the recipient of the game. T is the set of *topics* the speaker wants to communicate; here $T = \{\text{meat, animal}\}$. L is the *lexicon*, which contains expressions for the appropriate animal. C is a set of different contexts the participants could be in, represented here by $C = \{\text{dinner, farm}\}$. The label *dinner* depicts a context where the topic 'meat' is highly probable; likewise for *farm* and 'animal.' These dependencies are modelled by the *topic probability function* P_T[3]. Finally the *context probability function* P_C depicts the probability of being in a specific context and depends on the speaker's social status $\sigma \in \mathbb{N}$. We assume that the higher the social status of a speaker, the higher the probability that he will be in a dinner context and not in a farm context. With this in mind we defined nine different social statuses from 1 to 9, where a speaker with social status σ is in a dinner context with probability $\sigma/10$ and in a farm context with probability $1 - (\sigma/10)$, as depicted in Table 2.

Status σ	1	2	3	4	5	6	7	8	9
P_C(dinner)	.1	.2	.3	.4	.5	.6	.7	.8	.9
P_C(farm)	.9	.8	.7	.6	.5	.4	.3	.2	.1

Table 2: Speakers social status σ depicts his probability P_C of being in a dinner context or farm context.

A round of such a game between a speaker S and a recipient R can described as follows: nature N chooses a context $c \in C$ with probability $P_C(c)$. Then nature chooses a topic $t \in T$ with probability $P_T(t|c)$. Now speaker S communicates topic t by using an appropriate expression l of his lexicon L. Then recipient R has to construe the received expression l with a topic $t' \in T$. Communication is successful if R reconstructed the topic S had in mind, thus if $t = t'$. Then both participants get a *utility value* of 1, otherwise 0 for miscommunication. Both participants know the given context, but only S knows the given topic. The way a speaker allocates expressions to topics and a recipient allocates topics to expressions is called their *strategy.* Figure 1 depicts the *extensive form* of the CSG with $L = \{$“swine”, “pork”$\}$.

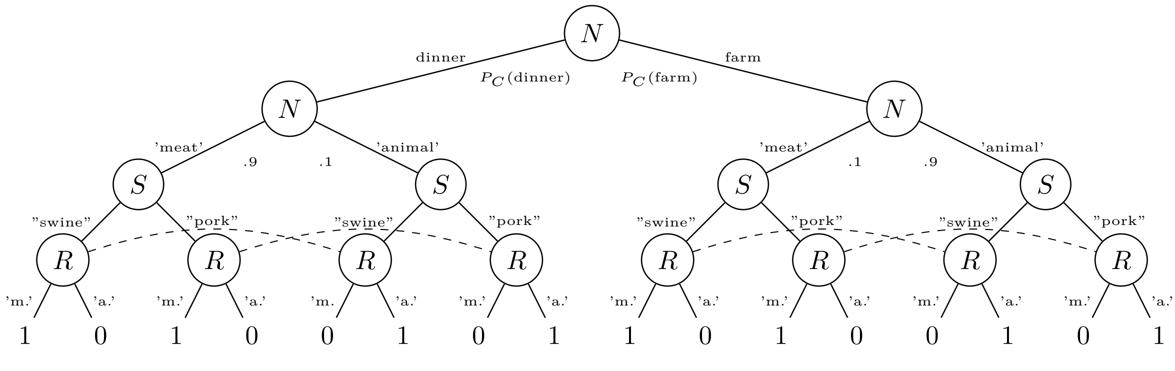


Figure 3: The extensive form of the CSG. (N.B. dashed lines connect situations the recipient cannot distinguish; leaves resulting in 1 depict successful communication.)

[3]The probability function $P_T \in \Delta(T)^C$ returns the probability for $t \in T$ being topic in a context $c \in C$. Here we chose $P_T(\text{meat}|\text{dinner}) = .9$ and $P_T(\text{animal}|\text{farm}) = .9$. Accordingly: $P_T(\text{animal}|\text{dinner}) = .1$, $P_T(\text{meat}|\text{farm}) = .1$.

3. Experimental Set-up & Results

300 agents with a randomly chosen social status and arranged on a social network with *scale-free properties* (Jackson, 2008) played the *CSG* repeatedly as both speaker and recipient with their neighbors on the network. Based on previous encounters, agents choose the *best response* to according to a *belief learning dynamic* (see e.g. Nachbar, 2008). At a simulation's start the agents' lexicons contain only the expression "swine". Right after all agents have learned the same strategy, we simulate the conquest of 1066 by replacing 10% of the agents by Norman invaders, agents whose lexicons contain only the expression "pork". When an agent as a recipient encounters an unknown expression, he adopts it to his lexicon. The simulation ends when every agent's lexicon contains "swine" and "pork" and a unique strategy governs the whole society. In this way we conducted three different experiments, which differ in the conditions of which agents are possible candidates to be replaced. Figures 4, 5 and 6 depict an example of all three experiments.

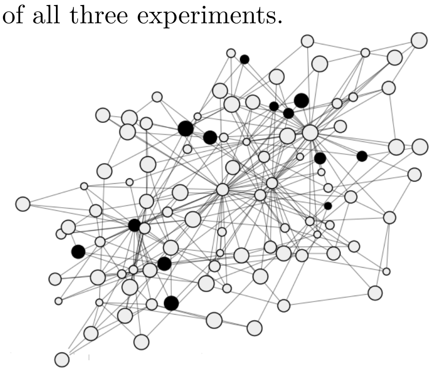


Figure 4: Experiment 1: randomly chosen agents are replaced by Norman invaders (black circles). Note: the larger the circle, the higher the social status.

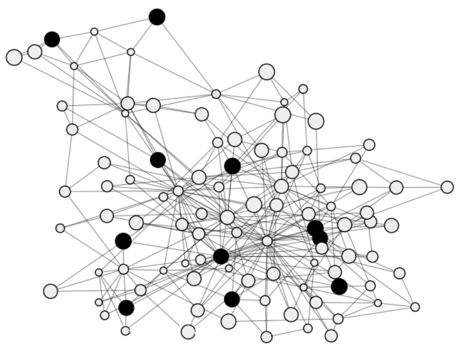


Figure 5: Experiment 2: only agents with a social status higher than a given lower limit are candidates to be replaced. Note: only large circles are replaced by Norman invaders.

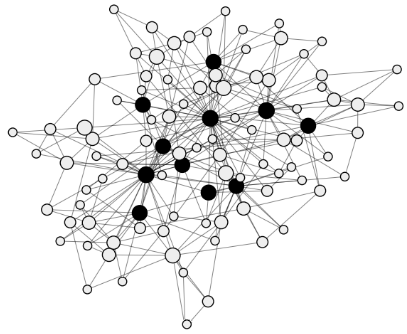


Figure 6: Experiment 3: agents must have a high social status paired with a high degree of centrality (number of neighbors) to be candidates to be replaced. Note: the network features a correspondence of social status and local influence.

3.1 Result 1: Context (In-)Dependence

The agents' initial strategy as a speaker is to use expression "swine" for each topic in each context (it's the only one in the lexicon). As recipient the agents' strategy is context-dependent: in the dinner context they construe "swine" as 'meat', in the farm context as 'animal'. Informally, the agents simply learn that topic 'meat' is more probable in the dinner context and topic 'animal' in the farm context when construing "swine". In Experiment 1 the invasion is done by replacing 30 randomly chosen agents with invaders (e.g. Figure 4). It cause the word "pork" to enter every agent's lexicon, spreading over the society. Every agent learns a new *context-independent* strategy. At the end of each trial every agent has learned one of two possible strategies: S_1 for using the word of French origin for the meat and word of Germanic origin for the animal or S_2, the other way around (Figure 7). Hundreds of simulation runs revealed that i) in each trial only one strategy spreads and stabilizes society-wide and ii) both strategies' emergence is equiprobable. But if it is a question of chance whether S_1 or S_2 emerges, then how can we explain that language use according to S_1 is predominant?

$$S_1: \begin{array}{rcl} \text{'meat'} & \longleftrightarrow & \text{"pork"} \\ \text{'animal'} & \longleftrightarrow & \text{"swine"} \end{array}$$

$$S_2: \begin{array}{rcl} \text{'meat'} & \nwarrow\!\!\!\nearrow & \text{"pork"} \\ \text{'animal'} & \swarrow\!\!\!\searrow & \text{"swine"} \end{array}$$

Figure 7: The two strategies S_1 and S_2, whose emergence is equiprobable if the replaced agents are chosen randomly, like in Experiment 1.

3.2 Result 2: Influence of Social Status

In the previous experiment, randomly chosen agents were replaced by invaders. In Experiment 2 and 3, we account for the fact that the Normans probably occupied high social positions in two ways: first by tying social status to context, then to connectivity.

To model the first, in Experiment 2 we consider the social status σ of the replaced agents as follows: we set a lower limit $\beta \in \mathbb{N}$ in such a way that only agents with a social status $\sigma \geq \beta$ could be replaced by invaders. Beginning with Trial 1 ($\beta = 1$) we incrementally increased β stepwise up to Trial 9 ($\beta = 9$) with 300 simulation runs per trial (see e.g. Figure 5 for $\beta = 8$). The simulation results showed: the higher β and therefore the higher the social status of agents replaced by invaders, the higher the probability for the emergence of the expected strategy S_1 (see Figure 8).

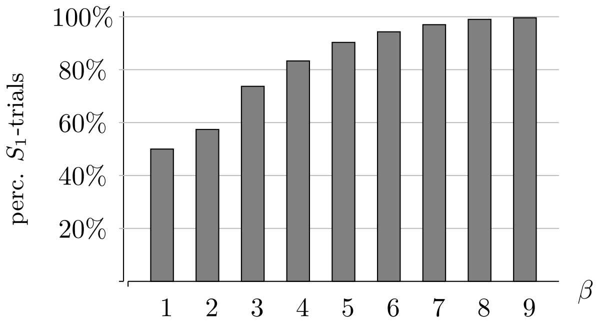


Figure 8: Result of Experiment 2: by increasing the threshold β of the replaced agents, the probability for a society-wide spread of expected strategy S_1 raises.

3.3 Result 3: Influence of Social Structure

For Experiment 3, we attached a second condition to the social status parameter σ, namely a value for the measure of centrality (number of neighbors) d as follows: the higher the social status σ, the higher d and therefore the higher the local influence (see e.g. Figure 6). This aligns with work depicting high status agents as more influential in a society (Nettle, 1999). Interpreting wealth as number of business (or speaking) partners, we derive a rationale for replacing the hubs in the original society with Norman invaders for high β values. We found that this additional parameter accelerated the probability for the adoption of the expected strategy (see Figure 9).

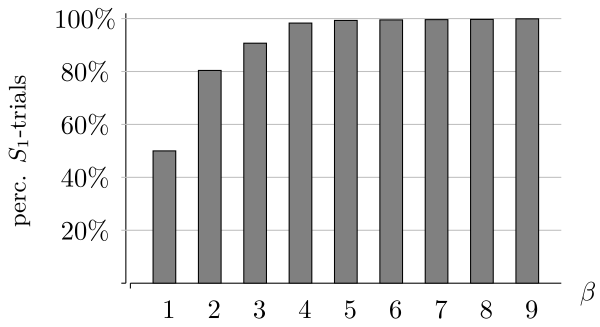


Figure 9: Result of Experiment 3: if the social status σ is also correlated with a higher degree d and therefore a higher local influence, the increasing of threshold β of the replaced agents leads to an accelerated raising of the probability for a society-wide spread of expected strategy S_1.

4. Discussion

Our simulations showed that agents in a social network playing the CSG with only one word in their lexicon resolve its polysemy in a context-dependent fashion. The simulated Norman invasion provided alternatives between words, allowing speakers to distinguish between previously context-dependent meanings. But without further assumptions, the new words could have described any meaning equiprobably. By considering social status and structure, the probability shifted markedly to the expected strategy: the words of the invaders tend towards meanings of upper-class contexts, while the words of the conquered associate more with the lower class.

4.1 Cultural Contact

What does this portend for cross-cultural contact on a more general level and those interested in it? Observe that while these signaling conventions were linguistically motivated, they are more general. Consider first, the signals are costless to learn for the agents. It is the potential for liberation from dependence on context that fosters the new convention. This may not be the case in contact on the ground. Observe also that we are introducing a new signal into an effectively monolexical society, thus effectively doubling the available lexicon. Other research in signaling games (e.g. Mühlenbernd, 2011) details the consequences of more costly signaling approaches.

While the signals are costless to learn, another feature of the model is that there is no inherent dyadic power difference among agents. I.e. the invading agents receive the same penalty for miscommunication as the invaded society. Further experiments into the payoff matrix that best represents an asymmetric dynamic might reveal a different impact on learning the convention. This also recalls that there is no guarantee that the expected convention will emerge without the added social pressure of context or structural placement.

Last, observe that it is not only the invaded population that adopts the convention but also the invaders. Under more stringent conditions, refusal of the invaders to adopt some of the signals of their new society can lead to new, less efficient equilibria.

4.2 Notes on Implementation and Application

While motivated by linguistic history, these results apply to more general phenomena of sociological modeling. We highlight three salient features of the model to that end. First is the Lewisean Signaling Game, a model of communication with applications ranging from linguistics to biology and on scales from bacterial to multi-national. As more permuations of decision spaces and strategies exist to arrive at the game's equilibrium than addressed here, there is great flexibility in implementing such a model. These include potentially infinite type, message, or action spaces; different update dynamics; or message costs.

Second, the interaction protocols do not exist merely between agents, but within them as well. E.g., the update dynamics for the signaling game represent a recognized and tractable implementation of cognitive capacity and theory of mind applied to a dynamic decision process. Various approaches, like those seen in Mühlenbernd (2011), give a more comprehensive view. Underlying this update dynamic is also a risk-neutral calculation procedure. Were the game, as mentioned above, subject to a different payoff matrix, risk aversion might also be a more realistic paradigm to implement.

Last, while the communication and learning of conventions occurs in a dynamic fashion, the overall structure of the network is fixed into a scale-free topology. The choice for this was based on results in data on preferential attachment (Jackson, 2008), something we assume a realistic social network to be a result of. Cultural contact not only brings new conventions or signals into a network, but it also can alter the network's topology or interaction mechanisms. For this reason, a network that included strategic link deletion or formation might also shed new light on the way that a convention spreads through the network through a signaling game.

5. Selected References

Blutner, R. (2000). Some Aspects of Optimality in Natural Language Interpretation. *Semantics, 17(3)*, 189-216.

Jackson, M. O. (2008). *Social and economic networks.* Princeton: Princeton University Press.

Lewis, D. (1969). *Convention.* Cambridge: Harvard University Press.

Mühlenbernd, R. (2011). Learning with Neighbors. *Synthese, 183*, 87-109.

Nachbar, J. (2008). Learning and evolution in games: belief learning. *The new Palgrave Dictionary of Economics.*

Nettle, D. (1999). Using Social Impact Theory to simulate language change. *Lingua, 108*, 95–117.

Author Biographies

JASON QUINLEY is a PhD student in Linguistics at the University of Tübingen, Germany with a B.S. and M.A. in Mathematics along with an M.A. in Computational Linguistics. His research interests include Public Goods Games and the Evolution of Cooperation, Politeness Theory and Prestige, Social and Economic Networks, and Language and Power.

ROLAND MÜHLENBERND is a PhD student in Linguistics at the University of Tübingen, Germany. With a B.S. in Computer Science and an M.S. in Media Science, his interdisciplinary background is also displayed by his multiple interests, especially in Game Theory, Social Sciences & Networks, Artificial Intelligence and Language Evolution.

Social Network Analysis and Simulation of the Development of Adversarial Networks

Razvan Orendovici, Computer Science and Engineering
Frank E. Ritter, College of IST
Pennsylvania State University
University Park PA
razvan.orendovici@gmail.com, frank.ritter@psu.edu

Keywords:
network visualization, social network, simulation, adversarial networks, development of networks

ABSTRACT: *We present a novel way to monitor and analyze the time course of adversarial networks through a simulation tool and supporting mathematical analysis. Recent work on social networks has been used in the analysis of adversarial networks and their underlying structure with hopes of detecting and preventing future activity. In this paper we consider an adversarial network to be a network subgroup that works against the interests of the group studying it. ANA, the software package presented here, can portray the structure of such networks and allow analysts to find patterns and key players in the network while watching the network evolve. Finally, this work uses output from ANA to study how a simulated adversarial scenario grows in structure and compares it to more traditional social networks on standard measures and also analyze the changes over time of this network on those same dimensions.*

1. Introduction

We present a novel tool to monitor, analyze, and examineadversarial social networks through a visualization tool and supporting mathematical analysis. This tool is demonstrated by presenting a simulated adversarial network that was used to guide the development of the visualization tool. Finally, we analyze how this network evolved and present the implications for education, network science, social network analysis and measurement.

1.1 Motivation

Social network analysis and visualization of adversarial networks is a very powerful method of understanding networks and keeping track of relevant information about the network as it evolves and becomes more defined. Recent events and political pressures have led to a lot of research into adversarial networks and their identification. This, combined with recent popularity of social network analysis, has made a network centric analysis of these networks interesting and useful. Throughout this work we look at allowing users to take mock intelligence gatherings on the communications and individuals involved in a possible adversarial network and keep track of this information while visualizing it and performing statistical analysis.

After describing the background to this work, we present a visualization tool in Section 2, tailored specifically at the style of data that intelligence agencies capture that allows the user to store the information and visualize the graph as it changes. Once this information is entered and visualized, the tool provides additional features to allow this data to be analyzed over its evolution and new patterns detected. Section 3 looks at this analysis for the adversarial network created by intelligence data with analysis found on more traditional networks seen in the literature.

The adversarial network viewed in this paper is a series of intercepted communications in a mock terrorist plot (Shemanski, 2011). The communications are made to appear as intelligence reports from various national agencies and present links between the various actors in the plot. This tech report is used in security and risk analysis classes for students to analyze in the College of Information Sciences and Technology at The Pennsylvania state University.

1.2 Background

Social network analysis has been a growing field since the work done by Moreno (1978) and others (Anthonisse, 1971; Beauchamp, 1965; Freeman, 1979; Holland & Leinhardt, 1971, 1972; Sabidussi, 1966) to establish measures and calculations of the field in the 1950's to the 1970's. With the adoption of the Internet and growth of social network platforms this analysis has become even more widespread and been applied to a variety of fields.

Communities of authorship (Albert & Barabasi, 2002; Barabasi, Jeong, Nelda, Ravasz, Schubert, & Vicsek, 2002; Newman, 2003; Qiu, Ivanova, Yen, Liu, & Ritter, 2011), Economic communities and online games (Bakshy, Simmons, Huffaker, Teng, & Adamic, 2010), and many other fields have benefited from the application of network science and graph theories. In the past 20 years there has been some focus at looking at how adversaries and competition inside social networks (e.g., Baldwin, Bedell, & Johnson, 1997) changes a network's performance and influences it.

This competition and rivalry was studied in classroom settings (Yang & Tang, 2003), and also became a crime prevention and understanding topic in the early 2000s with the September 11th terrorist attacks. Krebs (2002) was the first to look at the terrorist plot as a social network and to diagram it out and present it to the world as a social network between the terrorists. At the same time Klerks (2001) looked at organized crime and mapped out the social network and money flow networks of criminal organizations. While it is difficult to understand these networks, the ability to target particular nodes in the network and disrupt the entire network is a very powerful feature of social network analysis (Carley, Lee, & Krackhardt, 2002). These specific applications to crime and terrorism are just a small portion of work done in adversarial networks.

We use this foundation to analyze incoming intelligence reports about a mock terrorist plot and view the entire organization as a social network with leaders and subgroups.

Most work in the analysis of social networks takes a snapshot of an existing network and analyzes this network based on multiple centrality measures, distance measures, existence of power law patterns transitivity and clustering values. In this paper, however, we find that looking at a static view of the network is not as informative and useful as viewing how the network evolves over time. The evolution of a network is hard to analyze in most work due to difficulty in obtaining the data at different times or difficulty in observing the social network through its growth.

By using a social network that evolves from the communication of terrorists we can view the network from the beginning as it is seen through the intelligence reports about it. This allows us to look at standard social network measures mentioned earlier but with respect to how they change over time. This paper proposes these measures as being more interesting to analyze with respect to time, and also presents some inherent difficulties that exist in most social network analyses but that are not apparent until you view them with respect to time.

2. Adversarial Network Analyzer (ANA)

To best see these networks evolve and meet the requirements of the data set, this paper presents a graph visualization tool tailored to handling the simulated adversarial network. The Adversarial Network Analyzer (ANA) is a Java applet that allows users to input new connections about the graph and visualizes the state of the graph at all-time intervals. To provide powerful graph visualizations ANA is written on top of the Prefuse visualization toolkit (Heer, Card, & Landay, 2005). This feature-rich library provides a visualization library useful for displaying many aspects of datasets. The feature most commonly used in ANA is the use of graph visualization through Nodes and Edges. Prefuse handles all the calculations and work to layout and render the graph.

The application is a Java UI applet so that it could best be used by developers and analysts interested in network evolution or adversarial networks.

ANA 1.0 is planned to be used in the spring of 2012 by a security and risk analysis course taught in the College of Information Science and Technology at the Pennsylvania State University. The students who will be using this software will be studying simulations like the Shemanski (2011) dataset to understand an adversarial plot and identify it.

The UI shown in Figure 1 is broken down into four distinct sections, each providing a useful interface to managing the network graph. The various panels map to the supported tasks of ANA, described in Table 1, with some functionality moved into the menu.

Table 1 ANA 1.0 Supported Tasks

(a) Visualize graph
(b) Add nodes to existing graph
(c) Add edges to existing graph
(d) Modify existing edges and add more details
(e) Modify existing nodes and add more details
(f) Playback of graph expanding from the first node
(g) Save and reload graph
(h) Export state of graph to standard XML for mathematical analysis by ORA

An important part of the visualized data set is the requirement to handle unknown and incomplete data. As communications are intercepted and found by various intelligence organizations nobody knows the full layout of the cell network or the identities of everybody involved. The network has to be able to accept incomplete data and allow us to later fill in the blanks as more information becomes available. This is the reason behind allowing all information to be edited, and why nodes are added to the graph one at a time.

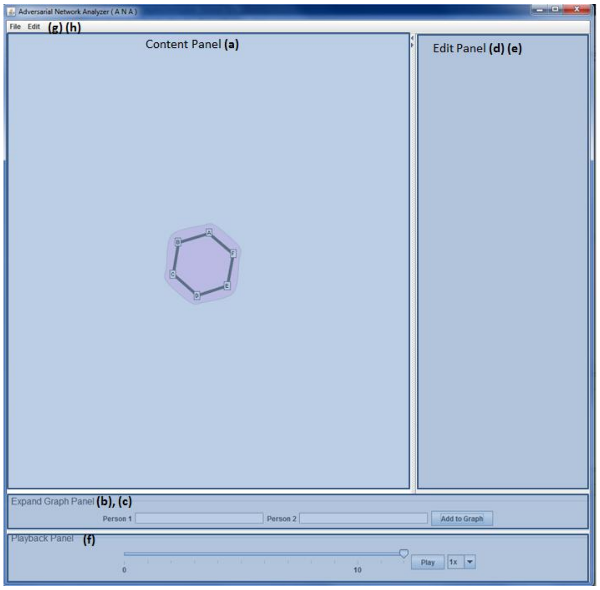


Figure 1 ANA 1.0 User Interface

Social networks do not just appear with hundreds of nodes and interconnections; they evolve slowly from nodes connecting to each other and new communications between actors showing up. We monitor this slowly from intelligence gatherings and intercepted communications.

ANA records all changes made in the graph, as edges and nodes are added or modified. This change detection and recording gives ANA the ability to play through the evolution of a graph from the beginning and animate it for the benefit of the user.

To run the mathematical analysis on any network created in ANA we built an exporting feature in ANA that allows the network to be output any time slice to a standard file format that tools such as the Organization Risk Analyzer, ORA (Carley & Reminga, 2004) can read. From here the calculations and analysis provided in Section 3 can be quickly computed.

3. Network Analysis

For the work in this section we entered the entire contents of the Shemansky (2011) simulation into ANA in order of how the intelligence reports appear. This simulation of 73 incident reports and 15 background reports were used to build the social network of the terrorist plot, as any analyst looking at the plot would see. We then exported the simulation to an ORA file format to compute the network statistics. The network created contained 30 nodes connected by 46 edges over a total of 117 time frames. We exported the state of the graph at intervals of 10 frames to visualize the time course of networks inside ORA.

Figure 2 presents this final network with labels and added color coding of the subgroups in the network.

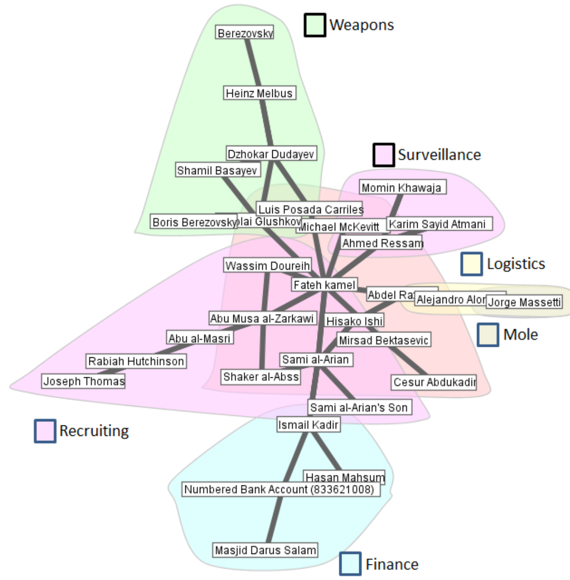


Figure 2 Final network of the Shemanski (2011) adversarial network simulation

3.1 Static Measures, Global and Local

In Table 2 we present the results of this analysis both on a global scale of the entire network, but also on a local scale where we look at particular agents and see how their position in the graph changes over time. The two players are Fateh Kamel, and Shaker al-Abssi who are the respective playmaker and leader of the plot.

Table 2 Static measures of the graph and key players

Measure	Graph Average	Fateh Kamel	Shaker al-Abssi
Degree Centrality	.053	.207	.034
Distance Centrality	.146	.305	.225
Betweenness Centrality	.048	.430	.012
Clustering Coefficient	.080	.042	.000
Distance	3.440		
Transitivity	.047		

The playmaker, Fateh Kamel, remains a more central node in the network than the average of the network and, more importantly, than the leader of the network.

This network is highly decentralized and spread out to protect the identities of its nodes. Compared to more traditional networks studied in the literature (e.g., co-authorship in various fields, World Wide Web, and movie actors) it has node degrees that are an order of magnitude smaller than some networks, and also clustering coefficients that are an order of magnitude smaller than previously studied social networks.

Distance in the network remains rather small both due to the size of the network and the connectivity of the playmaker. This network is much more similar to networks of things like the World Wide Web and power grids that are *non-social networks*.

Table 3 Comparison of Shemanski adversarial network with more standard social networks (Albert & Barabasi, 2002).

Network	Size	Average Degree	Average Distance	Clustering Coefficient
Shemanski Network	***30***	***1.53***	***3.44***	***.080***
WWW	153,127	35.21	3.10	.1078
Movie Actors	225,226	61	3.65	.79
LANL co-authorship	52,909	9.7	5.90	.43
MEDLINE co-authorship	1,520,251	18.1	4.60	.066
SPIRES co-authorship	56,627	173	4.00	.726
NCSTRL co-authorship	11,994	3.59	9.70	.496
Math. Co-authorship	70,975	3.9	9.50	.59
Neurosci. Co-authorship	209,293	11.5	6.00	.76
Power Grid	4,941	2.67	18.70	.08

3.2 Global Dynamic Measures

Rather than just comparing snapshots of the network, we analyze how it changes over time for various global measures of centrality and clustering of the network.

Figure 3 a, b, c presents the results for the change in centrality values over the time course of the network evolution. These values are already small, less than 50%, for all of the measures but present an interesting pattern of decreasing over the time of the simulation. Degree centrality and distance centrality show how both values decrease once the graph achieves a size of greater than 5 nodes, and then stabilize at a very small value. The deception forces inside this network keep it from becoming too centralized so that it may maintain its cell like structure remains hard to detect or infiltrate.

For the non-centrality based measures, clustering coefficient, transitivity, and average distance, presented in Figure 3 we see very similar patterns of the graph aiming to be more spread out and less tightly connected as more actors are brought into the network.

The clustering coefficient does increase between frame 40 and 50 due to a few connections developing. These connections bring the finance and weapons subgroups closer together so they can more effectively work inside of their subgroup. Over time this clustering coefficient does not continue to grow and the groups grow farther apart. These changes may be indicative of normal network growth or may be an artifact of this simulated network. In any case, the changes suggest that studying the time course of network growth can be interesting.

Distance across the network continues to increase over the entire time course of the network as more nodes are added to the far ends without connecting them to the center of the graph for quick communication paths. This keeps the two far ends of the network far apart and allows one end of the network to remain safe if anything were to compromise the other end.

Finally, transitivity only increases when weapons and finance subgroups become connected but decreases afterwards similar to the clustering coefficient. This shows that very few ties are created between triads of actors and instead the network chooses to communicate through the longer pre-existing chains of command and communication.

3.3 Local Dynamic Measures

Many of the values calculated in the previous sections can be calculated for particular actors. In Figure 4 we show a comparison between centrality and clustering of the two key players of the network, Fateh Kamel and Shaker al-Abssi. The important pattern seen in all of these graphs is that while both actors have low values for all of these network measures, the leader tries to remains less central and less visible compared to the playmaker. The leader, Shaker, is always looking to be more obscured by the surrounded network, while the playmaker, Fateh Kamel, is at times looking to grow more connections so that he can more quickly work with the various subgroups and leaders of those subgroups. Surprisingly, the leader's *distance centrality* in the graph becomes lower over time.

The betweenness centrality, degree centrality, and clustering coefficient for the playmaker actually increase in the graph as he becomes more tightly coupled to some of the people he is directing and organizing. This allows him to be effective, and yet leave the actual leader less detectable.

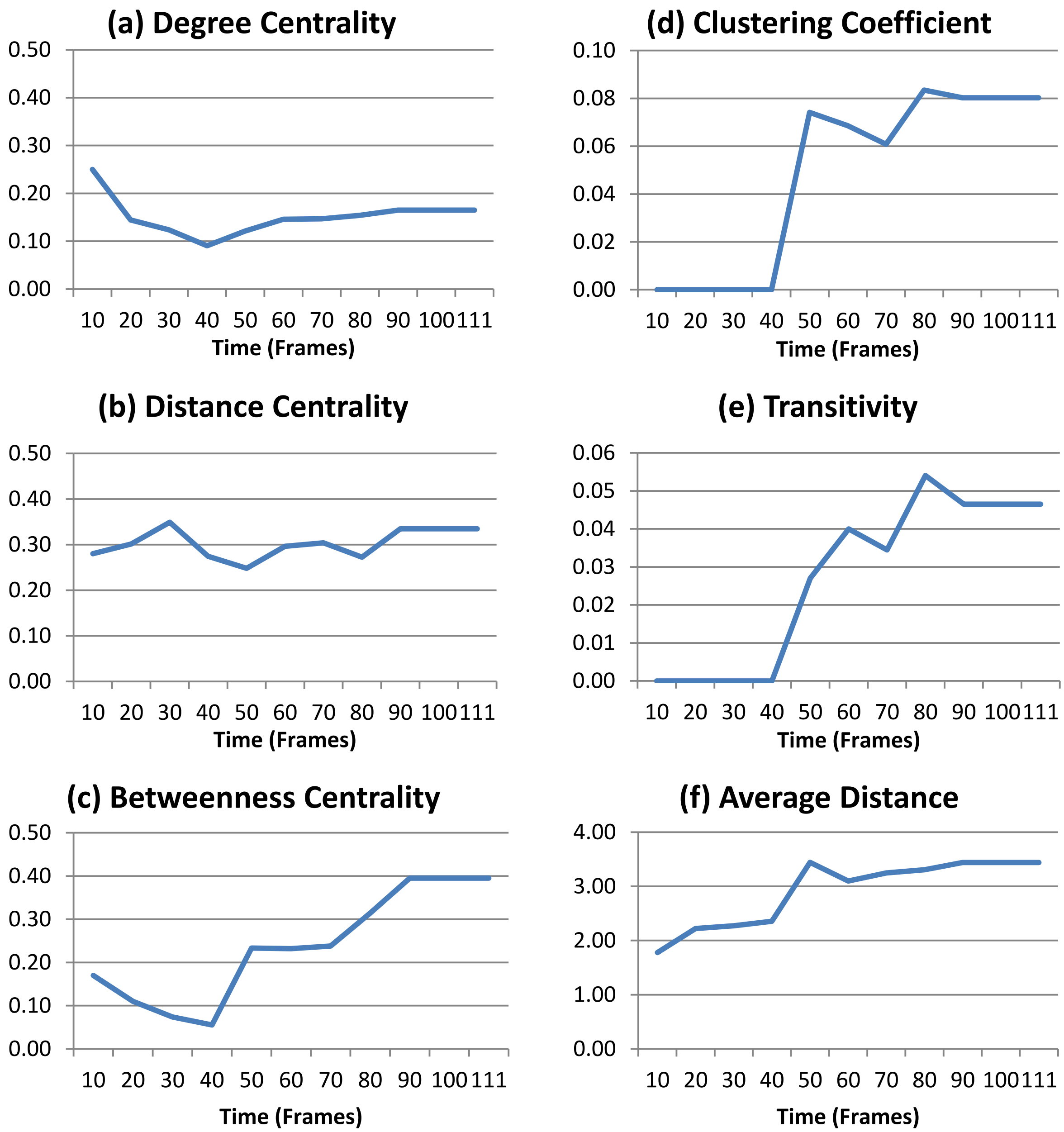


Figure 3 Global Dynamic Measures a) Degree Centrality b) Distance Centrality c) Betweenness Centrality d) Clustering Coefficient, e) Transitivity f) Distance of the entire network measures over the development period of the network.

3.4 Summary

Adversarial networks maintain a much smaller degree centrality than other networks. They are not interested in having each person be connected to as many other people in the network as possible. Each person is only connected to one, perhaps two other people that are strictly necessary to accomplish tasks. Additionally the clustering coefficient is small to minimize triangles of connections between actors.

In terms of dynamic measures, these networks do not follow a simple pattern of increasing connectivity, centrality, and clustering. The values actually decrease over the time of the network, and the average distance between members of the network increases. The networks push to be more spread out and increase the distance between recruits with high risk of being compromised and the leaders and playmaker of the network that carry out key actions.

The leader of such a network remains hidden from almost all members of the network. While this dataset does not specifically say how many people inside the network could identify the leader, our information and analysis of this shows that almost nobody would be able to identify the leader. Shaker al-Abssi remains nearly invisible from all but one or two members of the network. These members are the playmaker who actually carries out all of his orders and missions, and an additional buffer person. These covert strategies allow the playmaker to be the most visible person in the network. If he was to be compromised, he could be easily replaced by another member to carry out the orders of the leaders and bring the various functions together for a successful plot. The leader knows operational details of the group but does not actively use the existing links in the network.

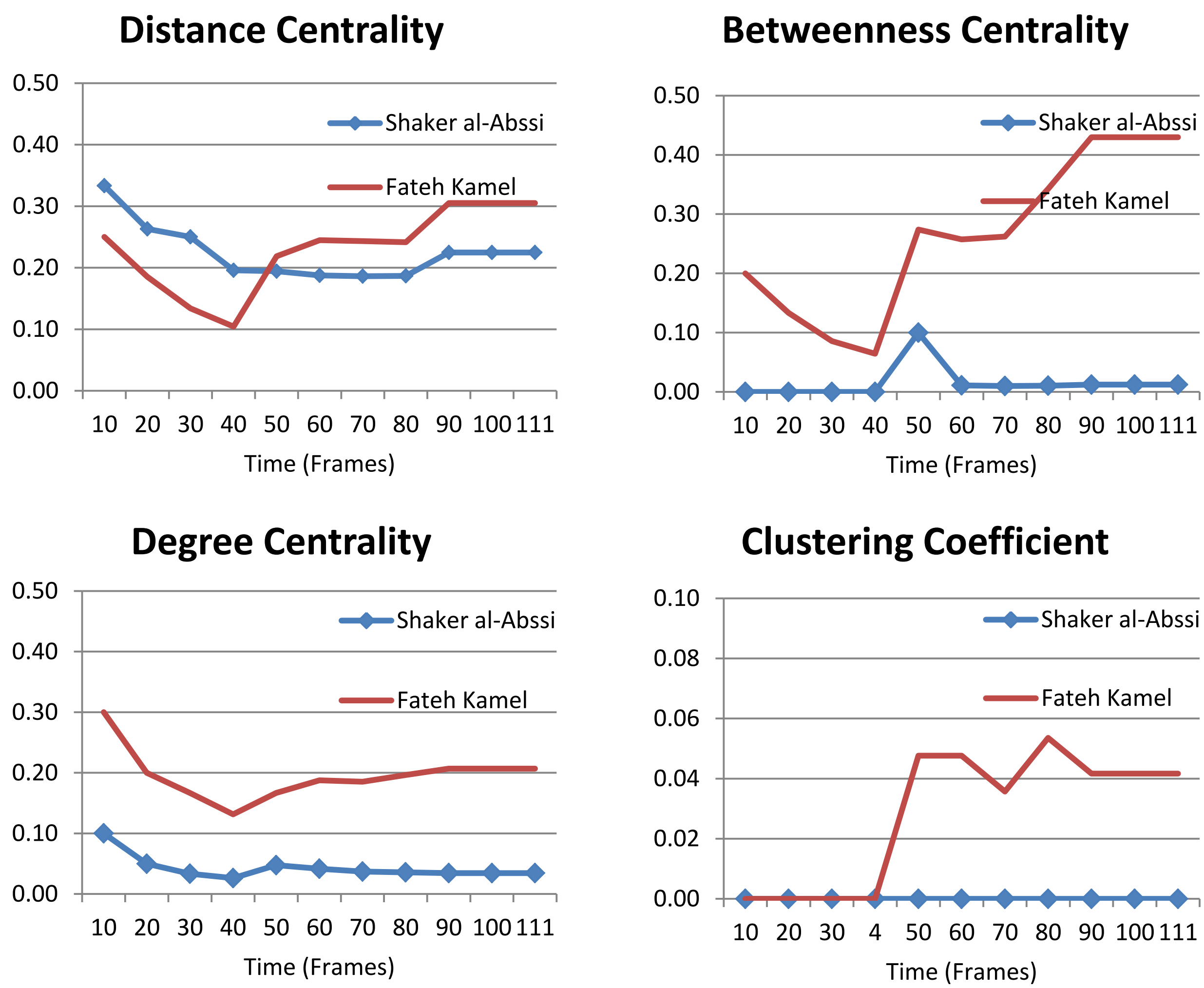


Figure 4 Local Dynamic Measures a) distance centrality b) betweenness centrality c) degree centrality d) clustering coefficient of the key players in the network measured over the development period of the network.

4 Conclusions

This paper introduced a new way to look at social networks, in particular adversarial networks and to analyze them for new patterns. These networks are seen through the eyes of a new animated visualization tool, ANA, that can build the network as information about its actors and connections emerge. One such network, the Shemanski (2011) simulation of an adversarial plot, is visualized through ANA, and using ANA's interface to standard file formats it is mathematically analyzed through social network measures.

In a time-wise analysis of the social network for two important actors, this analysis of these two actors shows additional differences between them. The leader, as expected, avoids building more connections and rather allows himself to be less and less central to the network as the network evolves. He is always less central and connected than the playmaker. The playmaker while not being heavily connected must build a number of connections between actors so the functional subgroups can work together.

Time-based analyses of adversarial social networks show concretely how these differ fundamentally from normal social networks. The network does not evolve to be more connected, nor does it evolve to be more central. The actors remain spread far apart so that each subgroup is separated and protected from problems that may occur in other parts of the network.

4.1 Limitations

Social Network analysis of these types of networks is a challenging task due to the nature of the networks. The network itself is adversarial and remains covert or tries to hide its underlying structure to improve its own performance. This causes error in the data that is obtained about the network and can complicate the analysis of such a network.

The observation methods used to record and look at social networks suffer from an inherent lag that can cause error in the analysis. All analysis is done on the network evolution of when we observe connections to be created. This is an analysis of our understanding of the social network rather than an analysis of the underlying evolution of the social network. Not all of the connections that are appearing through communications between actors are the first interaction between them. Many of these connections could have been formed days, months, or even years earlier but only been called into action when we observed it.

Inherent differences between our view of the network and the underlying structure of the network presents a source of error and remains as something to be looked at in future time-based analyses of social networks. Even work that does not study adversarial networks suffers from such a lag. Connections on popular social networks (Facebook, Twitter, LinkedIn) are not formed in a vacuum and are usually representative of an earlier interaction between actors. The same limitation applies to studies of publication networks that are frequent in network science. These publication databases suffer from a lag between when those researchers met each other and began sharing ideas and working together and the time when a collaborative paper is published.

4.2 Future Work

ANA in its current state does a good job of fulfilling a use case for students and intelligence analysts with a very simple interface and easy to use features. Future improvements can add more strength to ANA by providing support for more types of networks. The three additional types of networks that should be supported by future versions of ANA include networks with positive and negative relationships, directional relationships, and multi-mode networks (Qiu, Ivanova, Yen, Liu, & Ritter, 2011) to support the inclusion of events, and multi-person meetings. These features would allow more complete modeling of the interactions of an adversarial network but would have to be carefully implemented as to not overly complicate the interface and visualization of the network.

5. Acknowledgements

This work was supported by a grant from the Defense Threat Reduction Agency (HDTRA1-09-1-0054). We would like to thank Don Shemanski, Chris Dancy, and Jonathan Morgan for their expertise and help with this work.

6. References

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Review of Modern Physics, 74,* 47-97.

Anthonisse, J. M. (1971). *The rush in a graph.* Amsterdam: Mathematische Centrum.

Bakshy, E., Simmons, M. P., Huffaker, D. A., Teng, C.-Y., & Adamic, L. A. (2010). The social dynamics of economic activity in a virtual world. *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media.* Washington DC, 1-10.

Baldwin, T. T., Bedell, M. D., & Johnson, J. L. (1997). The social fabric of a team based M.B.A. program: Network effects on student satisfaction and performance. *Academy of Management Journal, 40*(6), 1369-1397.

Barabasi, A.-L., Jeong, H., Nelda, Z., Ravasz, E., Schubert, A., & Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A 311*, 590-614.

Beauchamp, M. A. (1965). An improved index of centrality. *Behavioral Science, 10*, 161-163.

Carley, K. M., & Reminga, J. (2004). *ORA: Organization Risk Analyzer.* Carnegie Mellon University, School of Computer Science, Institute for Software Research International, Technical Report.

Carley, K. M., Lee, J.-S., & Krackhardt, D. (2002). Destabilizing Networks. *Connections 24(3)*, 79-92.

Freeman, L. C. (1979). Centrality in Social Networks: I. Conceptual Clarification. *Social Networks, 1*, 215-239.

Heer, J., Card, S. K., & Landay, J. A. (2005). Prefuse: A toolkit for interactive information visualization. *SIGCHI Conference on Human factors in Computing Systems* (pp. 421-430). ACM.

Holland, P. W., & Leinhardt, S. (1971). Transitivity in structural models of small groups. *Comparative Group Studies, 2*, 107-124.

Holland, P. W., & Leinhardt, S. (1972). Some evidence on the transitivity of positive interpersonal sentiment. *American Journal of Sociology*. 72, 1205-1209.

Klerks, P. (2001). The Network paradigm Applied to criminal Organizations: Theoretical nitpickign or a relevant doctrine for investigators? Recent developments in the Netherlands. *Connections 24(3)*, 53-65.

Krebs, V. E. (2002). Mapping networks of terrorists. *Connections 24(3)*, 43-52.

Moreno, J. L. (1978). *Who shall survive?: Foundations of sociometry, group psychotherapy, and sociodrama.* Beacon, NY: Beacon House.

Newman, M. E. (2003, June). The structure and function of complex networks. *SIAM Review, 45*, 167-256.

Qiu, B., Ivanova, K., Yen, J., Liu, P., & Ritter, F. E. (2011). Event-driven modelling of evolving social networks. *Int. J. Social Computing and Cyber-Physical Systems, 1*(1), 13-32.

Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika, 31*, 581-603.

Shemanski, D. R. (2011). *Stop the Terrorists! Team-based Simulation of an International Terrorist Plot to Acquire and Use a Weapon of Mass Destruction.* ACS Tech Report 2011-1.

Yang, H.-L., & Tang, J.-H. (2003). Effects of social network on students' performance: A web-based forum study in Taiwan. *Journal of Asynchronous Learning Networks, 8*(3), 93-107.

Author Biographies

RAZVAN ORENDOVICI is a graduate of the Master's program in Computer Science and Engineering from the Pennsylvania State University and is now working at Microsoft.

FRANK E. RITTER is a professor in the College of Information Sciences and Technology at the Pennsylvania State University.

Soldiers, robots and local population - modeling cross-cultural values in a peacekeeping scenario

Saad Ahmad Khan
Taranjeet Singh Bhatia
Ladislau Bölöni
Dept. of Electrical Engineering & Computer Science.
University Of Central Florida
4000 Central Florida Blvd, Orlando FL 32816
skhan@eecs.ucf.edu, tsbhatia@eecs.ucf.edu, lboloni@eecs.ucf.edu

Keywords:
Human Robot Interaction, Social and Cultural Modeling

ABSTRACT: *We consider a near-future peacekeeping scenario, where a group of soldiers of various ranks and a robot interact with the local population. The goal is to quantify, analyze and predict the public perception of the soldiers and the robot. Instead of integrative statistical approaches, we develop a model which traces individual interactions. Our model assumes that human beings are considering collections of concrete and intangible values which are not, in general, directly and linearly convertible into each other. We argue that satisfactory modeling accuracy can be achieved by restricting the considered intangibles to a small set of* ***culture sanctioned social values****. For these values, the culture provides a name, calculation methods, as well as associated rules of conduct. We validate our model by comparing the predicted values with the judgment of a large group of human observers cognizant of the modeled culture. We use the model to evaluate the tradeoffs between various long term strategies to maintain security as well as to increase the trust and goodwill of the local population.*

1. Introduction

Soldiers on peacekeeping missions need to balance their own security and military objectives with the need to maintain friendly relationships with the local population. Our goal is to create a quantitative, operational model of the ways in which various actions taken by the soldiers and robots, as well as the members of the local population impact their respective cultural values and perceptions of each other. Some of the obvious challenges of this work include:

- The difficulty to assign numerical metrics and calculations to values dependent on social, cultural and personal perception.
- The need to consider the interaction between multiple players, some of them individuals (soldiers, members of the local population, the robot) while others groups of people (e.g. the participants in a crowd).
- The need to consider the evolution of values (such as gaining of trust) over a longer amount of time and series of interactions while simultaneously considering the fact that single, individual interactions can also have a long lasting impact.

Although the literature on cultural interactions is vast, most models are descriptive in nature and do not generate an *operational* model. Even when explicit numerical values are given (such as in Hofstede's models (Hofstede et al., 2010) the values are averaged over the populations and over specific situations.

In contrast, our model aims to provide automated analysis of specific scenarios with individual participants, and it needs to make predictions not about general trends but about the ongoing scenario. The model had been designed to provide input to the decision making system of a robot (which can be autonomous or tele-operated). The system can be also used as part of a training or assessment tool.

2. The Market Checkpoint Scenario

To anchor our modeling work in a plausible real world scenario, we shall use as a running example a situation frequently encountered in peacekeeping missions. The scenario takes place at a military checkpoint at the entrance of a busy market. We assume the location to be a Middle Eastern country (although the scenarios would unfold roughly similarly in other parts of the world - with the necessary adaptations for the cultural specifics). The checkpoint is manned by a sergeant (S), a private (P) and a robot (R). A street vendor (V) takes advantage of the traffic slowdown by positioning its cart near the checkpoint at one of the four locations L1-L4 (see Figure 1) at which are at an increasing distance from the checkpoint. We are concerned with the interactions among these actors over the course of several weeks. Let us now informally describe the various values, considerations and possible actions which are at stake at this scenario through the point-of-view (POV) of the checkpoint team and the street vendor.

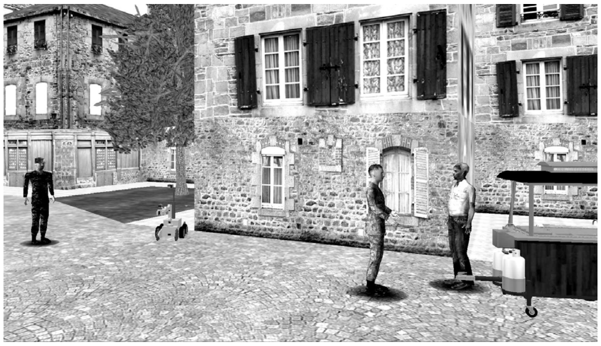


Figure 1. The private P is interacting with vendor V, with the sergeant S and robot R in the background.

The POV of the checkpoint team: the efficiency of the checkpoint and their personal security require maintaining a free and uncluttered area around the checkpoint. On days with a high alert level the perceived security is lower, and due to the more thorough inspections the traffic through the checkpoint slows down. The presence and location of the food vendor affects the security risks. Security threats can come from the street vendor itself, from creating additional crowding near the checkpoint, and from blocking lines of sight (either directly, or through the crowding).

The checkpoint team considers desirable to maintain good relations with the local population (in general), and the food vendor (in particular). Friendly interactions (informal conversations, exchange of gifts) increase friendship and trust. Unfriendly actions (such as ordering around or threatening) negatively impact the relations.

The POV of the street vendor: it is in the financial interest of the vendor to position its cart closely to the checkpoint. He will try to maintain friendly relations with the members of the checkpoint team, and will remember past interactions with the individual soldiers, appropriately reciprocating friendly or unfriendly behavior. Overall, the vendor will use its own cultural norms in assessing the behavior of the soldiers. However, he is aware of factors such as high alarm level (which can mitigate a specific intransigence from the checkpoint team).

3. Modeling Social Values

Our model assumes that the agents explicitly maintain a collection of *values*. These values are *visible* (as opposed to hidden), and the agents can explicitly quantify them if requested. The values are not independent, but are neither, in general, directly and linearly convertible into each other. The actions of the agent and external events change these values through linear addition or subtraction. The vector of these values determine the utility of the agent, through a non-linear utility function.

We group the values into *concrete* values such as financial worth or time as well as *intangibles* such as dignity and politeness. Concrete values have a rigorous definition, come with their native measurement units (e.g. dollars or euros for financial worth, seconds or minutes for time) and they are easily measurable.

Our approach limits the intangible values to *culture sanctioned social values* (CSSVs). We say that a culture *sanctions* a value if it provides for it a *name* and an evaluation *algorithm*. Cultures expect their members to continuously evaluate these values and to obey *rules of conduct* which depend on these values. A person can know more than one culture, and simultaneously evaluate values according to multiple cultures. However, evaluating the CSSVs can be a significant cognitive load, and busy people might not necessarily perform highly detailed evaluations of their ongoing environment. Similarly, there is no guarantee that everybody would obey the rules of conduct associated with CSSVs. Depending on how attentive is *the agent* in the evaluation of the values, and the level of immersion in the culture, an agent might or might not be aware of the transgression.

Different persons, educated in the same culture, would evaluate the values similarly. This has the important implication that in an environment where agents are in the presence of peers and general public, the agents are able to evaluate the CSSVs from the perspective of their peers and the general public. This evaluation can be normally made by evaluating the *epistemic state* of the subjects. An action witnessed by the agent changes the subject's perception of the value only if he *knows* about the action. The rules of conduct associated with a value usually extend to the peer / public values as well.

The algorithms provided by the culture for the calculation of values are obviously not numerical: rather, they rely on certain *keywords* to identify the gradations among the values (e.g. courteous → polite → neutral → rude → offensive). In our work we shall map these values to a scale of 0 to 1.

4. Analysis of the market checkpoint scenario

Let us now analyze and model our scenario using the CSSV model. We shall use the following collection of values:

- **Financial worth (V):** the income of the seller. It is dependent on the location, scaled by the traffic of the given day, and limited by the maximum amount of clients the seller can handle. It is measured in the local currency. It is only relevant to the seller.
- **Perceived security level (S, P, R):** is a metric of the level of threat as perceived by the soldiers. It depends on the alarm level, on the level of traffic, and the crowd created by the vendor.
- **Dignity (S, P, V):** The perception of the personal dignity by the soldiers and the vendor. The two parties apply different evaluation algorithms. The soldiers use a generic Western cultural model adapted to their status as soldiers ("being defied on an open order decreases dignity''). The seller uses its own cultural model - for the actions of this scenario, for instance involves that ("being ordered around decreases dignity", "declining an offered gift is an offense").
- **Politeness (S, P, V)**: The perceived politeness level of the soldiers and the vendor (with appropriate, culture and status specific evaluation algorithms).

4.1. Beliefs and public perception

The impact of an action on a CSSV is not a constant. Rather, it is modulated by the beliefs of the agent about specific aspects of the current context. A culture requires its members to maintain these beliefs as accurate as possible - the correctness of beliefs is necessary for the culture to operate as expected. Nevertheless, it is quite possible for an agent to have incorrect beliefs, especially in inter-cultural exchanges, when the agent might misinterpret the social signals (computers are especially bad at this, see (Vinciarelli, 2009)). As agents will act and calculate CSSVs according to the beliefs, we need to trace the belief values even when they are not correct. If an agent considers another one a friend, it will act accordingly and judge the actions of the other agent in this context, regardless of the fact that if the friendship is mutual or not.

In the agent literature, the beliefs of the agent are frequently considered to be a "model of the world". Creating such a model, for human participants, is clearly impossible. We argue, however, that by carefully choosing a small number of numerical belief values we can adequately model the influence of beliefs on the CSSVs.

Beliefs are higher level conscious judgments, and we posit that they are less subjected to the phenomena *psychological adaptation* than the values. For instance values such as politeness or dignity perception will tend to return to their average values over time spans of days. Beliefs, however, evolve more slowly, and they do not have natural trends towards average values. This does not mean, however, that beliefs are not affected by time spans without other actions - for instance, the perception of friendship might diminish in the presence of long spans of time without actions reconfirming this friendship.

We model the agent's beliefs using the Dempster-Shafer theory of evidence (Shafer, 1976; Yager, 1987) in the following way:

- The agent's current beliefs are fully encoded in the mass function - no previous evidences are remembered.
- The incoming evidence is be weighted by significance.
- For every incoming piece of evidence, the belief is updated using the standard Dempster's rule of combination (conjunctive merge).
- The value for the positive belief is used as the indicator of the belief.

Although, in general, the semantics of the Dempster-Shafer model is controversial, the results obtained with this model represent a good match to our intuitive understanding of the scene -- which, in fact, is what it is exactly what our objective was. We do not want the real probabilities of the events, rather to simulate the algorithms used by humans to maintain their beliefs.

We will use the following beliefs in the modeling of the checkpoint scenario. As with CSSVs, beliefs can be perceived from the self, peer or public perspective.

$\boldsymbol{B^{SPR}_{threat}}$ the soldiers belief that the vendor itself represents a threat (this does not include the belief that the congestion created by the vendor's presence can represent a threat). The perceived threat level starts up at a constant value, dependent on the soldier's training and personal perception. In general, the passing of time and human interactions decrease this belief. This belief affects the soldier's judgement of the security level function of the vendor location.

$\boldsymbol{B^{V\rightarrow x}_{friend}}$ the vendor's belief that the soldier x is a friend. Friendly actions (casual conversation, exchange of gifts, requests delivered with high mitigation level, lenience in accepting reactions to commands) increase the friendship belief. Actions which are considered rude (unmitigated commands, refusal of gifts) decrease the belief of friendship. The belief also decreases (albeit more slowly) in the absence of friendship maintenance actions (e.g. casual conversation).

4.2. Action repertoire

We model the possible scenarios using a series of possible actions. An action is performed either by a single actor (e.g. the vendor V moving from L1 to L3) or is the interaction between an actor and a recipient (the vendor V giving a gift to sergeant S). From the point of view of our model, the actions are fully described by their impact on the values of the actor and (if applicable) the recipient. Our modeling approach here is to define a relatively small number of actions, but to characterize them with parameters which describe, for instance, the destination of a movement or the verbal style in which a request or command is delivered. These actions are listed in Table 1.

	Action	Actors	Targets	Param.
A1	moves	V		Location
A2	declines-to move	V		Loudness, Offensiveness
A3	offers-gift	V	S,P	
A4	initiates conversation	V,S,P	V,S,P	
A5	accepts-conversation	V,S,P		
A6	orders-to-move	S,P,R	V	Loudness, Offensiveness
A7	passes-order	S,P	P, R	
A8	accepts-gift	S,P	V	
A9	declines gift	S,P	V	Offensiveness
A10	pushes	S,P,R	V	Loudness, Offensiveness
A11	overnight	S,P,R,V		

Table 1. Possible actions for the participants in the Market Checkpoint scenario (with specific possibilities for actor and target)

4.3. Case study: the impact model of action A6

One of the most critical and interesting actions is A6, where the one of the members of the checkpoint team (S, P or R) requests the vendor V to move the cart to a farther location (which is against the financial interests of the vendor). The request can be delivered in a number of different manners, which the impact on the values both of the request, and the possible responses (which can be A1 or A2).

We describe the manner in which the request is delivered through a parameter specifying the *mitigation level* of the order - according to the classification recently popularized by Malcolm Gladwell (Gladwell, 2008)[1]. To the six mitigation levels discussed by Gladwell, which culminate in command, we add three more levels which model the threat of a physical action and actual physical actions, respectively.

Note that the values in the table have been calibrated (using a survey) from a Middle Eastern perspective. Certain cultures such as Korean or Japanese, would put a significantly higher penalty on unmitigated speech. On the other hand, Northern European cultures would put no penalty on direct speech (and high levels of mitigation would probably be incomprehensible).

Name	Example	$P_{S/P}$	D^V
L1: Hint	Seems like the tree at position X would provide you with a better shade	1.0	1.0
L2: Preference	I would prefer you to use the position X today.	0.81	1.0
L3: Query	Shouldn't you be moving to position X?	0.68	1.0
L4: Suggestion	You should push the cart to position X.	0.56	0.91
L5: Obligation statement	You must move the cart to position X.	0.44	0.73
L6: Command	Move to position X!	0.36	0.63
L7: Threat of physical action	Move to position X or else I'll take action!	0.22	0.49
L8: Minor physical action	Pushing the cart manually away	0.11	0.28
L9: Major physical action	Taking the vendor in custody	0	0

Table 2. The impact of action A6 on the politeness of soldiers S or P and the dignity of the vendor using various levels of mitigated speech

These values are only the starting point for the calculation of the impact, which is further qualified by the performing agent, the relationship to the target, as well as the loudness of voice (which affects the knowledge of the bystanders of the interaction). To illustrate the type of expressions we reach, the peer-politeness of the sergeant due to action A6 is:

$$f(s5, a6)\ v = -H(x - 5)\ e^{0.1(x+y)}$$

while the effect on the dignity of the kebab-seller due to action A6 is:

$$f(s3, a6)\ v = -H(x - 4)\ e^{0.1(x+y+z)}$$

where x is the level of mitigated speech, y is the loudness level and z is offensiveness. Here, H() is the Heaviside step function, which is zero for negative and 1.0 for positive values.

A special situation applies when the actor of action A16 is the robot. The robot is not expected to know the subtleties of polite conversation, thus its use of direct command mode carries less offense - and its own politeness is irrelevant and not measured. This fact opens interesting possibilities for action strategies from the point of view of the team.

5. Survey based calibration of the model

Our model relies on the fact that the culture enforces a uniform method to calculate each CSSV. From this, it results that there will not be a significant difference between the peer-CSSV of the external observer and the self-CSSV of the direct participant, as long as they

[1] Note however, that similar ideas are present in the literature for a long time - e.g. in Brown and Levinson's politeness model (Brown, 1987)

are immersed in the same culture. This means that we can validate (and, if necessary calibrate) the action impact models by performing a *survey* in which persons cognizant with the respective culture will judge the impact on the social values from an external, peer-perspective.

5.1. Representativeness of the survey

One of the important considerations is the representativeness of the survey: are the results of the survey representative of the CSSVs of the target population? It is well known that many academic surveys suffer from the problem of using respondents who are in many ways divergent from the general population and are, in certain ways, "weird" (Henrich et al., 2010).

In the following we will discuss some of the obstacles we perceive in the representativeness of our results.

- The culture of the survey takers (Pakistan) might not be an exact match of the target culture. This is an unavoidable bias - for a perfect localization, one would need to use respondents from the exact geographical location we model.
- There might be a possible misunderstanding between the culture sanctioned values covered by the specific names. Our modeling target was a hypothetical, Arabic speaking Middle-Eastern environment. Our respondents have been primarily Urdu speaking, with a good knowledge of English, and many with at least some level of Arabic. We are confident that the use of English names, together with the Urdu and Arabic translations, have provided a sufficiently clear definitions of the values considered (see Table 3 for some of translations used).
- The distorting factor of social class: the survey subjects have been drawn from significantly higher social strata (students, engineers, doctors) than the average composition of the market we considered. Our conjecture is that people can accurately adapt their peer-CSSV assessment to the social strata and power positions of the actors in a narration.
- The impact of persons cognizant of multiple cultures. Many of the respondents have received some level of Western or Western-style education. It is to be determined whether this impacts their evaluation of the CSSVs – *i.e.* would they judge according to a western cultural model or can they estimate from the local perspective (as instructed). Our conjecture is that people cognizant of multiple cultures are able to evaluate separate CSSVs (within the limit of the cognitive load they can handle). Then, they decide which CSSV dependent rules of conduct apply in the current situation (which might be a combination of rules), then plan their actions in function of (not necessarily in obeisance to) these rules. This behavior model implies that even people who do not follow rules according to these CSSV settings, will still be able to calculate them.

5.2. Survey results

The methodology of the survey was as follows:

- The participants were presented with the scenario in a story-board style, with screenshots and explanation of the ongoing action.
- The participants scored the value of the perceived social value from the point of view of the seller (answering of questions of the type: rate the perceived politeness of the X on a scale of 0 to 10). The storyboard referred to CSSVs in English, with appropriate Urdu and Arabic translations provided (see Table 3).

The participants were 91 persons from various regions in Pakistan. While space limits us from analyzing the full output of the survey here, Figure 2 shows a representative case. The figure shows the histogram of answers for the public and peer politeness values for action A6(6, 5) - order to move using mitigation level 6 (L6) and moderate voice level and A6(1, 5) using maximally mitigated speech (L1). The graph shows that there is a remarkable consistency in the estimated CSSV, but also some level of distribution around mean values.

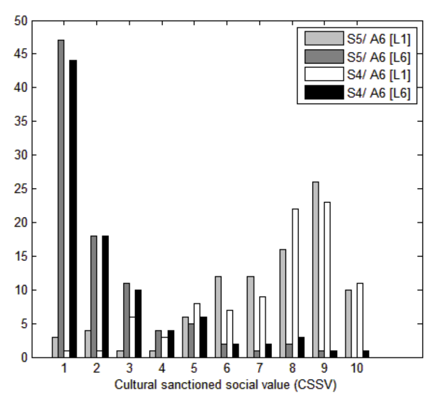


Figure 2. The survey histogram for public politeness [S4] and peer politeness [S5] in view of the vendor when the sergeant performs action [A6] (order to move)

Social Values	Urdu	Arabic
Politeness	مہــذب , شائســته	التهـــذيب والســـلوك
Dignity	وقـار, عـزت	احـــترام الذات
Friendship	دوسـت	صــداقة
Security	محفــوظ	أمن

Table 3. Names of CSSVs in English, Urdu and Arabic colloquial terminologies

The validation of the effect of actions on CSSVs have been done as follows. We have fitted normal distribution curves to the histograms of the values. Figure 3 shows these curves and vertical lines corresponding to the values which had been predicted by our model. While the match is not perfect, there is a very good correlation between the center of weight of the curve and the predicted values, which validates the predictive power of our model.

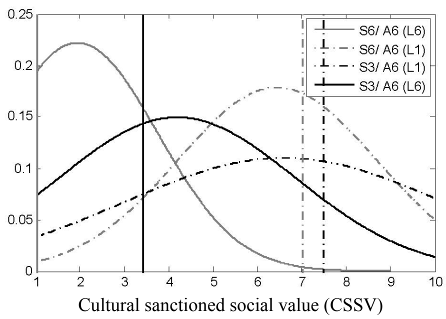


Figure 3. Normal distribution for dignity (S3) and friendship (S8) due to actions of order-to-move (A6) with mitigated level of speech L1 and L6

6. Experimental Study

In the following scenario, we describe the results of a simulation study, which traces the CSSV of the Market Checkpoint scenario in a 3D virtual world simulation. The CSSV action and belief models have been implemented using the YAES simulation environment (Bölöni, 2005). The participants have been placed in a 3D virtual environment based on OpenWonderland and collection of third part tools. As the objective had been to trace the CSSV of specific strategies, all the participants have been controlled by human players or scripted.

The modeled scenario represents instances of the Market Checkpoint scenario and its associated actions over the course of 14 days. The scenario also models the existence of external factors beyond the control of the soldiers and population: we assume that a medium (orange) alert happens on Day 8 and high (red) alert on Day 12. In the model we also include action A11 (overnight), that would shift the peer politeness and dignity back to the normal value. We assume that over the weekend, action A11 happens which justifies the rational that a person's dignity is less affected as an accumulative results of bygone days. But the belief is still affected and it maintains the value over the course of interaction.

In the following we will the evolution of CSSVs for five possible strategies of the soldiers at the checkpoint.

1. *Strict rule following*. In this scenario, the soldiers consistently use unmitigated command language (L6) and do not react positively to social interaction openings by the vendor. As Figure 4 shows, this lack of human interaction is perceived as rude by the vendor and is propagated to the beliefs of the general population. The positive side of this scenario is that the perceived security level remains high. However, the perceived politeness is low, the vendor is offended in his dignity, and the public belief is that the soldier and the vendor are not friends. The vendor is incurring some level of financial losses as it will regularly need to occupy unfavorable locations.

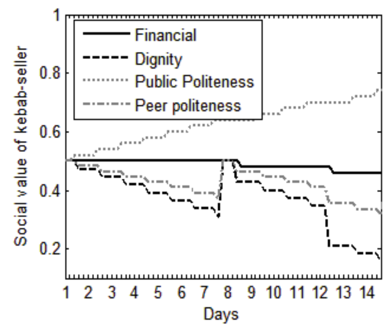


Figure 4: Strict rule following strategy.

2. *Consistent friendliness strategy*. In this scenario the soldiers consistently make choices to maximize their perceived friendliness. They achieve this by consistently using highly mitigated speech (L1-L2) when performing action A6, and responding positively to openings of social interactions. As at this mitigation level, the seller can often ignore the command, the scenario is financially advantageous to the seller. It leads, however to a low level of perceived security. Figure 5 shows that the rude behavior of vendor affect the dignity of the seller, and creates the perception of that the vendor is impolite. Naturally, that such a strategy is not a realistic option for a checkpoint due to low level of perceived security, and it is considered here only as a reference.

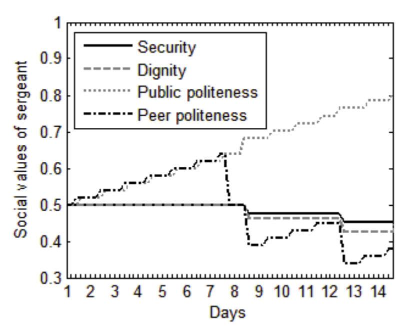


Figure 5: Consistent friendliness strategy

3. *Radical adaptation to alarm level.* In this scenario the members of the checkpoint team use consistent friendly behavior on days without alerts, with strict rule following behavior on days with orange and red

alerts (Day 8 and Day 12, respectively). The objective is to acquire a perception of polite and friendly behavior, while achieving a high perceived security on high alarm days. The simulation (some output values presented in Figure 6) shows, however, that the strategy is problematic. The overall politeness perception is lower than expected from the fact that the soldiers are following a friendship maximizing strategy on most days. The reason for this is that the cost of actions depends on the beliefs: commanding behavior from persons considered to be friends is more damaging to dignity than commanding behavior from a stranger.

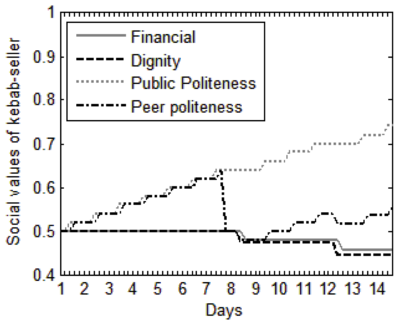


Figure 6: Radical adaptation to alarm level strategy.

4. *Adaptation by escalation*. In this strategy, the soldiers are trying to minimize cost of alarm level adaptation by starting with a high mitigation level on alarm days as well, but escalating the commands until the seller complies with the request.
Figure 7 shows the result of the scenario. We notice that the politeness loss is lower, however the necessity to escalate requests over time reduces the perceived security level.

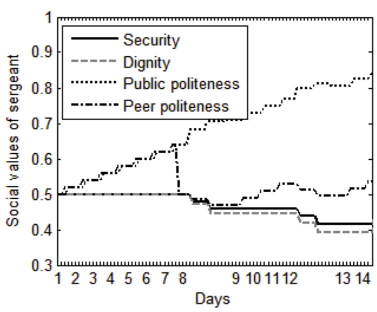


Figure 7: Adaptation by escalation strategy.

5. *Delegation of unpleasant tasks.* In this case the soldiers are following a strategy which tries to shift the socialization versus stability behavior by involving the robot in the interaction with the vendor. In most no-alarm days the soldiers will act socially and be, in general, permissive. In some days however, they will delegate the task to request the vendor to move to a farther position to the robot. The robot will deliver messages using non-mitigated speech (L6 and L7), and will (naturally) not participate in social interaction. In high alarm days, the soldiers will use the robot. The simulation of this model is shown in Figure 8. The results are in general positive (high level of perceived politeness, and high security).
Note, however, that the realism of this simulation is dependent on how accurate we are in inferring the transfer of perception of the robot to the soldiers. This depends on the perception of the autonomy of the robot – if the perception is that the robot is remotely operated by the soldiers, its social action impacts are directly transferred to them.

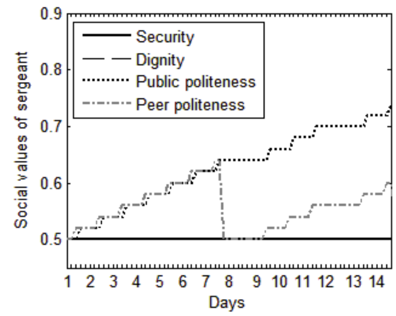


Figure 8: Delegation to the robot.

7. Related Work

There is an extensive body of work which analyzes population sentiment in reaction to the presence of foreign military forces. Many of these body of work assumes general, high level policies, involving overall directives, and had been done in the context of policy decisions, sociology and integrative simulations. Our work involves a direction which had been much less thoroughly investigated, which involves the action of individual soldiers, over the course of several weeks.

Famously, general Petreaus said that the american soldiers have to "drink a lot of tea" with local Afghan leaders, to establish normal relations. In recent years, a number of approaches, similar in spirit to ours, are working towards modeling individual interactions. In contrast to this approach which formalizes neurological theories of emotion, Miller et al. (Miller, 2008) propose to operationalize the Brown and Levinson politeness model (Brown, 1987). The implementation, the Etiquette Engine, is used to assess the politeness of a number of custom crafted social-interaction vignettes involving common culture but different rank (the interaction between a corporal and a mayor). The values were compared against the evaluation by human observers (unfamiliar with the Brown and Levinson model).

The discrete event social simulator DESS (Alt & Lieberman, 2010) provides a generic overview to social simulation with idea of embedding multi-agent system within a DES framework. The same authors explore the implications of applying TPB (Theory of Planned Behavior) and show the importance of using representative survey data for such action choice models (Alt & Lieberman, 2010). The authors of HCA (Holon cognitive architecture) model the culture as an epidemiology of representations and discuss the modeling of cultural frame shifting using the example of Swedish model (Young & Patterson, 2011). The authors in (Miller et al., 2007) propose an approach to produce culture specific, politeness-appropriate utterances and perceptions of utterances in a game setting, in aspect of culture specific language interpretation based etiquette generation

8. Conclusions

In this paper we described a method to model the impact of the actions of soldiers and robots through the model of culture sanctioned social values. Through a number of simulations involving a realistic near-future peacekeeping scenario, we had shown that is possible to develop a model which gives realistic predictions over a wide range of scenarios, and we have shown how the components of the model can be calibrated using surveys. One of the interesting findings of the research had been the importance that robots with real or partial / perceived autonomy can play in social interactions, a topic which is the focus of our future work.

9. Acknowledgements

The research reported in this document/presentation was performed in connection with Contract Number W911NF-10-2-0016 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory, or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation heron.

10. References

Brown, P. and Levinson, S.C. (1987). *Politeness: Some universals in language usage vol 4.* Cambridge Univ Pr.

Gladwell, M. (2008). *Outliers: The story of success.* Little, Brown and Company.

Henrich, J. A et al. (2010). The weirdest people in the world. *Behavioral and Brain Sciences*, 33(2-3) 61--83.

Hofstede, G. H et al. (2010). *Cultures and organizations: software for the mind.* McGraw Hill Professional.

Miller, C. et al. (2008). A computational approach to etiquette: Operationalizing Brown and Levinson's politeness model. *IEEE Intelligent Systems*, *23(4)* 28-35.

Miller, C. et al. (2009). Culture, politeness and directive compliance. *Proc. of the 5th Int. Conf. on Universal Access in Human-Computer Interaction*, pp. 568-577.

Shafer, G. (1976). *A mathematical theory of evidence vol 1.* Princeton university press Princeton, NJ.

Vinciarelli, A. and Pantic M. and Bourlard H. (2009). Social signal processing: Survey of an emerging domain. *Image and Vision Computing, 27(12)* 1743-1759.

Yager, R. (1987). On the Dempster-Shafer framework and new combination rules. *Information sciences 41(2)* 93-137.

Alt, J., & Lieberman, S. (2010). Modeling the Theory of Planned Behavior from Survey Data for Action Choice in Social Simulations. *Proc. for 19th conf. on Behavior Representation in Modeling and Simulation*, pp. 126-133.

Alt, J., & Lieberman, S. (2010). Agent Frameworks for Discrete Event Social Simulations. *Proc. for 19th conf. on Behavior Representation in Modeling and Simulation*, pp. 134-139.

Young, M., & Patternson, S. (2011). Modeling Cultural Frame Shifting. *Proc. for 20th conf. on Behavior Representation in Modeling and Simulation*, pp. 98-105.

Miller, C. et. al. (2007). *Proceeding for 16th conf. on Behavior Representation in Modeling and Simulation*, pp 26-29

Bölöni, L., & Turgut, D. (2005). YAES: A modular simulator for mobile networks. Paper presented at the *Proc. of the 8th ACM Int. Symp. on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (MSwim), pp. 169-173.

Author biographies

SAAD AHMAD KHAN is a Ph.D student at the Dept of Electrical Engineering and Computer Science of University of Central Florida.

TARANJEET SINGH BHATIA is a Ph.D student at the Dept of Electrical Engineering and Computer Science of University of Central Florida.

LADISLAU BÖLÖNI is an Associate Professor at the Dept of Electrical Engineering and Computer Science of University of Central Florida.

Effect of Decreasing Accuracy in the Temporal Processor for Attention Switches in a Complex Dual Task

Brian McClimens
Derek Brock
Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375
202-404-3798
brian.mcclimens@nrl.navy.mil, derek.brock@nrl.navy.mil

Keywords:
Attention, Human factors and Human-computer interaction, Human movement modeling Model comparison

ABSTRACT: *Research at the Naval Research Laboratory (NRL) has shown that the use of auditory cueing can dramatically improve operator performance in dual-task environments for which efficient task-switching plays a crucial role (Ballas, 1992, Brock, 2004, Brock, 2006). In order to better exploit the benefits of auditory cueing for the purpose of attention management in multitasking environments, the Navy desires a model-based understanding of the mechanisms driving human performance in these scenarios. Empirical studies utilizing a complex dual task and related cognitive modeling work developed with the EPIC cognitive architecture [5], have focused on understanding the methods subjects employ to effectively time their transitions between tasks. These models support the notion that time spent on the primary, relatively stateless, tracking task is regulated by state information retained from the secondary, radar task. However, the models do not sufficiently capture the benefits observed in conditions utilizing auditory cuing to assist in attention management. A minor modification to these models results in a dramatic change in model performance, provides insight into when and how auditory cues provide benefit, and raises questions about the methods used by the models to time attention switches between tasks.*

1. Introduction

A series of studies conducted at NRL, utilizing the Ballas dual task presented on widely separated monitors, have repeatedly shown a robust improvement in performance on both tasks when auditory cues are used to assist in attention management between the tasks *(Ballas, 1992, Brock, 2004, Brock, 2006).* In order to better understand the factors contributing to improved performance, cognitive modeling has been employed to examine viable strategies for directing attention in conditions with and without the presence of auditory cues.

In this dual task environment, proactive attention management is critical because the two tasks are presented on screens that are separated by a ninety-degree arc. Figure 1 shows a workstation similar to those used in empirical studies at NRL. The tasks utilize the far left and far right screens, while the center screen is left blank. As a result, subjects attending to either task cannot receive visual information from the other. In conditions for which no auditory cues are present, all attention switches between tasks must be self-directed, and cannot rely on external cues.

Cognitive modeling work at NRL has supported the notion that state information from one task could be used to determine the duration of an attendance to the other task (McClimens, 2011). Such information could be used to set an internal timer prior to switching tasks, and performance benefits could be realized without requiring a simulation of progression on one task while attending to the other. These models did not sufficiently replicate the benefits observed from the use of auditory cues to aid in attention management. This paper examines an adjustment to these models intended to address that deficiency.

Figure 1. Three-screen console configuration of the Common Display System, the new information workstation being acquired for the U.S. Navy's modernization program and next- generation surface ships. The described dual task utilizes the far left and far right monitors in a similar workstation.

2. Background

2.1 EPIC

The EPIC cognitive architecture (Kieras, 1997) has been used to build several models of this dual task in the past (Kieras, 2001, Brock, 2006, Hornof, 2010). The models in this paper are an extension of previous modeling work at NRL, and again use the EPIC architecture. These models also make use of a custom-designed encoder for the hostility property of blips on the radar screen, and a timing mechanism that regulates the amount of time spent on the tracking task between attendances to the radar task.

2.2 Ballas Dual Task

The Ballas dual task consists of a simple yet demanding, continuous tracking task in which subjects are required to follow the motion of a target object onscreen using a joystick, and an intermittent decision task loosely based on a radar display, in which subjects must classify incoming objects of three types as either hostile or neutral based on rule sets unique to each of the three object types. These two tasks are presented on monitors separated by a ninety-degree arc such that subjects focusing on one of the two tasks cannot receive visual information in their periphery for the other task.

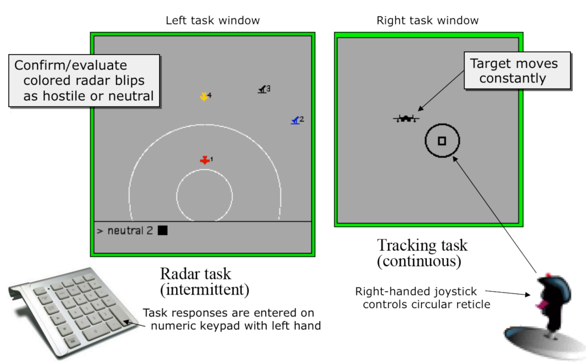


Figure 2. A depiction of the radar and tracking tasks.

The tracking task is presented to subjects as their primary task. Although there are neither complex decisions to be made nor critical events in the tracking task, it is a task that demands constant attention in order to perform well. Subjects control a reticle on the screen via joystick, and the performance criteria is simply the average distance between the reticle and a target object over the duration of the experiment. The target moves around the screen with quick, irregular movements that are difficult to predict. These movements are small enough that most subjects are able to track the target relatively well while attending to the task, but even quick glances to the radar screen incur a rapid drop in performance. As a result, the percentage of time spent attending to the tracking screen is a very good predictor of performance. As a general rule, subjects spend between seventy and eighty percent of their time focused on the tracking task depending upon the experimental condition.

The radar task consists of a series of classification judgments based on the motion of three types of objects, collectively referred to as blips. Over the course of a thirteen minute scenario, subjects are required to make sixty five classifications (one every 12s on average). The pace of activity changes throughout the scenarios, but in the studies this model is based upon, there are never more than five blips on the screen at any given time. Blips appear near the top of the screen, and move down towards the bottom of the screen over the course of approximately twenty seconds. When a blip initially appears on the screen, it is black and subjects are not permitted to enter a classification for that blip. When a blip has moved about halfway down the radar screen, it changes color to signify that the subject should enter a response. In experiment conditions that utilize auditory cues, the blip's color change is accompanied by an alert sound so that subjects are made aware of blips ready for classification even when they are attending to the tracking task.

2.3 Performance Measures

Recent research at NRL regarding the Ballas dual task has focused on the role that attention management plays in the performance of the radar and tracking tasks. The benefits of auditory cues have been measured primarily using three key performance measures: reaction times for blip assessments, the percentage of time spent on the tracking task, and the number of attention switches between the radar and tracking tasks.

The reaction times on blips in the radar task are defined as the amount of time that passes between a blip changing color and the completion of a response by the subject. These reaction times are more than a simple measure of performance on the radar task. Because blips often change color while a subject is attending to the tracking task, the reaction times increase if subjects fail to effectively manage their attention.

The number of attention switches between tasks, and the percentage of time spent on each task are measures of how much effort a subject is putting in to staying aware of blips on the radar task. In this setup, with the two screens set ninety degrees apart, attention switches are costly, and directly result in poorer performance on the tracking task. Although additional attendances to the radar screen should improve performance on that task, previous research has shown that when auditory cues are present to assist in attention management, subjects are able to improve performance on the radar task while making fewer attention switches and spending less time on the radar task.

3. Modeling

3.1 2009 data

Early models of the Ballas dual task at NRL suffered from a lack of precision in the data collected regarding the head movements of subjects. Data was recorded during an experiment in 2002 by hand. To record a head turn, the experimenter tapped buttons on a PDA, and time stamps were recorded to the nearest second. Though this process was sufficient for showing significant differences between the sound and no-sound conditions, a greater degree of resolution would be required for more sophisticated modeling efforts.

In 2009, a study was conducted using the Ballas dual task to evaluate the use of new presentation method for auditory cues. This provided the opportunity to collect more detailed empirical data regarding the allocation of attention between the radar and tracking tasks. Previous studies had reported the number of head turns subjects made during the experiment, but collection of data via a head-mounted tracking device was added to allow for more detailed analyses of individual attention switches between tasks. The head-tracking data allowed for the measurement of the durations of each attendance to a task, and allowed for an association between individual task attendances and the states of each task at that time. It was predicted that subjects would maintain an awareness of the radar task's state, and as a result would spend less time in episodes of tracking when there was more activity on the radar task. In other words it was thought that when a subject left the radar task to perform the tracking task, they made note of the current state of the radar task, and used that information to determine when they should return to the radar task. A subject who saw that there were no blips on the radar screen before attending to the tracking screen, would be likely to track for a longer period than they would if the radar screen had a large number of blips on it when they looked away to begin tracking. Data collected in a pilot study supported this prediction, and a model was created to test the impact of allowing state information from the radar task to guide the timing of attention switches between tasks.

3.2 Modeling Radar-Driven Task Switches

In order to test the effectiveness of using the radar task's state to guide attention switches, a model was created and run in three modes on two conditions. The two conditions were a "sound" condition, in which an auditory alert was presented whenever a blip was ready to be classified, and a "no-sound" condition in which no alert was used. In both conditions, blips would change color when they were ready for classification, so that a subject attending to the radar task at the time would be aware of the state change. In the sound condition, the audio alert allowed participants to be aware of this state change while attending to the tracking task as well. In each of these conditions the model was run in three modes that used progressively more information from the radar task to inform the temporal processor how much time should be spend on any given attendance to the tracking task. The first mode, referred to as 'mono', was a baseline in which the temporal processor was simply used to ensure that the model spent the same average time on the tracking task as was observed across all tracking attendances in the human subject study. The second 'dual' mode divided the tracking episodes into two categories: those in which tracking began while there were zero blips on the radar screen, and those in which tracking began with one or more blips on the radar screen. When blips were present on the radar screen, the model would spend less time on the tracking screen, in accordance with observed human behavior. In the third, 'multi' mode, four divisions of tracking attendances were made: one group for instances in which there were zero blips on the radar task as tracking began, one for instances with one blip on the radar screen, a third for instances with two blips, and a final group for instances with three or more blips on the radar screen.

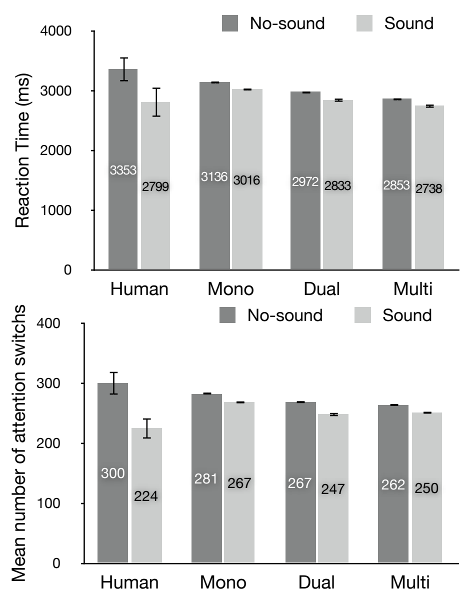


Figure 3. These graphs show improved reaction times for classification events on the radar task and a reduction in the number of attention switches between tasks as the model makes more use of state information from the radar task to regulate time spent on the tracking task. This data is reported in (McClimens and Brock, 2011).

As seen in figure 3, the models that made greater use of the radar task's state information to govern attention switches were able to perform better on the radar task, most notably with reduced reaction times for radar task classifications, and also showed a slight decrease in multitasking overhead, reflected by the decreased number of attention switches (McClimens and Brock, 2011). Unfortunately, while there was a small improvement for reaction times in the sound conditions, the differences between sound and no-sound conditions were not nearly as pronounced in the model as was observed in human data. This was also reflected in the number of attention switches made by the model. At the time, the reason for this discrepancy between the model and human data was unknown, but it appears that the reason can be found within the temporal processor used to determine the amount of time spent on each attendance to the tracking task. The model was designed to approximate the average amount of time spent on each attendance to the radar task. In the no-sound condition, this approximation was relatively close (3518ms observed, 3888 modeled), but the addition of auditory alerts did not affect the model as anticipated, and the disparity between the model and observed data increased (4620ms observed, 3986ms modeled). An examination of the distribution of the attendance times reveals an even more striking contrast between the model and observed data, as is shown in figure 4.

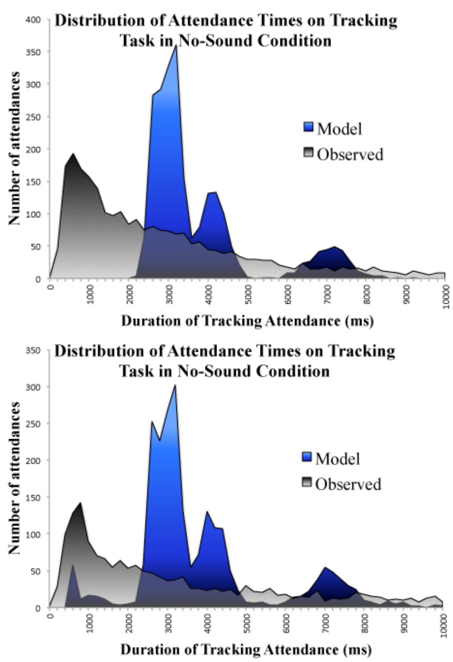


Figure 4. Compared to the empirical data, the model showed little variance in the amount of time spent on individual tracking attendances.

3.3 Performance Effect of Attendance Distribution

The model's variance in time spent on the tracking task was much narrower than the observed behavior. It was hypothesized that this characteristic of the model might be a key factor in the lack of distinction between the model's performance in the sound condition as opposed to the no-sound condition.

To see how the shape of this distribution can affect performance, consider the effect of an auditory cue on a subject attending to the tracking task. When a subject leaves the radar task to begin tracking, they make an internal estimate of when they need to return to the radar task. Our model works under the assumption that the desired return time coincides with a blip activation, when a blip changes color and is ready for a response. In the sound condition, this event is accompanied by an auditory cue. If a subject is accurate in their estimation, they will arrive on the radar task as the auditory cue sounds. If the subject returns early, the auditory cue will not have been presented. In these two cases, there should be no difference in performance between the sound and no-sound conditions. If the subject makes a poor time estimate, and would be late returning to the radar screen a performance difference results from the two conditions. In the sound condition, the auditory alert will interrupt the long tracking attendance, and prompt the subject to return to the radar task. In the no-sound condition, the subject will continue tracking, unaware that an active blip is on the radar screen, and their reaction time for classification events will suffer as a result. A distribution of attendance times on the tracking screen with less variance provides fewer opportunities for an auditory alert to be beneficial.

3.4 Adjusting Noise in the Temporal Processor

The internal clock used in EPIC's temporal processor is an implementation of a timing mechanism developed by Taatgen (Taatgen, 2007). This timer is based on a pacemaker-accumulator model, in which pulses are generated, and an accumulator keeps track of how many pulses have passed. These pulses are not evenly spaced, but rather grow gradually more distant as time passes. Three parameters govern the production of these pulses. The first is an initial pulse length. The second parameter, α, determines how quickly the pulse lengths grow. Each pulse length is on average α times the previous pulse length. The final parameter, β, determines variability within the internal clock. When each pulse's timing is calculated, noise from a logistic function determined by the current pulse length times the third parameter is added. The Ratkin et al. (1998) experiment was used as a benchmark task to find approximate values for these parameters, and values were estimated at 11ms for the initial pulse length, 1.1 for α, and 0.015 for β.

$$t_{n+1} = \alpha t_n + noise(M = 0, SD = \beta * \alpha t_n)$$

The initial model used these same values for the three parameters. As seen in figure 3.2, this results in a narrow distribution around the desired tracking durations.

In order to remedy the differences between the variance of tracking attendance times observed and those produced by the model, the β parameter was heavily modified. Assuming the new model would have a distribution of target attendance times similar to the old model, analysis showed that increasing the value of β from .015 to .2 would increase the variance enough to result in a much closer approximation to the distribution of tracking attendance times in observed data.

4. Results

The 'multi' mode of the old model was rerun with a β parameter value of .2 in both the sound and no-sound conditions. The resulting distribution of attendance durations is shown below in figure 5 alongside the distributions from figure 4.

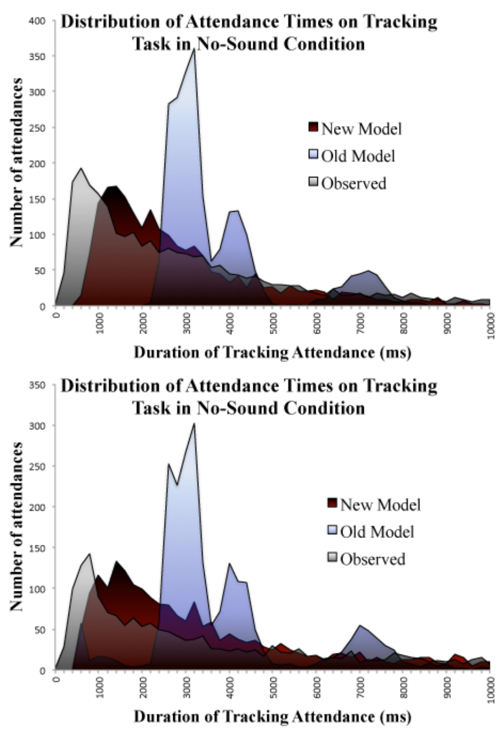


Figure 5. Adjusting the β parameter results in a distribution of tracking attendance times that better fits empirical data. Note that despite the apparent differences between the distributions generated by the two models, they use the same formula, with just one parameter modified.

The distributions from the new model peak slightly later than observed data, but increasing the β parameter does result in a much closer approximation of human performance in tracking attendance times. Note too in figure 7, that the mean in the no-sound condition, and the standard deviations are a better fit in the new model. Although the mean tracking time in the sound condition is further from the observed data, note that the difference in means between the no-sound and sound conditions is greater. The original model spent sixty-five percent of its time on the tracking task in both no-sound and sound conditions. The model with increased β spent sixty-six and sixty-seven percent of its time in the no-sound and sound conditions respectively. Empirical data shows subjects spending seventy-five and eighty percent of their time on the tracking task in no-sound and sound conditions.

	No Sound		Sound	
	Mean	St. Dev.	Mean	St. Dev.
Observed	3518	94.3	4620	140.2
Old Model (β 0.015)	3888	28.8	3986	39.1
New Model (β 0.2)	3534	109.5	3850	114.0

Figure 6. The mean and standard deviation of tracking attendance durations for observed data, original and increased β parameter, in no-sound and sound conditions.

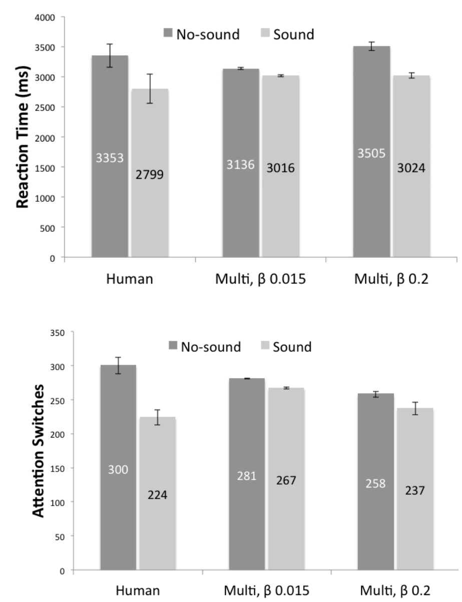


Figure 7. As predicted, the new model (the rightmost columns) shows an increased distinction between the no-sound and sound conditions as compared to the old model (middle columns).

5. Discussion

Increasing the β parameter for the timer in EPICs temporal processor from 0.015 to 0.2 resulted in a model that was a better approximation of the observed human performance data. Reaction times for classification events on the radar task showed a distinction between the sound and no-sound conditions, and the distribution of tracking attendance durations was more representative of the empirical data. Despite these results, one can question the validity of adjusting a parameter in the temporal processor's formula to fit a single case. In Taatgen's work (Taatgen 2007), it is noted that the behavior of the timing module is assumed to be task-independent, and as such the parameters should be determined by fitting performance to a single benchmark task and then be left alone.

Alternative methods should be explored to determine whether it is possible to fit the empirical data for tracking attendance durations without adjusting the β parameter. Recall that this model uses simplified state data from the radar task to bin tracking attendances into one of four categories (0,1,2 and 3+ blips on radar). For each category, there is a single target tracking duration. A narrow distribution of estimates of these times results in the sharp peaks shown in figures 4 and 5. A model may produce a more realistic distribution by using a continuous function to determine the desired tracking duration rather than discrete categories.

The presented model demonstrates that strategies utilizing state information from the radar task to regulate time spent on the tracking task, can benefit from the effects of auditory cuing. Previous such models that had shown a lack of distinction between no-sound and sound conditions can instead attribute this quality to a lack of fidelity to empirical data in the distribution of tracking attendance durations. Though a simple solution can be attained by increasing the β parameter in EPICs temporal processor, a more sophisticated solution should first replace the discrete function used in the model to determine the desired tracking attendance duration with a more nuanced continuous function.

References

Ballas, J. A., Heitmeyer, C. L., & Perez, M. A. (1992). Evaluating two aspects of direct manipulation in advanced cockpits. *Proceedings of ACM CHI '92: Conference on Human Factors in Computing Systems*, 127-134.

Brock, D., Ballas, J. A., Stroup, J. L., and McClimens, B. (2004b). The design of mixed-use, virtual auditory displays: Recent findings with a dual-task paradigm. *Proceedings of the 10th International Conference on Auditory Display*. Sydney, Australia.

Brock, D., McClimens, B, Hornof, A, and Halvorson, T. (2006). Cognitive models of the effect of audio cueing on attentional shifts in a complex multimodal dual-display dual-task. *28th Annual Meeting of the Cognitive Science Society*.

Hornof, A.J., Yunfeng, Z. (2010). Task-Constrained Interleaving of Perceptual and Motor Processes in a Time-Critical Dual Task as Revealed Through Eye Tracking. *Proceedings of ICCM 2010: The 10th International Conference on Cognitive Modeling.*

Kieras, D. and Meyer, D. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human Computer Interaction, 12, 391-438*.

Kieras, D. E., Ballas, J., & Meyer, D. E. (2001). Computational Models for the Effects of Localized Sound Cuing in a Complex Dual Task (*EPIC Report No. 13*). Ann Arbor, Michigan: University of Michigan, Department of Electrical Engineering and Computer Science.Hearn, J. and Mills, J. H. (2005). Finding the knowledge edge. CHIPS Magazine, U.S. Navy, vol. 13, no. 3, pp. 17-20.

McClimens, B. and Brock, D. (2011) Modeling Attention Allocation in a Complex Dual Task with and without Auditory Cues. *HCI International 2011 – Poster's Extended Abstracts, Part 1, 317-312*.

Rakitin, B.C., Gibbon, J., Penney, T.B., Malapani, C., Hinton, S.C., & Meck, W.H. (1998) Scalar expectancy theory and peak-interval timing in humans. *Journal of Experimental Psychology: Animal Behavior Processes, 24,* 15-33.

Taatgen, N. A., van Rijn, H., & Anderson, J. R. (2007). An integrated theory of prospective time interval estimation: The role of cognition, attention, and learning. *Psychological Review, 114(3), 577-598.*

Who says it best? A Comparison of Four Different Dialog Management Systems

James Hollister
Shane Parker
Avelino Gonzalez
Ron DeMara
Intelligent Systems Lab
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816
JHollister@isl.ucf.edu, Shane@isl.ucf.edu, Avelino.Gonzalez@ucf.edu, Ronald.Demara@ucf.edu

Keywords:
Dialog Management, Project Lifelike

ABSTRACT: *Dialog management systems are an essential component of modeling how an individual responds. The structure of how the knowledge is stored can greatly affect how humanlike or computer like the responses sound. This paper compares four different architectures: a context-based architecture, a semantic network, a contextual graph and a combination of the context-based architecture and a contextual graph, which was found to give the best results. A dialog manager was created using each of the four structures from the same knowledge base. Each dialog manager was tested individually to determine which of the four is the best method for modeling a human response.*

1. Introduction

Media and entertainment venues have produced fictional virtual characters that talk and act like a human. The most popular examples include Lucas' C3PO, Star Trek's Commander Data, and Kubrick's HAL. [Breazeal, 2005] All of these virtual characters can carry on a conversation with a person as if they were human, but these systems are fictional creations. Reality is often less impressive than fiction. Systems today can partially recreate the experiences seen on television and on the big screen. Avatars can be created that are lifelike in appearance. These avatars or virtual humans, can assist people in many ways; from an interactive kiosk to intelligent tutoring systems. Virtual humans have the means and opportunity to intelligently interact with users. Their dialog management system controls what a virtual human says during the interaction.

This paper is organized in the following manner. Section two contains information about why this research is important. Section three describes the different dialog managers. The testing procedure and results are found in sections four and five. Section six is our conclusion.

2. Why

Dialog management systems (DMS) are an essential component of modeling a person that communicates verbally in a computer. A DMS selects and composes a response to spoken or textual input that is sent to the DMS. In our system, project Lifelike, the dialog manager controls all of the interaction between the user and an avatar of a computer agent representing a person. The goal of project Lifelike is to push the state of the art in virtual humans to represent an actual, specific human being and maintain knowledge about a specific domain. As the National Science Foundation (NSF) funded this project, both the person and domain represent the NSF Industry & University Cooperative Research Center (I/UCRC) program. In project Lifelike, the system begins with the avatar asking the user, through use of a text-to-speech (TTS) system, what they would like to know about a certain topic. An automatic speech recognition (ASR) system takes the user's speech and generates what it believes to be his/hers words. However, the text generated by the ASR is often not exactly the same as what the user said originally. Depending on the ASR system and the speaker, the word error rate (WER) can vary. The WER is a percentage of errors found by dividing the number of errors by the number of words. In our previous testing, it was found that the ASR system had a WER of 60-70%. [Hung, 2010] This means that our system only correctly identified 3 to 4 words out of 10. These results may seem low but these results were acquired from testing utilizing users with no knowledge of how the system functions and with no training of the ASR system. After the ASR has interpreted the user response, the response is passed on to the dialog manager. The dialog manager interprets the user input speech and searches the knowledge base for an appropriate answer to the user response. The best answer is sent from the dialog manager to the avatar, which essentially speaks the information to the user via a TTS system (Neospeech Paul). This cycle repeats until the dialog manager or the user decides to discontinue the exchange.

As shown from the above interaction, the dialog manager must model the interactions of a human. This includes the ability to answer general as well as specific questions about a domain asked by a user. This paper compares four different architectures: a context-based architecture, a semantic network, a contextual graph and a combination of the context-based architecture and a contextual graph.

3. The Dialog Managers

Each of the four dialog managers were built with different criteria in mind. The CONCUR dialog manager was the first dialog manager built and is known as our original dialog manager. [Hung, 2010] It was designed to model the behavior of a specific person. The CONCUR dialog manager uses context-based reasoning as an architecture. [Gonzalez et al., 2008] The best example to explain how context-based reasoning functions is to examine how one drives a car. When driving a car, one does not use all of their knowledge of driving at the same time. One does not drive as if on a highway when they are in fact driving in a neighborhood. The CONCUR dialog manager functions in the same manner but instead of environmental triggers, conversational cues are used. The CONCUR dialog manager had two weaknesses: the first was in the ability to answer specific questions; the system was only able to provide general information. To address this, a semantic-based network dialog manager was built. The second issue of the original dialog manager was the loading speed. The original dialog manager had a delay between pressing the start key and asking the first question. The contextual graph dialog manager was built to be able to preload all the needed information into the system. The fourth dialog manager built was an attempt to combine the strengths of the original dialog manager and contextual graph dialog manager into one system to address these deficiencies.

3.1 The CONCUR Dialog Manager

The CONtext-centric Corpus-based Utterance Robustness or CONCUR is a context-centric architecture that allowed the system to overcome limitations of automatic speech recognition (ASR). This was done by using a lightly annotated corpus / knowledge base that is contextually organized. This organization allows the system to quickly process the corpus and categorize the knowledge into different contexts. During a typical interaction, the user's input is analyzed and the keywords are extracted. At this point, the system performs two separate searches. The first search is an exact search. Does what the user said match a contextual topic word for word? The exact search is quick and has the potential to find the proper context but has a low success rate, as most users tend not to describe the contextual topic word for word. The second search compensates for the variation of speech patterns of the user. This is done though use of the keywords. During this search, each keyword is compared to the different contextual topics. If a match is confirmed to a contextual topic, then that topic is added to a list of possible topics. After all the contextual topics have been compared, the system analyzes the list of possible topics. If this list only contains one topic, that topic is presented to the user. However, if the list contains multiple possible topics, the list of topics is presented to the user and asks them to disambiguate what was asked. A problem with CONCUR is that even if a question keyword is detected, the system has no method to refine the answer. A detailed explanation of the function of CONCUR can be found in Hung. [Hung, 2010]

3.2 The Semantic Network Dialog Manager

For this system, the text corpus is manually converted into an associative network in the knowledge engineering phase. First, complex sentences were broken down into simpler ones having the form {<noun phrase><verb phrase><noun phrase>}. Each of the nodes in the associative network represents a noun phrase. These nodes are connected via edges which represent the verb phrases. Thus, a full sentence can be represented by a node-link-node triple. This process results in a network structure which supports context in an intuitive way: given some topical node, all of the nodes which are connected to that node will likely also be relevant in the same context.

This structure creates a suitable foundation for the LifeLike dialogue engine. When a user asks a question, their query is superficially parsed to remove trivial words (e.g. articles such as “a” and “the”). Then, the meaningful words in the user's query are compared against the topical nodes in the associative network; the noun phrase with the highest degree of similarity is selected as the current node. When a successive query is made by the user, the system performs another keyword-based comparison of the query versus the neighboring connections to the current node. In this way, the system can take advantage of contextualized knowledge to generate a smaller search space.

3.3 The Contextual Graph Dialog Manager

The contextual graph dialog manager uses a contextual graph to organize and process the knowledge base. [Pomerol et al., 2002] The structure of a contextual graph is similar to a tree structure. In a contextual graph, there are three different types of nodes: action, contextual element, and activity. The action and activity nodes present information to the user and the contextual element allows the system to gain more information. The contextual graph dialog manager uses the contextual element and action nodes but adds a new type of node (goto). The goto node allows the system to move throughout the graph quickly. In the contextual graph dialog manager, the system performs a multiple-level keyword search. This search begins by looking at the keywords at the top node. If none of the keywords at the top node can be found in the request from the user, then the system moves throughout all of the contextual element nodes until a keyword is matched or all of the contextual element nodes have been checked. When a keyword is matched, the system presents the matched information to the user. After the information is presented, the goto node is used to send the system back to the top of the contextual graph. If keywords are not matched, the system asks the user to repeat the response. The contextual graph for the I/UCRC domain knowledge can be seen in figure 3.1. Each of the green squares represents a phase the dialog manager may output. The first blue circle indentifies the context of the user's input. The second layer of blue circles helps identify which phase is needed.

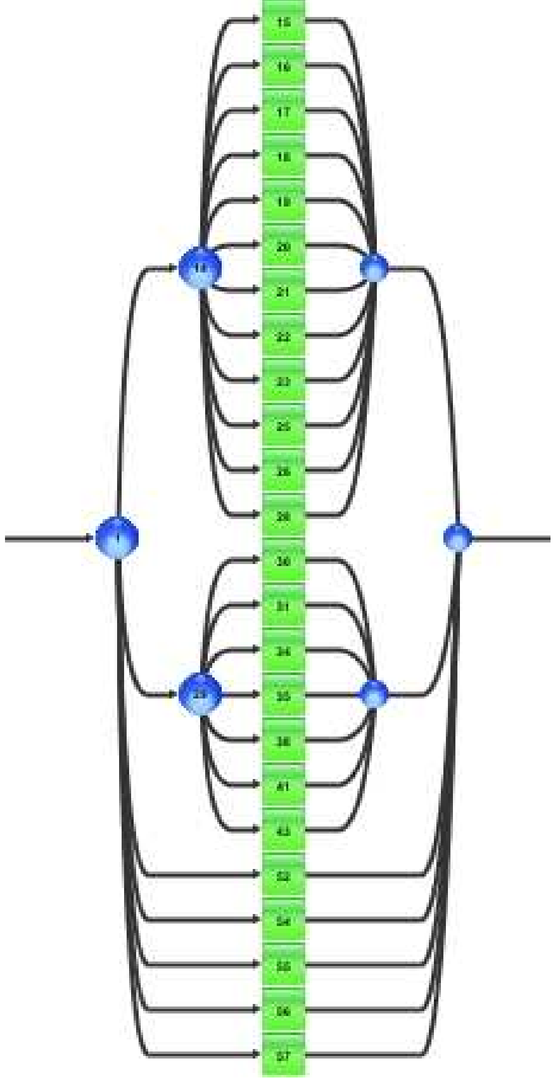


Figure 3.1: Contextual Graph of the I/UCRC Domain Knowledge

3.4 The Hybrid Dialog Manager

The hybrid dialog manager combines the strengths of the original dialog manager and contextual graph dialog manager. The original dialog manager is exceptional at identifying the topic the user wants to talk about. However, the original dialog manager can only give general information about the topic and cannot answer specific questions. The contextual graph dialog manager can answer specific questions when given enough information to process the knowledge in the contextual graph. The hybrid dialog manager uses the original dialog manager method for identifying the topic and then uses a contextual graph to attempt to give a better answer. The hybrid system only requires a little more time creating the corpus when compared to the original dialog manager by making a few simple contextual graphs and adds the filenames to the corpus. In the original dialog manager, if the system could not determine a topic, the system would convey a list of possible topics. For the hybrid system to work properly, the original dialog manager would need to select a topic and use the contextual graphs associated with it to refine the answer to give the user the knowledge/information they requested. To do this task, a best guess function was created. This function takes in a small list of possible topics and tries to make a guess about what the user wanted knowledge/information about. The function looks at the list of possible topics and tries to determine which context occurs most often in the list. A context is a specific topic within the domain knowledge. If a context cannot be determined, the system then looks for a common super-context on the list of possible topics. A super-context is a more general topic within the domain knowledge. If the function finds a common super-context, the first item in list of possible topics with that super-context is selected as the best guess. If no common super-context is found, the function acts just as the original dialog manager would in that situation. For example, let's say the user inquired about the "planning grant." The system would then generate the following list of possible topics.

- Overview of Planning Grant - About the planning grant
- Overview of Planning Grant - Considerations for Writing a Planning Grant Proposal
- Planning Grant Paper - Planning Grant Proposal
- Planning Grant Meeting - About the Planning Grant Meeting

There is no common context among the list of possible topics but there is a common super-context. The super-context "Overview of Planning" would be selected as the best guess and the first topic on the list with that super-context ("Overview of Planning Grant - About the planning grant") is returned as the topic about which the user inquired. At this point the hybrid dialog manager looks to see whether there is a contextual graph associated with that topic and uses the graph to refine the response. Another major change in the hybrid system is the way the contextual graphs are built. Figure 3.2 is an example the new contextual graphs. In the new contextual graphs, there are only three possible context nodes (questions) and they are "QandAResponseDetected", "QWordType", and "KeyWords." These question nodes all refer to a variable within the hybrid dialog manager. This allows the system to automatically process the contextual graphs internally allowing the system refine its own response quickly. In Figure 3.2, the "QandAResponseDetected" node is the first question node (1) in the graph. This node should always be first, if it is required. If it is not required, then the order of the other two nodes does not matter. Node 2 is the "QWordType" question and node 3 is the "KeyWords"

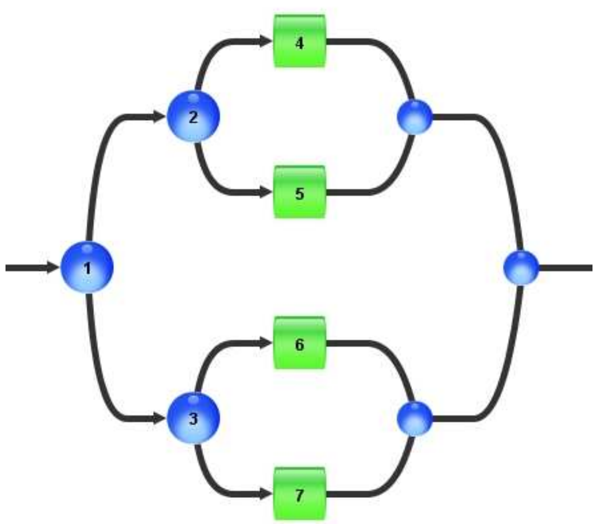


Figure 3.2: Example of Contextual Graph used by the Hybrid Dialog Manager

node. For example, if the user asked "When is the planning grant meeting?" The hybrid dialog manger would find the "About the Planning Grant Meeting" context and load the above, Figure 3.2, contextual graph and begin processing the graph. The first question node processing is the "QandAResponseDetected" node (1). This variable would be positive in the system as the hybrid dialog manger would detect the question word "when". This leads the hybrid dialog manager to the "QWordType" question node (2). This question node looks at the "question word" (QWordType). If the "question word" matches one of the paths on the contextual graph, then the system follows that path. If no match is made, the hybrid dialog manager stops processing the graph and uses the original answer that the system found.

4. Testing

The original dialog manager has been tested in the past in order to evaluate how it performs with the limitations imposed by ASR. [Hung, 2010] With this new set of tests, we sought to see how the semantic network, the contextual graph, and the hybrid dialog managers compare with the original dialog manager. All four of the dialog managers versions were created using the same knowledge base, the domain knowledge of the I/UCRC program. A comparison test was devised to determine which of the four dialog managers performs the best which is based upon previous work. [Hung et al., 2009] The comparison test is comprised of a tester interacting with the four dialog managers asking a set of 15 questions. The 15 questions are evenly divided into three groups: easy, hard, and impossible. The easy questions are simple questions that every dialog manager should be able to answer correctly. For example, one of the easy questions is "What is a planning grant meeting?" The hard questions are questions in which the knowledge is present within the knowledge base but the syntax and the vocabulary makes the question harder for a machine to understand. A human would be able to understand these questions with little problem. An example of a hard question is "What is the monetary amount of the planning grant award?" The impossible questions are questions whose answers may not be within the knowledge base or possibly off topic. The answers for most of these questions are not within the knowledge base and thus the system should respond that it does not know the answer. An example of an impossible question is "Do you have any friends that are avatars?" The dialog manager should respond that it does not know the answer to this question. All of the questions were selected from the actual questions that testers have asked the original dialog manager in previous testing situations. Six of the developers where selected to perform the comparison test. This way most of the testers are familiar with the functionality of each dialog manager allowing the greatest performance to be found for each system as some of the dialog managers required a specific protocol when interacting. The test consists of the testers interacting with the system and filling out an eight question survey about the interaction. This process was repeated until they have interacted with each of the dialog managers. The tester was never told which dialog manager they were interacting with until after all of the tests were complete. A system log was generated during each test and each session was audio recorded in order to determine how the ASR affected each dialog manager. All of the testing was done on the same machine using a desktop microphone in our lab with light to no ambient noise.

5. Results

Each response to the survey questions was answered on a scale from “strongly disagree” to “strongly agree.” A negative response to the survey question was worth -2 points and a positive response was worth 2 points. The points per survey were summed. If every response on the survey was answered positively, then the maximum score would be 16. The ASR results for one specific tester were determined to be a statistical outlier (a difference of over 20% from the average) and those results were removed from the tables. The score per dialog manager and the standard deviation can be seen in table 5.1.

	Average	Standard Deviation
Original	2.1	4.8
Semantic	-7.85	6.5
Contextual Graph	-5.7	4.5
Hybrid	3.4	5.9

Table 5.1: Summary of Survey Results (5 Testers)

The survey results were only part of the data collected during the testing. For every test run, the dialog manager’s response was recorded and analyzed along with information about the automatic speech recognition. If a question was answered correctly 1 point was given. However, if too much information was given, then a quarter of a point or more was deducted from the point given. The number of attempts per question was also recorded. This number shows how many times the user had to repeat each question. The results in table 5.2 are averaged across all of the test runs except for one tester. The ASR performed about the same across the four dialog managers. Thus, this factor can be eliminated as a possible difference between the results as no dialog manager was favored. In this small scale test, there is no statistical difference between the Hybrid and the Original dialog manager. The Hybrid dialog manager was able to answer seven of the fifteen questions correctly and only outright failed on one question. The next best dialog manager was the original which was able to answer only four questions correctly. It is interesting to note that one of the dialog managers (hybrid) was still able to answer over half of the questions from this individual. The results of the recorded data echo the survey results. This confirms that the hybrid dialog manager should outperform the rest in a larger test.

6. Conclusions

It was expected that the hybrid system would perform slightly better than the original dialog manager because the hybrid was built upon the original dialog manager. Due to the design, the hybrid dialog manager can only improve the results of the original dialog manager. The results of this testing are expected to hold true with larger groups as the original dialog manager was found to be similar to the results obtained during the previous testing. [Hung, 2010] The WER of these tests are better than the ones found previously but the hybrid has already shown it could function with a high WER. The structure of the knowledge base plays a large role in determining if the dialog manager will be able to respond as desired even if the knowledge is present. The best results may come from combining two different structures.

	Original		Semantic		Contextual Graph		Hybrid	
	Correct Response	Number of Tries	Correct Response	Number of Tries	Correct Response	Number of Tries	Correct Response	Number of Tries
Question 1	1	1	0	3.5	1	1	1	1
Question 2	1	1.3	0.3	3	0	2.3	1	1
Question 3	0.8	1	0.3	1.8	1	1	1	1
Question 4	0.8	1	0.3	2	0	1.3	1	1
Question 5	1	1	0	2.8	1	1	1	1
Question 6	1	1.3	0	2	0	1.8	0.8	1.5
Question 7	0	1.5	0	1.5	0	2	0.3	1.3
Question 8	0	1.8	0	2.3	0	2	0.3	1.3
Question 9	0.8	1.3	0	1.3	0	1.8	0.3	1.3
Question 10	0.2	1.5	0.3	1.8	0	1.3	0.3	1.5
Question 11	0.5	1.3	0.5	1.8	0.5	1.3	1	1
Question 12	0.2	1.5	0.3	1	0.5	1.8	1	1
Question 13	0	1.5	0	2.3	0	1.8	0	1.3
Question 14	0.5	1.5	0	2.5	1	1	0.3	1.3
Question 15	0.2	1.3	0.5	1.8	0.3	1.3	0.3	1.3
Average	0.5	1.3	0.2	2.1	0.4	1.5	0.6	1.2

Table 5.2: Average Results per Question for each Dialog Manager (5 Testers)

7. References

Breazeal, C. (2005). Socially intelligent robots. *Interactions, 12*(2), 19-22.

Gonzalez, A. J., et al. (2008). Formalizing context-based reasoning: A modeling paradigm for representing tactical human behavior. *International Journal of Intelligent Systems, 23*(7), 822-847.

Hung, V. (2010). *Robust Dialog Management through a Context-Centric Architecture.* University of Central Florida, Orlando, FL.

Hung, V., et al. (2009). *Towards a Method For Evaluating Naturalness in Conversational Dialog Systems*. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics.

Pomerol, J.-C., et al. (2002). Operational knowledge representation for practical decision making. *Journal of Management Information Systems, 18*(4), 101-116.

Smart Bandits in air-to-air combat training Combining different behavioural models in a common architecture

Jan Joris Roessingh, Ph.D.
Robbert-Jan Merk, M.Sc.
Pieter Huibers, M.Sc.
Remco Meiland, M.Sc.
National Aerospace Laboratory (NLR)
Amsterdam, the Netherlands
smartbandits@nlr.nl

Maj. Roel Rijken, M.Sc.
Department of Operational Research Evaluation
Ministry of Defence
The Hague, the Netherlands
rj.rijken@mindef.nl

Keywords:
Architecture, Computer Generated Forces, Opponent modeling, Cognitive modeling, Machine Learning

ABSTRACT: *The Smart Bandits project in the Netherlands aims at developing Computer Generated Forces (CGFs) exhibiting realistic tactical behaviour so as to increase the value of simulation training for fighter pilots. Although the focus lies on demonstrating adversarial behaviour in air-to-air missions, the results are more widely applicable in the simulation domain.*
CGF behaviour is traditionally governed by scripts that prescribe pre-determined actions upon a specific set of events. There are certain shortcomings attached to the use of scripts, for instance, the high complexity of scripts when considering full mission scenarios and the rigid and unrealistic behaviour that scripted CGFs tend to exhibit. To overcome these shortcomings, more sophisticated human behaviour models, combined with state-of-the-art Artificial Intelligence (AI) techniques are required. The Smart Bandits project explores the possibilities of applying these AI techniques.
This paper provides an overview of the entire system development and specifically explains the principal architecture that bridges the gap between theoretical behaviour models and their practical implementation in CGFs for fighter training purposes. The training environment in which the CGF are tested consists of four networked F-16 fighter aircraft simulators. This set-up is capable for providing experimental training to pilots for combat against enemy fighter formations (in the form of intelligent CGFs). The architecture is generic in the sense that it can cater for various human behaviour models, differing conceptually from each other in their use of AI techniques, the internal representation of their cognition, and their learning capabilities. Behaviour models based on cognitive theory (e.g. on theories of situational awareness, theory of mind, intuition and surprise) and behavioural models based on machine learning techniques are actually embedded into this architecture.

1. Introduction

Tactical training of fighter pilots in simulators is already widely used. An essential feature of the training of tactics is the presence of participants, other than the trainees. These participants may be team mates, e.g. other fighters in the formation, supporting forces, e.g. forward air controllers, neutral forces, e.g. civilians, or enemy forces, such as adversary fighters. In simulations, the roles of these participants can be performed by humans, Semi-Automated Forces (SAFs) or CGFs. SAFs have some functionality to perform role-related tasks, such that multiple virtual entities can be controlled by one human. However, the use of human experts to participate in tactical simulations may neither be cost-effective, nor operationally effective. First, these human participants are expensive assets. Second, as the simulation is not meant to provide training to them, they could be used somewhere else. Therefore, it is more effective to perform these roles by CGFs, insofar these CGFs are capable of performing these roles in an adequate manner.
However, the current state-of-the-art of CGFs is in many cases inadequate for tactical training purposes, because of their behavioural simplicity. Apart from aforementioned SAFs, four categories of CGF-

behaviour can be distinguished (Roessingh, Merk & Montijn, 2011):

1. Non-responsive behaviour, in which the CGF behaves according to a pre-determined action sequence, with minimal capability to observe or react to the environment; Such a CGF is, for example, able to follow a route defined by waypoints.
2. Stimulus-Response (S-R) behaviour, in which the CGF, in response to a certain set of stimuli or inputs from the environment, always exhibits a consistent behaviour; Such a CGF is, for example, able to intercept an aircraft when the aircraft position can be observed continuously.
3. Delayed Response (DR) behaviour, in which the CGF not only takes into account a current set of stimuli from the environment, but also stimuli from previous moments, which are stored in the CGF's memory. Such a CGF is, by means of remembering previous positions, able to intercept an aircraft, even though this aircraft is not continuously observable.
4. Motivation-based behaviour, which CGF combines S-R and DR behaviour but additionally takes into account its motivational states. These motivational states are the result of *internal* processes and may represent goals, assumptions, expectations, biological and emotive states. Such a CGF could, for example, make the assumption that a targeted aircraft is running low on fuel and that it will return to base. As a consequence, the CGF may decide to abort the interception. Alternatively, the CGF may anticipate the route change of the aircraft and decide to intercept the aircraft at a more favourable position.

A characteristic of the CGF that so far is not included in the discussion is learning behaviour or adaptive behaviour (in the sense of Russell and Norvig, 2003). CGFs that exhibit behaviour that is either S-R, DR or motivation-based, may be extended with the capability to adapt this behaviour on the basis of Machine Learning (ML). ML is a branch of Artificial Intelligence that focuses on adapting computer behaviour on the basis of examples. In general, ML techniques aim at improving a computer program's performance of a certain task through experience. ML-techniques enable the development of CGFs that are better tailored to the expertise of the trainee. Also, ML-techniques prevent the painstaking development of a set of rules (for example '*if-then* rules') that need to be derived for each specific problem or situation to be resolved, based on the 'manual' elicitation of operational expertise that is largely implicit and not simply explicated in terms of logical rules. Hence, our purpose of using ML techniques is not to recreate human-like learning, in the same way that other behaviours will be recreated using cognitive modelling techniques, but to implant knowledge in the CGF by letting it engage in (a large number of) training trials or episodes.

The goal of this paper is to illustrate the development of intelligent CGFs within the Smart Bandits project (2010-2013). This project seeks to implant humanlike intelligence into the CGFs that appear in simulated mission scenarios. With the project Smart Bandits, the Dutch National Aerospace Laboratory (NLR) strives to take a significant step forward in the area of simulated tactical fighter pilot training using expertise from the Royal Netherlands Air Force (RNLAF). The central message of this paper is that cognitive modelling is a powerful means to create motivation-based behaviour in CGFs. However, to reduce the elicitation of operational knowledge for the development of intelligent CGFs acting in complex domains, we advocate the additional use of ML techniques. It is demonstrated how cognitive models and machine learning techniques can be combined in a common architecture.

2. Modelling Motivation-based Behaviour

2.1 Smart Bandits models

One approach to generate intelligent behaviour is cognitive modelling. In this approach, computational models are designed to simulate human cognition. Within the Smart Bandits project, three cognitive models have been designed so far: a naturalistic decision making model, a surprise generation model and a situation awareness model. All three models have been evaluated using abstracted scenarios from the air combat domain.

<u>Naturalistic Decision Making (NDM)</u>
One particular decision making strategy that was considered for application is so called 'satisficing'. This is the process of creating a satisfying or workable solution rather than trying to create the best solution for the problem on hand (see Rouse, 1981). A related theoretical model for this type of decision making is NDM (in its applicable form referred to as "pattern recognition" or "recognition primed decision making", see Klein, 1989, 1993). The decision making strategy generally relies on long-term memory. Knowledge and past experiences of the pilot result in mental models of the air-to-air problems and the context in which these problems must be resolved. These mental models are stored in long-term memory. Because human decisions

are not purely driven by rational-analytic processes, some naturalistic models take into account the role of emotions and intuitions in the decision making process. Some emotions may result in more efficient decision paths, speeding up the decision process, while some emotions may also negatively affect memory operations, causing rational-analytic processes to be erroneous and more time-consuming.

A behaviourally realistic modification of such model was inspired by Damasio's Somatic Marker Hypothesis (Bechara & Damasio, 2004). The latter hypothesis assumes a central role to experienced emotions as an intuitive part of decision making while integrating this intuitive part with rational reasoning to form a two-stage decision making process. A description of the complete model is given in Hoogendoorn, Merk & Treur (2009).

Surprise generation

Surprise is considered a universally experienced human cognitive reaction to unexpected situations with recognisable impact on behaviour. However, there is little attention to the phenomenon of surprise in CGF research and few CGFs have human-like mechanisms for generating surprise intensity and surprised behaviour. This leads to impoverished and unrealistic behaviour of CGFs in situations where humans would react surprised. For air combat this forms a problem as the element of surprise is considered an important factor in military operations by many military experts.

For this reason, a model for generating surprise intensity and its impact on the behaviour has been developed (Merk, 2010). The model is based on various theories and empirical results from cognitive research on human surprise behaviour. Besides the unexpectedness of a situation, other cognitive factors such as the novelty of the situations are factored in.

Situation awareness

An important factor for effective decision making is Situation Awareness (SA). SA is especially important in work domains where the information flow can be quite high and poor decisions may lead to serious consequences. For this reason we designed a model based on Endsley's (1995) three levels of SA: (1) the perception of cues, (2) the comprehension and integration of information and (3) the projection of information into future events.

The basic SA model (see figure 1) used for intelligent CGFs in Smart Bandits (see Hoogendoorn, van Lambalgen & Treur, 2011) consists of five components: (1) observations, (2/3) belief formation of current situation, (4) belief formation of future situation and (5) the mental model. The beliefs on current situations and future situations are activated (receive an activation value) through a threshold function, a technique adopted from the neurological domain.

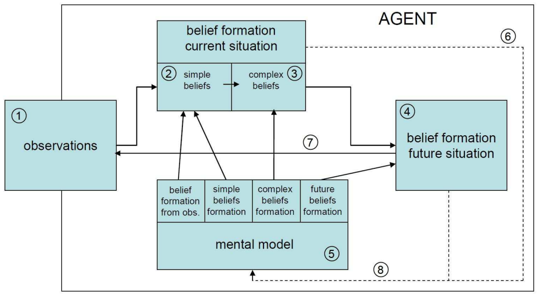


Figure 1: Cognitive model for situation awareness: overview

The SA model in figure 1 represents the knowledge of the domain that is used to form the beliefs. This model is based on assumption mentioned in the NDM section that pilots use dedicated mental models of air-to-air problems which represent the relationships between various observations and the formation of beliefs about the environment, which, in turn, direct the further observations to be performed.

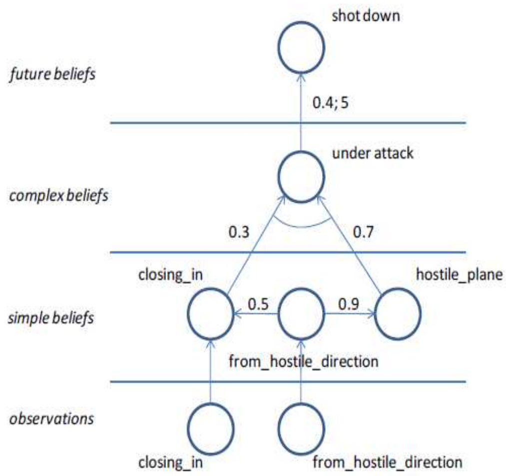


Figure 2: Example model for situation awareness (Hoogendoorn, van Lambalgen & Treur, 2011)

In the following, an overview of the basic algorithm will be provided. Figure 2 shows a simplified (part of a) model that is used to determine a certain set of beliefs regarding the hostility of a target. The model consists of 4 distinct layers: 1) observations, which constitute the directly observable aspects of the environment, 2) simple beliefs which are linked directly to either observations or other simple beliefs, 3) complex beliefs which are linked to different simple beliefs other complex beliefs and 4) future belies. The belief model is represented as a directed graph where each node represents a belief about the world state. Each belief has an activation value. The beliefs have connections between them which represent the cognitive links that different beliefs share. In Figure 2 we can see that this belief network contains a positive connection (strength 0.9 > 0) between the belief that a plane is coming from a hostile direction and the belief that this plane is in fact hostile. The knowledge about the domain is contained in the connections and the associated *strength values*.

The algorithm gives the observations activation values corresponding to the degree of certainty of these observations in the environment. The value is then propagated to the connected nodes by passing it through a threshold function and multiplying its result with the strength of the connection between the nodes. If a node has multiple connections pointing towards it, this effect is cumulative. Since certain beliefs have a positive effect on the complex beliefs while other have a negative effect on them, this algorithm can be used to reason about these complex beliefs in a quantitative manner based on simple observation.

Another important aspect is the degradation of SA that may arise in demanding circumstances. When time is limited, perception and the integration of cues is impaired leading to incomplete knowledge of the environment. In addition, humans will not always be able to make all necessary observations due to limitations in working memory. Depending on the amount of time available, knowledge on the situation can be further refined by considering less active beliefs. These characteristics are reflected in the behaviour of the intelligent CGFs. A detailed description of the above model can be found in Hoogendoorn, van Lambalgen & Treur (2011).

3. Machine Learning

3.1 Reinforcement Learning

A common distinction in machine learning techniques is between supervised and unsupervised learning (e.g. Russel & Norvig, 2003). In supervised learning, after each trial, the agent is presented with the responses that should match the input presentation (also called input example) on which he was supposed to act. The difference between the actual response and the desired response is used to train the agent, just as a trainer or supervisor would make a student aware of the desired response. For example, the agent could learn to fly a manoeuvre by being presented with the correct responses. In unsupervised learning, the agent is merely presented with input examples. The agent has to find hidden structures in the presented examples. Since the examples given to the agent are not accompanied by the responses, there is no difference signal to train the agent. The agent could e.g. learn to distinguish between friendly and enemy tactics.

Reinforcement learning has elements of both aforementioned learning techniques. Rather than being presented with the correct response after each trial, the agent receives feedback from the environment *during* the execution of each trial. Although the feedback may not necessarily represent the correct response for each individual action, the learning technique aims at providing aggregated feedback for the complete trial and therewith reinforcing the correct responses on average. However, this does not guarantee convergence to the correct response. Technical implementation of reinforcement learning is explained in Sutton & Barto (1998).

Reinforcement learning is particularly suited for agent application in simulated environments, because in such environment the agent is able to explore the environment such that a large number of successful and less successful responses can be evaluated. Also, in complex environments, the desired responses, e.g. the best possible opponent engagement tactic, is often unknown. Reinforcement learning provides a technique to improve responses with each trial, therewith discovering better tactics.

A common problem with reinforcement learning is that it requires a large amount of memory to store intermediate calculated values (responses combined with states of the agent in its environment, e.g. its position, speed and heading). In a realistic tactical environment this practically translates to an infinite amount of response-state combinations ('state-action-space'). In the Smart Bandits project, air-to-air engagements were simulated between two friendly aircraft and two enemy aircraft, the latter two represented by learning agents. In these engagements, the learning agents could only respond in four ways (left, right, forward and shoot). In this example, we stored the state-action-space in a table, which after an acceptable number of learning trials took in the order of 2 gigabytes of memory. Such memory-demand scales-up exponentially with additional parameters. The outrageous memory demands can be diminished by approximating the state-action-space, rather than keeping all the exact values. One way of approximating a large state-action space is by using Neural Networks (NNs), as will be explained in the next section.

3.2 Reinforcement Learning & Neural Networks

In a general sense, a Neural Network (Haykin, 1998) can be considered as a network that can model any mathematical function. In this case, we use a Neural Network (NN) to approximate the aforementioned state-action-space. The input for the NN is the current state of the agent in its environment. The output of the NN is a value for each possible action of the agent. The output of the NN is optimised on the basis of the data that is generated by the Reinforcement Learning (RL) algorithm.

For the implementation of the RL method we use an off-policy Temporal Difference (TD) control algorithm known as Q-learning, see Watkins (1989). We used its simplest form, *one-step Q-learning*, which is defined by

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]$$

Where Q denotes the action-value function for action a_t in state s_t; γ is the so-called discount factor $(0 \leq \gamma \leq 1)$, α is a constant step-size parameter and r_{t+1} is the reward the agent receives from the environment when performing action a_t (see Sutton & Barto, 1998).

To approximate the Q values in a state, we use a neural network structure. This neural network is of a simple feed forward form with one hidden layer. The current state, represented by a vector $\bar{s}_t$, is presented to the input nodes of the network. The output nodes of the network represent the approximate Q-values for this state. Hence, the number of output nodes of the network equals the number of possible actions in this state. In each state a choice can be made from a constant set of actions.

In our simulations we use an episode of finite length and update the weights of the neural network in the following way.

A step of the episode is started in a certain state $\bar{s}_t$ for time step t. In this state the neural network is characterized by a weight vector $\bar{w}_t$. We have chosen to execute action a_t and receive a reward r_{t+1} from the environment. After executing action a_t, we transition to state $\bar{s}_{t+1}$ where we find the approximated Q-values for each current state-action pair by calculating the neural network in this new state. The maximum Q-value is chosen to calculate the TD between the two states.

$$TD = r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)$$

The TD value, multiplied by the discount factor γ, is used to adjust the weight vector $\bar{w}_t$ of the neural network by using a standard back- propagation algorithm (see Haykin, 1998). In this way the Q-value $Q(s_t,a_t)$ is updated and a new step is initiated for state $\bar{s}_{t+1}$.

Using this method, the data of the RL algorithm does no longer need to be stored. In fact, the NN is trained using the data that becomes available from the RL algorithm. Where previously we needed 2 gigabytes of memory for resolving a relatively simple air-to-air problem, we now only require approximately 10 kilobytes of data to store the NN knowledge for this problem. This knowledge is represented by the weight-values of the NN. Also, the memory demand does no longer scale up exponentially with complexity of the problem, but only linearly. For this purpose relatively simple NNs of the feed-forward type can be used rather than recurrent NNs. However, we identified two reasons to develop alternative ML techniques for the type of agents that are needed to act in complex tactical scenarios.

Unlike domains, such as resolving problems in games like chess, where the optimal next action is completely determined by the current state of the world, the resolution of tactical problems is characterised by the need to use previous world states. For example, an air-to-air opponent may disappear for some time and may pop-up at a different position, which must be taken into account by the agent. In other words, tactical problems are characterised by imperfect or incomplete knowledge of the environment[1]. RL-techniques are known for not being overly robust for these types of problems and we have indeed experienced divergence from the correct response of our agents when confronted with more complex problems.

Some realistic tactical problems require memory of the previous states to be taken into account in current decisions. Because of this, RL-based agents are not well suited for realistic tactical problems. For applications in which Delayed Response behaviour or motivation based behaviour is required, RL may not be the preferred technique.

For more advanced problems in the air-to-air domain, *evolutionary techniques* are investigated as an alternative to RL in the next section.

3.3 Evolutionary Techniques & Neural Networks

Artificial autonomous systems are expected to survive and operate in dynamic, complex environments. The specific abilities of an agent, necessary to perform in such an environment, are hard to predict a priori, let alone to specify in detail. Artificial evolution of autonomous systems enables agents to optimise their behaviour in complex, dynamic environments, without the use of detailed prior knowledge of domain experts. Where RL-techniques assume solutions to the problem to possess the Markov Property (see footnote, earlier), evolutionary techniques (Bäck, Fogel & Michalewicz, 1997) are not bound by this constraint and are applicable to a larger set of problems.

Evolutionary techniques use an iterative process to search the fitness landscape in a population of solutions, in this case the solutions to a tactical

[1] In more formal terms, the solutions to these problems do not possess the so called Markov Property: the next state *s*' depends on the current state *s* and the decision maker's action *a*. But given *s* and *a*, it is conditionally independent of all previous states and actions.

problem. More successful instances in the populations are selected in a guided[2] random search using parallel processing to achieve the desired solution. Such processes are often inspired by biological mechanisms of evolution, such as mutation and cross-over. Many experiments in evolutionary techniques use NNs to control the agent. Neural networks offer a smooth search space, are robust to noise, provide generalisation and allow scalability (see Nolfi & Floreano, 2000). Furthermore, network architectures can be evolved or optimised to allow Delayed Responsive behaviour. These characteristics, combined with an evolutionary method to optimise the network, provide an interesting research area for complex, dynamic domains. As an example one could update the weights of the connection strengths of the SA model (see section 2.3) using an evolutionary technique in Smart Bandits.

Since cognitive models like the SA model usually have a large set of interrelated parameters, the determination of their (initial) value, using Subject Matter Experts, is cumbersome, speculative and labour intensive. This creates the need to use evolutionary learning techniques for the appropriate weights for the connections between the aforementioned observations, simple beliefs, complex beliefs and future beliefs.

In order to learn the connection weights of the network in Figure 2, two different approaches have been utilised (Gini, Hoogendoorn & van Lambalgen, 2011), namely a genetic algorithm application and a dedicated approach based upon the importance of the weights. The latter approach is called the 'Sensitivity Based' approach. Both approaches utilised a fitness function, which expressed how well a solution complied with the desired state. In this case, the fitness could be measured by the difference between actual activation levels and the activation levels estimated by a subject matter expert. The genetic algorithm performed significantly better than the sensitivity-based approach.

4. Architecture

4.1 Background

The architecture that was envisioned for the project would enable use of several AI techniques implemented in a programming language of choice. However, it was found that (commercially) available CGF packages that were candidates for use in the Smart Bandits project, offered limited prospect to directly connect AI to CGFs. Therefore, the need for an architecture based on a generic communication interface between AI and such CGF package was explicated. The communication interface enables the two-way communication with external 'client-defined' AI-based agents and an internal (to the CGF package) 'dumb' CGF, the latter primarily consisting of the models representing the aircraft and its weapons. Further, it was envisioned that the architecture should enable the combined intelligent CGFs to behave autonomously, abandoning the need for central intelligence. However, it was also realised that no single AI technique would be powerful enough to resolve all problems encountered in air-to-air combat. Each phase of combat (targeting, intercepting, engaging in an attack, self defence, etc.) may require its dedicated AI technique. It was therefore decided that each intelligent CGF could autonomously invoke one or more AI-based agent, depending on the problem it was facing or the phase of the air-to-air mission it was entering. This leads to a separation of responsibilities, in which each AI technique, i.e. each intelligent agent, has its specific responsibilities. The architecture is generic in the sense that it can cater for various and hybrid behaviour models.

4.2 Simulation Environment

The simulation environment that was used for CGFs in the Smart Bandits project is STAGE ™, a scenario generation and CGF software suite. As a basic scenario tool, STAGE provides us with a level of fidelity and abstraction which is well suited for the tactical air-to-air combat simulations that are currently considered. When a higher level of fidelity in platform, sensor or weapon models is required, the basic functionality provided by STAGE is extended. This ability to extend the basic functionality of the CGF environment is one of the reasons that STAGE was chosen as the main CGF software suite in Smart Bandits.

4.3. Middleware (Mediator)

Traditionally, Stimulus-Response (S-R) behaviour (see chapter 1) in agents can be realised in CGF software through the use of scripting and/or basic conditional statements. Combining these simple building blocks often provides a level of credibility to CGF behaviour, which may be adequate for many simulation training exercises. However, for more advanced problems and the associated agent behaviour, including learning behaviour, as described in sections 2 and 3, this method will not suffice. As argued in the previous sections, a wide array of techniques exists for developing CGF behaviour and controlling CGF in a simulation environment. A standard CGF platform does not cater for implementing these different techniques.

In order to use STAGE as the CGF platform in Smart Bandits while delegating the control of the CGFs to external software (i.e. specific software, built using a programming language of choice), an interface ('mediator') was developed through which external agents can receive observations from any CGF in STAGE and can command the CGF to perform actions in the simulation environment.

[2] In the sense of evaluation of a solution by a fitness function.

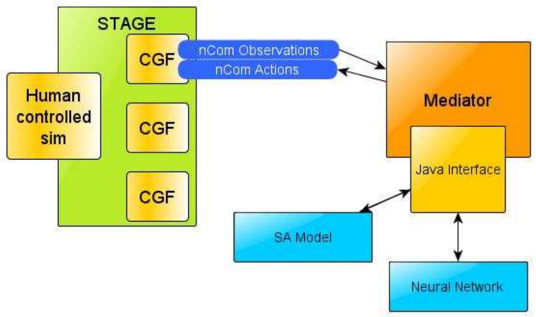


Figure 3: Architecture for including intelligent agents in a COTS CGF package (STAGE™). AI models, for example an SA model and a NN model, can use the C++ or Java interface to communicate with CGFs in STAGE via the Mediator

This middleware layer (the so-called Mediator in Figure 3) communicates in real-time with STAGE through a specific protocol (nCom, Presagis proprietary) and can send and receive the aforementioned observations and actions to and from different agents (possibly distributed over different computers). In order to communicate with the Mediator, external software uses a specific interface, defined in a library, which can easily be linked to the software, e.g. in Java or C++.

This approach creates a clean distinction between the simulated physical entities and the processes which guide their behaviour (agents). This loose coupling makes it easier to develop AI models, since we do not have to pay attention to the actual simulation environment nor do we have to create our own environment. And since the only connection with the actual platforms is through use of the observations\actions interface, it is also reasonably simple to couple multiple AI models to a single CGF entity.

This gives us a solid foundation for the hybrid approach, since multiple AI models can be recruited to guide the different aspects of the CGF's behaviour. Figure 3 gives an example of such a hybrid model in which two models are coupled to a single CGF entity. The SA model (as described in section 2) is used to generate SA based on the CGF's observations and it controls the higher level decision making (e.g. hold, flee, engage), while the neural network (in the sense of the AI described in section 3.2) is used to manoeuvre the CGF in an engagement with hostiles.

5. Conclusions & Discussions

Techniques for cognitive modelling and Machine Learning have been presented in this paper. Unfortunately, there does not seem to be one single technique to resolve all emergent tactical problems of intelligent CGFs engaged in an air-to-air mission.

Cognitive modelling is a powerful means to create motivation-based behaviour in CGFs. However to mitigate drawbacks of cognitive modelling, we advocate the additional use of Machine Learning techniques. Machine learning techniques are essential to reduce knowledge elicitation efforts for the development of CGFs acting in complex domains. This paper recommends combining different approaches into hybrid models.

The goal of the principal architecture presented here is three-fold:

– Different AI models can be recruited by CGFs in the tactical fighter simulation,
– The AI models are distributed at different clients and functionally separate from the simulation,
– The AI models can be activated when needed for specific mission tasks, simultaneously or subsequently.

Together, these three characteristics enable intelligent CGFs based on a hybrid modelling approach.

Within the Smart Bandits project, behaviour and design of intelligent CGFs should be tailored to the training objectives of the tactical training on hand. This paper has not dealt explicitly with training requirements. Implicitly, this paper assumes that required CGF behaviour for tactical training of operational fighter pilots comprises such aspects as the ability to surprise the human opponent, varied, seemingly random, in its responses which are nevertheless valid from a weapon platform perspective. The intelligent CGFs that have been created so far will be validated against training requirements in the coming project phase (2012/2013).

Hence, the two main items for future work within the Smart Bandits project are

– the implementation of hybrid models, in which cognitive modelling and ML are combined and
– tailoring the behaviour of intelligent CGFs to specific learning objectives or competencies.

6. Acknowledgements

This research has been performed under the Smart Bandits contract, awarded by the Dutch MoD to NLR.

An initial overview of this research was presented at NATO meeting RTO-NMSG-107 on December 2, 2011, with contributions of Carolynne Montijn and Tsu Kong Lue.

7. References

Bäck, T, Fogel, D.B. & Michalewicz, Z. (1997), *Handbook of Evolutionary Computation*, Oxford University Press, USA, Edition April 17, 1997.

Bechara, A. & Damasio, A. (2004). The Somatic Marker Hypothesis a neural theory of economic decision. *Games and Economic Behavior*, Vol. 52, pp. 336-372.

Endsley, M.R., (1995). *Toward a theory of Situation Awareness in dynamic systems*. Human Factors 37(1), 32-64.

Gini, M.L., Hoogendoorn, M., Lambalgen, R.M. van, (2011) Learning Belief Connections in a Model for Situation Awareness, In: Kinny, D., Hsu, D. (eds.), *Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent System*s, PRIMA'11. Lecture Notes in Artificial Intelligence, Springer Verlag, 2011, to appear.

Haykin, S. (1998), *Neural Networks: A Comprehensive Foundation*, Prentice Hall/ IEEE.

Hoogendoorn, M. & Lambalgen, R. van & Treur, J., (2011) An Integrated Agent Model Addressing Situation Awareness and Functional State in Decision Making. In: Kinny, D., Hsu, D. (eds.), *Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent Systems*, PRIMA'11. Lecture Notes in Artificial Intelligence, vol. 7047, pp. 385–397. Springer-Verlag, Berlin Heidelberg, 2011.

Hoogendoorn, M., Merk, R.J., & Treur, J., (2009), A Decision Making Model Based on Damasio's Somatic Marker Hypothesis. Presented at *the 9th International Conference on Cognitive Modeling*, ICCM'09, 2009.

Klein G.A. (1989). Recognition primed decisions. In W. Rouse (Ed.), *Advances in Man-Machine Systems Research*, Vol. 5, 47-92, Greenwich, Conn. (USA).

Klein G.A. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G.A. Klein e.a. (Ed.), *Decision making in action: Models and methods*, 138-147. Norwood, N.J. (USA).

Merk, R.J. (2010) A computational model on surprise and its effects on agent behaviour in simulated environments. In: Demazeau, Y., et al. (eds.), *Proceedings of the 8th International Conference on Practical Applications of Agents and Multi-Agent Systems*, PAAMS'10. Advances in Intelligent and Soft Computing Series, Springer Verlag, 2010.

Nolfi S. & Floreano D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books.

Roessingh, J.J., Merk, R.J. & Montijn, C. (2011), Modeling the Fighter Pilot's Situation Awareness and Decision Making Ability in Simulated Air-to-Air Engagements. *Proceedings of the 16th International Symposium of Aviation Psychology*, Dayton, Ohio, 2-5 May, 2011.

Rouse, W.B. (1981). Models of Human Problem Solving: Detection, diagnosis and compensation for system failures. In *Proceedings IFAC-conference on Analysis, Design and Evaluation of Man-machine systems*, Baden-Baden, Germany

Russell, S.J., Norvig, P. (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.

Stone, P. & Veloso, M. (2000), Multiagent Systems: A Survey from a Machine Learning Perspective, *Autonomous Robots, Volume 8, Number 3,345-383*, 2000.

Sutton, R. S. & Barto, A.G. (1998), *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

Watkins, CJCH (1989), *Learning from Delayed Rewards. PhD Thesis,* University of Cambridge,Cambridge, UK.

Author Biographies

JAN JORIS ROESSINGH is a scientist at the department of training, simulation and operator performance of the National Aerospace Laboratory.

ROBBERT-JAN MERK is a Ph.D. candidate at the department of training, simulation and operator performance of the National Aerospace Laboratory.

PIETER HUIBERS is a research engineer at the department of training, simulation and operator performance of the National Aerospace Laboratory.

REMCO MEILAND is a research engineer at the department of training, simulation and operator performance of the National Aerospace Laboratory.

ROEL RIJKEN is an officer at the Royal Netherlands Air Force and a Ph.D. candidate at the University of Amsterdam.

Model Decomposition for Reprogrammable Adversaries

Sean Guarino
Eli Stickgold
Richard Ho
Jonathan Pfautz
Sam Mahoney
Charles River Analytics Inc.
625 Mt. Auburn St.
Cambridge MA 02138
617.491.3474
sguarino@cra.com, estickgold@cra.com, rho@cra.com, jpfautz@cra.com, smahoney@cra.com

Keywords:
Human Behavior Models, Adversary Models, Agents, Adaptive Behavior, Model Reprogramming, Modularity, Moderation

ABSTRACT: *While many military training simulators rely upon adversary behavior models, or agents, to provide effective training content, it is often difficult to update these models to address changing adversary capabilities and behaviors. These changes are often critical to capture as new capabilities and behaviors become more dominant (particularly in domains such as electronic or cyber warfare). Model developers—specifically domain experts who may have limited modeling experience—need intuitive tools and approaches to support the rapid construction and modification of behavior models that address evolving adversary systems and tactics. In this exploratory effort, we extended our agent-based modeling framework with a modular modeling approach that decomposes adversary models into small model elements that can be easily replaced or modified as tactics, techniques, and procedures (TTPs) change.*

1. Introduction

In many military training domains, adversary tactics and behaviors continuously evolve as the adversaries develop new capabilities and/or learn about their own weaknesses. Real adversaries employ agile and rapidly changing tactics, techniques, and procedures (TTPs) to disrupt the TTPs used by US forces, and often attempt to disguise technological capabilities. Most current adversary models and simulations do not adequately support the changes required to reflect new adversary behaviors. Indeed, in modern systems, training for the technology and tactics of yesterday can be just as bad as training for the technology and tactics of previous wars. Effective adversary models for training must be designed both to capture dynamic adversary behaviors and to support rapid adaptations addressing unforeseen behaviors and capabilities.

Current processes for modeling adversaries within existing simulations and training systems are fraught with issues. It may take weeks, or even months of time, depending on the theater of operations, to: (1) observe and characterize new threats against US forces; (2) update models of adversary capabilities to effectively portray those new threats; (3) validate the new models; and (4) integrate new models into simulations or operational systems to support training and analysis. Therefore, new methods are needed to speed and enrich the process of acquiring and updating adversary models, validating those models for analysis and training purposes, and inserting those models into existing systems.

In previous work, we constructed a graphical agent development environment, AgentWorks™ (Furtak, 2009; Rosenberg et al., 2011), designed to support intuitive development of behavior models using our cognitive architecture, the Situation Assessment Model for Person-in-the-Loop Evaluation (SAMPLE) (Harper et al., 2000). SAMPLE is a hybrid computational modeling approach that uses appropriate technologies for different cognitive functions, including fuzzy logic engines for information gathering (Bellman & Zadeh, 1970; Zadeh, 1973), Bayesian networks (BNs) for situation assessment (Pearl & Russell, 2000), and expert systems for decision making (Rasmussen & Goodstein, 1987). Combined, AgentWorks and SAMPLE provides a parallel cognitive modeling approach to that found in other models, such as ACT-R (Anderson et al., 2004) and Soar (Laird, 1987).

In this exploratory effort, we built upon our hybrid computational modeling approach by incorporating new capabilities to streamline the model adaptation process, making it easier for domain experts to adapt existing behavior models to address new adversary capabilities, tactics, and behaviors. In particular, we support human and machine adaptation of behavior models by decomposing models into easily understood parts that a human can readily adapt to address new adversary behaviors, while maintaining the aspects of those behaviors that are unchanged.

2. Background

In this section, we provide a short summary of AgentWorks, followed by a review of other previously developed modular modeling approaches.

2.2 AgentWorks

AgentWorks (Furtak, 2009) provides a hybrid computational modeling framework for constructing complex behavior models, which can then be executed as underlying logic for agents. It incorporates a range of Artificial Intelligence components, as well as more generic computational systems that can be combined as desired into analyses or behavior models. When these components are not sufficient to perform required tasks, AgentWorks can be extended with new customized components or third party components (e.g., a MATLAB component that enables the integration of more complex physics or mathematics models).

AgentWorks is currently designed to use a combination of *Components* and *Actions*, where Components provide underlying computational capabilities (e.g., BNs to analyze a particular situation), and Actions perform operations using those components (e.g., posting data as evidence to a BN, computing, and extracting results). This approach in and of itself is a modular modeling approach, allowing the creation of complex behavior models that merge a range of physics models, analyses, and other behavior model elements within a single Action. However, the current approach does not support a further breakdown of the specific computational models used by an agent (e.g., the BNs, fuzzy logic systems, or expert systems).

AgentWorks also provides intuitive tools to construct and configure the computational underpinnings of models, as shown in Figure 1.

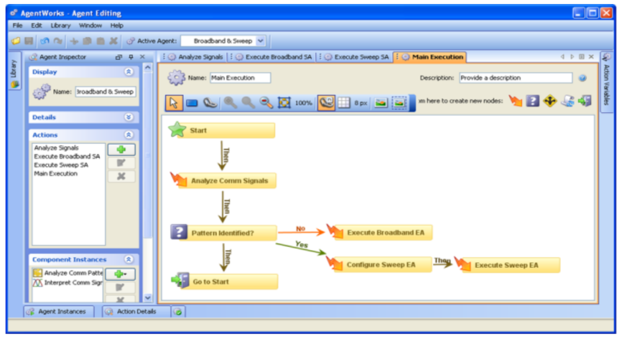


Figure 1: AgentWorks graphical agent development tool

2.3 Previous Modular Modeling Approaches

Modularity has been investigated in previous research. It is widely recognized that having the ability to change pieces of an agent is desirable, both to decrease the time needed to construct new agents, and to allow easier collaboration between groups of agent developers (Jones, Keating, & Porter, 2001; Leavesley, 1998). While several modular modeling solutions have been proposed and implemented, there are several problems that have not been addressed.

Many previous solutions are domain-specific modeling systems (Leavesley, 1998), which provide a modular approach within a specific research area. These systems contain a fixed set of components which can be combined to produce models, as well as an interface specification that allows model fragments to communicate effectively. Although these domain-specific approaches support easier collaboration, they do not provide a more general toolkit for *constructing* and *adapting* new elements within new domains. This limits their ability to effectively incorporate novel adversary behaviors and technical capabilities. Our approach focuses on providing not just a general modular modeling capability, but the tools needed to support the incorporation of those models elements into a larger behavior model.

Other previous solutions focus on a particular modeling paradigm, e.g. a specific implementation using modular state machines (Novak, 2008). While such solutions may include tools to support modular model development, they do not support a wide range of different model types. Because different modeling approaches are effective for different aspects of behavior and different technology capabilities, it is essential to cover a range of different modeling paradigms. Our approach does not limit itself to modularizing a single element, but rather seeks to modularize a range of different computational paradigms.

3. Modularizing Computational Models

To enable rapid adversary reprogramming in AgentWorks, we extended the modular modeling approach to the underlying components and actions within our agents. Our approach advocates the development of components as a series of simple model elements that can be combined into full-scope components. Specifically, in this effort we explored initial approaches to do this for BNs and fuzzy logic systems; a full-scope approach would extend this modularization to all AgentWorks components. Rather than developing a single model that addresses full behaviors for the adversary, users can develop a range of very simple model elements, and then use the new AgentWorks Element Library to import and merge those model elements as desired. When limited aspects of the model elements change, an expert can update and replace only the changed aspect, reusing unchanged model elements as desired. This approach has the added advantage that each model element is itself more simplistic, making it easier for less experienced modelers to understand and update previously developed models.

In our exploratory research, we investigated initial problems associated with modularizing Bayesian networks and fuzzy logic systems. In future work, we plan on addressing additional use cases for the modularization of these components, as well as use cases focused on other types of components, actions, and component merging.

3.1 Modularizing Bayesian Networks

Bayesian Networks (BNs) are probabilistic network models that provide a causality structure and conditional probabilities describing the relationships between nodes in that structure (Pearl et al., 2000). Within AgentWorks, we often use BNs to model situation assessment and fusion tasks, combining various input data into cues for decision-making. In this work, we implemented initial approaches for modularizing and merging BNs, and identified more complex problems that we hope to address in follow-on work.

When considering complex BNs, one can consider extracting large segments of the BN into separate models, creating a suite of simple model elements that can later be merged into the original BN. Consider a case in which one is designing an air combat adversary that analyzes a situation to determine if it is necessary to evade a target blue aircraft (e.g., the trainee). Using the previous AgentWorks approach, one might design a reasonably complex BN to perform this type of analysis, as shown in Figure 2. In this model, the adversary determines whether to evade based on a combination of the threat on his own aircraft, the threat on his mission, and his current morale level. Some of these elements incorporate additional complexity. Incorporating all of this information into a single model reduces the flexibility of the overall model. This becomes more evident as large BNs are created to address more complex analysis and decision-making requirements.

Our modular approach simplifies models like this by extracting each of the complex pieces into separate models, which are later automatically recombined at runtime. In Figure 2, each of the major components contributing to the Evade decision can be separated into model elements, leaving a much more simplistic BN at the lowest level (see Figure 3). Not only does this make each of the model elements easier to comprehend, but—just as in modular software development—it makes the replacement of model elements more simplistic. For example, one could easily use a different Own Ship Threat model that incorporates additional information available to a more sophisticated adversary (see Figure 4).

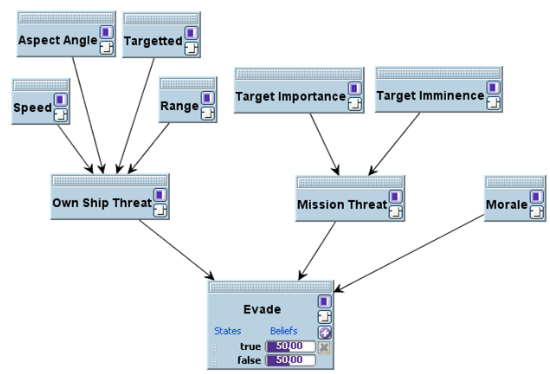


Figure 2: Full Evasion SA Model

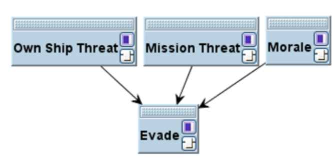


Figure 3: Collapsed Evasion SA Model

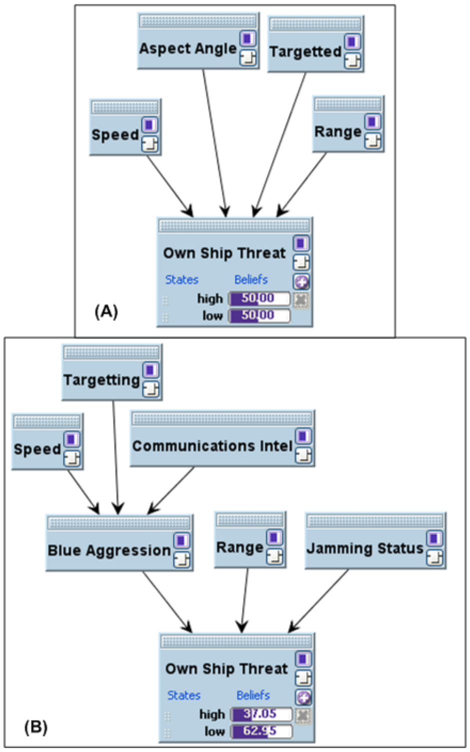


Figure 4: Alternative Own Ship Threat Model Elements for Evasion SA Model

More complex problems that we plan to address in future work will focus on merging BNs at different points (e.g., not at end points of a network). For example, one operation we will want to perform is to insert a new structure that we are newly linking to an outcome. Doing this would require redefining the related conditional probability table in a manner that may be previously undefined. Similarly, inserting new structures at middle layers of BNs may require expanding the CPTs in unforeseen way. In our continued effort, we plan to explore approaches to automate this process.

3.2 Modularizing Fuzzy Logic Systems

Fuzzy logic systems provide a means to fuzzify information sources, and probabilistically reason about those sources. In AgentWorks, we use fuzzy logic for information gathering because the fuzzification process resembles the way humans think about data. For example, a fighter pilot rarely thinks of a target as a specific distance away. Rather, he thinks of the target as too close, out of range, or in missile range (or at the verge between any of those categories) (Zadeh, 1965; Zadeh, 1973). AgentWorks fuzzy logic components uses fuzzification models to interpret raw data points in these categories, fuzzy rules to act on those cues, and defuzzification models to translate the results back into continuous data. Our initial focus in modularizing our fuzzy logic component has been on extracting these individual pieces.

Extracting fuzzification (and defuzzification) models is simple, as shown in Figure 5. These functions direct a fuzzy logic component on how to translate a single value, such as a range, into a fuzzy set that more describes the category or categories that range would be classified as. Extracting these models allows us to define a range of common fuzzification applications, which can then be reused when building new adversary behavior models. Furthermore, it allows us to separately address changes to fuzzification models, for example incorporating new ranges for missile functions. Defuzzification models have an identical structure, but are used in reverse to interpret categorical information in terms of a concrete value. However, because defuzzification models are not deterministic, we often instead use fuzzy values directly within the next reasoning component (i.e., as evidence to post within a BN).

Extracting fuzzy rules is more complex, as they are less independent than the fuzzification models. First, individual rules are not independent from one another. Rather, when exporting a set of rules, one should export all rules associated with particular data types (e.g. a rule set, rather than individual rules). Second, the inputs and outputs of fuzzy logic rules are defined by the fuzzification and defuzzification models. While rules can be saved without those models, doing so requires restructuring the fuzzy logic component, and identifying the template of fuzzification functions that can address the rule requirements. Our initial work has focused on exporting rule sets with defined templates for the required fuzzification and defuzzification models that enable the use of those rules.

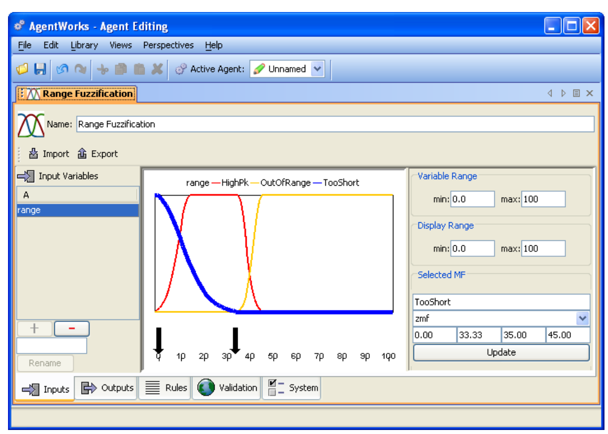


Figure 5: Extracted Fuzzification Function

More complex problems that we plan to address in future work will focus on further decomposition of the fuzzification models (e.g., extracting specific multi-function elements) and of the rules themselves. In addition, we plan to explore combining fuzzy logic model elements—specifically the fuzzification models—with BNs to translate raw data into data evidence for the BN.

4. Conclusions

In this exploratory effort, we explored methods to expand the modularity of our AgentWorks modeling framework, to provide tools to enhance modeler productivity. Using a modular modeling approach, we can decompose behaviors into smaller model elements that can be easily replaced or modified as adversary capabilities, behaviors, and TTPs change. We designed a library system for managing model elements in AgentWorks to manage model deconstruction and reconstruction. This capability has enabled our AgentWorks users to more rapidly update and reprogram behavior models. We tested the system informally by having two inexperienced users of AgentWorks instantiate and combine new model elements based on an Electronic Warfare (EW) scenario in which adversaries presented unexpected behaviors and capabilities. Our anecdotal results suggest that these users had little problem understanding and editing these simple model elements, whereas they had a great deal more difficulty constructing larger models from scratch. When this capability is only partially implemented, we expect a full version to enable AgentWorks to better address the development of adversary models in highly dynamic domains, such as electronic and cyber warfare.

In implementing these modifications in AgentWorks, we learned that it is significantly easier in that framework to adapt model components than model actions. In AgentWorks, model actions are feed-forward processes that can incorporate a number of interdependent steps. To modularize actions, we will need to redesign the actions to better encapsulate disparate pieces of logic. We plan to explore this in greater detail in follow-on research.

We also plan further work in extending the modular capabilities that we implemented for existing AgentWorks components. Specifically, we plan to extend our BN modularization to support more complex model insertion at various points within existing BNs, and we plan to extend plan to extend our fuzzy logic modularization to support the extraction of the sub-elements of fuzzification models. In expanding these components, we plan to address a number of more complex modeling issues as well, such as combining models across levels of analysis or models with overlapping subtasks. In addition, we plan to explore modularization of other components in AgentWorks, such as our social network modeling and analysis component. Finally, we plan to explore opportunities to combine elements of different model types (e.g., using fuzzification models to support BN inputs).

Finally, we recommend further work in implementing intuitive interfaces to access the model library. In particular, while we currently support simple, annotation-based search, it would be more effective to be able to search based on the details and even structure of the underlying model.

6. References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review*, 111(4), 1036-1060.

Bellman, R. E. & Zadeh, L. A. (1970). Decision Making in a Fuzzy Environment. *Management Science*, 17(4), B141-B164.

Furtak, M. (2009). Introducing AgentWorks. In *Proceedings of 14th Intelligent Agents Sub-IPT*. Ft. Monmouth, NJ.

Harper, K. A., Mulgund, S. S., Zacharias, G. L., & Menke, T. (2000). SAMPLE: Situation Awareness Model for Pilot-in-the-Loop Evaluation. In *Proceedings of 9th Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Jones, J., Keating, B., & Porter, C. (2001). Approaches to Modular Model Development. *Agricultural Systems*, 70(2-3), 421-443.

Laird, J. E. (1987). Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1), 1-64.

Leavesley, G. (1998). The Modular Modeling System (MMS): A Modeling Framework for Multidisciplinary Research and Operational Applications. ftp://brrftp.cr.usgs.gov/pub/mms [On-line].

Novak, P. (2008). Behavioral State Machines: Programming Modular Agents. In *Proceedings of AAAI Spring Symposium: Architectures for Intelligent Theory-Based Agents*.

Pearl, J. & Russell, S. (2000). *Bayesian Networks.*

Rasmussen, J. & Goodstein, L. P. (1987). Decision Support in Supervisory Control of High-Risk Industrial Systems. *Automatica*, 23663-671.

Rosenberg, B., Furtak, M., Guarino, S., Harper, K., Metzger, M., Neal Reilly, S. et al. (2011). Easing Behavior Authoring of Intelligent Entities for Training. In *Proceedings of Conference on Behavior Representation in Modeling and Simulation (BRIMS)*. Sundance, UT.

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control,* 8338-353.

Zadeh, L. A. (1973). Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. In *Proceedings of IEEE Transactions on Systems, Man and Cybernetics*. SMC-3(1): 28-44.

Acknowledgement

This work was performed for AFRL/RHAS. The authors would like to thank Capt. David Simmons, Mr. Glenn Cicero, and Mr. Craig Eidman for their engagement, support, and guidance in this effort. In addition, the authors would like to thank Lt. Col. Tim West (USAF, Ret.) and Maj. Kristin Steinke (USAF, Ret.) for organizing and supporting our associated knowledge elicitation in the EW domain, and our analysis of EW models developed in this effort.

Author Biographies

SEAN GUARINO is a Senior Scientist at Charles River Analytics Inc., and was the Principal Investigator for this effort. His expertise includes design and development of dynamic behavior models, model development frameworks, and distributed systems of interactive agents. He has led the development of a wide range of efforts focused on dynamic behavior models, including: air-to-air combat adversaries, counterinsurgency adversaries, situation assessment and bias analysis models to support US pilots, and virtual parent behavior models to guide children's entertainment.

ELI STICKGOLD is a Scientist at Charles River Analytics Inc., and was the Technical Lead for this effort His expertise includes machine learning techniques, cognitive models, and probabilistic modeling. He has experience developing intuitive model construction tools, and has worked on developing a number of extensions to our graphical agent development environment, AgentWorks.

RICHARD HO is a Scientist at Charles River Analytics Inc. He supported the design of the user interfaces for the Model Element Library and other tools added to AgentWorks in this effort. His expertise is focused in human factors and cognitive systems engineering, and he has worked on a range of interface design efforts for model development and visualization.

JONATHAN PFAUTZ is a Principal Scientist and Vice President – Cognitive Systems at Charles River Analytics Inc. in Cambridge, MA. His major research interests include cognition and perception-based visualization and user interface design, cognitive work analysis, social network analysis, human behavior modeling, and the applications of cognitive engineering to software system development across a variety of military and non-military domains. He received his PhD from Cambridge U.

SAM MAHONEY is a Senior Scientist at Charles River Analytics Inc. Mr. Mahoney's areas of expertise include system functional decomposition, data processing architectures, and computational knowledge representations. Mr. Mahoney is leading an effort developing a data-driven world model, capable of automated deduction of complex conditions based on incoming sensor information. Mr. Mahoney graduated from Stonehill College with a B.S. in Computer Science.

Building a Seeing Machine

Yunfeng Li
Tadamasa Sawada
Zygmunt Pizlo
Department of Psychological Sciences
Purdue University
703 Third Street
West Lafayette, IN 47907
765-494-6930
yunfenglee14@gmail.com, tada.masa.sawada@gmail.com, pizlo@psych.purdue.edu

Keywords: Figure-ground organization, 3D shape recovery

ABSTRACT: *It is now commonly acknowledged that intelligent robots will not be able to interact with humans unless they can see as well as we do. Until very recently emulating human visual abilities was considered an insoluble problem. The two main functions of human vision that allow us to operate in everyday life are Figure-Ground Organization and 3D shape recovery. We present the very first computational models of these two functions, compare the performance of the models to the performance of human subjects and implement these models in a robot.*

1. Introduction

The modern scientific study of human visual perception started with the work of Gestalt Psychologists who completely changed the way perceptual psychologists viewed the underlying mechanisms (Wertheimer, 1923/1958; Koffka, 1935). Gestalt Psychologists pointed out that the percept of the visual world is always a combination of the information provided by the retinal image(s) and an *a priori* simplicity principle. By doing this they elaborated the ideas that had been put forth by Ernst Mach (1906/1959). Gestaltists emphasized the importance of perceptual constancies in vision. Consider shape constancy, as their primary example. Shape constancy refers to the fact that the percept of the shape of a given 3D object is constant despite changes in the shape of the object's 2D retinal image, caused by changes in the 3D viewing direction. Perceptual constancies lead to a veridical perception of the world's permanent characteristics. By veridical it is meant that we see things the way they are out there. This paper addresses two fundamental aspects of veridical perception: Figure-Ground Organization and shape constancy.

2. Human Vision as an Inverse Problem

Figure-Ground Organization (FGO) refers to the task of identifying how many objects there are in front of an observer and where they are. Figure 1(a) shows an example. It is easy to see 5 pieces of furniture in the center of the floor. The 6th object is substantially occluded, but with some effort it can also be seen on the back right. The fact that FGO is typically solved by the human visual system so effortlessly is deceptive. From a computational point of view FGO is extremely difficult. The main reason is that FGO is, as most important visual functions, and ill-posed "inverse problem." The classification of problems into direct or forward vs. inverse is due to Tikhonov (Tikhonov & Arsenin, 1977; see also Poggio et al., 1985, for the introduction of this classification into vision science). A forward problem in vision refers to producing a 2D retinal image of a 3D scene. This forward problem is easy (well-posed) because for a given 3D scene and a given viewpoint the 2D image is uniquely specified. In fact, in real imaging systems such as a robot camera or a human eye, it is the laws of optics that "solve" the forward problem (produce the image).

(a) (b)

Figure 1. A pair of stereo images for an indoor scene with six pieces of furniture. The 6th object is in the back right of the scene. It is substantially occluded by the chair in front of it.

An inverse problem in vision refers to inferring the 3D scene from a 2D retinal image. This problem is difficult (ill-posed) because there are always infinitely many possible 3D interpretations (solutions) for a given 2D retinal image. One way to see this is to realize that the object points can be moved on their projecting lines arbitrarily without changing the 2D image. The fact that the human visual system almost always arrives at a single and correct 3D interpretation is truly amazing. The early enthusiasm of the machine vision community in the 1950s and 1960s to solve the vision problems and emulate or even surpass the abilities of the human visual system ended with a complete failure (but see successful examples of 3D interpretation of line drawings, such as the work of Guzman, 1968; Clowes, 1971, Huffman, 1971, Waltz, 1972, and Marr, 1977). As a result, the computer vision community abandoned the real 3D problem about 30 years ago and switched to the task of extracting 2D statistical features from 2D images. The 3D problem has been considered insoluble despite (i) the effort of hundreds, if not thousands of computer vision laboratories around the world, (ii) increasingly better computational and mathematical methods for signal processing and machine learning, and (iii) increasingly faster computers with growing computational power. Interestingly, many psychologists studying human vision followed the lead of the computer vision community and concluded that the 3D problem cannot be solved either by a scientist or by the human visual system. This resulted in a popularity of "multiple view theories" of 3D shape and scene perception, in which it is assumed that our percept of a 3D world consists of 2D representations. This is like claiming that the Earth is flat.

Let us explain in some detail the nature and the degree of the difficulty of FGO. Consider a camera image with 6 million pixels. Such cameras do exist and they provide a reasonable analogy with the human retina which contains 6 million cones. Solving FGO may require evaluating all possible partitions of the 6 million pixels and deciding which partitions most likely represent objects. Ignore for a while how a criterion for making this decision can be constructed. Even if a very good criterion were available, evaluating all possible partitions of a single 2D image cannot be practically done in any reasonable amount of time even if the fastest computers were used. The reason is that the number of all partitions of n elements (called the Bell number) grows with n faster than n!, and n! grows very fast. Any problem that requires performing a number of computations proportional to n! is considered intractable. Even for small n, n! is very large. For example, 61! is equal to 10^{81}, which is equal to the number of atoms in the universe. 6,000,000! is equal to $10^{38,000,000}$. It should be obvious that the number of all partitions of 6 million pixels cannot be analyzed. Brute force approach, based on machine learning methods will not do. It is important to realize that the human visual system solves FGO within a fraction of a second despite the fact that neurons are fairly slow. If a neuron in the human brain is compared to a transistor in a CPU of a computer, then the neurons are 6-9 orders of magnitude slower than the transistors (i.e., 1 million to 1 billion times slower). So, the algorithm used by our visual system to solve FGO must be very smart and if anyone is able to emulate it in a seeing robot, this will be a real breakthrough. Section 3 describes this algorithm and the performance of a robot using it.

Next, consider the second insoluble vision problem, 3D shape recovery. Figure 2 shows a 2D image of a 3D abstract and unfamiliar polyhedron. The reader can surely see the 3D polyhedron, but again the fact that 3D shape perception seems so effortless is deceptive.

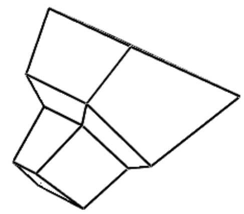


Figure 2. An image of a 3D polyhedron.

Assume that a 3D shape is represented by N points in the 3D space. These points could be points on visible surfaces of the 3D object. If the object subtends the central 20 deg of the visual field, as many as half of the 6 million cones are stimulated because most of the cones are in the central part of the human retina. So, N could be as large as 3 million. Now, as already Bishop Berkeley (1709) pointed out, each retinal point could be an image of any of the infinitely many points in the 3D space "out there", all the points being located on the line emanating from the retinal point and going through the center of perspective projection in the eye. Assume that the visual system tries to reconstruct the 3D points with spatial resolution of 1mm. Furthermore, assume that the object in front of the observer has the range in depth of 1m. This means that instead of considering infinitely many points on each projecting line, we *only* need to consider 1000 points. It follows that for a given 2D retinal image represented by 3 million points, the number of all possible 3D interpretations is $1000^{3,000,000}$ which is equal to $10^{9,000,000}$. Another astronomically large number. We know that the human visual system solves this insoluble problem in a fraction of a second despite the fact that the neurons in the brain are quite slow. How this is done will be described in Section 4. The last section will provide evaluation of what it means to have a machine that sees like us and what, if anything is missing in our effort of emulating human vision.

3. Algorithm for Solving Figure-Ground Organization Problem

As with every ill-posed inverse problem, a successful solution critically depends on the ability to impose effective *a priori* constraints on the family of possible interpretations. This is also true with FGO. There must be a way to restrict the number of all partitions of 6,000,000 cones to just one. What are those constraints? First, an *object always comes in one piece*, rather than in many spatially discontinued pieces. Second, *objects are typically closer to the observer, than the background*. Assume that each object subtends a solid angle of 10 by 10 deg (this area is about the size of one's hand at the arm's viewing distance). If there was no occlusion, we could stack about 200 such objects in front of the observer (assuming that the visual field is equivalent to a surface of a hemisphere, whose solid angle is 20,626 deg^2). So, if an observer is trying to find objects in front of him, he will need to examine only 200 spatial neighborhoods on the retina, rather than $10^{38,000,000}$ possible partitions. Furthermore, these 200 spatially separate neighborhoods can be analyzed simultaneously in the human visual system due to the massively parallel architecture of the visual system. Clearly, *a priori* constraints can be quite effective. The constraints of spatial contiguity and large size dramatically reduced computational complexity of the problem. Next, we will describe which constraints are needed to actually find objects in the 3D scene and in the 2D image.

Consider the case of a binocular observer, or a monocular active observer. We simulate this case using a Pekee II robot equipped with a BumbleBee stereo camera. The robot's height is about 1m and the two lenses of the stereo camera are separated by 12 cm, which is roughly twice as large as the separation between the human eyes. The robot does not use any other sensors. In particular, the robot does not use the laser range sensor for reconstructing depth map. The robot looks at an indoor scene containing children furniture like that in Figure 1. Its task is to determine the number, positions and sizes of objects. Note that the floor is highly textured and it contains shadows and specular reflections. All this makes it very difficult to detect objects in the 2D images using conventional methods of texture and contour analysis. Building on Julesz's (1971) powerful demonstrations, in which a human observer was able to solve the binocular correspondence problem with random dot stereograms, our robot begins with establishing binocular correspondence of texture points, using the Sum of Absolute Difference Correlation algorithm (SADC) (Wong, Vassiliadis, & Cotofana, 2002). This stage is not error free, of course and it cannot be the basis of reliable 3D scene reconstruction. But the robot's goal is to solve FGO, not to reconstruct the 3D scene. Because objects are always spatially contiguous and important objects tend to be large (see the previous paragraph), the objects can be detected in the scene because there are always large number of depth samples close to each other in the 3D space. Using the distance between its cameras, the robot computes a 3D depth map (scales the binocular disparities) of visible points. The next step is to improve the signal to noise ratio by using another set of *a priori* constraints.

The next set of constraints is the *a priori* knowledge that all objects rest on a *common horizontal ground* due to gravity. *The ground is at a known distance below the cameras* (this distance is called the height of the observer). Using these two constraints, it is natural to estimate, as the next step, the floor in the 3D depth map and eliminate it from further processing (Faugeras, 1993, p.209). This is quite easy because in a typical scene many 3D points are actually floor points. Furthermore, if the robot, like a human observer, knows the *orientation of its cameras relative to the gravity*, the estimation of the floor calls for nothing more than fitting a known plane to 3D points and determining which points are close to the floor or below the floor (3D points below the floor represent noise). All these points are removed from the depth map because the floor points represent background and we are interested in detecting objects, called "figures". It is important to point out that a richly textured floor is not a problem for our robot. In fact, the texture is actually helpful for establishing the binocular correspondence. In contrast, conventional algorithms cannot work with richly textured backgrounds because this makes the separation of figure from ground impossible.

After floor points are removed from the 3D depth map, only object points remain. Now, the robot "mentally" rotates the remaining 3D depth map to simulate viewing the scene from above. There are two good reasons to perform this rotation. First, *all natural objects have prominent vertical structures* such as legs, surfaces and edges. This is true about animal and human bodies, as well as architectural constructions and furniture. In the presence of vertical gravitational force, vertical legs and surfaces are mechanically more stable. Because there are many vertical structures in the natural environment, there are many texture points representing these structures that lead to our 3D depth map. When these 3D points are projected orthographically onto a horizontal surface, such as floor, there is a very strong signal indicating the presence of the objects.

The second reason to perform "mental rotation" to simulate looking at the scene from above is related to the fact that most *objects reside on a common horizontal ground plane*. Exceptions are lamps hanging from the ceiling or a book lying on a desk. This means that if the 3D scene is actually viewed from above there will not be many partial occlusions of some objects by others. Partial occlusions are a rule rather than exception in ordinary visual images like that in Figure 1. Occlusions are common because some objects are farther away than others from the observer. If a camera were mounted on the ceiling, there will not be many occlusions. Instead of mounting a physical camera on the ceiling (which is not practical), the robot "mentally" rotates the visible 3D points to simulate looking from above.

The result of the orthographic projection of the 3D points in our depth map on the horizontal plane is shown in Figure 3. When the viewing distance is large, the depth error from binocular disparity becomes large, too. Therefore, in this example, the algorithm only detects the objects that are within 4 meters in front of its cameras.

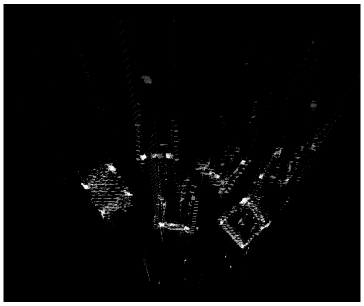


Figure 3. Orthographic projection of 3D points representing objects in Figure 1 on the horizontal plane.

Now is the time to identify individual objects in the 3D scene because these objects correspond to clusters of points. Any of the standard clustering methods can be used, but additional *a priori* constraints can greatly improve the result. For example, we expect *rectangular objects* whose *size is within some range*. We assume that the strong signal in the orthographic projection is caused by the vertical structures of the object. As a result, the projection of the 3D points onto the floor represents the shapes of the orthographic projection of the individual objects. So, we fit rectangles to the projection and estimate the position, size and orientation of each rectangle. The result of such fitting is shown in Figure 4. The green boxes show the fitted rectangles and the red boxes show the ground truth.

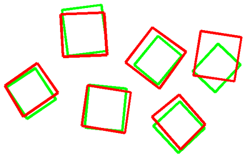


Figure 4. The green rectangles were fitted to the points in Figure 3. The red boxes show the ground truth.

In the case of the severely occluded object (the rightmost stand in the second row), our algorithm detects it and computes its position successfully, although it fails to compute its orientation accurately. This failure results from small amount information about this object. At this stage we can claim that our *robot solved FGO in the 3D representation*, on the floor. This solution can be used for planning visual navigation in the scene.

We want to emphasize one new and very important aspect of the 3D FGO. As you can see from Figure 4, the robot produces a spatially global map of its environment (the floor plan) from a single viewing position. Specifically, the robot recovers the invisible spaces behind the objects. As a result, the robot knows how much space is between objects regardless whether this space is directly visible or not. This is essential because it allows the robot to plan its navigation path in the scene even before it starts to move. This contrasts with conventional SLAM (Simultaneous Localization And Mapping) methods where the robot has to explore the environment and to reconstruct the visible surfaces from different places in advance to build the map and, at the same time to localize itself (Durrant-Whyte, Bailey, 2006). In dynamic environments, building spatially global map must be "instantaneous". Otherwise, by the time the map is produced, the environment would have changed. But solving 3D FGO is not the end of processing. More can be accomplished with these results.

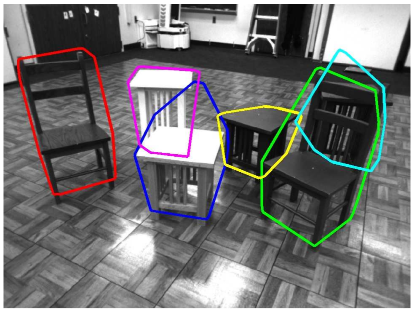


Figure 5. Detected objects in the image are circumscribed by color polygons.

The robot has an estimate of each object's height, which means that a 3D circumscribing box can be formed for each object. We then project the occluding boundaries of these boxes to one of the 2D perspective images that were used to solve the 3D FGO. The result is shown in Figure 5. The color curves in the image are estimated convex hulls of 2D projections of original objects. Some of these curves partially overlap as they should because the objects themselves partially overlap. These color curves represent the *solution of FGO in the 2D image.*

Figure 6 is the diagram of the algorithm and it shows how to identify individual objects from a pair of stereo images. The algorithm was tested on a DELL T5500 computer, and on average, it can acquire and process about 4 pictures (with the resolution of 512x384 pixels) per second.

Once we know which regions in the 2D image represent individual objects, we can proceed with extracting important 2D contours that represent essential aspects of the 3D shape. How a 3D shape can be recovered from a single 2D perspective image will be described next.

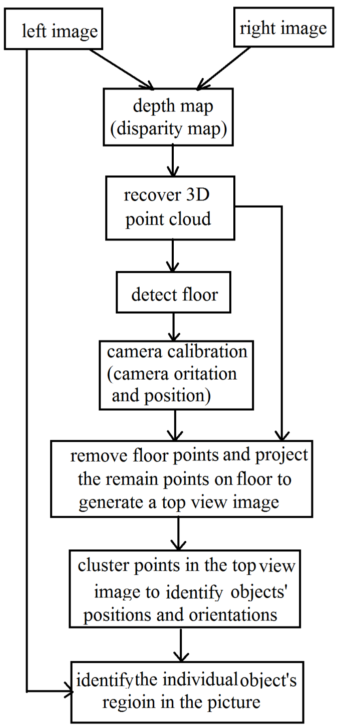


Figure 6: The diagram of the algorithm.

4. Algorithm for recovering a 3D shape from a single 2D image

Unlike all prior algorithms and models for 3D shape reconstruction, we begin by asking about the nature of *a priori* constraints, rather than about the nature of visual data. Constraints proved so useful in solving FGO that the reader should be confident that they will be essential in 3D shape recovery, as well. Computational complexity of 3D shape reconstruction is so large (see Section 2) that without *a priori* constraints it just does not seem possible to reconstruct 3D shapes and scenes accurately and reliably. This observation has been supported by enormous amount of results, both theoretical and empirical. In human vision, researchers have been testing for dozens of years the observer's ability to judge depth relations among points and surfaces, as well as the ability to judge 3D orientation of surfaces. Most of these experiments were done with amorphous stimuli that precluded the visual system from using *a priori* constraints. This was done on purpose; the researchers wanted to study 3D perception not "contaminated" with *a priori* constraints. These experiments universally showed that perception of depth and surface orientations is very poor: there are large systematic errors across viewing conditions, across observers and even across replications of the same experiment with the same observer. These results should have provided a warning sign that visual data without *a priori* constraints is not the way to go. But the researchers wanted to study the most general type of stimulus devoid of familiarities or constraints of any kind. But shapes of natural objects are not devoid of regularities; shapes of all animals are mirror symmetrical. Their parts, as pointed out by Biederman (1987), conform to translational symmetry (Biederman called the family of shapes that are used to represent parts, "geons"). Finally, flowers often represent rotational symmetry. Once we acknowledge that most, if not all important natural object are symmetrical, then the symmetry *a priori* constraint is no longer a "contamination" of one's experiment, but it becomes its essential part. Using Brunswik's (1956) terminology, a stimulus in one's laboratory experiment must be "ecologically valid." How far can one go with a symmetry constraint? The answer is, all the way. The symmetry constraint can often reduce the enormous family of possible 3D interpretations to a unique and the correct one.

Similar efforts with similar outcomes took place in the machine vision community. The researchers became aware of such *a priori* constraints as symmetry at least as early as 1981(Kanade, 1981; Gordon, 1989) , but the use of strong *a priori* constraints has not become the main stream of research. Marr's (1982) paradigm, with its emphasis on surfaces, dominated the field. When this paradigm failed, instead of exploring the role and availability of constraints, machine vision community switched to 2D operations on 2D images. The hope was that modern machine learning methods will be able to extract invariant signatures of 3D objects. Despite some moderate progress, recognition performance of these 2D "appearance models" did not come even close to the performance of human observers. During the last decade, the present authors have provided psychophysical evidence showing that *a priori* constraints such as 3D symmetry and planarity of contours are essential in 3D shape perception (Pizlo, 2008; Pizlo et al., 2010). If these constraints cannot be applied to the family of possible 3D interpretations of a 2D image, shape constancy performance is at chance level. When these constraints can be applied, performance is close to perfect. So, now we do not have to deal with a question as to whether or not constraints should be used. They should and they are used. The real question is how to design a 3D shape recovery model whose performance will match that of a human observer.

Consider a set of N points in a 3D space forming a mirror symmetric configuration. This means that there is a plane in 3D such that N/2 of the 3D points are mirror images of the other N/2 points with respect to this plane. Recall that in the absence of a symmetry constraint, when the task is to reconstruct the depth of N points from a single 2D perspective image, there are N free (unknown) parameters, the depth values of all N points. When these N points form a 3D mirror symmetrical configuration, there is no unknown! The 3D configuration is uniquely recovered! Recall the degree of uncertainty that we estimated in Section 2 for the case of 3D shape recovery. It should now be obvious that when a designer of a robot vision system is faced with a decision of adding additional visual data vs. an effective *a priori* constraint, she should choose the latter. This is what we did.

We showed not only how to apply a symmetry constraint to a 2D perspective image, but also to a 2D orthographic image (Li et al., 2009; see also Vetter & Poggio, 1994). Next, we showed how the human visual system combines the 3D symmetry constraint with two 2D retinal images (Li et al., 2011). This is the case of a binocular observer or an active monocular observer. Besides 3D symmetry and planarity of contours constraints, we showed that the human visual system uses 3D compactness constraint (Li et al., 2009). 3D compactness is a well known concept in mathematical physics (Polya & Szego, 1951), but it has never been used as a constraint in visual perception. A 3D compactness is defined as V^2/S^3, where V and S are the volume and the surface are of an object or of its convex hull.

Figure 7 shows three views of a 3D shape recovered by our model based on the 2D image shown in Figure 2. We want to point out two important aspects of our shape recovery model: (i) it recovers 3D shape without measuring 3D distances; 3D distances can be reconstructed after the 3D shape is recovered, and (ii) it often recovers the back invisible parts of the object as well as its front visible ones. This contrasts with Marr's (1982) 2.5D sketch. According to Marr, the visual system can inform the observer only about the visible surfaces of the 3D object. The back, invisible ones are completed by memory. Our robot can "see" the entire object. This is critical because it gives the robot knowledge of where the object ends on its back, "invisible" side. Note that so far, the symmetry correspondences for the image of a 3D shape have been established manually. Once the symmetric points are identified in the 2D image, the recovery is instantaneous.

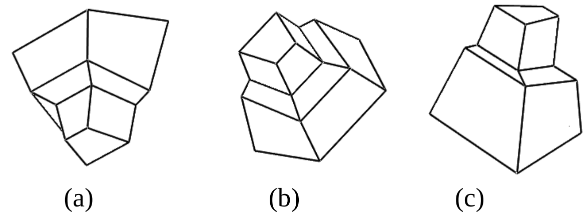


Figure 7. Three views of the recovered polyhedron from the 2D image in Figure 2.[1]

To compare the recovery of our model with human performance, we measured the subjects' percept of the symmetric polyhedra, like that in Figure 2, in binocular and monocular viewing conditions. In the case of binocular condition, the subject viewed the stereo images of polyhedra through stereoscopic shutter glasses. On each trial, two objects were shown. The reference 3D shape was a stationary object shown on the left (monocularly or binocularly). The test 3D shape was shown on the right monocularly. The test shape was rotating around the vertical axis so that the subject could see many views of this shape. The slant of the symmetry plane of the reference 3D shape was 15, 30, 45, 60 or 75 degrees. The subject was asked to adjust the aspect ratio of the test 3D shape so that it matched the reference 3D shape. The rationale behind this adjustment task is related to the fact that a single 2D orthographic image of a symmetric 3D shape, determines this shape up to only one free parameter – its aspect ratio (Vetter & Poggio, 1994). Four subjects were tested and each ran 100 trials for each viewing condition (20 trials per each of the five slants).

Figure 8 shows the comparison of monocular and binocular performance of a human subject in 3D shape recovery to the performance of our model. The bottom line is that the model's performance is very close to the subject's performance. Specifically, in monocular viewing, there are systematic errors in recovering the aspect ratio of the 3D shape when the slant of the symmetry plane is close to 0 or 90 deg. Slants 0 and 90 deg represent "degenerate views". They are called degenerate because with such views the 3D symmetry constraint is ineffective. Note that the systematic errors of the model are very similar to the systematic errors of the subject. In binocular viewing, all systematic errors disappear. Both, the subject and the model see the 3D shapes veridically. This is the first empirical and computational demonstration of perfect 3D shape perception.

[1] Our polyhedral stimuli consisted of two boxes whose bottom faces were coplanar. So, each polyhedron had 28 edges and all of the edges were included in the object's representation, despite the fact that some of them were coplanar, like those seen in Figure 7c.

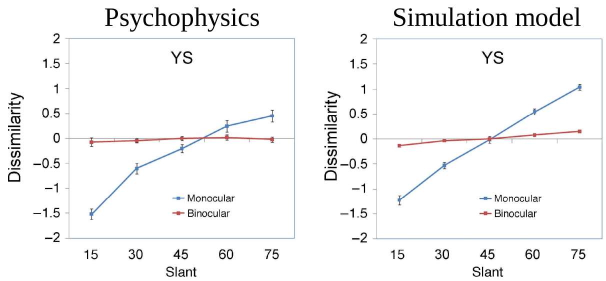


Figure 8. Performance of one subject (YS) is shown on the left. Performance of our model simulating YS's performance is shown on the right (from Li et al., 2011). The subject and the model performed a 3D shape recovery task based on the information provided by one or two images (monocular vs. binocular viewing). The horizontal axis shows the slant of the symmetry plane of the 3D reference shape. The vertical axis shows the error in the recovered aspect ratio of the test 3D shape using a log scale.

5. Conclusions

We explained two fundamental functions of the visual system: Figure-Ground Organization and 3D shape recovery. These two functions have been universally considered insoluble problems because of their inherent ill-posedness: there are simply too many possible solutions. We showed that these problems can be solved and a unique and correct interpretation produced when effective *a priori* constraints are used. We "explained" these functions in the sense that we formulated computational models and implemented them in a seeing robot. By doing this, we followed Richard Feynman's proposal "what I cannot create, I don't understand".

Is there any other problem to solve before the robots can interact with us? There is only one such problem. To apply the symmetry *a priori* constraint, the robot must be able to detect where, in the 3D object, the symmetry transformation is present. This is not trivial because the 2D image of a 3D symmetrical object is, itself asymmetrical. To detect a 3D symmetry from perspective images, one has to use perspective invariants of symmetry and formulate a robust method of using them with real images of real scenes. We have already made good progress in solving this problem. For more examples illustrating how our algorithm solves FGO problem, as well as demos of 3D scene and shape recovery of synthetic and real objects, as well as robot navigation, please refer to the following links http://web.ics.purdue.edu/~li135/Demo.html and http://web.ics.purdue.edu/~li135/Demo/People/objdet.mov.

6. References

Berkeley, G. (1709/1910). *A new theory of vision.* New York: Dutton.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review, 94 (2)*, 115-147.

Brunswik, E. (1956). *Perception and the representative design of psychological experiments.* Berkeley: University of California Press.

Clowes, M.B. (1971) On seeing things. *Artificial Intelligence, 2*, 79-116.

Durrant-Whyte, H., Bailey, T. (2006). Simultaneous Localization and Mapping (SLAM): Part I the Essential Algorithms. *Robotics and Automation Magazine 13(2)*, 99-110.

Faugeras, O. (1993). *Three-dimensional computer vision: A geometrical viewpoint*. Cambridge: MIT press.

Gordon, G. G. (1989). Shape from symmetry. In *Proceedings of SPIE, Intelligent Robots and Computer Vision VIII: Algorithms and Techniques, 1192*, 297-308.

Guzman, A. (1968) Decomposition of a visual scene into three-dimensional bodies. *Proceedings of AFIPS Conference, 33*, 291-304. Washington, DC: Thompson.

Huffman, D.A. (1971) Impossible objects as nonsense sentences. In: Meltzer, B. & Michie, D. (Eds.), *Machine Intelligence, vol. 6* (pp. 295-323) Edinburgh: Edinburgh University Press.

Julesz, B. (1971). *Foundations of cyclopean perception.* Chicago: University of Chicago Press.

Kanade, T. (1981) Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence, 17*, 409-460.

Koffka, K. (1935) *Principles of Gestalt Psychology*. New York: Harcourt, Brace.

Li, Y., Pizlo, Z. & Steinman, R. M. (2009). A computational model that recovers the 3D shape of an object from a single 2D retinal representation. *Vision Research, 49 (9)*, 979-991.

Li, Y., Sawada, T., Shi, Y., Kwon, T. & Pizlo, Z. (2011). A Bayesian model of binocular perception of 3D mirror symmetrical polyhedra. *Journal of Vision, 11 (4):11*, 1-20.

Mach, E. (1906/1959) *The Analysis of Sensations*. New York: Dover.

Marr, D. (1977) Analysis of occluding contour. *Proceedings of the Royal Society of London, B 197*, 441-475.

Marr, D. (1982). *Vision*. San Francisco: Freeman.

Pizlo, Z. (2008). *3D shape: Its unique place in visual perception.* Cambridge: MIT press.

Pizlo, Z., Sawada, T., Li, Y., Kropatsch, W. & Steinman, R. M. (2010). New approach to the perception of 3D shape based on veridicality, complexity, symmetry and volume. *Vision Research, 50 (1)*, 1-11.

Poggio, T., Torre, V. & Koch, C. (1985). Computational vision and regularization theory. *Nature, 317 (6035),* 314-319.

Polya, G. & Szego, G. (1951). *Isoperimetric inequalities in mathematical physics.* Princeton: Princeton University Press.

Tikhonov, A. N. & Arsenin, V. Y. (1977). *Solutions of ill-posed problems.* New York: Wiley.

Vetter, T. & Poggio, T. (1994). Symmetric 3D objects are an easy case for 2D object recognition. *Spatial Vision, 8 (4)*, 443-453.

Waltz, D. (1972/1975) Understanding line drawings of scenes with shadows. In: Winston, P.H. (pp. 19-91) New York: McGraw-Hill.

Wertheimer, M. (1923/1958) Principles of perceptual organization. In: D.C. Beardslee & M. Wertheimer (Eds.) *Readings in Perception*, pp. 115-135. NY: D. van Nostrand.

Wong, S., Vassiliadis, S., & Cotofana, S. D. (2002). A sum of absolute differences implementation in fpga hardware. *Proceedings 28th EUROMICRO conference.* 183-188.

Acknowledgements

This project was supported by AFOSR, DOE, DOD, NSF.

Author Biographies

YUNFENG LI is a postdoctoral research fellow in the Department of Psychological Sciences at Purdue University. He received his Ph.D. degree in Psychology from Purdue University in 2009. His research interests are in 3D shape and scene recovery, figure-ground organization, invariants of symmetry. His work combines psychophysical experiments, computational and mathematical modeling.

TADAMASA SAWADA is a postdoctoral research fellow in Department of Psychological Sciences at Purdue University. He received a Ph.D. degree in Psychophysics from Tokyo Institute of Technology in 2006. He has been working on human perception of 3D shape, 2D and 3D symmetry, figure-ground organization and perception of 3D scenes. His work involves psychophysical experiments and computational modeling.

ZYGMUNT PIZLO is a Professor of Psychology and of Electrical and Computer Engineering at Purdue. He received his Ph.D. degree in Electronic Engineering in 1982 and his Ph.D. degree in Psychology in 1991. He served as the president and vice president of the Society for Mathematical Psychology. He published over 100 journal and conference papers on all aspects of visual perception, as well as motor control, problem solving and image quality. In 2008 he published the first monograph on human shape perception.

IV. POSTERS

Papers are in order as they appear on the agenda.

A Demonstration for BRIMS: New Enhancements to MindModeling@Home Project

Jack Harris, Thomas Mielke, L. Richard Moore, Jr., Clayton Stanley, Thomas Olaes

The MindModeling@Home[1] project (http://MindModeling.org) strives to provide a simple, efficient and reliable resource for evaluating models of cognition. By uniting heterogeneous computational resources ranging from high performance computing systems to local grids to workstations of volunteers around the globe, the MindModeling@Home system avails large-scale computational power that enables exploration of models at a deeper level and across a larger number of contexts than is typically possible using a single research workstation. Over the past 4 years, the MindModeling@Home system has facilitated billions of cognitive model runs, using thousands of computers located in dozens of countries. The system has contributed to a broad spectrum of research, ranging from robust decision making to understand the underlying cognitive mechanisms of fatigue.

For BRIMS, we would like to present a full-system usage demonstration of MindModeling@Home, with an emphasis on some of its newest features. We will submit a cognitive model to the system for evaluation, show how it is executed across distributed MindModeling@Home resources, and demonstrate real-time visualization of the data as it is acquired.

Specifically, a web portal will be utilized to demonstrate the ease by which an ACT-R model can be submitted to the system for evaluation across an array of different contexts. Following submission, resources from around the globe will be utilized to execute model simulations and a world map will be updated to display where models are currently being executed. A local computing resource will also be online to demonstrate what takes place on these distributed resources during model execution. Finally, as model simulations complete, the resulting data will automatically be consolidated and presented to the Hurricane visualization system. Hurricane[2] provides an interactive 3-D data visualization environment for exploring model data and identifying optimal model parameterizations.

This exhibition will demonstrate the utility of the MindModeling@Home system for providing easy access to large-scale computation resources for cognitive model execution, and also for providing a means for visually analyzing the resulting data.

---

[1] Harris, J. (2008). MindModeling@Home: A large-scale computational cognitive modeling infrastructure. In *Proceedings of the 6th Annual Conference on Systems Engineering Research*. Los Angeles, CA

[2] Moore, L. R. (2010). Cognitive model exploration and optimization: a new challenge for computational science. In T. Jastrzembski (Ed.), *Proceedings of the 2010 Behavior Representation in Modeling and Simulation (BRIMS) Conference*. Orlando,FL: Simulation Interoperability Standards Organization.

Annual Cognitive Modeling Competition

Kevin Gluck
Air Force Research Laboratory
711 HPW/RHAC
2620 Q Street, Bldg 852
Wright-Patterson AFB, OH 45433-7955
937-938-3552
kevin.gluck@wpafb.af.mil

Keywords:
Cognitive, Modeling, Competition

1. Opportunity

Although the cognitive and behavioral modeling communities now have a rather lengthy history and there is ongoing research and development in these areas across dozens of academic, industrial, and government research laboratories, very little of the work has been explicitly competitive in orientation. That said, there are precedents for modeling competitions in this area. Two examples include the PokerBot Competition (Lebiere & Bothell, 2004) and the Dynamic Stocks and Flows Model Comparison Challenge (Lebiere, Gonzalez, & Warwick, 2010). Both of these were successful and interesting events. However, they were also both single shot modeling competitions that did not evolve into annual events in the spirit of the Robocup (2012) robotic soccer competitions.

I propose it is time to establish a recurring annual cognitive modeling competition. This poster is an opportunity for community discussion of the pros, cons, infrastructure requirements, and design parameters that should be considered in developing such an event within the cognitive and behavioral sciences.

2. References

Lebiere, C., Gonzalez, C., & Warwick, W. (2010). Cognitive architectures, model comparison, and AGI. *Journal of Artificial General Intelligence, 2*, 1-19. (see also http://www.hss.cmu.edu/departments/sds/ddmlab/modeldsf/index.html)

Lebiere, C., & Bothell, D. (2004). Competitive modeling symposium: Pokerbot World Series. In *Proceedings of the 2004 International Conference on Cognitive Modeling*. p.32. Mahwah, NJ: Erlbaum. (see also http://www.lrdc.pitt.edu/schunn/ICCM2004/index.html

RoboCup. (2012). RoboCup. Retrieved from the web January 6, 2012. http://www.robocup.org/

Author Biography

KEVIN GLUCK is a Senior Cognitive Scientist in the Cognitive Models and Agents branch at the Air Force Research Laboratory. He chairs AFRL's Robust Decision Making Strategic Technology Team and is leading the Human Effectiveness Directorate's Cognitive Modeling Expansion initiative.

Effects of Load on Movement-Related Operational Tasks in Agent-Based Constructive Simulation

Daniel Rice
Mitha Andra
Cortney Eldridge
Alexander Kennedy
Adam Peloquin
Technology Solutions Experts, Inc.
209 West Central Street, Suite 300
Natick, MA 01760
508-655-2232
daniel.rice@tseboston.com, mitha.andra@tseboston.com, cortney.eldridge@tseboston.com, alex.kennedy@tseboston.com, adam.peloquin@tseboston.com

Steven Yuhaski, Jr.
Natick Soldier Research, Development, and Engineering Center (NSRDEC)
1 Kansas Street
Natick, MA 01760
508-233-5072
steven.yuhaski@us.army.mil

Keywords:
encumbrance, load, constructive simulation

1. Introduction

Soldiers today are faced with a number of challenges related to the operational environment (materiel, strategic, personnel, etc.) that impact task performance and overall mission effectiveness. These challenges are exacerbated by the effects of Soldier equipment load on task performance. “Weight remains a major obstacle for the Army as it tries to equip soldiers with all of the gear needed to remain safe and connected to other soldiers on the battlefield – medics and mortar operators in Afghanistan are carrying the most weight: roughly 133 pounds for a three-day mission” (Brannen, 2010). With each additional piece of vital equipment, the individual Soldier’s load increases, thus impacting Soldier movement behavior and task performance. The result of this tradeoff between a lighter load and carrying more necessities can mean the difference between life and death.

Technology Solutions Experts, Inc. (TSE) is evaluating the tradeoff between Soldier load and mission effectiveness within operational environments to provide decision-support tools to military analysts and decision makers. TSE developed the Soldier Load and Speed-Regulation methodology and implemented it in the Infantry Warrior Simulation (IWARS), a force-on-force, agent-based constructive simulation. The prototype implementation of the algorithm developed to evaluate Soldier task performance under the burden of equipment load in a simulation tool provides cost-effective and operationally functional scenarios to investigate many different circumstances with varied environmental, materiel, strategic, personnel, and task factors that impact task performance and overall mission effectives. Furthermore, Modeling and Simulation (M&S) solutions are being used increasingly by government departments such as the Department of Defense (DoD) and the Department of Homeland Security (DHS) to aid with decision challenges regarding personnel and equipment interaction in operational environments (Corley & Lejerskar, 2003).

Using the Soldier Load and Speed-Regulation methodology in simulation, military analysts and decision makers can evaluate the tradeoffs among various equipment configurations, operational environment, and Soldier task performance. This tradeoff analysis helps to identify critical relationships between the Soldier and the operational equipment, which plays an essential role in military mission planning and equipment-acquisition decisions.

2. Soldier Load and Speed-Regulation Methodology

TSE collaborated with the Natick Soldier Research, Development, and Engineering Center (NSRDEC) Modeling and Analysis (M&A) Team to develop the

Soldier Load and Speed-Regulation methodology. This methodology incorporates algorithms that predict sustainable movement rates for short- and long-duration combat tasks and evaluates the tradeoff between additional Soldier load and the speed the Soldier can sustain for given movement-related tasks. TSE used Equation 2.1 (Pandolf, 1976) and *Army Field Manual 21-18* (Army, 1990) as a basis to form this model. The relevant Soldier and environmental factors considered in the equation are described in Table 2.1.

$$M=1.5W+2.0(W+L)(L/W)^2+\eta(W+L)[1.5V^2+0.35GV]$$

Equation 2.1

Soldier factors	Environmental factors
M, metabolic rate, watts	G, terrain grade, %
W, weight of Soldier agent, kg	η, terrain factor
L, load carried, kg	
V, velocity of walking, m/s	

Table 2.1

The Soldier Load and Speed-Regulation methodology includes several Soldier agent behavior algorithms: (1) speed regulation over the course of time for short-duration (0-30 min.) tasks; (2) speed regulation over the course of time for long-duration (30 min. or longer) tasks; and (3) the impact of a Soldier agent's prior movement during a given scenario on a Soldier agent's current movement rate. This methodology calculates the maximum sustainable movement speed for 0 to 10 hours of activity and incorporates smoothed average functions to account for prior movement speed when predicting the maximum sustainable metabolic rates.

3. Implementation

TSE implemented the Soldier Load and Speed-Regulation methodology in IWARS and created a behavior template to be used in the mission builder. To use this model, users must include the template as part of the overall mission. Figure 3.1 shows the details of the behavior template.

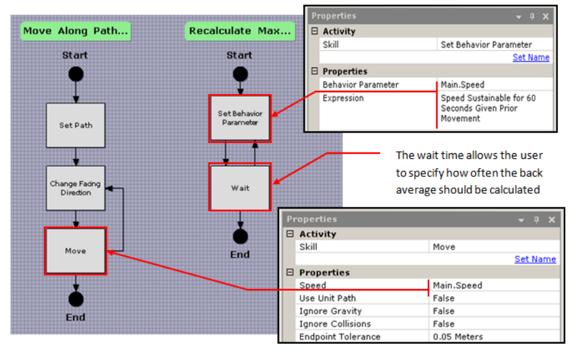


Figure 3.1 IWARS implementation – behavior template

The implementation of the methodology was internally tested and verified. TSE conducted a comparitive analysis to test the predictive Soldier movement rate using data resulting from the Soldier Load and Speed-Regulation methodology versus the existing IWARS methodology, using a t-Test to see if statistical significance existed at the 95% confidence interval. The comparison of the task completion times using the two different methodologies revealed statistically significant differences between the data.

4. Conclusion

The Soldier Load and Speed-Regulation algorithms implemented in IWARS overcome challenges in existing simulations to represent the impact of Soldier equipment load on movement rate. The methodology predicts sustainable movement rates for short- and long-duration tasks and represents the relationship between prior movements and future movement ability during a given scenario.

5. Future Work

Currently, TSE is working on including other environmental factors that may affect sustainable movement rate such as altitude and its effect on maximum aerobic capacity and sustaining movement rate. TSE also is developing the methodology further to consider short-duration/high-intensity movements affected by anaerobic supply (e.g., sprinting, digging trench, etc.) to encompass both anaerobic and aerobic energy.

6. References

Army, U. S. (1990). *Army Field Manual 21-18 Foot Marches.* Washington, DC: Dept. of the Army.

Brannen, K. (2010). *ArmyTimes*. Retrieved June 2011: http://www.armytimes.com/news/2010/10/defense-army-lightening-soldier-load-101310/

Corley, J., & Lejerskar, D. (2003). Homeland Defense Center Network – Capitalizing on Simulation, Modeling and Visualization for Emergency Preparedness, Response and Mitigation. *2003 Winter Simulation Conference*, (pp. 1061-1067).

Pandolf, K. G. (1976). *Predicting energy expenditure with loads while standing or walking very slowly.* Natick: USARIEM.

Author Biographies

STEVEN YUHASKI, JR. is an operations research analyst at the Natick Soldier Research, Development, and Engineering Center (NSRDEC) in Natick, MA. He has extensive experience in formulating mathematical models of physical and logistical activities and in developing and applying heuristics for the design and optimization of those systems, including the Infantry Warrior Simulation (IWARS). He has experience in artificial neural network (ANN) design and application, network graph theory, and genetic programming heuristic development and application. He holds a Ph.D. in industrial engineering & operations research from the University of Massachusetts, Amherst and an M.E. in mechanical engineering from the University of Florida.

DANIEL RICE is vice president and the director of research and development at Technology Solutions Experts, Inc. (TSE) in Natick, MA. He conducts research in operations, information systems, and computer security. He holds a Ph.D. in information systems and an MBA in finance from the University of Connecticut and a B.S. in naval architecture and marine engineering from the United States Coast Guard Academy.

MITHA ANDRA is a senior operations research analyst at Technology Solutions Experts, Inc. (TSE) in Natick, MA. Her work has included research, analysis, development, and evaluation of systems, procedures, and models in assigned functional areas including operations analysis and modeling and simulation (M&S). She focuses much of her work on developing models and implementing recommendations for the Infantry Warrior Simulation (IWARS). She holds an M.S. in industrial engineering from Northeastern University.

CORTNEY ELDRIDGE is a senior operations research analyst at Technology Solutions Experts, Inc. (TSE) in Natick, MA. She researches, analyzes, develops, and evaluates systems, procedures, and models in assigned functional areas such as operations analysis, modeling and simulation, and data fusion. Much of her work is focused on developing models and implementing recommendations for the Infantry Warrior Simulation (IWARS). She holds an M.S. in operations research from Northeastern University.

ALEXANDER KENNEDY is an operations research analyst at Technology Solutions Experts, Inc. (TSE) in Natick, MA. He has contributed extensively to developing algorithms and models for implementation in the Infantry Warrior Simulation (IWARS). As part of the Task Related Effects of Encumbrance Methodology (TEEM) project, he contributed to the development of a performance degradation model to represent the interactions between Soldiers, materiel, environment, and tasks. He holds a B.S. in industrial engineering and a B.A. in economics, both from the University of Massachusetts, Amherst.

ADAM PELOQUIN is a senior operations research analyst at Technology Solutions Experts, Inc. (TSE) in Natick, MA. His experience includes verification, validation, and quality-assurance testing for the Infantry Warrior Simulation (IWARS), feature and design recommendations, and the creation of complex behaviors for simulation studies. His other research experience includes topics such as safety, quality assurance, ergonomics, process improvement, and inventory reduction. He holds a B.S. in industrial engineering and operations research from the University of Massachusetts, Amherst.

A Preliminary Assessment of Adversary Behaviour Using Synthetic Environments

Andrew Green
Dr Beejal Mistry
Dstl
Human Systems Group
Information Management Department
Portsdown West
Hampshire PO17 6AD, UK
+44 (0) 2392 53 7597, +44 (0) 2392 53 7489
apgreen@mail.dstl.gov.uk, bmistry@mail.dstl.gov.uk

Keywords:
Synthetic Environment, adversary behaviour, decision- making

1. Introduction

Understanding the behaviour of adversaries is vital to understanding how they may be deterred, and to develop strategies to inform the development of transport security policy. Currently there is no existing method to investigate the behaviours and decision-making of adversaries undertaking tactical-level offensive operations against transport security infrastructure. Testing the deterrent impact of security interventions will enable identification of the obstacles faced in securing national rail infrastructure, protecting the public and maintaining a balance between deterrence, detection and public reassurance, in order to ensure a safe and fully-functioning rail network. For those involved in the development of transport security strategy policy making, a Synthetic Environment (SE) is currently considered a cost effective method of exploring these behaviours and decision making.

1.1 Aim

This preliminary experiment aimed to evaluate the deterrent impact of security interventions on adversary decision-making and behaviour. The primary aims of this experiment were:

To compare the impact on behaviour and decision-making of different types of security intervention, for example presence of police officers.

To compare the impact on behaviour and decision-making three adversary activities:

a) Hostile Reconnaissance (HR);
b) Person Bourne Improvised Explosive Device (PBIED);
c) Left Device (LD).

To compare the impact of the above security interventions on three adversary activities.

The secondary aims of the experiment were to; assess the impact of the security intervention on the daily commuter; and to test the effectiveness of immersion strategies to engage participants into the SE.

Overall it was expected that the study would help create a SE that could determine the effectiveness of proposed deterrence and detection interventions in a safe, repeatable and auditable way, to inform future transport security.

A Concept Demonstrator for the SE was developed using Virtual Battlespace 2™ (VBS2). A final SE build was developed by using Unity®, provided by an industry partner. This was further modified via iterative design and acceptance by Dstl Information Management Department (IMD) in collaboration with Dstl Policy and Capabilities Studies Department (PCS) to meet the requirements of the experiment. The SE was presented on a laptop with simulated train station announcements etc played through headphones. The SE also had a comprehensive After Action Review capability.

2. Methods

A mixed design was used; a between groups design was applied to the security intervention and a within groups design was applied so that all participants conducted activities across three adversary activities; HR, PBIED, and LD. The order of the adversary activities was counterbalanced for each level of security intervention to remove any order effects.

A number of strategies were employed to immerse participants in the Synthetic Environment by enhancing their motivation. Other strategies were intended to emulate the social psychological pressures experienced by terrorists whilst specifically avoiding the adoption of psychological wrapping.

3. Results and Conclusions

The findings from this preliminary study show that the method used has been successful in identifying significant differences in the deterrent impact of security interventions and scenarios. This method of experimentation shows much promise for future studies. The experimental design needs refinement to ensure the social and psychological pressures have the desired impact.

4. References

Available on request.

Author Biographies

ANDREW GREEN is an Ergonomist at the Defence Science and Technology Laboratory, Portsdown West, UK.

DR BEEJAL MISTRY is a Psychologist at the Defence Science and Technology Laboratory, Portsdown West, UK.

Reasoning, Planning, and Goal Seeking: A Cognitive Architecture for Small Combat Unit Constructive Simulation

Peter Amstutz
Mitha Andra
Daniel Rice
Technology Solutions Experts, Inc.
209 West Central Street, Suite 300
Natick, MA 01760
508-655-2232
peter.amstutz@tseboston.com, mitha.andra@tseboston.com, daniel.rice@tseboston.com

Keywords:
reasoning, planning, goal selection, autonomous behavior, cognitive architectures, constructive simulation

1. Introduction

The current state of military Modeling and Simulation scenario creation is difficult, requiring too much time and effort on the part of Subject Matter Experts (SMEs) and analysts to produce scenarios that are sufficiently realistic for valid analysis (James, 2008). Constructive simulations such as the Infantry Warrior Simulation (IWARS) provide a user interface that allows SMEs and analysts to build missions based on a graphical scripting language paradigm. While relatively accessible to non-technical users, constructing large, realistic simulation scenarios through scripting is painstakingly slow, due to the burden on the analyst to plan out the behaviors to respond to every possible result of every decision point.

In this poster session, Technology Solutions Experts, Inc. (TSE) will present research into new approaches to behavior modeling for Small Combat Unit (SCU) constructive simulation.

2. Reasoning, Planning and Goal Seeking

TSE developed the Reasoning, Planning, and Goal Seeking (RPGS) framework as a high-level cognitive model for agents. RPGS establishes a flexible approach to modeling automated SCU and Soldier-agent decision processes in constructive simulations and to building operationally relevant decision support tools.

"Reasoning" is the act of adding new facts to the agent's knowledge base and using these facts to select the agent's goal. This includes perception of the environment, spatial reasoning, and the application of knowledge rules to determine new facts.

"Planning" is the process of finding a sequence of actions that will achieve a goal. By understanding and anticipating the outcomes of specific actions, and acting with intent, the agent can engage in problem solving, which is impossible with scripted behaviors.

"Goal Seeking" links the reasoning and planning processes to actual behavior, translating planning operators into actions in the underlying simulation.

The approach to agent decision and task automation with RPGS is based on the idea that Soldier agents can be given a goal, the means to work toward that goal, and the conditions under which the goal is fulfilled. The Soldier agent then can determine how to best to accomplish the goal on-the-fly. Dynamic factors of the scenario feed back into the agent's behavior to determine steps to take and provide Situational Awareness (SA) for decisions in order to fulfill the goal.

Open-ended action selection enables more realistic human behavior as measured by an agent's ability to react or work around obstacles without requiring explicit, pre-planned guidance from the analyst. Agent behavior can also be constrained by physical stressors or incomplete information, resulting in different action plans than would be produced with perfect knowledge or perfect capability. This enables SMEs and analysts to study the secondary effects of bounded decision making on the outcome of the scenario.

The simulation agent runs the RPGS process continually during the simulation, monitoring changes in situation and adjusting the goal or plan in response to new information. RPGS supports hierarchical planning, for example, delegating path planning to specialized route selection algorithms.

3. Implementation

TSE implemented a proof of concept of the RPGS framework as a behavior engine component for IWARS. The modular, plug-in architecture of IWARS meant that the experimental autonomous behavior engine required no changes to the IWARS core and could be used alongside the conventional script-based behavior engine.

TSE chose to use first-order symbolic logic as the basis for abstract reasoning, and selected SWI-Prolog (Wielemaker et al., 2011) for the inference engine. Agent perception and knowledge is translated from IWARS data structures to Prolog clauses. Inference rules that leverage agent knowledge into higher-level knowledge abstractions are expressed directly in Prolog. Agent goals are represented as rules that are unified with agent knowledge to determine if the goal rule should apply, and a goal state which is used as input to the planner. Unification allows generic rules (fire if a target exists) to generate specific goals (fire at target X). Goals are prioritized and the highest priority goal is selected. A simplified example is shown in Figure 1.

For planning, TSE used a planner based on the Stanford Research Institute Problem Solver (STRIPS) model (Russel & Norvig, 2003) with each operator consisting of a set of preconditions (conditions that must be true in order to apply the operator), a set of anticonditions (conditions that cannot be true), an add list (clauses to add to the current state), and a remove list (clauses to remove). For the proof of concept, TSE used a simple forward-chaining planning algorithm which returned the first plan found by the search. TSE's implementation permitted arbitrary Prolog clauses in conditions enabling unification between conditions and state changes. To reduce the planning search space, goals specify which operators are available to achieve the goal.

```
plannergoal(1, killEnemy, [E]) :-
      enemy(E, true).

opn(shoot,
    [enemy(E, true), alive(E, true)]
    [alive(E, true)],
    [alive(E, false)])

Initial state: [alive(red, true)]
Goal state: [alive(red, false)]
Plan: [shoot(red)]
```

Figure 1: Example Prolog clauses describing the goal, operator, starting state, goal state, and generated plan to achieve the goal state.

Once the plan is generated, it is translated into primitive agent actions, which are called "skills" in IWARS. TSE used the same skills as they are presented in the conventional IWARS scripting engine with no modifications. Each skill is executed in sequence, and the situation is periodically re-evaluated. When all skills are complete, a skill fails, or a higher-priority goal becomes active, then a new goal is selected, a new plan generated, and the process repeats.

4. Results

TSE was successful in using the RPGS behavior engine to control an IWARS agent. The agent was given goals comprising a "seek and destroy" mission to move among a set of predefined waypoints, taking cover from and engaging enemy agents that were discovered. The RPGS agent was able to maneuver in the environment, act intentionally to acquire needed information, react to danger, engage the enemy, and select appropriate tactics given the situation. The agent required no special prior knowledge of the scenario; all actions were taken in response to knowledge acquired during the scenario.

The RPGS agent demonstrated dynamic behavior and the ability to act proactively in ways that would be difficult, tedious, or impossible to accomplish using conventional mission scripting or state machines. TSE concludes that the RPGS framework is a promising approach to developing autonomous agent behavior for constructive military simulation and will be conducting ongoing research into ways to make this capability accessible and useful to non-technical analysts and SMEs who are using constructive simulation for study.

5. References

James, D. R. (2008). *Exploring the Potential Value of OneSAF at the Small-Unit Level.* U.S. Army Research Institute for the Behavioral and Social Sciences.

Russel, S., & Norvig, P. (2003). *Artificial Intellegence: A Modern Approach 2nd ed.* Upper Saddle River, NJ: Pearson Education.

Wielemaker, Schrijvers, Triska, Lager. (2011). SWI-Prolog. Theory and Practice of Logic Programming, arXiv:1011.5332v1

Author Biography

PETER AMSTUTZ is a senior computer scientist at Technology Solutions Experts, Inc. (TSE) in Natick, MA. He is one of the lead developers of the Infantry Warrior Simulation (IWARS) and holds a M.S. in computer science from the University of Massachusetts, Amherst.

Towards a Speech Capable Intelligent Semi-Automated Force

Joseph F. Reni
Jennifer Pagan
Melissa M. Walwanis
Naval Air Warfare Center Training Systems Division (NAWCTSD)
12350 Research Parkway
Orlando, FL 32826
joseph.reni@navy.mil, jennifer.pagan1@navy.mil, melissa.walwanis@navy.mil

Keywords:
Semi-automated force, role-player, human behavioral model, intelligent agents, computer generated force

1. Introduction

Live, Virtual, and Constructive (LVC) training environments provide opportunities for generating operational readiness at a reduced cost through immersive, mission-relevant exercises. Delivering a realistic environment requires a large number of friendly and adversarial entities (human and simulated). Currently, this is achieved using human role-players and Semi-Automated Forces (SAFs). The use of human role-players can be challenging due to the difficulty of obtaining personnel with the correct expertise, increased costs of hiring outside contractors, training time foregone by volunteer role-players, and additional operational equipment (Abbott, Basilico, Glickman, & Whetzel, 2010). SAFs are pre-scripted simulated forces, controlled and monitored by a human, that can provide a more cost efficient alternative because a single operator can monitor multiple SAFs. However, the pre-scripted nature of SAF behaviors presents some inherent challenges. This paper outlines these challenges and proposes a framwork for developing a Speech-Capable Intelligent Semi-Automated Force (SCI SAF) to overcome the presented challenges.

2. Current SAF Technology Challenges

SAFs use finite state machines to generate linear behaviors from rules defined by domain knowledge (i.e., subject matter experts and manuals), which results in a set of independent behaviors programmed one after another to provide a sense of continuity. This lack of dynamic behavior often causes SAFs to appear as nothing more than target drones that provide limited training utility (Abbott et al., 2010) as trainees learn to anticipate SAF behaviors. In order to facilitate more realistic combat behaviors, each mission variation must be pre-scripted (Cox & Fu, 2005). This burdens instructional personnel with scripting multiple scenarios for a large number of entities.

Another challenge with using traditional SAFs is their inability to individualize behaviors. Because SAFs typically do not use human performance data to model human variability, they are often seen as being machine-like (e.g., flying exact routes or trajectories). This may become problematic in LVC environments if trainees can easily distinguish between live and constructive entities. These challenges limit simulation fidelity and increase costs, thus research efforts have focused on developing more realistic SAFs that incorporate environmental and human performance data to adapt behaviors in real-time.

3. Proposed Solution

Figure 1 illustrates a framework for addressing the shortcomings of traditional SAFs that is grounded in existing SAF technologies, Intelligent Human Behavioral Models (IHBM), Intuitive Decision Making (IDM), and a Natural Language capability. In the SCI SAF framework, instructional personnel interact with an IHBM designed to replicate human behavior by using IDM to search for persistent and dynamic environmental factors to adapt the SCI SAF actions. Once the appropriate actions are selected, IDM tasks the SAF with the appropriate behaviors. To increase realism, trainees communicate with the SCI SAF via Natural Language technology. By providing an integrated solution, the SCI SAF provides a framework for a dynamic capability with individualized behaviors to increase realism.

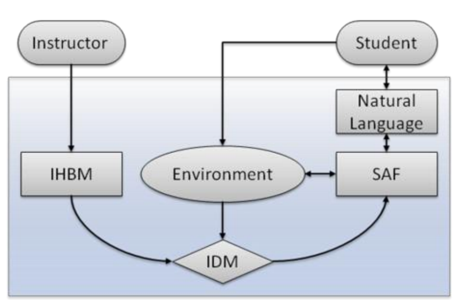


Figure 1. Speech Capable Intelligent Semi-Automated Force (SCI SAF) Model

Intelligent Human Behavioral Model. The IHBM uses computer-based models (e.g., cognitive architectures, machine learning algorithms) to replicate human behaviors (Giordano, Reynolds, & Brogan, 2004). This increases the tactical realism of entities by providing more robust behavior variations and allowing SCI SAFs to adapt their behavior in real-time based on environmental inputs. These dynamic behaviors add an element of unpredictability to training scenarios that traditional SAF models do not (Cox & Fu, 2005). While IHBMs help enhance the realism of traditional SAFs, in order to be indistinguishable from a human, they must be capable of responding to the same environmental cues a human operator would, which requires the ability to incorporate the uncertainty of human decision making. Therefore, the SCI SAF IHBM uses IDM to incorporate environmental and performance factors to adapt the way SAFs carry out behaviors.

Intuitive Decision Making. IDM requires making accurate decisions using multiple vague cues (e.g., imprecision, uncertainty, and partial truths). IDM enables SCI SAFs to accept such imperfect environmental data using soft computing methods (e.g., genetic programming, neural networks, Bayesian networks) within a situational awareness model (i.e., perception, comprehension, and projection; Endsley, 1988). The IDM also integrates performance mediating factors (e.g., cognitive, affective) to create individual differences between SAFs.

Natural Language Capability. Natural language processing provides the highest fidelity communication in a virtual environment and reduces the need to learn a new system interface **(**Guinn & Montoya, 2009). In order to fully support Natural Language (e.g., anticipating, receiving, interpreting, and executing), the SCI SAF requires three Natural Language components: 1) Dialogue System for information management, 2) Speech Recognition System for breaking audio into machine readable structures, and 3) Speech Synthesis System (i.e., text-to-speech) for converting text-based commands to computer generated speech (Allen et al., 2001).

4. Conclusion

While LVC training provides an alternative and robust training environment that increases fidelity and decreases total ownership costs, to meet its full potential there is a need for simulated forces that demonstrate human-like responses (e.g., behavior, communication). To this end, the SCI SAF framework proposes the integration of advanced technologies capable of dynamic behaviors, increased awareness, and speech-based communication to provide more effective training. While the maturity of component technologies vary, current capabilities provide vast strides in the movement toward realistic simulated forces. Therefore, through consideration of an integrated SCI SAF approach, future technology will deliver capabilities that reduce manpower requirements, workload, and training costs.

Authors' Note. The views expressed herein are those of the authors and do not necessarily reflect the official position of the organizations with which they are affiliated.

5. References

Abbott, R. G., Basilico, J. D., Glickman, M. R., & Whetzel, J. (2010). Trainable automated forces. *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, 2132-2142.

Allen, J., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L., & Stent, A. (2001). Toward conversational human-computer interaction. *AI magazine*, *22*(4), 27.

Cox, C., & Fu, D. (2005). AI for automated combatants in a training application. *Proceedings of the Second Australasian Conference on Interactive Entertainment*, Sydney, Australia.

Endsley, M. R. (1988). Design and evaluation for situation awareness enhancement. *Proceedings of the Human Factors Society*, Santa Monica, CA.

Giordano, J. C., Reynolds, P. F., & Brogan, D. C. (2004). Exploring the constraints of human behavior representation. *Proceedings of the 2004 Winter Simulation Conference*, Washington, D.C.

Guinn, C. I., & Montoya, R. J. (2009). Natural language processing in virtual reality training environments. *Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, Orlando, FL.

Authors' Biographies

JOSEPH F. RENI is a Research Psychologist at NAWCTSD. His research interests include Human-Computer Interaction (HCI), artificial intelligence, and computational perception. He is currently pursuing a M.S. in HCI from Iowa State University.

JENNIFER PAGAN is a Research Psychologist at NAWCTSD, Orlando FL. Her research interests include performance assessment and measurement, personnel selection, test validation, and training effectiveness evaluation. She has a M.S. degree in Industrial/Organizational (I/O) Psychology from the University of Central Florida.

MELISSA M. WALWANIS is a Senior Research Psychologist at NAWCTSD. Her research interests include distributed mission training, simulator instructional tools, and LVC. She has a M.S. degree in I/O Psychology from the University of Central Florida.

Using Micro-Architectures of Cognition to Model Macro-Cognitive Systems: A Warfighting Video Game Example

Sylvain Pronovost
CAE Professional Services
1135 Innovation Drive
Ottawa, Ontario, K2K 3G7
+1 (514) 963-2035
s.pronovost@cae.com

Dr. Robert L. West
Institute of Cognitive Science
1125 Colonel By Drive
Ottawa, Ontario, K1S 5B6
+1 (613) 520-2600 x. 2258
robert_west@carleton.ca

Keywords:
ACT-R, cognitive architecture, cognitive modeling, GOMS, hierarchical command structures, macro-cognition, micro-cognition, SGOMS, sociotechnical systems, tactical planning, World of Warcraft

1. Overview

The use of micro-cognitive architectures to model macro-cognitive systems is motivated by the open ended question as to whether the theories of cognitive science pitched at the level of individual mental processes can be successfully linked to, or bridged with, macro-cognitive phenomena involving multi-agent activity in complex sociotechnical systems. Previous research attempts to use the GOMS human information processor model in sociotechnical systems (West & Nagy, 2007 for references) have shown that GOMS is ill suited to model macro-cognition. SGOMS, or sociotechnical GOMS (West & Pronovost, 2009, West & Somers, 2011 for recent developments) was created as an extension to GOMS in order to cope with such shortcomings, mainly in dealing with interruptions and task switching, two critical issues in sociotechnical systems modeling.

The question motivating the present research is whether SGOMS can implement complex, multi-agent, strategic (cooperative and competitive) activities as-is, or requires theoretical and methodological extensions to cope with such interactions. More precisely, what are the implications of cognitive processing constraints on hierarchical command structures? Should the design of command structures consider cognitive constraints, and are there ways of making command structures better? We present an implementation of SGOMS in the ACT-R cognitive architecture, extending SGOMS to strategic (in this context, tactical) activities situated in a virtual environment. SGOMS models were developed, based on task analyses of the massively multi-player online roleplaying video game *World of Warcraft* tm, focusing on hierarchical command structures, using two varieties of tactical planning processes, the MDMP (Military Decision-Making Process) and the RPM (Recognition Planning Model).

The results will be used to evaluate whether either or both SGOMS and ACT-R require amendments in order to model macro-cognitive processes adequately. Firstly, ACT-R may require additional 'goal' buffers to maintain a plausible view of higher level cognitive and behavioral activity, in order to manage the complexities of planning and execution in the strategic activities extracted from the SGOMS model, in a 'top-down' fashion. Secondly, the addition of a parallel external monitoring capability, in the form of an additional production system, is being tested to manage 'bottom-up' interruptions, which are otherwise impossible to handle from within ACT-R. These amendments to SGOMS and ACT-R are likely to improve the degree to which the candidate model predicts the proper courses of actions, accounting for interruptions, task shedding, and handling the planning and execution phases of tactical planning appropriately.

2. The Cognitive Architecture

An overview of the extended ACT-R micro-architecture is presented on the next page (fig. 1). It consists of additional goal buffers for the four levels of SGOMS constructs (planning unit, unit tasks, methods, and operators), two imaginal buffers to represent the problem state of the agent (internal and external states), as well as a simple parallel production system (and buffer) representing the amygdala, to account for external disruptions to the primary production system.

3. The SGOMS Model

The tactical planning model framed in SGOMS is implemented in ACT-R via production rules and declarative memory elements representing the levels of SGOMS constructs. Examples of planning units are the various stages of the RPM (sit rep, COA development, COA analysis, execution), and the COAs themselves (engage enemy at defended location, defend own base, scout location, etc.). Unit tasks can be 'communicate through VoIP', 'move to area', 'engage enemy', etc. Finally, the GOMS methods and operators are low level sequences of (or unique) short and precise tasks such as 'target enemy', 'keystroke sequence', 'click location', 'view interface, 'listen to VoIP', etc. (fig. 2)

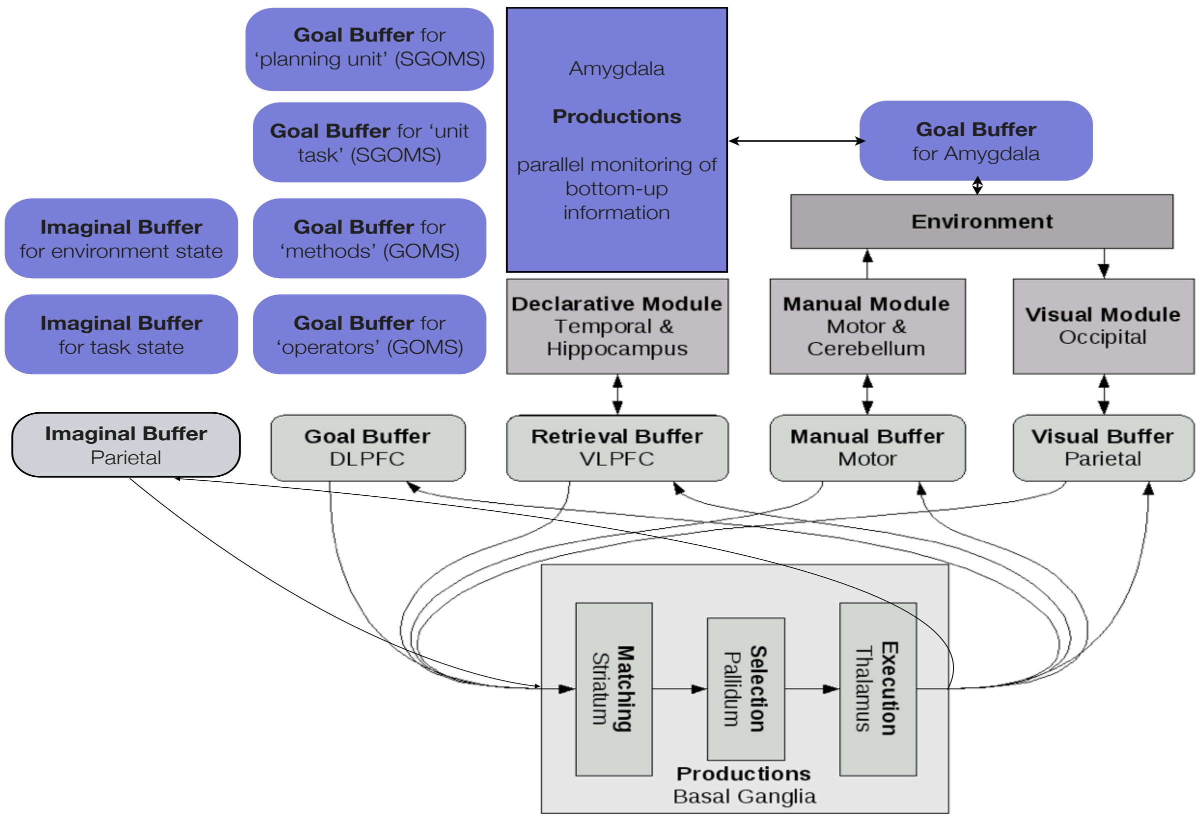


Figure 1 – The extended ACT-R architecture. Original image (in grey) from Stewart, T. C., Tripp, B., and Eliasmith, C. (2009). Python scripting in the Nengo simulator. *Frontiers in Neuroinformatics, 3* (7).

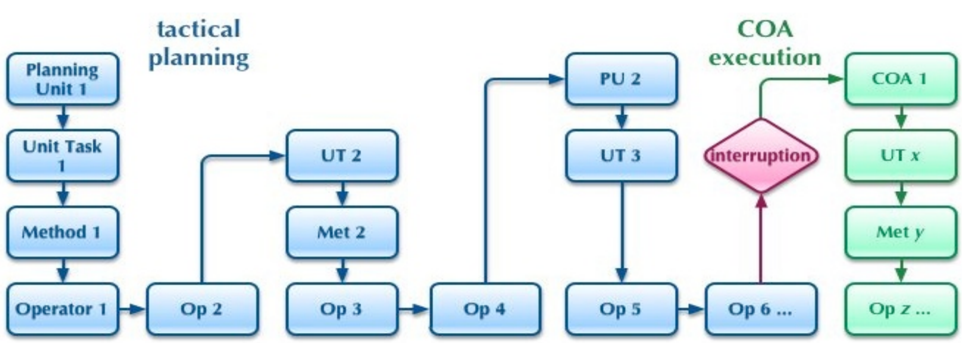


Figure 2 – An overview of the flowchart for the World of Warcraft tactical planning SGOMS model in ACT-R.

4. References

West, R. L., Nagy, G. (2007) Using GOMS for Modeling Routine Tasks Within Complex Sociotechnical Systems: Connecting Macro Models to Microcognition. *JoCEDM,* 1 (2): 186-211.

West, R. L., Pronovost, S. (2009). Modeling SGOMS in ACT-R: Linking Macro- and Microcognition. *JoCEDM, 3* (2): 194-207.

West, R. L., Somers, S. (2011). Scaling up from Micro Cognition to Macro Cognition: Using SGOMS to build Macro Cognitive Models of Sociotechnical Work in ACT-R. *Proceedings of the 33rd Annual Conference of the Cognitive Science Society.*

5. Author Biographies

SYLVAIN PRONOVOST is a Human Systems Integration consultant at CAE Professional Services, as well as a PhD candidate in cognitive science at the Institute of Cognitive Science of Carleton University. He specializes in cognitive and behavioral modeling and simulation in dynamic synthetic environments.

ROBERT L. WEST, PhD is an associate professor at the Institute of Cognitive Science and the Department of Psychology of Carleton University. His main area of research is the application of cognitive modeling to interactive human tasks, including game playing and sociotechnical systems.

Adjust Distance for Facilitating Better Creation through Memory Lost Control

Yu.Xiong
Yukio.Ohsawa
Graduate School of Engineering
The University of Tokyo
7-3-1Bunkyo-ku, Tokyo, 113-8656 Japan
yu.xiongyu@gmail.com, ohsawa@sys.t.u-tokyo.ac.jp

Keywords: iconic memory, backtracking, working memory, eye movement pattern, KeyGraph, creation, cognition

Abstract: *During creation, we believe a more creation- friendly environment can facilitate creation work be done more smoothly and efficiently. Furthermore, we proposed a method to decide the appropriate distance of objects in creation environment and the distance between viewer and the objects. We found the distance should be kept within certain range to achieve high productive creation work. As a matter of fact, to facilitate the backtracking much more efficient, we raise the hypothesis that the distance between objects should be controlled in one's memory holding time and retrievable time.*

1. Introduction

Vision attention affects vision performance of our daily lives (M. Lotze et al., 2000). Through attention control, we can achieve much more efficient cognition, memory and creation.

There are amounts of information stored in iconic memory of which the capacity is said to be infinite (Barbara Sakitt, 1976), but just a few received attentions and finally be utilized, and the rest decay soon. It is believed that all the objects entering in our visual field goes into iconic memory (Neisser, U., 1967), and then our brain goes through the very efficient and quick information selection process(Hawkins, J.&Blakeslee, S., 2004). Information of those objects selected to receive attention go to visual working memory, and the rest remain in iconic memory and decay with the passage of time (Shiffrin, R. M.&Atkinson, R. C., 1969). Half our visual brain power is directed to processing less than 5 percent of the visual world allegedly. This is the reason why we have to move our eyes. We have to admit that it is handy of our brain to grasp and extract the major information we need at that moment, but it also makes us miss the chance of accessing other information available at the same time which decays soon from our memory (Colin Ware. 2008). We know complex cognitive work, especially creation work, usually needs several times of grouping and rehearsal (Zelinsky, G.J., & Loschky, L.C., 2009), so our research therefore focuses on reducing the part of iconic memory loss and enhances the utilization of iconic memory in order to achieve better creation job.

2. Adjust Distance for Facilitating Better Creation

During creation work, we often need to use many kinds of information simultaneously. For example, when a company develops a new business plan, one common way is to add new functions to existing products. Thus, creation work usually involves two (or more) items into thinking process at the same time. KeyGraph is one of the tools which visualizing relations among items [4], where nodes represent items/words in data and the links and distances between nodes visualize correlations of items, computed according to how frequently they closely appear in the data. Viewers can accomplish creation work through looking at KeyGraph and then trying various grouping and combination.

In order to increase the potential work to use multiple items, time of backtracking for retrieving information of previous item and time of jumping to another item should be within iconic memory time span which is 500ms (Barbara Sakitt, 1976) to reduce the chance of memory loss during eye movement. Thus during backtracking time span, the iconic memory loss can be reduced, and then visual information input for complex cognition work can increase consequently.

3. Hypothesis

Limited time span of iconic memory (Barbara Sakitt, 1976) leads to backtracking strategies which is to reinstate information into working memory. Yet, due to the memory loss, some tasks are hard to accomplish. The creation task, which requires many times of grouping and rehearsal, particularly need to cut down the memory loss otherwise the rehearsal can be hard to accomplish.

In order to reduce the memory loss, we hypothesize the distance between items in creation environment should be kept within a certain distance to reduce memory loss due to time spend in eye movement.

To be specific, items next to each other should be kept within iconic memory retrievable distance. According to Shiffrin, R. M.&Atkinson, R. C.(1969)'s memory model, if one object has been glanced once by an informant but without being paid an explicit attention, the visual information of this object will decay very soon and completely get lost after 0.5 second then the object will lose its chance to be used. There are two ways to increase its chance to be utilized: first, turn its information into working memory, which is to pay attention to the items. Some works have been done by other researchers about how to make certain item stands out among a large number of items (Colin Ware, 2008). However, this method rather suits the situation in which we already know which item should be emphasized. The second way and also what we are trying to do is to facilitate the information processing which utilize both former and current information, and ensure this work be done before any memory gets lost.

In most circumstances, inspiration is triggered by new information combined with old information in our memory. According to Todd M. Thrash and Andrew J. Elliot (2003), inspiration involves previous information evoking new thought and leading to transcendence of the ordinary preoccupations or limitations of human agency. When people lost the previous information, inspiration will hardly happen. Then in order to use both items, backtracking to retrieve their information is needed. However, retrieval and recognition cost amounts of time and most of them are considered meaningless to be repeated. As a result, this kind of repeating should be avoided. In order to increase the potential work to use both items (previous and present) and reduce the time of backtracking for retrieving, the retrievable distance should be decided by the speed of attention shift and time of jumping to another item, which is within iconic memory time span 500ms.

As we see things and shift our attention through eye movement, so the distance between items (S) depends on the distance between viewers' eyes and item (H) (See Figure 1).

S= 2*X= 2*H* tan (r/2)*t

H stands for the distance between eye and item; X is half of the distance between items. r is the velocity of eyes rolling and t is the time span during which information can still be hold in memory. In different circumstances, the distance H is fixed respectively and easy for us to measure in our experiment.

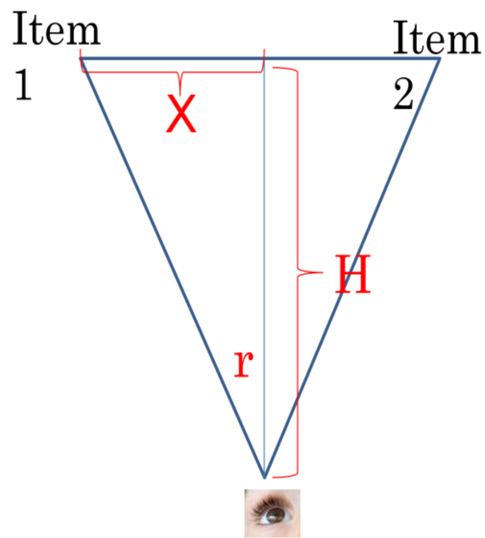


Figure 1. Distance between items and distance between viewer and items.

Before beginning calculation of distance between items, we have to know the mechanism of our visual system. It is well known that we see the world through our eyes, and when we shift our attention, we roll our eye balls. There is several different velocity of rolling the eye balls according to different purposes of attention shifts.

Saccade (60-600 deg/sec): Viewers intentionally shift the attention lead to saccade, the speed of which is comparatively very quick.

Pursuit: (5-30 deg/sec): Middle-velocity movement occurring in looking at a moving object.

Slow Saccade Line (10-40 deg/sec): When viewers acquire new interpretations of an image in his/her long term memory, the velocity will be slower than saccade but faster than pursuit. (Ohsawa Y.& Maeda Y. 2009)

By analyzing data we collected from other preliminary experiment we got the average velocity of shifting attention between items during regular scanning to be around 20 degree/second, together with the knowledge about which we already known of velocity of eye movements, that the eyes move at the velocity between saccade velocity and pursuit velocity when scanning over whole picture. Eyes draw lines in the velocity at

Slow Saccade Line when viewer reaches an insight (Ohsawa, Y.& Maeda, Y., 2009), of which the velocity is 10 deg/sec to 40 deg/se and it leads to backtracking to the certain item which triggered the insight. Therefore, we take 20 deg/sec of Slow Saccade Line's velocity (higher that pursuit velocity and lower than saccade velocity) as the velocity of backtracking to r in this formula.

Furthermore, when viewer sweeps objects in the environment, as long as these objects have not attracted view's attention, these objects goes into viewer's iconic memory, which only last 0.5 second, indicates the longest time span of backtracking movement for information holding and retrieving. In addition, it is proved the reaction time to change attention is around 100 mini second (C.Ware, 2004), so the iconic memory holding time is left to 0.4 second (0.5-0.1). Since the backtracking involves going back and forth, so the time should divided by 2, whereas here we assign 0.2 (0.4/2) to t. As a result, the formula becomes as follow:

S= 2*X= 2*H* tan (r/2)*t
=2* H*tan(20/2)*0.2=0.07*H

We raise the hypothesis that when people in an environment for creation, either the distance between items or the distance between viewer and items are fixed, we can adjust another variable to get the best distance for creation.

In our next experiment, the objects are the items on the KeyGraph, and the longest distance from the graph on the screen to viewer is 60 centimeter away due to the limitation of our eye tracking equipment. Therefore H is 60 centimeters, so the distance between items should be kept within 4.2 centimeters by doing the simple math.

4. Experiments for Verification

4.1 Comparison Experiment

4.1.1 Experimental Objective:

To set up a standard line of performance of idea creation, we did the comparison experiments to seven informants separately.

4.1.2 Experiment Approach

1. Visualize the raw data in KeyGraph

KeyGraph is a tool for visualizing relations among events (Kenichi H. et al., 2007). Meanwhile, the raw data is from ordered dishes of a restaurant in one month. The data includes weather, date, each dish's name and its number being ordered by men and women.

Table 1. Partial Data of Raw Data

温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	男 (male)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	男 (male)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	女 (female)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	女 (female)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	女 (female)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	豚カツ (fried pork)	女 (female)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	唐揚げ (fried chicken)	男 (male)
温 (warm)	晴れ (sunny)	水曜日 (Wednesday)	唐揚げ (fried chicken)	男 (male)

By visualizing above raw data (See Table 1) through KegGraph, we got the visualization map to illustrate the correlation of the dishes being ordered by male and female customers in various circumstances (See Figure 2). Most of the distance between items is around 7 centimeters.

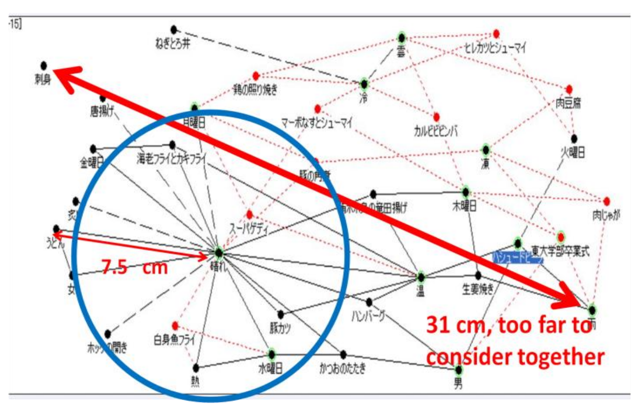


Figure 2. Visualization of data by KeyGraph in Comparison Experiment

2. Let informants think out new dishes or business for this restaurant by looking at the Key-Graph.

The time being given to each informant for watching and thinking is equivalently 60 seconds.

3. Finish the questionnaire

Table 2 Questionnaire

Questionnaire

Gender: (Male) (Female)

Age: 10~20 20~30 30~40 40~50 above 50

How many ideas did you think out？

What is/are your idea/ideas? How did you think out the ideas?

After the picture viewing, informants are required to fill the questionnaire for idea output (see Table 2), which consists of their age, gender and ideas about new dishes or businesses.

4.2 Verification Experiment

4.2.1 Experimental Objective:

In order to verify whether our hypothesis about the distance between items in creation environment is correct, a verification experiment has been carried out.

4.2.2 Experiment Approach

In the verification experiment, the informants were asked to do the exactly the same thing as in the comparison experiment to think out new dishes or business for this restaurant by looking at the same but smaller Key-Graph.

The distance between each item of the smaller Key-Graph is about 53% of the former one which makes most of the distance between items are around 4 centimeters (see Figure 3).

Given the same time span 60 seconds of picture viewing and idea thinking, the performance is supposed to be better than in the former size.

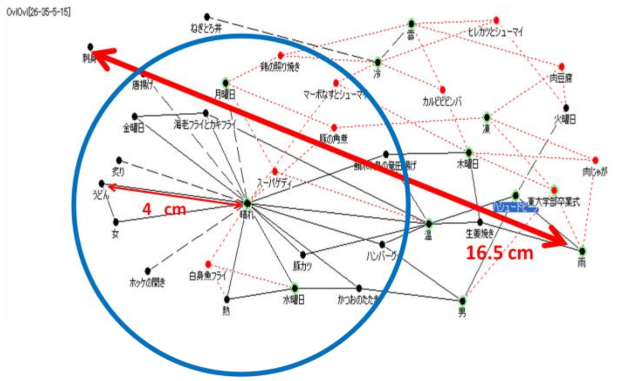


Figure 3. Visualization of data by KeyGraph in Verification Experiment

4.3 Experiment Result

Table 3. Number of Ideas Being Brought Out in Two Experiments

Informant	Number of ideas
1	4
2	1
3	0
4	4
5	3
6	1
7	2
Average	= 2.142857
a	5
b	6
c	6
d	4
e	5
f	4
g	5
Average	=5

This chart shows the number of ideas being brought out by each informant in comparison *(1-7) and verification (a-g) experiment*

Like the comparison experiment, seven informants took part in the verification experiment this time, among which six are university students and one is high school student and the ratio of sexuality is 4:3(man: woman), the same with the comparison experiment. By comparing the number of ideas being raised in two experiments, it is obvious that significant improvement has been realized. Given the same time span, viewers proposed about two 2 ideas in average in the previous experiment whilst in the verification experiment viewers thought out about 5 ideas in average. Since the only changed variable is the distance between items, so this 250% increase of idea productivity should give credit to the memory lost control by setting appropriate distance between items.

5. Conclusion and Future Work

More verification experiment should be done and we are going to evaluate the quality of the ideas being raised in the future for the sake of much fairer comparison. At the same, another comparison experiment should be done in case the items is too close to each other.

Our distance calculation is mainly based on iconic memory span aims to reduce iconic memory loss during backtracking memory since iconic memory is the start information input and processing (A.O.DICK, 1974). It is proved to be effective by keeping distance of items within such distance. However, due to working memory also has span and limited capacity, in order to further reduce memory loss, other conditions related to working memory should be considered as well.

Furthermore, our work is based on static picture and environment, but in real world, information input can be dynamic. Plus, memory span and capacity is different for individuals, dynamical adjustment to graph and environment by observing each individual's performance in real time should be more efficient for creation work. And we believe our research can serve the base of realizing such dynamic operation.

REFERENCES

[1] A. Baddeley. (1992). Working Memory. *Science*, Vol. 255 (5044), 556-559.

[2] A. Baddeley. (1992). Working Memory: The Interface between Memory and cognition. *Journal of Cognitive Neuroscience*, Vol.4 (3), 281-286.

[3] A. O. Dick. (1974). Iconic memory and its relation to perceptual processing and other memory mechanisms. *Perception&Psychophysics.* Vol. 16(3), 575-596.

[4] Barbara Sakitt. (1976). Iconic Memory. *Psychological Review*, Vol. 83(4), 257-276.

[5] Christine Burton & Meredyth Daneman (2007). Compensating for a Limited Working Memory Capacity During Reading: Evidence from Eye Movements. *Reading Psychology*, 28:2, 163-186

[6] Colin Ware. (2004). Information Visualization. San Francisco: Elsevier.

[7] Colin Ware. (2008). Visual Thinking for Design. Burlington: Elsevier.

[8] C. L. Colby&J. R. Duhamel& M. E. Goldberg. (1996). Visual, presaccadic, and cognitive activation of single neurons in monkey lateral intraparietal area. *AJP - JN Physiol*. Vol. 76 (5),2841-2852.x.

[9] Engel RW, Kane M, Tuholski S. (1999). Individual differences in working memory capacity and what they tell us about controlled attention, general fluid intelligence, and functions of the prefrontal cortex. Mechanisms of Active Maintenance and Executive Control, ed. AMiyake, P Shah. NewYork: Cambridge Univ. Press.

[10] F. Woodman&J.Luck. (2004). Visual search is slowed when visuospatial working memory is occupied. *Psychonomic Bulletin &Review*. Vol.11 (2), 269-274.

[11] Hawkins, J.&Blakeslee, S. (2004). On Intelligence. New York: Times Books.

[12] Kenichi H., Yoshiharu M., Yukio O. (2007). Data Crystallization applied for designing new products. *Journal of Systems Science and Systems Engineering,* Vol. 16 (1), 34-49.

[13] M. Lotze, B. Treutwein, T. Roenneberg. (2000). Daily rhythm of vigilance assessed by temporal resolution of the visual system. *Vision Research,* Vol. 40, 3467-3473.

[14] Neisser, U. (2009). Cognitive Psychology. Grolier Multimedia Encyclopedia. Retrieved July 17, 2009, from Grolier Online http://gme.grolier.com.ccnyproxy1.libr.ccny.cuny.edu/cgi-bin/article?assetid=0066790-0.

[15] Neisser, U. (1967). Cognitive Psychology. New York: Ap-pleton-Century-Crofts.

[16] Nelson Cowan, Emily M, et. (2005). On the capacity of attention: Its estimation and its role in working memory and cognitive aptitudes. *Cognitive Psychology,* Vol. 51(1), 42-100.

[17] Ohsawa, Y.& Maeda, Y. (2009). Eyes Draw Auxiliary Lines in Interpreting Images. *Journal of Computers,* Vol. 4(10), 1012-1021.

[18] Posner, M. I. (1980). Orientation of attention. The VIIth Sir Frederic Bartlett lecture. *Quarterly Journal of Experimental Psychology*,32A, 3–25.

[19] Saccade. (n.d.). In Wikipedia. Retrieved, December 22, 2011,from http://en.wikipedia.org/wiki/Saccade

[20] Shiffrin, R. M.&Atkinson, R. C.(1969). Storage and retrieval processes in long-term memory. *Psychological Review*, Vol. 76(2), 179-193.

[21] Todd M. Thrash and Andrew J. Elliot. (2003). Inspiration as a Psychological Construct. *Journal of Personality and Social Psychology,* Vol. 84(4), 871– 889.

[22] Yukio Ohsawa. (2005). Data Crystallization: A Project Beyond Chance Discovery for Discovering Unobservable Events. *IEEE International Conference on Granular Computing*,

Beijin.

[23] Yuko Ohsawa. (2005). Designing Systems for Chance Discovery. *The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology*, Plenary Lecture.

[24] Yukio Ohsawa. (2005). Data Crystallization: Chance Discovery Extended for Dealing with Unobservable Events. *New Mathematics and Natural Computation*, Vol.1 (No.3), pp.373 – 392.

[25] Zelinsky, G.J. (2008). A theory of eye movements during target acquisition. *Psychological Review*, Vol. 115(4), 787-835.

Author Biographies

Yu Xiong graduated from Shanghai University, China and received a Bachelor degree in Management. Now she is an master course student in Department of Systems Innovation, Graduate school of Engineering in the University of Tokyo. She belongs to the laboratory of Ohsawa&Nishihara, mainly conducts researches about people' cognition process and behavior pattern and her research interests are in chance discovery. She received Best Inventor Award from International Workshop on Innovating Service Systems (ISS2010) Program.

Yukio Ohsawa is a professor in the School of Engineering, The University of Tokyo. He received Ph.D in Communication and Information Engineering from The University of Tokyo. He worked also for School of Engineering Science in Osaka University (research associate, 1995-1999), Graduate School of Business Sciences in University of Tsukuba (associate professor, 1999-2005), and Japan Science and Technology Corporation (JST researcher, 2000-2003). He initiated the research area of Chance Discovery, defined "discovery of events significant for decision making" in 1999, and series of international meetings (conference sessions and workshops), e.g., the fall symposium of the American Association of Artificial Intelligence.

Game Information Dynamics and Its Applications in Congkak and Othello

Hiroyuki Iida, Takeo Nakagawa ,Nguyen Quoc Huy, Siti Mardhiah Hasai, Azizul Nur Husna, Apimuk Muangkasem, Shogo Sone, Taichi Ishitobi
School of Information Science, Japan Advanced Institute of Science and Technology
Nomi, Ishikawa 923-1292, JAPAN
apimuk, iida@jaist.ac.jp

Keywords:
Uncertainty of game outcome, Information dynamic model, Congkak, Othello, Entropy, Entertainment

1 Abstract

This paper is concerned with uncertainty of game outcome in Congkak and Othello. Firstly, information dynamic model on uncertainty of game outcome is derived based on fluid mechanics. Secondly, data analyses on Congkak and Othello have been done. It is found that Congkak is a unique regional game at South-East Asia, while Othello is one of the best games in view of entertainment in the globe. It is suggested that Shannon's entropy provides a measure of uncertainty of game outcome, but not itself. The true uncertainty is given by the present proposed model.

2 Introduction

Fundamental problem of information communication is that of reproducing at one point either exactly or approximately an information selected at another point. Frequently the information has meaning; that is, it refers to or is correlated according to some system with certain physical or conceptual entities. The significant aspect is that actual information is one selected a set of possible information. In the present paper, the selected information is data such as evaluation function scores either in Congkak, or in Othello. Information of game outcome here represents the data which is the uncertainty of game outcome. We consider that information is produced as the motion of particles, for stationary particles provide only trivial information. In this regard, it has been inferred by Solso(1994) that motion of visualized fluid particles, for example, is detected by the eye almost instantaneously through light having enormous high speed, 3×10^8 m/s, and is mapped on the retina. It may be evident that during this process, motion of fluid particles is transformed into that of "information particles" by light carrying the images of fluid particles. The eye and brain may work together in collecting the light reflected from the visualized fluid particles and processing the information particles, which flow in our brain.

Shannon(1948) has introduced quantities of the form

$$\mathrm{H(X)} = -\sum p_i \log p_i,$$

which plays a central role in information theory as a measure of information, choice or uncertainty. The measure H is normally called the entropy of the set of probabilities $p_1, p_2, ..., p_n$. The quantity H has a number of interesting properties which further substantiate it as a reasonable measure of information. For example, (1) H=0 if and only if all the p_i but one are zero, this one having the value of unity. Thus, only when we are certain of the game outcome, does H vanish. Otherwise, H is positive, and (2) for a given n, H is a maximum and equal to log n when all the p_i are equal, i.e., $\frac{1}{n}$. This is also intuitively the most uncertainty situation.

The concept on intelligence transmission velocity has been proposed by Nyquist (1924): The velocity at which intelligence can be transmitted over a telegraph current with a given line speed, i.e., a given rate of sending of signal elements is expressed approximately by the following formula.

$$\mathrm{W} = \mathrm{K} \log \mathrm{m},$$

where W is the intelligence transmission velocity, m the number of current values employed, and K a constant. By the technical term, intelligence transmission velocity is here meant the number of character, representing different letters, figures, etc., which can be transmitted in a given length of time assuming that the circuit transmits a given number of signal elements per unit time. Iida & Nakagawa(2011) has inferred that when information velocity become equal to the speed of light time stops completely. Can we find what happens if the intelligence transmission velocity reaches at the speed of light?

When we speak of the capacity of a system to transmit information, some sort of quantitative measure of information must be specified(Hartley 1928). In the first place, there must be a group of physical symbols, such as words, dots and dashes or the like, which convey certain meanings to the parties communicating. In any given communication, the sender mentally selects a particular symbol and by some bodily motion, as of his vocal mechanism, causes the attention of the receiver. By successive selections, a sequence of symbols is brought to the listeners attention. At each selection, all of other symbols may be eliminated. As the selections proceed, more and more possible symbol sequences are eliminated, and we say that the information becomes more precise. In this study, as the most precise information of game, the evaluation function scores are used(e.g. Tsuruoka et al 2002).

The main purpose of the present paper is two fold:
(1) To derive the information dynamic model on uncertainty of game outcome
(2) To dig out data of Congkak and Othello with aiming at their future improvement regarding entertainment.

3 Modeling

The modeling procedure of information dynamic model on uncertainty of game outcome is summarized as follows :

(a) Assume a flow problem as the information dynamic model and solve it.

(b) Get the solutions, depending on the position and time.

(c) Examine whether any solution of the problem can correspond to game information.

(d) If so, visualize the assumed flow with some means. If not, return to the first step.

(e) Determine the correspondences between the flow solution and game information.

(f) Finally, obtain the mathematical expression of the information dynamic model.

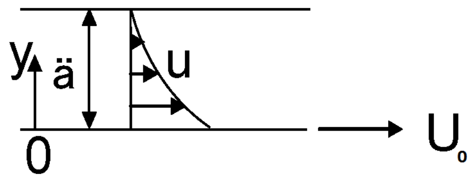


Figure 1: A definition sketch of flow between two parallel flat walls, one of which is at rest, the other is suddenly accelerated from the rest to a constant velocity U_0

The modeling procedure of information dynamics based on fluid mechanics has been established by Iida et al(2011a). Another information dynamics model for a series of approximate solutions of the flow between two parallel flat walls, one of which is at rest, the other is suddenly accelerated from the rest to a constant velocity U_0, Fig.1, will be constructed by following the above procedure step by step.
Flow near a flat plate which is suddenly accelerated from rest and moves in its own plane with a constant velocity is solved by Stokes(1851). For a brief sketch of the solution, see Schlichting (1968).
(a) Let us assume the flow between two parallel flat walls, one of which is at rest, the other is suddenly accelerated from the rest to a constant velocity U_0 . Fig.1. Note that the walls are two-dimensional, horizontal and infinitely long.

Since the system under consideration has no preferred length in the horizontal direction, it is reasonable to suppose that the velocity profile are independent of the horizontal x-direction, which means that the velocity profile u(y) for varying distance x can be made identical by selecting suitable scale factors for u and y. The scale factors for u and y appear quite naturally as the lower wall velocity U_0 and gap between the two walls δ. Hence, the velocity profile after the time t > 0 can be written as the function in the following way.

$$\frac{u}{U_0} = f(\frac{y}{\delta}) \tag{1}$$

(b) Get the solutions. The velocity profile is here accounted for by assuming that the function f depends on $\frac{y}{\delta}$ only, and contains no additional free parameter. Since the fluid particles are fixed on the surface of two walls due to the viscous effect, the function must take the value of 1 on the lower wall(y=0) and the value of 0 on the upper wall(y=δ) . The boundary conditions are:

$$t \leq 0: \frac{u}{U_0} = 0 \text{ for } 0 \leq \frac{y}{\delta} \leq 1$$

$$t > 0: \frac{u}{U_0} = 1 \text{ for } \frac{y}{\delta} = 0; \ \frac{u}{U_0} = 0 \text{ for } \frac{y}{\delta} = 1.$$

When writing down an approximate solution of the present flow, it is necessary to satisfy the above boundary conditions for $\frac{u}{U_0}$. It is evident that the following velocity profiles satisfy all of the boundary conditions.

$$\frac{u}{U_0} = (1 - \frac{y}{\delta})^q, \tag{2}$$

in the range $0 \leq \frac{y}{\delta} \leq 1$, where q is positive real number parameter. Equation (2) is considered as the approximate solutions on the flow between two parallel flat walls, one of which is at rest, the other is suddenly accelerated from the rest to a constant velocity U_0 , where each solution takes an unique value of q. The value of q must be determined by the boundary conditions and the Reynolds number R_e=$U_0 \cdot \frac{\delta}{v}$, where is the kinematic viscosity of the fluid. It is known that the transition from laminar to turbulent flow in the boundary layer is governed by the Reynolds number $R_e = U_\infty \cdot \frac{d}{v}$, where U_∞ is the free stream velocity, d the boundary layer thickness. The critical Reynolds number (Re)crit., at which the transition is initiated, is of 2,800 approximately(e.g. Hansen 1928, Schlichting 1968). In case of the present flow, as shown in Fig.1, at 1 atmospheric pressure and temperature at 20°C, water has the kinematic viscosity $=1.004\times10^{-2}$ cm^2/s. When water is chosen as the fluid, and the constant velocity U_0 = 10 cm/s and the gap between the two walls δ=10 cm are set, we obtain the Reynolds number $R_e \simeq 10^4$. The result of this calculation clearly illustrates how the flow is liable to be turbulent under an ordinary situation. The solution (2) is smooth analytical functions and thus this is only valid for laminar flow.

The fundamental equations for fluid mechanics are the Navier-Stokes equation. This inherently nonlinear set of partial differential equations has no general solution, only several exact solutions, which are trivial in practice, have been found(Wang 1991). All of these exact solutions are for laminar flows, and no turbulent flow solution is available yet. However, it is considered that each of the laminar solutions in (2) represents an approximate turbulent solution. In this regard, we consider that the solutions (2) are applicable for laminar flow as well as turbulent flow to some extent.However, it should be noted that the applicability of the present solutions to turbulent flow is severely limited.

(c) Let us examine whether this solution is game information or not. The non-dimensional velocity $\frac{u}{U_0}$ varies from 1 to 0 with increasing non-dimensional distance $\frac{y}{\delta}$ in many ways with changing the parameter q. It can be considered that $\frac{u}{U_0}$ represents the uncertainty of game outcome. This is why uncertainty of game outcome takes the value of 1 at start, and it decreases with increasing the game length and becomes the value of 0 at the end of game.

Table 1: Correspondences between flow and game information

Physical world(flow)	Informatical world(game)
u: flow velocity	I: current uncertainty- -of game outcome
U_0: plate velocity	I_0: initial uncertainty- -of game outcome
y: vertical distance	L: current game length
δ: gap between two walls	L_0: total game length

(d) Visualize the assumed flow with some means. Imagine that the assumed flow is visualized with neutral buoyant particles. Motion of the visualized particles is detected by the eye almost instantaneously through light and is mapped on our retina(Solso 1994), so that during these processes, motion of the fluid particles is transformed into that of the information particles by light carrying the images of fluid particles. This is why motion of the fluid particles is intact in the physical space, but only the reflected lights, or electromagnetic waves consisting of photons can reach the retina. Photons are then converted to electrochemical particles and are passed along the visual cortex for further processing in parts of the cerebral cortex(Solso 1994). Photons and /or electrochemical particles are considered to be information particles. It is, therefore, natural to expect that the flow in the physical world is faithfully transformed to that in the information world, or brain including eye, which is referred to informatical world here after. During this transformation, the flow solution in the physical world changes into the information in the informatical world.

(e) Proposed are correspondences between the flow and game information, which are listed in Table 1.

(f) Obtain the mathematical expression of the information dynamic model. Considering the correspondences in Table 1 and (2) can be rewritten as

$$\frac{I}{I_0} = \left(1 - \frac{L}{L_0}\right)^q \tag{3}$$

Introducing the following non-dimensional variables in (3),

$$\xi = \frac{I}{I_0} \text{ and } \eta = \frac{L}{L_0},$$

we finally obtain the mathematical expression of the uncertainty of game outcomeξas

$$\xi = (1 - \eta)^q for 0 \leq \eta \leq 1, \tag{4}$$

where η is the non-dimensional current game length, and q the positive real number parameter. We expect that the greater the value of q is, the greater the strength difference between the two teams (or players) in a game is, and vice versa.

4 Verification of Model

4.1 Congkak

History of Congkak: Congkak(Culin 1804, Hellier 1907, Overbeck 1915) is short for Main Congkak, which is Indonesian for cowrie shell, but some people believe that actually the name of the game originated from the word congak, which in old Malay language means mental calculation without writing it down. Congkak is a popular mancala game in Malaysia, Brunei, Singapore and Indonesia. Many Indonesians believe that the game originated in Malacca Kingdom where it became very popular and spread to the South-East Asia region. This spread was due to the many travelers who visited the kingdom because it was a trading city. In the early days, Congkak was mostly played by the royal family and palace residents, however later it spread to the general population of the kingdom and today it is usually played by girls and women. As the Congkak board is often shaped like a boat it is believed that it is based on the legend of a fisherman unable to go to the sea during rainy season who lost his income during this time. To prevent boredom she or he created this game which is similar to her or his boat.

Today many Congkak tournaments are organized for children in Malaysia, e.g. in Kuala Lumpur, Kuala Terengganu, Pekan and Seremban. Several hotels in southern Borneo offer Congkak course to tourists. Since 2004, the Malaysian Embassy and the Malaysian Association in France sponsore each year a Congkak tournament to spread Malaysian culture in Europe. Another tournament is held in Wales during the Cardiff European Games, an annual meeting of Malaysians from all across Europe. In Brunei, Congkak is also played during the night of royal ceremonials such as the Istiadat Malam Berjaga-jaga at the palace or nobilitysresidence Congkak consists of: Congkak uses an oblong game board called papan congkak, which has two rows each one with five to ten playing pits. These pits are called lubang kampong(”village”) or lubang anak(”child”) in Malaysia. Most widespread are boards with 2 × 7 playing pits. In addition, there is at either end a larger hole to store the captured counters. The store is called lubang rumah(house) in Malaysia. Each player owns the store to her or his left. Each of the small pits contains at the beginning of the game as many counters(usually cowrie shells or tamarind seeds called anak-anak buah in Malaysia) as each row counts small pits. How to play Congkak: 2 players sit opposite each other. Each player owns the row of houses directly in front of her or his houses and the storehouse on her or his left.

(1) Players play simultaneously beginning with anyone of their hoses and dropping seeds clockwise into each house until each the player is finished with all the seeds in her or his hand. On her or his round, a seed is placed in a players storehouse but not her or his opponents.

(2) On ending her or his round, the player takes all the seeds of the house that she or he has dropped her or his last seed in and the process is repeated until the last seed is dropped into an empty house.

(3) If the last seed falls in a house that is part of a players village, she or he can pick all the seeds from her or his opponents house that lies opposite it and put them in her or his storehouse.

(4) If it drops in her or his storehouse, she or he can continue the game, picking a house of her or his choice from her or his side.

(5) When the last seed drops in an empty house, she or he is considered mati(dead) and ends her or his turn. Her or his opponent continues until she or he similarly ends her or his turn.

Data analyses: Mardhiah plays Congkak against Husna under the rules mentioned in the above. The non-dimensional advantage $\alpha(\eta)$ is defined as

$$\alpha(\eta)= \frac{[S_M(\eta)-S_H(\eta)]}{S_T} \text{ for } 0 \leq \eta \leq 1,$$

where $S_M(\eta)$ is Mardhiahs current scores, $S_H(\eta)$ Husnas current scores, S_T the total scores for the two players in the game, and η the non-dimensional game length. Sign of the non-dimensional advantage is defined to be positive when Mardhiah gets advantage, while it is negative when Husna advantage. The uncertainty of game outcome ξ is derived by

$$\xi = \begin{cases} 1-|\alpha(\eta)| & \text{for } 0 \leq \eta < 1 \\ 0 & \text{for } \eta = 1 \end{cases}$$

Fig.3 shows how the non-dimensional advantage$\alpha(\eta)$ and uncertainty of game outcome ξ depend on the non-dimensional game length η. Mardhiah leads the game until $\eta \simeq 0.369$, but after this point Husna gets advantage and keeps it until the end. However, uncertainty of game outcome ξ is kept within 0.8 and 1 until very end of the game, so this game is considered to be quite tight one. Furthermore, in this figure the best fit model curve ξ=(1-η)$^{0.15}$ to the experimental data has been plotted concurrently.

4.2 Othello

History of Othello: Othello is a board game involving abstract strategy and played by two players on a

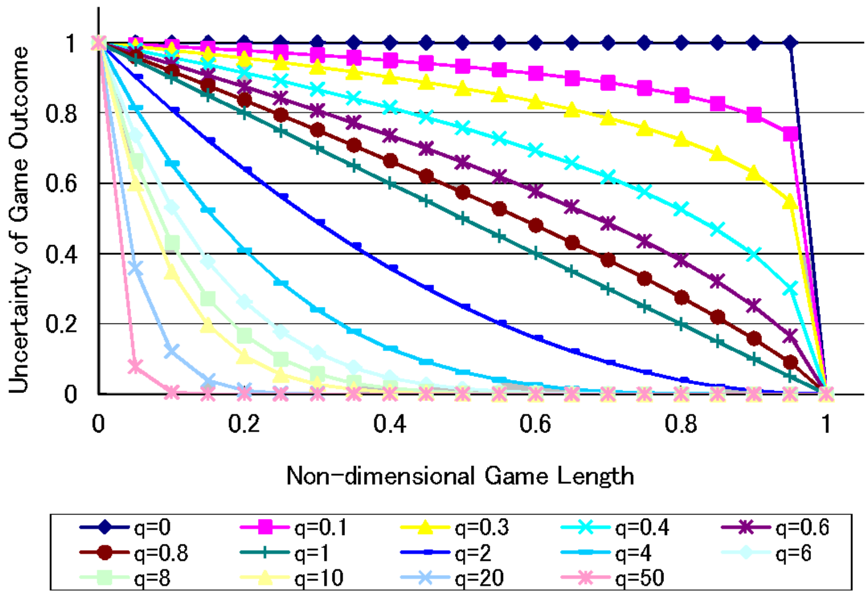


Figure 2: Uncertainty of game outcome ξ against non-dimensional game length η

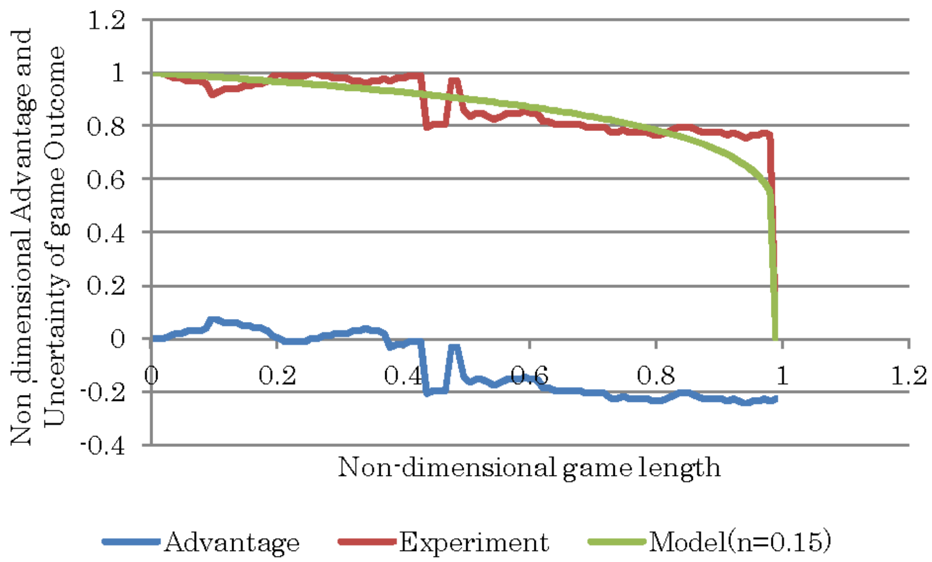


Figure 3: Non-dimensional advantage $\alpha(\eta)$ and uncertainty of game outcome ξ against Non-dimensional game length η for Congkak.

board with 8 rows and 8 columns and a set of distinct pieces for each side(Iwatas Kasai 1994, Victor 1994). Pieces typically are disks with a light and a dark face, each side belonging to one player. The player's goal is to have a majority of their pieces showing at the end of the game, turning over as many of their opponent's pieces possible. The modern rule set used on the international tournamentstage originated in Mito,Japan.

How to play: Word, "outflank" means to place a disc on the board, so that your opponents row(or rows) of disc(s) is bordered at each end by a disc of your color. A "row" may be made up of one or more discs. Othello rules are summarized as follows.

(a) Black always moves first.

(b) If on your turn you cannot outflank and flip at least one opposing disc, your turn is forfeited and your

opponent moves again. However, if a move is available to you, you may not forfeit your turn.
(c) A disc may outflank any number of discs in one or more rows in any number of directions at the same time-horizontally, vertically or diagonally. A row is defined as one or more discs in a continuous straight line.
(d) You may not skip over your own color disc to outflank an opposing disc.
(e) Disc(s) may only be outflanked as a direct result of a move and must fall in the direct line of the disc placed down.
(f) All disc(s) outflanked in any one move must be flipped, even if it is to the players advantage not to flip them all.
(g) A player who flips a disc which should not have been turned, may correct the mistake as long as the opponent has not made a subsequent move. If the opponent has already moved, it is too late for change and the disc(s) remain as is.
(h) Once a disc is placed on a square, it can never be moved to another square later in the game.
(i) If a player runs out of discs, but still has an opportunity to outflank an opposing disc on her or his turn, the opponent must give the player a disc to use. This can happen as many times as the player needs and can use a disc.
(j) When it is no longer possible for either player to move, the game is over. Discs are counted and the player with the majority of her or his color discs on the board is the winner. Note that it is possible for a game to end before all 64 squares are filled.

Othello has fast become one of the most popular and most often played games in our history, spawning contests, and tournaments on regional, national and even worldwide levels. And the rules of Othello explained as above, are very simple and the final destination is clear enough, but what exactly you are supposed to be trying to do in the early and middle stages of the game is unclear. Data analyses: The present Othello game is played by Huy, who acts as both black and white players. The non-dimensional advantage$\alpha(\eta)$ is defined as follows,

$$\alpha(\eta) = \frac{Ad(\eta)}{ACT(1)} \text{ for } 0 \leq \eta \leq 1,$$

where Ad(η) is the advantage or evaluation function scores, ACT(1) the total advantage change at the end of game. ACT(η) is expressed by

$$ACT(\eta) = ACT(\frac{m}{N}) = \sum_{1 \leq i \leq m} |Ad(i) - Ad(i-1)|,$$

where m is current move, N the total moves at the end of game, and i the positive integer. And, $\eta = \frac{m}{N}$ the non-dimensional game length Uncertainty of game outcome ξ is expressed by

$$\xi = \begin{cases} 1 - |\alpha(\eta)| & \text{for } 0 \leq \eta < 1 \\ 0 & \text{for } \eta = 1 \end{cases}$$

Fig.4 shows how non-dimensional advantage $\alpha(\eta)$ and uncertainty of game outcome ξ depend on the non-dimensional game length η.

It may be evident in Fig.4 that non-dimensional advantage $\alpha(\eta)$ is always positive, so that Black keeps advantage through the game, though it is smaller than 0.1. In this figure, the best fit model curve ξ=(1 - $\eta)^{0.04}$ to the experimental data has been plotted concurrently.

5 Discussion

This section describes how uncertainty of Soccer game outcome change with increasing the game length, where the goal scores of 2010 FIFA World Cup 3rd Place(Germany vs. Uruguay) are used for the illustration. Germany wins the game against Uruguay by the score 3 to 2: This game is full of thrill, with alternating changes from offense to defense, or from defense to offense many times. The game is balanced at the start, and then Germany gets the first goal. Uruguay makes the game balanced by taking the second goal, and then Germany is reversed by Uruguay due to the latters third goal. The game is made balanced again by Germany's fourth goal. Finally, Germany gets the fifth goal near the end and keeps her lead until the end of game.

To begin with, the advantage $\alpha(\eta)$ is defined by

$$\alpha(\eta)= \frac{[S_1(\eta)-S_2(\eta)]}{S_t} \text{ for } 0 \leq \eta \leq 1,$$

where $S_1(\eta)$ is the current score sum for team 1, $S_2(\eta)$ the current score sum for team 2, S_t the total score(s) for the game, and η the normalized game length. The sign of advantage is defined in such a way that it is positive when team 1 keeps advantage, while it is negative when team 2 takes advantage.

It may be worth noting the remarkable similarity between logarithmic uncertainty of game outcome $\xi_{lu}(\eta)$:

$$\xi_{lu}(\eta) = \begin{cases} -\sum_{i=1}^{2} p_i(\eta) \log_2 p_i(\eta) & \text{for } 0 \leq \eta < 1 \\ 0 & \text{for } \eta = 1 \end{cases}$$

where $p_1(\eta)$ and $p_2(\eta)$ are winning rates for teams 1 and 2, respectively, and the entropy $H(X)$ defined by Shannon(1948): Information theory has been used to

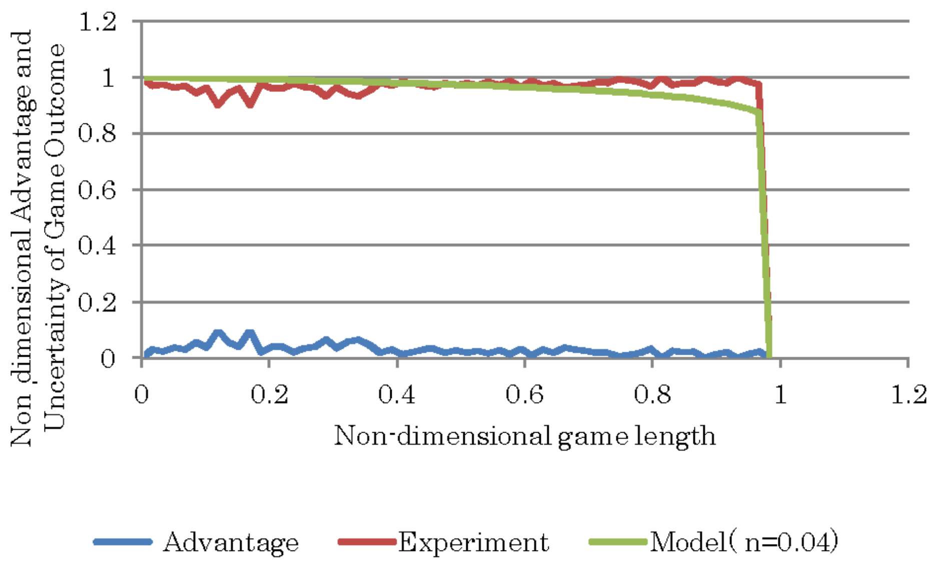


Figure 4: Non-dimensional advantage $\alpha(\eta)$ and uncertainty of game outcome ξ against non-dimensional game length η for Othello.

study the properties of random variables. If a random variable X can assume the state x, and $P(X = x)$ is the probability for X to assume the specific state x, we can define a measure $H(X)$ called entropy as

$$H(X) = -\sum_x [P(X = x)] \log[P(X = x)],$$

This is often described as the uncertainty about the outcome of X gained if one is to observe the state of x, without having prior knowledge about X. Note that in the expression of $\xi_{lu}(\eta)$ when base of the logarithm is 2, the unit of $\xi_{lu}(\eta)$ is "bit", when the base is Eulers number η, the unit is "nat", and when the nbase is 10, the unit is "digit". We choose the value of 2 as the base, for it is unnecessary to normalize $\xi_{lu}(\eta)$ in this case. It may be evident in Fig. 5 that $\xi_{au}(\eta)$:

$$\xi_{au}(\eta) = \begin{cases} 1 - |\alpha(\eta)| & \text{for } 0 \leq \eta < 1 \\ 0 & \text{for } \eta = 1 \end{cases}$$

is always smaller than the logarithmic uncertainty of game outcome $\xi_{lu}(\eta)$. It may be evident that the logarithmic uncertainty of game outcome $\xi_{lu}(\eta)$ obscures the uncertainty of game outcome by introducing the logarithmic value of winning rate $p_i(\eta)$. Thus, it is here suggested the logarithmic uncertainty of game outcome $\xi_{lu}(\eta)$ or Shannon's entropy provides only a measure of uncertainty of game outcome, but not itself. The uncertainty of game outcome is considered to be given by the present proposed advantageous uncertainty of game outcome $\xi_{au}(\eta)$.

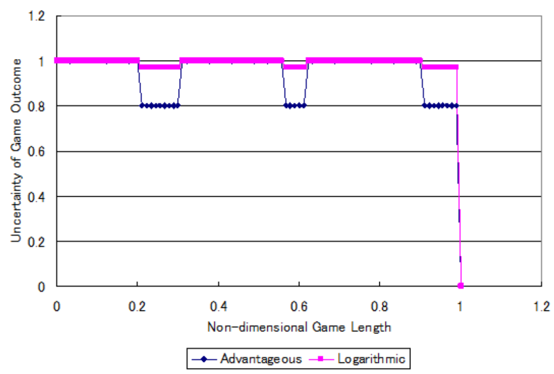


Figure 5: Uncertainty ξ_u of game outcome against normalized game length η for 2010 FIFA World Cup, 3rd Place.

6 Conclusion

New knowledge and insights obtained through the present study have been discussed and summarized as follows.

(a) Uncertainty of game outcome ξ for the present game record of Congkak is approximated with the model curve:

$$\xi = (1 - \eta)^{0.15},$$

while for that of Othello with the model curve:

$$\xi = (1 - \eta)^{0.04},$$

where η is the non-dimensional game length. This means that Othello is more balanced than Congkak in the games shown in this study, and thus it is considered that the former is more exciting than the latter. However, it must be noted that this conjecture is neither universal nor objective, because the results are highly depending on individual feeling or emotion of game players. According to the classification by Iida et al(2011b), Othello can be classified as "one-sided game", while Congkak as "seesaw game".

(b) It is inferred that the logarithmic uncertainty of game outcome or Shannon's entropy(1948) provides only an order of uncertainty of game outcome, but not itself: It is considerd that the value required is given by the present proposed advantageous uncertainty of game outcome.

(c) An information dynamic model representing the uncertainty of game outcome has been derived based on the fluid mechanics. Its usefulness has been confirmed by comparing with the present actual game experiments due to Congkak as well as Othello.

(d) Congkak has been recently introduced into Japan for the first time, as far as the present authors are aware of, and analyzed in order to explore the game potential with aiming at the future improvement regarding its entertainment. As the result, it is realized that Congkak is a unique regional game at South-East Asia, having a high possibility to spread out widely.

7 References

Shannon, C.(1948) A mathematical theory of communication. Bell System Technical Journal, 324-423, and 623-656.

Nyquist, H.(1924) Certain factors affecting telegraph speed. Bell System Technical Journal, 324-346.

Hartley, R.V.L.(1928) Transmission of information. Bell System Technical Journal, 535-563.

Iida, H., Nakagawa, T., Spoerer, K.(2011) Information dynamic models based on fluid mechanics. Entertainment and Computing(to appear).

Iida, H., Nakagawa, T., Spoerer, K., Sone, S.(2011) Three elemental game progress patterns. IScIDE2011(to appear)

Iida, H., Nakagawa, T.(2011) Game information dynamics. J. Anacleto et al(Eds.) ICEC2011, LNCS6972, pp.403-6, 2011.

Stokes, G.(1851)On the effect of the internal friction of fluid on the motion of pendulums. Cambridge Philosophical Transactions, , 8(1851); Math. And Phys. Papers, , 1-141, Cambridge 1901.

Shlichting, H.(1968) Boundary-Layer Theory(translated by J. Kestin) 6th Edition, McGraw-Hill, new York, 82-3.

Hansen, M.(1928)Die Geschwindigkeitsverteilung in der Grenzschicht an einer eingetauchten Platte. ZAMM 8, 185-99.

Wang, C.Y.(1991) Exact solutions of the steady-state Navier-Stokes equations. Annu. Rev. Fluid Mech. 23:159-77.

Solso, R.(1994) Cognition and the Visual Arts. pp.294, MIT press, Cambridge.

Tsuruoka, Y., Yokoyama, D., Chikayama, T.(2002)Game-tree search algorithm based on realization probability. ICGA Journal, 25, 145-52.

Culin, S.(1894) Mancala: The National Game of Africa. Report of the National Museum, Philadelphia, USA, 597-611.

Hellier, M.(1907) Notes on the Malay game Jongkak. Journal of the Royal Asiatic Society, Straits Branch, 49-93

Overbeck, H.(1915) New notes on the game of Chongkak. Journal of Asiatic Society, Straits Branch, 58:8 .

Iwata, S., Kasai, T.(1994) The Othello game on an n*n board is PSPACE-complete. Theor. Comp.Sci. 123, 329-40.

Victor, A.(1994)Searching for solutions in games and artificial intelligence. Ph.D Thesis, University of Limburg, Maastricht, The Netherlands.

Data-Driven Modeling of Target Human Behavior in Military Operations

Elizabeth Mezzacappa, Ph.D.
Gordon Cooke, MEME
Gladstone Reid, MSBMS
Robert DeMarco, MSBMS
Charles Sheridan BA
John Riedener, MSSE
Target Behavioral Response Laboratory
Armaments Research, Development, and Engineering Center (ARDEC)
RDAR-QES-D Building 3518
Picatinny Arsenal
(973) 724-8067
john.riedener@us.army.mil

Keywords:
data, human behavior, model building, verification, validation, crowd simulation, non-lethal weapons

ABSTRACT: *This paper describes the Army-funded exploratory work in progress at the Target Behavioral Response Laboratory. The final objective of the project is to develop "lab to laptop" experimental data-based general approaches to modeling and simulation of human behavior and quantitative methods of verification and validation. Crowd behavior data were collected under controlled laboratory conditions. Mathematical models of human behavior were derived which were then coded into computational models to produce predicted paths. These processes allow visual and statistical comparisons between outputs from simulations and behavioral data collected in the laboratory from human subjects. The results of these preliminary efforts will initiate further work in the methods of incorporating human behavioral data into models and validation procedures.*

1. Introduction

One of the most daunting challenges for the modeling and simulation (M&S) of human behavior is the lack of data to both build computational models and to validate resulting programs (Zacharias, MacMillan & Van Hemel, 2008; Zhou et al, 2010). We propose that one solution is a closed feedback loop collaboration between the operations research analysts developing predictive M&S and researchers of human behavioral responses. The US Army's Target Behavioral Response Laboratory (TBRL) has conducted human behavioral experiments since 2004, examining human behavioral response to a variety of stimuli relevant to non-lethal weapons (light, sound, blunt impact). Human behavioral responses to non-lethal weapons are examined in a number of testbeds including vehicle controlled entry point, convoy protection, sniper defeat, and room entry scenarios (https://defensemetawiki.pae.osd.mil/DAC/index.php?title=Target_Behavioral_Response_Laboratory).

Most mature of the TBRL's research programs are the protocols examining crowd behavior in response to non-lethal weapons and systems. Access to data on the behavior of crowds faced with a control force has led the TBRL to develop a unique and innovative approach to analytical and predictive M&S. By leveraging these crowd data, the TBRL has developed processes by which data from behavior of real persons in tactically relevant scenarios are the analytical link to the computational model. TBRL received funding from an ARDEC In-house Laboratory Independent Research (ILIR) award to 1) develop and document methods and processes to generate computational models from mathematical models calculated from crowd human behavioral data, and 2) to develop and document methods and processes to quantitatively verify and validate crowd human behavioral models. The specific goal of the study is to contribute to the creation of M&S operational planning tools to provide commanders with the capability to predict crowd response to non-lethal weapon tactics, techniques, and procedures. The broader goal is to develop a symbiosis between lab and laptop, where experimental designs are configured to facilitate M&S model building and validation, and M&S outputs are used to extend theory and support experimentation.

In this paper we first give a brief overview of the relevant tenets of Lewinian field theory which provides both the conceptual framework for our experimentation and the conceptual model for our mathematical and computational models of crowd behavior. We then describe the laboratory experiment that generated crowd behavioral data for our model building. Then an overview of the mathematical procedures that generated the computational model and simulations will be presented, followed by analyses and discussions of the resulting output. The paper will conclude with possible directions in which these initial efforts will lead our future M&S efforts.

2. Conceptual Framework/Conceptual Model

2.1 Lewinian Field Theory

The crowd research program was developed under the theoretical framework of Lewinian Field Theory (Lewin, 1935, 1936). This orientation is especially valuable to the M&S of crowd behavior in response to non-lethal weapons because of its mathematical underpinnings and its specific focus on the crowd behavior most important to Soldiers (Cooke et al, 2010; Mezzacappa, Cooke & Yagrich, 2008).

2.1.1 Mathematical Underpinnings

This metatheory states that human behavior is a function of the person and of the person's environment, where the state of the person and that of the environment are not independent of each other (Figure 1). To understand or to predict behavior, the person and the person's environment have to be considered as one constellation of interdependent factors. Thus, the task for crowd behavioral researchers is to identify the critical variables affecting people in the crowds and then to derive the equations linking the predictor variables to the outcome behavior. These equations then form the algorithm needed by M&S efforts.

$$\vec{B} = f(p, e)$$

Figure 1

Lewin's early methods of psychological research consisted of observations of children's movement (locomotions) toward positively valenced areas (attractors, e.g., a candy) and away from negatively valenced areas (repulsors, e.g., a bully). He theorized that locomotion behavior is induced by the resultant force arising from the constellation of positive and negative psychological forces toward and away from goal areas (Figure 2).

$$\vec{B} = \overline{G_1} + \overline{G_2} + \cdots + \overline{G_n}$$

Figure 2

2.1.2 Military Relevance

The focus on understanding determinants of movement toward and away from goal areas is critical to understanding and evaluating the effectiveness of non-lethal weapons. Capability requirements developed for non-lethal weapons and Soldier reports demonstrate very clearly that the primary function of the non-lethal weapon is to control locomotion behavior of crowd members (Joint Non-lethal Weapons Program, 2011; North Atlantic Treaty Organisation, 2009; Mezzacappa, Cooke, & Merenda et al, 2011). That is, Soldiers routinely employ non-lethal weapons in crowd situations to drive crowds away or stop their approaching a protected area. The locomotion (in an x,y coordinate plane) of a crowd member toward an attractive target area (e.g., the embassy that is a target for throwing rocks) can be expressed mathematically within the field theoretical framework (Figure 3).

$$\overrightarrow{G_{Target}} = \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = [\alpha] \begin{bmatrix} p \\ e \end{bmatrix} = [\alpha] \begin{bmatrix} x_i \\ y_i \\ x_G \\ y_G \\ \vdots \\ M \end{bmatrix}$$

Figure 3

The situation where a Soldier with a threatening weapon (i.e. a negatively valenced source of repulsive force, Figure 4) stands in defense of the area of attraction is a more complex constellation of positive forces toward the the target and negative forces away from the weapon (Figure 5).

$$\overrightarrow{G_{Weapon}} = \begin{bmatrix}\Delta X\\ \Delta Y\end{bmatrix} = [\beta]\begin{bmatrix}p\\ e\end{bmatrix} = [\beta]\begin{bmatrix}x_i\\ y_i\\ x_G\\ y_G\\ \vdots\\ M\end{bmatrix}$$

Figure 4

$$\vec{B} = \vec{G_1} + \vec{G_2} = \overrightarrow{G_{Target}} + \overrightarrow{G_{Weapon}}$$

Figure 5

2.2 Lewinian Experimental Configurations

In reference to Figures 3 and 4, experimentation on crowd response to non-lethal weapons seeks to identify the specific person (p) and environmental (e) constellation of variables that are capable of generating locomotion behavior. Subsequent vector regression analyses are then carried to estimate the strength of the relationship between these variables and locomotion behavior, the α and β coefficients and an error term. The resulting mathematical equation becomes the computational basis of simulations of crowd behavior in response to non-lethal weapons.

2.3 Data-based model building

These methods stand in contrast the typical approach. The typical approach for human behavioral representation is to select theories of human behavior thought to be relevant to the specific scenario and turn relationships among variables specified in the theory into code (Moya, McKenzie, & Nguyen, 2008; Zacharias, MacMillan & Van Hemel, 2008) with a heavy reliance subject matter experts or development and validation of models and resulting simulations. Criticisms of the current state of the art include lack of data on human behavior and quantitative methods of verification and validation, incomplete and conflicting theories of human behavior, difficulty in turning theories of human behavior into code, architecture-specific models,. Most critically, there is recognition of the lack of real-life data to provide guidance for these M&S efforts, and methods to assess how well these M&S efforts relate to actual real life human behaviors (Zhou, et al 2010).

TBRL's procedures focus on modeling of human behavior based on human behavior data collected under controlled tactically relevant experimental conditions, as opposed to theory or opinion. This approach may be seen as providing methods of model generation that are generally free from constraints related to the underlying conceptual modeling of agents or a specific cognitive architecture. Resulting models are then more likely to be reusable, composeable, and interoperable.

3. The Crowd Experiment

3.1 Subjects

All procedures were approved by the local human research subjects ethics board (ARDEC IRB #10-0002, "Effectiveness Testing for Crowd Management with Non-Lethal Weapons). Participants were recruited from the general population to participate in an investigation on "Crowd Movement." Fifty-two men and women participated in one of seven experiment days. Subjects were healthy local residents or Picatinny Arsenal employees, all over the age of 18.

Subjects targeted a protected area with simulated rocks for points/money. The area was protected with control force tactics utilizing foam projectiles, directed energy, and acoustic weapons in an attempt to cause the subjects to lose points/money (See Mezzacappa, Cooke, Sheridan et al, 2011 for more information). After two no-weapon comparison baselines, each directed energy and projectile weapon condition was tested four times (4 trials); the acoustic weapon and the no-weapon conditions were tested twice (2 trials). Exposure to the acoustic weapon was limited to protect the subjects against permanent auditory damage. Subjects earned money for scoring points during the test and lost money for being hit by the control force during the test. Subjects also were paid $20.00/hr for participation. The single session experiment lasted 4-5 hours long.

3.2 Data Collected

Following from our previous work, several measures were recorded (Cooke, et al, 2010; Mezzacappa, Cooke, Reid, DeMarco, Sheridan & Riedener, 2011; Mezzacappa, Cooke, & Yagrich 2008). Simple behavioral data were recorded, and include rock throwing success, rock throwing attempts, and number of weapon hits sustained by the crowd members. Standard demographics such as age and gender were also recorded.

Underlying psychological variables were assessed. Before each of the trials, subjects were given a questionnaire asking them about their plans and expectations for the next encounter with the control force and his weapon ("I plan to run away from the control force"). After each of the trials, the subjects were asked about their experience during the encounter ("I could escape"). Responses to these questionnaires are meant to capture the constellation of psychological forces impinging on the subject, directing and energizing his or her behavior.

Sociometric data were also recorded through questionnaire and observation. Subjects were asked to identify any prior acquaintance among the other subjects (friends and family sometimes participated together). At the conclusion of the trials, subjects also were asked to identify other subjects who they felt exhibited leadership behaviors. Video recordings were taken during the trials to assess social interactions. Sociometric questionnaires and coding of recorded social interactions create the adjacency matrix for input into social network analyses (Mezzacappa, Cooke, & Yagrich, 2008; Mezzacappa et al, in press). These psychosocial measures are assessed based on the assumption that social relationships among the crowd members have a significant effect on individual and collective behavior.

All of these behaviors are of interest in understanding crowd behavior in tactical situations. However, TBRL's mission is to test the effectiveness of non-lethal weapons and systems. To reiterate, effectiveness of non-lethal weapons in crowd scenarios is assessed by evaluating how well the weapon controls the location and movements of the crowd members. To that end, during the experiment a computer recorded the subjects' location, orientation, and locomotion through the testbed. Specifically, motion capture cameras and video recording cameras (visual and audio) recorded the behavior of the members of the crowd during the entire experiment. TBRL has used this paradigm successfully to develop processes for assessing relative effectiveness of fielded and simulated non-lethal weapons, regardless of specific energy or mechanism employed by the device (Mezzacappa, Cooke, Sheridan et al, 2011).

In the present effort we have used the data to develop mathematical and computational models to simulate crowd member locomotions in response to non-lethal weapons. Preliminary efforts are detailed in the following sections.

4. Modeling and Simulation Procedures

4.1 Creation of the Mathematical Model from Subject Data

Two different scenarios were modeled. The first situation was of unimpeded approach toward the target—no control force and no weapon was used against the crowd when they approached the target to throw rocks (field determined by a single source of attraction). The second situation was of the crowd's approach to the target that was defended by a stationary control force wielding a projectile weapon (field determined by a source of attraction and source of repulsion). These two situations are thought to represent differing psychological constellations with different resultant behaviors.

As an initial step, coders created mathematical models of human crowd behavior based on motion capture data collected for this purpose. Following from the TBRL research program, the primary behavior of interest was locomotion toward a target or goal Figure 6. Motion capture methods were used to capture X,Y coordinate locations, and therefore the paths persons took toward the target.

The raw motion capture data were first processed using an input module created in MatLab. The primary results of this step are an output matrix of the subject data and a matrix of predictor variables. For these initial efforts, subject's current location and locomotion were the variables to be predicted from subject's initial and previous location. The results of this step were passed to a statistical model module. Based on the predictors (X,Y location), the module computed a best fit to a non-linear model predicting the velocity vector in both the X and Y components, thus generating model coefficients for change in location in X and Y coordinates based on empirical data. For the more complex model with the control force, components include a relative radial and tangential component.

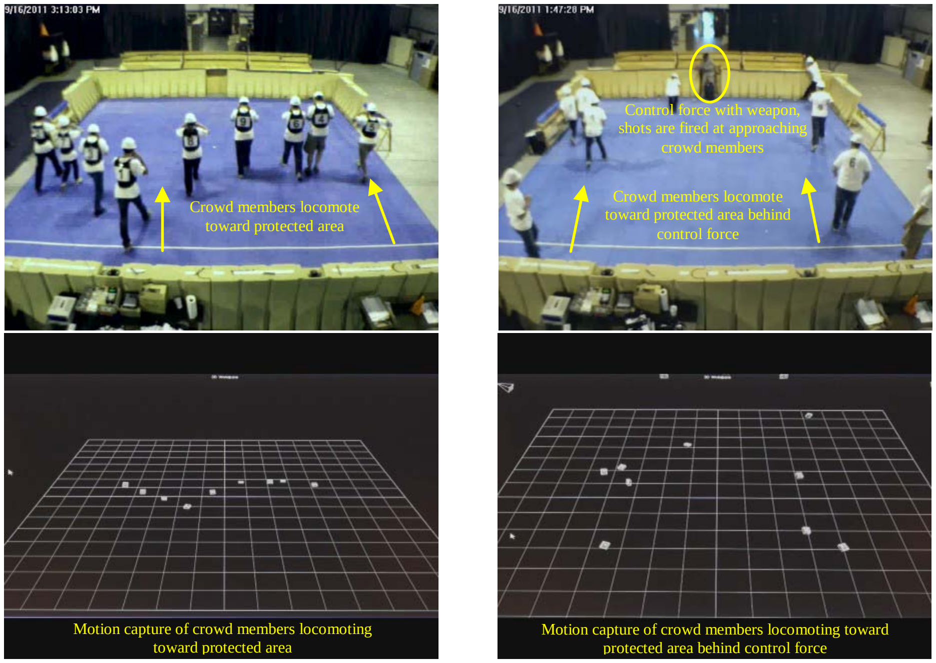


Figure 6

4.2 Calculation of Agents' Predicted Paths

These resulting regression equations were then used as the computational model underlying a simple simulation algorithm to predict a simulated crowd member agent's change in location over time. The simulation module was a MatLab file that was built into an independent function that executed for each agent a time stepped simulation of a real subject's behavior, based on the provided model and start conditions. At each time step, the new locations were calculated and time advanced. The calculated current state was updated and appended to the resulting file. Then the function stepped through each iteration calculating the change in X and Y directions, the change in distance, and the change in position for each agent.

4.3 Comparison of Subject Data and Agent Paths

The display module was a MatLab file that was built into an independent function that displayed the time plots of the real subjects' data and predicted agent paths. That is, the function created plots of the raw captured data and the simulated movement pattern for each agent crowd member, allowing for a side-by-side view of movement patterns. The function scaled the plots appropriately so that both plots had the same axis limits.

5. Results

A graphical visual comparison can be made between the mathematical representations or parsing of the behaviors as they are modeled in the computer and the data recorded in laboratory (Figure 7, Figure 8). In addition, calculation of the mean squared error (MSE) provided a quantitative measure of model fit to data. To our knowledge, this is the first instance where output from a simulation of human crowd behavior has been directly compared with data collected from actual human crowd behavior relevant to military operations with non-lethal weapons.

5.1 Locomotion induced by a single positively valenced goal area

Figure 7 shows the recorded locomotions and the locomotions predicted by the simulation during the baseline condition of unimpeded crowd approach toward the target (toward top of graph).

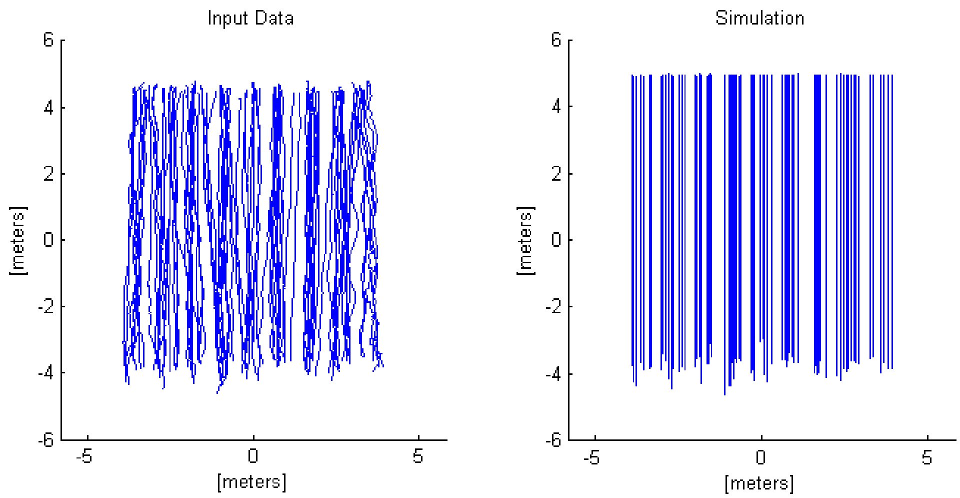


Figure 7

Data are shown on the left; simulation output are on the right. One observation is that the data show a sinusoidal pattern along the direction of the x-axis. These undulations correspond to the person's shifting of weight from one foot to the other while walking. Because the primary locomotion behavior is going in the forward y-direction toward the target area, this movement was deemed operationally non-relevant and not included in the computational model underlying the simulation. As a measure of model fit, the mean squared error (MSE) was calculated over all the sequential regression analyses. The MSE for the baseline model was 0.2523.

5.2 Locomotion induced by a combination of a positively valenced goal area and a negatively valenced goal area

Figure 8 shows the recorded locomotions and the locomotions predicted by simulations during the experimental condition involving crowd approach toward the target protected by a control force with a projectile weapon. Data are shown on the left, simulation output are on the right. In order to run the simulation according to the computation model, MatLab shifted the coordinate origin to the position of the control force; therefore the graph differs from the baseline condition in this regard.

In comparing the starting point of the locomotions, the graphs seem to indicate different initial starting locations. This appearance is an artifact of the data processing—the input data reflect a longer timespan than the simulation and are therefore are actually showing more data points-- in addition to the location points recorded during the trial, the graph of the data points also reflect the position of the subjects immediately before the start of the trial, before intiating approach behavior, behind the start line. The model is coded to initiate the locomotion prediction program once the subject crosses the start line, shown in the graph at -8m. In addition, the simulation outputs show a wider spread in the x-axis direction compared with the data. This difference in path spread results from differences in boundaries. As shown in the picture Figure 6 subjects' trajectories were bounded by the edges of the blue test bed; this feature was not included in the simulation. Model fit was then estimated; the calculated MSE was 0.4191 and 0.2958 for model fit of the radial and tangential components of the crowd movement data, respectively.

6. Discussion

The preliminary results indicate that the TBRL was successful in developing an overall process for collecting data from real people, creating simple mathematical and computational models

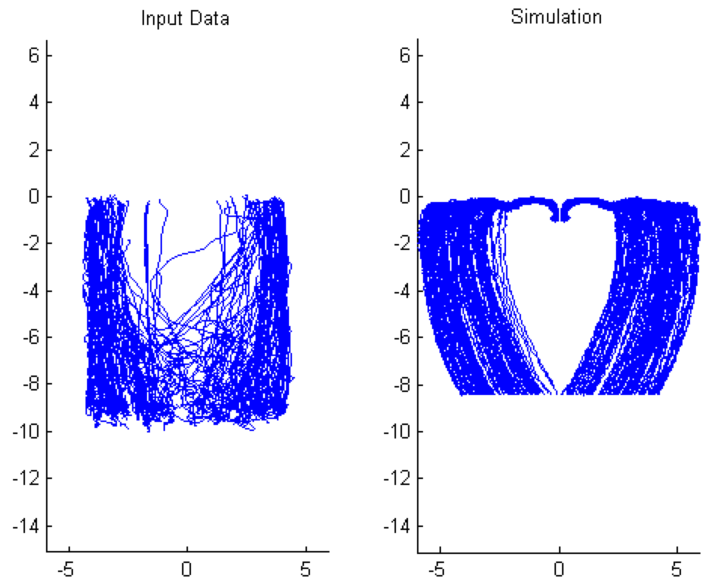


Figure 8

of crowd member movement, and generating output that is directly comparable to the initial data collected in the laboratory. This is the first critical step in creating symbiosis between lab and laptop.

Although we have succeeded in the primary goals of getting from data to model to comparisons between human data and model output, the model needs refinement. TBRL has been funded with a second ILIR to improve the initial locomotion model. Future plans include continuing model development to leverage and incorporate the other psychological, sociometric, and demographic data collected in the crowd program of research. TBRL possesses data that allows analysis among psychological variables of the crowd, weapon characteristics, rules of engagement and other operationally relevant variables such as rock throwing, martyr or instigator behavior. Finally, we plan to move our testing from small crowds in small indoor settings to our higher fidelity outdoor crowd testbed, a 1250 square meter field that can fit more than 1200 individuals. Preliminary testing of the outdoor testbed motion capture system was successfully conducted with a crowd of approximately 90 persons (Reid et al, 2011). Different crowd behavior models taking into account a richer array of predictors can then be compared using model fit statistics (mean squared error, R^2, and K-S Goodness of fit).

Access to these human behavior data and development of the processes that go from data to model building to computational model expand future avenues of TBRL's efforts in human behavioral representation in modeling and simulation. Beyond crowd modeling, TBRL also possesses data modeling driver behavior at vehicle controlled entry points, aggressive acts against convoys, and sniper behavior in response to non-lethal weapons. We will leverage our research efforts and data to create, evaluate, and identify mathematical models that will yield more accurate computational models; and simulations whose outputs will more accurately predict both the actual behavior of humans recorded in the laboratory, as well as the actual behavior of humans in theaters of operation.

7. References

Cooke, G., Mezzacappa, E., Sheridan, C., DeMarco. R., Tevis, K., Short, K., Reid, G., Jaffery, N., Riedener, J. (2010). Topology and individual location of crowds as measures of effectiveness for non-lethal weapons. *Proceedings of the 27th Army Science Conference.*

Joint Non-Lethal Weapons Program (2011). Non-Lethal Weapon Requirements Fact Sheet, http://jnlwp.defense.gov/pressroom/nlwr.html Accessed 29 December 2011.

Lewin, K. (1935). *A dynamic theory of personality*. New York:McGraw-Hill.

Lewin, K. (1936). *Principles of topological psychology*. New York: McGraw-Hill.

Mezzacappa, E. S., Cooke, G., Yagrich, K. (2008). Network science and crowd behavior metrics. *Proceedings of the 26th Army Science Conference*.

Mezzacappa, E.; Cooke, G.; Merenda B.; Jaffery, N.; Galonski, L.; Hedderich, E.; Sheridan, C.; Riedener, J. (2011). Crowd Dynamics and Military Interactions Review Updated. (TBRL-NLE10-TR-11002). Picatinny Arsenal, NJ: Target Behavioral Response Laboratory.

Mezzacappa, E.; Cooke, G. Sheridan, C.; DeMarco, R.; Reid, G.; Tevis, K.; Short, K.; Jaffery., N.; Riedener, J. (2011) Effectiveness testing for crowd management with non-lethal weapons (TBRL-NLE10-TR-11003). Picatinny Arsenal, NJ: Target Behavioral Response Laboratory.

Mezzacappa, E.; Cooke, G.; Reid, G.; DeMarco, R.; Sheridan, C.; Riedener, J. (2011). Mathematical capture of human crowd behavioral data for computational model building, verification and validation. *Proceedings of the 20th annual Behavior Representation in Modeling & Simulation (BRIMS) Conference*, March 21-24 2011, Sundance UT.

Mezzacappa, E. S., Cooke, G., Short, K. R., Sheridan, C., Reid, G., Tevis, K., DeMarco, R., Riedener, J., Jaffery, N., Yagrich, K. (in press). Social network analysis of crowds.

Moya, L. J., McKenzie, F. D., & Nguyen, Q. H. (2008). Visualization and rule validation in human-behavior representation. *Simulation & Gaming*, 39, 101-117.

North Atlantic Treaty Organisation (2009) SAS-078 Requirements Descriptions (18 Nov 2009). North Atlantic Treaty Organisation, Research and Technology Organisation.

Reid, G.; Mezzacappa, E.; Sheridan, C.; Tevis, K.; Galonski, L.; DeMarco, R.; Cooke, G.; Riedener, J. (2011). Target Behavioral Response Laboratory Outdoor Crowd Test Bed Evaluation (TBRL-NL10-TR-10013). Picatinny Arsenal, NJ: Target Behavioral Response Laboratory.

Zacharias, G. L., MacMillan, J., & Van Hemel, S. B. (Eds.). (2008). *Behavioral modeling and simulation: From individuals to societies.* Washington, D.C.: National Academy of Sciences.

Zhou, S., Chen, D., Cai, W., Luo, L., Low, M., Tian, F., Tay, V., Ong, D., Hamilton, B. (2010). Crowd modeling and simulation technologies. *Transactions on modeling and computer simulation*, 20(4).

Author Biographies

ELIZABETH MEZZACAPPA, PHD, is a Principal Investigator at the TBRL with expertise in social psychology, stress, and human behavior modeling and simulation issues.

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.

GLADSTONE REID, MSBME, is a Project Lead Engineer with expertise in experimental automation design and implementation.

ROBERT DEMARCO, MSBME, is a Project Lead Engineer and Certified LabVIEW Associate Developer.

CHARLES SHERIDAN, BA, is a Human Research Technician who has conducted recruitment, screening, and running of hundreds of volunteers through studies.

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. Before coming to the TBRL, Mr Riedener was the Performance Management Lead for the ARDEC Software Engineering Center and was integral in leading the that organization to an elusive CMMI Level 5 maturity rating.

Driving with Smith:
A Scenario-Aware Driver Model for Driving Simulation

Zhitao Xiong
Institute for Transport Studies & School of Computing
38 University Road
University of Leeds, Leeds, LS2 9JT
+44 (0)113 34 31752
tszx@leeds.ac.uk

Oliver Carsten
Hamish Jamson
Institute for Transport Studies
38 University Road
University of Leeds, Leeds, LS2 9JT
+44 (0)113 34 35348, +44 (0)113 34 35730
O.M.J.Carsten@its.leeds.ac.uk , A.H.Jamson@its.leeds.ac.uk

Anthony G. Cohn
School of Computing
E C Stoner Building 9.23
University of Leeds, Leeds, LS2 9JT
+44 (0)113 3435482
a.g.cohn@leeds.ac.uk

Keywords:
Ontology, Ontology for Scenario Orchestration, Scenario, Autonomous Vehicles, Driver Model, Driving Simulation

ABSTRACT: *Autonomous vehicles are being used in driving simulation to create realistic traffic environment but scenarios need to expose the Subjects to some reproducible and limited interactions, so autonomy is limited and the traffic environment can become dull with no ability to stand interference. In this paper, a driver model is proposed to make intelligent virtual drivers be able to operate one vehicle or a flock of vehicles based on their understanding of scenario requirements, which can make vehicles not only follow the scenario requirements and perform pre-defined actions, but also tolerate any interference and endow rich behaviours if permitted. Ontology for Scenario Orchestration (OSO) is also proposed to be the data source and can be used to build standard scenario library and standardize scenario description. An experiment is used to test the basic functionality of the model: communication mechanism, interpretation of simple Assignments and Overtake behaviour. It shows that the communication and interpretation are both working properly while lane-changing trajectory needs to be improved. This model is being tested with large scenarios at present.*

1. Introduction

Driving simulator, a facility for driving simulation, is constructed with a virtual world and a driving interface in order to imitate the driving activities of the real world. With a driving simulator, a wide variety of researches can be performed, e.g. the human factor research, because scenarios can be used in driving simulator. A scenario is a pre-defined environment that experimenters need a Subject (the participant or simulator driver) to experience; it includes the physical scenes, pre-defined traffic flow, interactions with some vehicles and measures that need to be collected. There is one scenario for one experiment.

By having scenarios, driving simulator has several advantages: 1) it has reproducible and consistent scenarios; 2) it has a risk-free environment and 3) it can create some "dangerous" scenarios. However, it is not straightforward to make a driving simulator a valid tool for researchers due to the two basic requirements: 1) the simulated vehicles in driving simulation should behave in a natural and realistic manner and 2) the scenario in the driving simulation should be reproducible. Hence, how to produce scenarios in a reproducible, realistic and natural manner is a major concern when designing the computing environment in driving simulation. Three approaches have been proposed: script-based, autonomous vehicle-based and director-based (see Table 1)

Table 1. Common Scenario Orchestration Techniques

	Script-based	Autonomous Vehicle-Based	Director-Based
Controllability	Tight	Loose	Balanced
Behaviours	Limited	Rich	Rich
Scenario Maintenance	Manual, Hard	Not Specified	Dynamic ,Easy
Scenario Replanning	No	No	No or Not Specified
Example	STISIM[1]	DRIVERSIM[2]	ISAT[3]

[1] see Allen et al. (2003)
[2] see Wright (2000)
[3] see Papelis et al. (2003)

Although autonomous vehicle-based approach has gained great attention in driving simulation by designing autonomous vehicles in order to make scenario realistic, scenarios should be handled in a user-friendly (Scenario Maintenance) and intelligent (Scenario Replanning) manner. At present, some issues may arise when running a scenario (Papelis et al., 2003), e.g., 1) designers may forget to fully consider the willingness of the Subject to engage in a scenario and 2) scenarios may fail, e.g., the vehicle fails to follow the scenario instructions of overtaking. Those issues are caused by varieties of reasons, e.g.

1) There lacks a communication mechanism between experiment designers and programmers, so they cannot fully understand the limitations/advantages of some scenario;

2) Not every proposed action can be successfully triggered because of any monitor failure and

3) There lacks a "real" intelligent driver who can drive those virtual vehicles with the knowledge of "what I need to do" and "what I know", so failures may sabotage the scenario and behaviours of simulated vehicles can be dull.

In order to find a solution for the problems above, a "*The Matrix*" metaphor has been taken to design a driver model for the computing environment in driving simulation (A detailed description of the film trilogy "*The Matrix*" can be found in The Matrix (franchise) (2011)). In short, in the film trilogy "*The Matrix*", machines created a simulated world called "*The Matrix*" and managed to get energy from humans who are kept in pods and implanted to the virtual reality "*The Matrix*". However, not all the humans in "*The Matrix*" are human avatars and controlled by humans, some of them are AI programs used for maintenance purposes, e.g., Agent Smith (Agent Smith, 2011). He can terminate any human avatar in order to prevent any instability of "*The Matrix*". Moreover, he finally became a virus in the virtual reality and got the power "*to take control over the simulated body of any human wired into the Matrix*" and " to *communicate with each other instantaneously and perceive what other humans wired into the Matrix do via a type of shared consciousness*"(Agent Smith, 2011).

Hence, being inspired by this metaphor, this research separated driver with the vehicles. Drivers are treated as "Agent Smith", and vehicles in the simulation are treated as "simulated humans" in "*The Matrix*" (at least one of them is driven by a Subject or participant). Drivers therefore should have the ability of "Agent Smith", e.g., taking control of entities in driving simulation dynamically by using Role Matching and Pair Link. As an AI program like "Agent Smith", they can understand the scenario instructions and become an autonomous, intelligent virtual driver who act according to the instructions with rich, appropriate and natural behaviours. Hence, a scenario-aware driver model is proposed. Virtual drivers equipped with this model maintain memory of outside world and itself, which includes an action plan with action profiles (constraints) and pre-/post-conditions so that the virtual driver equipped with this model can have their own understanding of scenario and outside world. Decisions are then made based on the understanding and automated planning. The virtual driver can operate one vehicle or a flock of vehicles according to the scenario Assignment.

This model can 1) make vehicles be operated by virtual drivers who can "understand" the scenario requirements and 2) make vehicles/drivers commit to the scenario - any failure can force them retry if necessary. Its data source – OSO (Ontology for Scenario Orchestration) can 1) make the communication between simulator users more smooth and easy and 2) standardize scenarios, not only their procedure but also descriptions. SDF (Scenario Definition File) is generated by using OSO. Some essential concepts have been given as follows,

Assignment is what Smith needs to do during a scenario and includes (influenced by Willemsen(2000)) 1) how often to trigger (frequency of the Assignment: "when", "whenever", "as long as" and "every"), 2) what to monitor (event or state) and 3) what to do (actions or Assignments, so Assignments can be nested);

Flock refers to a platoon of vehicles that has a leader vehicle driven by a Smith. This Smith is being called as Leader Smith or Virtual Leader Driver, which manages the flock with appropriate orders;

Phase means a phase of a scenario as indicated by its literal meaning. Phases in a scenario will be related in some way but have different measures and interactions;

Role Matching is a mechanism used to decide which vehicle or flock a Smith should drive. It is determined by the Assignment, which includes required vehicle model and spatial goal. At present, the only factor Smith needs to consider is the spatial goal;

Situation refers to some standard and common interactions that a Subject commonly be exposed to, which always involves more than one vehicle, e.g. a "Following-Pressure" Situation involves two vehicles and a flock. One vehicle pushes the Subject as a follower and another vehicle will be the Subject's leader. A flock is used to generate an oncoming traffic flow that prevents the Subject from overtaking;

Smith is a virtual entity that can control corresponding objects in simulation, e.g., a dog, a vehicle, etc. At present, Smith is just the virtual driver and can control vehicles only. Smith is inspired by Agent Smith (2011);

This paper will summarise the progress in designing and implementing scenario-aware driver model and its data source - OSO (Ontology for Scenario Orchestration). It is organised like this: Section 2 will first have a general description of OSO followed by Section 3 that is dedicated to show the driver model. Section 4 will include an initial experiment and finally, Section 5 will summarise the paper.

2. Ontology for Scenario Orchestration (OSO)

OSO is being developed using Protégé (Protégé, 2011) and designed to be the data source. It contains two levels: the first level is General Scenario Model (GSM), which contains three sub-levels:

1). Domain Knowledge: It contains 33 super-classes, which specifies abstract objects that can be used in driving simulation except the Logic Road Network., e.g., class ***Assignment*** *s*pecifies what smith needs to do;

2). World Knowledge: It specifies the Logic Road Network with 8 super-classes, e.g., class ***Intersection*** specifies the properties of an intersection;

3) Action Knowledge: It specifies what actions Smith can use and related action profiles (constraints), e.g., class ***ActionProfile*** indicates the constraints when executing some action.

The second level is Specific Phase Model (SPM) and contains just one sub-level: Phase Knowledge. It is built upon the instances generated from the knowledge mentioned above and stores what Smith needs to know in a scenario. Phase knowledge has four parts, which are listed below with examples:

Smith Feature: ***smith1*** is an instance of ***VirtualDriver***, it relates ***aggressive1*** through an object property called ***hasIndividualFeature***. ***aggressive1*** is an instance of ***Aggressive*** and indicates values of related parameters, e.g., in Lacroix et al. (2007), parameter “maximal speed” for an aggressive driver is between 140 km/h and 160 km/h;

Vehicle Model: ***vehicle***1 is an instance of ***Caravan*** ;

Logical Road Network: ***roadsegment1*** is an instance of ***RoadSegment***, which has the speed limit of 50 mph (The Logical Road Network in OSO is highly influenced by OpenDrive (Dupuis et al., 2011)) and this is specified using the data property called ***hasSpeedLimit***.

Assignment: ***assignment1*** is an instance of ***Assignment***, it will trigger an action ***break*** that is specified by objet property ***hasAction***. ***break*** is an instance of ***Action*** and related with class ***AdaptSpeed*** through property of ***has-ActionType***. ***break*** is also related to individual ***ds1*** that is an instance of ***DesiredSpeed*** through property of ***has-ActionProfile***. ***ds1*** is related to number 0 through ***hasGeneralValue*** property. All the properties associated with ***Assignment*** has been given in Table 2.

Table 2. Properties of ***Assignment***

Category	Property	Related Class
What to Trigger	***hasAction***	***Action***
	hasSituation	***Situation***
	hasAssignment	***Assignment***
What to Monitor	***hasMonitorOperator***	***MonitorOperator***
	hasReference	***Entity***
	hasStateVariable	***StateVariable***
	hasAssignmentType	***AssignmentType***
	hasEventType	***EventType***
	hasRefValue	***ReferenceValue***
Additional Information	***hasAssignmentStatus***	***AssignmentStatus***
	hasTimeStamp	***TimePoint***
	isCarSwapAllowed	***Boolean number***

isCarSwapAllowed is a data property that relates ***Assignment*** with a Boolean number. It indicates if Smith can link to another vehicle if this Assignment failed. ***hasAssignmentStatus*** is used to indicate the initial status of the Assignment: status of ***PENDING*** makes the ***Assignment*** effective from the very beginning while status of ***INITIAL*** indicates that the Assignment needs other Assignment to trigger. ***hasTimeStamp*** is no longer used due to the adoption of time nodes in ***Action*** that are used for temporal reasoning in the algorithm described in Section 4, but it is still kept to represent the start time of an assignment;

Properties in What-to-Trigger category are used to specify which instance of Action/Situation/Assignment should be related (one to one or one to many) so that their represented action/situation/assignment can be triggered.

hasMonitorOperator specifies the frequency of trigger. ***hasReference*** specifies who will be monitored. ***hasStateVariable*** specifies which state variable of the reference to be monitored. ***hasAssignmentType*** specifies what to trigger: actions, Assignments or Situations; ***hasEventType*** specifies event type of the monitor: it's a state transition (***Event***) or a state interval (***State***); ***hasRefValue*** specifies the operator and right value of a monitor. Operator is based on the event algebra in Willemsen(2000).

At present, OSO has about 360 classes and 120 properties. Moreover, the knowledge in OSO is designed to reflect the hierarchy of the driver model, e.g. ***IndividualFeature*** is used to specify the knowledge in the Memory level. The design of Smith will be elaborated in the next section.

3. Smith (Virtual Driver)

A PDA (Perception-Decision-Action) model is adopted and interpreted in a hierarchical goal-oriented manner derived from ECOM architecture (Engström et al., 2007). The model is illustrated in Figure 1.

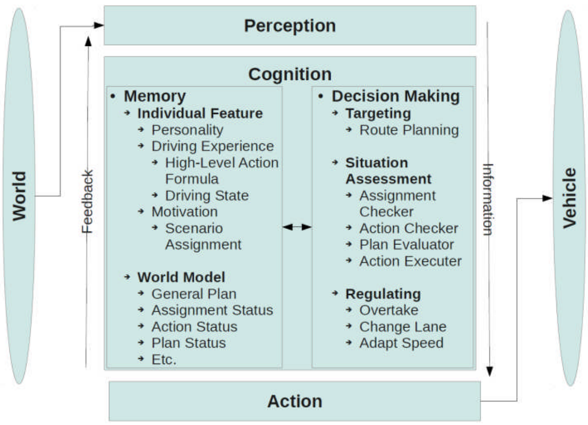


Fig. 1. Scenario-Aware Driver Model

Perception is used to sense the outside environment and make necessary interpretation. As a driver, he/she should have the following sensing abilities (Peters et al., 2007): vision, hearing, touch and proprioception. At present, the information received by Smith is concentrated on the visual information, so the sensing ability is now based on vision.

Cognition is used to maintain memory and make decisions. The driver contains two sets of memories: Individual Feature and World Model. Individual Feature contains exactly 14 structures derived from Dewer et al. (2007), e.g. Experience, Personality, Emotions etc. When Smith is initialised, it adopts different parameters based on the identifier of those individual differences, e.g. "Aggressive" Driver will adopt aggressive-related parameter from the SDF. Smith has only three sets of Individual Feature at present, which are Personality, Driving Experience and Motivation. Driving Experience makes Smith know 1) how to perform high-level actions, e.g., the "Chase" action will let Smith have spatial goal of "be that vehicle's follower" and 2) which Driving State he/she is in, namely Junction, Rural Road, Urban Road, Motorway, etc. Every state has different sets of memories regarding available actions, traffic rules, etc. Motivation refers to Assignments specified by SDF. The World Model includes the logical road network, traffic rules, weather, previous Memory of World Model, etc.

The Decision Making layer has three sub-layers, namely Targeting, Situation Assessor, Regulating. Targeting is used to plan the route for Smith and to set goals for the Situation Assessor. The Situation Assessor is used to supervise the Assignment, route and any safety requirements from Memory. Regulating is to use any relevant tactical driving behaviour to drive safely and satisfy the goal from Situation Assessor. Situation Assessor is using an algorithm called "Autonomous Local Manoeuvre and Scenario Orchestration Based on Automated Action Planning" to maintain the General Plan, which is the plan that Smith needs to follow in order to finish the whole scenario. This algorithm is influenced by the temporal reasoning algorithm in Hadad et al. (2003) and some terms/concepts are also borrowed from this paper, e.g., metric and precedence constraints. Before a description of the algorithm, some background concepts will be given first.

An action can be a High-Level or a Low-Level Action (High-Level Action means any action that can be done in different ways while Low-level Action means an action that can be done in just one way). Each action is associated with the following parameters: ***ID, d, D, r, s, f.*** ***ID*** represents the name of the action performer. ***d*** represents the deadline of an action. ***D*** represents the duration of an action. ***r*** represents the release time of an action. ***s*** and ***f*** represent the start time and finish time of each action that will be used to generate General Plan in Memory - a temporal constraints graph Gr_{α} (Hadad et al., 2003).

A Smith has a top High-Level Action that can be either Perform-scenario or Free. Free makes Smith ignore any scenario Assignment and autonomously navigate the virtual world, however, in this case, route or a destination will be randomly chosen. Perform-scenario is divided into four sub-actions, namely, β_0, β_1, β_2, β_3,

β_0(Get-to-the-initial-state) adopts initial speed. β_1 (Generate-formation) means that Smith should "drive" vehicles to their proposed formation position in order to perform the corresponding Assignment-actions. β_2 means Perform-assignment Action, so it can be further divided into several Assignment-actions, which are represented as λ_0 through λ_n (n is the number of Assignment-actions a Smith needs to perform). β_3 (Clean-up) can have several actions that in most circumstances should be specified by experimenters as an Assignment-action, however, it can be autonomous by changing to perform the Free action. This hierarchical action network is illustrated in Figure 2

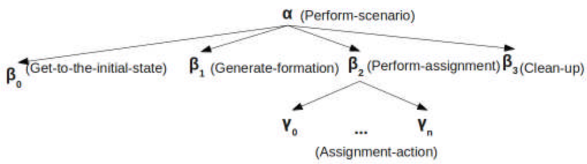


Fig. 2. Action Formula Tree for Virtual Driver (Smith) In Driving Simulation

The algorithm is described as follows:

1). Initialization Procedure:

Build a General Plan according to the formula tree in Figure 2 and Assignments from Memory (Figure 1) along with metric and precedence constraints; if it's consistent, go to 2, if not, go to 4;

2). Manoeuvre Loop:

Run the following procedure until the end of the General Plan:

(a) Assignment Checker Procedure:

Check if some monitor has been triggered according to the monitor statement in Memory or precedence constraints, set the online release time r_{β}^{online} of any triggered action β and add β to Action Execution Queue;

(b) Action Checker Procedure:

i. Check if there is some action in Action Monitor Queue, if yes, check its post-condition, if true, set the adjusted deadline of that action β: d_{β}^{adj};

ii. if the action did not finish after its duration *D*, set it to "Failure" and go to 3;

iii. Check if there is some "Failure" pre-condition regarding the action β, if it becomes true, set the status of β to "Failure" and go to 3;

(c) Regulating Procedure:

Autonomously navigate the local area according to the information stored in Memory (Figure 1), execute any behaviour when necessary by setting its online release time r_{β}^{online}. However, when executing β_2, change-lane and overtake are both forbidden unless stated by scenario Assignment.

(d) Action Execution Procedure:

i. Monitor the Action Execution Queue and check if r_{β}^{online} is consistent with the General Plan, if yes, execute the action whose r_{β}^{online} is earlier than present time, delete the action from the Action Execution Queue and add it to Action Monitor Queue; if no, go to 3;

ii. Monitor the Action Monitor Queue and check if d_{β}^{adj} is consistent with the General Plan, if yes, delete it from the Action Monitor Queue and General Plan, set Action Status and corresponding Assignment Status if necessary; if no, go to 3;

3). Scenario Replanning Procedure:

(a) Relocate position and add action β_1 to the General Plan with any new constraints; run Plan Evaluator that uses Floyd-Warshall algorithm to check its consistency (Dechter et al., 1991), if consistent, go to 2;

(b) Do Role Matching and find another vehicle that can meet the vehicle model requirement and General Plan constraint, link to that vehicle and go to 2, if Role Matching failed, go to 4;

4). Failure Broadcast procedure: Broadcast "Failure".

At present, Assignment-actions are not limited to Low-Level Actions, e.g., Overtake can be used as an Assignment-action, because every Assignment-action that is a High-Level Action can be further divided into Low-Level Actions. The output of this algorithm is "Smith Order", which is sent from drivers to vehicles and based on two Low-Level Actions: "Change-Lane" and "Adapt Speed", that is, target lane and target speed with corresponding parameters, e.g., target acceleration rate.

4. Experiment and Result

As this driver model is designed in a microscopic level, so several questions should be asked to evaluate its effects: 1) is it validated? 2) is it controllable regarding the scenario instructions? and 3) is it able to restore a scenario from failures without being noticed by the Subject? Moreover, as its data source, OSO should also be evaluated. Evaluation of OSO is based on ontology evaluation criteria (e.g. Krummenacher et al. 2007) and its usefulness in assisting scenario design. Further discussions regarding evaluation will not be dealt with in this paper but an experiment is designed to test the data exchange, interpretation of simple Assignment (and commit to it) and Overtake behaviour (Figure 3).

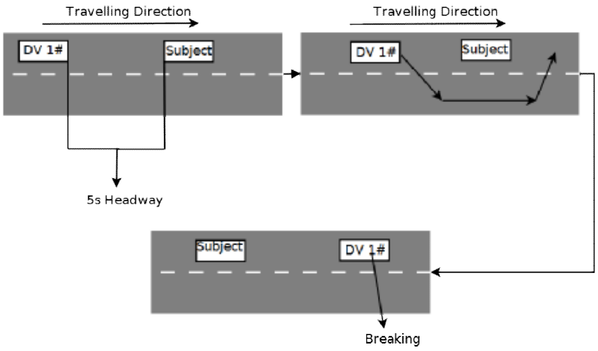


Fig. 3. Illustration of Experiment

The desktop version of the University of Leeds Driving Simulator (UoLDS) (Jamson et al., 2007) is used to run the experiment, which consists of a plain rural two-lane straight road. The Subject (block labelled “Subject”) is instructed to drive along the road without any requirement. A vehicle (block labelled “DV1$^{\#}$”, which is a Driver-Vehicle unit driven by a Smith) needs to be the Subject's leader first and then break. Because the initial position of the vehicle is placed behind the Subject, the Smith who drives “DV1$^{\#}$” needs to chase the Subject and be his/her leader first. As the high-level actions have not been developed yet, instead of describing the Smith's Assignment as “BeLeader”, the SDF specifies Assignments in this scenario simply according to the following informal description:

1) go at 20 mph, overtake the Subject and break;
2) if Subject keeps accelerating, stick to the Assignment until the end of the road.

Hence, there are three sub-Assignments: 1) overtake the subject ahead, 2) break when the follower is the subject and the Overtake action has already finished and 3) keep trying Assignment one as long as Assignment one has not been tagged as *SUCCESS*. Since the RDF/XML description is verbose, a snapshot of the Assignment one in Protégé has been given in Figure 4 and all the instances/individuals have been stated as different. Please note that the properties have been described in Section 3.

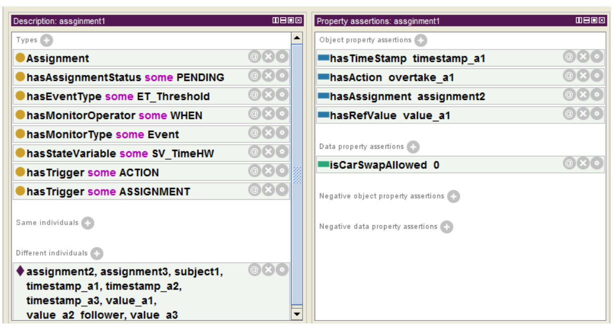


Fig. 4. Scenario Orchestration using Protégé and OSO

Experiment shows that 1) the communication mechanism is working properly; 2) Smith is able to interpret and commit to simple assignment that contains one action and one paired vehicle and 3) Lane-changing behaviour needs to be enhanced regarding its trajectory.

5. Conclusions

The driver model proposed in this paper can “understand” scenario requirements, so advanced features can be developed, e.g., Role Matching, Scenario Replanning etc. It has two major advantages:

1) Autonomous actions and scenario actions are no longer separated from each other because all actions are generated based on driver’s judgement. Scenario requirements are now a part of Memory and related decision making algorithm is also embedded into Smith’s decision making layer along with algorithms used for autonomous local manoeuvre. This feature makes it possible to endow the vehicles driven by virtual drivers with rich behaviours and persistent goals;

2) This driver model adopted a goal-oriented hierarchical architecture, which can be used to assist the design of in-vehicle devices and adopt findings in related areas, such as driver model, driving behaviour, psychology etc. Different methodologies for decision making can be also used or tested easily due to its modular structure.

This driver model is the foundation of a new framework called SOAV (Scenario Orchestration with Autonomous simulated Vehicles). It is designed to be an architecture that can orchestrate scenarios with autonomous vehicles, so that Subjects can experience rich, appropriate and reproducible scenarios. For scenario orchestration, OSO and SDF can standardize scenarios and make them shareable among different simulators; for scenario interpretation, the driver model proposed can naturally combine autonomous actions and scenario actions. SOAV can be a platform to test new driver model or driving behaviours. Moreover, different groups can have different drivers and compete with each other on a Sim in order to test which one is more efficient and intelligent.

SOAV, esp. the driver model is now being tested in the desktop version of UoLDS using large scenario that is involving six Assignments and two flocks in a simulated 23 km long rural road.

6. References

Agent Smith(2011). Agent Smith. Retrieved from the web October 16, 2011. http://en.wikipedia.org/wiki/AgentSmith

Allen, R. W., Rosenthal, T. J., Aponso, B. L. and Park, G. (2003). Scenarios Produced by Procedural Methods for Driving Research, Assessment and Training Applications. In *Proceedings of Driving Simulation Conference North America, DSC'03*.

Dechter, R., Meiri, I. and Pearl, J. (1991). Temporal Constraint Networks. *Artificial intelligence, vol49*, 61-95.

Dewar, R.E. (2007). Individual Differences. In Dewar, R.E. and Olson, P.L(Authors), *Human Factors in Traffic Safety, Second Edition*, (pp. 83 - 102). Lawyers & Judges Publishing Company.

Dupuis, M. et al. (2011): OpenDRIVE Format Specification V1.3. OpenDRIVE: VIRES GmbH.

Engström, J. and Hollnagel, E. (2007). A General Conceptual Framework for Modelling Behavioural Effects of Driver Support Functions. In Cacciabue, P. C (Editor), *Modelling Driver Behaviour in Automotive Environment* (pp. 61 - 84). Springer.

Hadad, M., Kraus, S., Gal, Y. and Lin, R.(2003). Temporal Reasoning For A Collaborative Planning Agent In A Dynamic Environment. *Annals of Mathematics and Artificial Intelligence, vol37*, 331-379.

Jamson, A. H., Horrobin, A. J., and Auckland, R. A. (2007). Whatever Happened to the LADS? Design and Development of the New University of Leeds Driving Simulator. In *Proceedings of Driving Simulation Conference America (DSC-NA),DSC'2007.*

Krummenacher, R. and Strang, T. (2007). Ontology-Based Context-Modeling. In *Third Workshop on Context Awareness for Proactive Systems* (CAPS07).

Lacroix, B., Rouelle, V., Kemeny, A., Mathieu, P., Laurent, N., Millet, G., and Gallee, G. (2009). Informal rules for autonomous vehicles in SCANeR. In *Proceedings of Driving Simulation Conference europe, DSC'09*, pages 59–68.

Papelis, Y., Ahmad, O., and Watson, G. (2003). Developing Scenarios to Determine Effects of Driver Performance: Techniques for Authoring and Lessons Learned. In *Proceedings of Driving Simulation Conference North America, DSC'03*.

Peters, B. and Nilsson, L. (2007). Modelling the Driver in Control. In Cacciabue, P. C(Editor), *Modelling Driver Behaviour in Automotive Environment*, (pp. 85 - 104).

Protégé. (2011). Protégé-OWL Ontology Editor, http://protege.stanford.edu

The Matrix (franchise) (2011). The Matrix (franchise). Retrieved from the web February 2, 2012. http://en.wikipedia.org/wiki/The_Matrix_(franchise)

Willemsen, P. (2000). *Behavior and Scenario Modeling for Real-Time Virtual Environments*. PhD thesis. The University of Iowa.

Wright, S. (2000). *Supporting Intelligent Traffic in the Leeds driving simulator*. PhD thesis. School of Computing Studies, University of Leeds.

Author Biographies

ZHITAO XIONG is a PhD student who is co-supervised by the Institute for Transport Studies and School of Computing, University of Leeds. His present research is to orchestrate scenarios in driving simulation with autonomous vehicles so that subjects can experience realistic, natural and reproducible scenarios. He is interested in Autonomous Land Vehicle, scenario control in driving simulation, driver model and driving behaviour.

ANTHONY G COHN is Professor of Automated Reasoning in the School of Computing, University of Leeds where he is Director of the Institute for Artificial Intelligence and Biological Systems. His research interests focus on knowledge representation and reasoning, and in particular on spatio-temporal representation and reasoning.

OLIVER CARSTEN is Professor of Transport Safety at the Institute for Transport Studies, University of Leeds. His major research focus is on driver interaction and safety with new driver support systems. He led the UK national project on Intelligent Speed Adaptation.

HAMISH JAMSON is a Principal Research Fellow and Manager of the University of Leeds Driving Simulator (UoLDS). He was awarded his PhD in Motion Cueing in Driving Simulators in 2011 having previously obtained a BEng in Mechanical Engineering from the University of Sheffield and a MSc. in Flight Dynamics from Cranfield University’s College of Aeronautics. He has worked on various publicly and privately funded projects involved with driver performance, road safety and traffic engineering. He also was responsible for the development of the large UoLDS, one of the leading such facilities worldwide. His current work and published interests include driver behaviour, driver distraction and driving simulation.

A Methodology for Composing Behaviour Models for Military Simulations

David P. Unrau
CAE Professional Services Canada, Inc.
1135 Innovation Drive
Ottawa, ON K2K 3G7
613-247-0342
david.unrau@cae.com

Bruno Emond
National Research Council Canada
1200 Montreal Road
Ottawa, ON K1A 0R6
613-993-0154
bruno.emond@nrc-cnrc.gc.ca

Keywords:
Decision-making, Ontology, Military Simulation, Standardization

Despite the consistent requirement for Human Behavior Models (HBMs) in defence simulation, surveys of Computer Generated Force (CGF) Artificial Intelligence (AI) reveal consistent limitations that restrict use of CGF's in training and experimentation (Parkinson, 2009). A lack of standardization and issues with scaling make it difficult to transition cognitive models from the laboratory to operational environments (Douglass & Mittal, 2011). Operational environments contain a diversity of computing devices, simulation specifications and languages, as well as a diversity of human expertise, with a strong requirement to share common representations (Kramer, Miller, Unrau, & Armstrong, 2010). Methodologies that deliver robust, re-useable, scalable HBMs for operational use will have a large, positive impact.

This poster outlines a methodology for the composition of different behavior model components into a single simulation application. If deploying composite models was straightforward, simple decision models could be combined to emulate more complex aspects of human decision-making. Sophisticated cognitive models could be combined with procedural AI to extend the capabilities of existing CGF tools. The same high-level decision model could easily be used in two different synthetic environments. This work is based on previous simulations built with multiple levels of system abstractions (Emond, 2011; Miller, Unrau, & Kramer, 2010), and the application of ontology (Tzeng, Hsu, Cheng, & Huang, 2009) and semantic web technologies (Horrocks et al., 2004; W3C OWL Working Group, 2009) as a means to provide interoperability between heterogeneous military simulation systems (Turnitsa, Padilla, & Tolk, 2010).

For HBMs of different levels of abstraction to exist in a single application, two new processes must be added to a classical sense, decide, and act loop (RTO/NATO, 2009; Wray, Laird, Nuxoll, & Jones, 2002). In figure 1, a higher-level decision model (Decide #2, red) is combined with a lower level model (Decide #1, orange). The base information flow is shown in green - the lower level HBM senses the environment, makes decisions, and acts on the synthetic environment in a conceptual representation natural to that model. To combine this HBM with a higher-level model, the information flow in blue is introduced. The more concrete concepts 'sensed' by the lower level must be abstracted for use in the higher level. Conversely, the abstract decisions must be made more specific to be actionable at the lower level.

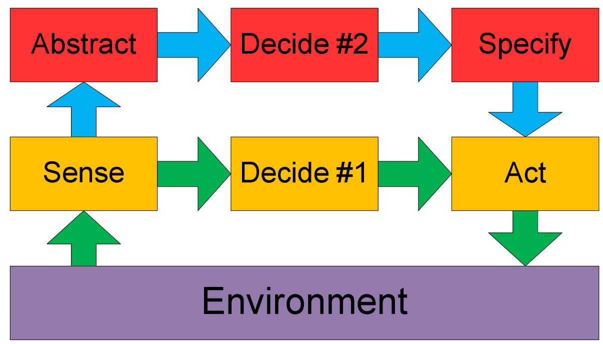


Figure 1: Abstracting and specifying as a methodology for combining decision-making models

Following this methodology, in Miller, Unrau & Kramer (2010), a motivation framework was used to specify waypoints that controlled the actions of Bohemia Interactive VBS2's AI. In recent work, a civilian model was developed by integrating a motivation framework with Presagis' STAGE. The AI in STAGE controlled the entity navigation using cost-minimization routing. The higher-level process modified the terrain feature costs based on observed military activity. In this fashion, the motivation framework specified *how* the STAGE AI should act.

The abstraction and specifying processes can also be discussed as 'translation' interfaces. In a conventional interface example (figure 2) two components transfer entity information.

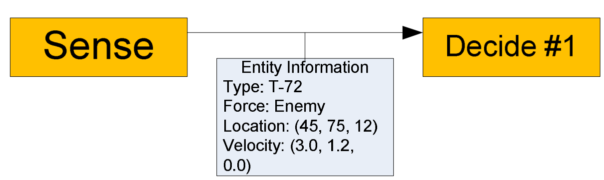


Figure 2: A conventional interface

Critically, the two components must agree exactly on the details of this interface, and the *Decide #1* component receives information exactly as it is sent from the *Sense* component, tightly coupling the internal representation frameworks of the two components. In figure 3 we introduce a 'translation' interface. As before, a *Sense* component produces information on detected enemies. Now, an ontology formally specifies the relationship between *Sense* concepts such as *Type* and *Force*, and *Decide #2* concepts such as *Threat-Level*. Importantly, the ontology bridges differences between the inputs and outputs of the two components, and the data received is not identical to the data sent.

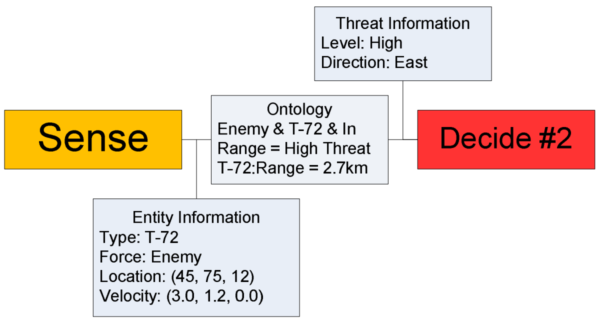


Figure 3: Definition of a 'translation' interface

Software is required to make this interface work. For instance, a reasoner could be used to deduce the inputs for the *Decide* component based on the ontology and the outputs of the *Sense* component. Alternatively, the ontology could be used to specify a Bayesian inference network that relates the *Sense* outputs to the *Decide* inputs in a probabilistic fashion.

The short example above outlines a methodology for integrating multiple, dissimilar sensory and motor element into a single HBM. Translating interfaces compartmentalize functionality, enabling re-use and scalability. The methodology can be summarized as follows: 1) Define the full scope of the HBM; 2) Define a set of simple models that together address the full scope; 3) Define the model inputs and outputs in concepts that are natural for the models; 4) Define ontologies that relate the defined concepts; 5) Integrate the simple models with translating interfaces. For re-use in a new application, the following process is applied: 1) List the concepts defined by the existing models; 2) Relate these concepts with ontologies; 3) Unsatisfied mappings now define additional models or synthetic environment elements required to complete integration.

The next step of this research is to introduce standardization through military specific information exchange and messaging standards such as the Coalition Battle Modelling Language (Blais, Galvin, & Hieb, 2005) and the Military Scenario Definition Language (MSDL Product Development Group, 2008).

References

Blais, C., Galvin, K., & Hieb, M. (2005). Coalition Battle Management Language (C-BML) Study Group Report. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.2176&rep=rep1&type=pdf

Douglass, S., & Mittal, S. (2011). *Using Domain-Specific Languages to Improve the Scale and Integration of Cognitive Models*. Paper presented at the Behavior Representation in Modeling and Simulation Conference.

Emond, B. (2011). *An abstract spatial device for human behavior models, user performance data collection, and game control user interface*. Paper presented at the Behavior Representation in Modeling and Simulation, Sundance, Utah.

Kramer, C., Miller, J., Unrau, D., & Armstrong, J. (2010). *Defining Spatial Information Requirements for Asymmetric Threat Behaviour in Simulation*. Paper presented at the RTO Human Factors and Medicine Panel (HFM) Symposium (RTO-MP-HFM-202 AC/323 (HFM-202) TP/ 365).

Miller, T., Unrau, D., & Kramer, C. (2010). *A Multi-resolution Cognitive Model of Improvised Explosive Device Emplacement*. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).

MSDL Product Development Group. (2008). Military Scenario Definition Language (MSDL). Simulation Interoperability Standards Organization SISO-STD-007-2008.

Parkinson, G. (2009). *AI in CGFs comparative analysis – summary report*. Defence Research and Development Canada – Ottawa CR 2009-201.

RTO/NATO. (2009). Human Behaviour Representation in Constructive Simulation (RTO-TR-HFM-128) (pp. 196).

Turnitsa, C., Padilla, J. J., & Tolk, A. (2010). Ontology for modeling and simulation, Baltimore, MD.

Wray, R. E., Laird, J. E., Nuxoll, A., & Jones, R. M. (2002). Intelligent opponents for virtual reality trainers.

Author Biographies

DAVID UNRAU is a Senior Consultant in the simulation systems group at CAE Professional Services Canada. His focus is the development of simulations for military training and operational research.

BRUNO EMOND is a Senior Research Officer with the National Research Council Canada Institute for Information Technology. His current research focuses on cognitive modelling technology applied to learning environments and simulation.

Adaptive Decision-making for Distributed Assets (ADDA)

Stacy Lovell Pfautz, Eric Jones, Gabe Ganberg, Andy Chang, Fred Diedrich
Aptima, Inc.
12 Gill Street, Suite 1400, Woburn, MA 01801
781-935-3966
spfautz@aptima.com

Paul Scerri, Nathan Brooks
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
pscerri@cs.cmu.edu

Ron Steed
UpScope Consulting Group, LLC
125 Ensign Drive, Mystic, CT 06355
ronaldsteed@gmail.com

Keywords:
Adjustable Autonomy, Distributed Decision Making, Human-System Collaboration, Dynamic Function Allocation

1. Introduction

There is a need for submarine and surface vessels to communicate and coordinate assets during mission planning, dynamic replanning, and mission execution to ensure that information about contacts is being shared; tasking is assigned and understood; and accurate, informed decisions are made in a timely manner. There are a number of constraints and complications that challenge this increased coordination including limited crew members and resources; uncertainty and ambiguity of sensors, data feeds, and communications; asynchronous, distributed networks; complex information needs; and shifting priorities.

To facilitate collaboration, coordination, and communication as additional onboard and off-board sensors are integrated into the current control room, we developed the Adaptive Decision-making for Distributed Assets (ADDA) decision aid. ADDA augments mission planning systems by providing methods to improve coordination and reduce uncertainty about status and tasking to enable more resilient teams. The goal of ADDA is to help distributed teams make informed, timely decisions about shared assets as they are working to accomplish multiple missions, resulting in more efficient system employment both within and across assets. ADDA aims to dynamically allocate tasking and workflow across distributed resources (manned and unmanned) to fit mission priorities given constraints, and uses adjustable autonomy to determine whether and when transfers of decision-making control should occur to optimize human-system collaboration.

The following sections describe the theoretical and algorithmic underpinnings of the ADDA effort, which resulted in a software prototype that demonstrates the potential for our approach.

2. Approach

To address these challenges, ADDA uniquely combines several technologies and innovations. A use case and concept of operations for an Intelligence, Surveillance, and Reconnaissance mission was developed to guide the development of ADDA. Aptima's Common Context Representation Framework (CCRF) models organizational structure, mission constraints and priorities, resources, assets, and capabilities. For humans and automated systems to effectively work together to perform tasks in a shared environment, all participants need access to a shared representation of the context for those tasks. This context consists of the tasks themselves, the environment, the goals and capabilities of the people and systems involved, as well as the interactions between the various elements. A significant challenge when developing this model of context is in representing it at a level of abstraction that is understandable to a human and usable by an automated system. Aptima's CCRF (Ganberg et al., 2011) provides a framework for representing the types of information needed for entities to interact and cooperate with human users in the process of pursuing common goals. For ADDA, we created CCRF models to describe the elements of our use case.

Next, coordination algorithms intelligently allocate tasking and workflow across distributed resources and teams according to the constraints and priorities characterized in the CCRF models. Intelligent agents act as "proxies" for the heterogeneous teams. ADDA utilizes adjustable autonomy between these proxies to enable scalable human-automation and transfer of control schemes, where the decision making power can reside with human operators, while also providing a flexible method for automation or agents to make decisions when required. In ADDA, we are concerned about algorithms for dynamically allocating functions, managing joint resources, information sharing, reaching joint conclusions, and other routine coordination tasks. We investigated initial allocation and coordination algorithms that would facilitate distributed tasking across human teams and human-automation teams. We leveraged a function allocation algorithm called Machinetta, developed by Carnegie Mellon University, which uses a form of Low communication Approximate Distributed Constraint Optimization Problem (LA-DCOP). LA-DCOP is specifically aimed at working well in domains where communication is highly limited and tasks change frequently, but where finding an absolute optimal solution is not critical.

Finally, novel human computer interfaces and interaction methods display recommended plans and courses of action to decision makers. In ADDA, CCRF models automatically configure proxy agents within Machinetta. The proxies cooperatively manage the coordination for the team, getting input from the agents as required and letting them know their responsibilities. This overall concept is described in Scerri et al. (2004). A simulation testbed was developed to demonstrate the models and algorithms by following our use case. Several interface mockups that illustrate how human decision-makers and team members might interact with ADDA were also developed.

3. Discussion and Future Work

ADDA was a seven month research and development project focused on improving distributed human-system collaboration and coordination through the novel application of context models, dynamic function allocation strategies, and adjustable autonomy. At the end of our effort, we successfully accomplished the following objectives: (1) Developed an initial concept of operations and a detailed use case; (2) Extended and applied CCRF to model the organizational structure, capabilities, resources, assets, environment, constraints, and other context; (3) Developed an approach for using adjustable autonomy and implemented dynamic function/task allocation algorithms for distributed team coordination within the Machinetta infrastructure; (4) Developed and implemented a prototype of ADDA that integrates Aptima's CCRF models with CMU's Machinetta software; (5) Developed initial interface mockups that illustrate how human decision-makers and team members might interact with ADDA; and, (6) Described and identified example evaluation criteria and methods to assess system effectiveness and performance.

We recommend several areas of focus for future work, including expanding the complexity of the use case, implementing the interface mockups within software, extending the coordination algorithms so they produce multiple plans that can be evaluated according to operational risks and tradeoffs, including how the allocation of decisions or tasks to human decision makers or automation may impact team resilience, and evaluating the effectiveness of ADDA for improving the performance of human-human and human-automation teams.

6. References

Ganberg, G., Ayers, J., Schurr, N., Therrien, M., & Rousseau, J. (2011) Representing context using the context for human and automation teams model. Presented at the *Activity Context Representation workshop at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI-11.

Scerri, P., Pynadath, D., Schurr, N., Farinelli, A., Gandhe, S. and Tambe, M (2004). Team Oriented Programming and Proxy Agents: The Next Generation, In *Proceedings of 1st international workshop on Programming Multiagent Systems*, Springer, LNAI 3067.

Acknowledgments

This work was funded by Dr. William Krebs from the Office of Naval Research (ONR) under Contract N00014-11-M-0296. Any opinions, findings, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of ONR or the authors' employing organizations.

Author Biographies

STACY LOVELL PFAUTZ is a Senior Scientist and leads the Human-Systems Analysis team at Aptima, where she designs and develops methods to provide seamless integration of humans and technology to enhance the performance of sociotechnical systems. Ms. Pfautz has led the research and development of multiple projects that focus on human-system interaction, adjustable autonomy, and intelligent interface development for unmanned systems.

PAUL SCERRI is an Associate Research Professor at the Robotics Institute at Carnegie Mellon University. His research focuses on multi-agent coordination and human interaction with intelligent systems. His work has contributed to the areas of adjustable autonomy, multi-agent systems and teamwork, and he developed the Machinetta coordination software.

RONALD STEED (CAPT, USN Ret.) has over twenty-five years of service as a Captain in the United States Navy's Nuclear Submarine Force. As a former Submarine and Squadron Commanding Officer, he is uniquely qualified to advise teams desiring to improve decision-making, human-error avoidance, and organizational effectiveness.

Behavior Modeling and Classification via Probabilistic Context Free Grammars

Sahin Cem Geyik, Boleslaw K. Szymanski
Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
geyiks@cs.rpi.edu, szymansk@cs.rpi.edu

Keywords:
Team behavior/interaction/coordination, Pattern recognition, Socio-cultural modeling

Social networks analysis includes examining the actions of entities in a social setting. These actions can be either interactions between entities (e.g. talking, exchanging items etc.), or actions which do not include interactions, but nevertheless are happening in a social context, hence are influenced by social relations. Such actions often contain behavioral patterns that are specific to the actions involved. It is important to understand such patterns to be able to model social environments reliably.

In this work, we introduce a novel method for modeling and classifying behavior of nodes in a social network using Probabilistic Context Free Grammars (PCFGs). Informally, PCFGs are regular context free grammars (consisting of START symbol, terminals, nonterminals and production rules), augmented with probabilities assigned to the production rules to denote how likely each rule is to be used when producing a sentence from this grammar. Given a set of action sequences, a PCFG can be automatically constructed [Geyik, 2009] to derive probabilities for these actions and concisely represent behavioral patterns based on the input data. This PCFG can either be used to predict or classify future behaviors, or to understand the relationships between these patterns via manual inspection possible thanks to conciseness of the PCFG representation.

To evaluate our proposed methodology, we present the results from processing the Mission Survival Corpus (MSC-1) dataset [Pianesi, 2008] collected by Project FBK (Fondazione Bruno Kessler). This dataset basically contains time-stamped annotations of 11 meetings of people deciding on how to proceed in a disaster scenario. The annotations include the social role label and task role label of each meeting attendant as well as an indication of who speaks at the time-stamp. Social roles (supporter, protagonist, attacker, and neutral) mainly represent the attendant's attitude towards the group's function while the task roles (giver, seeker, orienteer, neutral) represent the individual's function and technical skills. We examine the task roles taken (as a sequence) by a meeting attendant while undertaking a social role (socio label), or vice versa (we call this metric "*Which Roles Go Together*"). In our previous work [Geyik, 2010], we presented the constructed grammars and provided our interpretation of them based on manual examination that led us to understanding the underlying properties of this application domain. As a simple example, the grammar constructed for the *attacker* social role within the MSC-1 dataset is shown in Figure 1.

START → N0 M1 (0.18) M1 (0.18) M1 N0 (0.55) N0 (0.09)	N0 → n (1.0) M1 → s (0.5) | g (0.5)

Figure 1. A Sample Grammar for the Attacker Socio-Label (n-neutral, s-seeker, g-giver)

An inspection of this grammar shows that for this dataset, seeker (terminal s) and giver (terminal g) have equal probability (as defined by the right hand side of the production defining M1) for the attacker social role (i.e. an attacker is equally likely to ask questions as it is to provide facts). Such domain specific property can be easily observed due to the concise representation achieved by the PCFG modeling.

As an extension to our previous efforts, we present here the classification results of the PCFG modeling on the MSC-1 dataset. For this purpose, we utilize the aforementioned metric, "*Which Roles Go Together*", to classify the roles of meeting attendants. We first provide the separation capability of the PCFGs between roles when we utilize the whole dataset; and then, we separate the dataset into training and test partitions and evaluate the classification ability of PCFGs on the unseen data.

Table 1 presents the classification results for the social roles in MSC-1 dataset. First, four grammars (for each of four social roles) are constructed automatically, and then we try to parse the sequence that is to be classified by these four grammars. Similar to a Naive Bayesian Classifier, the classification metric depends on the sequence's production probability given a grammar, and the prior probability of the class itself (according to how frequently the role is observed in the training data). To account for unseen sequences in the training data, we inserted a smoothing nonterminal into each grammar to accept all sequences (but with a very small probability if it was not in the training data).

Between the Roles of	Separation Accuracy	2-Fold Cross Val.	10-Fold Cross Val.
Protagonist and Supporter	69.7 %	65.4 %	65.3 %
Neutral and Supporter	71.4 %	69.2 %	66.6 %
Neutral and Protagonist	63.5 %	63.6 %	60.3 %
Attacker and Supporter	94.3 %	92.6 %	93.7 %
Attacker and Protagonist	96.1 %	95.4 %	95.8 %
Attacker and Neutral	97.3 %	97 %	96.9 %
Neutral, Protagonist, and Supporter	51.7 %	51.1 %	48 %
Attacker, Protagonist and Supporter	68.2 %	63.7 %	63.7 %
Attacker, Neutral, and Supporter	70.1 %	67.8 %	65.2 %
Attacker, Neutral, and Protagonist	62.6 %	62.5 %	59.2 %
Attacker, Neutral, Protagonist, and Supporter	51.1 %	50.4 %	47.3 %

Table 1. Separation and Classification Results for the Social Roles in MSC-1 Dataset

From the above table, it can easily be observed that the attacker social role is easily distinguished from the rest of the social roles, while the others are harder to distinguish according to their action role sequences. We also see that the 10 and 2-fold cross validation results are not much worse than testing on the training data itself, which demonstrates the classification ability of the PCFG model on unseen data.

Table 2 presents the classification results for the task roles in MSC-1 dataset. From the table, it can be observed that the task role classes neutral and giver are easy to distinguish from the task role classes seeker and orienteer in terms of their social role taking, but the classification performs worse between the classes in these two groups (i.e. neutral is difficult to distinguish from giver, and seeker is difficult to distinguish from orienteer).

Future work directions in this area include various other application domains within social network context where the PCFGs can be used for prediction and classification. Examples include role recognition in automatic analysis of videotapes of clandestine meetings or the monitoring and classification of normal vs. abnormal behavior of people in other security applications. Furthermore, a question that is interesting for future work direction is whether a certain social stance is distinguishable for different application domains (i.e. the question “Is attacking behavior always as distinct as it is for the MSC-1 dataset?”).

Between the Roles of	Separation Accuracy	2-Fold Cross Val.	10-Fold Cross Val.
Orienteer and Seeker	63.6 %	66.9 %	67.7 %
Neutral and Seeker	88 %	87.8 %	86.7 %
Neutral and Orienteer	84.4 %	82.1 %	81.3 %
Giver and Seeker	87.9 %	87.2 %	87.4 %
Giver and Orienteer	83.8 %	81.9 %	81.8 %
Giver and Neutral	59.2 %	57.2 %	55.5 %
Neutral, Orienteer and Seeker	75.8 %	73.5 %	72.4 %
Giver, Orienteer and Seeker	75.3 %	72.9 %	73.1 %
Giver, Neutral, and Seeker	55.3 %	53.4 %	51.8 %
Giver, Neutral, and Orienteer	54.1 %	51.5 %	49.6 %
Giver, Neutral, Orienteer, and Seeker	50.8 %	48.4 %	46.6 %

Table 2. Separation and Classification Results for the Task Roles in MSC-1 Dataset

This work was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and by the Office of Naval Research Grant No. N00014-09-1-0607.

References

Geyik, S. C., Szymanski, B. (2009). Event Recognition in Sensor Networks by Means of Grammatical Inference. *In Proc. IEEE INFOCOM*, 900-908.

Pianesi, F., Zancanaro, M., Not, E., Leonardi, C., Falcon, V., Lepri, B. (2008). Multimodal Support to Group Dynamics. *Personal and Ubiquitous Computing, Vol. 12, Iss. 3*, 181-195.

Geyik, S. C., J. Xie, J., Szymanski, B. (2010). Behavior Modeling with Probabilistic Context Free Grammars. *In Proc. Int. Conf. on Information Fusion*.

Author Biographies

SAHIN CEM GEYIK is currently a Ph.D Candidate in the Computer Science Department, at Rensselaer Polytechnic Institute (RPI), Troy, NY. He received his BS degree in Computer Engineering from Bogazici University in Istanbul, Turkey, in 2007.

BOLESLAW K. SZYMANSKI is the Claire and Roland Schmitt Distinguished Professor of Computer Science and the Founding Director of the Center for Pervasive Computing and Networking at Rensselaer Polytechnic Institute (RPI), Troy, NY. He received his Ph.D. degree in computer science from the National Academy of Sciences, Warsaw, Poland, in 1976.

Data-Driven Modeling of Human Behavior in Military Operations

John Riedener, MSSE
Elizabeth Mezzacappa, Ph.D.
Target Behavioral Response Laboratory, Armaments Research, Development, and Engineering Center
RDAR-QES-D Building 3518, Picatinny Arsenal, (973) 724-8067
john.riedener@us.army.mil

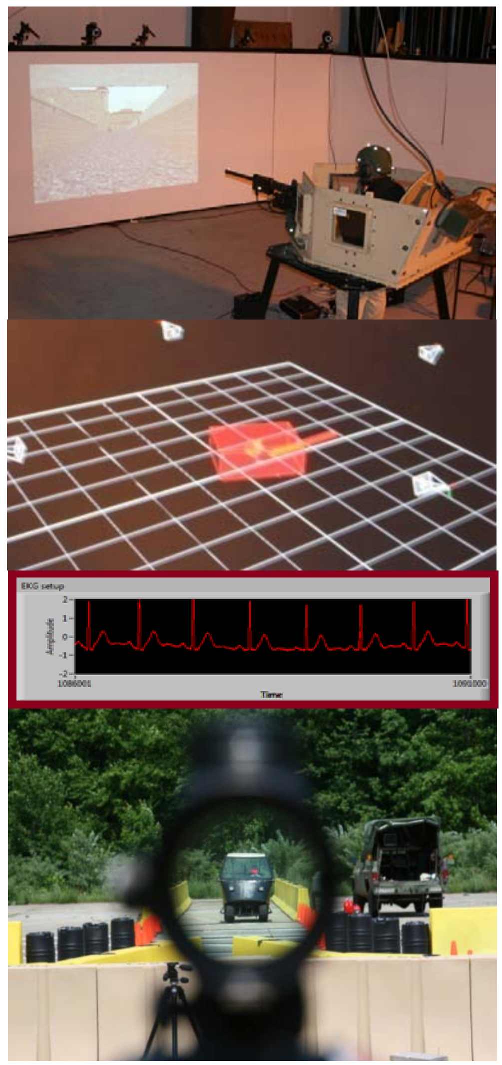
Keywords:
data, human behavior, model building, verification, validation, non-lethal weapons

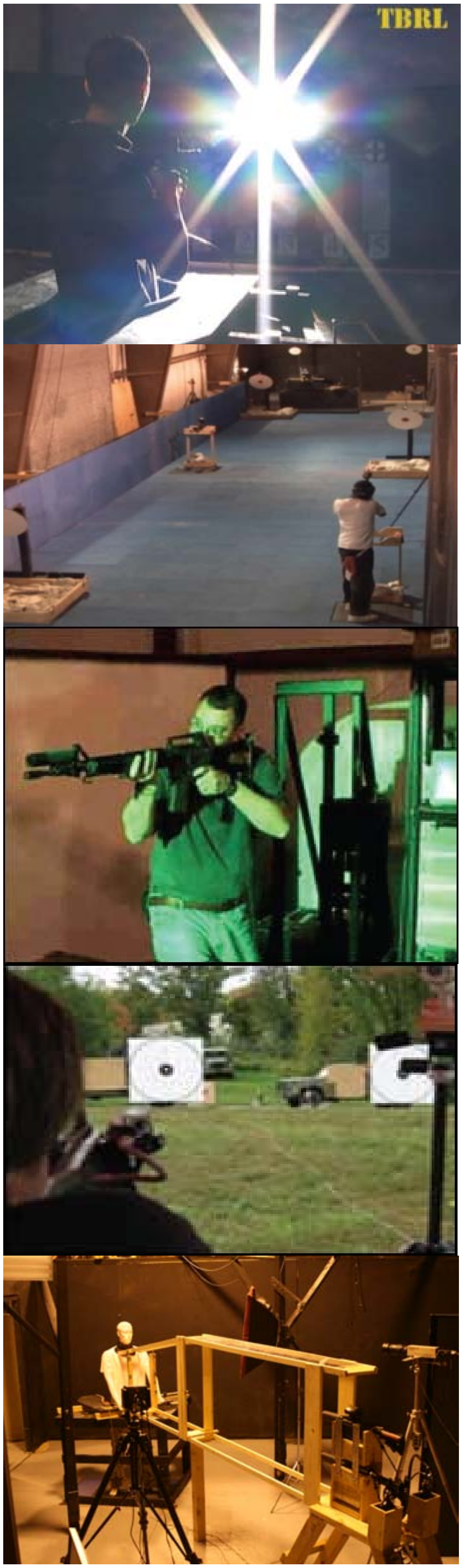
ABSTRACT: *This paper describes the work at the Target Behavioral Response Laboratory to develop data-based general approaches to modeling and simulation of human behavior.*

1. Introduction

The current theaters of operation have sharpened focus on analytics relevant to irregular warfare (National Research Council, 2011). A critical tool for operations research and systems analysts is the modeling and simulation of tactically relevant human behavior. There are several specific criticisms of the current state of the art. Most critically, there is recognition of the lack of real-life data to provide guidance for these M&S efforts. Moreover, also lacking are methods to assess how well these M&S efforts relate to actual real life human behaviors. One might propose that the lack of data on human behavior is caused by a lack of M&S researchers who are studying human behavior.

The Target Behavioral Response Laboratory (TBRL) is one such collection of scientists and engineers. TBRL's primary mission is to test the effectiveness of non-lethal weapons and systems, and has conducted experiments examining human behavioral response to a variety of stimuli relevant to non-lethal weapons (light, sound, blunt impact) (Cooke, Mezzacappa, Yagrich, & Riedener, 2010; Cooke et al, 2010; Mezzacappa, Cooke, & Yagrich, 2008; Short Riedener, & Cooke, 2010; Short Riedener, Cooke, & Minor 2010) and developed general methodologies by which data on actual human behavior in the laboratory serves as a basis for development of mathematical models describing human behavior (Mezzacappa, Cooke, Reid, DeMarco, Sheridan & Riedener, 2011). This presentation proposes facilitating close end-to-end collaborations between the laboratory and modeling researchers.





2. References

Cooke, G. ; Mezzacappa, E.; Yagrich, K.; Riedener, J (2010). Effects of lasers on driving. *Proceedings of the 2010 Directed Energies Professional Society Meeting*, 15-19 November 2010.

Cooke, G., et al. (2010). Topology and individual location of crowds as measures of effectiveness for non-lethal weapons. *Proceedings of the 27th Army Science Conference*.

Mezzacappa, E. S., Cooke, G., Yagrich, K. (2008). Network science and crowd behavior metrics. *Proceedings of the 26th Army Science Conference*.

Mezzacappa, E.; Cooke, G.; Reid, G.; DeMarco, R.; Sheridan, C.; Riedener, J. (2011). Mathematical capture of human crowd behavioral data for computational model building, verification and validation. *Proceedings of the 20th annual Behavior Representation in Modeling & Simulation (BRIMS) Conference*, March 21-24 2011, Sundance UT.

National Research Council. (2011). *Sociocultural data to accomplish Department of Defense Missions: Toward a unified social framework: Workshop summary*. Washington, D.C.: National Academy Press.

Short, K.; Riedener, J.; Cooke. G. (2010).Targeting of Convoy Vehicles is Not Disrupted by a Green Laser: Moving, Predictable Targets in Bright Lighting. *Proceedings of the 2010 Directed Energies Professional Society Meeting*, 15-19 November 2010.

Short, K.R.; Reid, G.; Cooke, G.; Minor, T. R. (2010). Can repeated painful blunt impact deter approach toward a goal? *Proceedings of the 27th Army Science Conference*.

Author Biographies

JOHN RIEDENER, MSSE, is the Technical Director of the TBRL.

ELIZABETH MEZZACAPPA, PHD, is a Principal Investigator at the TBRL.

Modeling Resilient Submarine Decision Making

Robert F. Stark, Michael P. Farry, Wayne A. Thornton, Arthur B. Wollocko
Charles River Analytics, Inc.
Cambridge, MA 02138
bstark@cra.com, mfarry@cra.com, wthornton@cra.com, awollocko@cra.com

David D. Woods, Alex M. Morison
Ohio State University
Columbus, OH 43210
woods.2@osu.edu, morison.6@osu.edu

Keywords:
Agent-Based Modeling, Resilience, Submarines, Decision Making

ABSTRACT: *Decision making, especially in high pressure environments such as submarines, is dependent on the use of a large amount of information. To aid decision makers dealing with increasing amounts of complexity and stress, computational models can be used to process information relevant to decisions. However, effective outcomes often rely on the decision maker's ability to mitigate risk. In risk management, different goals present their own constraints or risks that can be difficult for computational models to reason about. In particular, unforeseen circumstances present novel challenges to decision making, which can make brittle approaches fail. To address these challenges and introduce a greater degree of resilience, we present a novel agent-based approach in the context of submarine decision making.*

1. Introduction

A submarine captain is in charge of a crew in the Pacific Ocean. His submarine is ordered to be ready to launch at Tomahawk Land Attack Missiles (TLAM) strike package at a moment's notice. In the meantime, his mission is to search for enemy submarines and track them. These two missions have conflicting requirements, including depth, speed, location, stealth, time, and resources. These requirements need to be negotiated in a way that reduces the risk of mission failure or damage to your submarine or your crew. Furthermore, there is a large amount of data to be processed to enable objective, timely, and accurate decision-making during the execution of these missions. For example, sonar signals may indicate the presence of an enemy submarine, heat readings from the nuclear reactors onboard are needed to verify the submarine is in healthy operating condition, and communications from a superior officer may arrive at any time to change or modify your missions.

Unexpected events may also affect your missions. Even when risk-reducing decisions can be made, team-based decision making can fail under unforeseen circumstances (Woods & Branlat, 2010). For example, if the submarine becomes "wounded" from an event that creates a seawater leak, then it is noisier, less maneuverable, and slower. In such cases, the submarine commander needs to make quick and accurate decisions on how to modify his mission plans, or courses of action (COAs), to account for these changes. Developing a COA modification scheme from the beginning will make the human-computer team more *resilient* in the face of challenge events (Hollnagel, Woods, & Leveson, 2006).

Existing work has explored decision making in terms of risk management and constraint satisfaction. The Navy has codified risk management for its officers as Operational Risk Management (ORM; OPNAVINST 3500.39C), involving five steps: (1) Identify hazards; (2) Assess hazards; (3) Make risk decisions; (4) Implement controls; and (5) Supervise (watch for changes). ORM is generally only used for routine checkups and in-depth planning. During execution of tasks, on the other hand, the Navy has a simplified version of ORM called Time-Critical Risk Management (TCRM; OPNAVINST 3500.39C), which has only four steps following an "ABCD" mnemonic: (A) Assess the situation; (B) Balance your resources; (C) Communicate risks and intentions; and (D) Do and debrief (take action and monitor for change). We build on these efforts with an agent-based model that offers assistance in complex, knowledge-based decision making by introducing a critiquing agent.

2. Our Agent-Based Model

To support the TCRM process, the agent-based model starts with a three-step implementation of the Recognition-primed Decision Making (RPD) framework (Mulgund et al., 2000) with Resource Agents, Task Agents, and Risk Management. We added to this model with Resilient Critique based on cognitive walkthrough of actual COA replanning cases.

Resource and Task Agents were based on the first two steps of the RPD implementation and corresponds to the first step of TCRM. The Resource Agents process submarine and environmental information, while the Task Agents assess the situation based on Resource Agent values to determine if their constraints are satisfied. The Risk Management Agent corresponds to the second step of TCRM: this agent performs constraint satisfaction on the Task Agents to balance resources. To communicate risks and intentions, the Risk Management Agent suggests COAs to the human user.

The cognitive walkthrough of submarine decision making taught us that plans rarely work as expected on submarines. Therefore, we modified the three-part model to stay resilient in unexpected circumstances. In particular, we implemented the final TCRM step to include a critique that checks the generated COA for weaknesses and to stimulate replanning processes. The goal of this Resilient Critique Agent is to monitor the environment via Resource Agents compared to the constraints of active Task Agents. It then recommends re-planning to the Risk Management Agent, resulting in the sort of *meta-adaptation* that is necessary for resilience (Hollnagel et al., 2006).

We combined these agents into a multi-agent model (shown in Figure 1) to work together and assist commanders in making decisions. The Resilient Critique Agent is central to our structure so it can efficiently monitor all agents and provide feedback to the Risk Management Agent.

Figure 1: Diagram of our multi-agent model

For example, a Resource Agent that monitors depth may be used by a TLAM strike Task Agent to determine if conditions are right to launch the strike. The Risk Management Agent might compare the TLAM Strike agent with another Task Agent dedicated to ASW search and track to determine the most appropriate COA. If the submarine's depth is too deep for a TLAM strike, though, then the Resilient Critique Agent will notify the Risk Management Agent that its COA is no longer valid.

3. Discussion

Although we designed this model with submarine decision making in mind, it can also be applied to enable resilient decision making in other domains where multiple COA may be underway and handling unexpected events requires balancing multiple interactions between plans, goals and constraints such as occurs in disaster response. Initial feedback from subject matter experts (SMEs) has indicated that such a decision support system would help with COA generation.

However, because we do not expect the intelligent agents to be sufficiently resilient by themselves, we plan to build on this work with a user interface that enables users, such as submarine commanders, to specify constraints and explore COA options. We also plan to continue validating our intelligent agents and COA exploration approach in several ways. We plan to start by continuing our informal evaluations with SMEs through heuristic evaluations and cognitive walkthroughs. These informal evaluations will pave the way to formal user studies with representative users and, later, testing for operational effectiveness with real submarine users.

4. Acknowledgements

This material is based upon work supported by The United States Navy under Contract No. N00014-11-M-0297.

5. References

Hollnagel, E., Woods, D. D., & Leveson, N. (2006). *Resilience Engineering: Concepts and Precepts*. Altershot, UK: Ashgate.

Mulgund, S. S., Harper, K. A., Zacharias, G. L., & Menke, T. (2000). SAMPLE: Situation Awareness Model for Pilot-in-the-Loop Evaluation. In *Proceedings of 9th Conference on Computer Generated Forces and Behavioral Representation*. Orlando, FL.

Woods, D. D. & Branlat, M. (2010). Basic Patterns in How Adaptive Systems Fail. In E. Hollnagel, J. Paries, D. D. Woods, & J. Wreathall (Eds.), *Resilience Engineering in Practice*. Aldershot, UK: Ashgate.

NATO RTO HFM 209 - Emotion and attitude in constructive modelling

Dr. Wouter Lotens
Chairman HFM 209
TNO Human Factors,
Kampweg 5, PO Box 23, 3769 ZG Soesterberg, The Netherlands
+31 (0) 346 356 211
wouter.lotens@tno.nl

Mr. Brad Cain
Defence Research and Development Canada Toronto
Human System Integration Section, Learning and Training Group
1133 Sheppard Avenue West, Toronto, ON
M3K 2C9, Canada
+1-416-635-2195
brad.cain@drdc-rddc.gc.ca

Keywords: emotion, attitude, human modelling, constructive simulation, non-kinetic warfare

1. Introduction

Recent interest in non-kinetic warfare, aims to expand the options available to a military for achieving its strategic goals beyond those that have traditionally been employed. These non-kinetic alternatives often rely on psycho-social concepts that lack quantitative precision. The broadness of the decision making process, the considerations associated with a Three Block War (Krulak, 1999) concept and the requirement for up-to-date input data, make outcome prediction a scientific and practical challenge.

Further, modelling and simulation (M&S) are increasingly being applied to training and mission rehearsal in non-combat activities. Domains such as medical operating rooms, first responder triage exercises, interaction with non-combatants, critical event Crew Resource Management and team coordination, as well as collaboration among Other Government Departments (OGDs) and multinational command teams incorporating cultural differences require a different set of cues and responses than does conventional warfare modelling because interactions happen at the individual level, usually at close proximity.

The North Atlantic Treaty Organization (NATO) Research and Technology Organization (RTO) has considered a range of non-kinetic warfare concepts, including psychological, organizational and inter-cultural factors (Human Factors and Medicine (HFM) Research Technology Groups (RTGs) 120 and 140, Simulation, Analysis SAS 49), stress and psychological support (HFM 81 and 134), gender and child combatant issues (HFM 158 and 159), PSYOPS and I-OPS (HFM 160 and SAS 57) and exploration of serious games (MSG 37, 51, 55 and 59). Non-kinetic warfare is about enticing others to your perspective out of a belief that by accepting your fundamental principles they will be better positioned to satisfy their basic human needs (safety, shelter, food, recognition, development). Thus, non-kinetic warfare deals with a broader range of issues than the historical scope of classical warfare, involving many technological and human sciences disciplines.

Expansion of the existing body of knowledge on modelling of human behaviour representation through rational approaches from Artificial Intelligence to include emotional factors is seen as an important research area to better capture the human element. Incorporating models of emotions, attitudes and affective drivers are regarded as feasible, but the solution for current military M&S is not entirely obvious.

The development of the brain is related to different responses to stressful events. Emotion and emotional stress are integral elements in cognition and behaviour, playing roles in perception, learning, decision making and performance. It is widely assumed that rational decision making is the basis for most behaviour and that emotions lead to sub-optimal choices beyond cognitive control, however, there is increasing evidence that human behaviour is more complex than this simplistic perspective (Frank, Cohen & Sanfey, 2009). While it may be the case that severe emotional distress can lead to irrational behaviour (in the colloquial use of the term irrational), neurophysiological evidence suggests that even moderate levels of stress result in an interactive, coupled process between rational cognition and emotion-driven responses.

Current theories in decision making include two processes: an unconscious affective appraisal and an explicit, cognitive process (Picard, 2000; Sanfey, 2007). Emotion seems to also be driven by phases in the explicit process, sense making and judgment, reinforcing unconscious evaluations, and in periods of high time-stress, may be the only mechanism capable of responding in time (irrespective of the quality of the choice.) Interesting, the consumer research literature seems to indicate that what we think of as rational decision making (cognition) is more a rationalization of an affective choice. An extreme perspective is that all behaviour is driven by emotional responses to an imbalance in some goal or motivation. A considerable literature exists on the role of emotion and attitude in perception, judgement and actions, but little has been incorporated into applied human behaviour representation.

Human factors relevant to constructive simulation of conventional military operations have been researched and methods proposed for greater representation of human capabilities and limitations in constructive simulation in order to better predict or represent human behaviour. While the need for this added level of detail could be debated for models of force-on-force conflict, few doubt its need at the level of one-on-one or few-on-few interactions typical of today's Three Block War scenarios.

The RTG HFM-209 will assess the mainstream models of emotion, attitude and motivation for suitability and applicability to military simulation, then provide a scientific basis for including such non-traditional human modelling factors into what is expected to be an emerging trend in military modelling: that is, human-centric simulation within a "physics and phenomenon" framework compatible with more traditional military M&S approaches.

2. Objective

HFM 209 proposes to explore, test and make recommendations on modelling the effects of emotion and attitudes in constructive entities, particularly for Effects-Based Operations (EBO) models (Dompke, 2006). The methods focus on attitude changes and the associated consequences for security, including the 'hearts and minds' issue and concept of 'national popular support'.

Determination of the scientific validity of various emotional models is not the focus of the RTG; rather, the RTG will rely on the scientific literature for such guidance. The RTG will instead focus on the viability and validity (fitness for purpose) of the models for capturing the resultant behaviour from a dual-process, rational-irrational framework in a range of simulations for military applications.

3. Approach

Constructive modeling is an important element of option analysis, tactics development, training and, more frequently, mission rehearsal. HFM 128 and 143 (Cain et al., 2007; Lotens et al., 2009) promoted ways to lift constructive modeling from the traditional, organizational or force-on-force level to the individual human level where the understanding of cognition and human error is a greater challenge than weapon-effects modeling. This approach paves the way for non-kinetic actions as alternatives to the use of force in military M&S by adopting a goal directed behaviour approach consistent with Annett's Hierarchical Task Analysis (Annett, 2003; Annett & Duncan, 1967).

The current plan of work will begin with an assessment of current theories on the relationship between emotions, attitudes and culture, and their resultant effect on behaviour and decision making. Central to this assessment will be a determination of the elements of emotion and attitudes that are thought to be important in a non conventional, Three Block War (Krulak, 1999) context where EBO are likely to be applied. The assessment will look for common and unique model characteristics, and attempt to select a framework that seems compatible with current or envisioned constructive simulations.

Following the assessment, the RTG will propose and, if possible implement, a framework. The intent is to create a demonstration of the consequences of incorporating emotion and attitude in military simulation by applying it to a representative scenario. This is a challenging step for RTGs that will likely rely on concurrent national programs for implementation.

The RTG will consider how to apply the resulting framework to other RTO M&S research, such as stress and psychological support to own units (HFM 081 and 134), psychological methods to manage attitudes and terrorist behaviour (HFM 140 and 160, SAS 049), irregular warfare (SAS 071) and social sciences in the conduct of operations (SAS 074 and HFM 172). There is a growing body of quantitative research that may be exploited, both for model development and validation.

4. Way ahead

RTG HUM 209 has received approval to proceed and we are attempting to attract members interested in the domain to participate and define a formal programme of work. There is commitment from NLD and CAN with interest expressed from FRA, SWE and GBR, but more participation is needed, desired and welcome.

5. References

Annett, J. (2003). Hierarchical task analysis. In E. Hollnagel (Ed.), *Handbook of cognitive task design* (pp. 17-35): Lawrence Erlbaum.

Annett, J. & Duncan, K. D. (1967). Task analysis and training design. *Occupational Psychology, 41*.

Cain, B., Armstrong, J., Allender, L., Belyavin, A., Castor, M., Fraser, W., Gluck, K. A., et al. (2007). *Human Behaviour Representation in constructive modelling.* Paper presented at the HFM 143/RSM “Human Behaviour Representation in Constructive Modelling” Canadian Forces College, Toronto, Ontario, Canada.30-31 May. 165.

Dompke, U. (2006). *Modelling & Simulation Support to the Effects Based Approach to Operations – Observations from Using GAMMA in MNE 4.* Paper presented at the Transforming Training and Experimentation through Modelling and Simulation, RTO-MP-MSG-045 16.

Frank, M. J., Cohen, M. X. & Sanfey, A. G. (2009). Multiple systems in decision making. *Current Directions in Psychological Science, 18*(2), 73-77.

Krulak, C. C. (1999, January). The Strategic Corporal: Leadership in the Three Block War. *Marines Magazine*.

Lotens, W., Allender, L., Armstrong, J., Belyavin, A., Cain, B., Castor, M., Gluck, K. A., et al. (2009). *Human Behavior Representation In Constructive Simulation* (No. RTO-TR-HFM128): North Atlantic Treaty Organization (NATO), Research and Technology Organization (RTO) ,Human Factors and Medicine (HFM).

Picard, R. W. (2000). *Affective computing*. Cambridge, Massachusetts: Massachusetts Institute of Technology.

Sanfey, A. G. (2007). Decision Neuroscience: New directions in studies of judgment and decision making. *Current Directions in Psychological Science, 16*(3), 151-155.

D2P/CLS: A Tutor for Combat Lifesavers

J. Nicholas Hobbs
Frank E. Ritter
Jonathan H. Morgan
Applied Cognitive Science Laboratory
The Pennsylvania State University, University Park, PA
jnh5057@psu.edu, frank.ritter@psu.edu, jonathan.h.morgan@gmail.com

Keywords:
Tutor, Medicine, Combat Lifesavers, Battlefield Trauma

1. Introduction

Tactical Combat Casualty Care (TCCC) has been shown to save lives on the battlefield (NAEMT, 2010). These guidelines suggest that Combat Lifesavers (CLS), specially trained Marines operating at the platoon level, are pivotal for reducing causalities because they reduce the time between injury and care. These skills include both the assessment and the treatment of injuries, such as hemostasis, airway management, etc. Importantly, these skills require the proceduralization of interdependent time-sensitive skillsets.

To provide Combat Lifesavers with the opportunity to learn and maintain the skills necessary to implement TCCC techniques, we are starting to create a tutor called D2P/CLS. The D2P (Declarative to Procedural) architecture has been used previously in a marksmanship tutor (Hiam, Ritter, & Morgan, 2012; Ritter, Morgan, Hiam, & Kim, 2011). D2P uses iterative learning loops to introduce the material, followed by dynamic exercises that increase in complexity as the learner progresses through the tutor. We believe this approach will not only provide a training aid but also act as a form of sustainment training. The simulations within D2P/CLS will provide background information on the situation, present an injury to the learner, and present the learner with a visualization of the injury. After the learner has attempted to treat the injury, the D2P Interpreter will provide feedback to the learner to allow the learner to recognize mistakes.

2. Declarative To Procedural (D2P)

The design of the D2P tutoring architecture is focused on supporting the learner in learning declarative information and proceduralizing that information (Hiam, Ritter, & Morgan, 2012). This approach is based on a recent review of learning theory (Kim, Ritter, Koubek, in press). Figure 2.1 illustrates the implementation of this theory in D2P

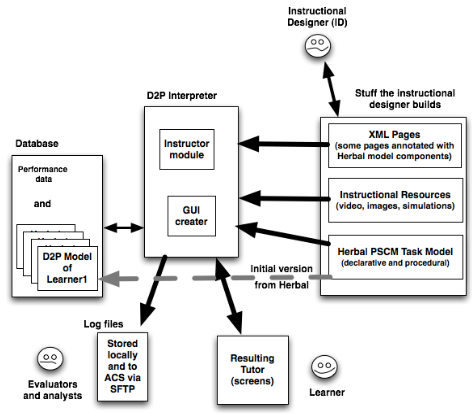


Figure 2.1. D2P implementation diagram

The D2P Architecture is designed around five major components: instructional design, the database, D2P interpreter, evaluation and analysis tools, and the user interface. The instructional designer provides content for the tutor by developing XML pages using provided templates and media. D2P allows us to easily incorporate insights from Subject Matter Experts (SMEs) into the tutor through the XML Pages. In addition, it supports the export of PDFs of the tutor to support iterative development by allowing SMEs to easily review the tutor's content.

D2P provides a tailored user experience by supporting output-based navigation; navigation occurs based on the success or failure of the learner throughout the course of the tutor. The database stores performance data and current models of the learners that are used to determine what content the learner needs to review. The D2P interpreter uses the content from the instructional designer and database to provide feedback to the users and evaluators. The resulting user interface appears from the D2P Interpreter by using the other tutor components, while the Herbal model will support a help function.

3. D2P/CLS: D2P applied to CLS tasks

To introduce the learner to CLS tasks, the tutor will provide the learner with a variety of combat-related situations. These situations will first introduce singular injuries to the learner and require the learner to assess the casualty and perform basic interventions. These introductory skills will be required throughout the remaining portions of the tutor and will allow the learner to identify where they are weak early on.

As the learner progresses through the tutor, the situations will become more difficult, and the system will introduce sessions where the patient has multiple and more complex injuries. The movement from basic to more complex situations provides the learner with the opportunity to not only gradually build their knowledge but also to learn basic task management.

The tutor will use content pages, media, and simulations to introduce basic assessment, injury types, and interventions, as well as care management. Using a mixture of content, practice, and results pages, we aim to develop a set of declarative and procedural skills that the learner can apply across multiple settings. The simulations will also move from conceptual to more realistic representations as the learner demonstrates mastery of TCCC concepts. Figure 3.1 shows an early conceptual representation.

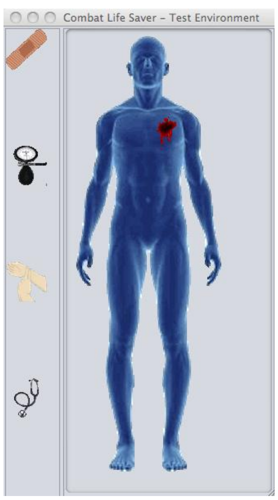


Figure 3.1. Initial draft D2P/CLS interface. Selectable tools are shown on the left.

4. Conclusions

D2P/CLS will provide an introduction on learning to become a Combat Lifesaver by teaching the identification and treatment of battlefield injuries. The D2P architecture facilitates the proceduralization of knowledge and allows for a customized user experience in the tutor. Learners will be taught using a combination of images, videos, text, tests, and simulations.

The complexity of the challenges the tutor will present the learner will increase as the learner progresses through the tutor. A potential end state scenario includes both declarative tests, as well as a realistic situation where the learner must correctly identify and treat multiple injuries. In summary, we hope the tutor will help save lives on the battlefield by teaching the required medical knowledge to learners in such a way that the information becomes proceduralized. We are currently developing D2P/CLS at Penn State, and will be testing it in conjunction with the Marine Warfighting Lab (MCWL) at Quantico, VA..

5. References

Hiam, J. W., Ritter, F. E., & Morgan, J. H. (2012) Declarative To Procedural Manual.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (in press). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science*.

NAEMT. (2010). *PHTLS: Prehospital trauma life support military edition* (Vol. 7 edition). St. Louis, MO: Mosby/JEMS.

Ritter, F. E., Morgan, J. H., Hiam, J.W., & Kim. J. W. (2011). The Moving Target Tutor (MTT)- Teaching the declarative knowledge to shoot moving targets (pp. 1 page handout).

6. Acknowledgements

D2P/CLS is developed with support from ONR through CRA (N091-086/P10008). The learning theory was developed with support from ONR Div 30 (N00014-11-1-0275/N00014-10-1-0401). D2P/CLS uses Herbal (N091-086/P10008 / N00014-06-1-0164). Ray Pursel, Paul Andre, Henry Warren, Keith Barry, and Art Jones, have served as SMEs. Mark Cohen has helped create part of the D2P system and is creating core parts of the CLS.

Author Biographies

J. NICHOLAS HOBBS is a researcher in the Applied Cognitive Science Laboratory at the Pennsylvania State University.

FRANK E. RITTER is a professor in the College of Information Sciences and Technology at the Pennsylvania State University.

JONATHAN H. MORGAN is a researcher at the College of IST's Applied Cognitive Science Lab with expertise in tutor design and organizational learning.

V. TUTORIALS

Papers are in order as they appear on the agenda.

Cognitive Systems and the Soar Architecture

Randy Jones (Senior Advisor on Strategy and Technology, Soar Technologies)

Full-Day Agenda:

- Cognitive systems and cognitive architecture overview
- History of Soar
- Soar overview
- Basic functionality in Soar
- Soar complex reasoning
- Use of impasses and sub-states in Soar
- Soar's learning mechanisms: Chunking, Reinforcement Learning, Semantic Memory, Episodic Memory

Abstract:

This tutorial provides an introduction to applied cognitive systems in general and the Soar cognitive architecture in particular. The discussion of applied cognitive systems will present the advantages of cognitive architectures over other software paradigms, as well as an introduction to how knowledge-based systems can be designed and engineered within a cognitive architecture. The remainder of the tutorial will focus specifically on the Soar cognitive architecture, including an overview and history of Soar, as well as hands-on exercises to learn how Soar implements knowledge-based decision making, structured and reactive goal management, and various forms of learning.

Network-Centric Simulation and Virtual Experimentation

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

Keywords:
Socio-Cultural Modeling, Model Validation

ABSTRACT: Based on a segment of an annual course offered at Carnegie Mellon, we will discuss virtual experimentation and model consumption. We will introduce Construct, a network-centric simulation tool that may be of interest to modelers focused on larger group behavior. Construct incorporates key insights from sociological literature. We will teach attendants how to critically consume model results and how to design experiments involving models. Attendees will be taught how to use Construct and will execute simple virtual experiments for Construct.

1. Introduction

Simulation is a very powerful method useful to people in many research areas. This tutorial is designed to provide attendants with skills in critically consuming simulation results and in the design of simulation experiments. We use Construct (Carley, 1991), a network-centric simulation of information diffusion, as an example to discuss these concepts. We will discuss Construct and place the simulation within a taxonomy of modeling methods. We will provide instruction in how to use Construct through a UI and also provide material for the advanced user on how to fully explore the range of options Construct offers.

1.1 Why is this material important?

As simulation methods become more common-place, it becomes more and more important for researchers in many fields, as well as policy makers, to be able to usefully consume simulation results. Being able to evaluate experimental design and to critically evaluate model assumptions will help researchers from diverse fields work together and improve the quality of simulations that are developed.

Construct, as a network-centric simulation tool, is an interesting perspective on organizational modeling. Like many of the modeling methodologies presented at BRiMS, Construct has individual actors with unique characteristics. Yet Construct actors exist within a larger context defined by their access to knowledge and resources and their connections with other actors. Modelers of many disciplines may be interested in examining the possibilities of integrating network-centric techniques into their work, especially as network data becomes easier to access and more understandable to model consumers.

1.2 Tutorial Purpose

The tutorial intends to educate attendants in how to critically evaluate simulation work and to educate them in the construction of simulation experiments. 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 simulations and simulation techniques. It provides a larger context on simulation that may be useful to these attendees. 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.

2. Contents

The tutorial has three sections, these are 1) Introduction to Construct, 2) Virtual Experimentation, and 3) Practical Construct Use through ORA.

The first section, *Introduction to Construct*, defines and describes Construct in a larger modeling context, describing the unique groups of features that Construct possesses and key concepts relevant to Construct from the simulation literature. It also describes the mechanics of Construct at a high level.

The second section, *Virtual Experimentation*, takes advantage of the first section by using Construct as an object for practical exercises. Following this exercise, we define virtual experimentation and discuss when such a technique is useful. We discuss, at a high-level, methods for parameter assignment.

The third section, *Practical Construct Use through ORA*, returns to Construct from an operational perspective. ORA, the Organizational Risk Analyzer, is an application which includes both best-of-breed network analytics as well as a simulation UI to create construct experiments. We will discuss how to use this interface and perform some simple experiments.

The remaining sub-sections describe each section in more detail.

2.1 An Introduction to Construct

The tutorial will begin by introducing and describing Construct. Construct is a turn and agent-based simulation useful for modeling information and belief diffusion. Each aspect of this previous definition will be defined and attendees will be encouraged to identify elements from their own experience that share these characteristics.

We approach modeling as a method of representation that removes those details not pertinent to the reason to model. A complete and accurate representation of an object would no longer be a model, but instead be the thing itself. As such, we present the attendee with the idea that models are common-place and everywhere, from recipes to toy cars to military simulations.

We describe Construct's turn mechanics, which are similar to Newell's decision cycle (Newell, 1990) and to Boyd's OODA loop (Boyd, 1987). We focus on the importance on perception as opposed to perfect apprehension of reality and the importance of bounded rationality to realistic modeling of human effort. We describe the importance of agency and also reference the importance of randomness in turn-based simulations to avoid accidental primacy among agents.

We also discuss two important human drives for socialization, based on the sociological literature: homophily (McPherson and Smith-Lovin, 1987) and expertise-seeking. Homophily is the preference for same, while expertise-seeking is the desire to interact with actors with rare knowledge. These two drives are more or less in action in many interesting modeling contexts.

From here, the tutorial will take a very brief break and then move on to the next section, Virtual Experimentation.

2.2 Virtual Experimentation

We begin this section by defining the term "Virtual Experiments". They are experiments operated upon a model. We remind ourselves that unlike experiments on reality, we are not discovering universal truths but instead what the model would predict. Models may be misapplied to problems that their assumptions poorly reflect. We keep the modifier "virtual" to remind us of this possibility. Finally, we remind the attendant of the useful maxim: "All models are wrong, but some are useful."

We suggest that virtual experiments are best used when testing reality would be unethical, infeasible, or incredibly expensive. Many real-world problems, particularly those involving people, exhibit these characteristics.

Next, the attendant is asked to think critically about Construct and its assumptions, based on the previous section of the tutorial. We introduce the term "model assumption" and note that such assumptions are integral to the model, as opposed to how the model was used in a particular instance. After the exercise, we identify these key assumptions for Construct.

Next, each attendant is asked to identify phenomena where these assumptions may not hold (or be useful). This exercise is used to promote a group discussion. We then remind the attendant that all models they produce or consume have their own set of assumptions, and that considering whether those assumptions are appropriate for the problem at hand is nearly always a useful exercise.

We then return to the larger issue of Virtual Experiments, identifying some key issues for discussion as they apply to simulation work, these are:

- Independent Variables
- Dependent Variables
- Method
- Control Conditions
- Generality
- Power

Independent variables, for simulation purposes, that are simulation parameters that we intend to manipulate over the course of the experiment. We encourage the attendant, for each independent variable, to ask themselves several questions, such as "Why am I changing this variable?" and "Why these values?". We also remind the attendant of the potential problem of combinatoric explosion.

Dependent variables, for simulation purposes, are the things that are being measured. We introduce the idea of construct validity in connection with dependent variables, suggesting that it is important that what they're measuring in the simulation should, in some way, relate to what they would want to measure in the real world.

Method is one of the largest differences between standard experiments in reality versus experiments involving simulation. In simulation, this is largely confined to setting other variables that must be set for simulation operation to an appropriate constant setting. We offer three strategies for setting these constants:

- No Impact – Set the variable so that the process it informs is non-operational, this is appropriate if the process is orthogonal to your particular purposes
- A reasonable base-line – This is an appropriate strategy when the process is clearly important to your experiment but not central. Ideally, previous work has found that this is a useful operational setting.
- Monte Carlo - a random distribution – If the relevant process is important to your phenomena but a single setting is difficult to defend, it may be all the user can do to define an appropriate range and distribution that the variable should fall into. As such, this variable will be randomly set for every instance of the simulation and provide useful noise.

Control conditions, in simulation work, are model settings where the process those conditions inform have very little effect on the phenomena of interest. For network-centric simulation, Erdos-Renyi networks (Erdös & Rėnyi, 1960) are often used as a control condition because these networks have very little inherent structure. Yet, Erdos-Renyi networks are very unlikely in human phenomena!

We discuss the issue of generality in the sense that all model parameters should be informed from literature, and should not be described merely in the context of their particular situation. A particular network-centric example is provided.

We also discuss the issue of power. Simulation, because it can be repeated as many times as the modeler has computer horsepower, will often find statistical significance in the comparison of even the most minute differences. We encourage the attendant to focus on trends in their simulation results, rather than specific percentage differences. The use of explicit percentages often encourages the model consumer to believe that those percentages are likely to be found as-is in the real-world as well.

The model section concludes with a discussion of validity. Face validity, simple validity, and docking are described.

A brief break separates this section from the next. The break will be primarily used to make sure that those interested in the final section have the software installed on their machine.

2.3 Practical Construct Use through ORA

This section is hands-on exercise using ORA to create simple Construct experiments. ORA is a Windows-only tool. Attendees will be provided the software at the start of the tutorial (if possible, asked to install it before the conference).

The tutorial will focus on how to start ORA and on how to load networks into ORA. These networks will be used for the Construct exercises.

Once the attendees have loaded the networks, we will move to the Simulation UI, which allows ORA users to access Construct.

ORA's construct UI will be discussed and an example will be demonstrated. Users will define a base case and alternatives for comparison. They will run Construct and view the results through ORA.

Advanced materials (Hirshman, Carley, & Kowalchuk, 2007a; Hirshman, Carley, & Kowalchuck, 2007b) on how to use XML to initialize ORA will be offered, but not discussed in depth in this section

3. Conclusion

As a powerful method, Simulation is useful to people in many research areas. We have designed this tutorial to provide attendants with skill in the critical evaluation of simulations and simulation experiments. We feel that this will be valuable both to simulation designers and simulation consumers.

Construct offers an interesting and different perspective on agent-oriented simulation with insights that may be useful to others, either as a stand-alone tool or technologies they may want to consider applying to their own tools.

4. References

Boyd, J (1987) A discourse on winning and losing. Air University, Maxwell Air Force Base

Carley, K. M. (1991). A theory of group stability. *American Sociological Review*. 56, 331-354.

Erdös, P., and Rėnyi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*. 5, 17-61.

McPherson, J. M., Smith-Lovin, L. (1987). Homophily in Voluntary Organizations: Status Distance and the Composition of Face-to-Face Groups. *American Sociological Review.* 52, 370-379.

Newell, A (1990) Unified theories of cognition. Harvard University Press, Cambridge

Hirshman, B., Carley, K.M., and Kowalchuck, M.J. (2007a). Loading networks in Construct. *Technical Report CMU-ISRI-07-116*. Pittsburgh: Carnegie Mellon University.

Hirshman, B., Carley, K.M., and Kowalchuck, M.J. (2007b). Specifying agents in Construct. *Technical Report CMU-ISRI-07-117*. Pittsburgh: Carnegie Mellon University.

Author Biographies

GEOFFREY P MORGAN is a PhD student in Carnegie Mellon's Computation, Organization, and Society program. Morgan has experience in developing agent-based models, cognitive models, and network-centric simulation. He is interested in organizational impacts on individual decision-making and on optimizing organizational performance.

KATHLEEN M CARLEY is 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.

Practical Aspects of Running Experiments with Human Participants

Frank E. Ritter (frank.ritter@psu.edu)
Jonathan H. Morgan (jhm5001@psu.edu)
College of Information Sciences and Technology
The Pennsylvania State University

Jong W. Kim (jong.kim@ucf.edu)
Department of Psychology
University of Central Florida

Keywords: Psychology experiments, human participants, ethics for experiments, experimental design

ABSTRACT: *There can often be a gap between theory and its implications for practice when gathering data on human behavior. This gap can be particularly significant outside psychology departments. While most students at the undergraduate or early graduate levels in psychology are taught how to design experiments and analyze data in courses in psychology and statistics, there is, unfortunately, a dearth of materials providing practical guidance for running experiments. In this tutorial we provide a summary of a practical guide for running experiments with human participants (Ritter, Kim, Morgan, & Carlson, in press).*

1. Introduction

There are many skills required for simulation. In addition to creating the simulations, you sometimes have to gather new data to help build the simulation. The knowledge about how to gather data through experimentation is partially covered in experimental design. How to analyze the data is covered in statistics courses and books.

On the other hand, the lack of materials on the details of running human experiments can lead to a gap between theory and practice, which is particularly acute outside of psychology departments where less studies are run. Consequently, labs developing simulations frequently must teach these practical skills to students informally because it is not included in their formal or informal education. Researchers in psychology often end up appalled by the lack of this common but undocumented sense when these studies are reported by researchers applying experimental psychology methods outside of psychology.

The details about how to run the studies themselves, how to interact with subjects, and the details are often learned solely through apprenticeship in a psychology or HCI lab. However, many researchers who are running or want to run studies do not have access to this tacit knowledge.

This tutorial provides a practical guide on the practical aspects of how to prepare and run experiments with human participants.

1.1 Why do we need a practical guide?

In general, scientific inquiries in the areas of human-computer interaction (HCI), human factors, cognitive psychology, and cognitive science involve human participants. One distinguishing factor of these disciplines, and thus experiments in these areas, has been the centrality of the human participant. Consequently, working in these areas requires not only understanding the theoretical and ethical issues for running human participants but also the practical aspects of the process itself. To start to frame this discussion, we are working to provide an overview of this process and related issues.

1.2 Purpose of this tutorial

In this tutorial we will present a summary of a practical guide (Ritter, Kim, & Morgan, 2009; Ritter, Kim, Morgan, & Carlson, in preparation) that can help people run experiments more effectively, more safely, and more comfortably for both the subjects and experimenters. Our purpose is to provide hands-on knowledge about experimental procedure.

We are generally speaking here from our background running cognitive psychology, cognitive ergonomics, and human-computer interaction (HCI) studies.

Because it is practical advice, we do not cover experimental design or data analyses and it may be less applicable in more distant areas.

1.3 Who is this tutorial useful for?

We believe that the tutorial will be useful to anyone who is starting to run research studies, training people to run studies, or studying the experimental process. Attendees will be provided copies of the written materials. A lecture and discussions will be used to present the material. If the conference allows it, the materials may be partly provided ahead of time for reading ahead. Particularly, the tutorial will be useful for students, teachers, lab managers, and researchers in industry. It will also be useful to computer scientists and other technologists who need to run empirical studies to gather data used to test and develop models.

2. Contents

This tutorial focuses on topics that are important for running repeatable studies. The tutorial will make the case for the importance of repeatable and valid experiments and some of the potential ethical issues that can arise. Table 1 notes several of the major components of studies explained in the larger report.

Table 1. Several important study components.

Component	Explanation
Scripting	What will be done with participants, writing it down in a script.
Missing subjects	How do you deal with subjects who do not show up?
Decorum	How do you dress and how do you address the participants?
Recruiting	How do you recruit a diverse yet representative set of participants without unwanted bias?
Literature	What literature should you read as background prep. for running a study?
Debriefing	How to debrief after a study session.
Payments	How to arrange payment for the participants, and the importance of getting this correct.
Piloting	The need to run pilot subjects to practice the method and also to find where the method (e.g., the script) needs to be modified.
Simulator studies	The role for simulated studies and how to treat model results as data.
Chances for insights	The need to keep your eyes and ears open for further insights while running studies.

In the tutorial we will address these topics, which we will briefly explain here to give a flavor of the material.

2.1 Repeatability and Validity

When running an experiment, insuring its repeatability and validity are of greatest importance, assuming the experiment is conducted ethically. Repeatability arises from running an experiment the same way for each participant. In addition, reducing unwanted variance in the participants' behavior is important as well. Ensuring this repeatability is partly the job of the research assistants (RAs), who often are not informed about these concepts and their practical application. Thus, RAs should strive to provide each participant with a consistent and comfortable but neutral testing experience.

Understanding how subjects will complete the task and working towards uniformity across all iterations of the procedure for each subject are important. The repeatability of the experiment is a necessary condition for scientific validity. There are, however, several well-known effects that can affect the experimental process. Chief among these is the experimenter's effect, or the influence of the experimenter's presence on the participants and how this effect can vary across experimenters. Depending upon the experimental context, the experimenter effect can lead to either increased or decreased performance. The magnitude and type of effect that can be attributed to this effect generally depends upon the type and extent of personal interaction between the participant and experimenter. Thus, you should strive to provide each participant with a comfortable but neutral testing experience.

Besides the experimenter effect, there are other risks to the experimental process. We highlight some here and illustrate how to avoid them, either directly or through proper randomization. Randomization is particularly important because you will most likely be responsible for implementing treatments, while understanding the other risks will help you take steps to minimize them. Finally, there are other experimental effects that are outside of your control—we do not cover these here. Even though you cannot eliminate all contingent events, you can note idiosyncrasies and with the principle investigator either correct them or report them as a potential problem.

Another common source of variation across trials is the effect of the experimental equipment. For instance, if you are having subjects interact with a computer or other fixed display, you should take modest steps to make sure that the participant's distance to the display is the same for each subject—

this does not mean, necessarily, putting up a tape measure, but in some cases, it does. It is necessary to be aware that the viewing distance can influence performance and in extreme cases can lead to blurred vision, irritated eyes, headache, and movement of torso and head (e.g., Rempel, Willms, Anshel, Jaschinski, & Sheedy, 2007). Furthermore, if subjects are picking up blocks or cards or other objects, the objects should either always be in the same positions, or they should be always randomly placed because some layouts of puzzles can make the puzzles much easier to solve. There will be other effects where variation in the apparatus can lead to unintended differences, and you should take advice locally to learn how to reduce them.

2.2 Ethics

There are several topics that you need to keep in mind when running subjects that we will cover. Chief among these are the ethics pertaining to the running of participants, and the gathering and reporting of data including published and unpublished documents. If you have any questions, you should contact the lead researcher (or principal investigator), or other resources at your university.

We would like to generalize the results to a wide population, indeed, the whole population. It is useful to recruit a representative population of subjects to accomplish this. It has been noted by some observers that experimenters do not always recruit from the whole population. In some studies, this is a justifiable approach to ensure reliability (for example, using a single sex in a hormonal study) or to protect subjects who are at greater risk because of the study (for example, non-caffeine users in a caffeine study).

Where there are not threats to validity, experimenters should take some care to include a representative population. This may mean putting up posters in a broad range of locations, and it may include paying attention to sex balance and even age balance in a study, and then correcting the balance by recruiting more subjects with these features. As a research assistant, you can be the first to notice this, and to bring it to the attention of the investigator, and help to address this.

It is necessary to avoid any procedures in a study that restrict participants' freedom of consent, that coerce their participation in a study. Some participants, including minors, patients, prisoners, and individuals who are cognitively impaired are more vulnerable to coercion. For example, enticed by the possibility of payments, minors might ask to participate in a study. If, however, they do so without parental consent, this is unethical because they are not old enough to give their consent—agreements by a minor are not legally binding.

Students are also vulnerable to exploitation. The grade economy presents difficulties, particularly for course where a lab component is integrated into the curriculum. In these cases, professors must not only offer an experiment relevant to the students' course-work but also offer alternatives to participating in the experiment.

To address these problems, it is necessary to identify potential conditions that would compromise the participants' freedom of choice. For instance, in the second example, recall that it was necessary for the professor to provide an alternative way to obtain credit. In addition, this means ensuring that no other form of social coercion has influenced the participants' choice to engage in the study. Teasing, taunts, jokes, inappropriate comments, or implicit quid pro quo arrangements are all inappropriate. These interactions can lead to hard feelings (that's why they are ethical problems!), and loss of good will towards experiments in general and you and your lab in particular.

When preparing to run the study, you should prepare how to deal with sensitive data as well. There are at least two issues here—data that you anticipate is sensitive and unanticipated data that arises that is sensitive. Data that is intrinsically sensitive should be handled carefully. Personal data is the most common. Information on an individual, such as related to race, creed, gender, gender preference, religion, friendships, and so on, must be protected. This data should not be lost or mislaid. It should not be shared with people not working on the project, either formally if you have an IRB that requires notice, or informally, if your IRB does not have this provision (this may occur more often outside of the US). You should seek advice from your colleagues about what practices are appropriate in your specific context. In some situations, you are not allowed to take data from the building, and in most cases, you are encouraged to back it up and keep the backed-up copy in another safe and secure location. In nearly all cases, anonymising data, that is, removing names and other ways data can be associated with a particular individual, removes most or all of the potential problems.

The second type of sensitive data is data that can arise where the subject's responses have implications outside of the scope of the study. This can include subjects implicating themselves in illegal activity, or unintentionally disclosing an otherwise hidden medical condition. For example, if you are administering caffeine, and you ask the subject what drugs they take (to avoid known caffeine agonists or antagonists), you may find information about illegal drug use. If you take a subject's heart rate or blood pressure measurements, you may discover symptoms of underlying disease.

You should know what to do in these cases before they arise. Generally, preparation for a study should involve discussions about how to handle sensitive data, and if there is a chance that the study may reveal sensitive data about the participants. You should fully understand how your institutions policies regarding sensitive data, and how to work with the subjects when sensitive information becomes an issue. If you have questions, you should ask the principle investigator.

2.3 Recruiting

Recruiting participants for your experiment can be a time consuming and potentially difficult task, but it is a very important procedure to produce meaningful data. An experimenter should thus carefully plan out with the lead researcher (or the principal investigator) to conduct successful participant recruitment for the research study. Ask yourself, "What are the important characteristics that my participants need to have?" Having a coherent reason for which participants are allowed or disallowed into your study is important.

First, it is necessary to decide a population of interest from which you would recruit participants. For example, if an experimenter wants to measure the learning effect of foreign language vocabulary, it is necessary to exclude participants who have prior knowledge of that language. On the other hand, if you are studying bi-lingualism you will need to recruit people who speak two languages. In addition, it may be necessary to consider age, educational background, gender, etc., to correctly choose the target population.

Second, it is necessary to decide how many participants you will recruit. The number of participants can affect your final results. The more participants you can recruit, the more reliable your results will be. However, limited resources (e.g., time, money) often force an experimenter to use the minimum number of participants. You may need to refer to previous studies to get some ideas of the number of participants, or may need to calculate the power of the sample size for the research study, if possible (most modern statistical books have a discussion on this, and teach you how to do this, e.g., Howell, 2008). Finally, you will upon occasion have to consider how many Ss are too many. It is believed to be the case, that running large number of subjects is both wasteful of time and effort, and also that the types of statistics that are typically used become less appropriate with large sample sizes. With large sample sizes effects that are either trivial or meaningless in a theoretical sense become significant (reliable) in a statistical sense. This is not a normal problem, but if you arrange to test a large class you might get close to this problem.

There are several ways that participants can be recruited. The simplest way is to use the experimenters, themselves. In simple vision studies, this is often done because the performance differences between people in these types of tasks is negligible and knowing the hypothesis to be tested does not influence performance. Thus, the results remain generalizable even with a small number of participants.

The next way that subjects can be recruited that we will consider is a sample of convenience. Samples of convenience consist of people who are accessible to the researcher. Many studies use this approach, so much so that this is not often mentioned. Generally for these studies, only the sampling size and some salient characteristics are noted that might possibly influence the participants' performance on the task. These factors might include age, major, sex, education level, and factors related to the study, such as nicotine use in a smoking study, or number of math courses in a tutoring study. There are often restrictions on how to recruit appropriately, so stay in touch with your advisor and/or IRB.

In studies using samples of convenience, try distributing an invitation email to a group mailing list (e.g., students in the psychology department or an engineering department) done with approval of the list manager and your advisor. These posts should be reasonably likely to appeal to the list members and they should be likely candidates (Cheyne & Ritter, 2001). Also, you can post recruitment flyers in a student board, or an advertisement in a student newspaper. Use efficiently all resources and channels that are available to you.

There are disadvantages to using a sample of convenience. Perhaps the largest is that the resulting sample is less likely to lead to generalizable results. The subjects you recruit are less likely to represent a sample from a larger population. Students who are subjects are different from students who are not subjects. To name just one feature, they are more likely to take a psychology class and end up in a subject pool. And, the sample itself might have hidden variability in it. The subjects you recruit from one method (an email to them) or from another method (poster) may be different. We also know that they differ over time—those that come early to fulfill a course requirement are more conscientious than those that come late. So, for sure, randomly assign subjects to the conditions in your study.

The largest and most carefully organized sampling group is a random sample. In this case, researchers randomly sample a given population by carefully applying sampling methodologies meant to ensure statistical validity and equal likelihood of selecting each potential subject. Asking students questions at a football game as they go in does not constitute a random sample—some students do not go (selection bias). Other methods such as selecting every 10th student based on a telephone number or ID introduce their own biases. For example, some students do not have a publicly available phone number, and some subpopulations register early to get their ID numbers. Truly choosing a random sample is difficult, and you should discuss how best to do this with your lead researcher.

One approach for recruiting participants is a subject pool. Subject pools are generally groups of undergraduates who are interested in learning about psychology through participation. Most Psychology departments organize and sponsor subject pools.

Subject pools offer a potential source of participants. You should discuss this as an option with your lead researcher, and where appropriate, learn how to fill out the requisite forms. If the students in the study are participating for credit, you need to be particularly careful with recording who participated because the students' participation and the proof of that participation represent part of their grade.

The theory is that participating in a study provides additional knowledge about how studies are run, and provides the participant with additional knowledge about a particular study. The researchers, in turn, receive access to a pool of potential subjects.

2.4 The Related Literature

This short document does not assume that you have a background in statistics or have studied experimental design. To help run a study you often do not need to know these areas (but they do help!). If you need help in these areas, there are other materials that will prepare you to design experiments and analyze experimental data. In addition, most graduate programs with concentrations in HCI, cognitive science, or human factors engineering feature coursework that will help you become proficient in these topics.

Many introductory courses in statistics, however, focus primarily on introducing the basics of regression and ANOVA. These tools are unsuitable for many studies analyzing human data where the data is qualitative or sequential. Care, therefore, must be taken to design an experiment that collects the proper kinds of data. If ANOVA and regression are the only tools at your disposal, we recommend that you find a course focusing on the design of experiments featuring human participants, as well as the analysis of human data. We also recommend that you gather data that can be used in a regression because it can be used to make stronger predictions, not just that a factor influences a measure, but in what direction (!) and by how much.

So, it is generally useful to have read in the area in which you are running experiments. This reading will provide you with further context for your work, including discussions about methods, types of subjects, and pitfalls you may encounter. For example, the authors of one of our favorite studies, an analysis of animal movements, note that data collection had to be suspended after having been chased by elephants! If there are elephants in your domain, it is useful to know about them. There are, of course, less dramatic problems such as common mistakes subjects make, correlations in stimuli, self-selection biases in a subject population, power outages, printing problems, or fewer participants than expected. While there are reasons to be blind to the hypothesis being tested by the experiment (that is, you do not know what treatment or group the subject is in that you are interacting with, so that you do not implicitly or inadvertently coach the subjects to perform in the expected way), if there are elephants, good experimenters know about them, and prepared research assistants particularly want to know about them!

As a result, the reading list for any particular experiment is both important and varies. You should talk to other experimenters, as well as the lead researcher about what you should read as preparation for running or helping run a study.

2.5 Piloting

Conducting a pilot study based on the script of the research study is important. Piloting can help you determine whether your experimental design will successfully produce answers to your inquiries. If any revision to the study is necessary, it is far better to find it and correct it before running multiple subjects, particularly when access to subjects is limited. It is, therefore, helpful to think of designing experiments as an iterative process characterized by a cycle of design, testing, and redesign. In addition, you are likely to find that this process works in parallel with other experiments, and may be informed by them (e.g., lessons learned from ongoing related lab work).

Thus, we highly recommend that you use pilot studies to test your written protocols (e.g., instructions for experimenters). The pilot phase provides experimenters with the opportunity to test the written protocols with practice participants, and is important for ironing out misunderstandings, discovering problematic features of the testing equipment, and identifying other conditions that might influence the participants. Revisions are a normal part of the process; you should not hesitate to revise your protocols. This will save time later. There is also an art to knowing when not to change the protocol. Your principle investigator can help judge this!

It is also useful at this stage to write the method section of your paper. Not only is your memory much fresher but also you can show other researchers your method section and receive suggestions from them before you run the study, a good time to get suggestions. These suggestions can save you a lot of time, in that these reviews essentially constitute another way of piloting the study.

2.6 Chance for insights

Gathering data directly can be tedious, but it can also be very useful and inspiring. Gathering data gives you a chance to obtain insights about aspects of behavior that are not usually recorded, such as the user's questions, their postures, and their emotional responses to the task.

Obtaining these kinds of insights and the intuition that follows from these experiences is important for everyone, but gathering data is particularly important for young scientists. It gives them a chance to see how previous data has been collected and how studies work. Reading will not provide you with this background or the insights associated with it; rather this knowledge only comes from observing the similarities and differences that arise across multiple subjects in an experiment.

So, be engaged as you run your study and then perform the analysis. These experiences can be a source for later ideas, even if you are doing what appears to be a mundane task. In addition, being vigilant can reduce the number and severity of problems that you and the lead investigator will encounter. Often, these problems may be due to changes in the instrument, or changes due to external events. For example, current events may change word frequencies for a study on reading. Currently, words such as bank, stocks, and mortgagees are very common, whereas these words were less prevalent a few years ago.

3. Conclusions

Once a science is mature enough practitioners will know the methods, while a science is growing, the method will have to be more explicitly taught. While a method is moving between areas, such as behavioral studies moving from psychology to computer science and engineering, the method will need to be made more explicit, and it can be useful for a method to become more explicit.

In our tutorial we will provide practical advice regarding the important and basic inquiry of how to run an experiment with human participants. We are working on extending and polishing a written guide that will be useful to anyone who is starting to run research studies, training people to run studies, or studying the experimental process (Ritter, Kim, Morgan, & Carlson, in press, 2012). This tutorial and guide will particularly help researchers in industry and government in addition to students who are not in large departments or who are running participants in departments that do not have a large or long history of experimental studies of human behavior.

Currently, the report is in use at eight universities in the US, Canada, and England for graduate and advanced undergraduate courses in cognitive science, human factors, information science, and in human-computer interaction courses.

As a colleague noted, this contains just common sense. We have found that this common sense is not so common, and that new researchers, both students and those taking up a new methodology, need a good dose of common sense, and that researchers who have many fields to master as in simulation and training will need and can use help in learning the tacit knowledge in psychology about how to gather human data.

4. Acknowledgements

Developing the report of practical advice on running studies was sponsored by ONR (W911QY-07-01-0004, N00014-10-1-0401, and N00014-11-1-0275).

5. References

Cheyne, T., & Ritter, F. E. (2001). Targeting respondents on the Internet successfully and responsibly. *Communications of the ACM, 44*(4), 94-98.

Rempel, D., Willms, K., Anshel, J., Jaschinski, W., &

Sheedy, J. (2007). The effects of visual display distance on eye accommodation, head posture, and vision and neck symptoms. *Human Factors, 49*(5), 830-838.

Ritter, F. E., Kim, J. W., Morgan, J. H., & Carlson, R. A. (in press, 2012). *How to run experiments: A practical guide to research with human participants*. Currently 112 pages. Thousand Oaks, CA: Sage.

Author Biographies

FRANK RITTER is on the faculty of the College of IST, an interdisciplinary academic unit at Penn State to study how people process information using technology. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Human Factors, Cognitive Systems Research,* and *IEEE SMC IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans.*

JONATHAN H. MORGAN is a researcher at the College of Information Sciences and Technology's Applied Cognitive Science Lab with expertise in modeling social interactions and social moderators, including the moderating effect that relative spatial positions have on group performance.

JONG W. KIM is a researcher at the University of Central Florida. His research interests are in learning and forgetting of skill. He received a PhD in Industrial Engineering in 2008 from Penn State.