



AFRL-RH-WP-TR-2014-0094

**ROBUST DECISION MAKING FOR IMPROVED MISSION
ASSURANCE**

Kevin Gluck

United States Air Force

**June 2014
Final Report**

Distribution A. Approved for public release; distribution unlimited. (Approval given by 88 ABW/PA, 88ABW-2014-4086, 28 Aug 2014.)

**AIR FORCE RESEARCH LABORATORY 711TH
HUMAN PERFORMANCE WING, HUMAN
EFFECTIVENESS DIRECTORATE,
WRIGHT-PATTERSON AFB, OH 45433
AIR FORCE MATERIEL COMMAND UNITED
STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RH-WP-TR-2014-0094 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signature//
KEVIN GLUCK, Work Unit Manager

//signature//
MERRICE SPENCER, Col, USAF
711 HPW/RHA
Human Effectiveness Directorate
711 Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show “//signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YY) June 2014		2. REPORT TYPE Final		3. DATES COVERED (From - To) 17 Aug 2011 – 30 Sep 2013	
4. TITLE AND SUBTITLE Robust Decision Making for Improved Mission Assurance				5a. CONTRACT NUMBER In-House	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Kevin Gluck				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER H00E/1123AC03	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Materiel Command Air Force Research Laboratory 711 th Human Performance Wing Human Effectiveness Directorate RHA Division RHAC Branch Wright-Patterson AFB, OH 45433				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command Air Force Research Laboratory 711 th Human Performance Wing Human Effectiveness Directorate RHA Division RHAC Branch Wright-Patterson AFB, OH 45433				10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL/RHXX	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-RH-WP-TR-2014-0094	
12. DISTRIBUTION/AVAILABILITY STATEMENT Distribution A. Approved for public release; distribution unlimited. (Approval given by 88 ABW/PA, 88ABW-2014-4086, 28 Aug 2014.)					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The Robust Decision Making (RDM) Strategic Technology Team (STT) proposed and was approved to receive funding for a set of four research projects advancing foundational decision science and technology over a three year period of performance. At the time it was approved, the initiative involved 27 collaborating scientists and engineers from five technical directorates and AFIT. They brought expertise spanning cognitive science, industrial/organizational and experimental psychology, computer science, electrical and computer engineering, mathematics, statistics, and human factors. A common thread across these diverse perspectives was a view of current and future missions as enabled by integrated human-machine cyber-physical decision systems. The vision was improved mission assurance, made possible through increased robustness in decision making processes and better decision outcomes, despite the pressures and perturbations that are present in contested environments.					
15. SUBJECT TERMS Robust Decision Making; decision making processes					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT: SAR	18. NUMBER OF PAGES 189	19a. NAME OF RESPONSIBLE PERSON (Monitor) Kevin Gluck 19b. TELEPHONE NUMBER (Include Area Code) 937-938-3552
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			

Contents

- 1. Abstract..... 5
- 2. INTRODUCTION..... 5
- 3. OBJECTIVES. 5
- 4. AIR FORCE BENEFIT..... 6
- 5. AFRL BENEFIT..... 6
- 6. PROGRAM..... 7
- 6.1 Modeling Sense-Making as Abduction-Based Inquiry in a Cyber Effects Testbed 7
- 6.2 Saliency, Expertise, and Intuitive and Analytical Reasoning in the use of Decision Support Tools 7
- 6.3 Context Switching Methods to Calibrate Trust for Mission Assurance 7
- 6.4 Remotely Piloted Aircraft (RPA) Testbed for Research on Robust Asset Management and Decision Making..... 7
- 7. References 9
- Appendix A. 10
- Appendix B.: Abduction’s Role in Reverse Engineering Software 16
- Appendix C. Saliency, Expertise, and Intuitive and Analytical Reasoning in the use of Decision Support Tools 22
- Appendix D. RDM STT: Context Switching Methods to Calibrate Trust for Mission Assurance 61
- Appendix E. Sensors Directorate Technologies for Robust Decision Making for Improved Mission Assurance 63
- Appendix F. RDM STT INITIATIVE: ROBUST DECISION MAKING FOR IMPROVED MISSION ASSURANCE 135
- Appendix G. Remotely Piloted Aircraft (RPA) Testbed for Research on Robust Asset Management and Decision Making 164

1. Abstract.

The Robust Decision Making (RDM) Strategic Technology Team (STT) proposed and was approved to receive funding for a set of four research projects advancing foundational decision science and technology over a three year period of performance. At the time it was approved, the initiative involved 27 collaborating scientists and engineers from five technical directorates and AFIT. They brought expertise spanning cognitive science, industrial/organizational and experimental psychology, computer science, electrical and computer engineering, mathematics, statistics, and human factors. A common thread across these diverse perspectives was a view of current and future missions as enabled by integrated human-machine cyber-physical decision systems. The vision was improved mission assurance, made possible through increased robustness in decision making processes and better decision outcomes, despite the pressures and perturbations that are present in contested environments.

2. INTRODUCTION.

As he neared the end of his tenure as Commander of AFRL, MajGen Bedke shared this observation with the laboratory in one of his weekly What I'm Thinking (WIT) emails:

“In the 21st Century, the Really Tough Problems are simply too complex and too integrated to be able to solve within any one Technical Directorate. The TDs need to be able to work together to solve these mega-problems.” (24 January 2010)

Demonstrable improvement in the quality and robustness of decision processes and outcomes, particularly in complex, uncertain, dynamic, and time constrained environments like those encountered in Air Force operations, is a “Really Tough Problem” that warrants this multi-disciplinary, cross-TD investment. Indeed, it is exactly the sort of challenge the Strategic Technology Teams (STTs), which existed in AFRL from about 2006 to 2012, were intended to address. Today’s military missions involve people, vehicles, and computers that are all information processing components of integrated cyber-physical decision systems. The humans and machines in these systems engage in decision making processes that result in a decision to act. Military decisions (actions) have consequences for safety, security, and survivability. It is important that we establish a sustained investment in decision science and technology that will transform the ways we assure the accomplishment of our military missions.

Mission assurance is achieved through robustness in military decision making processes, where robustness is resilience in the face of dynamic environmental pressures, such as degradations in components of the decision system, increases in uncertainty, or time pressure. In the Robust Decision Making team, we envision a future in which achieving the mission is assured despite such perturbations because the process of making military decisions is robust to them. Where optimality is provable, we endeavor to optimize the process of decision making and the goodness of the decision outcome. Where optimality is not tractably achievable, we maximize the probability of mission success with novel, integrative decision methods, the foundations for which was to be provided by the research conducted under the umbrella of this initiative.

3. OBJECTIVES.

Our general objective is fundamental scientific and technological advancements that provide the foundation for future technology options which create for the Air Force the Robust Decision Making future described in the previous section. The specific objectives span a naturally and desirably heterogeneous set of four research projects. Their objectives are:

1. Empirical study and computational process modeling to establish a scientific understanding of abductive reasoning for sensemaking, implemented in the context of software protection reverse engineering, and evaluated with a cyber effects testbed in the C2 Wind Tunnel.

2. Systematic investigation of the factors that determine the effectiveness of decision support tools, with a focus on (a) analytical versus intuitive decision processes, (b) saliency of underlying task structures, and (c) expertise level of the decision maker.
3. Broaden scientific understanding of the roles and dynamics of context, trust and systemic portrayal biases in complex decision environments and develop and evaluate strategies such as “knowledge glyphs” for optimizing decision robustness through communication and portrayal technologies that capitalize upon this understanding.
4. Develop, demonstrate, and evaluate a unique Integrated System Readiness Monitoring testbed that will be used to discover new methods, metrics, and models for quantitative measurement-based mission-level asset management and to provide support for tactical and operational decisions regarding overall mission system readiness and adaptation.

4. AIR FORCE BENEFIT.

Decision making pervades every stage of Air Force and joint missions. These decisions are increasingly complex, cognitively demanding, and consequential as a result of high uncertainty, urgency, and the rapidly changing nature of the joint fight. Massive amounts of relevant information are now available from disparate and powerful biological and artificial sensory systems to inform these decisions; however, the task of quickly extracting and understanding the relevant actionable knowledge from this overwhelming flow of information is daunting, especially given the severe time pressure for making decisions, the dynamic nature of the battlefield environment, and the complexity of dealing with distributed or net-centric team decision tasks. To complicate matters further, it is increasingly clear that the traditional boundaries between human and machine roles are disappearing. The future vision of integrated human-machine decision systems is already upon us. Hence, there is escalating pressure on AFRL researchers to better understand the basic science of mixed human – machine decision making, and make use of this science to develop increasingly automated knowledge-extraction tools and intelligent machine-based decision aids that help optimize, speed up, and adaptively adjust inference, prediction, and decision processes in order to adequately inform human decision makers in the loop. This need is conveyed in Air Force requirements documents and the recent reports of scientific groups who have examined potential technical solutions. Such reports include the 2004 report of the Air Force Scientific Advisory Board (SAB), *Human-System Integration in Air Force Weapon Systems Development and Acquisition*; portions of the 2006 report of the National Research Council, *Basic Research in Information Science and Technology for Air Force Needs*; and even more recently, the 2008 Air Force SAB study on *Defending and Operating in a Contested Cyber Domain*. In their 2008 study, the SAB called for an emphasis on mission assurance, and recommended the development of agent-based models for modeling and simulation (M&S)-based training, as well as a new emphasis on the fundamentals of human-computer interaction (HCI) when dealing with compromised, or perceived-to-be-compromised, cyber systems (pp. 60-62). This RDM STT directly addressed these scientific and technological challenges through a coordinated, collaborative, multi-disciplinary research effort.

5. AFRL BENEFIT.

The necessity and ubiquity of decision making in all military operations and in our daily life suggest the need for a scientific and technological foundation for new decision technologies that will be well positioned for maturation and transition to a wide assortment of domains. Establishing this research foundation is consistent with AFRL’s position at the beginning of the acquisition process and AFRL’s role as an engine of discovery and innovation in important, relevant research areas. Beyond the solid positioning of the RDM initiative’s research agenda in the core of the AFRL mission, the research proposed here also is responsive to several recommendations from AFRL’s Scientific Advisory Board. The SAB’s report from their most recent review of the lab’s portfolio included AFRL-level recommendations to:

1. Increase investments in basic and early applied research, balancing relevance across air, space, and cyber
2. Explicitly integrate consideration of threats and countermeasures in all S&T planning/execution
 - Evaluate S&T approaches and products in terms of robustness/resiliency as well as performance
4. Increase engagement across CTCs, FLTCs, and AFRL Directorates
7. Create and implement a research portfolio addressing deficiencies in the validation and certification of software controlling complex adaptive systems
 - Address V&V needs for large-scale, non-deterministic systems that include significant human-system interactions

The RDM STT addressed these four of the SAB's seven recommendations.

6. PROGRAM.

The research program included four research projects, involving a coordinated set of related research questions addressed concurrently. All four research projects involve cross-TD, multi-disciplinary collaborations, with in-house government scientists and engineers serving as principal investigators. In total, the RDM STT spanned five Directorates (RB, RH, RI, RX, RY) and AFIT.

Here are the four research lines within the RDM STT, with the project lead and initial set of performing scientists and engineers identified:

6.1 Modeling Sense-Making as Abduction-Based Inquiry in a Cyber Effects Testbed

Project Lead: Scott Douglass, RHAC
Co-PIs: Shelby Barrett, RISB; Adam Bryant, RYTA; Tim Busch, RISB;
 Dawn Trevisani, RISB; Kirk Weigand, RYTC

Summarized in: Appendix A, Appendix B

6.2 Saliency, Expertise, and Intuitive and Analytical Reasoning in the use of Decision Support Tools

Project Lead: John Salerno, RIEF
Co-PIs: Dick Deckro, AFIT; Warren Geiler, RIEF; Robert Patterson, RHAE;
 Byron Pierce, RHAE; Matthew Robbins, AFIT

Summarized in: Appendix C

6.3 Context Switching Methods to Calibrate Trust for Mission Assurance

Project Lead: Kirk Weigand, RYTC
Co-PIs: Oscar Garcia, RHAS; Felicia Harlow, RYTC; Joseph Lyons, RHXS;
 Mike Manno, RIEF; Charlene Stokes, RHXS; Gina Thomas, RHCP

Summarized in: Appendix D, Appendix E, Appendix F

6.4 Remotely Piloted Aircraft (RPA) Testbed for Research on Robust Asset Management and Decision Making

Project Lead: Mark Derriso, RBSI
Co-PIs: James Christensen, RHCP; Justin Estep, RHCP; Kevin Gluck, RHAC;

Mike Grimaila, AFIT; Raymond Holsapple, RBCA; Roman Ilin, RYHE; Jeremy Knopp, RXLP;
Leonid Perlovsky, RYHE

Summarized in Appendix G

Unfortunately, after years of cross-directorate planning and preparation costing hundreds of thousands of dollars in this STT alone, AFRL chose to eliminate most of the STTs prematurely. This decision was not made due to technical performance problems within this or the other STTs. It was purely a financial convenience, on the heels of an over-committal of Section 219 funds by the AFRL Corporate Board. Ironically, theirs was one of the least robust decisions that could have been made. It created enormous frustration and waste. Despite the management perturbation, we managed to remain focused on extracting as much value as possible out of what was left of our funding and period of performance. The content of the Appendices describes our advancements and, in some cases, remaining challenges.

Something not represented in the content of the Appendices is an investigation undertaken by the STT Chair into the nature of robustness and development of a method for quantifying robustness and stability, resulting in the following publications:

7. References

- Gluck, K. A., McNamara, J. M., Brighton, H., Dayan, P., Kareev, Y., Krause, J., Kurzban, R., Selten, R., Stevens, J. R., Voelkl, B., & Wimsatt, W. C. (2012). Robustness in a variable environment. In J. R. Stevens & P. Hammerstein (Eds.) *Evolution and the Mechanisms of Decision Making* (pp. 195-214). Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press.
- Walsh, M. W., Einstein, E. H., & Gluck, K. A. (2013). A quantification of robustness. *Journal of Applied Research in Memory and Cognition*, 2, 137-148.
- Walsh, M. W., & Gluck, K. A. (in press). Mechanisms for robust cognition. *Cognitive Science*.

Appendix A.

RDM-STT Sub Project Summary

Sub-Project Title: “Modeling Sense-Making as Abduction-Based Inquiry”

Sub-Project Contributors:

- RH: Adam Bryant, Scott Douglass, Jo-Ann Hamilton, Jeff Luehrs, Veda Setlur, Rachel Taylor
- RI: Shelby Barrett, Tim Busch, Tim Lebo
- RY: Ron Hartung, Kirk Weigand
-

Research Question

This sub-project set out to understand, model, and simulate sensemaking and follow-on decision making in tractable mission-relevant tasks. The research focused on modeling sensemaking as a type of abduction-based inquiry. Specifically, this research studied and modeled the abduction-based inquiry process underlying a series of simple inference tasks and aspects of reverse engineering tool use.

The overall scientific question motivating this research was, “What are the cognitive processes used by reverse engineers endeavoring to understand and circumvent software protection schemes?” In order to answer this question, this research effort employed a series of case studies and experiments to ask a series of more focused research questions:

- How do individuals reason from data to likely explanations when they are attempting to identify individuals?
- How do analysts reason from data to likely explanations when they are attempting to assess the presence of counter insurgents in fictional medium-sized towns?
- How do reverse engineers use debuggers, disassemblers, and inference to make sense of software executables.

Air Force Relevance

Air Force Benefits

One expected outcome of this research is that the empirical and computational cognitive models produced during the research will inform the development of tools that minimize the amount of time reverse engineers need to understand and break software protections. Through this research, better systems interfaces might be developed to allow reverse engineers to be trained faster in order to more quickly enhance the cyber security work force of the Department of Defense.

AFRL Benefits

To the extent that the domain-specific languages (DSLs) developed during this research capture the cognitive/computational foundations of sensemaking in an executable formalism, it will support cross-directorate sensemaking and decision making modeling and simulation efforts in effects test-beds such as the C2 Wind Tunnel (Neema, Hemingway, Green, Sztipanovits, Karsai, 2009)

Model specification and execution capabilities developed in this research will help future AF cognitive scientists incorporate cognitive models of sensemaking into broader models of decision making and software engineering paradigms such as Model Integrated Computing (MIC). Modeling frameworks

supporting MIC such as the Generic Modeling Environment (GME) have a long history of AFRL support (Sztipanovits, Karsai, & Ledeczi, 2002; Balogh, Neema, Hemingway, Williams, Sztipanovits, & Karsai, 2008). MIC/GME-based projects such as “C2 Wind Tunnel - Human Centric Design Environments for Command and Control Systems” and “Resilient Architectures for Integrated C2 in a Contested Cyber Environment” are currently being supported by AFOSR and AFRL. This research is relevant and important in the context of such on-going AFOSR-sponsored MIC/GME research because it:

- Leveraged the findings and products of related AFRL supported MIC/GME research.
- Generated products that can be interoperated into AFRL MIC/GME systems.
- Introduced new cognitive/behavioral modeling capabilities to MIC.

S&T Background

This research examined, modeled, and simulated the way individuals: (1) make sense of situations; and (2) consequently make better decisions. This research investigated sensemaking in task contexts requiring people to incrementally develop and refining competing hypotheses about information and decisions about possible actions. Conventional decision theory cannot be used to study this type of decision and action. To study decision and action in these circumstances, additional knowledge about the interleaving of situational assessment and decision making processes is required.

Sensemaking is a process through which people attempt to understand complex and ambiguous situations so that they can make reasonable decisions and act effectively (Klein, G., Phillips, J., Rall, E., & Peluso, D., 2007). The unifying core of this research is a conceptualization in which sensemaking is defined as *abduction-based inquiry*. Abduction can be thought of as a type of inference that plays a role in a process through which inquiry reduces doubt (Schvaneveldt & Cohen, in press; Fann, 1970). Figure 1 below illustrates the concepts and relations that underpin the conceptualization of sensemaking as abduction-based inquiry.

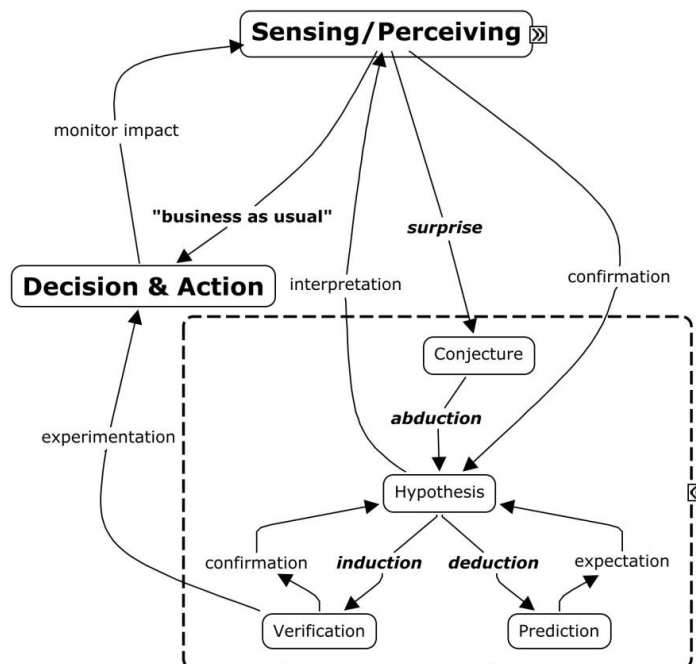


Figure 1: Central concepts, relations and constraints in model of sensemaking as abduction-based inquiry.

After a person assesses and understands the context in which they are trying to act effectively, they either: (a) find that it's "business as usual" and act according to routine; or (b) are surprised by unexpected observations and try to make sense of things through inquiry. A surprised person uses *abduction*, a type of reasoning from observations to likely explanations or causes, to generate new ideas (hypotheses) about their situation. Through *deduction* and *induction*, these hypotheses are expanded and confirmed. If necessary, follow-on actions can refine and evaluate explanations and hypotheses. This research empirically investigated the details of the diagrammatic model in Figure 1 and captured the underlying cognitive and computational processes in a *domain-specific language* (DSL). This DSL was then used by contributing researchers to formally specify and simulate models of sensemaking.

Summary of Results

The research project was divided into 4 sub-projects that were executed by collaborating researchers from RH, RY, and RI. The sub-projects were:

[1] An empirical sub-project that used structured interviews, cognitive task analyses, observational studies, protocol analyses, empirically manipulated task variations, and follow-on case studies to understand how humans reverse engineer software protections.

In this sub-project, skilled individuals reverse engineering software protections were observed. Their actions and verbalized thought processes were recorded and analyzed to uncover how they systematically made sense of the software systems they were endeavoring to understand.

Adam Bryant completed the observational studies described above and described all tasks, analyses, results, and models of abduction and sensemaking in:

Bryant, A. (2012). Understanding How Reverse Engineers Make Sense of Programs from Assembly Language Representations, Dissertation, Air Force Institute of Technology, WPAFB, OH.

Due to personnel changes during the course of the RDM initiative, reverse engineering was de-emphasized in this sub-project and empirical studies focusing on the general nature of abduction-based inquiry were conducted. These studies investigated abduction-based inquiry in information gathering and decision making tasks. The results of a comprehensive analysis of human subject protocols obtained during a study of evidence-driven decision were described in a paper and conference proceeding:

Vickhouse, R., Bryant, A. & Bryant, S. (2012). Investigating Hypothesis Generation in Cyber Defense Analysis through an Analogue Task. In Proceedings of the 8th International Conference on Information Warfare and Security (ICIW 2013), Denver, Colorado, USA.

A follow-on empirical study exploring abduction and decision making in an intelligence gathering task was conducted. This study explored how analysts generate and test hypotheses about insurgent threats in a fictitious Afghan province. Situational factors analysts used to infer levels of insurgent threats were based on content from: (1) The Defense Science Board Task Force on Defense Intelligence report, "Counterinsurgency (COIN) Intelligence, Surveillance, and Reconnaissance (ISR) Operations"; and (2) Kilcullen, D. (2009). "Measuring Progress in Afghanistan"

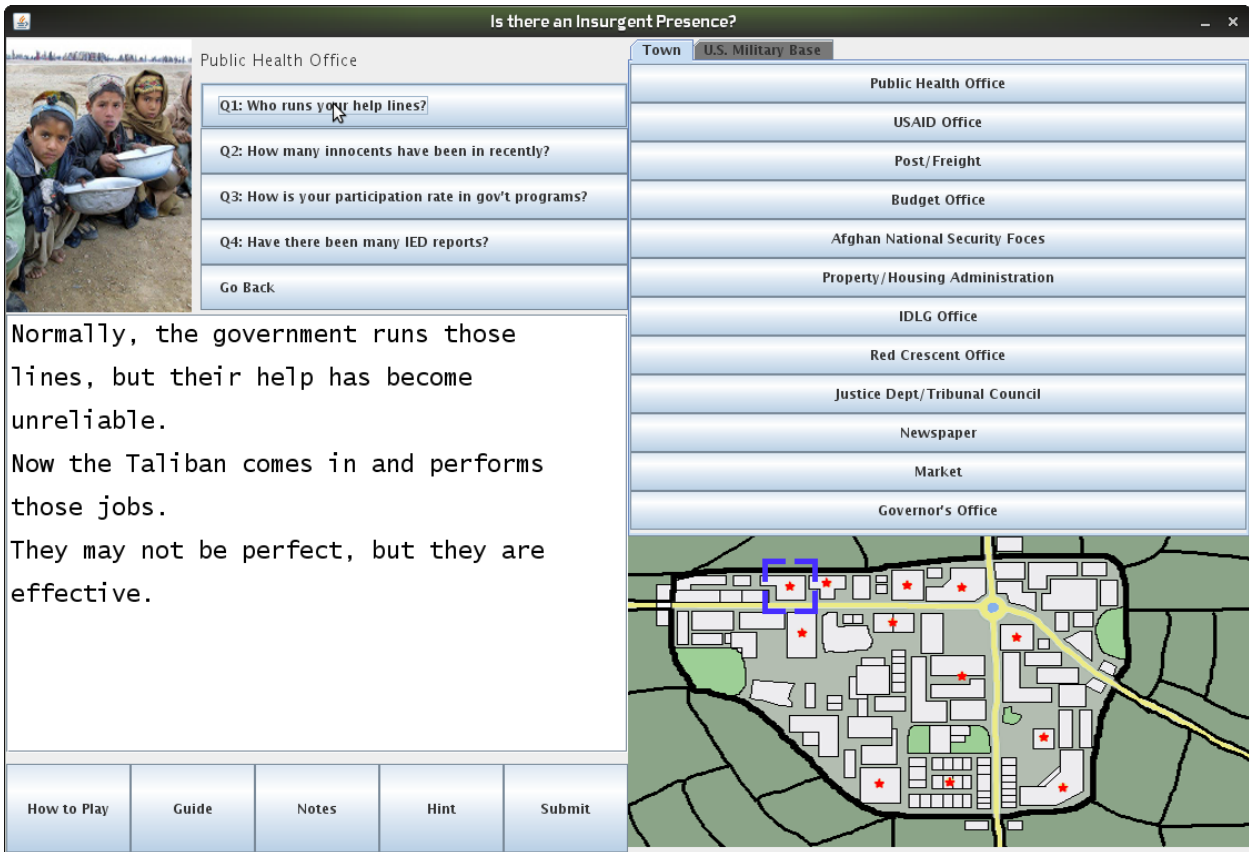


Figure 2: Screen image showing the COIN empirical task.

During the hypothesis generation task, participants interacted with a number of intelligence sources, engage in an evidence-based decision making process, and ultimately provide estimates of: (1) likelihood of insurgent presence; and (2) public sentiment concerning insurgents. The COIN task, shown in Figure 2, is described in:

Kunkle, C., Luehrs, J. & Setlur, V. (2013) Final Report for CDRL A001 Final Report, Task Order 0054: Robust Decision Making for Improved Mission Assurance.

A model of the evidence-based reasoning demonstrated by participants will be specified and executed in the CECEP cognitive M&S framework. Models and agents in the CECEP architecture will “mine” structural and relational domain knowledge capturing situational factors, metrics, and cues in order to mimic participant hypothesis generation and decision making.

[2] A cognitive modeling sub-project developed formal computational accounts of abduction-based inquiry.

This sub-project used the Generic Modeling Environment (Molnár, Balasubramanian, & Lédeczi, 2007; Lédeczi, et al., 2001), to research and develop a DSL specifically tailored to meet the demands of modeling these processes as abduction-based inquiry. To develop the DSL, called the research modeling language (RML), researchers in this sub-project combined the findings from the empirical sub-project with knowledge and experience in the authoring of cognitive models using cognitive architectures.

Budget-driven de-scoping exercises and personnel changes over the course of the RDM initiative led this sub-project to focus on developing a web-delivered RML authoring environment.

Veda Setlur developed a RML model/agent specification environment called “ModelerStudio.” The specification environment is web-delivered for easy distribution and presents a modeling GUI that helps non-programmers specify models and agents. The ModelerStudio is comprehensively described in:

Kunkle, C., Luehrs, J. & Setlur, V. (2013) Final Report for CDRL A001 Final Report, Task Order 0054: Robust Decision Making for Improved Mission Assurance.

Scott Douglass finalized the development of a domain-specific language (DSL) tailored to the needs of modeling the abduction-based inquiry processes uncovered by the Bryant’s analyses. This DSL, called the research modeling language (RML), is described in:

Douglass, S. A. (2013). Learner Models in the Large-Scale Cognitive Modeling Initiative. In Design Recommendations for Adaptive Intelligent Tutoring Systems Learner Modeling (Volume I), Springer-Verlang.

Douglass, S. A., & Mittal, S. (2013) A Framework for Modeling and Simulation of the Artificial. Andreas Tolk (Ed.), Ontology, Epistemology, and Teleology of Modeling and Simulation, Intelligent Systems Series, Springer-Verlag.

Douglass, S. A., & Mittal, S. (2011). Using domain specific modeling languages to improve the scale and integration of cognitive models. In the proceedings of the 20th Annual Conference on Behavior Representation in Modeling & Simulation. Provo, UT.

Mittal, S., & Douglass, S. A. (2011). From Domain Specific Languages to DEVS Components: Application to Cognitive M&S. In the proceedings of the Workshop on Model-driven Approaches for Simulation Engineering, SpringSim'11, Boston, MA.

Mittal, S., & Douglass, S. A. (2011). Net-centric ACT-R-Based Cognitive Architecture with DEVS Unified Process. In the proceedings of the DEVS Symposium, Spring Simulation Multiconference, SpringSim'11, Boston, MA.

Taha, T., Atahary, T. & Douglass, S. A. (2013). Hardware Accelerated Mining of Domain Knowledge. In R. E. Pino (Ed.), Network Science and Cybersecurity, Advances in Information Security Series, Springer-Verlang, NY.

Taha, T., Atahary, T. & Douglass, S. A. (2013). Hardware Accelerated Cognitively Enhanced Complex Event Processing Architecture. In Proceedings of the 14th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Honolulu, HI.

[3] A command and control resiliency modeling and simulation sub-project will incorporate cognitive models developed in the cognitive modeling sub-project into a derivation of the C2WT, a software-intensive modeling and simulation framework.

Budget changes to the overall RDM effort required that this sub-project be de-scoped.

[4] A knowledge facilitation and higher abstraction modeling sub-project used interdisciplinary knowledge elicitation methods to systematically relate the project-level conceptual common ground

developed in the scope of this research to broader conceptualizations of sensemaking

This activity investigated more comprehensive approaches to computational abduction. After Adam Bryant (the investigator with expertise in reverse engineering) withdrew from the research effort, researchers in this sub-project took on challenge of developing sensemaking models and agents in the RML DSL. The results of the broad study of computational abduction were published in two conference proceedings:

Weigand, K. A., & Hartung, R. (2012). Abduction's role in reverse engineering software, IEEE National Aerospace and Electronics Conference (NAECON).

Hartung, R. & Weigand, K. A. (2013). Microgenetic Critic for Situation Assessment Supporting Abduction and Surprise, In Proceedings of the 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing (GEM'13 & ICAI'13).

A detailed description of a RML model/agent capable of employing abductive inference during reverse engineering is described in:

Hartung, R., Garcia, R., Morrow, C. & Culbertson, T. (2013). Sensors Directorate Technologies for Robust Decision Making for Improved Mission Assurance Project Final Report.

Appendix B.: Abduction's Role in Reverse Engineering Software

Kirk A. Weigand

Air Force Research Laboratory Dayton, Ohio,
USA kirk.weigand@wpafb.af.mil

Ronald Hartung

The Design Knowledge Company Fairborn,
Ohio, USA ronaldhartung1@gmail.com

Abstract— As software has become an integral part of most systems, so too have cyber threats become an expected attack vector. This has made the task of reverse engineering software an increasingly necessary and critical skill. Software systems are regarded as the most complex of human designed technologies. Software can be difficult to understand when the source code is provided, but a reverse engineer is restricted to machine code and often intentionally obscured machine code. This makes reverse engineering an extreme technical challenge. This work examines the reverse engineer's cognitive task as abductive reasoning. Abductive reasoning has received significant theoretical attention in the last decade resulting in a broader account of abduction types and methods. Abduction, as the only generative means of inference is essential to hard diagnostic tasks and scientific exploration that require non-deductive and non-inductive hypothesis generation. In particular, we explore manipulative abduction and meta-diagrammatic abduction employed by a reverse engineer to counter falsification of a hypotheses and surprise. With this basis, we are studying the work of reverse engineering with the dual goals of understanding the task and looking at ways AI systems can be constructed to augment reverse engineering. Process philosophy principles of panexperientialism and consciousness are used to form a critique of current AI approaches and some tenants of a novel abductive AI framework are justified.

I. INTRODUCTION

There are a number of possible approaches for studying complex human tasks like reverse engineering. This paper follows a top down approach by introspecting about the task itself and then applying work from philosophy. This is inherently a black box approach, and as such suffers from the problems inherent in studying human behavior from the outside. This approach is not meant to be exclusive of approaches used by the cognitive sciences, but rather an effort to study the problem from the outside with the hope of later tying it to cognitive models to form a whole. This work draws inspiration from work in abductive reasoning and intuition. The other concept that enters into the work is the concept of surprise. Surprise has a range of meanings. Here surprise is used to describe the situation when a person has been pursuing a course of action based on a hypothesis and recognizes a situation that falsifies the hypothesis. This should cause shifts

in her or his plans and hypotheses. The ability to seize the correct shifts and responses appears to be a clear distinction between expert and novice practitioners.

II. ABDUCTION

Reasoning by abductive inference was introduced by Charles Sanders Peirce [1]. Since that time, there has been work in applying abduction in AI systems [2], as well as in further defining and classifying abduction. For our work, two sources of definition have been most useful. Hoffmann provides a classification of abduction into types [3]. Magnani's focus is on scientific work, especially theory formation, thus clearly at the exceptional level of performance. However, his characterization of manipulative abduction is a core aspect of reverse engineering tasks. The other influence on abductive view of reverse engineering is from Klein's work with critical decision making [4].

Hoffmann uses two dimensions to construct a 3 by 5 matrix of abductive types as shown in Table 1. The dimension shown in the columns of this table is the source of the hypothesis and defines the applicability of the new hypothesis to society. The hypothesis can be selected from the memory of the agent (for example, a Selective Fact), one that exists in our culture but is psychologically new to the creator (such as P- creative) or one that is historically new (e.g. H-creative). For our current work we consider only hypotheses which exist in the mind of the Reverse Engineer (RE) and prefaced with "Selective". This is not to imply that the other kinds are unimportant. The reverse engineer needs to use the cultural dimension in order to succeed; knowledge discovery in cyber security is an ongoing process of psychological learning. Reverse engineers may also encounter a novel approach and so may produce historically-new knowledge.

The rows of Table 1 describe the kind of abduction. These range from simplest kind of abduction at the top of the table to the most abstract kind of abduction at the bottom of the table. The simplest form is to select a fact that explains or supports the inference. Moving down the table, the next row describes a selecting a type or concept to support the inference. These appear to be ubiquitously employed in almost all human

TABLE 1.
Hoffmann's Taxonomy of Kinds of Abduction versus Breadth of Usefulness

	Exists in the mind	Exists in Culture	Historically new
Fact	Selective Fact	P-Selective Fact	H-Selective Fact
Type or Concept	Selective Type	P-Selective Type	H-Selective Type
Explanatory hypothesis or Law	Selective Law	P-Selective Law	H-Selective Law
Theoretical Model	Selective Model	P-Selective Model	H-Selective Model
Representation	Selective Meta-diagrammatic	P-Selective Meta-diagrammatic	H-Selective Meta-diagrammatic

activity. These first two kinds of abduction are often applied in In Fig. 1 Magnani describes the reasoning process applied to many diagnostic AI systems [2]. The third kind of abduction is hypothesis formation [1]. He allows induction and selective law abduction where a law or stereotype is used. This abstraction/abduction as dual paths to generate a diagnostic abduction step is to select a law that supports the observations hypothesis. Other authors, especially Josephson, view induction and satisfies as an explanation. This, like the first two kinds, as a special case of abduction, Magnani gives it a first class appears to be common in humans, especially with a domain status as parallel to abduction. Deduction's role becomes the expert.

The last two kinds of abduction in the bottom two rows of the table are the forms that are most interesting from a reverse engineering perspective. Selective model abduction selects a model to support the abduction. Reverse engineers typically select hypotheses that are models. While selective law abduction is used to verify the assumptions in the model and the conclusions derived from it, is the choice of the model is what drives the reverse engineer's path of exploration.

Meta-diagrammatic abduction is a kind of abductive inference where a shift in representation of the problem leads to the inference. When surprise occurs, meta-diagrammatic abduction is one of the tools to clarify the situation and to adopt a new model or change the perspective by which the problem is viewed. This is often done by invoking the tools to obtain a different view of the situation [1].

Magnani provides a clear view of a critical part of the reverse engineering process. Reverse engineering uses tools in the manipulative abduction. Both Magnani and Peirce use mathematical reasoning via diagrams as the illustrative example of manipulative abduction. In mathematics, diagrammatic reasoning has often been the pivotal source of insight into finding the solution to previously unsolvable proofs. At a more applied level, engineers have relied on diagrams to visualize, problem solve, and record designs. Given that software artifacts are considered to be among the most complex objects developed by human kind, the tools used must support visualization and recording. Software becomes too complex to comprehend without multiple levels of abstraction and tools capable of navigating and exposing software structure and behavior supporting those abstractions. The challenge of designing a software system pales in comparison to reverse engineering that same software, because much of the designer's thinking is lost in the translation to an artifact of executable code. Thus, reverse engineering debugging tools are used to expose the workings of the software and to form theories about what its operation may be and, ultimately, what the goal of the software system may be. It is this discovery process where manipulative abduction appears to be most relevant.

verification process for the hypothesis. These deductive inferences produce expected data to be observed. Abstraction is the process to locate or select the observed data to be explained, and from process philosophy may be seen as perceptual abduction with potential ties to Recognition-Primed Decision-making of Naturalistic Decision Making [4]. Peirce asserted that all reasoning was based upon these three kinds of inference [5], and we believe this is a logical process for problem solving in a complex domain like reverse engineering. For the current work, the induction path is not the focus since the expected observations from deduction are not met and surprise requires the abstraction/abduction loop to generate a novel diagnostic hypothesis.

The experienced reverse engineer is able to make abductive leaps because they have a large body of knowledge to apply. The work of Naturalistic Decision Making with critical decision making offers some useful insights into mechanism behind an expert's ability to abductively reason [4]. An expert's experience enables recognizing situations encountered in the past and establishes their perceptual cue saliency, expectancy and biases. However, there is another side to this capability that is described in Klein's work at the macrocognitive sense-making level, which is the recognition that a pattern is not right [6]. Both of these capacities are important in reverse engineering. Both can trigger abduction since the human mind does not require a complete match, but can find partial matches that trigger sub-goals to evaluate to the pattern. But the ability to see that a pattern that has been accepted as a hypothesis is not correct can be a strong trigger for advancing new theories or new ways to view the situation (selective model abduction and meta-diagrammatic abduction). This can be termed surprise in the sense that an unexpected change occurs in the problem solving activity.

Introspective analysis of reverse engineering finds that detecting when a chosen hypothesis is falsified is crucial to their success [7]. This would appear to be especially true when the target code has been designed to obfuscate, mislead or hide. The subjects studied by Bryant showed marked tendencies to be misled into wrong directions or be lead down the garden path per theories of Naturalistic Decision Making.

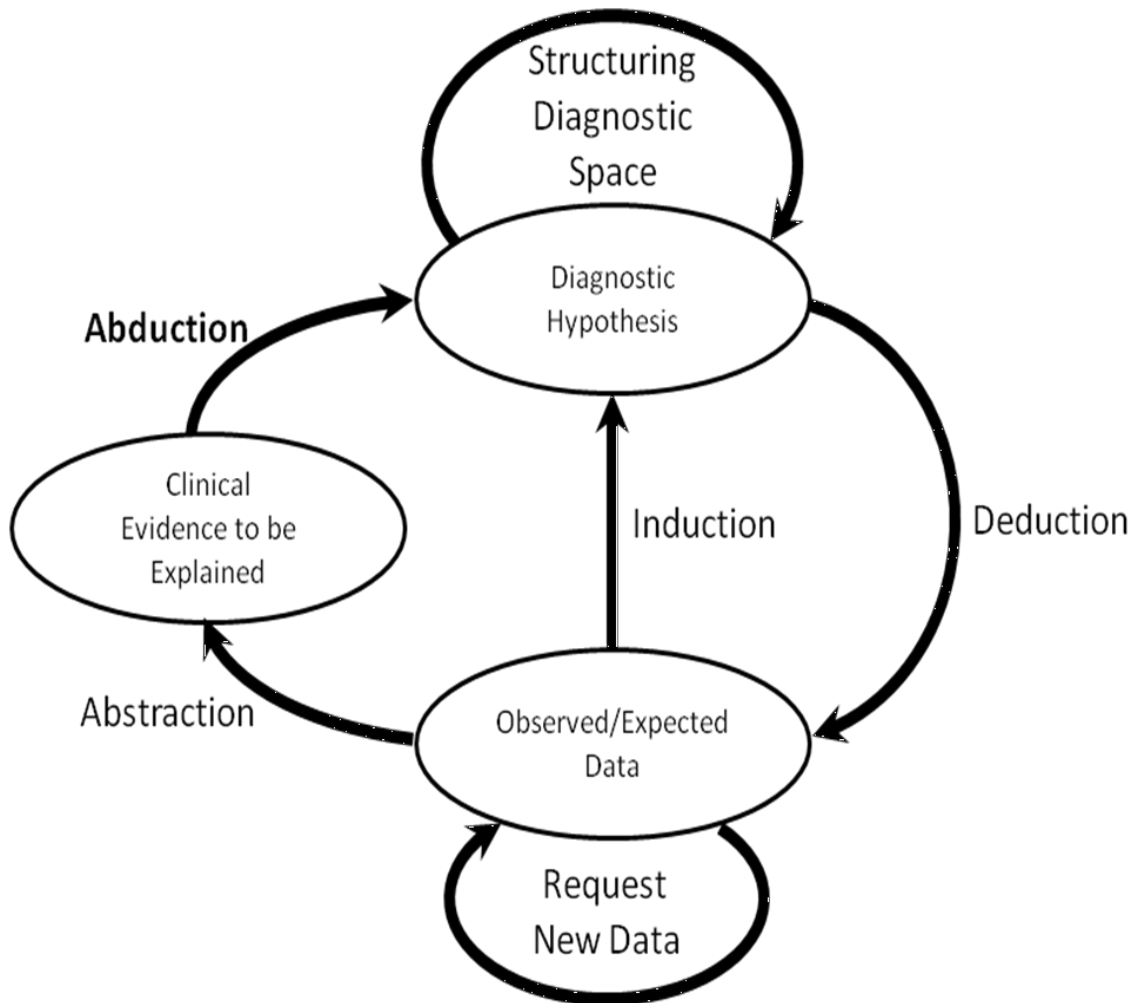


Figure 1. Lorenzo Magnani's Abduction vis-à-vis Induction and Deductive Inference

A strong suspicion, but unverified experimentally, is that failure to see when the evidence is against a hypotheses or the clinging to a hypothesis by trying to patch over problems can be the cause of this affirmation bias [8]. Certainly the history of science would lead to the conclusion that humans can go to great length to preserve a current theory [9]. On this basis, the detection of a falsified hypothesis is an essential aspect of the model successful abductive inquiry in reverse engineering.

III. REVERSE ENGINEERING'S PROBLEM SPACE

Reverse engineering is a broad activity and encompasses a number of sub-categories defined by the intended goal. The

tools are common across this space and the domain knowledge is based on a common base set. The knowledge required to successfully reverse engineer is enormous due to the complexity of software systems. This includes knowledge of low level hardware and language implementation, the operating system details, general computer science area like algorithms, and specific tricks used by attackers. This knowledge occurs at multiple levels of abstraction. Bryant's task analysis work shows that the practitioner's focus shifts between these levels as they explore the problem [7].

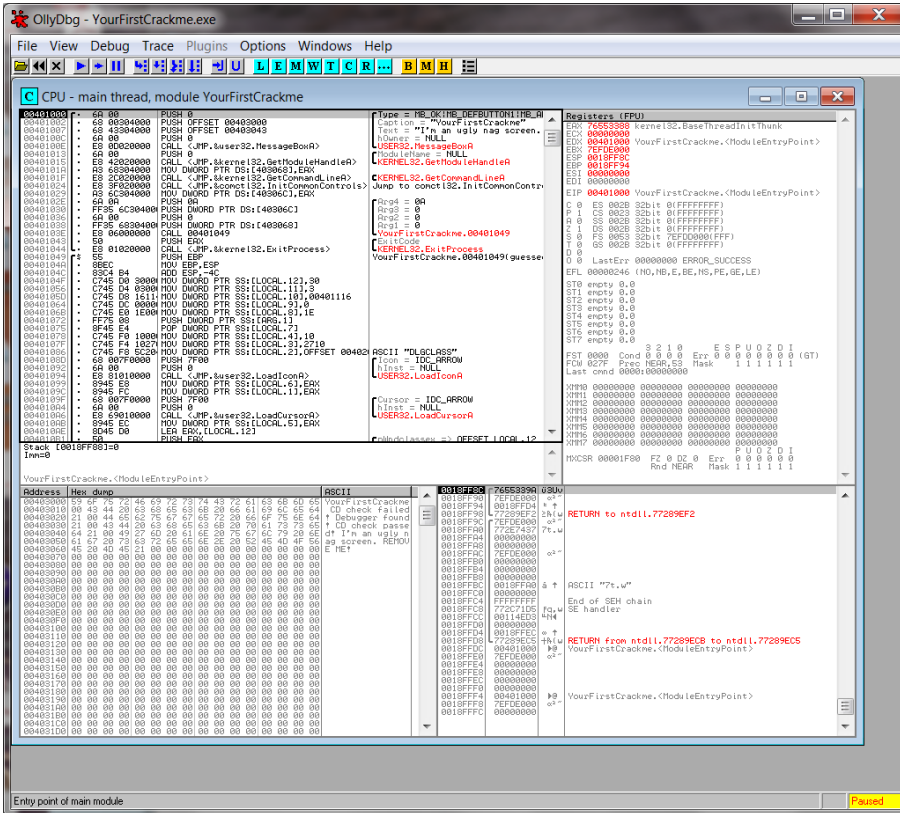


Figure 2. A Typically Complicated Screen-shot from a Reverse Engineering Tool

Software can be difficult to understand when the source code is provided, but a reverse engineer is restricted to machine code and often intentionally obscured machine code. Machine code, or its assembly language interpretation, is generally very large, much larger than the high level language source, and has a complicated data flow and control flow structure. Because of its low level, there can be multiple levels of abstraction between the machine code and the usable meaning the reverse engineer seeks. Complete coverage of the code is intractably large even with automated methods. In any case, automated methods have yet to be devised that can achieve the level of sensemaking required for reverse engineer software. As shown in Fig. 2 reverse engineering tools require complicated use of many windows.

Our interest in reverse engineering is from a cyber-defense perspective. In this case, the reverse engineer is trying to find malicious code fragments within a program. Using reverse engineering for other domains can have a more relaxed deadline, cyber-defense adds a time driven threat to the problem. It also drives the practitioner to triage the critical elements of the code, not to apply a broad attempt to study all parts of the code. The ability to successfully triage is a necessary skill for a reverse engineer. In this problem space, the critical elements may have obscured and hidden by a variety of means. The library code, considered irrelevant in some reversing efforts, can become a target for the reverse engineer; because the suspect code may be using existing flaws in other code or systems to achieve a malicious purpose. The faster an incorrect triage is detected, the larger the savings

in both time and cognitive effort. Again, reinforcing the thesis that detecting falsification and sensing when a pattern is not correct are critical drivers in abductive software exploration.

This is a difficult task and requires skilled individuals. The complexity of a given piece of software can be high and requires the grasp of detailed information in order to understand the operation. The tools used by the reverse engineer are used in what Magnani refers to as manipulative abduction. The tool provides a visual reminder of what has been discovered, as a source for the discoverer and when a theory is falsified, the means to abductively shift the view to explore a new model or representation (Hoffmann's model abduction and meta-diagrammatic abduction).

To limit the domain in our research, the focus will be on obfuscation in binary executables in the Windows operating system. This will exclude the common language runtime based .NET executables and other operating systems. This reduces the set of tools and approaches, but still leaves a large problem space. It is a rich domain for studying abduction as human levels of sense making are required and following false paths is easy, resulting in the need to detect and adjust to surprise.

Obfuscation techniques [10-12] are used for protection of proprietary software as well as means of hiding code. There are a range of techniques applied to obfuscation. There are tools which will automatically obfuscate code. The clever programmer can also restructure code to make it less obvious and to provide defenses against the disassembler and debugger

as well as the human. It is interesting to note that techniques that fool a human and techniques that fool a machine are different.

IV. ABDUCTION APPLIED

The work to apply abduction as a way to understand and to characterize the tasks in reverse engineering leads to a focus on the selective model and meta-diagrammatic abduction in a manipulative abductive mode. This is not to imply that the other levels are not used, but rather this looks like a dividing point between highly skilled vs. novice practitioners. The knowledge used to guide the abduction is split into two types. The direct situation recognition is the knowledge type applied in all knowledge based AI systems and cognitive modeling systems. This is the “I know what to do here” kind of knowledge. It is easier to deal with, because it is triggered by direct evidence and tends to have a closed form.

The other kind of situation recognition is the metacognitive sense-making, “Something isn’t right” variety. This is harder because it tends to have indirect forms. Klein et. al. show a number of examples of expert behavior that show the power of this kind of perception [6].

It can be that a critical fact is missing from the situational description. This is a very interesting problem, given that abduction tries to fill in the missing pieces. At some point, the expert’s processes override the assumptions about missing pieces. An observation may also just not fit the current hypothesis based on the deductive path predicting that observation.

Abduction leads naturally to two important issues. First of all surprise, surprise is a trigger for abduction. When we struggle to deal with an unfamiliar situation, surprise, abduction is one of the powerful tools to reach conclusions quickly and efficiently. Likewise, abduction can lead to further surprise; it is not guaranteed to reach valid conclusions. This leads to the other remark about human reasoning under abduction, it is non-monotonic. Prior conclusions have to be abandoned as reasoning and problem solving progresses so premises evolve with understanding. This sometimes requires whole chains of reasoning to be removed from consideration. Fortunately for a human, holding contradictory positions is not a limitation. Unfortunately, current AI systems have difficulty with non-monotonic reasoning.

The abductive example used in this paper is not the only, nor the most complex, example in reverse engineering. Reverse engineering is heavily invested in abduction. The area of code obfuscation is approachable with less overall computer science knowledge, making it more approachable for study. It is a mid-level problem, above tracing and code reading, but below determining intention and recognizing algorithms.

V. CODE OBFUSCATION

The transformations used in obfuscation can be classified into four main categories with a number of subtypes. This is clearly a finite set of possibilities; however the job of recognizing and untangling them can be difficult. The problem

is twofold, first recognizing what kind of transformation was applied and then changing the representation back to a form that displays the intent. This involves both meta-diagrammatic and model abductions. There are several conceivable cases. The transformation may be enough and no model shift is required. Or, the transformation may invoke the need for a model change. Or even that a model shift may precede the meta-diagrammatic transformation. In all cases, there is a fundamental recognition that the code seen is not showing the right properties and something is hidden.

It is worth noting that obfuscation can mislead and definitely can slow down the reverse engineering. However, the computation being performed must be preserved, hence the necessary information is still present for sense making by a reverse engineer.

The four categories are layout obfuscation, data obfuscation, control obfuscation and preventative obfuscation. While a complete study of these is not the goal of this paper, some instructive examples are required to make the argument for the claim of abductive reasoning. Layout obfuscation removes some formatting and naming information. This is considered a low potency transformation [11]. For that reason, we dispense with any discussion of these transformations. In data obfuscation, one approach is to convert constants to procedures. Here the recognition is that the computation is not required and a simple constant is all that is needed. The RE will have to trace the computation to put it into a form where it is clear that a constant is the result. This often requires some knowledge of math to recognize the formula in use and understand its properties.

Another similar complication is to split a variable. This requires the RE to detect that two or more values are always used together and then transform them into the real variable. Again this is a meta-diagrammatic transformation and it leads to a more compact representation. A model shift may occur because a detail of an algorithm may be unveiled by the transformation.

Data obfuscation can be achieved with aggregations. This can be to merge independent data or to split dependent data. For example, arrays can be split, folded or merged to hide the intent of the contained data.

These methods all make location of structures found in the higher level language representation more difficult to discern in the code.

Control obfuscation can seek to obscure control flow, how the code is aggregated or how the code is ordered. An interesting approach here is to reorder code in ways where a high level language analog does not exist. This will require the RE to derive the intent by low level tracing methods. Another technique is Inlining or outlining code. Inlining removes procedures (methods) and outlining creates extra procedures. Again the RE must group the methods and look for how they are used in order to infer intent. This can be more model abduction prior to a meta-diagrammatic shift.

A very useful technique for misleading the RE is to insert dead or irrelevant code. This is achieved by opaque predicates that insure branches proceed to live code and the dead code is

never executed. This requires the RE to identify the dead segments so they are ignored and to meta- diagrammatically replace the opaque predicates.

Preventative transformations target the automated deobfuscator tools. They are nonetheless a potential confusion for the human RE. An example is to reverse the direction of a loop. This can make an algorithm appear different and will require more work on the part of an RE to correlate the reversed form of the loop with its more conventional forward form.

The combinations of all of these transformations result in the need to abductively manipulate the code to reorder and reassemble it into a form where it can be recognized and understood.

VI. FUTURE WORK TO COMBINE PROCESS PHILOSOPHY WITH COGNITIVE MODELING

The view of abduction described in this paper, now needs further testing. This will proceed in two dimensions. More human observations need to be undertaken in the restricted domain of obfuscation. And second, an AI system to demonstrate the abductive reasoning applied to obfuscation needs to be constructed. The outcome of the two approaches will allow validation of the theoretical view given in this paper.

Cognitive Domain Ontologies (CDOs) [13] may be useful for knowledge representation of abductive reasoning needed to devise decision support systems for reverse engineering of software. Process philosophy may enable a refinement of abstractions used in CDOs to better articulate the roles of behavior models, declarative memory models and ontologies that store constraints about context of use. Means-end goal structures appear to be necessary to enable purpose modeling that is needed to operate these models across diverse contexts of use and various environmental constraints. Ultimately, work in process philosophy indicates that some level of awareness of experience may be needed for AI agents to be adaptive across these diverse ecologies of use. Self-awareness drives the need for a high level of experience that may ultimately tie to work in consciousness studies. A consciousness measure, Φ , Phi, is being developed based on Integrated Information Theory of Giulio Tononi & Gerald Edelman [14] and this work is supported by the panexperiential theory of Alfred North Whitehead's process philosophy [15].

We speculate that if successful AI agents might use information on state space of the reverse engineer and combine that with valuations defining goals as means-end affordances, then these could be used by agents using CDOs to generate behavioral and procedural actions that could help the reverse engineer during abductive inference. This might entail enabling the reverse engineering tool to make a meta-representational or theoric shift as proposed by Magnani and Hoffmann.

VII. CONCLUSIONS

The thesis of this work is that abduction plays a critical role in reverse engineering. This is an interesting point of

departure for two approaches. From an attempt to provide cognitive models of the reverse engineering task, abduction clearly will be needed. The other point of this work is how to help the human involved in the task of reverse engineering. While it does not appear to be a domain where automated tools can solve the problem, a symbiosis of human and machine may be useful. If so the machine needs to respect and follow the abductive leaps of the human. In addition, the machine should be able to provide alternative paths to consider for the human. This appears to be a system that can be built.

ACKNOWLEDGMENT

This research was funded by the Robust Decision Making Strategic Technology Team, Air Force Research Laboratory, Wright-Patterson AFB. Approved for public release, case number 88ABW-2012-4895.

REFERENCES

- [1] L. Magnani, *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Berlin/Heidelberg: Springer Verlag, 2009, p. 16.
- [2] J. R. Josephson and S. G. Josephson, *Abductive Inference: Computation, Philosophy, Technology* Cambridge: Cambridge University Press, 1996.
- [3] M. H. G. Hoffmann, "Theoric Transformations' and a New Classification of Abductive Inferences," *Transactions of the Peirce Society*, vol. 46, no. 4, pp. 570-590, 2011.
- [4] G. Klein, *Sources of Power: How People Make Decisions*. Cambridge/London: MIT Press, 1999.
- [5] C. Hookway, *Peirce*. London/New York: Routledge, 1992, p. 30.
- [6] G. Klein, J. K. Phillips, E. Rall, & D. A. Peluso, "A data/frame theory of sensemaking," in *Expertise out of context: Proceedings of the 6th International Conference on Naturalistic Decision Making*, R. R. Hoffman, ed. New York: Erlbaum, Taylor and Francis, 2007.
- [7] A. R. Bryant, R. F. Mills, G. L. Peterson, M. R. Grimaila, "Software reverse engineering as a sensemaking task," *Journal of Information Assurance and Security*, vol. 6, no. 6, 2011, pp. 483-494.
- [8] E. Yudkowsky, "Cognitive biases potentially affecting judgment of global risks" in *Global Catastrophic Risks*, Bostrom and Cirkovic, eds. Cambridge: Oxford University Press, August 2011.
- [9] T. Kuhn, *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press, 1996.
- [10] G. Wroblewski, "General Method of Program Code Obfuscation," *The 2002 International Conference on Software Engineering Research and Practice (SERP'02)*.
- [11] C. Collberg, C. Thomborson, and D. Low, "A Taxonomy of Obfuscating Transformations," *Technical Report #148*. New Zealand: Department of Computer Science, The University of Auckland, 1997.
- [12] E. Eilam, *Reversing Secrets of Reverse Engineering*. New York: Wiley, 2005.
- [13] S. A. Douglass and S. Mittal, "A Framework for Modeling and Simulation of the Artificial," in *Ontology, Epistemology and Teleology for Modeling and Simulation*, Intelligent Systems Series, ed. A. Tolk, Berlin/Heidelberg/New York: Springer-Verlag, 2012, pp. 271-317.
- [14] Christof Koch, *Consciousness: Confessions of a Romantic Reductionist*. Cambridge: MIT Press, 2012.
- [15] D. R. Griffin, *Whitehead's Radically Different Postmodern Philosophy: An Argument for its Contemporary Relevance*. Albany: SUNY Press, 2007.

Appendix C. Saliency, Expertise, and Intuitive and Analytical Reasoning in the use of Decision Support Tools

Project Lead: John Salerno, AFRL/RIED

PIs: Christine Covas-Smith, AFRL/RHAS; Dick Deckro, AFIT/ENS; Major Matthew “JD” Robbins, AFIT/ENS

Contributors: Dr. Robert Patterson, (AFRL/RHXM), Dr. Lisa Tripp (AFRL/RHAS), Ms. Christina Kunkel (Leidos, Inc), Prof Rong Pan (Arizona State University (ASU)); Jason Smith (ITT); Adam Kwiat (CUBRC); Major Jonathan S. Findley (AFIT)

1.0 Introduction

One of the most important decision making skills an individual can possess is the ability to project potential alternative courses of action into the future. Having the ability to make such projections is one component of robust decision making. This ability develops through training and experience and it can also be enhanced by using decision-support tools. Decision support tools are systems designed to support and even enhance human decision making augmenting the amount of information that individuals can process and in many cases, aiding with information integration. Decision support tools are especially valuable and needed when the domain in question is highly complex, dynamic in nature, and requires incorporating large amounts of data from multiple sources (Klein & Calderwood, 1991). The objective of the current research was to determine the effects of several factors on learning to use a complex, decision support tool to facilitate robust decision making.

The decision support tool used for the current research is the National Operational Environment Model (NOEM). NOEM is designed to provide environmental, economic, and societal-level decision support for strategic- operational- and tactical-level military decision makers and analysts. NOEM is a large-scale stochastic model that represents the environment of a given geopolitical region. NOEM allows one to identify potential problem regions, test a variety of socioeconomic policy options, and investigate the implications on local and national communities. NOEM was designed for use by intelligence analysts and strategic decision makers to assist in the integration of large amounts of information that must be interpreted and synthesized every day. The goal for this tool is to enable analysts to combine information modeled in the software with current operations information before making a decision.

Decision making includes both an analytic rule-based process and an intuitive, highly implicit process which involves the detection of situational and environmental patterns (Kahneman & Klein, 2009; Evans, 2008; Pretz, 2008; Sloman, 1996). Since many decision-support tools involve integration of large amounts of information, as well as simulating higher-order relationships among multiple complex factors, acquisition of skill in using such tools would involve the intuitive decision making process (Berry & Broadbent, 1984). The current research investigated the effects of saliency on learning (Experiment 1), the development of analytical versus intuitive decision making (Experiment 2), and the effect of the order of analytical versus intuitive training on learning and the development of decision making expertise with complex, decision support tools (Experiment 3). Each of these factors is thought to influence the

development and use of robust decision-making processes in complex, dynamic decision making. In Section 2 we provide an overview of the NOEM (its tool and model) and present an example scenario. This scenario has been used as part of the experiments (discussed in Section 3), use of Decision Analysis Techniques to investigate optimum solution sets in Section 4 and investigation of possible metrics for defining the “best” solution for the decision maker in Section 5.

2.0 NOEM Overview

The overall goal of the NOEM is to provide an infrastructure in which an analyst can investigate/explore the complex state space that forms the environment in which we all live. In order to accomplish this objective, the NOEM is divided into three components: (1) A set of tools available to exercise the model; (2) the model itself and (3) a set of Support utilities that maintain data currency and baseline state. The yellow boxes (Figure 1) represent the collection of these components while the blue boxes detail the products generated or supported by the tool. The Model Definition Environment (MDE) allows one to configure/define the existing modules for a given country.

2.1 NOEM Tools

The first tool available in NOEM is the Consequence Analysis (CA) Tool. The purpose of the CA Tool is to provide a geo-spatial view into the NOEM model and allow one to explore the consequences or ramifications of exercising a given weapon or collection of weapons within or over a designated area. Figure 1 depicts the tools available in the NOEM including the CA tool. The tool is independent of the weapon type. A weapon is chosen from the list by selecting the “Choose Weapon” button. Choosing the weapon will then tell NOEM what effects will be generated. Clicking on the map will place the weapon and a ring of destruction on the map (provided the weapon has collateral damage associated with it). A list of targets can also be imported. After the target set is identified the user simply clicks on the “Examine Consequences” button. At this point NOEM will take the rings of destruction, identify any/all assets with the rings and degrade them accordingly. This process will generate an input set used to load the simulation and run the simulation. The results, once completed can be viewed through the spider graphs, as displayed on the right side.

2.2 The Model

The NOEM model supports the simulation and the analysis of a nation-state’s operational environment. Within the MDE, the models can be configured for one or more regions, where each region is composed of a group of highly interconnected modules which simulate subsystems such as a region’s demographics, economy, or critical infrastructure. These modules, in essence, relate to the major pillars of a nation-state based on stability operations theory (Governance, Security/Rule of Law, Economy and Social Well-Being). Figure 2 provides a representative view of the overall model. It is envisioned that additional modules can be added as needed. Such additional modules can include: education, health infrastructure, mining, etc.

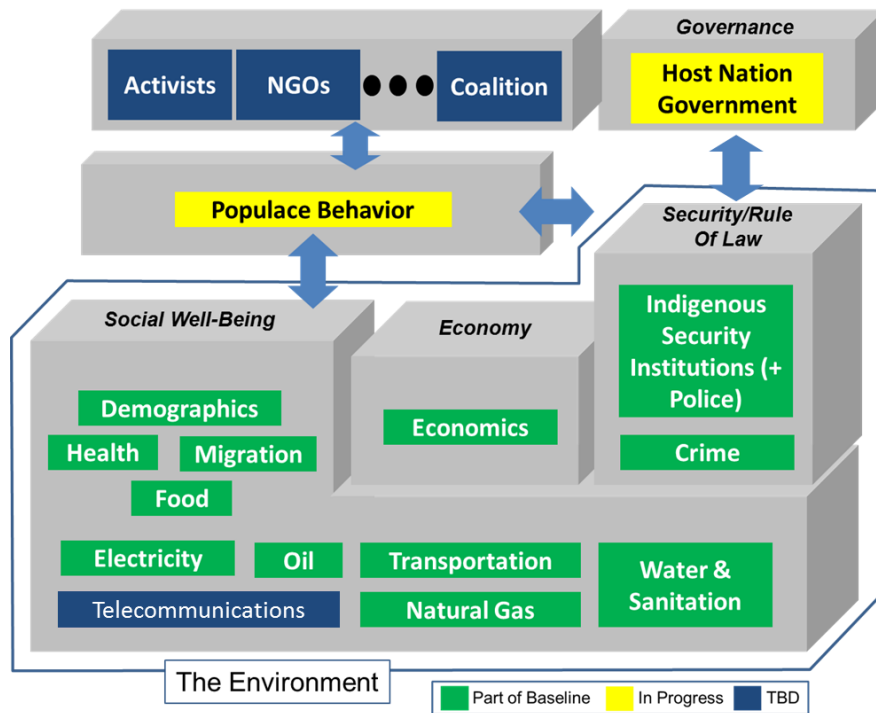


Figure 2 - NOEM model and its constituent modules

Security/Rule of Law is defined as protecting the lives of the populace from immediate and large-scale violence, and ensuring the state’s ability to maintain territorial integrity. The Security/Rule of Law pillar is currently comprised of two modules: Indigenous Security Institutions (ISI) to include local security and Crime. The ISI module is divided into: Border Patrol, Civil Defense, Facility Protection Services, Indigenous Military and Police. Economy is defined as a system made up of various economic policies, macroeconomic fundamentals, free market, and international trade that exchanges wealth, goods, and resources mostly free of economic criminal activity. Governance is defined as a public management process that involves a constituting process, governmental capabilities, and participation of citizens. Social Well-Being is defined as sustenance of life and relieving of suffering by way of humanitarian aid, best practices, human rights, essential services, and emergency response systems. The Social Well-Being pillar is composed of the majority of the modules and includes: Demographics, Health, Migration, Food,

and fundamental Utilities (Electric Power, Telecommunications, Natural Gas, Oil, Transportation, and Water & Sanitation).

The agent-based Populace Behavior Module forms the heart of the NOEM model in the sense that all other modules (resources and security) are in place to support the populace. If the populace is not happy or satisfied to a certain degree of expectation, they could become activists and rebel against the host-nation government. Whether or not segments of the populace become activists depends on many factors, including their perceived hardship, the legitimacy of (or belief in) their government, their level of risk aversion, and the amount/visibility of security forces. Insurgents, Coalition forces, NGOs, and Host Nation Governments within the NOEM are not modeled as agents, but are characterized by the policies or strategies that they implement. Policies implemented by such groups will affect either the overall security within the environment or the services/resources provided to the people.

2.3 The Scenario

The scenario used for the current research is based on the Democratic Republic of Congo. The DR Congo is only used to provide a framework for infrastructure and demographics and the situation or crisis that is provided to participants and used as part of the model is fictitious. Consider that a baseline forecast was run based on the current situation in this country. The baseline forecast indicated that DR Congo will be undergoing financial instability in the near future if it does not quickly change its current fiscal policies. The country's borrowing ability will soon be denied since their maximum allowable debt (based loosely on the IMF's policies) will exceed 150% of their GDP. Figure 3 provides two charts. The first one (on the left) displays the amount of debt and the maximum allowable debt (in USD dollars) for a two year run. The chart on the right provides the number of people considered to be activists (plot over time). Based on the forecast, this will occur around day 28 after our simulation starts. The country is headed for bankruptcy. Therefore, the objective will be to minimize the amount of borrowing (i.e., the debt). One way to accomplish this would be to stop spending or increase revenues (i.e., taxes). However, there may be many downfalls should we attempt to do this. While our primary objective is to minimize the debt, we add a secondary objective: minimize the number of people becoming upset (or in our case, the number of people willing to rise up) in reaction to the policies implemented for the purpose of reducing debt.

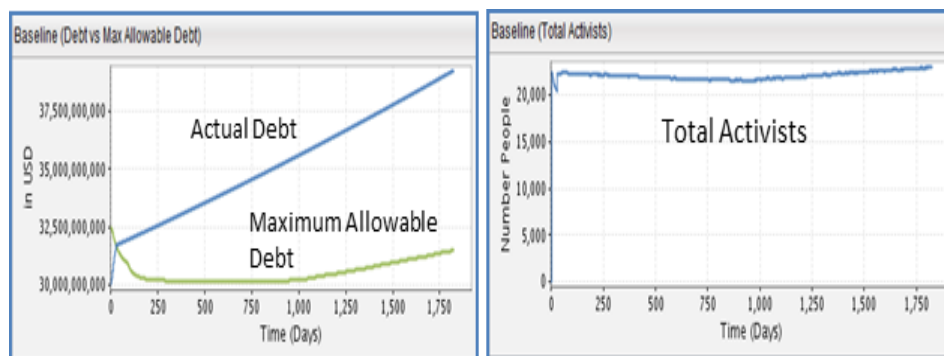


Figure 3 - Baseline Forecasts

Now that we have defined our problem and have identified our goal/measure of effectiveness, we next generate an initial set of policies that we believe would likely provide us the “best” solution. Typically, these potential solutions or policy sets are debated within a group, revised, and then an agreed upon direction forward is adopted. However, how do we know whether any given policy set chosen was the best one (having taken the right actions), the most robust, or the most resilient? How does one know that it is the lowest cost, and how long could it be until one begins to see appreciable results? How long until the full effects can be realized? How does one even know what actions one should take to achieve the most effective possible solution? These are the types of questions policy makers have to answer each day -- questions that the NOEM and its suite of Decision Support Tools strive to support. We will address these metrics and what constitutes a “good” decision in Section 4.

2.3.1 Defining the Objective/Goal

In the current example, our primary objective is to minimize the amount of long term debt accrued by our country, DR Congo. Another objective is to avoid inciting a riot or revolt; i.e., any actions we take should not increase the discontent of the populace (measured in terms of the number of populace willing to become activists). Thus, our initial goals had been to minimize debt and the number of activists. We found that what we needed to do was to maximize the difference between the maximum allowable debt and the actual debt. This is due to the fact that the maximum allowable debt is a function of the GDP and only provides an upper ceiling as to the amount of money a country can borrow. Solely driving the debt lower does not provide us an insight into whether we are below the maximum allowable debt, since the GDP could decrease and in such a case so would the maximum allowable debt. We weight both the number of activists and the distance between maximum debt and actual debt equally. The next question is to determine to what extent it is possible to achieve this objective. Of course, we can set the goal for both the debt and the number of activists to be zero, but this is unrealistic. With that in mind, what are the realistic values? One way is to simply state a number for each based on expert opinion. A second option might be to derive them based on a set of runs to determine what is achievable. We opted for the second approach, and do not specify the actual values for our objective function, which can be written as:

**GOAL = MAXIMIZE (MAXIMUM ALLOWABLE DEBT – ACTUAL DEBT) AND
MINIMIZE (TOTAL ACTIVISTS)**

2.3.2 Identifying the Inputs/Actions

After identifying the outputs of interest, we next need to define what inputs or actions we believe are necessary to achieve our desired objective. To identify these actions, we have developed a query or associate tool. The tool allows the user to select a given output in order to view a list of inputs that most directly affect it. The number of inputs displayed will be based on the degree of significance that the input contributes to the selected output (user selectable). Highlighting the inputs will automatically select them for the next step in the process – creating an initial set of policies. In our example, fifteen inputs have been identified. Table 1 provides this list along with their default values and chosen acceptable ranges.

Table 1 - Input Variables

Inputs	Default Value	Range
Adjudication Rate (in %)	0.7	0.1 – 0.9
Government Corruption Theft Percentage (in %)	0.1	0.05 – 0.15
Government Infrastructure Spending Percent (in %)	0	0 – 1.0
Government Services Spending Percent (in %)	1.0	0 – 1.0
Government Stimulus Spending Percent (in %)	0	0 – 1.0
Government Wages (in \$'s)	2,835.84	1,400 – 4,200
Initial Police Forces (in Number People)	22,000	112,500 – 137,500
Interest Rate (in %)	0.045	0.0225 – 0.0675
Jail Term (in Months)	30	3 – 100
Long Term Government Share Of Employed (in %)	0.001	0.001 – 0.05
Mean Adjudication Processing Time (in %)	0.5	0.1 – 0.9
Military Acting As Police Percent (in %)	0.1	0.0 – 0.2
Police Forces Goal (in Number People)	22,000	112,500 – 187,500
Stimulus (in \$'s)	10,000,000	0 – 6,000,000
Tax_Rates (in %)	0.0176	0.03 – 0.50

2.3.3 Generating a Set of Policies

This next step is to create an initial set of policies. This is accomplished by using the inputs/outputs defined in the previous steps, their specified ranges, and space filling techniques such as Latin Hypercubes Sampling (LHS) or Leap Halton. The goal is to create an initial set of policies that uniformly cover the design space. This design space is based on the number of inputs, their acceptable ranges, and the total number of allowable runs, which is user selectable. Each simulation run consists of one input or policy set along with the list of desired outputs. These are sent to the NOEM, which executes the given simulation and returns the results. The results of each run can now be compared to the objective function to examine how close to “optimal” the given policy set has come. If one or more of the results or policy sets are within a given tolerance, we are done.

In our example, a total of 1,000 runs were conducted. In this case, since we are minimizing only two values, we can simply plot them as a two dimensional scatter plot - debt versus total activists. Figure 4 provides this plot for one-year and five-year projections. We can see that, as time progresses, the dispersion of the results increases.

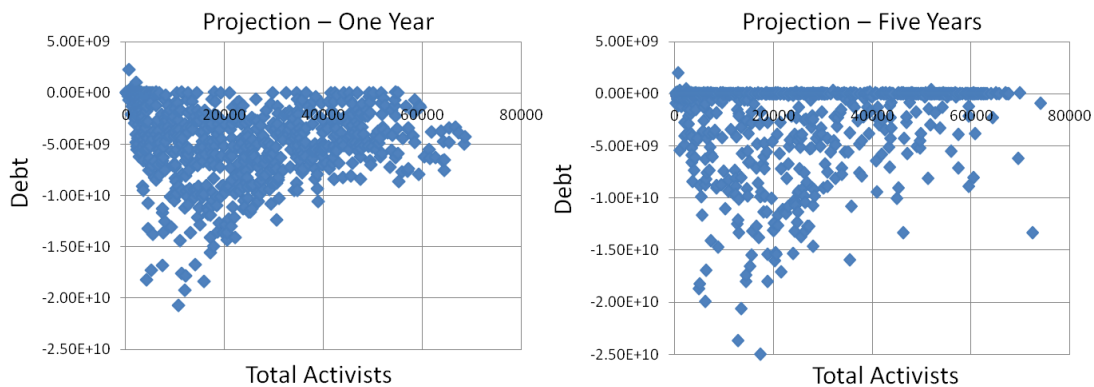


Figure 4 - Results of 1000 runs at the end of one and five years

2.3.4 Analyzing the Results

Based on these runs, we can now find our goal values (the best that can be done independently) for both the delta between the maximum allowable debt and the actual debt, and for total activists. This is accomplished by searching through each run for a given day and finding the minimum (or maximum) value for that output. In our example we have:

Table 2- Goal Values

Output	One Year	Five Years
Maximum Allowable Debt – Actual Debt	2.3E+09	2.00E+09
Minimum Number of Total Activist	622	716

In a two variable world, we can simply find the policy set that is closest to the objective. In a multi-dimensional world we can still use the concept of closeness, but generalized for an n-dimensional Euclidean space. We use the goal as the target point and compute a distance between it and each policy set result. Table 3 provides a listing of the closest ten policy sets after one- and five-year projections.

Table 3 - “Best” Policies

Rank	1 Year		5 Year	
	Run No.	Distance	Run No.	Distance
0	160	0.006	160	0.007
1	984	0.063	984	0.062
2	142	0.087	142	0.064
3	145	0.095	49	0.067
4	411	0.096	630	0.070
5	214	0.096	867	0.070
6	810	0.096	500	0.070
7	581	0.096	389	0.070
8	419	0.096	766	0.070
9	515	0.096	681	0.070

Comparing the two projections, we can see that the top three runs are the same, Run 160, 984, and 142. We next examine these top policies for each of our fifteen variables. Table 4 provides these values:

Table 4 - “Best” Policies (Input Values)

Inputs	Run 160	Run 984	Run 142
Adjudication Rate [0.7]	0.03	0.08	0.07
Government Corruption Theft Percentage [0.1]	0.06	0.08	0.07

Government Infrastructure Spending Percent [0.0]		0.24	0.75	0.96
Government Services Spending Percent [1.0]		0.007	0.009	0.033
Government Stimulus Spending Percent [0.0]		0.807	0.985	0.503
Government Wages [2,836]		1,479	4,062	1,417
Initial Police Forces [22,000]		124,897	136,228	107,183
Interest Rate [0.045]		0.034	0.058	0.0398
Jail Term [30]		46	96	74
Long Term Government Share Of Employed [0.001]		0.0457	0.0372	0.0079
Mean Adjudication Processing Time [0.5]		0.23	0.48	0.57
Military Acting As Police Percent [0.1]		0.17	0.03	0.095
Police Forces Goal [22,000]		113,976	168,146	142,586
Stimulus [10,000,000]		471,071	42,325	565,747
Tax_Rates [0.0176]		0.4549	0.1699	0.42498
Total Activist	[End of Year 1]	622	2057	872
	[End of Year 5]	716	2299	985
Debt	[End of Year 1]	3.47E+10	3.15E+10	3.37E+10
	[End of Year 5]	5.21E+10	4.18E+10	5.05E+10
Max Allowable Debt	[End of Year 1]	3.70E+10	3.25E+10	3.40E+10
	[End of Year 5]	5.41E+10	4.36E+10	5.08E+10

We can now display the results of our policy sets through the NOEM and see how well we did. Figure 5 provides the results for runs 160, 984, and 142. The two smooth lines (green and blue) represent the maximum allowable debt and the actual debt, respectively. The third curve (purple) provides the number of activists.

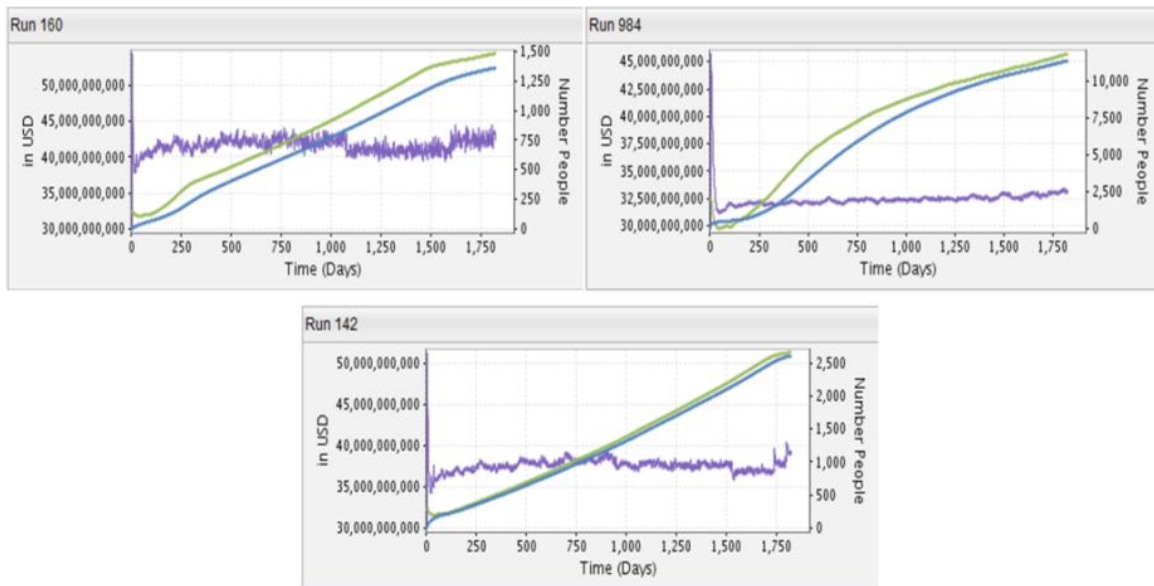


Figure 5 - Results for runs 160, 984 and 142

A second way to view the results is through the use of a Decision Tree. A decision tree is a tree like graph of decisions and their possible consequences. To illustrate, we used the one-year data and

labeled the top ten policies as good (Table 4) and all of the other policies as bad. Figure 6 is an image generated using the rapidminer package (see www.rapidminer.com). The tree had been constructed using the information gain criterion (branching based on most information). Based on this decision tree, rules can be derived and English-like explanations can be generated, essentially describing how a good policy for the given scenario is composed.

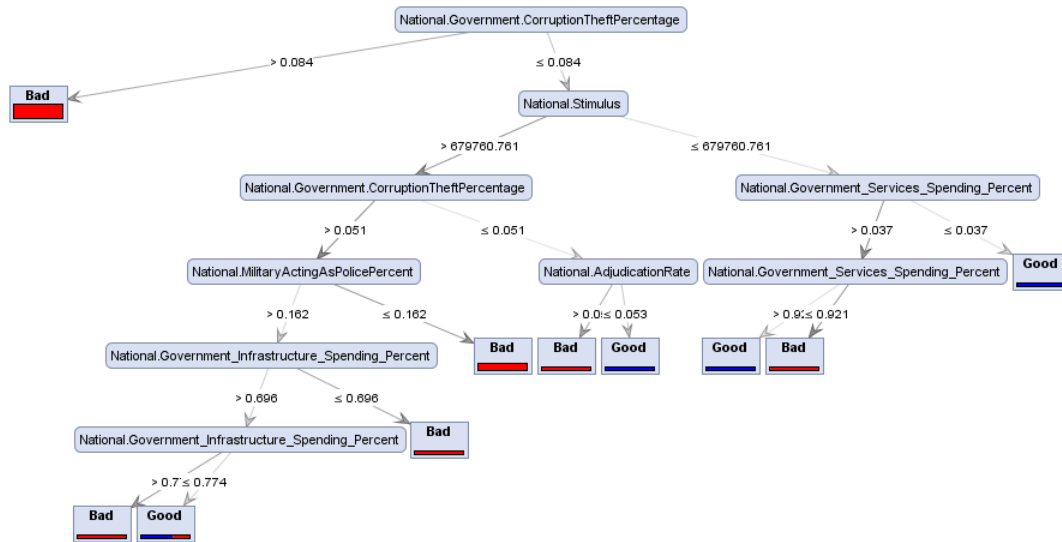


Figure 6 - Decision Tree

Figure 7 is a text view of the same graph. The text view provides a set of rules which can be used to form English-like explanations.

```

National.Government.CorruptionTheftPercentage > 0.084: Bad {Good=0, Bad=658}
National.Government.CorruptionTheftPercentage ≤ 0.084
| National.Stimulus > 679760.761
| | National.Government.CorruptionTheftPercentage > 0.051
| | | National.MilitaryActingAsPolicePercent > 0.162
| | | | National.Government_Infrastructure_Spending_Percent > 0.696
| | | | | National.Government_Infrastructure_Spending_Percent > 0.774: Bad {Good=0, Bad=12}
| | | | | National.Government_Infrastructure_Spending_Percent ≤ 0.774: Good {Good=2, Bad=1}
| | | | | National.Government_Infrastructure_Spending_Percent ≤ 0.696: Bad {Good=0, Bad=42}
| | | | National.MilitaryActingAsPolicePercent ≤ 0.162: Bad {Good=0, Bad=243}
| | | National.Government.CorruptionTheftPercentage ≤ 0.051
| | | | National.AdjudicationRate > 0.053: Bad {Good=0, Bad=4}
| | | | National.AdjudicationRate ≤ 0.053: Good {Good=2, Bad=0}
| | National.Stimulus ≤ 679760.761
| | | National.Government_Services_Spending_Percent > 0.037
| | | | National.Government_Services_Spending_Percent > 0.921: Good {Good=2, Bad=0}
| | | | National.Government_Services_Spending_Percent ≤ 0.921: Bad {Good=1, Bad=30}
| | | | National.Government_Services_Spending_Percent ≤ 0.037: Good {Good=3, Bad=0}

```

Figure 7 - Text view of the Decision Tree in Figure 6

If we follow the left branch of the tree we see that if National Government's Theft due to corruption is over a threshold of 0.084, then we will have loss of money and likely a higher debt. The people will see problems with their government, so there will be a high number of activists and hence we always end up with either bad debt or high activists and hence a bad policy (0 good policies out of 658 cases). For deeper ending nodes, it is only a matter of finding an end node (one with a "Good" or "Bad" label), and tracing through the tree down to that node, to get an English explanation. Let's look at the "Good" end node below Government Services Spending Percent, the furthest to the right in the tree above.

The path through the tree is:

Corruption Theft Percentage → National Stimulus → Services Spending Percent.

Applying the tipping points depicted in the tree, the rule for this end-node becomes:

If the Government's Corruption Theft Percentage is less than .084, and National Stimulus is less than or equal to 679,760.761, and Government Services Spending Percent is less than .037 (3.7%), then the simulation always ends satisfactorily (3 good policies out of 3 cases).

3.0 Experimental results

In the previous section we have presented an overview of a decision support system and a sample scenario. These two items provide us with a starting point and will be referenced as part of this and the next section. In this section we describe the results of studies designed to investigate the effects of saliency on learning (Section 3.1, Experiment 1), the development of analytical versus intuitive decision making (Section 3.2, Experiment 2), and the effect of the order of analytical versus intuitive training on learning and the development of decision making expertise with complex, decision support tools (Section 3.3, Experiment 3). Each of these factors is thought to influence the development and use of robust decision-making processes in complex, dynamic decision making.

3.1 Experiment 1- Effects of Model control and saliency of variables

Experiment 1 assessed the development of intuitive and analytical decision making processes as a function of the level of control of the underlying model or saliency of the task (Covas, Patterson, Kunkle, & Tripp, 2013). Saliency here is related to the model control in that the variables with greater control of the model have a greater impact on the outcome of the model, i.e. their inputs are more salient to the output of the model. Berry and Broadbent (1988) found that analytical decision making was beneficial for controlling interactive computer-implemented tasks when the underlying task structure was salient (see Reber, et al., 1980). This finding suggests that the saliency of the underlying task structure and relatedness of the variables underlying complex decision support tools should be important for determining whether they can be learned analytically as well as intuitively, and that methods should be tried that combine the two processes.

In Experiment 1, two groups of participants were trained to use NOEM and were presented with the DR Congo Scenario/Problem and a predetermined set of variables manipulating the underlying model in either a high (3 variables) or a low (8 variables) model control condition. Table 5 lists the variables used in both the high control and the low control condition. This study was a 2 between-subjects (level of control—high and low) x 3 within-subjects (test sessions) mixed design. The performance with NOEM was measured in composite error score which represents a combination of the debt and number of activists in the region. A lower composite error score indicates that there is more stability in the nation overall.

Table 5 - List of Variables

High Control	Low Control
Initial police forces	Initial police forces
Jail term	Jail term
Stimulus	Stimulus
	Government stimulus spending %
	Government services spending %
	Government infrastructure spending %
	Police forces goal
	Tax rates

In Experiment 1, participants were asked to manipulate the variables in their assigned condition to lower the national debt and the total number of activists in the DRC. Participants were asked to complete twelve practice trials and three test sessions with the NOEM interface. The control variables were selected from the total set of variables available in the model to maximize the degree of correlation the models had on the outcome of the manipulations. For example, in the high variable control condition (3 variables), the selected variables had the highest correlation with the model outcome and exerted a high level of control on the outcome and the model. These variables, due to their high correlation with the model are also considered to be highly salient as to the outcome of the model. Participants also completed an adapted version of the questionnaire developed by Berry & Broadbent (1984, 1988) used to measure whether explicit learning and analytical decision making were occurring with the decision support tool. We predicted that when the control level was high (and inputs were more salient), participants would demonstrate a better ability to control the model as compared to conditions where they were asked to control variables that were less influential on the outcomes of the model, i.e. their inputs were less salient to the outcome of the model. Figure 8 provides an overview of the procedures and variables that were investigated in Experiment 1.

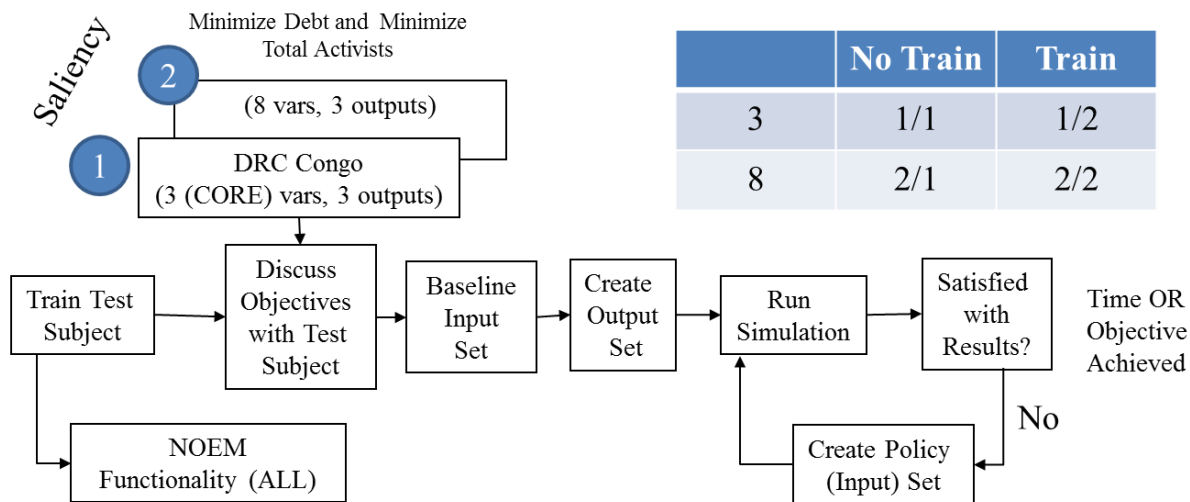


Figure 8 - Experiment 1 Overview

Figure 9 depicts the results from Experiment 1 in the mean composite error score for the two variable control conditions and the three test phases. Error bars represent +/-1 Standard Error of the Mean (Arrow depicts baseline score from the model). As shown in Figure 1, we found that participants learned to interact with the complex, stochastic decision support model without any direct instruction on the interactions within the model and no feedback to guide their learning. Thus, participants were able to use the decision support tool to develop intuitive decision making skills with the model. The learning was not evident in an explicit, analytically driven questionnaire designed to measure explicit knowledge (replicating methodology of Berry & Broadbent (1984, 1988)). Therefore, even though learning was demonstrated with NOEM, it was not able to be explicated into a form where participants could predict the mathematical outcomes of certain sets of input variables on the composite error score.

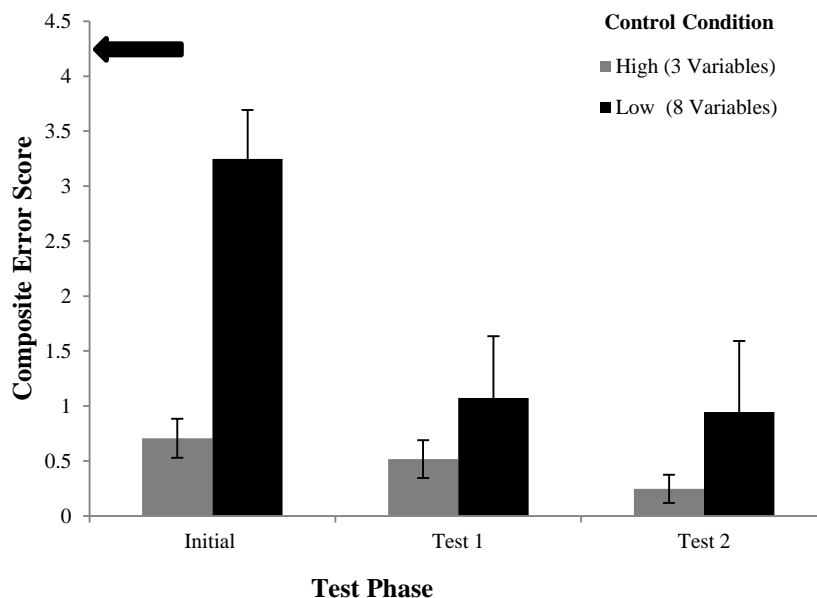


Figure 9 – Results from Experiment 1

Overall, the results of the first study demonstrated that participants were able to learn to use the NOEM interface and interact with the model with little to no guidance or feedback. Participants also demonstrated higher levels of learning with a low level of control over the interrelationships within the model than with the higher level of control. However, the group with the higher level of control also had lower scores because of the amount of control that the variables exerted on the underlying model. The group presented with the high level of control did show a trend towards decreasing composite error; although, this trend was non-significant. Experiment 1 was designed to assess the effect of levels of model control and the development of analytical and intuitive decision making. Experiment 2 was designed to assess the effects of different training regimes on learning within the NOEM tool.

3.2. Experiment 2- Training Analytical and Intuitive processes with NOEM

Experiment 2, assessed the effects of different training regimes on the development of decision making expertise with NOEM. The objective of this study was to collect some preliminary data to determine the ideal training combination for investigation of the order of analytical and intuitive training and to assess the development of decision making expertise for Experiment 3. To this end, participants were trained in either an intuitive, explicit or storyboard condition. The storyboard condition was added to assess whether we could get better learning from the standard, passive intuitive decision making conditions or whether grounding the learning in a story would create better conditions for the intuitive learning conditions. It was predicted that intuitive training would outperform either the analytical or the storyboard training due to the inherent complexity of the underlying model (DR Congo) for NOEM.

The study was a 3 between-subjects (training regime—intuitive, explicit, storyboard) x 3 within-subjects (test sessions) mixed design. This experiment used the eight variables from the low control condition in the first experiment. Participants in the intuitive condition received no feedback prior to start of the experimental trials; those in the explicit condition completed a pre-experiment variable parameter questionnaire with feedback prior to the experimental trials. All of the variable parameter questionnaires were adapted from Berry & Broadbent (1984, 1988). Participants in the storyboard condition were given a PowerPoint Presentation before the start of the experimental trials, framing the activities and training with NOEM as being the aftereffects of an earthquake in the region with the participants' task to help repair the economy by determining the effect of the different policy options. Figure 10 provides an overview of the process.

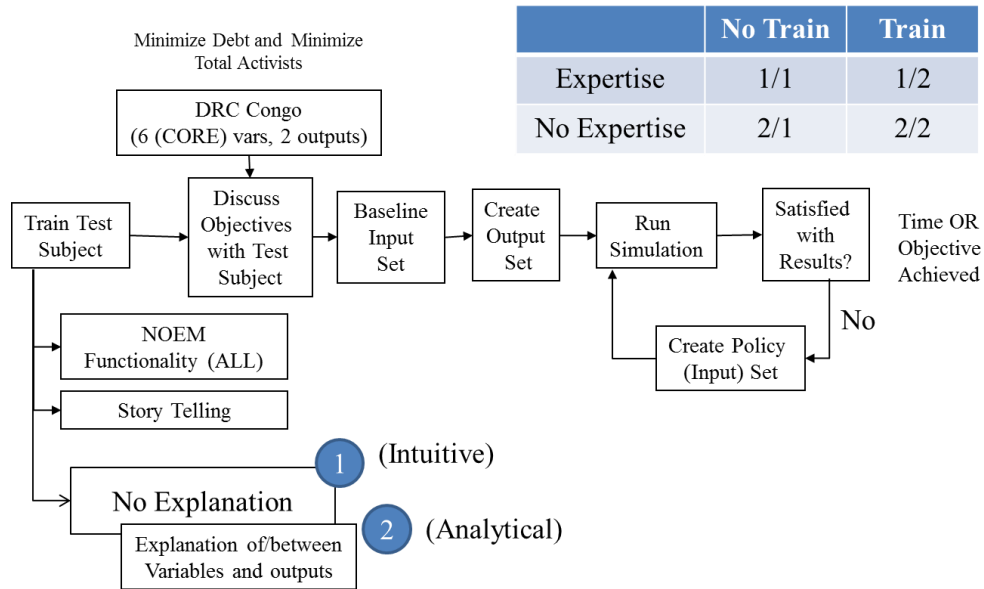


Figure 10 - Experiment 2 Overview

After the pre-experiment tasks were completed, participants were given background information on the DR Congo scenario/problem set and instructions for the experiment. Next, participants were shown how to use the NOEM software and were given directions for the first test trial. Again, the experimenter assisted the participant with the NOEM interface but did not provide any feedback regarding their task performance. After the final test, participants completed the post-experiment relating concepts Pathfinder task, followed by a post-experiment variable parameter questionnaire. Participants also completed a brief interview with the experimenter regarding their strategies and decision making. Lastly, all participants completed the usability questionnaire.

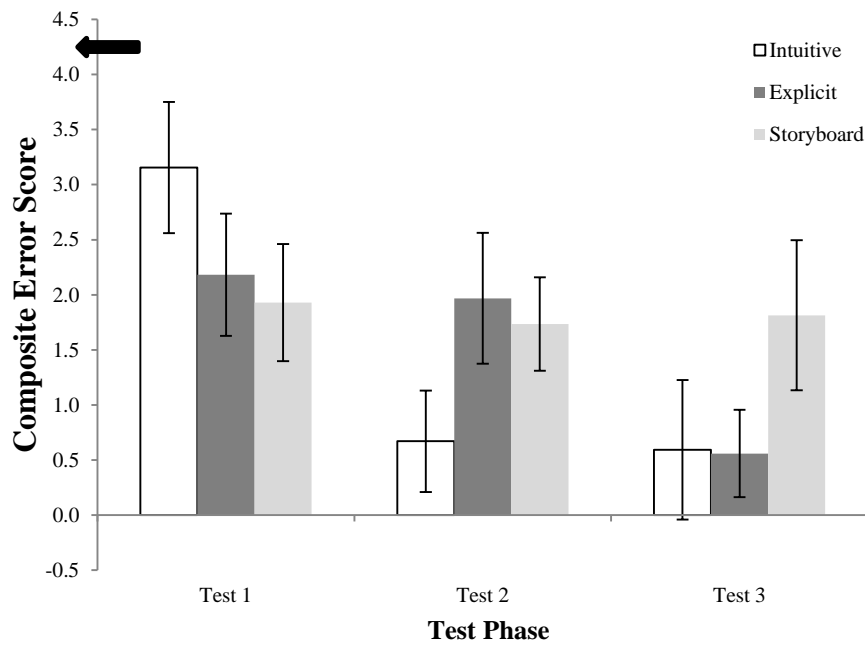


Figure 11 - Results from Experiment 2

Figure 11 depicts the results from Experiment 2 in the mean composite error score for the three training conditions and the three test phases that each of twenty four participants completed. The results indicated that in each condition performance did improve below that of the baseline model conditions. The greatest amount of learning took place for the intuitive learning condition, followed by the explicit condition. The storyboard condition did not exhibit much change across the three test phases even though initial performance was better than both the explicit and implicit learning conditions. The pathfinder data has not yet been analyzed to determine whether there was an effect of the learning manipulations on the associative knowledge of participants.

Experiment 2 found that participants demonstrated greater levels of learning with the implicit and explicit learning conditions. Initially, participants in the storyboard condition demonstrated lower performance (as shown in Figure 11), their performance did not continue to decrease over time to the same levels as either the implicit or the explicit conditions. Therefore, the implicit and explicit training conditions were chosen for the investigation of analytical and intuitive decision making and development of expertise in Experiment 3.

3.3 Experiment 3- Analytical and Intuitive Processes and the Development of Decision Making Expertise

Experiment 3 assessed the effects of the order of analytical and intuitive training on the development of decision making expertise. Klein (2008) suggests that recognition-based decision making is a combination of both analytical and intuitive decision making processes. However, the order and the regime for the development of combined analytical and intuitive decision making processes is controversial. Sun, Slusarz and Terry (2005) argued that intuitive knowledge should be developed first and then followed by analytical knowledge, yet Reber, et al. (1980) suggested the opposite. There is not a consensus on how best to combine the two processes when developing training regimes for using decision-support tools. Furthermore, it is unknown what the ideal training regime is especially concerning whether the order of the development of these two decision making processes will affect expertise development. Pretz (2008) found that the intuitive process was appropriate for novices whereas the analytical process was appropriate for experienced individuals. This finding suggests that the training regime for complex decision-support tools should be adjusted, from an emphasis on the intuitive process to emphasis on the analytical process, as expertise is developed. The data collection for Experiment 3 is still underway. Data analysis will be completed after data collection has been completed.

In Experiment 3, two groups of participants were trained to use NOEM across two sessions. This study was a 2 between-subjects (expertise—1 vs. 2 sessions) x 2 (training order—intuitive versus explicit first) x 3 within-subjects (test sessions) mixed design. Two groups of participants were trained with either analytic, explicitly based instruction or intuitive, unsupervised instruction. The analytic training condition consisted of a set of questions similar to the questions used in the questionnaire used in Experiment 1 before beginning their training. This set will provide feedback in the form of the correct answer. After finishing the questions, participants are provided, detailed descriptions of the variables and the interrelationships within the variables prior to beginning any interactions with NOEM. In contrast, the intuitive training condition consists of the questionnaire with no feedback. Participants completed the

questionnaire and then initiate interaction with the interface. Figure 12 provides an overview of the process.

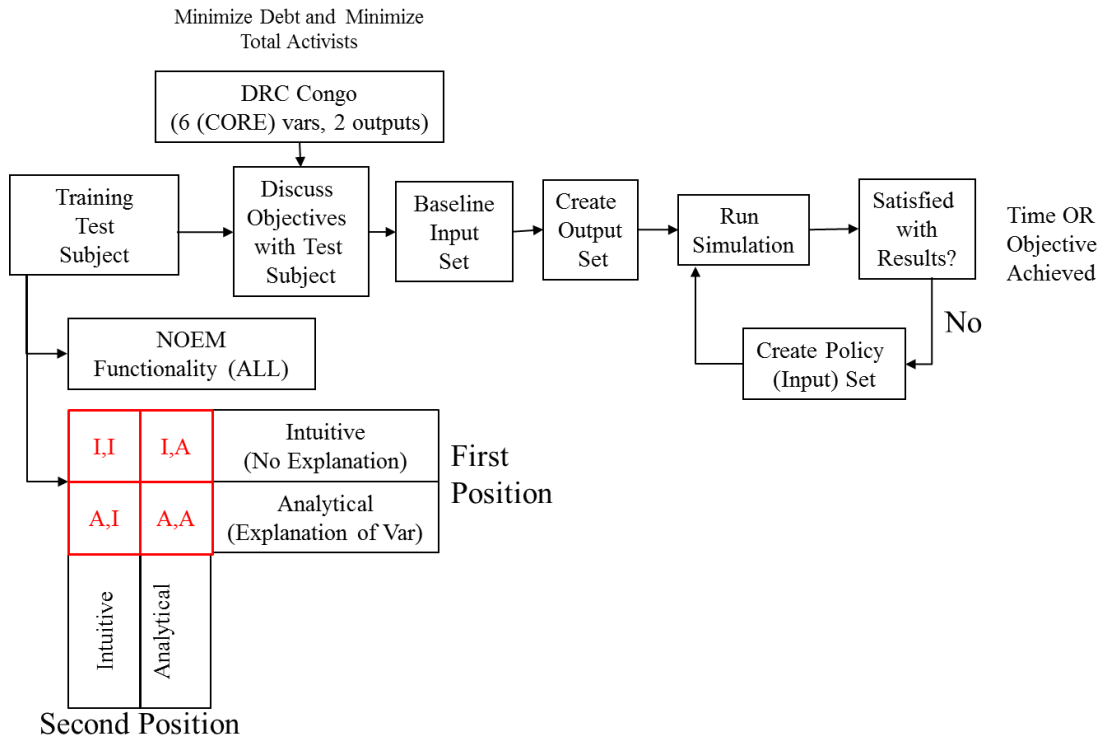


Figure 12 - Experiment 3 Overview

Participants in the single session group completed one session of either analytical or intuitive training consisting of 12 training sessions and 3 test sessions. Participants in the expertise group completed either an analytic or an intuitive condition first consisting of the 12 training and 3 test sessions. The analytic first condition was given a set of questions similar to the questions used in the questionnaire from Experiment 1 prior to beginning their training session. These questions will provide feedback in the form of the correct answer. Participants in the intuitive first condition will complete the questionnaire without feedback before beginning their training.

3.4 Summary

During this effort we studied three factors hypothesized to effect learning to use complex decision support tools: (1) saliency of underlying relationships among variables modeled by the tool; (2) expertise level of the decision maker; and (3) order of development of intuitive and analytical processes. Overall, findings of the current research indicate that participants were able to learn to use the NOEM decision support tool interface. Furthermore, participants interacted with the interrelationships of a complex model with little to no guidance or feedback, performance was not able to be measured with an analytically derived questionnaire, and therefore, learning was largely implicit. Participants demonstrated higher levels of learning with a low level of control over the interrelationships within the model than with the higher level of model control. Additionally, participants demonstrated greater levels of learning with the implicit and explicit learning conditions. Initially, participants in the storyboard condition demonstrated lower performance (as shown in Figure 11), their performance did not continue to decrease over time to

the same levels as either the implicit or the explicit conditions. Based on the findings of Experiment 2, the implicit and explicit training conditions were chosen for the investigation of analytical and intuitive decision making and development of expertise in Experiment 3. The findings of Experiment 3 will be updated and included upon completion of data collection and analysis. It is hypothesized that an ideal combination of order of training of intuitive and analytical components of decision making will be found that will contribute to the development of expertise with decision support systems.

4. A Decision Analysis Perspective

Decisions in which multiple objectives must be optimized simultaneously occur frequently in government, military, and industrial settings. One method a decision maker (e.g., a design engineer) may use to assist in multiple response optimization situations is the application of a desirability function. The decision maker specifies the desirability function parameters so as to express his or her own preferences with respect to the objectives under consideration. An informed specification of the parameters is essential so that the desirability function accurately describes the decision maker's value trade-offs and risk preference. Misapplication of the desirability function may result in the selection of an optimal policy that is inconsistent with the stated preferences. This section examines the desirability function from a decision analysis perspective. In particular, utility transversality provides the basis for an analysis of the implicit value trade-off and risk attitude assumptions attendant to the desirability function.

A limitation of the desirability function is its failure to explicitly account for response variability. A robust solution accounts for not only the expected response, but the variance as well. Assessing a utility function over desirability as a means to describe the decision maker's risk preference produces a robust operating solution that is consistent with those preferences. This thesis examines robustness as it applies to the desirability function, using a decision analysis perspective. In particular, a robust manufacturing solution is identified for a wire-bonding process, seen often in the quality and reliability engineering design literature. An exponential utility function over desirability is applied to regression equations developed from a Box-Behnken design. Monte Carlo simulation enables specification of the robust solution.

Using decision analysis methods, this methodology is applied to a practical problem currently facing the Air Force Research Laboratory (AFRL). Contributing to AFRL's Robust Decision Making Strategic Technology Team program, this section examines robustness in the context of national policy-making in country stability situation. Different levels of diplomatic, informational, military, and economic (DIME) instruments of national policy are investigated to examine how they affect the political, military, economic, social, infrastructure, and information (PMESII) systems of a nation. AFRL's National Operational Environment Model (NOEM) serves as the basis for the analysis of a scenario involving the Democratic Republic of Congo. A D-optimal design of experiments enables identification of a robust national policy. Employment of a multi-attribute utility function that satisfies the axioms of expected utility theory ensures that the policy is consistent with the decision maker's stated preferences.

4.1 National Operational Environment Model Experiment

To illustrate the decision analysis approach to robust optimization, a stability operations policy optimization problem is investigated within the Air Force Research Laboratory's (AFRL) National Operational Environment Model (NOEM). An experiment is designed to investigate how various diplomatic, informational, military and economic (DIME) instruments of national power affect the

Democratic Republic of Congo's (DRC) political, military, economic, social, infrastructure, and information (PMESII) systems. This experiment investigates the effect 14 DIME factors have on two PMESII responses. Table 6 lists the factors with brief descriptions. Table 7 lists the units and minimum and maximum values for the experimental factors. National debt and total number of activists are the two indicators of the DRC's PMESII systems. Minimizing both responses is preferred and is indicative of a more stable government. Each design point is run for three simulated years (1095 days) with two replications. A list of random seed generators is generated in Excel and each design point is assigned a seed from this list. NOEM outputs debt and activist data for every simulated day. This experiment considers the arithmetic mean of the last 30 days of the simulation's debt and activist output as the two responses for debt and activists respectively.

Table 6 - NOEM DRC Experiment Factors and Descriptions

Factor	Description
Diplomatic	
Stimulus maximum	Maximum amount of money allocated daily to government stimulus
Stimulus Spending %	Percentage of government funding earmarked for stimulus
Government Corruption	Proportion of government income not available for use
Military	
Initial Police Forces	Police forces at the start of the simulation
Police Forces Goal	Long term goal for police forces
Jail Term	Mean jail term for arrested activist
Mean Adjudication Processing Time	Mean time spend in adjudication process
Adjudication Rate	Average rate of adjudication process
Economic	
Tax Rate	Income and production tax rate
Interest Rate	Government debt interest rate
Long Term Government Employee Share	Long term proportion of workers employed by the government
Government Wages	Mean annual wage paid to government employees
Infrastructure Spending %	Percentage of government funding earmarked for infrastructure
Services Spending %	Percentage of government funding earmarked for providing services

Table 7 - NOEM DRC Experiment Factors, Units, Minimum Values, Maximum Values

Factor	Units	Min	Max
Diplomatic			
Stimulus maximum	\$	0	6,000,000
Stimulus Spending %	%	0	100
Government Corruption	n/a	0.05	0.15
Military			
Initial Police Forces	personnel	52,500	137,500
Police Forces Goal	personnel	112,500	187,500

Jail Term	days	3	100
Mean Adjudication Processing Time	day	0.1	0.9
Adjudication Rate	per day	0.01	0.13
Economic			
Tax Rate	n/a	0.03	0.5
Interest Rate	n/a	0.0225	0.0675
Long Term Government Employee Share	n/a	0.001	0.05
Government Wages	\$	1400	4200
Infrastructure Spending %	%	0	100
Services Spending %	%	0	100

The experiment executes a D-optimal design. D-optimality focuses on good model coefficient estimation. It does this by choosing design points so that the determinant of the moment matrix M is maximized (Myers, Montgomery, et al., 2009).

$$|M| = \frac{|X'X|}{N^p}$$

where X is the design matrix, N is the number of experiment runs and p is the number of parameters (Myers, Montgomery, et al., 2009). An experimental design consisting of 139 design points with two replicates each is chosen to create a full quadratic model with cubic terms.

A deterministic value function, V , is formulated to describe the preferred relationship between the two responses.

$$V = 18.164 - 5.75 \times 10^{-10}y_1 - 1.01 \times 10^{-4}y_2$$

where y_1 is debt in dollars, and y_2 is the number of activists.

$$t(y_1, y_2) = \frac{V'_{y_2}}{V'_{y_1}} = 176,439.50$$

The value function V is describing a tradeoff where the decision maker would be willing to increase the debt by \$176,439.50 in order to reduce activists by one. This particular form of a value function assumes a constant trade-off between the two responses throughout the entire response space. Consider two examples, one with \$3 billion of debt and 500 activists, and a second with \$3 billion of debt and 50,000 activists. It is unlikely a decision maker would have the same tradeoff between debt and number of activists in these two cases. Given \$3 billion of debt, a decision maker would more likely have a higher tradeoff when faced with 50,000 activists as opposed to when faced with 500 activists. Moreover, given a particular number of activists, a decision maker would more likely have a decreasing tradeoff as debt increases.

Consider an exponential utility function assessed over V

$$U = \begin{cases} 1 + e^{-\gamma V}, & \gamma < 0 \\ V, & \gamma = 0 \\ 1 - e^{-\gamma V}, & \gamma > 0 \end{cases}$$

After assessing a utility function over value, the risk aversion with respect to the two attributes can be calculated. The objective in this experiment is to maximize V which minimizes y_1 and y_2 . Since this is a decreasing value function, the risk aversion functions with respect to these attributes change some of the following two equations:

$$\gamma_{y_i}^V = \frac{V''_{y_i}}{V'_{y_i}}, i = 1, 2$$

$$\gamma_{y_i}^U = -\gamma_V^U V'_{y_i} + \gamma_{y_i}^V, i = 1, 2$$

This value function is additive, so its contribution to risk aversion is zero.

$$\gamma_{y_i}^V = 0, i = 1, 2$$

The risk aversion of the utility function with respect to the two attributes are:

$$\gamma_{y_1}^U = 5.75 \times 10^{-10}\gamma \text{ and}$$

$$\gamma_{y_2}^U = 1.01 \times 10^{-4}\gamma$$

In cases where constant risk aversion is indicated by the utility function, one must be cognizant of saturation effect. The utility function, $= 1 - e^{-\gamma V}$, equals one as V approaches positive utility. When V is greater than $\approx 1/\gamma$, the utility function is nearly a horizontal line. This saturation effect improperly models risk preference at these higher levels of V. This limitation of the exponential utility function must be considered when selecting reasonable values of γ .

4.2 Experiment Results

After the experiment is run, the data is analyzed in Design-Expert. Ordinary least squares (OLS) regression is applied and two regression equations to predict y_1 and y_2 are developed. The model for y_1 did not pass the lack-of-fit test. However the R-Squared, adjusted R-Squared and predicted R-Squared measures are all high so this model is accepted. After a natural log transform is applied to the y_2 response this model does pass the lack-of-fit test. Its R-Squared, adjusted R-Squared and predicted R-Squared are also high. These results are displayed in Table 8.

Table 8 - Statistics for the Two Regression Equations

	y_1	$\ln y_2$
R-Squared	0.9882	0.9920
Adjusted R-Squared	0.9773	0.9847
Predicted R-Squared	0.9631	0.9686

4.3 Deterministic optimum setting

The Hooke-Jeeves (HJ) algorithm finds the maximum value and corresponding operating solution. One thousand random starting points within the experimental region are generated using the uniform pseudo-random number generator in Microsoft Excel Visual Basic for Applications (VBA). This procedure produces 657 local maxima. The set of local maxima is reduced using a K means clustering

algorithm with Matlab's kmeans function. This algorithm classifies the 657 points into K groups by minimizing the sum of squares of Euclidean distances between the points and their corresponding cluster centroid (MacQueen, 1967).

To find the proper number of clusters K, K is chosen to vary between 2 and 50 and the total sum of squared distances in each case is graphed against K to find a point where the trade-off between the sum and K is balanced. Figure 13 displays this graph. Based on this graph, 13 clusters are chosen which gives a total sum of squared distances of 788.05. The maximum value point from each cluster is chosen to represent

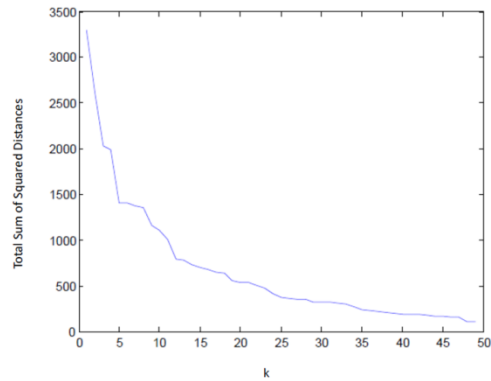


Figure 13 - Total Sum of Squared Distances in all K clusters by K

Table 9 - Local Deterministic Optimum Policies

Adjud Proc Time	Infra Spend	Govt Employ	Interest Rate	Police Goal	Tax Rate	Adjud Rate	Stim Pct	Govt Wage	Service Spend	Corrupt Pct	Police Init	Jail Term	Stim	V
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	
0.9	100	0.05	0.0675	165611	0.480	0.130	100	4188.61	0 0	0.141	52500	87	0	15.826
0.9	100	0.001	0.0225	164944	0.480	0.110	100	4200.00	0 0	0.130	52500	100	0	14.501
0.1	4.22	0.001	0.0225	166091	0.386	0.010	0	1400.00	0 0	0.050	137500	79	6000000	14.208
0.9	1.54	0.005	0.0225	166388	0.480	0.089	100	4200.00	0 0	0.125	137500	98	0	14.131
0.9	10.11	0.001	0.0225	167133	0.461	0.116	0	3985.57	0 0	0.050	137500	88	0	14.067
0.9	100	0.004	0.0225	162504	0.480	0.110	100	4200.00	0 0	0.130	137500	96	0	13.957
0.9	4.87	0.001	0.0225	169242	0.458	0.108	0	1400.00	0 0	0.050	137500	88	0	13.833
0.9	100	0.05	0.0675	169510	0.465	0.130	100	1400.00	0 0	0.138	52500	88	0	13.215
0.9	6.41	0.001	0.0225	167085	0.422	0.090	0	1400.00	0 0	0.050	137500	83	6000000	12.623
0.9	14.33	0.001	0.0225	166446	0.461	0.118	0	3970.42	0 0	0.050	52500	92	0	11.950
0.9	100	0.001	0.0225	157331	0.470	0.110	0	4200.00	0 0	0.118	137500	90	0	11.490
0.9	100	0.001	0.0225	153712	0.439	0.093	0	4101.30	0 0	0.123	137500	82	6000000	10.514
0.9	100	0.05	0.0675	155220	0.383	0.130	0	1400.00	80.11	0.129	52500	65	6000000	5.090

4.4 Monte Carlo Simulation

The residuals from the regression equations are used to estimate the marginal distribution functions for the two responses. The lower bound A for each response's Beta distribution is chosen by rounding its lowest residual down to the next integer value. The upper bound B for each response is chosen by rounding its highest residual up to the next integer value. Using these bounds, the α and β parameters are estimated using Matlab's fminsearch function. Table 10 contains the estimated parameters of the two marginal Beta distribution functions.

Table 10 - Estimated Parameters for the Marginal Beta Distribution Functions

	Residual y_1	Residual $\ln y_2$
α	17.7591	255.1359
β	7.9396	255.7977
A	-4.422E+09	-3
B	6.389E+09	6

The correlation matrix R is calculated from the residuals as shown in Table 11.

Table 11 - Correlation Matrix of the Residuals

1	-0.1912
-0.1912	1

Using Matlab's mvnrnd function and the normal multivariate copula method, a set of 10,000 sample residuals are constructed for the Monte Carlo simulation. To confirm this random sample is from the intended distribution, the Beta distribution parameter estimates and correlation matrix of the sample is calculated for comparison with the residual data estimates. Table 12 contains the Beta parameters fit to the sample.

Table 12 - Beta Distribution Parameters Fit to the Monte Carlo Samples

	Residual y_1	Residual $\ln y_2$
α	18.0711	260.9499
β	8.0698	261.6797

Table 13 contains the correlation matrix of the random sample. Both the parameters from the two marginal distributions and the correlation matrix from the random sample appear to resemble those calculated from the residuals.

Table 13 - Correlation Matrix for the Monte Carlo Samples

1	-0.1833
-0.1833	1

The Monte Carlo simulation is run using the random sample. In this model, the regression equation for y_2 contains a natural log transformation on the response. The random sample is not simply added to the

responses. The Monte Carlo simulation allows one to compute the expected utility for a particular vector x . Any point, x^* gives an expected utility,

$$\sum_{i=1}^n \left(\frac{1}{n} U_V \left(V(f_1(x^*) + \epsilon_{1i}, e^{\ln(f_2(x^*)) + \epsilon_{2i}}) \right) \right)$$

where $f_i(x^*)$ is the expected response y_i given an input of x^* ($i = 1; 2$), n is the number of random samples used in the Monte Carlo simulation, ϵ_{1i} and ϵ_{2i} are the i^{th} random samples from the distributions of the residuals for y_1 and y_2 respectively, V is the multi-attribute value function, and U_V is the utility function assessed over value. The optimization problem is to then choose a point x^* that maximizes expected utility.

The HJ algorithm is started at each local maxima displayed in Table 9. Risk preferences are modeled by varying in the appropriate utility function in Equation 5.4. Table 14 contains the levels of γ chosen and the corresponding risk preference with respect to y_1 and y_2 .

Table 14 - Levels of Sampled in Monte Carlo Simulation

γ	$\gamma_{y_1}^U$	$\gamma_{y_2}^U$
-0.01	-5.751E-12	-1.015E-06
0	0	0
0.01	5.751E-12	1.015E-06
0.02	1.150E-11	2.030E-06

Table 15 displays the robust solutions found when $\gamma = \square 0:01; 0; 0:01; 0:02$ along- side the deterministic solution. The robust solutions when $\gamma = 0; 0:01; 0:02$ are equal. They differ slightly from the deterministic solution and the robust solution when risk seeking behavior is modeled ($\gamma = 0:01$).

Table 15 - Compare Deterministic Solution with Robust Solutions for $\gamma = 0.01; 0; 0:01; 0:02$

	Deterministic Solution	Robust Solutions	
		$\gamma = -0.01$	$\gamma = 0, 0.01; 0.02$
x_1	0.9	0.9	0.9
x_2	100	100	100
x_3	0.05	0.05	0.05
x_4	0.0675	0.0675	0.0675
x_5	165481	165473	165473
x_6	0.47811	0.47814	0.47806
x_7	0.13	0.13	0.13
x_8	100	100	100
x_9	4188.76	4188.84	4188.84
x_{10}	0	0	0
x_{11}	0.14081	0.14081	0.14081
x_{12}	52500	52500	52500
x_{13}	87	87	87
x_{14}	0	0	0
E(V)	15.8259	15.8259	15.8259

This case illustrates a limitation in this method of finding a robust optimum solution. The methodology assumes constant variance in the response noise over the experimental region. The Monte Carlo simulation inputs noise to the responses, y_1, \dots, y_n , and calculates a robust solution by examining how this noise affects the multi-attribute value function. To illustrate this, consider the example robust solution displayed in Figure 14.

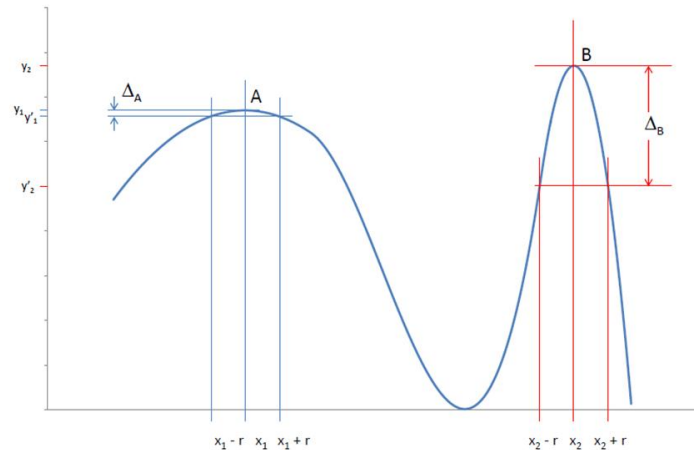


Figure 14 - Illustration of Robust Solution [16]

Here, the horizontal axis represents the responses (y_1, \dots, y_n) and the vertical axis represents $V(y_1, \dots, y_n)$. By properly specifying the decision maker's risk preference in the utility function $U_V(V(y_1; \dots; y_n))$, the proper robust solution can be found. This method is dependent on the form of the value function in use. The value function illustrated in Figure 14 has curvature present. The value function used in this situation is an additive value function. It exhibits no curvature. Therefore this method will choose a robust solution at or near the optimum.

These three points are input into NOEM with 10 replicates run for each point to test the prediction of these optimum solutions. Table 16 contains the expected values and 95% confidence intervals for these three points.

Table 16 - Average V and 95% confidence intervals from 10 replicates in NOEM for the three optimum points

	Deterministic	Robust Solution $\gamma = -0.01$	$\gamma = 0, 0.01, 0.02$
E(V)	0.6577	0.7156	0.1472
95% Conf Int	[0.5837, 0.7316]	[0.6073, 0.8238]	[-0.1963, 0.4907]

The expected values ($E(V)$) displayed in Table 16 do not fall within the 95% confidence intervals found by running these points in NOEM. This is most likely due to no design point from the experiment lies near these calculated optima. A space-filling design such as a Latin hypercube design may present better results due to the experiment design points being more uniformly spread throughout the experimental region. Augmenting this D-optimal design with additional runs may also have presented a better model of the response surface. Moreover, a Kriging model of the system would most likely have described the surface better than a regression model. A Kriging model makes no assumption about the form of the

underlying system response. Moreover, applications of Kriging models exist that model either variance homogeneity or variance heterogeneity (Kleijnen and van Beer, 2005).

4.5 Summary

A decision analysis method of calculating a robust optimum solution using a utility function assigned over a value function in a Monte Carlo simulation was applied to a national policy-making scenario within NOEM. The usefulness of this method depends on the form of the value function. Since random noise is assumed to be constant throughout the experimental region, certain value functions (e.g., an additive value function) are affected uniformly by this random noise. When the value function exhibits curvature (e.g., the desirability function), the effect noise has on the variance in the value function can be measured and a robust optimum solution can be found that is affected by changing the risk preference described by the utility function. Additional research in this area includes analyzing other multiple response optimization functions. Any multiple response optimization function makes implicit and explicit assumptions regarding the decision maker's preferences. These assumptions should be properly analyzed so that the decision maker can make informed decisions.

5. Measures of Robustness

What makes a good decision? Is it the “best” one, one that meets a given objective or the one that is the best one that has the greatest chance of succeeding? In Section 5 we look at a number of metrics that in their own right allow us to compare the various policies against the given objective, but can a single metric define “best” or is it a combination of metrics? In this section we present a number of metrics and one again use our DR Congo scenario to demonstrate what insight each metric can provide us in choosing the “best” solution set.

Given an environment defined as a set of inputs, a set of processes that model the environment and a set of outputs, we would like to find a set of potential solutions for a given objective or target goal that will effectively and quickly achieve the goal(s). But just because a solution achieves a given goal and it might be the closest to achieving the goal, it might not be the overall best. Optimality is not the only measure. Others measures include: robustness, resiliency, responsiveness and longevity. Given a set of potential solutions we first define what we mean by optimality. Our function is highly complex and composed of multiple input variables that provide a given set of outputs. The inputs can be interdependent while the optimum solution might not in all cases provide an optimum solution for each output. Simple stated our function is defined by:

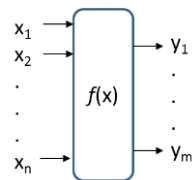


Figure 15 - Basic Model

Since we are mapping a multi-dimensional space into a second multidimensional space, a simple metric for optimality is using the n-dimensional Euclidian distance. The goal or objective function represents the center point and each solution is compared to it. The distance between the two is then used

to rank each solution. The solution with the lowest value (or closest to the objective) is considered the “best”. Any or all solutions that meet the objective are considered part of the solution set. Figure 15 portrays our distance measure in a three dimensional space.

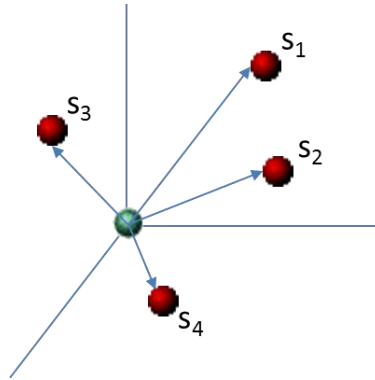


Figure 16 - Sample State Space

In our example the ranked solutions would look like (s_4, s_3, s_2, s_1) . Figure 17 summarizes the process to this point.

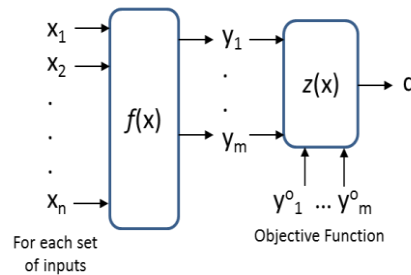


Figure 17 - Function with Comparison

Now that we have a set of potential solutions with some order we would like to determine how they stand up with respect to the other four dimensions (robustness, resiliency, responsiveness and longevity). Our first metric we will look at is robustness. In many cases the best or highest ranked solution can be a point solution. That is to achieve the given solution the values must be exact and cannot deviate. In reality it is nearly impossible to guarantee such accuracy. So the question is “How we can plan for such a situation?” This is where our first metric, “Robustness” comes into play. Given a function, $f(x)$ that maps the set of inputs to their respective outputs and our comparison metric as described above we would like to evaluate the effects that changes in an input has on the overall outputs and thus our goal.

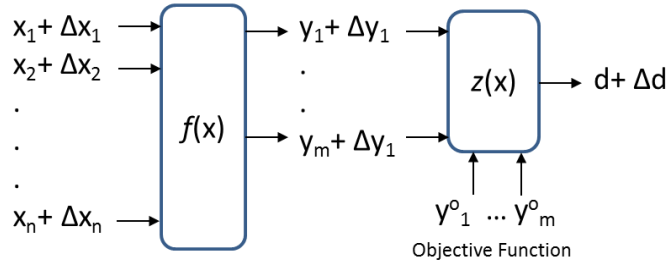


Figure 18 - Defining Robustness

Formally we define robustness as:

A solution is said to be robust if changes in the inputs minimally affects the differences in either the outputs or distance measure. How robust will depend upon how much the inputs can be changed before some change in either the output or distance is seen

We say that solution s_1 is more robust than solution s_2 if for all changes in the selected inputs the difference in the outputs or computed distance measure is less

Robustness assumes that all factors within the environment will stay the same and within the bonds identified as the deltas but what happens when an unexpected perturbation occurs? Will the solution be able to recover or will we need to find another? We call this metric, “Resiliency” (Figure 19).

We say that a solution s_1 is resilient if it has the ability to recover from a perturbation in one or more of the input variables. How resilient is measured by how quickly the system recovers to the original expected measure

We say a solution s_1 has a greater resiliency than solution s_2 if the amount of time it takes s_1 for the system to recover (i.e., get back to meeting or exceeding the stated objective) is less than s_2 .

Given A Solution (s_1):

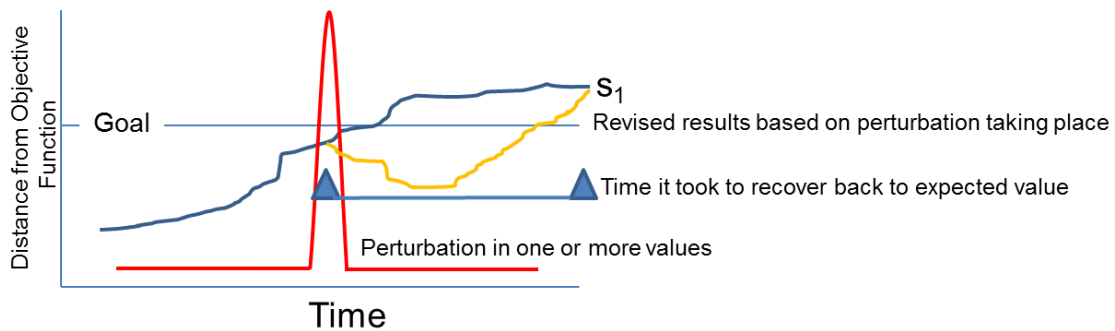


Figure 19 - Resiliency Example

The next question is how quickly can we achieve or get to our objective? We use the metric “Responsiveness” to provide us with this answer (Figure 20). We formally define responsiveness as:

We say that a solution s_1 is more responsive than solution s_2 if the time to achieve the stated objective for s_1 is less than s_2

We say a solution s_1 has greater longevity than s_2 if the amount of time that s_1 meets or exceeds the stated objective is greater than s_2

Given Two Solutions (s_1 & s_2):

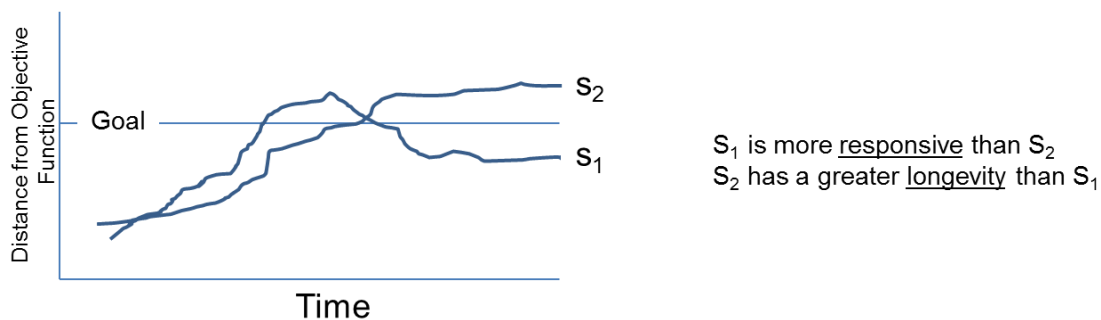


Figure 20 - Responsive vs. Longevity

Just because a given solution gets one to the objective the soonest it does not say the solution is good for the long term. In some cases, we can create additional problems by pushing to an objective sooner rather than slowly getting there. To measure this case we introduce another metric which we call “Longevity”. We formally define longevity as:

We say a solution, s_1 has greater longevity than s_2 if the amount of time that s_1 meets or exceeds the stated objective is greater than s_2

5.1 Methodology

Through preliminary work we found several challenges to the implementation of the proposed metrics in NOEM. First, the number of input variables is large, which induces a vast design space to be explored. For example, with 15 variables and each taking one of 5 discrete values there are over 30 billion combinations, so even a sampling plan with 1,000 design points accounts for only 0.000003% of the total design space. This “curse of dimension” phenomenon forbids any detailed exploration of a NOEM model. Second, the variability existed in output variable cannot be easily explained by inputs. This is a common problem in the model of system dynamics. Third, the response is a function of time, thus we need a smoothing function to represent the response. To overcome these difficulties, we will employ spline smoothing and kriging modeling tools to build meta-models to facilitate our analysis.

5.1.1 Spline smoothing

Smoothing by spline functions is widely used in functional data analysis. For data as presented in Figure 3, we want to introduce a smooth curve that can describe the global trend, as well as local features, such as the quick ascent from Day 1 to Day 3 and then a dip at around Day 50. Obviously, linear regression or polynomial regression cannot fit this type of curve well. We use a set of functional building blocks $\phi_k(t)$, $k = 1, 2, \dots, K$ called basis functions, which are combined linearly; i.e.,

$$y(t) = \sum_k c_k \phi_k(t),$$

where c_k are coefficients for regression. For noncyclical curve, we choose spline basis functions. Splines are piecewise polynomials, and typically the polynomials with order 3 are used as they are more flexible for fitting twists and turns of local features. We perform the curve fitting using the `fda` package in R[3,4].

5.1.2 Kriging model

NOEM is a computer model for modeling a nation-state's environment. The Kriging model is good at approximating a complex computer model because (1) it is flexible enough for constructing complicated response surfaces in a high-dimensional design space with a few parameters; (2) it considers the spatial correlation between responses when interpolating new design points; and (3) it provides a faster and cheaper surrogate model (meta-model) for the expensive computer model when the model-based optimization and sensitivity analysis are needed. A prediction of a kriging model is a realization of a regression model plus a random process; i.e.,

$$\hat{y}(x) = f(\beta, x) + z(\theta, x),$$

where $f(\beta, x)$ is a regression model with β being the regression coefficients, and $z(\theta, x)$ is the random process with θ being the parameters defined in correlation matrix. We use a MATLAB package, DACE [5,6], to estimate these parameters and to find the predictor $\hat{y}(x)$. Our analysis follows these steps:

1. Construct plots of the response variable at the end of year one versus 15 design variables. Identify the design variables that have significant correlations with the response variable.
2. Build the kriging model for the response variable at the end of year one. Perform model-based optimization and sensitivity analysis.
3. Plot function responses versus time. Define the goal value for response variable and find the responsiveness and longevity for each simulation run.
4. Build the kriging models for the responsiveness and longevity measures of corresponding simulation runs, and perform model-based optimization and sensitivity analysis.

5.2 Results

First, we focus on the response at Day 365 and investigate what policy sets can bring out desirable outcomes. Table 17 lists the ranges of the input and output variables. As there are 15 input variables, we plot the scatter plots of them versus the response (y_1 , the distance measure at Day 365) to show the correlation between them. A strong positive correlation is seen between x_2 and y , but not seen on other input variables (see Figure 20). However, because a scatter plot is a projection of observations from

multidimensional space to a two dimensional chart, the interactions of input variables or other complicated relationships between input and output variables cannot be revealed.

Table 17 - Ranges of input and output variables

Variable	Minimum	Maximum
x_1	0.01	0.13
x_2	0.05	0.15
x_3	0	1
x_4	0	1
x_5	0	1
x_6	1400	4200
x_7	52500	137500
x_8	0.0225	0.0675
x_9	3	100
x_{10}	0	0.05
x_{11}	0.1	0.9
x_{12}	0	0.2
x_{13}	112500	187500
x_{14}	0	6000000
x_{15}	0.03	0.5
y_1	0.0058	1.0477

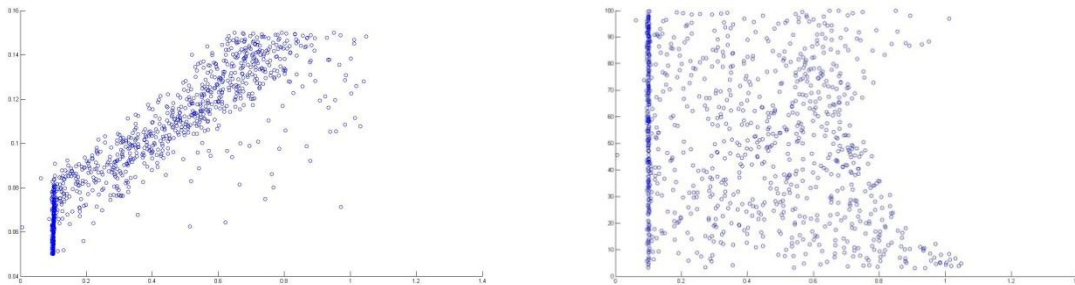


Figure 21 - Scatter plots of x_2 vs. y_1 (left) and x_9 vs. y_1 (right).

Thus, we proceed to a multiple regression analysis to identify significant input variables, which are highlighted in Table 18. Based on t tests, ten (10) variables are identified to be significant, although they do not show up in the scatter plots, and they are x_2 , x_3 , x_7 - x_{12} , x_{14} , x_{15} .

Table 18 - Multivariate regression analysis of distance measure at Day 365

Predictor	Coef	SE Coef	T	P	VIF
Constant	-0.41208	0.02912	-14.15	0.000	
x_1	-0.05319	0.07812	-0.68	0.496	1.017
x_2	8.10698	0.09359	86.63	0.000	1.014
x_3	-0.020223	0.009354	-2.16	0.031	1.012
x_4	0.012311	0.009393	1.31	0.190	1.020
x_5	-0.005817	0.009342	-0.62	0.534	1.009

x_6	0.00000264	0.00000334	0.79	0.429	1.011
x_7	0.00000094	0.00000011	8.53	0.000	1.018
x_8	0.6046	0.2087	2.90	0.004	1.019
x_9	-0.00244001	0.00009689	-25.18	0.000	1.022
x_{10}	-0.7269	0.1910	-3.81	0.000	1.012
x_{11}	0.02838	0.01175	2.41	0.016	1.021
x_{12}	0.39107	0.04660	8.39	0.000	1.005
x_{13}	-0.00000009	0.00000012	-0.70	0.483	1.015
x_{14}	0.00000000	0.00000000	2.77	0.006	1.019
x_{15}	-0.05231	0.01992	-2.63	0.009	1.016

We next performed functional data analysis on all distance measures over 365 days for each simulation run. As previously described, a typical curve shows a quick increase from the starting day and then a gradual decrease and tails off. The smooth curves of all simulation runs are shown in Figure 22, where one can see that in only a few simulation runs the distance measure can be greatly reduced over the course of one year, which indicates that most policy sets that were tried in simulation runs had failed to reach the goal of reducing the number of activists and reducing debts.

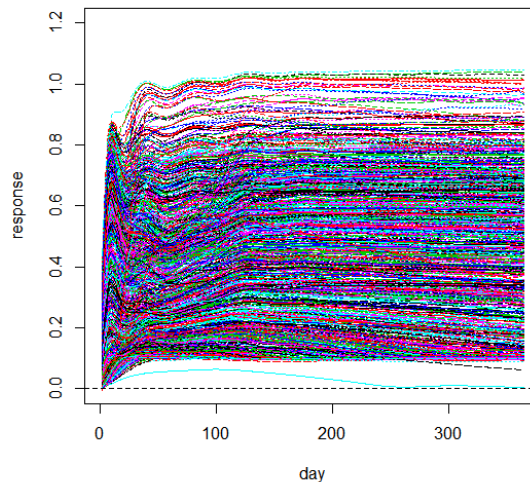


Figure 22- Functional responses of 1000 simulation runs

To study the responsiveness and resiliency of these policy sets, we defined two response variables – y_2 (Time), the date of the response curve goes below 0.1 during its gradual decreasing period, and y_3 (Duration), the days that the curve stays below 0.1. Here, we choose the distance value of 0.1 as the goal for the DR Congo scenario. Clearly, the response variable y_2 indicates the responsiveness of the distance measure to a policy set. The smaller is y_2 the more responsive is the policy performance. The variable y_3 represents the longevity of a policy set, and the larger is y_3 the more longevity is the policy performance. To study whether the policy performance is robust to random disturbance, we may conduct a sensitivity analysis of y_2 and y_3 based on their metamodels. That is, we can tell if a policy set produces a resilient performance to policy variable disturbance by measuring the deviations of y_2 and y_3 when small

perturbations are introduced into the inputs. Figure 23 gives the smooth curves fitted for simulation run 160, 984 and 142, the three runs that provide the best performance measures at Day 365.

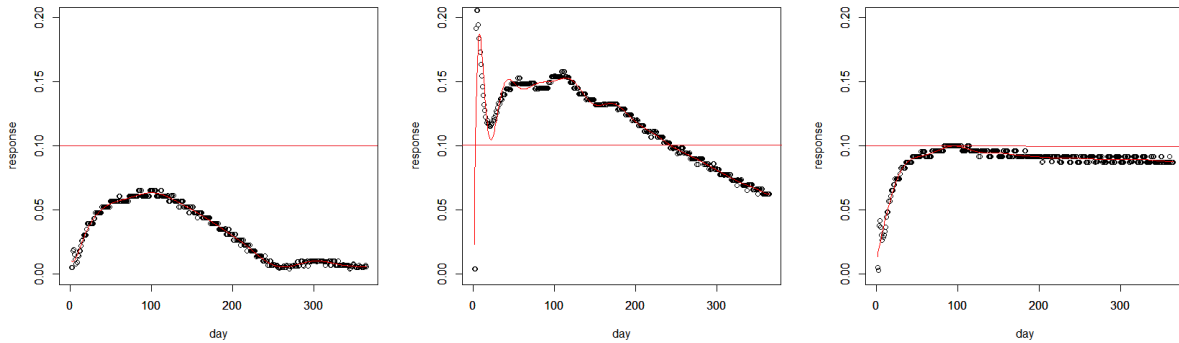


Figure 23 - Response curves fitted for simulation runs 160 (left), 984 (middle) and 142 (right).

Using these fitted curves we find the time (y_2) that the distance measure will first reach 0.1 after the initial rising period. If during the simulation run the distance measure has never passed 0.1, then we regard y_2 to be 1 and y_3 to be 365. There are 178 runs that have its responses reached the goal at some points. The ranges of input variables from these 178 runs are presented in Table 19. Comparing this table with Table 17, one can see that they cover the design space of nearly the same size, which implies that there is no particular input variable that dictates the responsiveness of the distance measure.

Table 19 - Ranges of input variables of 178 simulation runs

Variable	Minimum	Maximum
x_1	0.01	0.13
x_2	0.05	0.15
x_3	0	1
x_4	0	1
x_5	0	1
x_6	1400	4200
x_7	53244	136614
x_8	0.0225	0.0672
x_9	3.522	98.73
x_{10}	0	0.05
x_{11}	0.1	0.9
x_{12}	0	0.2
x_{13}	112942	187195
x_{14}	818	5991640
x_{15}	0.03	0.4943

5.2.1 Kriging models and sensitivity analysis of the DR Congo scenario

Previously, we have identified 3 alternative policy sets (Runs 160, 984 and 142) that will produce acceptable outcomes at the end of the first year. The first question to answer is “Which of these alternatives is less sensitive to disturbance on the input variables?” it will be a robust solution when a

decision maker is facing uncertainties in policy parameters. As we have identified 10 significant variables, in the following analysis we will utilize these variables to build a kriging model for the distance measure at Day 365 (y_1).

A kriging model consists of two components – a deterministic function, which is typically a polynomial regression model, and a stochastic function, which is a spatial correlation model. The regression models that we tried to fit include constant, linear and quadratic models, and the spatial correlation models include Gaussian model and exponential model. To choose the best model, we considered a 20-fold cross validation, in which the original data set was separated into 20 groups with each group having 50 observations, then one group was selected as the test group while data in other groups were used to fit a model. The results, as shown in Table 9, indicate that the second order polynomial with Gaussian or exponential correlation model provides the highest prediction power. We choose the Gaussian correlation model for convenience. A surface plot of this model, projecting on two input variables (x_2 and x_9), is given in Figure 24.

Table 20 - Cross validation of multiple kriging models for the response on Day 365

Model	$R^2_{\text{prediction}}$
First order polynomial + Gaussian correlation	0.878
First order with interaction + Gaussian correlation	0.878
Second order polynomial + Gaussian correlation	0.898
First order polynomial + exponential correlation	0.878
Second order polynomial + exponential correlation	0.898

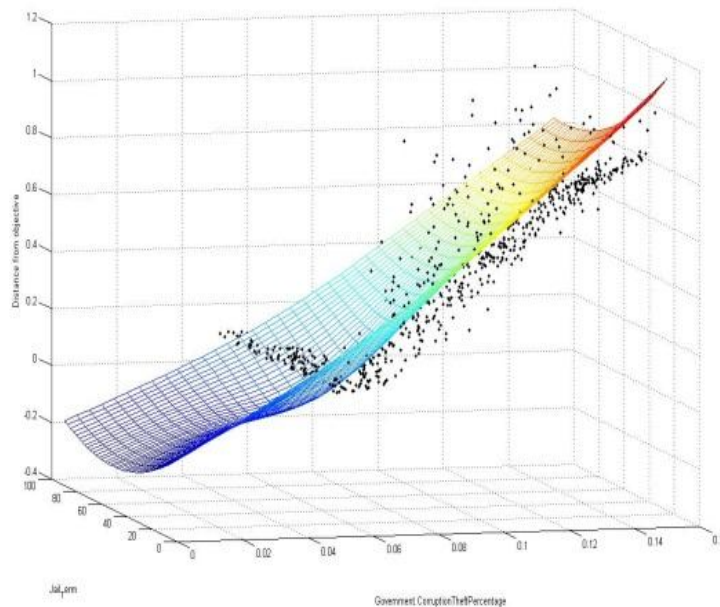


Figure 24- Response surface of the kriging model of distance measure at Day 365

Around each of the three best policy sets, we construct a hypercube in the 10-dimensional design space with the current policy set at the center. The length on each dimension of the hypercube is $1/100^{\text{th}}$ of the range of the corresponding variable. Within this hypercube, we randomly sample 10000 points using the LHS design and evaluate the response values of these samples based on the kriging model. The deviations of the response values of these samples to the response value of center point are used to assess the sensitivity of response surface to a small disturbance of input variables. We calculate the mean square deviation as defined by

$$MSD = \frac{1}{10^4} \sum_{i=1}^{10^4} (\hat{y}_i - y_0)^2,$$

where \hat{y}_i is a sample value and y_0 is the response value at the center of hypercube. Table 21 provides the MSDs, as well as the range of response, for the three best policy sets. One can see that the simulation run 142 has the least variation when there is a small random perturbation on input variables, thus it is the most robust solution for y_1 .

Table 21 - Mean square deviations and ranges of y_1 in the three simulation runs

Run	MSD	Range
160	5.3907e-6	0.0081
984	6.8642e-6	0.0091
142	4.4036e-6	0.0073

We apply the same kriging modeling and sensitivity analysis on the other two response variables – the date of the response reaching the goal value of 0.1 (y_2) and the days of the response staying below the goal value (y_3). Both of them use the Gaussian correlation model, but the polynomial regression model for y_2 is selected to be a second-order polynomial with variables $x_2, x_4, x_8, x_9, x_{10}, x_{14}, x_{15}$, and for y_3 it is a second-order polynomial with variables $x_1, x_2, x_3, x_4, x_7, x_9, x_{10}$. Comparing the three best simulation runs, it is found that the policy of Run 160 is the most responsive policy ($y_2=1$) and it has the longest sustained desirable effect ($y_3=365$); however, it is not the most resilient policy when we compare its MSD of y_3 with the other two runs. Table 22 provides the comparison of these three simulation runs.

Table 22 - Comparison of responsiveness, longevity and resiliency of three simulation runs

Run	y_2	MSD of y_2	Range of y_2	y_3	MSD of y_3	Range of y_3
160	1	56	19.65	365	1300	72.17
984	245	1697	85.95	121	64	22.15
142	98	679	53.38	268	1135	78.00

Run 984 and Run 142 are, however, not the best runs in terms of the response y_2 or y_3 . Thus, we list the three best runs for y_2 and y_3 , and study their robustness, respectively. The three best runs for y_2 are runs 160, 49 and 207, and the three best runs for y_3 are runs 160, 49 and 71. Tables 23 and 24 provide the measurements of robustness of these runs. Run 160 and Run 49 have the same responsiveness and

longevity, but clearly the results of Run 160 are most robust to small disturbance on its policy setting, thus it is more resilient.

Table 23 - Robustness of the three best runs for responsiveness

Run	y_2	MSD	Range
160	1	56	19.65
49	1	1435	78.04
207	32	18395	32.53

Table 24 - Robustness of the three best runs for longevity

Run	y_3	MSD	Range
160	365	1300	72.17
49	365	5088	154.07
71	331	8984	138.94

5.2.2 Global Optimum

Using kriging meta-models we can also search for the global optimum over the entire design space. We implemented a simulated annealing (SA) stochastic search algorithm to find the global optimums for y_1 , y_2 and y_3 . It turns out that the current solution, Run 160, is indeed the global optimum for all three responses.

Finally, we draw a spider plot with five features – distance, robustness, responsiveness, longevity and resiliency, where the distance feature is the response value of y_1 , robustness is represented by the MSD of y_1 , responsiveness is the logarithm of y_2 , longevity is the inverse of y_3 , and resiliency is represented by the MSD of y_2 . Therefore, for all features, a smaller value is better. Figure 25 is a spider plot of Run 160, Run 984 and Run 142. The scales of all features are standardized. It is clear that the plot of Run 160 covers the smallest area, so we conclude that the policy set of Run 160 provides the best solution for the DR Congo scenario and this solution is the overall “best” across all metrics compared to other alternatives.

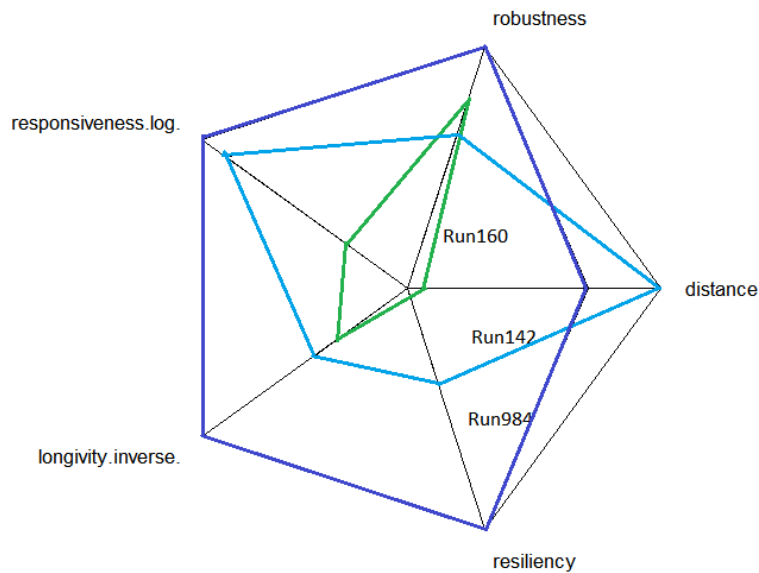


Figure 25 - Spider plot of three runs with five features

5.3 Summary

As the outputs from NOEM are typically functional outputs, we must pay attention to the definitions of parameters of interests and transfer them to some measurable response variables. In this section we demonstrate the use of functional data analysis, metamodeling and sensitivity analysis to evaluate policy sets for the DR Congo scenario. We believe that these data analysis techniques can be and should be integrated into the existing NOEM policy set analysis (PSA) tool, and they can become invaluable assets to decision makers.

6.0 Conclusion

In this chapter we presented a set of experiments that allows us to gain a better understanding as to how humans make decisions. An existing decision support tool (NOEM) was presented along with a sample problem set (DR Congo) with which we were able to develop a baseline answer. Using this work we were then able to compare how close humans can get to what an automated tool (assisted by the human) can do. We investigate Decision Analysis Techniques to investigate optimum solutions. We then used the same scenario/results to present a number of metrics (robustness, resiliency, longevity and response) that can provide insight to the decision maker as to the quality of the solution. Even though NOEM was used and a specific scenario was provided the results of this work can be easily extended to other decision support tools and the solutions/recommendations they provide.

References

- [1] AFRL/RIEA (2009). *NOEM System Description Document, Version 2.0*.
- [2] Berry, D. C., & Broadbent, D. E. (1988). Interactive tasks and the implicit–explicit distinction. *British Journal of Psychology*, **79**, 251-272.
- [3] Berry, D.C. & Broadbent, D.E. (1984). On the relationship between task performance and associated

- verbalizable knowledge. *Quarterly Journal of Experimental Psychology*, 36A, 209-231.
- [4] Evans, J.S.B.T. (2008). Dual-processing accounts of reasoning, judgment and social cognition. *Annual Review of Psychology*, 59, 255-278.
 - [5] Hammond, K.R. (2007). *Beyond Rationality: The Search for Wisdom in a Troubled Time*. N.Y.: Oxford University Press.
 - [6] Hogarth, R. (2001). *Educating intuition*. Chicago: University of Chicago Press.
 - [7] Kleijnen, J.P.C. (1997). Sensitivity analysis and related analyses: A review of some statistical techniques. *Journal of Statistical Computation and Simulation*, 57(1), pp. 111-142.
 - [8] Kleijnen, Jack P.C. and Wim C.M. van Beers. "Robustness of Kriging when interpolating in random simulation with heterogeneous variances: Some experiments", *European Journal of Operational Research*, 165(3):826-834, September 2005.
 - [9] Klein, G. (2008). Naturalistic decision making. *Human Factors*, 50, 456-460.
 - [10] Lophaven, S.N., Nielsen, H.B., Sondergaard, J. (2002). DACE, a MATLAB kriging toolbox, version 2.0. Available at <http://www2.imm.dtu.dk/~hbn/dace/>.
 - [11] MacQueen, J. B. "Some methods for classification and analysis of multivariate observations". *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*, 281-297. University of California Press, 1967.
 - [12] Milkman, K.L., Chugh, D. & Bazerman, M.H. (2009). How can decision making be improved? *Perspective on Psychological Science*, 4, 379-383.
 - [13] Myers, Raymond H., Douglas C. Montgomery, and Christine M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 3 edition). John Wiley & Sons, Inc., 2009.
 - [14] Patterson, R., Fournier, L., Pierce, B.P., Winterbottom, M. & Tripp, L. (2009). System dynamics modeling of the recognition-primed decision model. *Journal of Cognitive Engineering and Decision Making*, 3, 253-279.
 - [15] Patterson, R., Pierce, B.P., Bell, H., Andrews, D. & Winterbottom, M. (in press). Training robust decision making in immersive environments. *Journal of Cognitive Engineering and Decision Making*.
 - [16] Perruchet, P. & Pacton, S. (2006). Implicit learning and statistical learning: one phenomenon, two approaches. *Trends in Cognitive Sciences*, 10, 233-238.
 - [17] Pretz, J.E. (2008). Intuition versus analysis: Strategy and experience in complex everyday problem solving. *Memory & Cognition*, 2008, 36, 554-566.
 - [18] Ramsay, J.O., Hooker, G., Graves, S. (2009). *Functional Data Analysis with R and MATLAB*, Springer, New York, NY.
 - [19] Ramsay, J.O., Wickham, H., Gaves, S., Hooker, G. (2012). fda: Functional data analysis. Available at <http://cran.r-project.org>.
 - [20] Reber, A. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General*, 118, 219-235.
 - [21] Reber, A., Kassin, S. M., Lewis, S., & Cantor, G. (1980). On the relationship between implicit and explicit modes in the learning of a complex rule structure. *Journal of Experimental Psychology: Human Learning and Memory*, 6, 492-502.
 - [22] Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4), pp. 409-423.
 - [23] Salerno J.J., Smith J.E., Geiler W.M., McCabe P.K., Panasyuk A.V., Bennette W.D. and Kwiat A., (2013) The NOEM – A Tool for Understanding/Exploring the Complexities of Today’s Operational Environment. In: Subrahmanian V.S. (ed) *Handbook of Computational Approaches to Counterterrorism*. Springer, New York, pages 363-399.
 - [24] Schvaneveldt, R.W. (1990). *Pathfinder Associateive Networks: Studies in Knowledge Association*, Ablex Publishing Corporation, Norwood, NJ.
 - [25] Sloman, S.A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin*, 119, 3-22.
 - [26] Serman, J.D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. Boston, MA: McGraw-Hill.
 - [27] Tversky, A & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 185, 1124-1131.

Appendix D. RDM STT: Context Switching Methods to Calibrate Trust for Mission Assurance

Summary/Lessons Learned, Dr. Kirk Weigand, AFRL/Rywa

The project did not meet its goal of completing an integrating capstone experiment due to a major funding cut. The initial work on all three experiments was completed and those contributions are shown below. Due to the difficulty of collaborating across multiple technical directorates, funding and personnel needed to be carefully protected and valued at a corporate level. As lead for this sub-project, I can only surmise given the qualified personnel involved that had AFRL paid the price for this collaboration, this sub-project would have made significant headway on a difficult problem. I believe we made a good start to this work that should be continued. The complexity of the problem merits a cross-disciplinary collaboration.

Experiment 1, Dr. Gina Thomas & Ms. Krystal Thomas, 711 HPW/RHCP

Knowledge Glyphs, which were initially defined by the 711 Human Performance Wing (HPW)/RHC (Warfighter Interface Division), are advanced visualization artifacts whose sophistication has the potential to enhance information systems support to military commanders. Knowledge Glyphs are defined as *specialized icons affording users the ability to access extraglyphic information in such a form that the extraglyphic presentation is anchored with respect to the same entity (or another discrete object of reference)*. As such, a knowledge glyph both acts as a micro-interface and implements an intersection among two or more distinct referential contexts in a manner that treats the denoted entity as a juncture point. The goal of the experiment was to determine whether this method of context shifting provided an advantage when performing a strategic task.

In the experiment, participants were asked to plan a daily schedule given a list of tasks and the times required for performing each. In order to complete the schedule, participants were required to take into account not only the task times but also associated travel times and bus schedules. They were given both a geographic (map) view and a temporal (timeline) view for reference. The method for switching from one of these contexts to the other and back was varied between subjects. In one condition, participants could close one view and open another. In a second condition, participants could keep both views open as windows. In the final view, participants could select an object of interest and the new view replaced the old view in such a way that the object of interest maintained its screen location.

The experiment was planned for a minimum of 102 participants in order to yield a reasonable chance of detecting a performance difference. Four participants completed the study for the purpose of pilot testing of the experimental software; their data were discarded. In the time we had, we were only able to recruit twenty-five additional participants for the study; two of those participants failed to complete the task because they had to leave to go to work or to class. The data show extreme variability among participants, which makes it impossible to detect an effect if there were one. Additionally and unfortunately, when we looked at the data it appeared that some of them made a schedule and then thought they could do better, so they deleted it and started again. Then when the time ran out, they only had a few items scheduled, but we have no evidence of what they had completed before they deleted it. This problem adds to the difficulty of determining whether there is an effect. We would need more time to recruit closer to the anticipated number of participants to have any hope of overcoming these difficulties.

Support for Experiment 1, Dr Ron Hartung and TDKC staff

See attached report on TDKC support for RDM STT

Experiment 2, Michael Manno, AFRL/RIED

The following bullets provide a summary and overview of the work related to RDM applied to SITA.

- * Purchased 3 Dell Poweredge R415 servers that were set up a laboratory environment to develop and test the latest SITA Baseline.
- * Demonstrated SITA as part of the RDM STT, Context Switching Kickoff meeting at WPAFB RY January 2011. Provided updates to the SITA installation at Wright Patterson Air Force Base, WPAFB, RY Trust Lab in September 2011. The install was provided to test and evaluate the effects of the SITA program, the interface, and the 3 screen visualization (Past, Present and Future) as an improved visualization method.
- * Additional usability features were included in the user interface along with improved performance and stability.
- * The SITA interface was enhanced to include an alert notification feature which allows the user to update the displays when new data is detected, without automatically disrupting any current on-going analysis.
- * The SITA graph display was modified for improved user interaction.
- * Noise was also added to the AOC scenario for a more complete demonstration, increasing the observables from 39 to 474.

Experiment 3, Dr. Charlene Stokes, 711HPW/RHXS

See attached report on Experiment 3

**Appendix E. Sensors Directorate Technologies for Robust
Decision Making for Improved Mission Assurance
Project Final Report**

**Ronald L. Hartung, Ph. D. Raymund Garcia Colin
Morrow Tracey Culbertson**

**The Design Knowledge Company 3100 Presidential Drive, Suite 103
Fairborn, Ohio 45324**

August 2013

THIS IS A SMALL BUSINESS INNOVATION RESEARCH (SBIR) PHASE III REPORT.

Distribution B: Distribution authorized to U.S. Government Agencies only; Proprietary Information (DFARS SBIR Data Rights); August 2013. Other requests for this document shall be referred to AFRL/Rywa, Wright-Patterson AFB, OH 45433.

WARNING: This document contains technical data whose export is restricted by the Arms Export Control Act (Title 22, U.S.C., Sec. 2751, et seq.) or the Export Administration Act of 1979, as amended, Title 50 U.S.C. app. 2401 et seq. Violations of these export laws are subject to severe criminal penalties. Disseminate IAW the provisions of the DOD Directive 5230.25.

**AIR FORCE RESEARCH LABORATORY SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433 AIR FORCE
MATERIEL COMMAND
UNITED STATES AIR FORCE**

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

SBIR Data Rights Legend

Contract Number: FA8650-09-D-1512
Contractor Name: The Design Knowledge Company
Contractor Address: 3100 Presidential Dr., STE 103 Fairborn, OH 45324
Location of SBIR Data Rights: Pages 17-31
Expiration of SBIR Data Rights: Expires 5 years after completion of project work for this or any follow-on SBIR contract, whichever is later.

The Government's rights to use, modify, reproduce, release, perform, display, or disclose technical data contained in or computer software marked with this report legend are restricted by during the period shown as provided in paragraph (b)(Para number) of the Rights in Noncommercial Technical Data and Computer Software—Small Business Innovation Research (SBIR) Program clause (DFARS 252.227- 7018 (June 1995)) contained in the above identified contract. No restrictions apply after the expiration date shown above. Any reproduction of technical data, computer software, or portions thereof marked with this legend must also reproduce the markings. Any person, other than the Government, who has been provided access to such data must promptly notify the above named Contractor.

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

- THIS PAGE LEFT INTENTIONALLY BLANK -

Table of Contents

1.0	Executive Summary	1
2.0	Background	1
3.0	Context Switching Methods.....	3
4.0	Modeling Sensemaking.....	17
5.0	Abduction and Sensemaking Conclusions	30
6.0	References	31

Appendices

- THIS PAGE LEFT INTENTIONALLY BLANK -

Sensors Directorate Technologies for Robust Decision Making for Improved Mission Assurance

1.1 EXECUTIVE SUMMARY

The Design Knowledge Company (TDKC) was awarded a Phase III Small Business Innovative Research contract entitled “Argumentation-based Approaches to Enhance Dynamic Time Critical Decision-Making (ADTD)”. ADTD supports the Air Force Research Laboratory (AFRL), Sensors Directorate, Spectrum Warfare Division, Avionics Vulnerability Mitigation Branch (RYWA). The goal of the ADTD program is to research new, innovative methods and technologies that support the AFRL’s vision for distributed, mission assured layered sensing systems and desire for universal situation awareness. The Universal Situation Awareness (USA) S&T Strategic Vector involves acquiring knowledge of enemy intention, capability, and location at all times. With multiple layers and multiple sensor architectures, there are multiple possibilities for attack on a USA view of the battlespace. The user needs capabilities that include an argumentation-based approach to enhance dynamic time critical decision-support for mission assured multilayered sensing, and intuitive visualizations that promote efficiency in the analyst’s work routine. The previous SBIR research on “XTM Course of Action & Argumentation Technology (XCAT)” laid the groundwork for the more advanced ADTD research.

ADTD Task Order 04 is “Decision Support and Interdependency Visualization” includes a sub-project for research on technologies for AFRL Robust Decision Making initiative.

AFRL’s Robust Decision Making (RDM) Strategic Technology Team (STT) explored a range of new decision technologies for the warfighter’s tactical, operational, and strategic decision making in command and control, nation stabilization, air operations center, and remotely-piloted aircraft mission contexts. The RDM STT was proposed to address the following objectives:

- Develop integrated human-machine reasoning and decision processes to fight through cyber attacks
- Make strategic decision making intuitive and effective
- Incorporate and evaluate assessments of trust in complex operational-level decision systems
- Create and use a testbed for research on robust asset management and mission-level decision making

The team included scientists and engineers from multiple AFRL technical directorates (RH, RI, RY) with key scientific and technical skill sets from cognitive science, computer science and engineering, electrical and mechanical engineering, experimental and industrial/organizational psychology, mathematics, operations research, and physics. The Sensors Directorate team explicitly explored two research areas: context switching methods to calibrate trust for mission assurance, and modeling sensemaking as abduction-based inquiry for cyber effects.

2.0 BACKGROUND

- 2.1 AFRL’s Robust Decision Making (RDM) Strategic Technology Team (STT) explored a range of new decision technologies for the warfighter’s tactical, operational, and strategic decision making in command and control, nation stabilization, air operations center, and remotely-piloted aircraft mission contexts.

2.2 The STT is consistent with the AF Chief Scientist's 2010 report on Technology Horizons, A Vision for Air Force Science & Technology During 2010-2030. One of the major findings was that natural human capacities are becoming increasingly mismatched to data volumes, processing capabilities, and decision speeds and that science and technology (S&T) to augment human performance will become increasingly essential for gaining the benefits that many technologies can bring. The report recommended several Grand Challenges one of which was "Trusted Highly-Autonomous Decision Making Systems." The challenge was to "Explore, develop, and demonstrate technologies that enable current human-intensive functions to be replaced, in whole or in part, by more highly autonomous decision-making systems, and technologies that permit reliable verification and validation (V&V) to establish the needed trust in them." The Technology Horizons report stated that this challenge "*will enable technologies that can support substantial manpower cost reductions, and extend robust improved decision-making capabilities to highly stressing future applications that may involve decision time scales beyond human capacity.*"

2.3 Similarly, the June 2013 Global Horizons, USAF Global Science and Technology Vision report reinforced the need for superior decision making in an environment where there is global access to technology, worldwide connectivity, and increased access to all domains by our adversaries who will ultimately challenge our ability to dominate air, space, and cyberspace. The report notes that the combination of increasing threats, knowledge-based technologies, and fiscal constraints simultaneously demand and enable the development of resilient and innovative C4ISR game changing capabilities. One recommended game changer is the use of data analytics, neuromorphic computing, cognitive modeling, and flexible autonomy to integrate platforms, sensors, and highly trained/educated operators for superior decision making. Thus, robust improved decision making capabilities are a key technical objective central to the AF's S&T Vision for the next 20 years.

2.4 The Robust Decision Making STT was proposed to address the following objectives:

- a. Develop integrated human-machine reasoning and decision processes to fight through cyber attacks
- b. Make strategic decision making intuitive and effective
- c. Incorporate and evaluate assessments of trust in complex operational-level decision systems
- d. Create and use a testbed for research on robust asset management and mission-level decision making

The team included scientists and engineers from multiple AFRL technical directorates (RH, RI, RY) with key scientific and technical skill sets from cognitive science, computer science and engineering, electrical and mechanical engineering, experimental and industrial/organizational psychology, mathematics, operations research, and physics.

2.5 The Sensors Directorate team explicitly explored two research areas: context switching methods to calibrate trust for mission assurance, and modeling sensemaking as abduction-based inquiry for cyber effects.

2.5.1 The objective of "Context Switching Methods to Calibrate Trust for Mission Assurance" was to evaluate various methods for alternating contexts when viewing information in multiple contexts is required for decision making. This research will be used to assess the efficacy of focal context switching for use in display design. The approach was to test novel Knowledge Glyphs, define context switching, and test appropriate reliance of decision aids during the context switch. The benefit to the warfighter is improved trust and relevance of displayed information for ISR and cyber operators.

2.5.2 The objective of “Modeling Sensemaking as Abduction-based Inquiry in a Cyber effects Testbed” was to improve decision making when an operator is faced with a surprising situation. The approach was to model and test the cognitive process of abductive inference (wise hypothesis generation) for reverse engineers. Current decision making algorithms, such as Bayesian and Dempster-Shafer, are brittle and the complexity of system of systems operations often results in surprising states. The benefit of abduction technology to the warfighter is better situation understanding and better management of the decision making.

3.0 CONTEXT SWITCHING METHODS TO CALIBRATE TRUST FOR MISSION ASSURANCE

3.1 Introduction on Knowledge Glyphs

The Air Force Research Laboratory (AFRL) 711th Human Performance Wing (711HPW) Human Effectiveness Directorate (RH) is a leader in researching the use and display of meta-information. Knowledge Glyphs, which were initially defined by the 711HPW/RHC (Warfighter Interface Division), are advanced visualization artifacts whose sophistication has the potential to enhance information systems support to military commanders. Knowledge Glyphs are defined as specialized icons affording users the ability to access extraglyphic information in such a form that the extraglyphic presentation is anchored with respect to the same entity (or another discrete object of reference). As such, a knowledge glyph both acts as a micro-interface and implements an intersection among two or more distinct referential contexts in a manner that treats the denoted entity as a juncture point. Historically, the only tests of the effectiveness of these knowledge glyphs are based on specific examples within a single domain. There is a need to test the basic tenets of the knowledge glyph model in order to verify the potential for performance improvement across domains.

There are a number of open research questions regarding the effectiveness of using knowledge glyphs. The most pressing of these questions derives from the principle that, when the user wishes to view the entity in a different context, the new context will rotate in while the glyph remains the focal point of the display. Therefore, it is important to experimentally determine whether the hypothesized advantage of switching contexts without the need to shift visual attention holds in a general sense and, if so, how does that advantage change under a variety of viewing conditions and information requirements.

The Sensors Directorate (RY) and the Human Effectiveness Directorate (RH) partnered on conducting human-centered experiments on context switching for Knowledge Glyphs. The experiments were conducted in RY’s Nucleus facility. RY developed the software tool for conducting experiments and data collection on user interface concepts related to the idea of context switching. Data from the experiment will be analyzed by RH personnel and separately published. This research was initiated in 2011 on a project called 3-D Knowledge Glyphs. This project was initiated by Dr. Gina Thomas and Dr. Kirk Weigand.

3.2 The Knowledge Glyphs Project

The following objectives encompass the totality of 3-D Knowledge Glyphs.

- 1) Travel to AFRL Mesa, Arizona to meet with both government and contractor personnel who may have worked with original 3-D Knowledge Glyphs.
- 2) Conduct a “knowledge capture” of the previous 3-D Knowledge Glyph software and hardware and any documentation of the previous developed knowledge glyphs.

- 3) Develop 3-D Knowledge Glyphs in an open source code architecture that is scalable.
- 4) Conduct a demonstration with infrastructure of the 3-D Knowledge Glyph.

3.2.1 Knowledge Glyphs Methodology

The team used the following methodology to execute the Knowledge Glyphs Project.

- 1) Investigate and document previous design and prototype development.
- 2) Build a new prototype.

3.2.1.1 Investigation of Previous Prototype

In August of 2010, Chuck Anderson traveled to the AFRL facility at Mesa, Arizona and met with personnel involved with previous SGI project. The demonstration featured SGI Pixel Fusion Server displayed by a Sony4K projector. The system included several racks of hardware for video stream multiplexing, and presented video streams from multiple computers within a single 3-D environment. One of the quadrants of the screen was not working because of failed hardware. The demonstrated scenario represented an Air Operations Command environment, and showed a ground map projected onto a large rectangular area. Similar to Figure 1, there were several billboard style rectangles arranged vertically in the distance, containing images and information related to command-level objectives, global time, etc. The 3-D Knowledge Glyphs, Figure 18, represented aircraft and were displayed above the ground, animated to move along flight paths. Interactions with the 3-D Knowledge Glyphs were primarily by commands issued from a separate monitor, via controls displayed in a dialog window.



Figure 1. SGI Pixel Fusion Server.

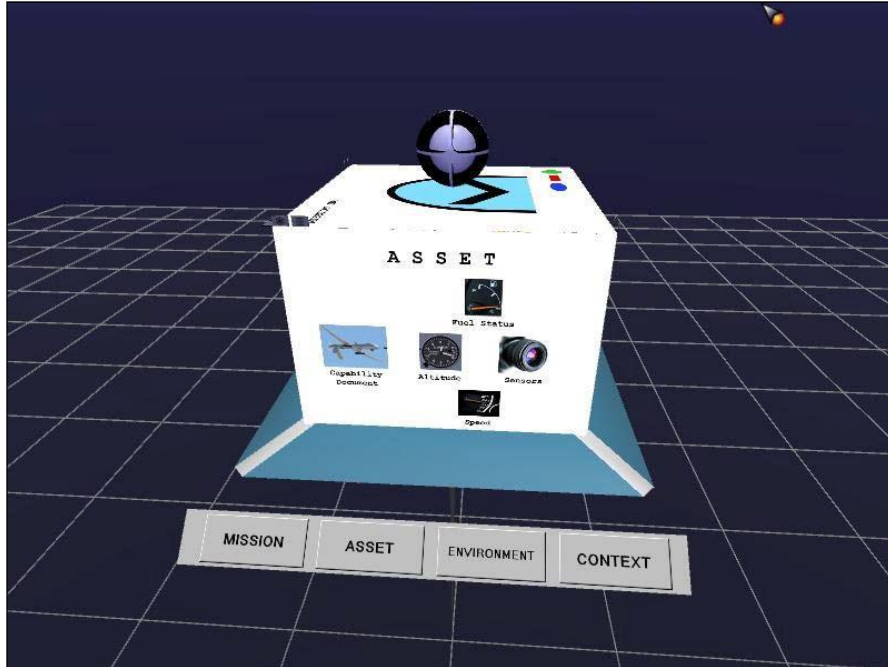




Figure 2. Original 3-D Knowledge Glyph Design

3.2.1.2 Three Dimensional Knowledge Glyph Implementation

The first step in process of integrating a 3-D Knowledge Glyph model into the UDOP framework was to replace the simple icons, as in Figure 19, used to represent satellites and ground sites in the 3-D World view with 3-D Knowledge Glyphs.

- Original Satellite icons 
- Original Ground Site icons 
- Replaced with interactive 3-D Knowledge Glyph cube, as in Figure 20.

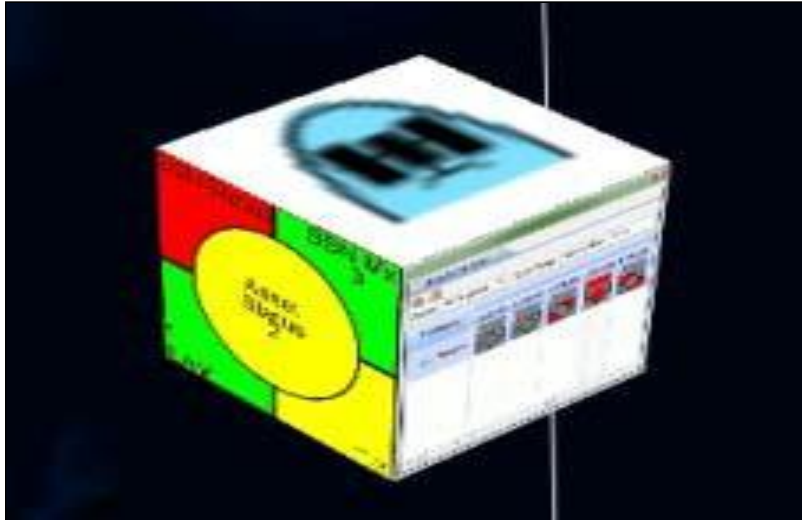


Figure 3. 3-D Knowledge Glyph Cube



Figure 4. 3-D Knowledge Glyphs in 3-D View

The cubes were created using OpenGL within the World Wind plug-in, with separate textures applied to each face. We also demonstrated, in an earlier prototype, the animation of the images on the cube faces. In addition, we experimented with the ability to play movie files on each face of a rotatable cube.

The cubes within the 3-D view are positioned at the proper time-based locations for the satellites or ground site, but the orientation of the cube is set relative to the camera view, such that the face, showing the Mil Spec unit icon, is always visible. Scaling of the cubes is also limited to keep them easily visible at farther distances.

We provided a new set of user interactions associated with the new 3-D Knowledge Glyphs.

- The user can use the right and left mouse buttons while the cursor is over a cube to rotate to the next face in that direction.
- The user can use the left or right mouse button to rotate the cube by one face, and drag left or right to rotate the cube freely around the vertical axis.
- Double clicking on a face opens supplemental information associated with the face image (see section 3.1.4.3).
- Double clicking on the top switches out of the 3-D view to a different context. (Currently this function is implemented on a face.)

A new button was added to the 3-D View toolbar to lock the camera to the currently selected satellite. This was needed to maintain focus on a single unit while the unit is moving in time.

3.2.1.3 Representation of 3-D Knowledge Glyph in 2-D Views

The concept of context switching requires maintaining a consistent focus on a unit of interest when switching between contexts. In the previous prototype design, the additional contextual views were to be displayed within the single 3-D environment. Our design approach was to integrate the 3-D Knowledge Glyph into existing 2-D views, maintaining a consistent interface between contexts. We tried three different technical approaches. Each approach manually positioned the 3-D Knowledge Glyphs over appropriate locations on the context view window. The assumption for all three approaches was that a Knowledge Glyph Layer would later be implemented to track and report the location of the selected unit on the screen.

The first approach was to add a new, optional, canvas layer to the base view object used by most existing UDOP views. This Knowledge Glyph Layer allowed the overlaying of a selectable, movable, graphic image over any existing view. Our prototype used trimmed screen captured images of the 3-D Knowledge Glyph from the 3-D view to simulate a 3-D object. The user was able to left and right click on the image to change faces, middle click to open the associated supplemental view, and click on the cube top to change contexts. One advantage to this approach was the ability to integrate the 3-D Knowledge Glyphs into many existing views. The drawbacks included the fact there was not a true 3-D engine associated with the displayed cube, so it would be difficult to provide smooth animation of the cube rotation. Also, it would be very difficult to dynamically change the images on the faces of the cube.

The second approach was to display a 3-D Knowledge Glyph within a small floating window, in front of the 2-D view, Figure 21. This allowed a fully interactive 3-D cube, using an OpenGL canvas, with identical functionality to that in the main 3-D view. An extension of this second approach was to create a new 3-D window type, with no borders or title bar.



Figure 5. Floating Window 3-D Knowledge Glyph

3.2.1.4 Context Functionality

The concept of a context was implemented by designating a few specific full-screen views showing some basic relationship between the selected unit of interest and other units or entities. Within a context, the user would use the 3-D Knowledge Glyph interface to bring up additional information about the unit. This supplemental information was displayed in floating windows over the current window, but not obscuring the 3-D Knowledge Glyph, Figure 23. The prototype used four hardcoded window locations.

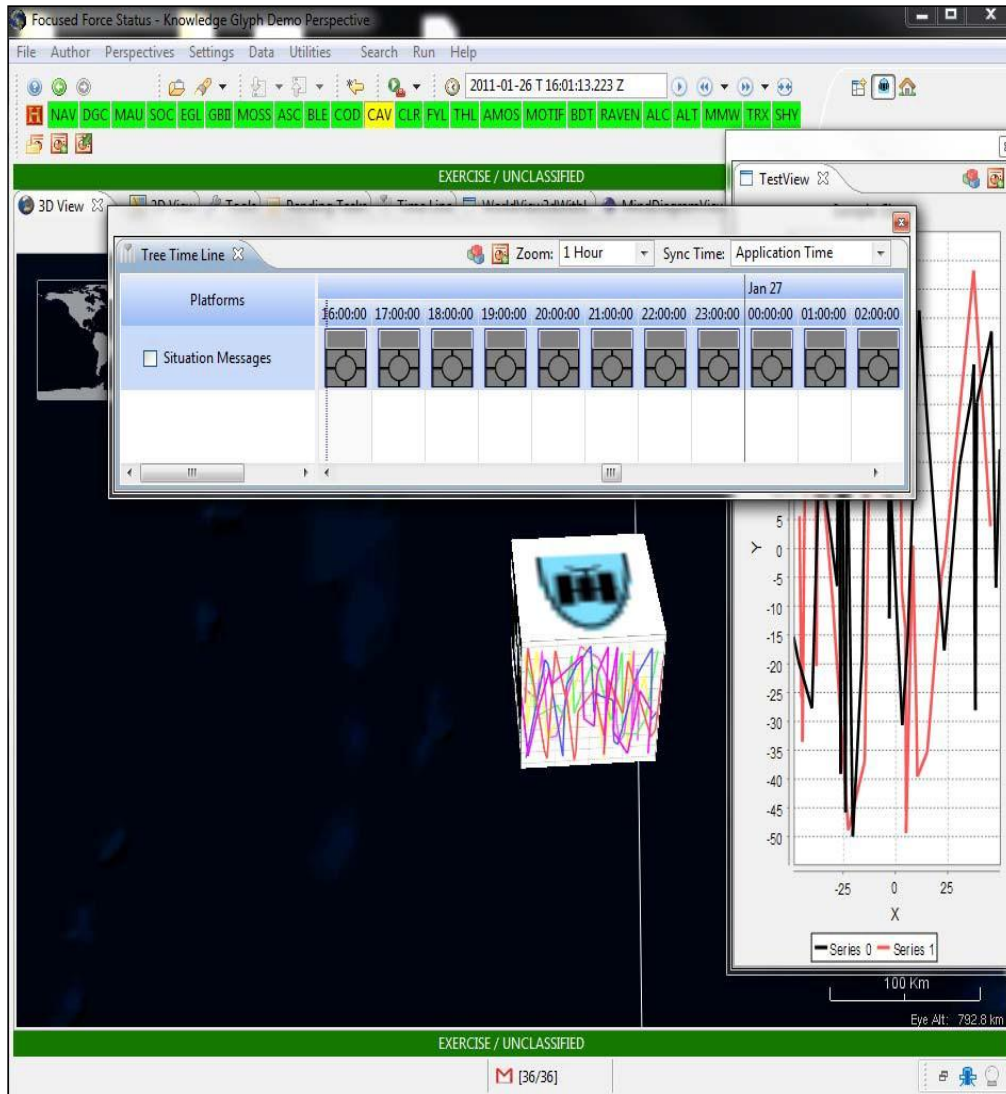


Figure 6. Supplemental Information Views

3.2.1.5 New Context Views

The first context view was the 3-D View that displayed satellites and ground sites as 3-D Knowledge Glyphs in a spatial context. The team built prototypes of two new Views to act as alternate Contexts—Timeline and Network. The use case concept was that the user will always have access to the 3-D Knowledge Glyph interface and can use that interface to switch between the three background view contexts.

The original Timeline context used the Glyph-based timeline view, Figure 23, but this turned out to be confusing and lacked some basic timeline controls. The new Timeline context uses a traditional Gantt chart display, Figure 24 following. The base Gantt chart widget was modified to place a 2-D image of a 3-D Knowledge Glyph at the current time location on the chart. The example shows multiple Satellites, with their glyphs located at the current time, and tasks show when they are scheduled to be in coverage of various ground sensors.

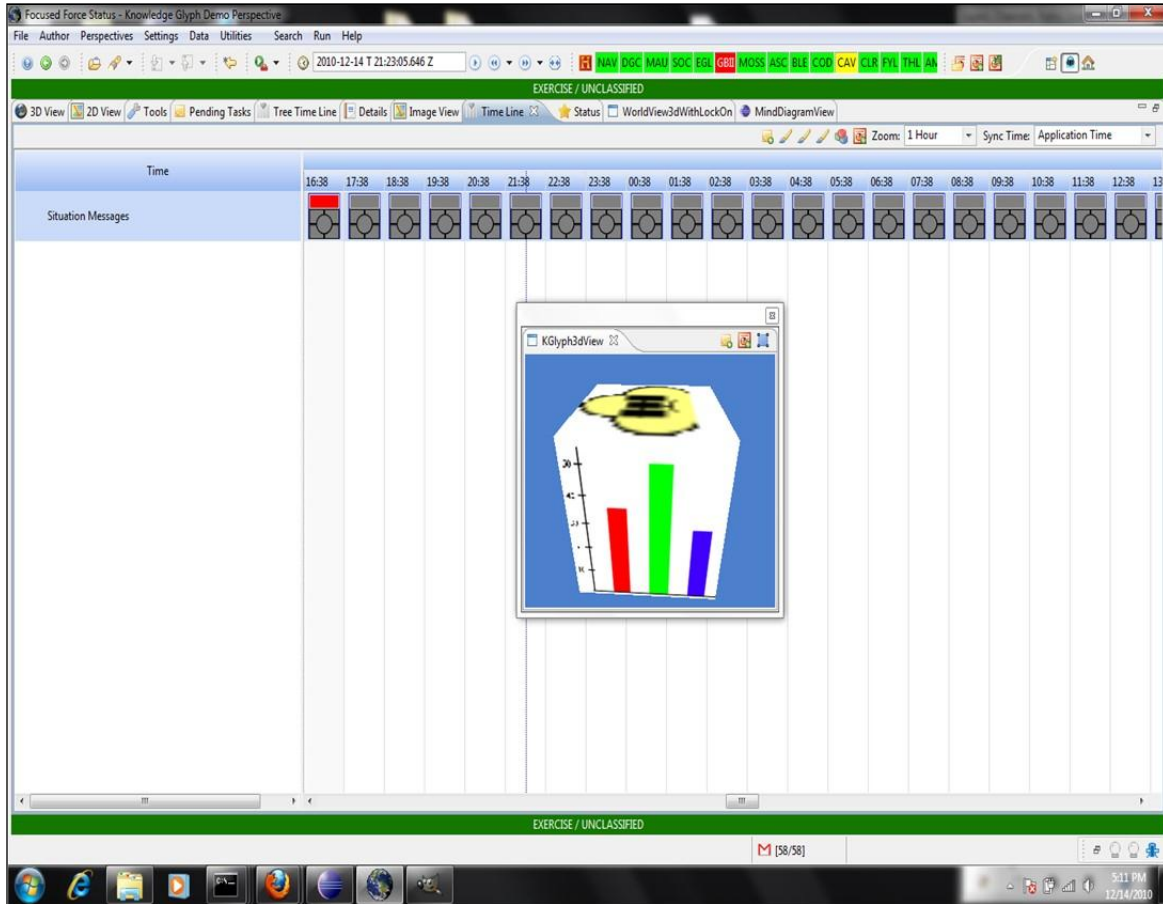


Figure 7. Preliminary Timeline Context

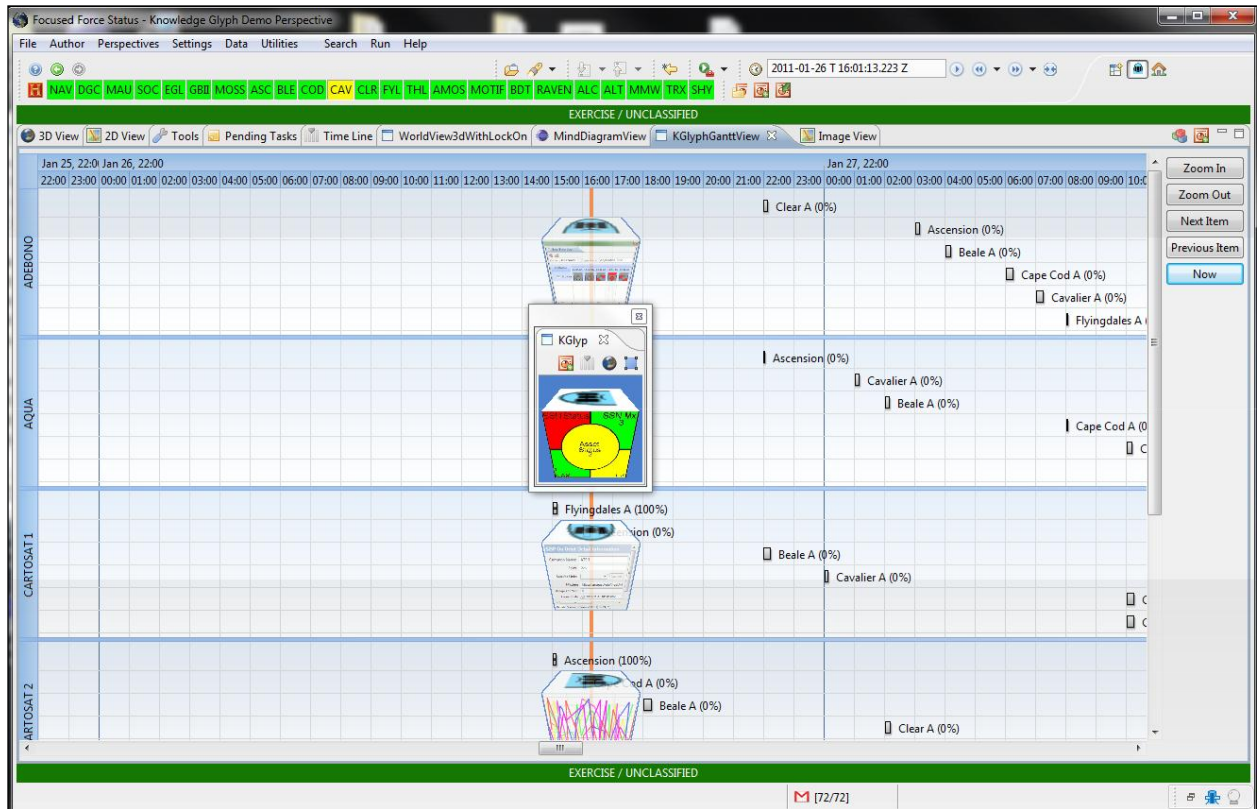


Figure 8. New Timeline Context

The new Network Context shows relationships between the selected unit and other units of interest with glyphs for each. The Network Context view uses the XMind tool to present several hardcoded graphs to illustrate some possible network diagrams that could be presented in a Network Context. Figure 26 shows the selected Satellite is visible to three ground sensors and has a communication link to one satellite and line of sight to another. Tabs at the bottom display alternate configurations. The first tab displays a tree graph representing the unit of interest at the top, with the subsystems as sub-nodes. This is similar to existing view used in the space UDOP. The third tab shows a similar subsystem tree, but arranged in a fishbone diagram to indicate a cause and effect relationship.

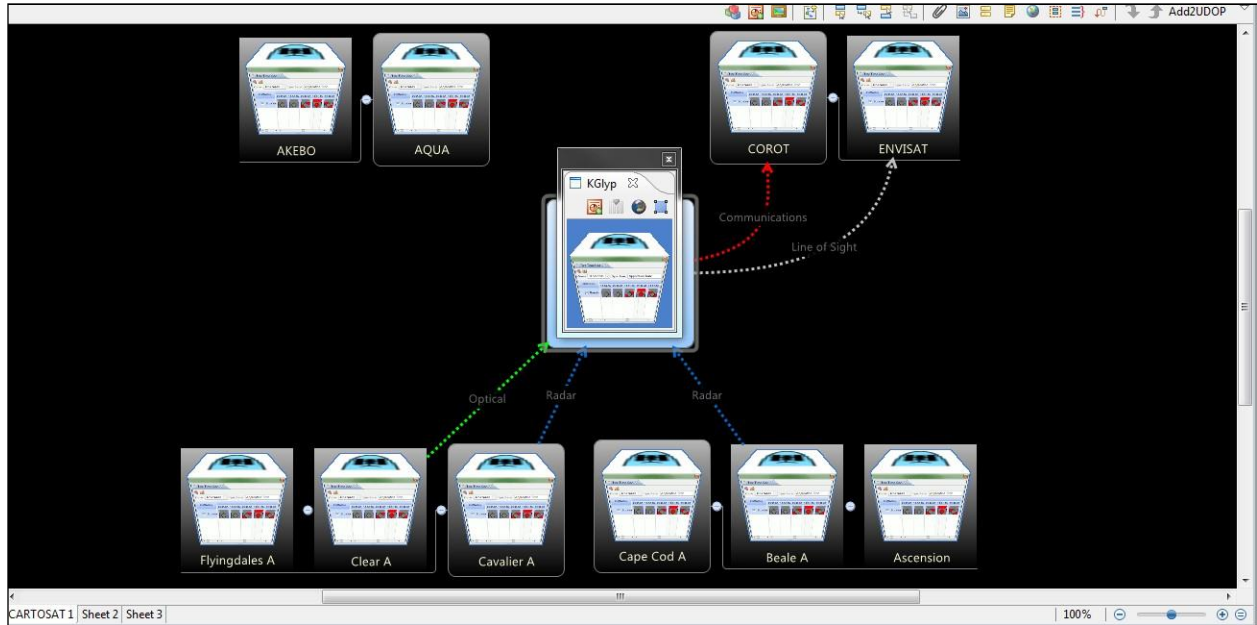


Figure 9. Network Context

3-D Knowledge Glyphs transitioned to a program called Context Switching.

3.2 The Context Switching Experiment

The displays were presented on a desktop computer monitor and the participants used a mouse and keyboard to input responses. Data was recorded using a software program designed for the study. The experiment took less than two hours for each participant. The objective was to have participants plan a daily schedule given a list of tasks and the times required for performing each. In order to complete the schedule, participants were required to take into account not only the task times but also associated travel times and bus schedules. They were given both a geographic (map) view and a temporal (timeline) view for reference. The method for switching from one of these contexts to the other and back was varied between subjects.

Three levels of the context switching methods were studied: switch context using tab; both contexts open in windows view; and full focal context switching in which the element of concern remains in the same location on the screen when the alternate context was selected. Upon arrival, participants were randomly assigned a switching method, until each method had been assigned 34 participants. Data collection included the number of tasks scheduled, the number of errors made during scheduling, and a log of the timing of task assignments. Figure 1 shows the overall high level flow for the experiment.

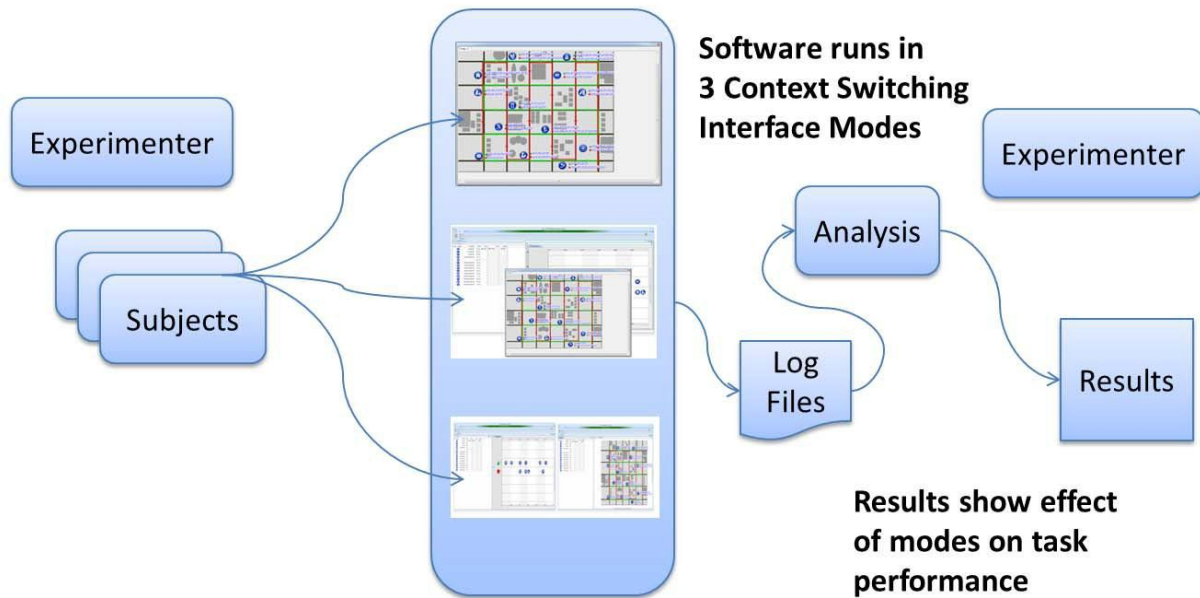


Figure 1. Overall Flow for Context Switching Experiment

3.3 Software Requirements for the Context Switching Experiments (CONTEX)

The CONTEX effort required a tool for conducting experiments on user interface concepts, specifically related to the idea of context switching. It provides a user interface for testing subjects on a defined set of test scenarios, and included the required usage data collection features. The design facilitates development and modification of user test scenarios, as well as future expansion of the types of interfaces to be tested.

3.3.1 Terminology:

Scenario – A hypothetical situation that is presented to the test subject. A typical scenario will involve a set of tasks, times, and locations that a subject will create a plan to accomplish.

Multi-Phase Scenario – A scenario may be designed such that after the user creates their initial plan, some details of the scenario change, requiring the user to modify the initial plan.

Session – A test subject may be asked to complete multiple scenarios sequentially. This full set of scenarios, completed as a set is considered a session.

Modes – the application will have two primary operation modes: Administrator and User, with different interfaces and capabilities available in each. While in User Mode, the application will operate in one of three experiment context switching modes, described in 3.3.3.

3.3.2 Development Synopsis

Views. Views are windows displays presented to the user for the purpose of completing an assigned set of tasks. The views may behave differently in different modes (see section 3.3.3 for description of experiment modes).

- *To Do List View*

Done	Where	Available	Allow	Bus #	From	Departs
<input checked="" type="checkbox"/>	home	at 8:00 AM	15 min	Bus 1	home	7:50 AM
<input type="checkbox"/>		at 12:00 PM	90 min			
<input type="checkbox"/>		at 2:00 PM	60 min			
<input type="checkbox"/>		4:00 PM-6:00 PM	15 min			
<input type="checkbox"/>			60 min			
<input type="checkbox"/>		10:00 AM-2:00 PM	40 min			
<input type="checkbox"/>		9:00 AM-5:00 PM	10 min			
<input type="checkbox"/>		9:00 AM-5:00 PM	15 min			
<input type="checkbox"/>		9:00 AM-6:00 PM	10 min			
<input type="checkbox"/>		9:00 AM-9:00 PM	30 min			
<input type="checkbox"/>		10:00 AM-9:00 PM	5 min			
<input type="checkbox"/>		10:00 AM-9:00 PM	60 min			
<input type="checkbox"/>		7:00 AM-10:00 PM	60 min			
<input type="checkbox"/>		9:00 AM-11:00 PM	5 min			

Figure 2. To Do List View

- A set of time-sorted goals for the user to try to achieve, consisting of a list of activities, locations, and time constraints. This list may change during the experiment user session.
 - *To Do List data elements*
 - *Scenario name*
 - *Update number – Original list will be version 1, with updates incremented.*
 - *Update time – time that the updated list comes into effect.*
 - *Row data*
 - *Due Date/Time – time that the action needs to be accomplished by.*
 - *Location – Name of the location where the action needs to be done at. (from a standard list of locations?)*
 - *Description – Text describing the action to be accomplished at the indicated time/location. Maybe an estimate of time needed to complete.*
 - In user mode, the user will be able to scroll through the list and select and copy cell contents. This could be used to paste data into the day planner list.
 - Experiment modes – this view will behave similarly in all experiment modes.

- *Map View*



Figure 3. Map View Example

- A map of the area of interest including all of the key locations listed in the Errands List and enough detail that the user can identify nearby cross streets that can be used to locate bus routes and stops on the route map.
 - At a minimum, the map may consist of a static image but should be scrollable.
 - In non-context switching modes the map may include Placemarks for locations listed on the To Do List destinations
 - In non-context switching modes it may provide pop-up menus to link to other views:
 - Display this location on Routes Map, which would put a place mark on the Routes Map.

- *Routes View*

The Routes view is integrated with the map view to locate the bus stops and bus routes that can be used to reach the destinations indicated on the To Do list.

- The map may be a static screen shot image, but should be scrollable and zoomable so that

the user can see the needed areas and level of detail needed.

- In non-context switched modes the user will be able to select a route and have it display the Schedule for that route (either as a pop-up or in the bus schedule view)

- *Bus Schedule View*

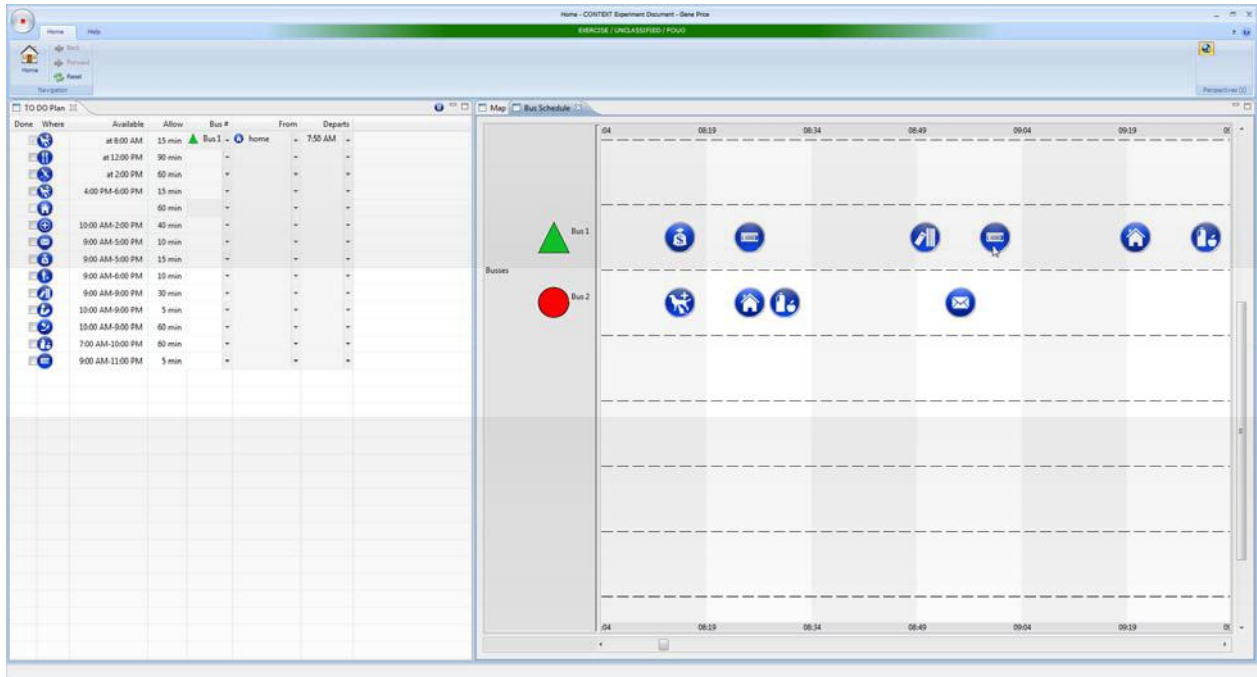


Figure 4. Bus Schedule View

- Displays the bus arrival times for each of the bus stops along a particular bus route. The displayed Route will be selected from a pull down list of available routes, or in non-context switched modes, by selecting a route on the Route List map view.
 - Applicable Route Schedule data will be pre-loaded by the administrator to match the Route Map.
 - List should be scrollable to see all entries.
 - In non-context switched modes the user should be able to select the bus stop name and have it display that stop on the Route map
 - In non-context switched modes selecting the route name will highlight the route on the Route map.
 - Cell should be selectable with various levels of copy possible based on context switching modes:
 - Copy time contents only
 - Copy time and stop and current route
 - Pop-up – Add this stop to Day Planner schedule

- *Experiment Administrator View*

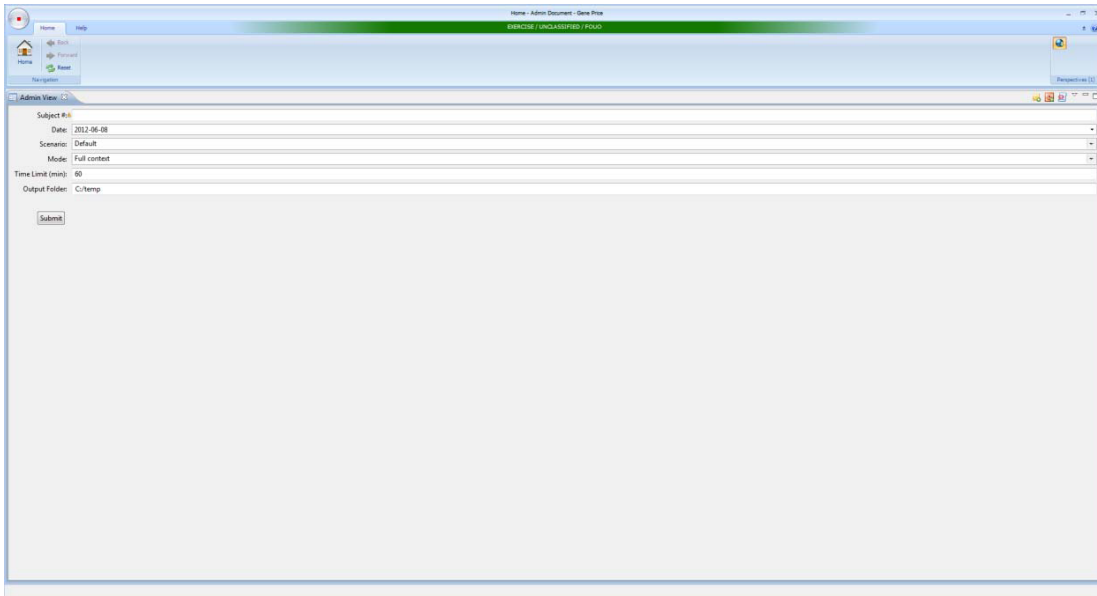


Figure 5. Administrator View

- This view provides functionality for the Experimenter to create the scenarios and data structures needed to perform the experiment.

View features:

The administrator user will be able to:

- Define and enter or import Data Sets of *map, routes, schedules, and key locations*.
 - Define Scenarios, either Single or Multi phase, entering a time, action, and location for each of multiple goals. (Duplicate feature would allow modifications of existing scenarios)
 - Define correct scenario solutions in the form of Day Planner lists that can be used to verify user entered solutions. There may be multiple solutions per scenario.
 - Define Standard User Sessions consisting of an ordered list of predefined Scenarios that a user can be assigned.
 - Define User Instructions to be presented to each user at the start of a User Session. This may vary based on Mode, Data Set, and/or individual Scenario.
 - *Initiate Test User Session. Allows the administrator to select a predefined Standard User Session, enter related User ID, (may be auto-generated) and select Run.*
- *After the administrator selects “Run” the application displays a Start User Session View (see next section) , and is ready for subject user to begin the experiment session.*

Data Sets: *(Regional set of data used by subjects to complete scenario tasks) Map
Bus Routes Route Schedules
Key Locations*

Scenario: (To Do List) Example data:

- Scenario Name: *Shopping at Mall - Multi Phase*
- Data Set: *Dayton*
- To Do List:
 - Phase1 0900
 - 0900 Start at TecEdge
 - 1100 Meeting at City Hall
 - 1300 Lunch at Paneras The Greene
 - 1500 Buy present at Home Depot – Presidential
 - 1700 End TecEdge
 - Phase2 1300
 - 1600 Emergency at Home
 - 1700 End TecEdge

Standard User Session Sets:

Session Set 1.

Seq.	Scenario Name	Mode
1	<i>Shopping at Mall– Single Phase</i>	<i>Full-Context Switched</i>
2	<i>Meetings Downtown– Single Phase</i>	<i>Partial-Context Switched</i>
3	<i>Class and Job– Multi Phase</i>	<i>Non-Context Switched</i>
4	<i>Shopping Downtown– Multi Phase</i>	<i>Full-Context Switched</i>
5	<i>Visit Friends – Multi Phase</i>	<i>Partial-Context Switched</i>

Session Set 2.

Seq.	Scenario Name	Mode
1	<i>Visit Friends – Multi Phase</i>	<i>Partial-Context Switched</i>
2	<i>Meetings Downtown– Single Phase</i>	<i>Partial-Context Switched</i>
3	<i>Class and Job– Single Phase</i>	<i>Non-Context Switched</i>
4	<i>Class and Job– Multi Phase</i>	<i>Non-Context Switched</i>
5	<i>Shopping Downtown– Multi Phase</i>	<i>Full-Context Switched</i>

- **Start User Session View** – Selecting the Start button will start the session timing and other logged metrics, then display the To List View in the appropriate user interface Mode (Partial, Non or Full Context Switching).

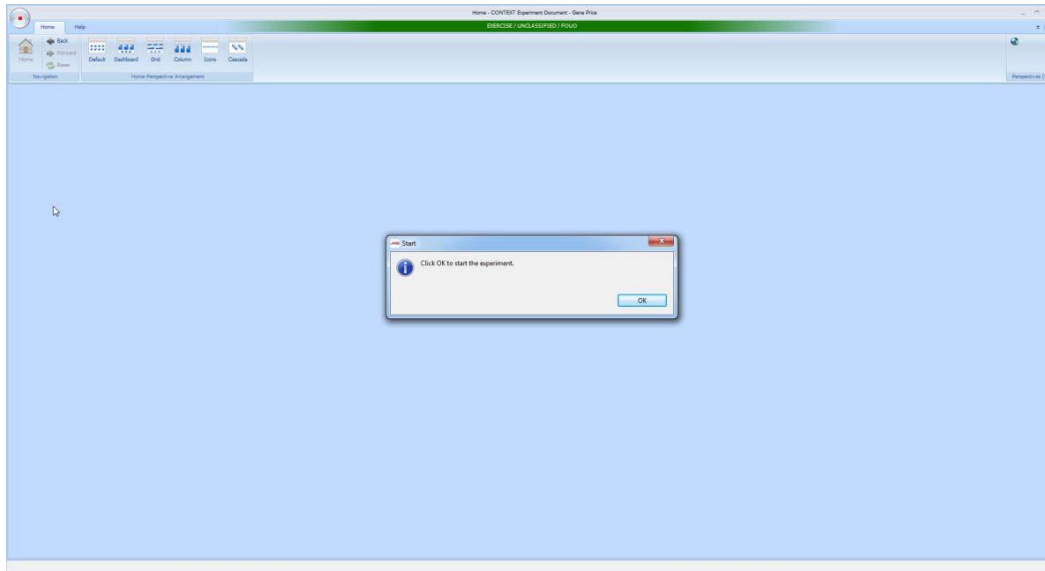


Figure 6. User Start Screen

3.3.3 Experiment Context Switching Modes – At the beginning of a user session, the administrator will select the Experiment Mode, along with the required user information, select the specific scenario to be run. One of the main goals of this software is to test the effects of different context switching mode on user performance. The three test modes provide different levels of information and interactivity to user who is attempting to complete the given task requiring collection and correlation of data from several different sources. The three modes will vary the interface such that they will require more or less screen switching, which will cause the user to switch contexts more or less. Differences in modal behavior for the different view are described in the view sections above.

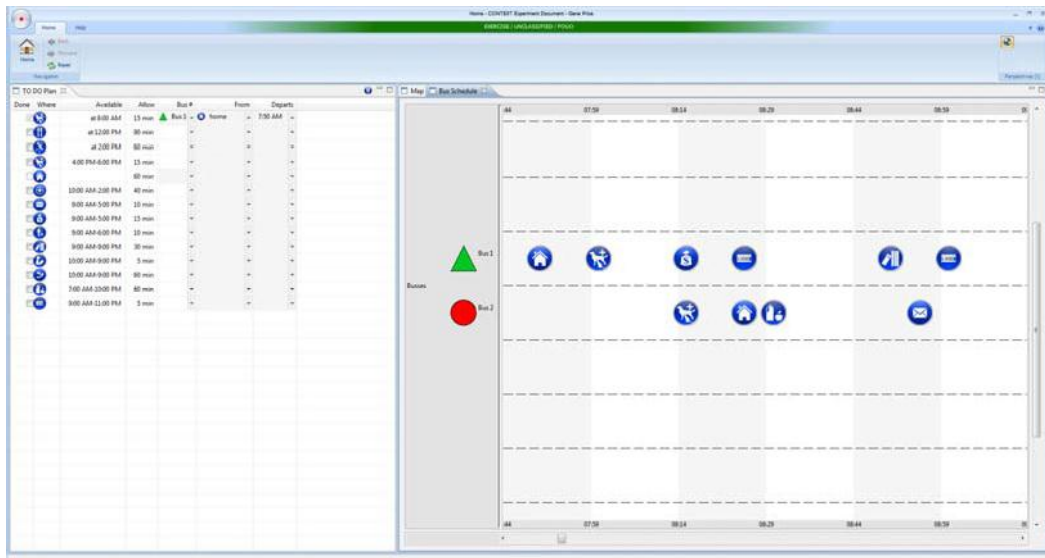


Figure 7. Full-Context Switched Mode Example – To Do List & Schedule

- Full Context Switch Mode – (Figure 7, and Figure 8)

- This mode will require the user to interact with only one view at a time. Each view will be presented in a single full screen window. A set of tabs (or buttons) will be used to switch between views when clicked on an icon on bus schedule or time on timeline.
- Views in this mode will generally not have features to dynamically link to other views.

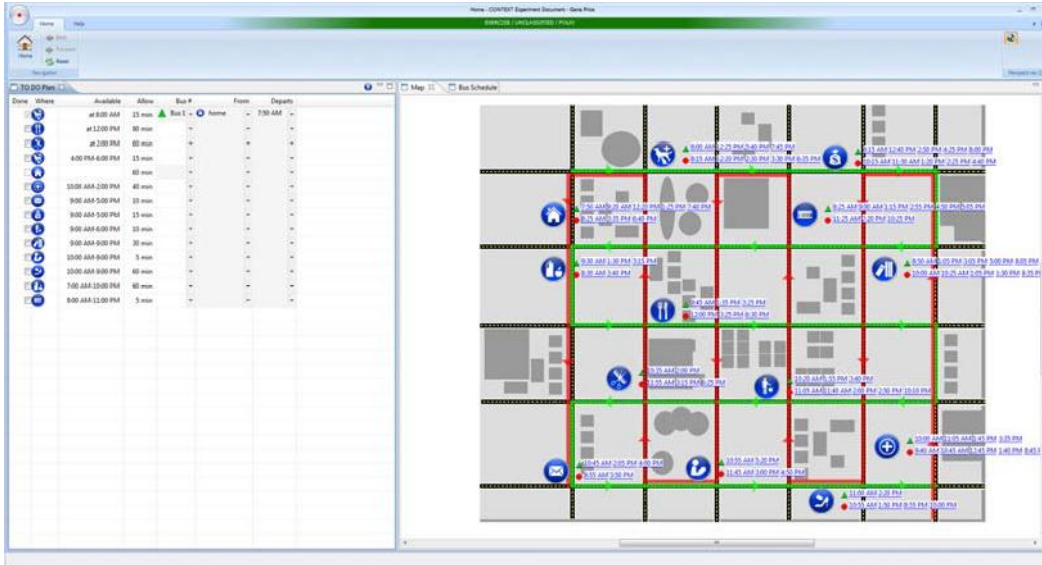


Figure 8. Full – Context Switched Mode Example –To Do List & Map

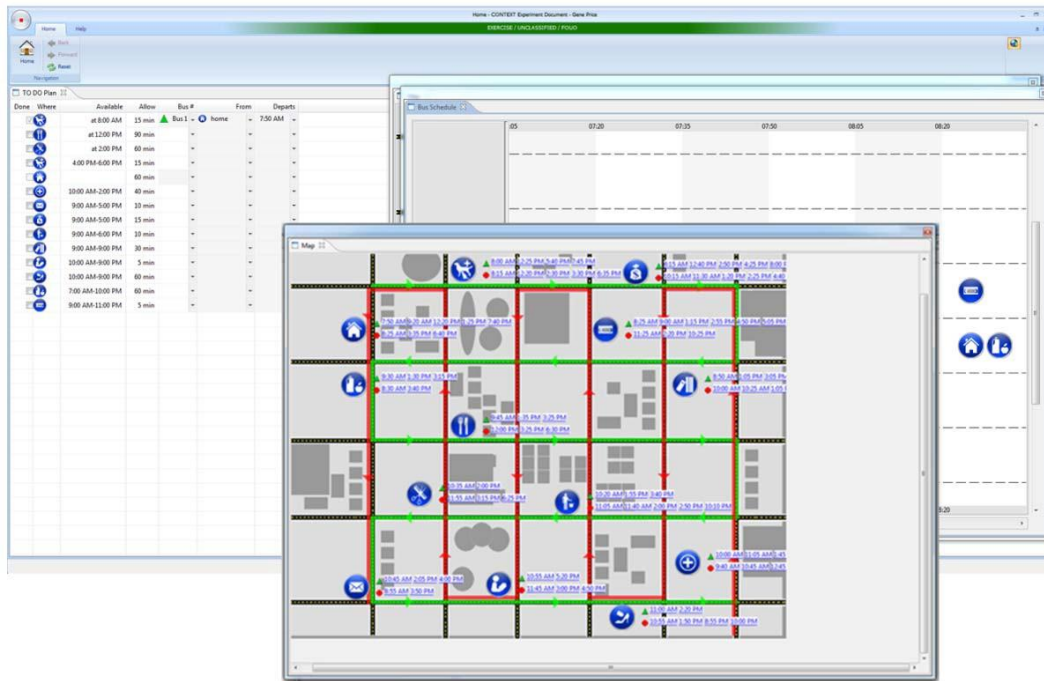


Figure 9. Partial context Switched Mode Example - Routes with Map and Bus Schedule floating windows

- Partial Context Switching Mode – (Figure 9) This mode will reduce the amount of view changes that the user will need to perform by allow some frequently used views to be displayed in floating, resizable windows. The Map and Bus Schedule will still be available as full screen windows accessible by Tabs along the top.

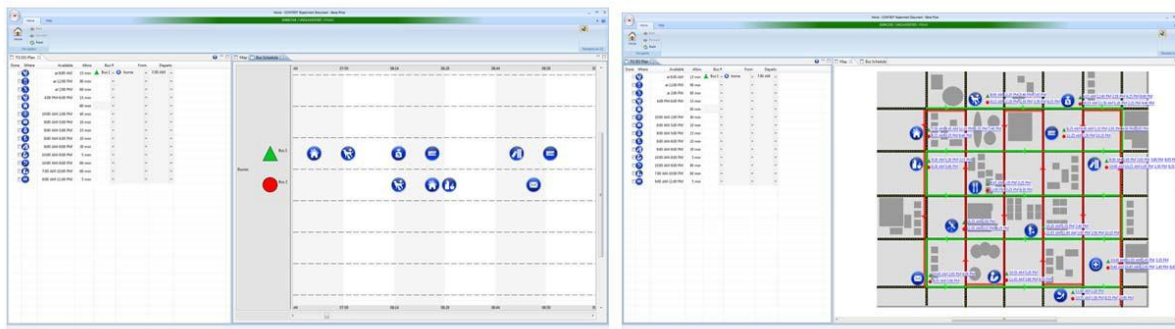


Figure 10. Non context Switched Mode Example

- o Non Context Switching Mode – (Figure 10) This mode is designed to maximize the amount of related information available to the user, to prevent context switching. Window content will be linked where possible so that selecting an item in one window will display related information in the other visible windows. For example, selecting a row on the To Do List view will display place mark on the Map for the indicated location. Selecting a Bus Route line on the Map will display the schedule for that route in the Bus Schedule View.

3.3.4 Data Logging and Analysis features

- o The analysis features will be driven by the needs of the experimenter but will probably, at a minimum include the following minimal set of data to be collected and exported from the application:
 - User entered data in the current Day Planner Schedule rows.
 - Header data identifying the user id, session id, experiment interface mode.
 - Basic timing metrics such as when the experiment session started and ended.
 - May include Accuracy metrics, comparing the user entered schedule to the “Correct” schedules entered by the administrator.
- o Basic data can be saved in a simple comma separated values text file or XML file.

3.3.5 Use Cases

These use cases center around the subject/user experience during a test session.

3.3.5.1 Full-Context Switched – Single Phase

A sample user interaction scenario. The goal for the user is to create a schedule of hypothetical actions needed to meet a set of transportation related goals presented to him as part of the exercise. The user will utilize multiple views to collect information to determine which bus routes would be needed to reach the

desired locations, as well as times to catch buses at particular bus stop. This case covers a static To Do List, but we may multi-part scenario, with the user being required to update the planned schedule at some point due to changes in the goal list, or “unforeseen” factors.

Scope: Standard user interaction scenario running in Full-context switched mode.

Primary Actor: *user/subject*

Supporting Actor(s): *Administrator / Test Conductor*

Preconditions:

2. *The CONTEX tool is assumed to be configured by the administrator with a standard set of test scenario data. This data will include: Map, Bus Routes, Route Schedules, To Do Lists, Correct answer Day Planner plans for each To Do List.*
3. *The Administrator selects the To Do List for this session and enters the user id, session id, and test modes.*
4. *Example data:*
 - *User Id: achuck*
 - *Session ID: achuckFullContext01*
 - *Context Mode: Full Context Switching*

 - *To Do List:*
 - *0900 Start at TecEdge*
 - *1100 Meeting at City Hall*
 - *1300 Lunch at Paneras The Greene*
 - *1500 Buy present at Home Depot – Presidential*
 - *1700 End TecEdge*

Triggers:

Experimenter selects Experiment start button.

Main Scenario

- User selects the To Do List tab, looks at list of locations he needs to visit. Current assumed virtual location and time is the first item in To Do List.
- User selects the Maps tab to locate the destinations listed.
- User selects the Route Map tab to see which bus routes go to the desired destinations, and stops on those routes closest to start and destinations.
- User selects the Day Planner tab and enters bus route, bus stop and time information.
- User repeats parts of the process until he has completed the Day Planner list of items that he thinks will meet the To Do List goals.
- When complete, user selects “Done” button,
 - This is a single phase session; session log information is saved to disk.

3.3.5.2 Full-Context Switched – Multi-Part Re-Plan

A sample user interaction scenario. The goals and general flow of the sessions will be the same a Use Case 1, but when the user completes the initial planning phase and selects the “Done” button, they will be presented with a modified version of the To Do List, representing a change to the initial list at some point in time. The user will be required to update the Day Planner list to meet the new objectives.

Scope: Standard user interaction scenario running in Multi-Part mode.

Primary Actor: *user/subject*

Supporting Actor(s): *Administrator / Test Conductor*

Preconditions:

User has completed the first phase of a multi-phase session.

Triggers:**Main Scenario**

- See use cases for Single Phase.
- When complete, user selects “Done” button,
 - This is a multi-part session; session log information is saved to disk.
- The application loads new data and displays the To Do to present a modified To Do list with a new start time.
- The Day Planner List is updated so that rows before the new current time are not modifiable.
- User repeats process for a normal Single Phase use case.

3.3.5.3 Partial-Context Switched – Single Phase:

- A sample user interaction scenario.

The goals and general flow of the sessions will be the same a Use Case 1, but the To Do List view and the Day Planner view are both visible at all times in floating (docked?) windows. This should change the number and sequence of user mouse clicks, and perhaps affect accuracy and speed for accomplishing the tasks.

Scope: Standard user interaction scenario running in partial-context switched mode.

Primary Actor: *user/subject*

Supporting Actor(s): *Administrator / Test Conductor*

Preconditions:

Same as Use Case 1, except administrator selects Partial-Context Switched Mode.

Triggers:**Main Scenario**

- User clicks in the To Do List window area,

Full-Context Switched – Single Phase:

- A sample user interaction scenario.

The goals and general flow of the sessions will be the same a Use Case 2, but the user interface will display all views in docked windows and will also provide logical (context sensitive) function relations between them. For example, selecting on a Route on the Routes Map will update the window displaying the Route Schedule to display the schedule for the selected route.

This should change the number and sequence of user mouse clicks, and perhaps affect accuracy and speed for accomplishing the tasks.

Scope: Standard user interaction scenario running in full-context switched mode.

Primary Actor: *user/subject*

Supporting Actor(s): *Administrator / Test Conductor*

Preconditions:

Same as Use Case 1, except administrator selects Full-Context Switched Mode.

Triggers:

Main Scenario

- Assumed virtual location and time is the first item in To Do List.
- User selects the location of the first row. (Starting location)
 - This causes the application to update the Map View to display a marker at that location, and on the Routes View.
- User can look at the marker in the Routes View and find the Routes and stops near the location. User records the stop in the day planner view as the first location.
- User repeats the processes for the second row in the To Do List, which will be the first destination.
- When the bus stop for the first destination is identified, the user will examine the Routes View to see if a single bus route connects the starting stop and the destination stop.
 - If so, the user selects that Route on the Routes Map
- The application updates the Route Schedule view to display the bus schedule for that route.
- User finds a row (bus) that arrives at the destination stop before the required time for the first destination. (including walk time from stop to destination)
- User looks for when that bus will be at the starting bus stop.
- User records that bus stop name and time in the Day Planner (copy paste? Click menu?)
 - And the destination bus stop name and time.
-
- User repeats the process until he has completed the Day Planner list of items that he thinks will meet the To Do List goals.
- When complete, user selects “Done” button,
 - This is a single phase session; session log information is saved to disk.
 - Sequence is the same, except the To Do List view and the Day Planner view are both visible at all times so user does not have to select a tab to view them.

3.3.3.5 Experiment Administrator:

- A sample administrator interaction includes the sequence of events for setting up a set of experiment sessions with a subject. A full user session may be designed to consist of a sequence of individual user sessions each using different modes and To Do Lists. Final experiment results will be calculated using data from multiple users over multiple full sessions. The administrator will design the number and sequence of these sessions before testing begins.

Scope: Standard administrator interaction to set up the application for use by a subject.

Primary Actor: *user/subject*

Supporting Actor(s): *Administrator / Test Conductor*

Preconditions:

Administrator has prepared To Do Lists, and other supporting data for the full experiment.

*Administrator has prepared the planned set of standardized user sessions, for example each user can be assigned an experiment session set consisting of a sequence of individual scenarios:
Subject user is present and ready.*

Triggers:

Main Scenario

- *Initiate Test User Session.*
 - *Administrator selects from a predefined list of Standard User Session Sets (see view description),*
 - *Enter related User ID, (may be auto-generated)*
 - *Selects Run.*
 - *The application displays a Start User Session View (see next section), and is ready for subject user to begin the experiment session.*
 - *User completes all scenarios the selected User Session Set,*
 - *Session Set Completed view - administrator reviews status of completed user session.*
 - *Administrator verifies that log file was recorded to the correct location and file name.*

4.0 MODELING SENSEMAKING AS ABDUCTION-BASED INQUIRY FOR CYBER EFFECTS

4.1 Introduction to Abduction and Sensemaking

This task was an investigation into abductive inference and sensemaking in the context of hardware, software, and malware reverse engineering for cybersecurity. It was conceived as a supporting task for the RDM STT objective to develop integrated human-machine reasoning and decision processes to fight through cyber attacks. The Generic Modeling Environment and DEVS Language based system under development in RH was to form the base of a demo system for reverse engineering. The demo was to be developed by personnel in RH that brought in knowledge of reverse engineer's cognitive models and the newly developing cognitive modeling platform. This subtask was to provide guidance in abduction for the modeling task. The subtask goal was initially stated as a "high-level abstraction of sensemaking and abductive inference process inspired by Whitehead and Peirce".

Abduction owes its origin to the work of Charles Sanders Peirce, who saw it as essential human reasoning and as an explanation of intuition. It is a third form of reasoning, differentiated from deduction and induction. There has been considerable work in recent years to apply abduction in artificial intelligence and as an investigation into complex problem solving. The results and insights into abduction form a principal outcome of this project.

Sensemaking is a familiar term to those in defense work. In this effort, the area of interest is what might be termed "hard problems" of sensemaking. That is the intuitive grasp of a situation that is the province of experts. The intuition of an expert can appear almost mystical. It is a property that the experts themselves are often hard pressed to explain. The works of Klein [2] have provided a many interesting case studies of this aspect of sensemaking.

The project began with a study of abduction and sensemaking in the context of reverse engineering. The study produced results summarized in published two papers[3][4], also included as Appendices A and B. The first paper examined abduction and sensemaking with respect to cybersecurity reverse engineering and further proposed an interesting related subtask -- sensemaking of obfuscated code. The second paper proposed an architecture for an artificial intelligence (AI) system capable of abduction and situation recognition, two key aspects of sensemaking.

The results of the project are best described by treating the topics and demonstration system in separate sections. In section 4.2, the research discussed in Appendix A is summarized and section 4.3 summarizes the demonstration system. Section 4.4 discusses the second paper and the proposal for an abductive system.

4.2 Applying Abduction

Abduction has been used in AI systems at least back into the 1980's. A major body of work was done by the AI Lab at The Ohio State University. A summary of this work can be found in Josephson and Josephson [1]. In characterizing abduction, Hoffmann [5] provides a taxonomy of abductive types. This was used as a guide to problem solving types subject to abductive methods. A first thing observed from the view of Hoffmann's taxonomy was that earlier AI systems used only the simplest levels of abduction. This works by supplying missing assumptions. The assumptions are selected based on typical cases. For example when observing a wet sidewalk, humans typically assume rain was the cause. This is decidedly a typical human ability, albeit a rather simple one.

The taxonomy covers a very broad range of abductive capability. Hoffmann uses two dimensions to construct a 3 by 5 matrix of abductive types. One dimension is the source of the hypothesis. The hypothesis can be in our mind, one that exists in our culture or one historically new. This dimension is not used in our work, for the moment we consider only hypothesis which exist in the mind of the reverse

engineer. This is not to imply that the other cases are unimportant. The reverse engineer needs to use the cultural dimension in order to succeed; knowledge discovery in cybersecurity is an ongoing process. It is also the case that a reverse engineer may encounter the new and novel approach and so may add to the knowledge of the culture. But for the moment, we leave consideration of these as future work.

Table 1. Hoffmann's Taxonomy of Kinds of Abduction versus Breadth of Usefulness

	Exists in the mind	Exists in Culture	Historically new
Fact	Selective Fact	P-Selective Fact	H-Selective Fact
Type of Concept	Selective Type	P-Selective Type	H-Selective Type
Explanatory hypothesis of Law	Selective Law	P-Selective Law	H-Selective Law
Theoretical Model	Selective Model	P-Selective Model	H-Selective Model
Representation	Selective Meta-diagrammatic	P-Selective Meta-diagrammatic	H-Selective Meta-diagrammatic

The case where the hypothesis exists in the mind is called selective by Hoffmann, implying the human selects the needed item from memory. The other dimension describes the type of abduction. This ranges in abstractness of the abduction. The first two forms are clearly the simplest. The simplest form is to select a fact that explains or supports the inference. The next level up is selecting a type or concept to support the inference. These appear to be ubiquitously employed in almost all human activity. The third level is selective law abduction where a law or stereotype is used. The abduction step is to select a law that supports the observations and satisfies as an explanation. This is the kind of abduction applied in many diagnostic AI systems (see Josephson). This, like the first two types, appears to be common in humans, especially with a domain expert.

The last two types are the forms that are most interesting from a reverse engineering perspective. Selective model abduction selects a model to support the abduction. As will be shown, reverse engineers continuously select hypotheses that are models. While selective law abduction is used to verify the assumptions in the model and the conclusions derived from it, it is the model that drives the reverse engineer's path of exploration.

Meta-diagrammatic abduction is where a shift in representation of the problem leads to the inference. The reverse engineer is working in a non-monotonic mode of reasoning. When surprise occurs, meta-diagrammatic abduction is one of the tools to clarify the situation and to adopt a new model. This is often done by invoking the tools to obtain a different view of the situation. This is clearly described by Magnani.[6]

It can be argued that humans use the lower (simpler) levels very often, but only apply the upper levels sparingly. The upper levels of abductive inference are used to develop new and innovative theories; this is argued at length in [6]. Magnani also makes the case for manipulative abduction, the use of tools manipulated to gain understanding. Magnani provides a clear view of a critical part of the reverse engineering process. Reverse engineering uses tools in manipulative abduction. Both Magnani and Peirce use mathematical reasoning via diagrams as an illustrative example of manipulative abduction. In

mathematics, diagrams often have been the pivotal source of insight into proof. At a more applied level, engineers have relied on diagrams to visualize and to record designs. Given that software artifacts are considered to be among the most complex objects developed by human kind, tools need to support visualization and recording. Software becomes too complex to comprehend without multiple levels of abstraction and tools capable of navigating and exposing software structure and behavior supporting the abstractions. While it is one thing to design a software system, it is harder problem to reverse engineer the same software. Too much of the designer's thinking is lost in the translation to executable code. Thus for a reverse engineer, the tools are used to expose the software and to guide theory formation about what its operation is doing and ultimately what the goal of the software system may be. It is this discovery process where manipulative abduction will be shown as a key solution method.

The third aspect explored in this phase was expert behavior in sensemaking, heavily inspired by Gary Klein's work [2]. In particular the issue of "not quite right" situations illustrates the intuitive behavior of experts. Experts, with a very high level of expertise, demonstrate the ability to sense when a situation is close, but not quite the expected situation. The experienced reverse engineer is able to make abductive leaps because they have a large body of knowledge to apply. Klein's work with critical decision making offers some useful insights into mechanism behind an expert's ability to abductively reason. An expert's experience enables recognizing situations encountered in the past. However, there is another side to this capability that is described in Klein's work, the recognition that a pattern is not right. Both of these capacities are important in reverse engineering. Both can trigger abduction since the human mind does not require a complete match, but can find partial matches that trigger sub goals to evaluate to the pattern. But the ability to see that a pattern that has been accepted as a hypothesis is not correct can be a strong trigger for advancing new theories or new ways to view the situation (selective model abduction and meta-diagrammatic abduction). This can be termed surprise in the sense that an unexpected change occurs in the problem solving activity. From both studies and some introspection, it is likely that a novice will continue down a path, while an expert will sense the falsification and adopt new approaches. This aids the malware writer in hiding the intent by making the code look benign by suppressing clues and helping the novice pursue the wrong course.

The work found in [7] shows that the reverse engineering task is full of opportunity to follow unproductive paths in exporting a code space. This ability to detect when a path of enquiry is not the correct one is critical to good and effective reverse engineering work.

4.2.1 The Reverse Engineering Problem Space

Reverse engineering is a broad activity and encompasses a number of subcategories defined by the intended goal. The tools are common across this space and the domain knowledge is based on common base set. Software systems are regarded as the most complex of human designed technologies. Software can be difficult to understand when the source code is provided, but a cybersecurity reverse engineer is restricted to machine code and often intentionally obscured machine code. Machine code, or its assembly language interpretation, is generally very large and has a complicated data flow and control flow structure. Complete coverage of the code is intractably large even with automated methods.

Our interest in reverse engineering is from a cyber-defense perspective. In this case, the reverse engineer is trying to find malicious code fragments within program. Using reverse engineering for other domains can have a more relaxed deadline, cyber-defense adds a time driven threat to the problem. It also drives the practitioner to triage the critical elements of the code, not to apply a broad attempt to study all parts of the code. The ability to successfully triage is a necessary skill for a reverse engineer. In this problem space, the critical elements may have been obscured and hidden by a variety of means. The library code, considered irrelevant in some reversing efforts, can become a target for the reverse engineer; because the suspect code may be using existing flaws in other code or systems to achieve malicious purpose.

This is a difficult task and requires skilled individuals. The complexity of a given piece of software can be high and requires the grasp of detailed information in order to understand the operation. The tools utilized by the reverse engineer are used in what Magnani refers to as manipulative abduction. The tool provides a visual reminder of what has been discovered, as a source for the discoverer, and when a theory is falsified, the means to abductively shift the view to explore a new model or representation (Hoffmann's model abduction and meta-diagrammatic abduction).

This project choose to explore the application of abductive reasoning by looking at a subtask for reverse engineering.

4.2.2 Base Knowledge

One of the difficult issues of this project was the base knowledge applied by a reverse engineer. This turns out to be very large. The base knowledge needs to include, at a minimum, basic machine level instruction set, operating system library routines, detailed machine architecture, and general algorithms. This is clearly a very large set. In addition, for expert level ability in malicious software detection, the knowledge base needs to include knowledge of common approaches, exploits, and attacker intentions.

4.2.3 Obfuscation Subtask

The subtask that offered an interesting domain for abduction was obfuscation. Obfuscation techniques are used for protection of proprietary software as well as means of hiding code. There are a range of techniques applied to obfuscation. For one there are tools which will automatically obfuscate code. The clever programmer can also restructure code to make it less obvious and to provide defenses against the disassembler and debugger.

4.2.4 Abduction in Reverse Engineering

The work to apply abduction as a way to understand and to characterize the tasks in reverse engineering leads to a focus on the selective model and meta-diagrammatic abduction in a manipulative abductive mode. This is not to imply that the other levels are not used, but rather this looks like a dividing point between highly skilled vs. novice practitioners. The knowledge used to guide the abduction is split into two types. The direct situation recognition is the knowledge type applied in all knowledge based AI systems and cognitive modeling systems. This is the "I know what to do here" kind of knowledge. It is easier to deal with, because it is triggered by direct evidence and tends to have a closed form.

The other kind of situation recognition is the metacognitive sense-making, "Something isn't right" variety. This is harder because it tends to have indirect forms. Klein et. al. show a number of examples of expert behavior that show the power of this kind of perception [7].

It can be that a critical fact is missing from the situational description. This is a very interesting problem, given that abduction tries to fill in the missing pieces. At some point, the expert's processes override the assumptions about missing pieces. An observation may also just not fit the current hypothesis based on the deductive path predicting that observation.

Abduction leads naturally to two important issues. First of all surprise: surprise is a trigger for abduction. When we struggle to deal with an unfamiliar situation (surprise) abduction is one of the powerful tools to reach conclusions quickly and efficiently. Likewise, abduction can lead to further surprise; it is not guaranteed to reach valid conclusions. This leads to the other remark about human reasoning under abduction, it is non-monotonic. Prior conclusions have to be abandoned as reasoning and problem solving progresses so premises evolve with understanding. This sometimes requires whole chains of reasoning to

be removed from consideration. Fortunately for a human, holding contradictory positions is not a limitation. Unfortunately, current AI systems have difficulty with non-monotonic reasoning.

The four categories of obfuscation are: layout obfuscation, data obfuscation, control obfuscation, and preventative obfuscation. While a complete study of these was not undertaken, some instructive examples are used to make the argument for the claim of abductive reasoning. *Layout obfuscation* removes some formatting and naming information. This is considered a low potency transformation [10]. For that reason, we dispense with any discussion of these transformations. In *data obfuscation*, one approach is to convert constants to procedures. Here the recognition is that the computation is not required and a simple constant is all that is needed. The reverse engineer will have to trace the computation to put it into a form where it is clear that a constant is the result. This often requires some knowledge of math to recognize.

Another similar complication is to split a variable. This requires the reverse engineer to detect that two or more values are always used together and then transform them into the real variable. Again this is a meta-diagrammatic transformation and it leads to a more compact representation. A model shift may occur because a detail of an algorithm may be unveiled by the transformation.

Data obfuscation can be achieved with aggregations. This can be to merge independent data or to split dependent data. For example, arrays can be split, folded or merged to hide the intent of the contained data.

These methods all make location of structures found in the higher level language representation more difficult to discern in the code.

Control obfuscation can seek to obscure control flow, how the code is aggregated, or how the code is ordered. An interesting approach here is to reorder code in ways where a high level language analog does not exist. This will require the reverse engineer to derive the intent by low level tracing methods. Another technique is inlining or outlining code. Inlining removes procedures (methods) and outlining creates extra procedures. Again the reverse engineer must group the methods and look for how they are used in order to infer intent. This can be more model abduction prior to a meta-diagrammatic shift.

A very useful technique for misleading the reverse engineer is to insert dead or irrelevant code. This is achieved by opaque predicates that insure branches proceed to live code and the dead code is never executed. This requires the reverse engineer to identify the dead segments so they are ignored and to meta-diagrammatically replace the opaque predicates.

Preventative transformations target the automated deobfuscator tools. They are nonetheless a potential confusion for the human reverse engineer. An example is to reverse the direction of a loop. This can make an algorithm appear different and will require more work on the part of an reverse engineer to correlate the reversed form of the loop with its more conventional forward form.

The combinations of all of these transformations result in the need to abductively manipulate the code to reorder and reassemble it into a form where it can be recognized and understood.

4.3 The Demonstration System

The demonstration system for RDM applied the 711HPW/RH developed cognitive modeling framework and attached an IDA Pro multi-processor disassembler and debugger (Hex-Rays SA, Belgium). This was undertaken by TDKC after discussions with RH and based on some initial investigation of Microsoft Windows accessibility features. The TDKC developed software is the adapter for the IDA Pro debugger and enables the RML framework to control the debugger. The original plan was for RH to use the framework to construct a human cognitive model of reverse engineering. Although during the project, RH's reverse engineering expert departed AFRL, TDKC continued development of a basic demo. The end result is a simple demonstration of capability, but well short of an reverse engineering system. IDA Pro was chosen because it is a commonly used reverse engineering tool and also used by the reverse engineers

in AFRL/Rywa.

The design of the experiment is described below and then followed by the current state of the demo. An important constraint on the design of this system is that this is not an AI type design; it is a human factors simulation system.

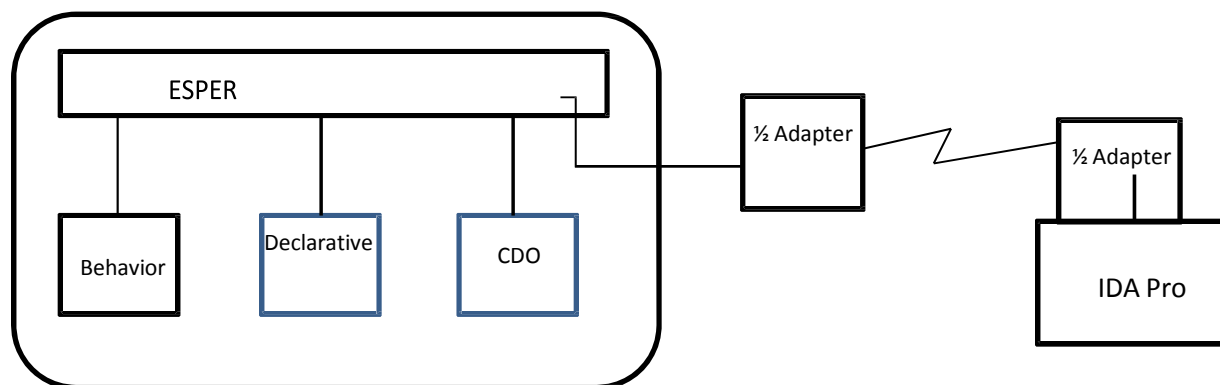


Figure 11. Demonstration System Diagram

The above figure shows the system. The RH framework runs in a Java environment. The debugger is in a Microsoft Windows and C# environment. These two domains are joined by an adapter with a TCP/IP network link with a simple applications protocol.

4.3.1 RH Framework

The RH component-based cognitive modeling and simulation framework employs the Discrete Event System Specification (DEVS) formalism [11] and uses the research modeling language (RML), one of the domain-specific languages (DSLs) supported by the framework. The architecture technically realizes a Discrete Event System Specification Modeling Language in a DEVSMML stack. From a user perspective, the framework consists of a set of DSLs that are automatically transformed into the DEVSMML and executed in a transparent M&S infrastructure. DSLs used in the DEVSMML stack are developed using the Generic Modeling Environment (GME), the centerpiece modeling technology of Model Integrated Computing (MIC), a general modeling and systems integration paradigm [12]. To develop a DSL, a meta-modeler specifies its abstract and concrete syntaxes in GME. The abstract syntax captures the concepts, constraints and relationships relevant to a domain using abstractions that exploit domain-specific knowledge and processes. The concrete syntax allows a modeler, acting more like an end user than a programmer, to visually/textually specify models that people with similar domain expertise can easily comprehend. To use a DSL, a modeler configures GME so that it supports the use of the DSL and then specifies models in the DSL's concrete syntax. The research modeling language (RML) is a DSL used to specify cognitive models in the DEVSMML stack. The abstract syntax of RML is influenced by the ACT-R cognitive architecture [13]. The concrete syntax of RML is designed so that a modeler with experience in ACT-R can specify behaviorally equivalent models at a higher level of abstraction.

The RH framework employed in this effort has 4 components. The backplane is ESPER, an open source event stream container. It is the communications link that binds the other components together. ESPER supports event objects that “flow” through the system and are delivered to the attached components. The

interface to ESPER is a SQL like language that declares what events are of interest to a component. The components can also source events onto the ESPER media.

4.3.2 Behavior

The framework uses the RML language described above. Behaviors are treated as small atomic units and are triggered by ESPER events. A behavior may in turn trigger other behaviors and actions via transmitting ESPER events. The RML behaviors allow the demo system to encode common procedural actions, opening a file or scrolling or searching on a screen.

The behaviors are contained in a flat space; that is, there is no hierarchy or organization to the set of behaviors. The only control mechanism used is event driven triggering.

4.3.3 Cognitive Domain Ontology (CDO)

GME has been used by 711HPW/RH to develop the Cognitive Systems Specification Framework (CS2F), a composition of domain-specific languages tailored to the requirements of specifying models and agents that base goal-pursuing behaviors on contingencies [14]. CS2F/CDO is a specification language based on system entity structure (SES) theory used to specify models of domain knowledge combining aspects of agents and the situational factors or contingencies constraining their behavior. SES theory is a formal ontology specification framework that captures system aspects and their properties [14]. Situational/agent properties, aspects and constraints can be formally captured in CDOs. CDOs are processed by an agent to determine what it should do. The framework constitutes a modeling architecture that explicitly supports the representation and processing of CDOs. This capability allows modelers to separate the what and how concerns and specify agents that generate process descriptions by using answers to the what question to identify and “soft-assemble” knowledge into contextually appropriate process descriptions. The cognitive domain ontology was extended by RH with a constraint solver system. This allows the CDO to declaratively reason. The import is that an entry in the CDO can express a relationship in a constraint and solve for missing terms. This is a very useful tool for knowledge used in tracing and understanding machine instructions.

4.3.4 Declarative memory

The declarative memory works like the Adaptive Character of Thought-Rational (ACT-R) declarative memory system. It stores the frames and retrieves them by similarity.

4.3.5 Debugger

The IDA Pro is a Windows based debugger and the plan was to use the accessibility layer of Microsoft Windows to gain access to the debugger in a mode that mimics the human user. The accessibility layer was implemented by Microsoft as a mechanism to support disabilities by providing other control and display modes for the user.

Since the IDA Pro debugger and accessibility interface logic run in a common language runtime environment on Windows, an adapter was need to get into the ESPER container (Java Virtual Machine). The choice was to use a network socket between the two environments and the adapter is constructed in two halves.

The adapter has two main functions. First it receives events from the ESPER stream and translates them into commands via the accessibility layer to control the debugger. This is relatively straight forward.

The second function, more complex than the control function, is to transmit screen contents into the ESPER backplane. The debugger screen is limited to text contents; the graphics is not used at this time. Adding the graphics would require a better vision system component on the model side and some form of either feature detection or raster scanning in the adapter.

The interface is constructed in a mode to simulate a human scanning the text screen. This is a compromise between the interface one would design for a pure computer tool and one that is purely a human vision scanning model. For a more human factors simulation, the adapter could be expanded to operate under control by the behavior models and would be designed to track the human's behavior. For the current adapter, the screen contents are broken into events that transmit a line of screen text. The behaviors can cause particular lines to be retransmitted as needed. This was to simulate the user scanning a new screen of data and also going back to relook at parts of the screen. This is the simplest interface given the ESPER stream model. Since ESPER stream events are transient, behaviors can get triggered but need to "back up the stream" to get to information that has passed by and was not part of the initial trigger events for the behavior.

4.3.6 State of the Demonstration

A basic demonstration was constructed. However, the CDO was not available and this limited what could be demonstrated. Second, the accessibility layer turned out to be less capable than what was indicated by the Microsoft documentation. This section will elaborate on what the demonstration was able to do and what needs to be done for the future.

The current demonstration uses RML behaviors to open and search debugger screens. The Microsoft accessibility layer turned out to be very cumbersome. First, the screen widgets do not all implement the accessibility layer, as the designers of IDA Pro did not work with the entire specification and the base classes could not always access the widget. This resulted in simulating the mouse operation instead of a direct access to the buttons and menu objects. Also, the id's of the widgets are not static but generated at runtime, requiring rescanning to find the id's. Since the screen contents were not directly available via accessibility, screen scraping was used. This resulted in a more complex design and more difficult than was first anticipated.

The RML behavior layer makes sense as a symbolic cognitive model; but stacking behaviors in a reentrant subroutine mode was not possible. Programmers are used to the typical subroutine model. Also, it was realized that encoding instruction set knowledge in behaviors would be difficult for two reasons. First, the lack of reentrant behaviors meant when evaluating one instruction requires the invocation of a sub-behavior that could loop back to a re-invocation of a behavior itself. Second, when instruction knowledge is encoded in behavior, multiple behaviors are needed for each instruction. The direction of search, forward in the code or backward needs separate procedures and separate procedures may be needed for each argument.

The CDO provides a useful tool for modeling knowledge of computer instructions. When reverse engineering it is necessary to be able to trace backwards and forwards through instruction sequences. Constraint propagation is a useful model of this knowledge, since we don't have to have separate models of each direction. Worse, if separate models of each direction are used, the models would also have to have variants for each combination of missing arguments. Another alternative would be a logic theorem prover that could invoke rules based on needed items.

With both the RML behaviors and the CDO, a control paradigm is needed to capture goal driven behavior. The problem of reverse engineering requires multiple goals to be maintained by the reverser. These goals often are satisfied in an opportunistic approach and new goals are constantly being identified. It is believed that such a model can be built in the CDO. Without a system to build the experiment, this remains

and untested thesis.

A walkthrough of the RML demonstration is in Appendix C.

4.3.7 Missing Opportunities

A missing aspect of the work is the access to graphic information. The IDA Pro debugger has graphic displays for gaining a more abstract and distilled view of the code. This is very useful to a human reverse engineer, and hence would make great sense to include in a cognitive model. The first question on designing this extension is how to deal with the graphic image. Humans scan and focus on the image. Also human processing is dependent on the application of feature extraction. On one extreme this system could extract a textual model of the graphic image preprocessed to make the reverse engineering task easier. On the one hand, one would mimic the human visual system. This choice has to do with the goal chosen: building an AI system for reverse engineering or modeling a human reverse engineer.

A goal for the work in RYWA was to look at the issues of abduction and sensemaking. The application of abduction was discussed in the paper [3]. The conclusion about the level of abduction available in the RH system is limited to the first level and perhaps the second level of the Hoffmann taxonomy. Again, without the CDO, it is hard to explore this issue to a satisfying depth.

4.4 An Abductive AI Architecture

This final phase of research in the task was to examine what kind of architecture might be required to enable abductive reasoning in an AI system. This is partially addressed by [4]. The paper represents the first steps toward an effective AI system. There is more work to be done to both demonstrate the capabilities of the system and to extend and add to it. This section will give an overview of the guiding principles and the system. This is followed by a prospectus of future work to enact this system.

From the studies that resulted in the paper in appendix A, a key conclusion was that two aspects were required to construct a system capable of stronger abduction. Stronger is used in the sense of going beyond the lower levels of Hoffmann's classification. The system has to have the ability to recognize situations by association. The use of abduction at the higher levels uses analogy and similarity. The similarity is not a quantitative, but rather a qualitative variety. These conclusions lead to a search for suitable ideas for building this kind of system and at the same time having some basis in human processing systems.

Another key driver in the search was the influence of Dr. Steven Rogers (Senior Scientist for Automatic Target Recognition and Sensor Fusion, Air Force Research Laboratory). Dr. Rogers has been concerned with the lack of progress in AI systems and has concluded that the missing element is consciousness. He defines consciousness by the ability to generate qualia. This position echoes concerns the author has had with parts of the AI field over the last several years.

The design of a successful abductive system is complex task. The second paper [4] is an initial step in the design of such a system. While the paper contains more detail, this section will give an overview of the proposed system. Cognitive scientists Alison Gopnik and Jeffrey Elman have both commented on the need to resolve the nature versus nurture argument found at the roots of the connectionist versus symbolic debate. This demands a bridge between the two or some newer model.

The study and the discussions with Dr. Rogers lead to an interesting thesis and realization. Symbolic AI systems have long underperformed their inventor's claims. One of the early arguments against these systems comes from John Searle's ubiquitously well-known Chinese Room. For over 20 years AI researches and philosophers have argued over this problem. There are two aspects of the problem that appear highly relevant to this work.

The first issue is ungrounded representation. In order for analogy and abduction to gain traction, the representation of knowledge must be rich and also grounded in the world of the problem. For a human this would mean grounded in the physical world. Linguists have long noted the effect of grounding in the world on language. Language is built on movement in the physical world and the objects found within it. Human processing power is built on perception in the physical world. From these observations, a viable conclusion is that the representations used in an AI system need to be grounded in the problem space with enough representational richness to enable analogy and abduction. This richness leads to recognizing situations and to allowing fine distinctions in the situations to drive abduction. This is discussed in the operation of the critic function in the architecture.

The second issue is self-reflection. Dr. Rogers will hold that the system needs consciousness in order to solve the kinds of complex problems represented in reverse engineering. While it is not hard to agree that such a goal will provide robust and interestingly capable systems, the thesis arrived at in this study proposes that there is a space outside Searle's room, but not fully conscious. In this space are many interesting systems that will be able to provide advanced capabilities.

These two aspects are credited as a thesis, because they are not yet proven.

There is one other argument that has come forward along with this thesis. Connectionism (artificial neural nets) is often proposed as a way around the limitations of symbolic systems. While there is good reason to examine this alternative, it is also easy to argue that many of the examples of neural systems, commonly found in the literature, are subject to the same limitation of a representational space as the symbolic systems. In essence, the inputs to the net are as constrained and ungrounded as the symbol systems. There are examples of systems that transcend this restriction, but they are few.

4.4.1 System Overview

Peter Gårdenfors proposed a bridge in [8] called conceptual geometry. The conceptual geometry system is used as sensor input subsystem (property concept layer in figure below). In order to recognize situations, a sparse distributed memory is placed above the conceptual geometry layer. The sparse distributed memory was designed by Pentti Kanvera [9]. The sparse distributed memory has three compelling features. First it supports very large address space (2^{1000}). Second it does associative look up. And, it can detect similar situations. While it can be implemented in a reasonable amount of storage, it is on the slow side.

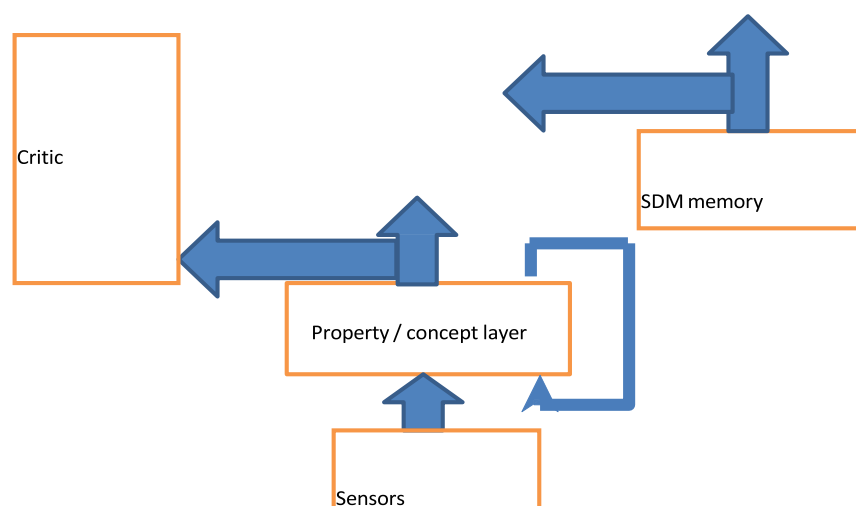


Figure 12. Abductive System Architecture

The third element is a critic which acts in a command by denial mode. The critic observes the situations and the property / concept layer outputs. It can detect marginal conditions by examining the property / concept layer. It can look for issues with the situations selected as similar, for example missing evidence or conflicting evidence. It is also a basis for extensions into analogous reasoning (future work). It will override the situation selected when issues are detected, similarly to an expert recognizing a situation is not quite as it seems.

These three components are discussed in more detail in the following sections.

4.4.2 Sparse Distributed Memory

The search for a mechanism that could recognize situations by associativity and could work with large representational spaces resulted in discovery of the work of Pentti Kanerva. The properties needed included a large space of inputs. This allows the use of much finer grain representations, below the symbolic level. While it is clear the human sensory system does feature extraction from the raw input, the feature space is still quite large. For example, the visual system (human and other mammals) has feature extraction prior to processing the visual scene to form a representation of that scene. A shortcoming of many symbolic and computational models of human cognition is the premature limitation of the representational space. It is not hard to postulate that the richer the situation representation is, the easier it is to detect subtlety in the situation.

The sparse distributed memory (SDM) as described by Kanerva is defined by a set of parameters. These are the logical space, the physical locations, and the access radius. These are not independent and must be chosen to work together. The word length to be stored is independent of the other dimensions, but usually includes the logical address to enable associative lookup (explained below). The physical locations are real memory locations, and far fewer than the actual logical space. The example given by Kanerva is a logical space of 2^{1000} in a space of 10^6 physical locations. Note that 10^6 is approximately $2^{19.9}$, which is only 2% of 1000. The access radius is 450 bits. The physical locations are randomly distributed over a binary hypercube of 1000 dimensions. The physical locations contain an 8 bit data value for each bit of data. For an associative memory, the location will hold 1000 bits. The access method for both reading and writing selects a set of physical locations by applying a circle placed on the surface of the hypercube with a suitable radius and a center at the logical address being accessed. The circle is calculated using Hamming distance. The radius is the distance from access address to the edge of a circle and is 450 bits in the example. (Hamming distance is the number of bits that differ between two binary numbers.) In order to read the memory, compute the average of the data values of the physical locations selected by the access radius. The resulting bit is one for a positive sum and zero for a negative sum. Data is written by adding its bits to the matching data byte in each physical location selected by the radius.

The access is associative, meaning that when an address is accessed the memory will return the value in closest location stored to the access address. In the normal mode, the memory has the data stored equal to the address it is stored with (in this case each location contains 1000 bits of data each stored in a byte). This address represents the situation and the data returned will then be the closed known situation, hence it is an associative lookup. It is also possible to add data to the data word in addition to the situation to allow the situation to be recognized along with an action to perform.

An extension to the base SDM is storing sequences. This is obviously useful for procedural memory. It can also be used to set up expectation, which is recognizing a situation then predicting the next situation that can be expected to follow. The memory is extended by doubling the word size. The word contains the address field as before and then a next address field. Similar to hash table implementations, the next

address is encoded by a mathematical operation to the address to help reduce collisions. The suggested operation is to permute the bits of the address.

In order to validate the claims and to ensure we could build SDM systems, Java prototypes were constructed for the basic memory and for the sequence extension. The physical memory was placed in a separate hard drive in a MySQL database. The MySQL database was used to make access to data easy and because MySQL is open source. In a production system, direct disk access would be easy to implement. Clearly the trade off in SDM is between time and both space and associativity. Test cases for the prototype run hours. No attempt was made to performance measure or to tune the system, since basic functionality was the goal. Also the code base is “graduate student code.” No attempt was made to build to production quality. The raw database size for the tests was 1.27 Gigabytes for basic and 2.38 Gigabytes for a sequence SDM, using the parameters specified above, e.g., 2^{1000} logical space. This is not at all unreasonable at current disk size / price. The system functioned well and lives up to its inventor’s claims.

4.4.3 Conceptual Geometry

The other base component of the architecture is conceptual geometry used to develop features, and concepts form the input space in a way similar to human capability. The use of the term geometry by Gårdenfors is in the sense of Euclid, relationships without recourse to measure. He draws on the ability of humans to deal with close and betweenness in domains like color and sound. The basic structure of the scheme defines domains, concepts and properties. Domains are groups of properties and concepts they are related. Properties are determined by sensory input alone, for example “red”. Concepts are built from both concepts and properties. Figure 13 below illustrates this structure.

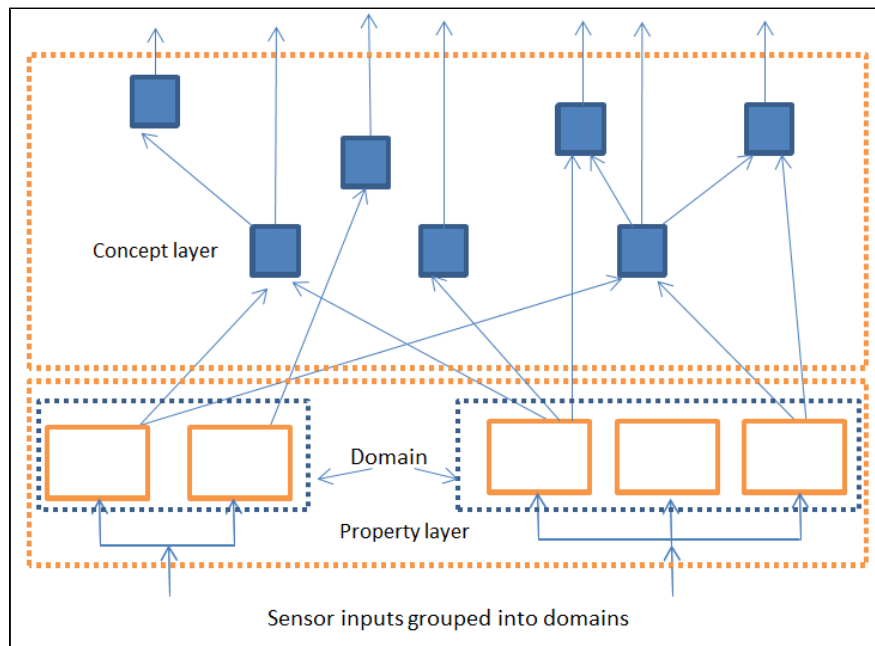


Figure 13. Conceptual Geometry

Each property or concept element in the figure is a computation over an input space. The input space is divided into convex regions (e.g., a tessellation) that identify a concept or property value. Gårdenfors

argues that humans can learn convex regions, but have trouble with properties or concepts that are defined by concave regions. Learning is easily implemented by adjusting the tessellations boundaries or adding in new tessellated regions.

4.4.4 The Critic

The overarching project has been to look at abduction and sensemaking problem solving. The present level of development of the critic is still weak in abductive ability by Hoffmann's taxonomy, but it is a step toward stronger abduction and also a step toward expert sensemaking. The sensemaking problem is often a straight forward recognition of a situation that has been encountered before. The challenge is recognizing when the situation is almost the one seen before, but misses some critical aspect.

While the system is designed to recognize situations and direct actions by setting goals, the critic has an oversight function with the ability to interfere. It has a focus on surprise and abduction. This is implemented in two modes – by exploration of alternatives and by reflection during backtracking on failure. As a practical note, there are two caveats on the current state of the critic. The description provided here is our current view and is still under development to strengthen the critic. Also, as we apply this system to different problems, the role of the critic can be restricted. Some domains demand a critic with restricted actions in order to assure the system behaves in a controlled fashion. In this case the critic can log its observations for latter study. This mode is useful in new domains.

Surprise can occur in two ways. There is the obvious mode when a goal, selected by the system, fails to be obtained. Such a failure triggers backtracking. The other is the preemptive detection of “not quite right” situations, which is covered first.

The SDM is used to find situations which match the current set of concepts and properties encountered in the environment. This is done by associative matching in the SDM. In order to consider detection of “not quite situations,” the agent must consider how these situations can arise. There appear to be two main cases that are tractable. The first examines the situational assessment for marginal values. The strength of Gårdenfors' system is its geometrical formulation, using relative values, not absolute numerical values. Thus in classifying properties and concepts, we can use the geometrical notions to mark properties that are close to classifying as a different property. This marginal boundary analysis can be used to look at other possible goals in contrast to the initial situation. Since this can be a costly activity, it is triggered by an examination of the “strength of evidence.” This is done by considering first the marginal properties and concepts in the situation and the concepts or properties that are in the found situation, but not in the input (missing evidence). If nothing appears to be marginal, the analysis is not triggered. Secondly, during the actual situation, the critic analyzes the concepts and properties expressed by the input to the sensors. Here, the system looks at concepts and properties not included in the situation produced from the SDM. Alternatives are generated by suppressing some of the concepts in the found situation and probing the memory for other possible situations.

On failure of a goal, the system backtracks and performs the same kinds of analysis describe above. However, the failed goal provides additional information, that of falsifying that choice of the goal as a counterfactual representation. This forces reconsideration of that goal and to a lesser degree the prior goals that supported the choice. The “blame” for the bad choice is passed back and is applied with a reduced degree of weight as the backtracking proceeds backward.

Now turning to abduction, Hoffmann uses two dimensions to construct a 3 by 5 matrix of abductive types. The simpler forms of abduction are implemented implicitly by the matching in the SDM. Since we are looking for the closest match to a situation, evidence for the situation that is missing will be assumed. The more interesting case is how to apply shifts in the structure of a representation. This is made possible by a meta-information level over Gårdenfors' system. By producing a meta-level over the domains, one can

rank similarity measures between domains inductively. This similarity can then be used to shift concepts.

Analogical reasoning can be used to modify a situation by substituting concept A for a concept B, but only when their underlying domains have relative similarity. This will be represented by law or model shifts in Hoffmann's taxonomy (rows 3 and 4 in table 1).

5.0 ABDUCTION AND SENSEMAKING CONCLUSIONS

The quest for adaptive responsive AI systems to employ in domains like reverse engineering can benefit from abductive reasoning. Further Hoffman and Magnani clearly show that abduction is a complex process. The demo system constructed for this task was aimed at cognitive modeling. While it established that the modeling platform can be successfully linked to a debugger, it is not the basis for an AI system. The possibilities for applying an AI system for reverse engineering are the subject of future work for this task area.

Much of the focus in this task was on studying the application of abduction in support of the cognitive modeling work that was planned for the RDM STT. However, the next phase of effort can be profitably focused on building abductive tools to apply to the reverse engineering and software protective problem areas. This matches the new research program for RYWA. There are two main approaches that can be used. First a classical AI system approach can be used to crack reverse engineering problems. Second, and more useful, is to build a "wingman" solution that acts to assist the human reverse engineer with their task. In either case the set of problems to be solved is the largely same. The wingman solution also requires some additional human computer interaction issues to be solved in order to make a symbiotic system.

In order to set a context for the task of reverse engineering, a classical AI agent view is used. That is, the reverse engineering work is carried out by a reverse engineering tool (RET). The RET is an agent working within an environment. While there can be many environments used in reverse engineering, the debugger is chosen as the environment of this RET. For the remainder of this discussion, when an RET is referenced, it is an agent interacting with a "debugger". The term debugger is placed in quotes for a very significant reason. A debugger is a tool designed for human use. For an artificial agent system, the interface provided by the debug tool unnecessarily limits the agent. However, for ease of discussion we will use the image of an agent interacting with a debugger. If a symbiotic system should be built, then the debugger interface would be used by the human and would be observed by the agent. However the agent's interface to the code under reverse engineering does not have to be limited to the debugger screen. However the debugger screen under human control would serve as a focus of attention for the agent.

One conclusion about abduction is that, like most AI problems, representation is critical. Searle's Chinese Room has long been a fun conundrum for AI research. There are several issues raised by the argument that bear on the current work. These are the restriction of the input space and the reasoning space. First of all is the lack of symbol grounding. By restricting the input to the "room" by passing in only Chinese texts, the room fails to have a rich environment to interact with. This causes the "room" to have nothing to reference the symbols in the real world. This lack of grounding helps insure the "room" cannot have a large space of possible reactions or "feelings" about the symbols. It also short cuts the "world is its own best representation" mantra of the "new AI" advocates (e.g. Rodney Brooks). Without something to reference a symbol to within a world, the system is pure syntax without semantics as Searle argues. Now, that is not to concede the ability of a symbolic system to have qualia or feelings, just that a pure symbol space is insufficient.

Second the "room" is given a very procedural oriented control system to enact. The reasoning space is therefore a limited as is the input space of the "room". The procedures of the room are first and foremost limited by the entirely syntactic input space. The procedures can therefore only compute on the basis of a set of symbols. In order to break this limit, the system needs to have something else to choose its actions. This seems to demand a way to evaluate choices in terms of something more. A plausible mechanism is to

have feelings, emotions or qualia about the symbols. Just as the human feeling “sad” may place a sad tone on the answers to the questions in the Chinese room, so too would a processing system respond with differently based on emotional content. This is a departure toward a conscious system.

The architecture proposed in the paper [4] is an approach to dealing with the above issues. However, it needs extension to include more “self-reflection” and a primitive machine consciousness. This remains an area to investigate. However there is a community of AI researchers that are also working into this space. In order to provide a bit of reason to what may sound as a farfetched plan; consider the philosopher Nagel and his argument about the bat . If a bat is conscious, it is hard for us to imagine what its consciousness is like. A conscious agent for reverse engineering would inhabit a world of program code, and a limited knowledge of human intention.

The human mind is the product of a long evolutionary process. While progress is being made decoding the mind’s secrets, this work cannot wait for that answer. However, two approaches exist to shortcut the evolutionary process and its time scale. First, knowledge engineering can be applied to gain a human reverse engineer’s knowledge set. Also, some of the knowledge needed is part of the software literature and knowledge, while not small, it is documented.

The second source is the application of machine learning. This is used to prime the system and also to cause the system to learn as it works. This is a new task area already underway in RYWA.

The approach described in this future work is open research, but the initial studies undertaken in the task lead to belief that such systems are feasible.

6.0 REFERENCES

1. Josephson J. R. and Josephson S. G., “Abductive Inference”, Cambridge Press, New York, NY, 1996.
2. Klein, Gary, “Sources of Power: How People Make Decisions”, MIT Press, 1999.
3. Hartung, R, and Weigand, K. “Abduction’s Role in Reverse Engineering Software”, Proceedings of NAECON 2013 Conference, 2013
4. Hartung R, and Weigand K. “Microgenetic Critic for Situation Assessment Supporting Abduction and Surprise”, ICAI 2013 Conference.
5. Hoffmann, Michael, H. G. “Theoric Transformations and a New Classification of Abductive Inferences”, Transactions of the Peirce Society, Vol. 46 No. 4 pg 570 – 590, 2011.
6. Magnani, L., “Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning”, Springer Verlag, Berlin/Heidelberg, 2009.
7. Bryant, A. R., R. F. Mills, G. L. Peterson, M. R. Grimaila, “Software reverse engineering as a sensemaking task,” Journal of Information Assurance and Security, vol. 6, no. 6, pp. 483-494, 2011.
8. Gärdenfors, Peter, “Conceptual Spaces, the geometry of thought”, A Bradford book, MIT Press, Cambridge Mass, 2000.
9. Kanerva, Pentti, “Sparse Distributed Memory”, A Bradford Book, MIT Press, Cambridge Mass, 1998.
10. Collberg, C, C. Thomborson, and D. Low, “A Taxonomy of Obfuscating Transformations,” Technical Report #148. New Zealand: Department of Computer Science, The University of Auckland, 1997.
101. Zeigler, B.P., Praehofer, H. & Kim, T.G. “Theory of Modeling and Simulation”, 2nd Edition. Academic Press, 2000.

12. Sztipanovits, J. & Karsai, G. "Model-integrated computing", *Computer* , 30 (4), 110-111, 1997.

13. Anderson, J.R. "How Can the Human Mind Occur in the Physical Universe?", Oxford: OUP, 2007.
14. Douglass, Scott and S. Mittal, "A Framework for Modeling and Simulation of the Artificial," in *Ontology, Epistemology and Teleology for Modeling and Simulation*, Intelligent Systems Series, ed. A. Tolk, Berlin/Heidelberg/New York: Springer-Verlag, pp. 271-317, 2012.

Additional Bibliography on Knowledge Glyphs:

1. Bisantz, Pfautz, Stone, Roth, Thomas(-Meyers), and Fouse (2006). "Assessment of Display Attributes for Displaying Meta-Information on Maps.", presented at Human Factors and Ergonomic Society Annual Meeting and published in proceedings.
2. Bisantz, Stone, Pfautz, Fouse, Farry, Roth, Nagy, and Daniels (now Thomas), "Visual Representations of Meta-Information." *Journal of Cognitive Engineering and Decision Making*, 3-1, pp67-91, 2009.
3. Gray, W. D., *Integrated Models of Cognitive System*, Oxford University Press, Oxford New York, 2007.
4. Josephson J. R. and Josephson S. G., *Abductive Inference*, Cambridge Press, New York, NY, 1996.
5. Preston, J. and Bishop, M., *Views into the Chinese Room: New Essays on Searle and Artificial Intelligence*, Oxford University Press, Oxford New York, 2002.
6. Repperger, Aleve, Thomas, Miller, and Fullenkamp, "Complexity of Visual Icons Studied via Signal Detection Theory." *Perceptual and Motor Skills* (105) 287-298, 2007.
7. Repperger, Thomas(-Meyers), Aleva, Fullenkamp, and Phillips (2007). "Investigating Display Icon Complexity via Information-Theoretic Constructs." Presented by Dr. Repperger at Aerospace Medical Association (AsMA) Annual Scientific Meeting and published in proceedings, 2007.
8. Repperger, Thomas(-Meyers), Aleva, Fullenkamp, "Improving Decision Making using Complex Icons and Information Theory," Spring review of the AFOSR 'Cognition and Decision' program, Fairborn, Ohio and published in the proceedings, 2006.
9. Thomas(-Meyers) and Whitaker, "Knowledge Glyphs as a Tactic for Multi-Planar Visualization of Simulation Products." Presented at Winter Simulation Conference and published in the proceedings, 2006.
10. Thomas, Whitaker, and Barbier. "Knowledge Glyphs: Iconography to assist in Command and Control", presented at Visual and Iconographic Language conference and published in proceedings, 2007.
11. Whitaker and Thomas, "Knowledge Glyphs: Visualization Theory Development to Support C2 Practice", presented at and in the proceedings of the 10th International Command and Control Research and Technology Symposium - The Future of C2, 2006.

Appendix A RML Stand-alone Demonstration Walk-Through

24 July 2013 - TDKC, Michael Carter (mcarter@tdkc.com)

Introduction

This document describes the steps to properly startup and run the unclassified version of the RML demo. This demo version is packaged with unclassified data sources.

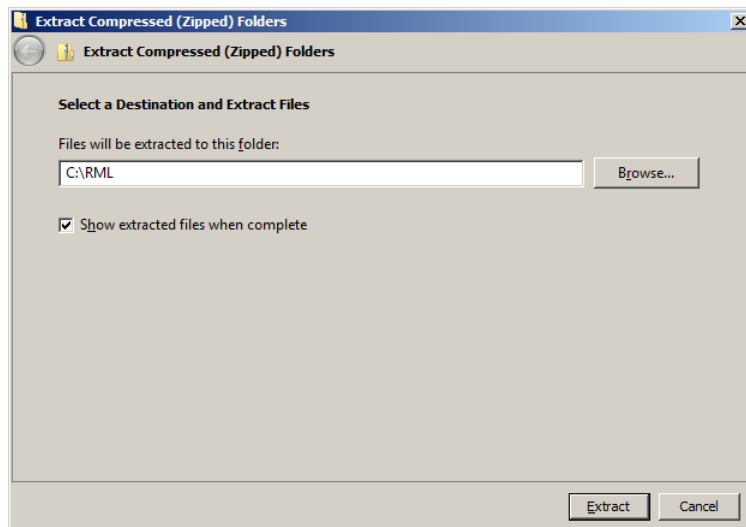
Requirements

This demo requires the following:

- Windows 2000 Professional / XP / VISTA / 7
- 2+ GB memory
- Java JRE/SDK 1.7.x

Installation

1) To install the RML Demonstration, unzip the installation package (RML_20130724.zip) into an appropriate directory. The easiest way to do this is to right click the zip file and select **Extract All...** This will bring up a dialog that allows you to select where to extract the files. For purposes of this document, the archive is unzipped into the directory C:\RML. This creates the directory C:\RML, which contains all the files necessary to run the demonstration.



Startup

- 1) Use File Explorer and navigate to the C:\RML directory
- 2) Double click on the **RML.bat** file.

Demonstration Component Overview

There are several main components to the RML Stand-alone demonstration environment:

- 1) **RML.jar**: This contains the compiled GME models for RML, as well as ESPER utilities and the

components needed to communicate to the IDA Pro Server backend.

- 2) **RMLCommunications.exe**: This is the C# program that communicates between ESPER and IDAPro. This requires the RDM.exe program to be located in the same directory.
- 3) **demo_stackframe.exe**: This is the program that will be used within the debugger for this demo.
- 4) **IDA*Pro**: This program needs to be loaded on the demo system and be available from the command line.

Demonstration Walkthrough

Starting the RML.bat batch file should start the two models used for this demo (runner and groupLines). A Command prompt window should appear, listing the registered events, as well as showing the number of models running (in this case it should be 2).

```

C:\Windows\system32\cmd.exe
C:\RML>java -classpath RML_lib -jar RML.jar CEPBMs.runner CEPBMs.groupLines

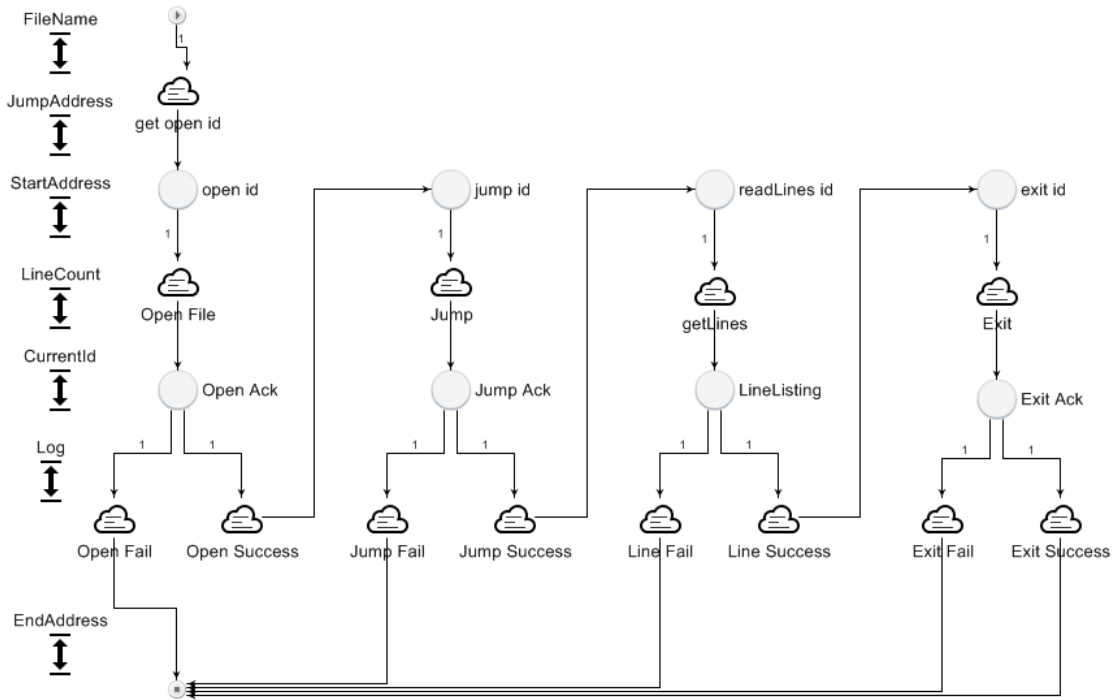
=== Event Types ===
Ack(String data, String id, Boolean success) -> ResponseEvents.Ack
ArrayList(boolean empty) -> java.util.ArrayList
CloseFile(String id) -> RequestEvents.CloseFile
GetLine(String id, Integer lineNumber) -> RequestEvents.GetLine
GetLines(Integer end, String id, Integer start) -> InternalEvents.GetLines
ID(String id) -> InternalEvents.ID
IDAProReady() -> ResponseEvents.IDAProReady
Line(String id, String line, Integer lineNumber, Boolean success) -> ResponseEvents
.Line
LineListing(String data, Integer end, String id, ArrayList lines, Integer start,
Boolean success) -> InternalEvents.LineListing
OpenFile(String filename, String id) -> RequestEvents.OpenFile
Scroll(Boolean down, String id, Integer numLines) -> RequestEvents.Scroll
groupLines_Transition(String stateFrom, String stateTo, Long tick) -> CEPBMs.gro
upLines$groupLines_Transition
runner_Transition(String stateFrom, String stateTo, Long tick) -> CEPBMs.runner$
runner_Transition
=====
2 model(s) running.

```

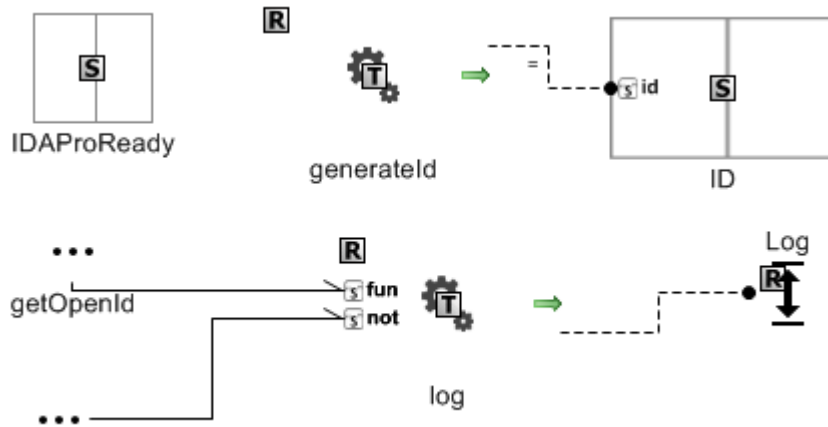
Next, the RMLCommunications program needs to be started by double clicking on the RMLCommunication.exe program. This will start a second command prompt, and then IDA*Pro will start up. The two programs will then begin communicating, passing messages back and forth. These can be seen in the RML screen.

The demo should run to completion with no human intervention. Watching in the command window for RML.bat, the user should see the following happen

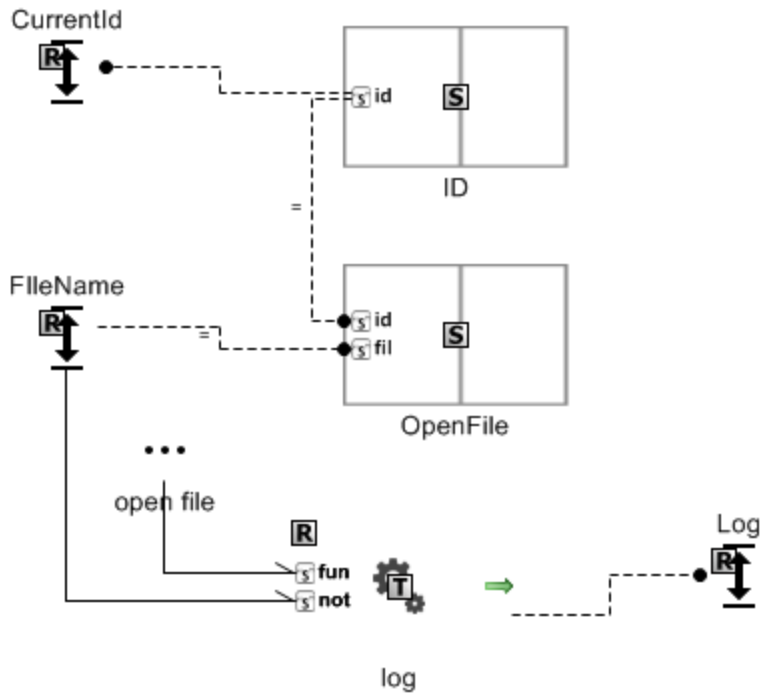
- 1) When RMLCommunications connects to the open socket established by the RML program, an “IDAProReady” event is generated to start the runner model.



- 2) The runner model transitions from the start state to the “open id” state, calling the calculation “generateId” to generate an id to be used to track the open operation, which is pushed as an ID event.



- 3) The runner model receives the ID event, transitioning to the Open Ack state and generates an OpenFile event that is pushed to ESPER.



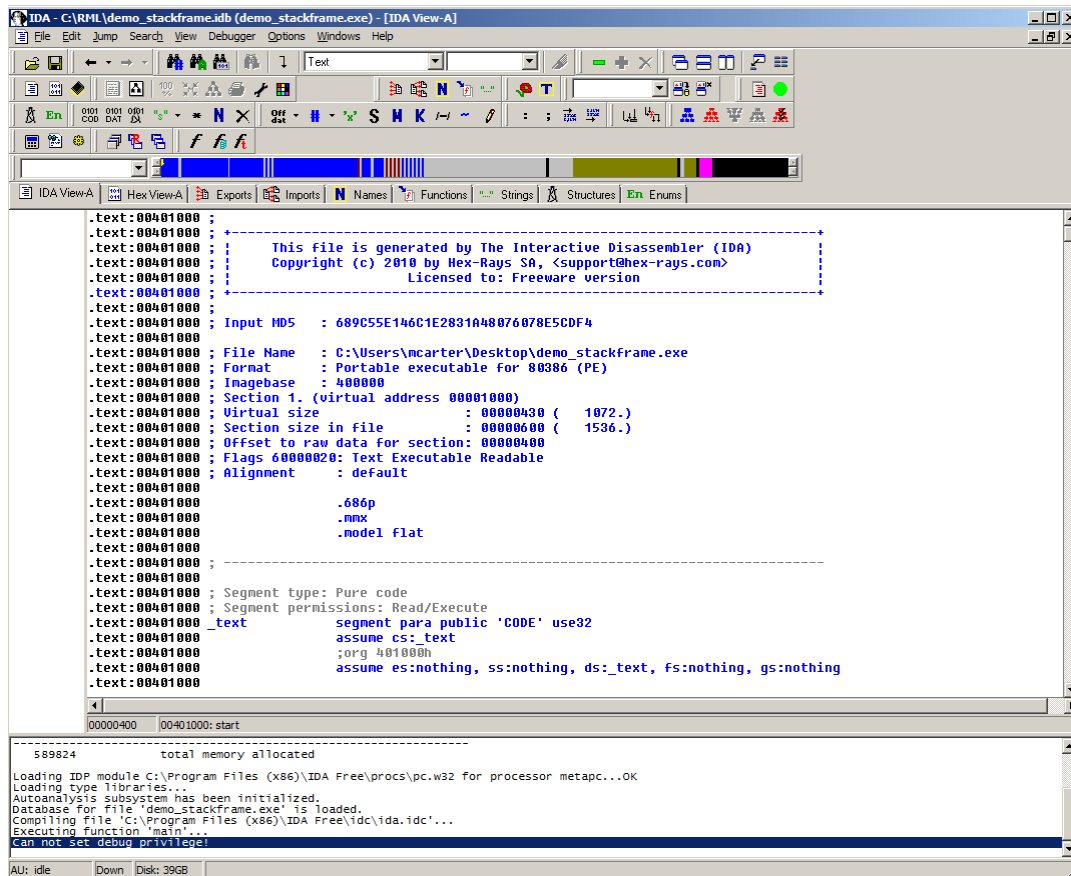
- 4) The OpenFile event is then transformed into a NetworkRequest, and sent over the socket to the RMLCommunications server.

```

<NetworkRequest id="1">
<command timeout="10000">open</command>
<arguments>
<argument name="path">demo_stackframe.idb</argument>
</arguments>
</NetworkRequest>

```

- 5) The request is received by the IDA*Pro controller, which after parsing the request, proceeds to emulate the user opening a new file



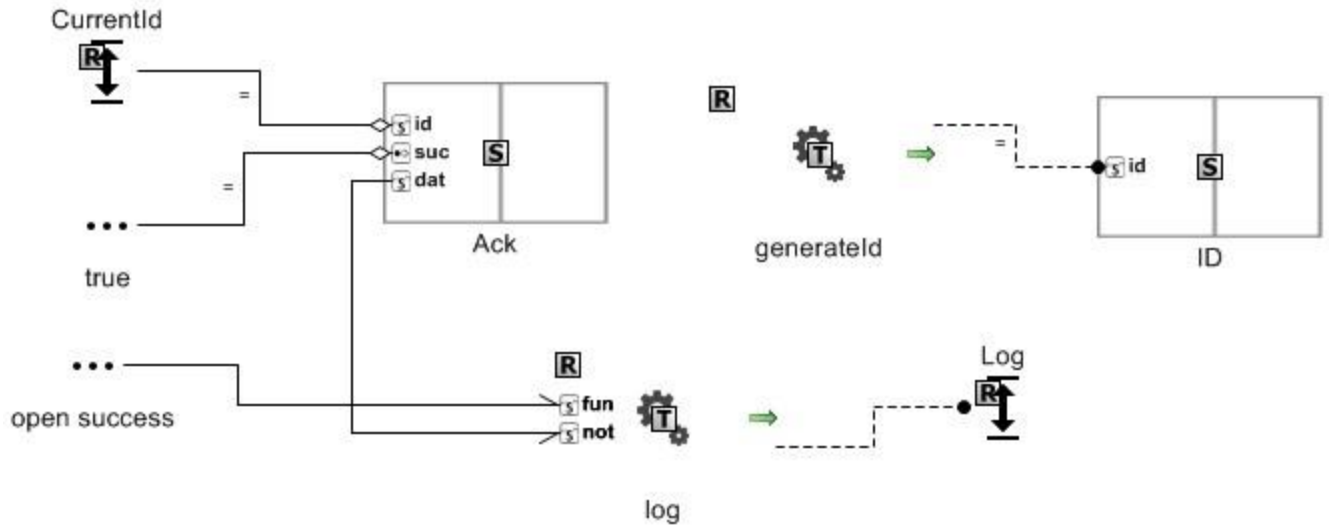
- 6) The controller then generates a NetworkResponse indicating the status of the previous request and puts it on the socket in response to the request.

```

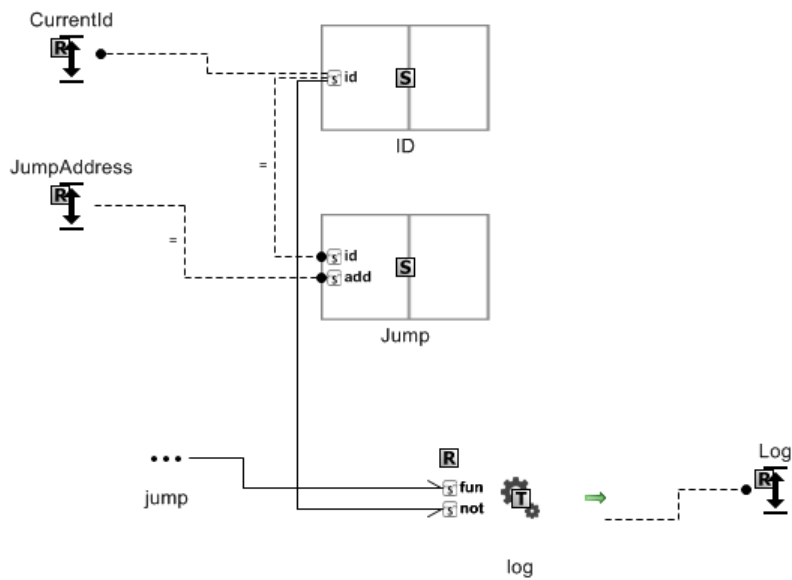
<NetworkResponse id="1" success="True">
<data></data>
<command timeout="10000">open</command>
</NetworkResponse>

```

- 7) The NetworkResponse is parsed and converted to an Ack event, which is pushed to ESPER.
8) The Ack event triggers a transition based on the value of the success attribute. False (the request fails) causes the model to transition to the end state. True transitions to the scroll id state, generating a new id and an ID event.



9) The ID event transitions the model to the Jump state, generating a Jump event to jump to the specified address in the debugger.



10) The Jump event is received, and transformed to a NetworkRequest that is pushed through the socket.

```
<NetworkRequest id="2">
<command timeout="10000">jump</command>
<arguments>
<argument name="address">00401320</argument>
</arguments>
</NetworkRequest>
```

11) The IDA Controller receives the request, and scrolls the window down, sending back a

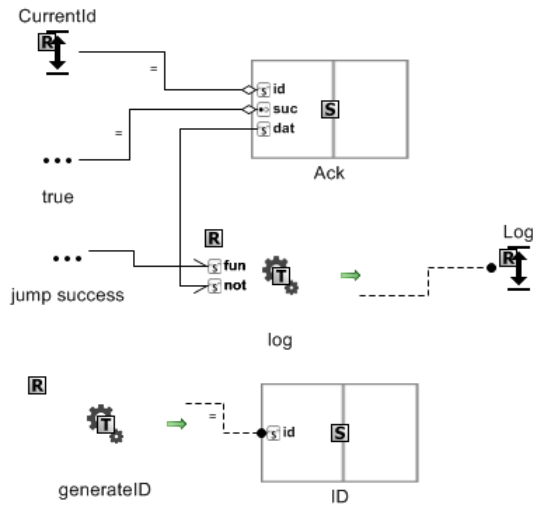
NetworkResponse with the status of this event.

```

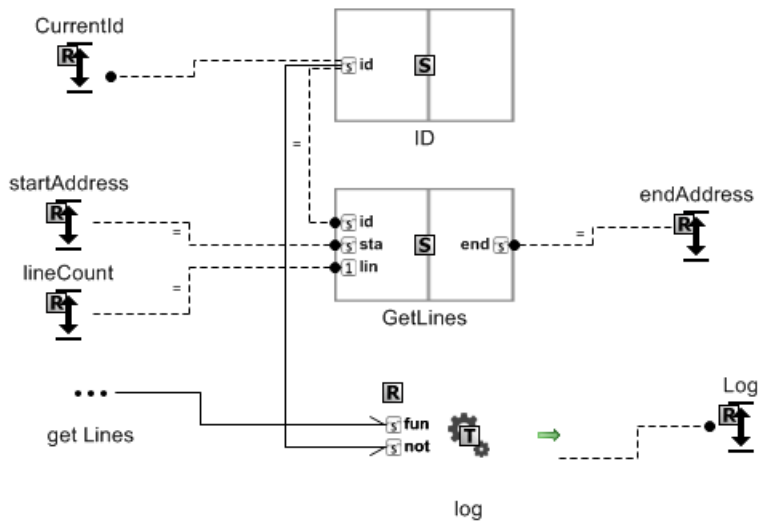
<NetworkResponse id="2" success="true">
<data/>
<command timeout="10000">jump</command>
</NetworkResponse>

```

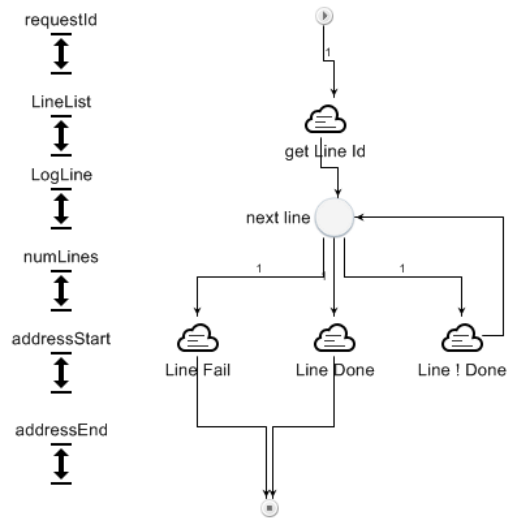
12) Failure of the scroll command will transition to the end state, success will generate a new id and ID event



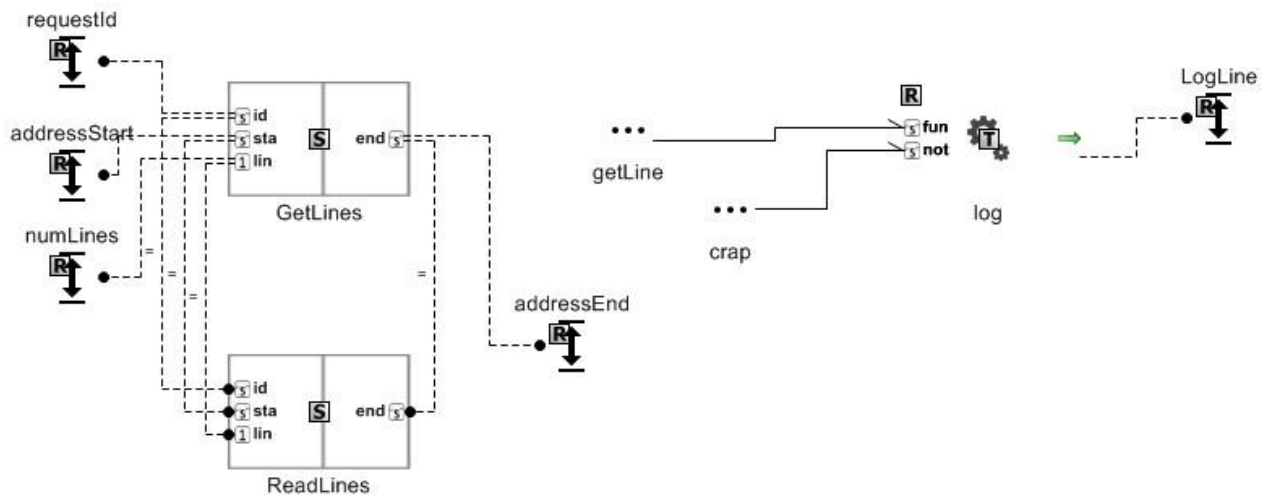
13) The ID triggers the transition to the LineListing event, generating a GetLines event. The model waits in this state until all the requested lines are returned.



14) The GetLines event triggers the groupLines model to transition out of the start state to the line id state, by generating a new id for the get line request



15) The ID triggers the transition to the next line state, generating a ReadLines request



16) The ReadLines is received and translated to a NetworkRequest and pushed onto the socket

```

<NetworkRequest id="3">
<command timeout="10000">read_line</command>
<arguments>
<argument name="startAddress">00401350</argument>
<argument name="endAddress">00401376</argument>
<argument name="numLines">8</argument>
</arguments>
</NetworkRequest>

```

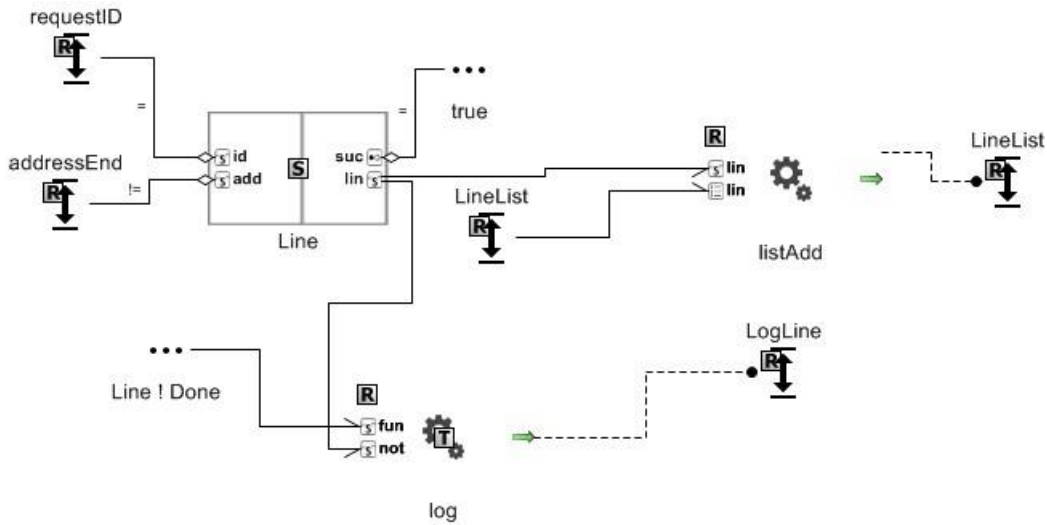
17) The IDA Controller receives this event, parses it, and performs the request, retrieving the line 8 lines requested. These are returned in a NetworkResponse across the socket.

```

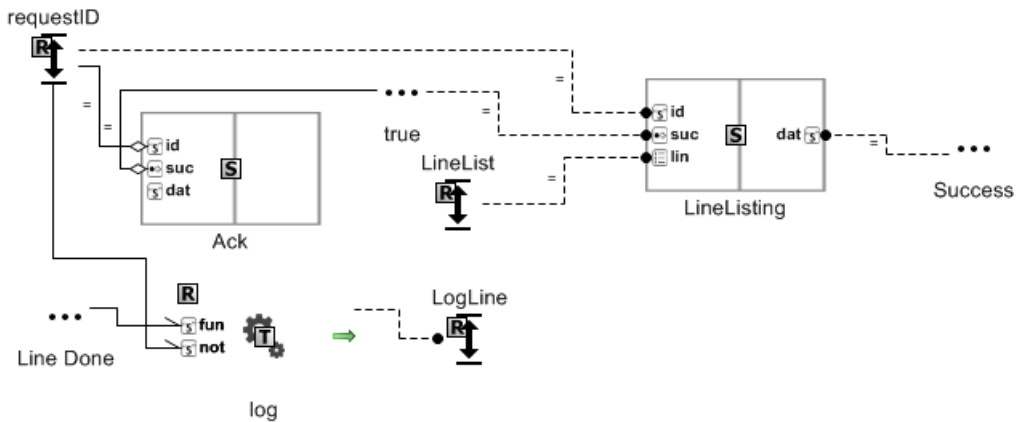
<NetworkResponse id="3" success="true">
<data.text:00401350 var_C          = dword ptr -0Ch
.text:00401350 var_8              = dword ptr -8
.text:00401350 var_4              = dword ptr -4
.text:00401350
.text:00401350
</data>
<command timeout="10000">read_line/command<
</NetworkResponse>

```

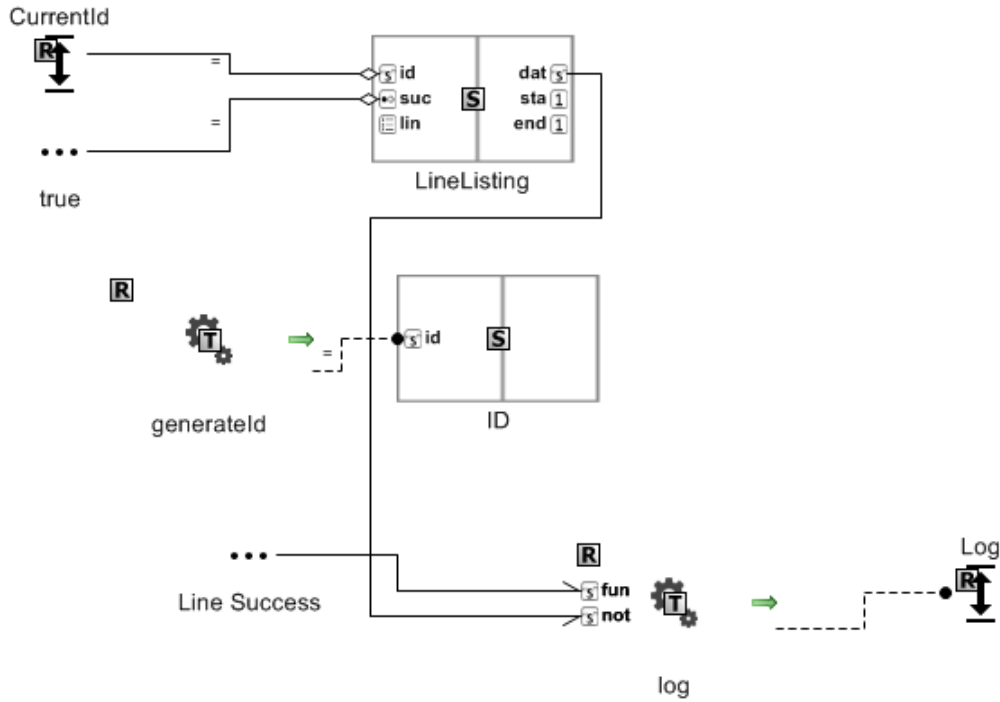
18) The NetworkResponse is parsed, and converted to a Line event, containing the data and the status of the request. The Line event can trigger one of three transitions based on the success flag and the line number returned. If the line returned is less than the maximum line for the request, then the line is added to a list, a new id generated, and the model is transitioned in a loop back to the line id state.



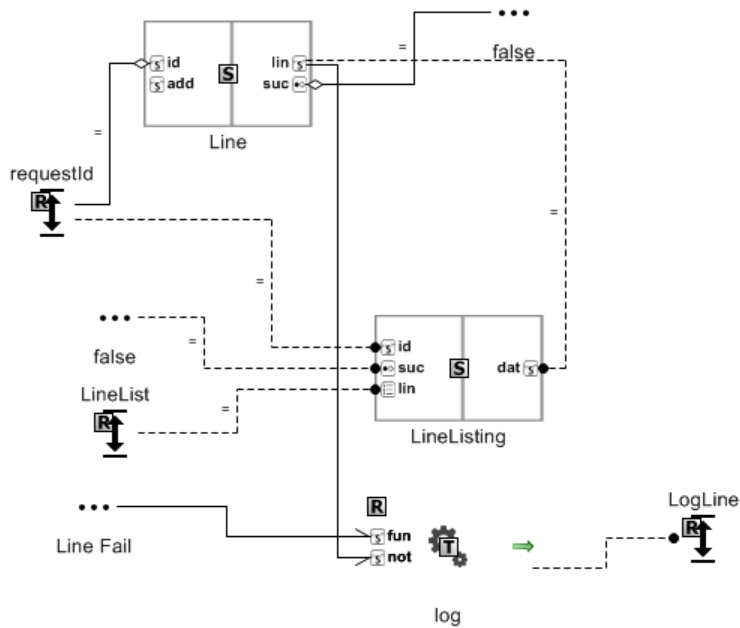
19) If the line number returned is equal to the maximum line for the request, a third transition occurs to the end state of this model, generating a LineListing event from the accumulated lines



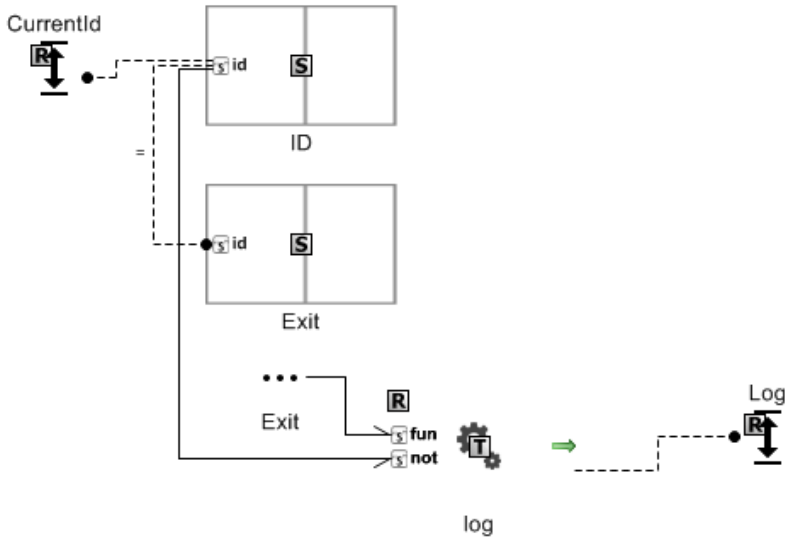
20) Receipt of the LineListing event triggers the transition back in the runner model to the close id state. A new transaction id is generated.



21) In the case of an error, the LineListing event is still generated, but this time containing the error.



22) The ID event causes the model to generate an Exit event, which closes the currently open file and exits the debugger.



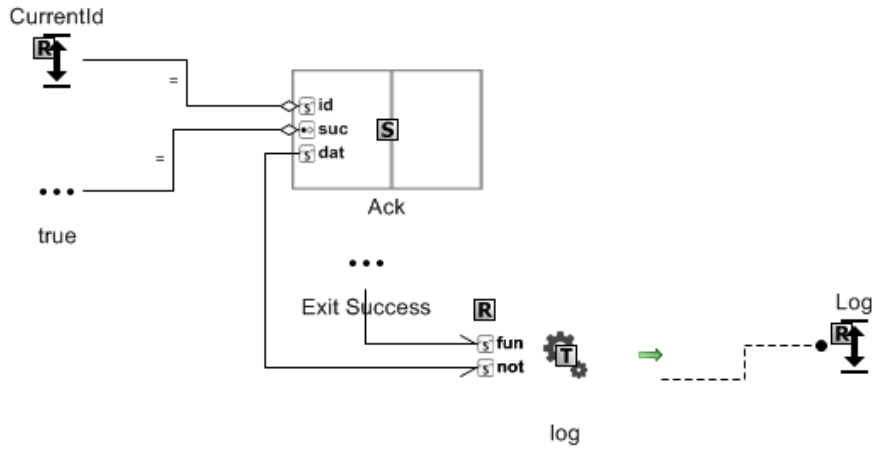
23) The CloseFile is translated into a NetworkRequest that is pushed over the socket to the IDA Controller

```
<NetworkRequest id="4">
<command timeout="10000">exit</command>
</NetworkRequest>
```

24) The controller parses the event, closes the file, and exits the debugger, returning a NetworkResponse with the status over the socket.

```
<NetworkResponse id="4" success="true">
<data/>
<command timeout="10000">exit</command>
</NetworkResponse>
```

25) The NetworkResponse is transformed to an ACK event, which transitions the runner model to the end state. This occurs for both the successful and failure cases here.



26) At this point the demo is complete. Type “quit” in the RML window to exit the RML model runner. Close the RMLCommunication window (via the window controls).

- THIS PAGE LEFT INTENTIONALLY BLANK -

Microgenetic Critic for Situation Assessment Supporting Abduction and Surprise

Ronald L. Hartung and Kirk A. Weigand

¹ The Design Knowledge Company, Fairborn, Ohio, USA

² Air Force Research Laboratory, Wright-Patterson AFB, Ohio, USA

Abstract - One of the core problems in cognitive systems is recognizing a situation and grounding that recognition to a sensed representation occurring in the world. This recognition also requires representations that can be analyzed and critiqued. This critic function challenges symbolic approaches, which become limited by the fuzziness of underlying meanings as contrasted to the crispness of symbols. Symbolic representations also lack the ability to translate sensor data to more abstract concepts while still preserving the underlying relationships. The proposed microgenetic abductive system is inspired by humans' adaptive solution to this representation problem through their cognitive evolutionary development. This paper describes a system and approach designed to advance abductive situation recognition by bridging form sub-symbolic input to concepts that allow critical analysis of surprising events.

Keywords: abduction, cognitive system, sub-symbolic representation, world representation, situation recognition

1 Introduction

This work comes from a project to apply research from cognitive science, logic and process philosophy to solve problems in software security. In particular, abduction and surprise have been a major focus of the study. [1]. The ability to recognize surprising situations and to abductively address problem solving in complex domains is an attribute found among human experts. These human abilities are both elegant and flawed in that they usually work yet sometimes fail as well. Abductive inference's ability to find the best explanation of many possible explanations developed from a long evolutionary heritage that enabled the human species to survive. Human's ability to adapt appears to come from learning embedded in long and short term processes of growth: 1) Development of the human species (phylogenesis), 2) development of individual humans (ontogenesis) and 3) development of the mind over the life of an individual (microgenesis). [2][3][4]

On a pragmatic bent, this framework attempts to engineer a working artificial intelligence system that straddles the divide between symbolic, linguistic and semantic models in Artificial Intelligence (AI) and bottom-up artificial neural network and automata systems. However the focus of the paper is on the component to recognize situations and also

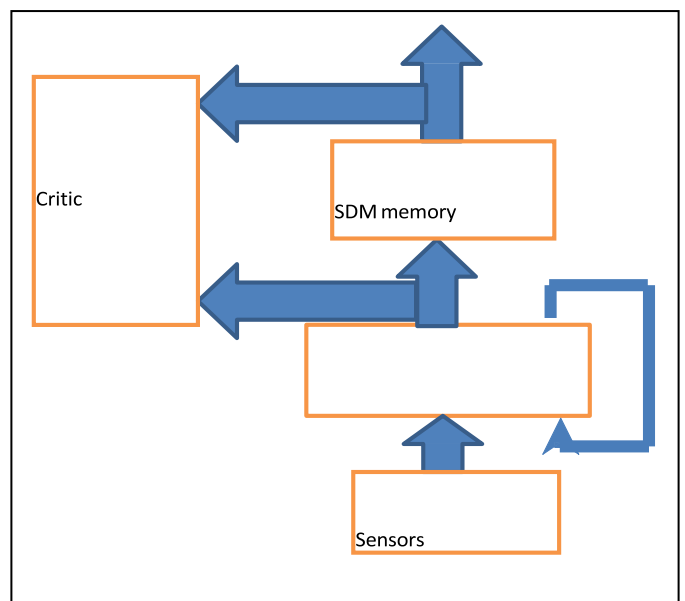
evaluate the situation in an effort to “detect when the situation is not what it seems;” that is, to encounter surprise and adapt in an efficacious manner that serves to benefit the agent's purpose with respect to the novel states encountered from the surprising situation.

2 Background

The proposed system is inspired by cognitive science but also results from an engineering approach. Inspiration from cognitive science stems from an ongoing debate between symbolic and connectionist schools of thought on the most productive scientific path for building cognitive models. Gopnik [5] and Elman [6] call for a systematic understanding of development in cognitive science to help resolve the Nature versus Nurture argument found at the roots of the connectionist versus symbolic debate. Gopnik calls for a bridging of the symbolic systems with connectionist systems using evidence from her studies of ontogenetic development. Microgenetic theory offers a top-down coherent way that humans may learn during moment-by-moment human thinking [2][3]. Gardenfors' theory of concept geometry offers a grounding for these desiderata [7].

3 Situation Recognition System

The situation recognition system is composed of three components, as shown in Figure 1.



The property and concept layer is constructed using conceptual spaces and is described first. The SDM (sparse distributed memory) is described next and is used to match and recognize. The critic, still in its infancy, is described last.

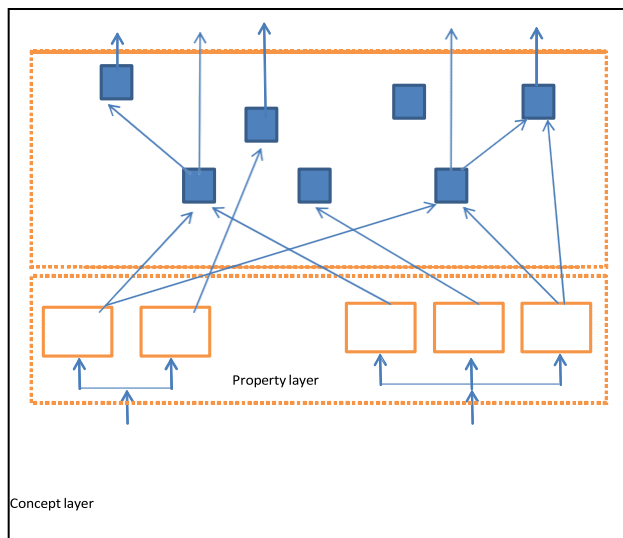
3.1.1 Conceptual Spaces

The model describes a process that takes sensory input to concepts, based on Gärdenfors' conceptual spaces [7]. The terms defined in the model are summarized as follows. The sensory system has some simple properties. Input signals are treated as relative. The important feature required by Gärdenfors' model is that similarity and betweenness are persevered. That is classical geometry, not precise measurement, is the essence of the output of the sensory system.

Sensors are grouped into related sets called domains. Gärdenfors' example of a domain is color, which allows the component sensor to measure hue, chromaticity and brightness as defined by the NCS color system [8]. These three dimensions produce a space which represents the domain.

Properties are partitions of a domain. For example, red is represented by a region of the color domain. A significant property of these regions is their convex shape. Convex shapes are easier to learn. Gärdenfors denotes these as natural properties. One possible method for the partitioning is Voronoi tessellation; there are other interesting possibilities as well such as Kohonen maps, and perhaps deep connectionist learning neural systems.

Properties can be combined into concepts by the same mechanism as properties are formed from domains. Concepts, unlike properties, are built by combining sensors from multiple domains, properties and other concepts (see figure 2).



The model allows the construction of new domains. These are derived from the domains directly related to the senses. There are several methods used. Domains represent abstract

constructs similar to symbolic representations in AI and cognitive modeling and thus serve as the top-down anchor in this framework. Higher order domains can be constructed from a set of domains. A simple example is the longerThan relation constructed as a partition of the space constructed by two length domains. Clearly following this can produce considerably higher dimension spaces. For example, a space for representing faces or persons will have a large number of dimensions.

Another approach is to produce a new partition on an existing domain. That is, to take a subspace of the domain and to partition that domain into properties. The example used is skin color. White and red have different meanings when applied to skin color than when applied to some other context such as a ball. Here, similarity is used to construct a new subdomain.

Gärdenfors uses the example of Newtonian mass as an abstract domain; while it is similar to weight, it is not something directly measured by the sensory system. This shows an approach to systems being grounded in the world by sensory systems, yet also being able to construct abstract models by analogy to physical sensory systems. There is no direct sensory input into Newtonian mass, yet it can still be represented as a domain.

An interesting and useful aspect of this approach is the structured sub-conceptual level. This captures the fluidness of human concepts and presents an approach that can be used to support analogy and to guide induction [7]. The fluid aspects of concepts can be achieved in several ways. Sensitivity to context can be used to downgrade the importance of some of the sub-conceptual properties and sub-concepts that support a given concept. This can be used to select a new concept or to look at the comparison of two concepts in the current environment (defined by the sensory input and goals). This is the key to binding symbolic concepts to the world by a connectionist sub-symbolic structure of meaning.

The approach also enables a bi-directional processing technique – concept to properties and properties to concepts. This is largely the function of the critic. The processing arrows are omitted from the diagrams for simplicity. On initially encountering an input from the sensory system, the flow is from inputs to properties to concepts. However, analysis can also work back from concepts to properties and can do sensitivity analysis by looking at the strength of property and concept selectors. This provides a very meaningful fuzzy membership for properties and concepts.

The input to the system will come through a collection of sensors. The sensors are specific to the problem domain. For example in the reverse engineering domain, the sensors will prehend the aspects of code that a human must build from lower level primitives. This will short cut the construction of concepts and properties that take place in a human perception and cognition during reverse engineering. Care must be taken in the construction of the sensor system. Raising the sensor level too high, on a scale of abstraction, will reduce the ability to make distinctions that will be needed to form a flexible system. Biological inspiration is taken from the evolution of

the human sensory and cognitive system as tuned to survival on the savannas, and later developed the ability to solve more conceptual problems.

3.2 Sparse Distributed Memory

The next layer is a memory built on the sparse distributed memory (SDM) design from Pentti Kanerva [9]. This is extended with the work found in [10]. The Learning Intelligent Distributed Agent (LIDA) system has made extensive use of SDM for a variety of memory types [11]. The memory in this model needs to fulfill three functions: situation recognition, sequence storage and action. Only situation recognition is considered in this paper. Situation recognition follows Kanerva's original scheme of applying a large address space so that the memory easily extracts similar situations. The memory stores the address in the word being stored and that provides the auto associative memory.

The use of the SDM to detect similarity in situations depends on the relationship between the access radius and the locations that are similar to the input being considered. The details can be found in the reference [9]. If the concept is stored in the memory as a single bit, then the access radius will capture locations with enough common concepts. However, a more interesting approach is to store situations with a strength value for the concept. The issue is then to maximize the use of the access radius. The way to do this is to Huffman encode the strength so the distance in coding units and the strength indication is aligned.

While the SDM is very good at, associatively, selecting the best match, in this application the additional step needed is to consider alternatives based on marginal, weakly selected concepts.

3.3 The critic function – abduction's entry

The overarching project has been to look at abduction and sensemaking problem solving. This is discussed in [1]. The present level of development of the critic is still weak in abductive ability [12][13], but it is a step toward stronger abduction and also a step toward expert sense making. The sense making problem is often a straight forward recognition of a situation that has been encountered before. The challenge is recognizing when the situation is almost the one seen before, but misses some critical aspect.

While the system is designed to recognize situations and direct actions by setting goals, the critic has an oversight function with the ability to interfere. It has a focus on surprise and abduction. This is implemented in two modes – by exploration of alternatives and by reflection during backtracking on failure. As a practical note, there are two caveats on the current state of the critic. The description provided here is our current view and is still under development to strengthen the critic. Also, as we apply this system to different problems, the role of the critic can be restricted. Some domains demand a critic with restricted actions in order to assure the system behaves in a controlled

fashion. In this case the critic can log its observations for latter study. This mode is useful in new domains.

Surprise can occur in two ways. There is the obvious mode when a goal, selected by the system, fails to be obtained. Such a failure triggers backtracking. The other is the preemptive detection of “not quite right” situations, which is covered first. The SDM is used to find situations which match the current set of concepts and properties encountered in the environment. This is done by associative matching in the SDM. In order to consider detection of “not quite situations,” the agent must consider how these situations can arise. There appear to be two main cases that are tractable. The first examines the situational assessment for marginal values. The strength of Gärdenfors' system is its geometrical formulation, using relative values, not absolute numerical values. Thus in classifying properties and concepts, we can use the geometrical notions to mark properties that are close to classifying as a different property. This marginal boundary analysis can be used to look at other possible goals in contrast to the initial situation. Since this can be a costly activity, it is triggered by an examination of the “strength of evidence.” This is done by considering first the marginal properties and concepts in the situation and the concepts or properties that are in the found situation, but not in the input (missing evidence). If nothing appears to be marginal, the analysis is not triggered. Secondly, during the actual situation, the critic analyzes the concepts and properties expressed by the input to the sensors. Here, the system looks at concepts and properties not included in the situation produced from the SDM. Alternatives are generated by suppressing some of the concepts in the found situation and probing the memory for other possible situations. This mode of analysis is inspired by process philosophy's explication of consciousness as amplified by inclusion of counterfactuals (D.R. Griffin, 2009).

On failure of a goal, the system backtracks and performs the same kinds of analysis describe above. However, the failed goal provides additional information, that of falsifying that choice of the goal as a counterfactual representation. This forces reconsideration of that goal and to a lesser degree the prior goals that supported the choice. The “blame” for the bad choice is passed back and is applied with a reduced degree of weight as the backtracking proceeds backward.

Now turning to abduction, Hoffmann uses two dimensions to construct a 3 by 5 matrix of abductive types. The simpler forms of abduction are implemented implicitly by the matching in the SDM. Since we are looking for the closest match to a situation, evidence for the situation that is missing will be assumed. The more interesting case is how to apply shifts in the structure of a representation. This is made possible by a meta-information level over Gärdenfors' system. By producing a meta-level over the domains, one can rank similarity measures between domains inductively. This similarity can then be used to shift concepts. Analogical reasoning can be used to modify a situation by substituting concept A for a concept B, but only when their underlying domains have relative similarity.

4 Conclusions

Our work is an intriguing approach to situation recognition. The current state of the system does not yet apply learning to the problem space. This is clearly the next step. Gärdenfors' work does address learning and this is an additional function to add to the critic. In addition the application of abduction in this work falls in the lower reaches of Hoffmann's taxonomy. Improvement will require extension into three reaches. First of all, the use of tool manipulation [Magani 2009] is heavily involved in the more advanced abductive processes. This allows us to bring in logic, mathematics and external tools like software systems. Second, the similarity and metaphor provided by conceptual geometry can be used for theoretic shifts. Third, elicited expert intuition can bootstrap microgenetic development of the concept geometry.

5 Acknowledgement

This work was approved for public release per PA Number 88ABW-2013-0867. The authors gratefully acknowledge the support of Drs. Kevin Gluck and Scott Douglas of the Air Force Research Laboratory's Robust Decision Making Strategic Technology Team.

6 References

- [1] Ronald Hartung, Kirk Weigand "Abduction's Role in Reverse Engineering Software", NAECON, 2012
- [2] J.W. Brown, "Microgenetics theory: Reflections and Prospects," *Neuropsychoanalysis*, 3: 61-74, 2001
- [3] J.W. Brown, "What is Consciousness?" *Process Studies*, 41:1 21-41, Spring/Summer, 2012
- [4] M. Pachalska & B. D. MacQueen, "The Microgenetic Revolution in Contemporary Neuropsychology and Neurolinguistics," in *Process Approaches to Consciousness in Psychology, Neuroscience and Philosophy*, Editors, M. Weber & A. Weekes, SUNY Press, New York, 2009
- [5] Alison Gopnik, <http://thesciencenetwork.org/programs/cogsci-2010/alison-gopnik>, 2012
- [6] Jeff Elman, *CogSci* <http://thesciencenetwork.org/programs/cogsci-2010/10-years-of-rumelhart-prizes-a-symposium>, 2010
- [7] Peter Gärdenfors, "Conceptual Spaces, the geometry of thought", A Bradford book, MIT Press, Cambridge Mass, 2000
- [8] Colin Ware, "Information Visualization", Elsevier, p. 121, ISBN 9780123814654, 2012
- [9] Pentti Kanerva, "Sparse Distributed Memory", A Bradford Book, MIT Press, Cambridge Mass, 1998
- [10] Javier Snider, Stan Franklin, "Extended Sparse Distributed Memory and Sequence Storage" *Cognitive Computation*, 4(2), 172-180. doi: 10.1007/s12559-012-9125-8, 2012
- [11] U. Faghihi, & Stan Franklin, "The LIDA Model as a Foundational Architecture for AGI" In P. Wang & B. Goertzel (Eds.), *Theoretical Foundations of Artificial General Intelligence* (pp. 105-123). Paris: Atlantis Press, 2012
- [12] Hoffmann, Michael, H. G. "Theoric Transformations" and a New Classification of Abductive Inferences, *Transactions of the Peirce Society*, Vol 46 No. 4 pg 570 – 590 2011
- [13] L. Magnani, "Abductive Cognition The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning" Springer Verlag ISBN 978-3-642-03630-9, 2009
- [14] D.R. Griffin, "Consciousness as Subjective Form: Whitehead's Nonreductionist Naturalism" in *Process Approaches to Consciousness in Psychology, Neuroscience and Philosophy*, Editors, M. Weber & A. Weekes, SUNY Press, New York, 2009.

**Appendix F. RDM STT INITIATIVE: ROBUST DECISION MAKING
FOR IMPROVED MISSION ASSURANCE**

**Sub-Project: Context Switching Methods to Calibrate Trust for
Mission Assurance**

– Trust in Automation (Experiment 3)

Dr. Charlene Stokes, RHXS

Project Team

Gov: Dr. Charlene Stokes & Mr. Alex Nelson, RHXS

Students: Ms. Corinne Wright & Mr. Patrick Clark, RHXS

Contract Support: Dr. Beth Veinott, Mr. Corey Fallon, Dr. Beth Crandall, Ms. Anna Grome,
Applied Research Associates, Inc.

1.0 ABSTRACT.

In support of the sub-project, Context Switching Methods to Calibrate Trust for Mission Assurance, the objective of this research project was to identify and validate a relevant trustworthiness dimension set for human-machine cyber-physical decision systems. Following a phased approach, a new trustworthiness model was developed based on extensive ISR analyst interviews. Subsequent phases of experiment validation and integration with “knowledge glyphs” was not accomplished due to cancellation of the project. However, the trustworthiness model and dimensions described herein significantly advance our understanding of trust in these complex decision systems and provide a critical first step to improve trust calibration and robust decision making in these environments.

2.0 INTRODUCTION.

Existing trustworthiness models are narrowly defined, focusing on human-human or human-single system interactions, and they do not account for the increasingly networked and complex environments of today’s decision systems. In addition to the complex structure of these decision systems, Air Force operations are required to engage in the decision cycles under uncertain, dynamic and time constrained environments. In order to improve the quality and robustness of decision processes and outcomes, an accurate understanding of how operators evaluate trust in the context of the larger network they now operate in must be developed. This research project provides an initial step in developing that knowledge, the requisite trustworthiness dimensions to evaluate it empirically, and the potential to integrate those dimensions in appropriate communication and portrayal technologies to further optimize decision robustness.

3.0 AIR FORCE BENEFIT.

As described in the full RDM STT proposal:

Decision making pervades every stage of Air Force and joint missions. These decisions are increasingly complex, cognitively demanding, and consequential as a result of high uncertainty, urgency, and the rapidly changing nature of the joint fight. Massive amounts of relevant information are now available from disparate and powerful biological and artificial sensory systems to inform these decisions; however, the task of quickly extracting and understanding the relevant actionable knowledge from this overwhelming flow of information is daunting, especially given the severe time pressure for making decisions, the dynamic nature of the battlefield environment, and the complexity of dealing with distributed or net-centric team decision tasks. To complicate matters further, it is increasingly clear that the traditional boundaries between human and machine roles are disappearing. The future vision of integrated human-machine decision systems is already upon us. Hence, there is escalating pressure on AFRL researchers to better understand the basic science of mixed human – machine decision making, *and* make use of this science to develop increasingly automated knowledge-extraction tools and intelligent machine-based decision aids that help optimize, speed up, and adaptively adjust inference, prediction, and decision processes in order to adequately inform human decision makers in the loop. This need is conveyed in Air Force requirements documents and the recent reports of scientific groups who have examined potential technical solutions. Such reports include the 2004 report

of the Air Force Scientific Advisory Board (SAB), *Human-System Integration in Air Force Weapon Systems Development and Acquisition*; portions of the 2006 report of the National Research Council, *Basic Research in Information Science and Technology for Air Force Needs*; and even more recently, the 2008 Air Force SAB study on *Defending and Operating in a Contested Cyber Domain*. In their 2008 study, the SAB called for an emphasis on mission assurance, and recommended the development of agent-based models for modeling and simulation (M&S)-based training, as well as a new emphasis on the fundamentals of human-computer interaction (HCI) when dealing with compromised, or perceived-to-be-compromised, cyber systems (pp. 60-62).

The present research directly benefits the Air Force by addressing the gap in the traditional fundamentals of HCI in the critical area of trust in and between systems. The Air Force can no longer afford to rely on traditional models of trust that simply do not apply to the complex, networked environment Air Force operations face today. Advanced technology is already implemented and new technology is being developed at an alarming rate when compared to the lack in advancement of our understanding regarding how humans interact with and in these environments. Without this commensurate understanding, devastating accidents in the Air Force and other services can, and do, occur. Considering the devastating consequences when such accidents occur in the Air Force, the present research explicitly sought out an Air Force sample (i.e., ISR analysts) to ensure direct Air Force relevance of the model developed.

As an example, Open-Source Intelligence (OSINT) analysts currently draw from a vast amount of open-source data and networked sources in the “Collection” phase of PCPAD. In moving forward through the PCPAD process, the usefulness of much of the collected information is based on the “human” gauged trustworthiness of the information and its source (e.g., is it accurate, useable information or planted/misdirection information). With an understanding of the trustworthiness dimensions being used for decision making in these environments, systemic trust-based vulnerabilities can be identified in order to prevent (e.g., through training, selection, and design) their negative impact in the PCPAD process, as well as exploit the associated vulnerabilities in the adversary. That is, ISR analysts could be trained to identify trust-based vulnerabilities and targets, and incorporate that information in the Analysis and Production phase of PCPAD so it can be converted into actionable intelligence for Dissemination.

OBJECTIVES.

Three objectives were sought in the present research:

1. Develop a new trustworthiness model based on relevant dimensions in human-machine cyber-physical decision systems - accomplished.
2. Validate the trustworthiness model through an empirical study using existing experimental platforms (e.g., Convoy Leader) and/or other relevant platforms (e.g., SITA) – not accomplished.
3. Incorporate the validated trustworthiness dimensions in communication and portrayal technologies (i.e., knowledge glyphs). In other words, integrating the ‘what’ to display with the ‘how’ to display it – not accomplished.

The remaining report describes the first objective, as the other objectives were not accomplished due to cancellation of the project.

5.0 PROJECT DETAILS FOR OBJECTIVE 1.

In collaboration with Applied Research Associates (ARA), extensive ISR analyst interviews, archival data analysis, and observation/focus groups were conducted to identify relevant trustworthiness dimensions in analyst operations. Following thematic analysis and comparison to traditional trustworthiness models, a comprehensive dimension set was developed – see Table 1. This new trustworthiness model went well beyond traditional models (e.g., Mayer et al., 1995; ability, benevolence, integrity) and captured relevant dimensions in networked environments derived directly from Air Force analysts - See Appendix A for Final Report from ARA. Initial questionnaire item development was also started, but not finished, as it was to be included in the experimental phase of the project.

Table 1. Trustworthiness Dimensions

Dimension	Definition and Example
Source Agenda	Source’s motives, reasons, incentives for providing information/data. These could be the source’s funding sources or political affiliation. Example: “MIT is more reputable than ‘Global Business.’ ‘Global Business’ might have investors in mind.”
Accessibility of Methods	Do you recognize, understand the methods that were used? Ability to see process/components for generating information and evaluate them for oneself; transparency. Is the scientific methodology for generating the data described in the document? Example: “If people cannot describe their premises and processes, then I don’t trust them (same for tools).”
Data Aggregation	Degree to which data is already processed or prefiltered vs. raw. Example: “Three to four people edit the data before it gets to the analyst and it can be compromised at any point. Need to look at how much it is compromised and how it is compromised. There can be compromises in the initial language.”
Presentation	Data packaging and presentation is appropriate. If it is a written document does it have typos? Are there charts, graphs and images? Example: “Overly flashy article is probably covering something up.”
Logic of Argument	The extent to which a prediction is plausible based on logic. Example: “If ‘A’ happens, this is what we would expect to see.”
Credibility	Source/data’s pedigree, reputation, level of education and/or training. What is it? Does their degree fit with what they are writing about (e.g., are the geologist writing about politics (less credible) or geology (highly credible)). Example: “One that looks good looks more like a scholarly journal. It tips you off because it is more likely to be peer reviewed.”
Reliability	Replication across time and/or people (findings, opinions). Example: Data from this year is consistent with the same data collected last year.
Convergent Validity	Extent to which the information is corroborated by other sources. Example: “When reporting matches up to other reporting, then confidence goes up try to line up multiple reporting streams. If reporting is more consistent, then the analyst will look at it more”
Data Recency	The time/date when the data originated. It was implied in the interviews that more recent data is better than older data. Example: “Latest findings”
Traceability	Extent to which analysis can be traced back to the raw data.

	Example: “I tell analysts to list top premises and the evidence for your premises and often analysts can’t trace it back. Once you present your propositions you need to be able to defend it. Trustworthiness through extraction of evidence is extremely important.”
Supervision	Extent to which the data/source is monitored by a trusted entity Example: “Deception and misinformation have a big impact on trustworthiness” [of a system].
Plausibility	Extent to which the information from the data/source is believable, probable and/or conceivable. Example: “Immediately check: Is it physically possible? Can it be done? If not, what makes the customer think it can be done?”

The widely accepted trustworthiness dimensions of traditional models, both interpersonal and automation (Mayer et al., 1995: ability, benevolence, integrity; Muir, 1994: dependability, predictability, faith) fail to capture the complexity users of networked systems, such as ISR analysts, face. Analysts not only have to consider a single other person or stand-alone system they are interacting with when making trust judgments and associated decisions in these environments, they must consider the vast array of system, source, and user connections the information/data they are viewing represents. The dimensions described in Table 1 highlight the far more complex nature of trust judgments in today’s networked operational environments.

CONCLUSION.

Although all objectives of the present research were not able to be achieved, the trustworthiness dimensions identified advance our understanding of the trust process in complex human-machine cyber-physical decision systems. Although awareness and appreciation of trust calibration is growing, there is a limited understanding of the significant impact trust can have on decision making, particularly across the networked ISR domain; this is a human-centric problem. Continued research, such as the present, is needed to mitigate trust-based vulnerabilities on the part of the ISR analyst and exploit trust-based vulnerabilities on the part of the actor/adversary. Although the ISR domain is one example, a deeper understanding of the trust process in these complex environments will provide an avenue for increased robustness in decision making and mission assurance across the spectrum of warfare.

APPENDIX A



**Extending Human and Machine Models of Trust based on
Analysts Operational Environments**

Subcontract No.:
RQ000746

Prime Contract No.:
FA8650-09-D-6939

Prepared by:
Elizabeth Veinott, Ph.D.
Corey Fallon
Beth Crandall
Anna Grome

Applied Research Associates
Cognitive Solutions Division
1750 Commerce Center Blvd. North
Fairborn, OH 45324-6333
(937) 873-8166

Prepared for:
David Snyder
Systems Research and Applications Corporation
4300 Fair Lakes Court
Fairfax, VA 22033
(P) 703-653-5310
(F) 703-502-1130
David_Snyder@sra.com

Date Submitted:
June 31, 2012

EXECUTIVE SUMMARY

Trust and trustworthiness are multidimensional concepts that are complex. Current theories of trust potentially miss operationally relevant insights. The current effort attempts to bridge this gap by reviewing conducting interviews with novice and expert analysts regarding trust and trustworthiness indicators in human-machine, team, and human mediated environments. From a series of three interviews with experts, observations during a single training exercise with 19 participants and two focus groups we identified 17 trust indicators that operators use. Using those indicators, we developed more than 100 potential self report trust items that could provide a new measure of trust in these operational environments. We examined the effectiveness of using trust as a sensemaking process.

TABLE OF CONTENTS

INTRODUCTION 1
MODELS OF TRUST 1
TRUST AS A SENSEMAKING MODEL 4
METHODS 5
Results Trust Dimensions..... 6
QUESTIONNAIRES ITEMS BASED ON TRUST DIMENSIONS 8
FINITE MIXTURE MODEL OF QUESTIONNAIRE DATA 8
CONCLUSION..... 10
REFERENCES..... 12
APPENDIX A: TRUST QUESTIONNAIRE ITEMS A-1

LIST OF FIGURES

Figure 1. Mayer, Davis and Schoorman (1995) model of interpersonal trust 2
Figure 2. Lee and See’s (2004) Conceptual Model of Human Machine Trust..... 3
Figure 3. The Data/Frame Model of Sensemaking..... 4
Figure 4. Framework of Trust as a Sensemaking Process in
Human-Machine Environments..... 5
Figure 5. Depiction of the six personality groups obtained in the present analysis..... 9

LIST OF TABLES

Table 1. Trustworthiness Dimensions 2
Table 2. Interview Method Developed..... 6
Table 3. Trust Dimensions 7
Table 4. Simple description of each obtained personality group 9

INTRODUCTION

Supporting analyst and team performance in networked-enabled environments includes improving trust calibration in teams where members are humans, automation, or machines. Our approach to understanding trust and trustworthiness is to view it as an active process in which theories of sensemaking are particularly relevant (Klein, Phillips, Rall, & Peluso, 2007). This approach has been used previously in experimental work (Lyons & Stokes, 2012). In order to improve this type of team performance, one needs a better understanding of how people calibrate their trust in these networked environments.

Trust and trustworthiness are complex concepts and there are many different theories of trust in the existing management, human factors and social literature (e.g., Handy, 1995; Lee & See, 2004; Muir, 1994). While there are several current theories of trust, they tend to include relatively few dimensions of trust and trustworthiness. For our research effort, we developed some preliminary concepts of trustworthiness based on interviews with analysts in operational environments.

Our objective was to interview different kinds of analysts to capture the range of trust examples in human-human and human-mediated environments. Then we used these examples to uncover trustworthiness themes and develop an initial conceptual model of trust. These concepts expand current theories of trust and trustworthiness and provide additional insight into trust in operational environments that can inform experimentation and new measures development. This report is an interim end of first year report and covers the interviews and questionnaire items developed for this effort.

In the first part of this report, we review several models of trust in an effort to understand where current models of trust stand, and then we describe our data collection efforts and findings based on our cognitive task interviews, observations, and focus groups at a training event. Following these data, a set of themes and an initial conceptual model of trust were developed. Based on these themes, initial questionnaire items were developed.

MODELS OF TRUST

We define trust as a willingness to accept vulnerability and to put one's self at potential risk (Zand, 1972). Rotter (1980) developed a scale to measure dispositional trust, while other researchers have developed several descriptive models of trust in a variety of disciplines (e.g., Handy, 1995; Lee & See, 2004; Muir, 1994). Some models, such as Muir's human-machine trust model, are descriptive and attempt identify key factors that affect trust in human-machine (Muir, 1987) or human-human contexts (Handy, 1995; Zand, 1972). Table 1 below highlights several different dimensions of trust that have emerged in the literature. Each theory tends to focus on two to three different dimensions.

Table 1. Trustworthiness Dimensions

	Mayer et al. (1995)	Remple (1985), Muir (1994)	Handy (1995)
Ability	X		
Interactivity			X
Integrity	X		
Predictability		X	
Dependability		X	
Affective			
Benevolence	X		
Social Presence			X
Faith		X	

Mayer, Davis, and Schoorman (1995) theory of interpersonal trust include three antecedents: ability, benevolence, and integrity. Ability includes the skills and knowledge the person has that makes one trust them more, integrity relates to whether that person should be trusted, and benevolence relates to the trustee’s intentions. Two of these components might mirror human-machine models of trust.

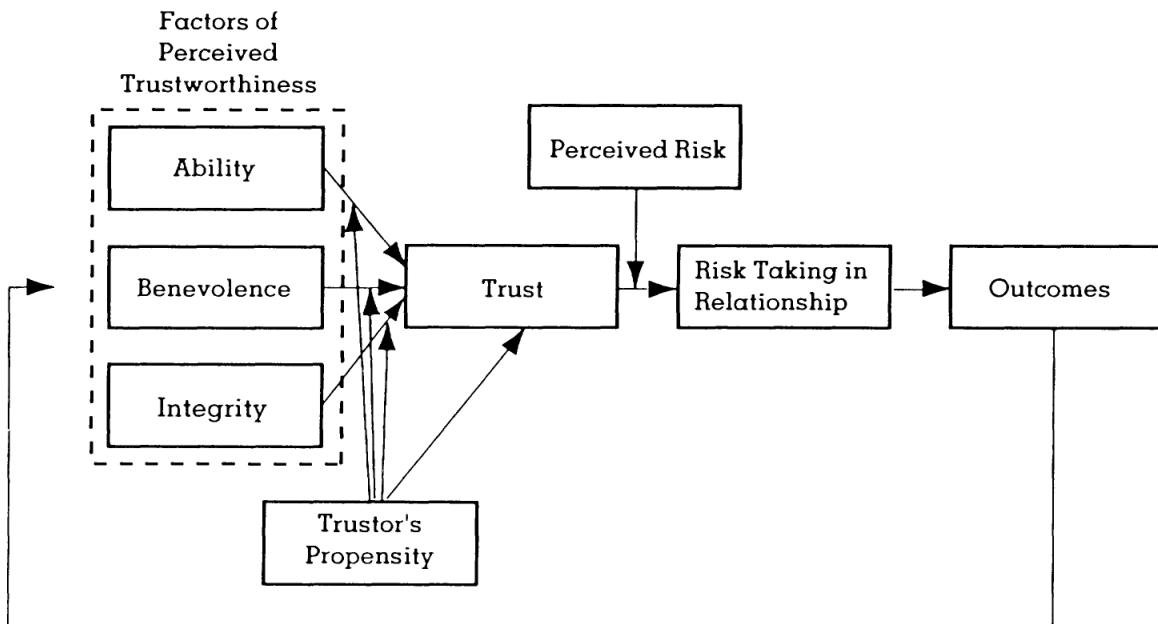


Figure 1. Mayer, Davis and Schoorman (1995) model of interpersonal trust.

Muir suggests that models of social trust such as Rempel, Holmes and Zana (1985) model might

also be applied to trust in human-machine environments. For Muir (1994), trust depends on a system's dependability, faith, and a system's predictability. For Muir (1987), dependability is

reliable or consistent system functioning. Faith is the belief that the system will function in the operator's best interest; predictability is the operator's ability to correctly anticipate the system's behavior. Muir's machine trust model has guided subsequent trust in automation research.

In contrast, Lee and See's (2004) model of human-machine trust focuses on the dynamic interaction between trust and other factors and the effects on a person's behavior and reliance on a system. Their model represents the evolution of human-machine trust and its relationship with other variables in the work environment, such as operator workload, self-confidence and system feedback (Lee & See, 2004, see Figure 1). Lee and See's model depicts a closed-loop process and illustrates the various stages that influence the impact of trust on behavior. According to Lee and See's model, operators' trust evolves based on their predisposition to trust a system and their perception of its trustworthiness. Feedback from the automation is a critical component of this interaction and once trust is compromised, reliance may cease entirely.

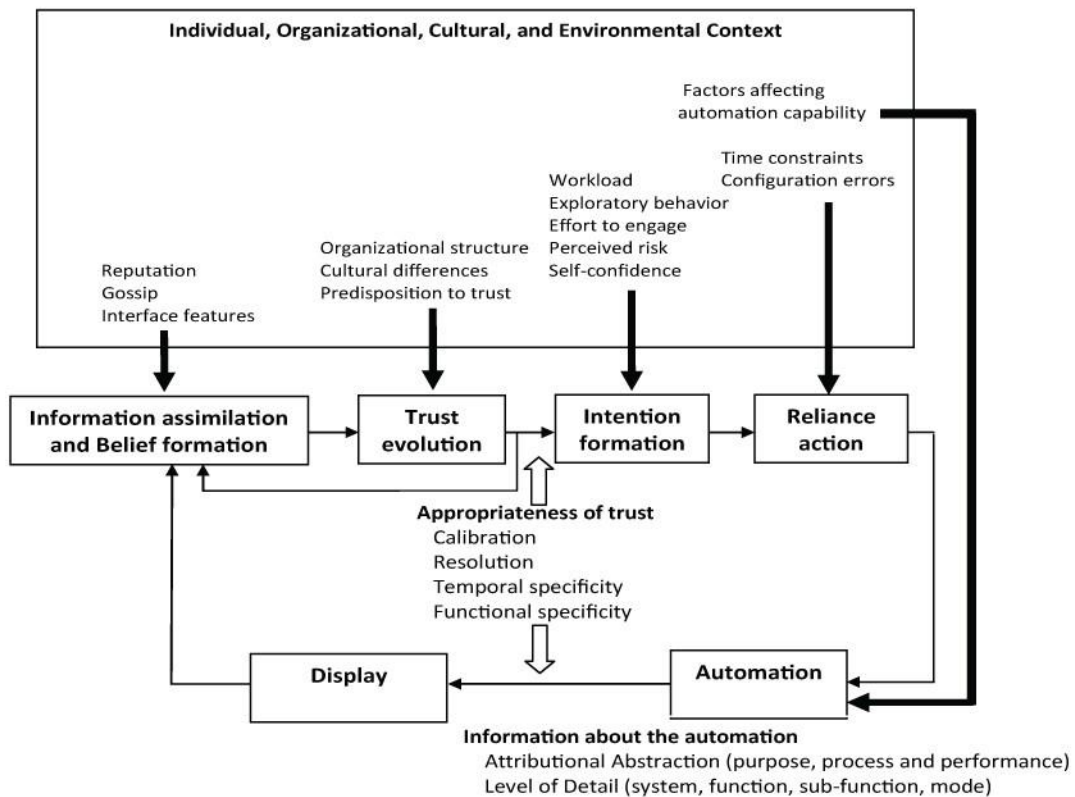


Figure 2. Lee and See's (2004) Conceptual Model of Human Machine Trust.

TRUST AS A SENSEMAKING MODEL

In this paper, we are leveraging Klein et al. (2007) data frame model of Sensemaking (DFM) (see Figure 3) to examine trust. This paper is not the first effort to think about sensemaking concepts in order to understand how analysts do their work. Pirolli, Lee and Card (2004) used sensemaking concepts to explain how intelligence analysts cope with uncertainty when trying to make sense of highly ambiguous data gathered in an uncertain context. However, the DFM model describes one way that humans may develop trust in the complex human-system environment (see Figure 2).

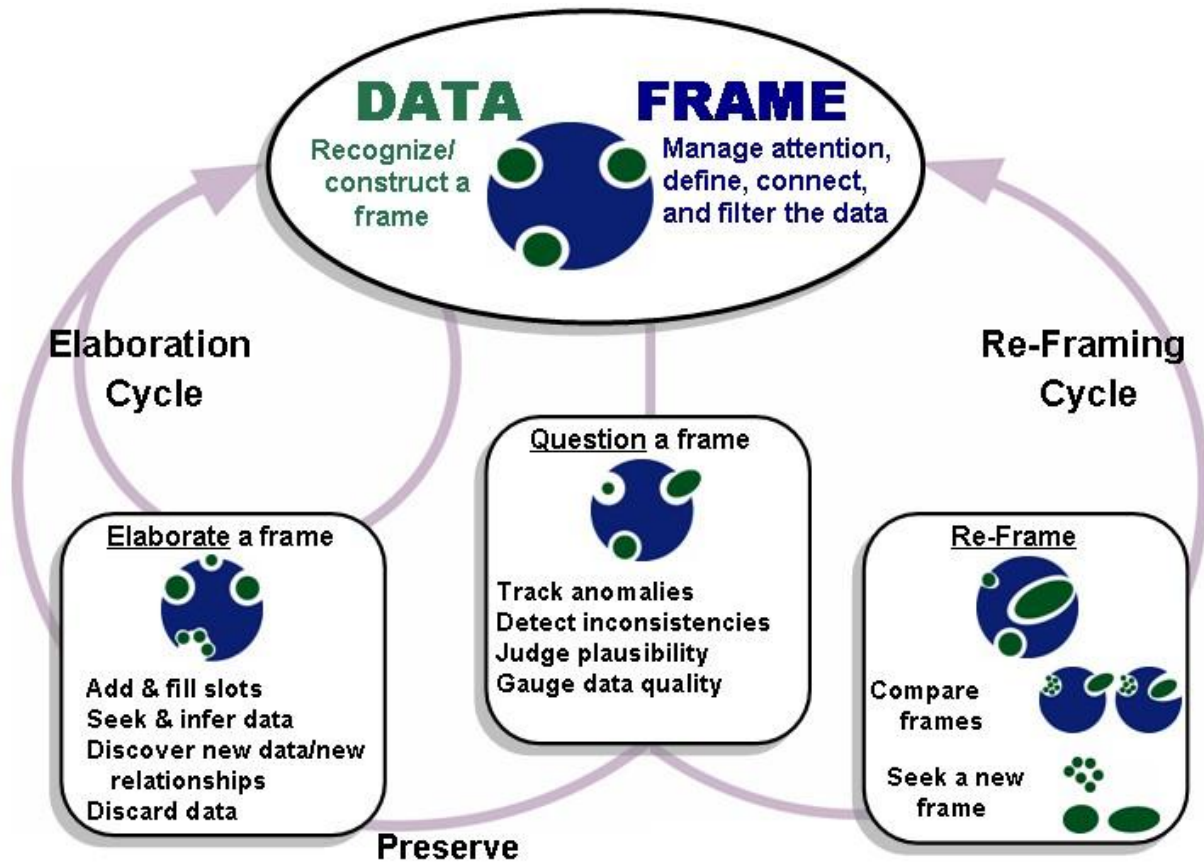


Figure 3. The Data/Frame Model of Sensemaking.

Analysts in operation settings depend heavily on systems, reports from remote teams, and open source information in order to document and diagnose current situations and anticipate future events. Because of this, these operators naturally develop a frame, in the form of an initial explanation of the situation. This frame can guide their actions and understanding of the situations as it evolves. Part of that frame is a set of expectations about the nature of the data, the situation, and how the situation will unfold over time. This can result in the operator's shifting level of trust

in the system or team at different points in their analytic process.

Figure 4 shows the sensemaking model is instantiated in the context of a human-machine-sensor environment in order to show how a networked environment might leverage trust. The key to Figure 4 is that there is an initial trust frame that may be based on several factors (represented in green and blue boxes). For the current research we are seeking to expand on the list of trustworthiness factors that affect “initial trust” or be considered as people recalibrate their trust (by adjusting or maintaining it).

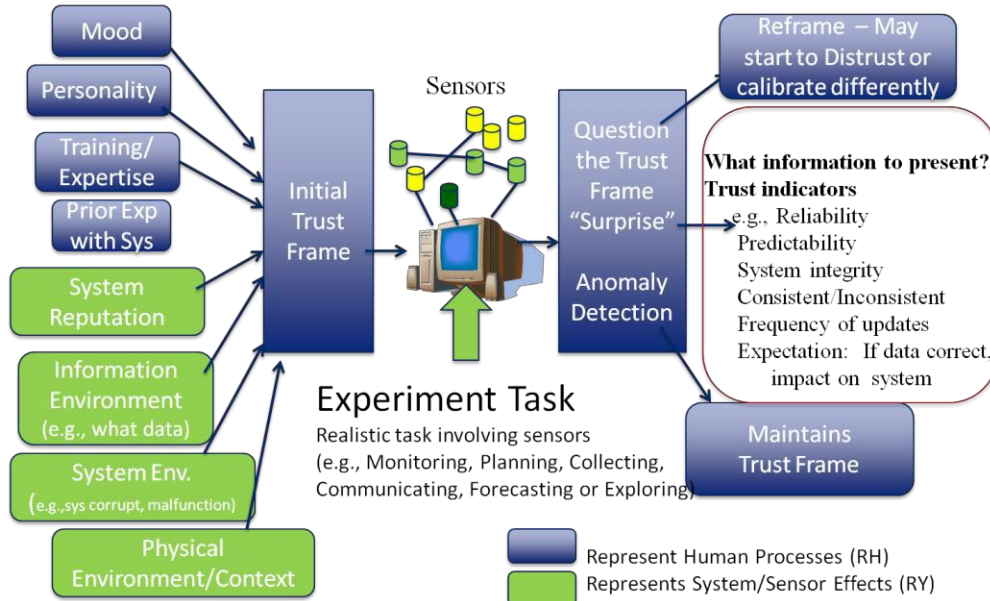


Figure 4. Framework of Trust as a Sensemaking Process in Human-Machine Environments.

Our objective was to interview different kinds of analysts to capture the range of trust examples and trustworthiness indicators in human-human and human mediated environments. Then we used these examples to uncover trustworthiness themes and develop an initial conceptual model of trust.

METHODS

Participants. We had planned to conduct 8-10 interviews. Instead, we conducted three interviews with expert open source analysts. We observed a 6-hour operational training event with 19 participants and conducted two-focus groups with groups of 9 and 10 participants respectively. All were open source analysts.

Interviews: We leveraged several different cognitive task analysis techniques for this effort that are described in Table 2. For expert analysts we used a Critical decision method where our initial query was to ask these analysts, “Can you tell us about an experience where at some point during the analysis process, something seemed “off” and it made you wonder whether you should trust

the information or source? Can you tell us about an experience where at first everything seemed fine until you noticed that something seemed ‘off’ and you knew you couldn’t trust the information or source?” For all analysts we developed task diagrams and conducted knowledge audits.

Table 2. Interview Method Developed

Method	Description
CDM	<ul style="list-style-type: none"> • Great for discovery • Depth--data that is specific & detailed • Works best with experienced operators/analysts to interview • Requires participants talk in detail • Provides sets of cases/examples for use in subsequent research
Knowledge Audit	<ul style="list-style-type: none"> • More structured...provides examples organized/elicited around the same set of dimensions. • More examples in shorter time • More confirmatory rather than exploratory
Task Diagram	<ul style="list-style-type: none"> • Identify the key tasks for position. • Useful for laying out information flow

Archival Analysis: We analyzed an existing interview corpus with analysts for trust themes. These interviews had focused on decision making in analytic contexts, so trust was not the primary focus. Trust was present in many of 17 of the archival interviews.

Observation/Focus groups: In addition, we conducted observations during a training exercise in which the teams were required to capture public opinion on a particular topic (e.g., global warming), or evaluate specific scientific accuracy (e.g., global warming real?)

Results Trust Dimensions

Prior to consideration of trust, the relevance of trust to the person, system, environment, task and/or data must be established. If trust is deemed to be a relevant (and not implicit) concern then the following dimensions of trust are important to consider. Trust could be implicit in which some of these dimensions had been covered.

Table 3. Trust Dimensions

Dimension	Definition and Example
Source Agenda	Source’s motives, reasons, incentives for providing information/data. These could be the source’s funding sources or political affiliation. Example: ”MIT is more reputable than ‘Global Business.’ ‘Global Business’ might have investors in mind.”
Accessibility of Methods	Do you recognize, understand the methods that were used? Ability to see process/components for generating information and evaluate them for oneself; transparency. Is the scientific methodology for generating the data described in the document? Example: “If people cannot describe their premises and processes, then I don’t trust them (same for tools).”
Data Aggregation	Degree to which data is already processed or prefiltered vs. raw. Example: “Three to four people edit the data before it gets to the analyst and it can be compromised at any point. Need to look at how much it is compromised and how it is compromised. There can be compromises in the initial language.”
Presentation	Data packaging and presentation is appropriate. If it is a written document does it have typos? Are there charts, graphs and images? Example: “Overly flashy article is probably covering something up.”
Logic of Argument	The extent to which a prediction is plausible based on logic. Example: “If ‘A’ happens, this is what we would expect to see.”
Credibility	Source/data’s pedigree, reputation, level of education and/or training. What is it? Does their degree fit with what they are writing about (e.g., are the geologist writing about politics (less credible) or geology (highly credible). Example: “One that looks good looks more like a scholarly journal. It tips you off because it is more likely to be peer reviewed.”
Reliability	Replication across time and/or people (findings, opinions). Example: Data from this year is consistent with the same data collected last year.
Convergent Validity	Extent to which the information is corroborated by other sources. Example: “When reporting matches up to other reporting, then confidence goes up try to line up multiple reporting streams. If reporting is more consistent, then the analyst will look at it more”
Data Recency	The time/date when the data originated. It was implied in the interviews that more recent data is better than older data. Example: “Latest findings”
Traceability	Extent to which analysis can be traced back to the raw data. Example: “I tell analysts to list top premises and the evidence for your premises and often analysts can’t trace it back. Once you present your propositions you need to be able to defend it. Trustworthiness through extraction of evidence is extremely important.”
Supervision	Extent to which the data/source is monitored by a trusted entity Example: “Deception and misinformation have a big impact on trustworthiness” [of a system].
Plausibility	Extent to which the information from the data/source is believable, probable and/or conceivable. Example: “Immediately check: Is it physically possible? Can it be done? If not, what makes the customer think it can be done?”

QUESTIONNAIRES ITEMS BASED ON TRUST DIMENSIONS

Based on the dimensions that were listed above, we developed potential self-report rating items from each of these dimensions. The purpose of a questionnaire items is for future scale development that would measure a person's level of trust in a system, person or team. In all about 100 self report questions were developed and the full list is in Appendix A. The items would be used with a 5 or 7-point Likert scale.

The trustworthiness items need to be answered with respect to a particular experiment context, or based on a particular scenario provided to the participant include questions. Respondents are asked to make assessments both of the source of the info (a system, person, or team), but also the information generated by the source (e.g., data, document, conclusions). Source could be a particular system, a person, or a team.

Overall, these items are all generic until we can ground them within a particular scenario (e.g., we need a specific source and specific set of information that participants are responding to. The items can then be tailored to those.)

FINITE MIXTURE MODEL OF QUESTIONNAIRE DATA

As part of the questionnaire development, we explored the use of a finite mixture modeling (Mueller & Veinott, 2008) technique we developed to identify groups of people who responded similarly on a scale. We examined the 50-question IPIP subscale administered to 286 respondents, which involved ten questions each related to five distinct personality constructs (1. Extroversion, 2. Agreeableness, 3. Conscientiousness, 4. Neuroticism, 5. Openness to experience). Of the 286 respondents, in the full study, 271 gave responses to all 50 questions. Questions were answered on a 7-point Likert scale, with half the questions in each dimension positively related to the construct, and half negatively related.

The base likelihood model assumed that responses were generated from a multivariate normal distribution, with mean and variance estimated directly from the data. This model has some limitations when examining Likert-type data, because of edge effects and other non-normality, but in practice it obtains reasonable results. A single-group model would estimate 100 parameters for the data set—50 means and 50 standard deviations.

We performed a finite mixture model on the data (cf. Mueller & Veinott, 2008) to determine whether there were any coherent personality subgroups. The model works by assuming a fixed number of groups, and using the E-M algorithm to determine the most likely membership of those groups. Separate models were fit for 1 through 10 groups, and by comparing models using the Bayesian Information Criterion (BIC), we determined that the simplest most explanatory model had six groups. Best-fitting models that assumed more than six groups all converged to six-group solutions (with one group being empty), indicating that a six-group solution was appropriate.

The makeup of those groups are found on the left panel of Figure 5. In the figure, mean values of the ten questions in each category (with proper reverse-recoding) are shown for each inferred latent personality group. It is apparent that overall, many of the groups are fairly similar, often

differing from one other group by a single personality factor. This variation can be seen most easily in the right panel of Figure 5, where responses are organized by mean personality category.

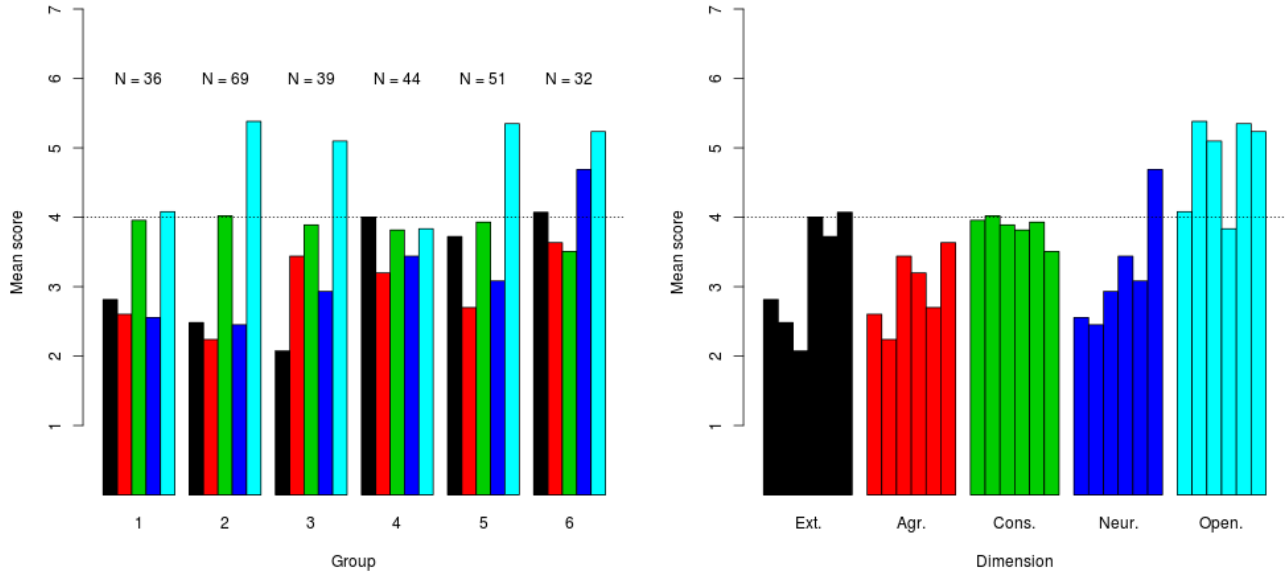


Figure 5. Depiction of the six personality groups obtained in the present analysis.

Left panel shows each distinct group; right panel shows difference of each personality factor across groups.

For example, there was very little variability across respondents on the ‘conscientiousness dimension; all groups had mean scores around 4 (the center of the scale). In contrast, four groups had positively-valenced scores on ‘openness to experience’ whereas two had scores near the center. Similarly, five groups had negatively-valenced scores on ‘neuroticism’ whereas one group had slightly positive scores. Table 4 below summarizes these results, with – and + signs indicating scores that are relatively above or below average.

Table 4. Simple description of each obtained personality group. +/- indicates whether mean score of group is higher or lower than the average score

(not the middle range of 4 on a 1-7 scale)

Group	Extraversion	Agreeableness	Conscientious-ness	Neuroticism	Openness to experience
1 (N=36)	-	-		-	-
2 (N=69)	-	-		-	+
3 (N=39)	-	+		-	+

4 (N=44)	+	+		-	-
----------	---	---	--	---	---

Group	Extraversion	Agreeableness	Conscientious-ness	Neuroticism	Openness to experience
5 (N=51)	+	-		-	+
6 (N=32)	+	+		+	+

These six groups each accounted for between 33 and 56 respondents, and suggest that the personality factors may over specify the actual personality types that exist in the general population. That is, if each factor was allowed to take on either high or low values, there would be $2^5=32$ distinct personality types; our analysis shows that just six subtypes accounts well for the population being tested. Furthermore, very little variance was accounted for by the conscientiousness dimension.

Based on the lack of empirical results in the literature, our initial candidate trust indicators for experimentation are as follows.

- Predictability:** Does the sensor or system behave as expected (consistent/inconsistent)? There are multiple ways to operationalize predictability. One type of predictability is reliability. Reliability typically refers to the accuracy and consistency of the data (e.g., 60% vs. 90%). Past research indicates that somewhere between 60% and 90% reliability is where people switch from trusting a system to not trusting. Other ways to operationalize predictability include that the data match expectations or the sensor output is consistent with the operator's expectation given context (e.g., raining). For example, a system could be a consistently poor performer only under certain conditions (predictable). In these cases, people might be more likely to develop workarounds for those conditions. As mentioned earlier, most experimental research on human-machine trustworthiness varies **reliability** and sometimes other **predictability** measures.
- Ability:** Is the system capable of doing what needs to be done? For example, learning why a system made an error (and why it will not again) recalibrates human trust.
- Supervision:** Is the system monitored by another system or person who is trusted? This could also include a rate of monitoring.
- Vulnerability:** This trust indicator would provide information on access and capability of a sensor to be compromised, which is one of RYWC's central concerns. This is a different operationalization than what the CLE used. It can be operationalized as: number of access points, type of access (secure, unsecure), who built system and software (reputation).
- Integrity:** Does the system perform functions without being degraded by internal or external disruptions (malfunction or malware could reduce integrity)? The vulnerability indicator relates to this indicator.

CONCLUSION

Trust and trustworthiness are complex concepts that have generated different approaches from a variety of academic disciplines. Using a combination of cognitive task analysis techniques that complement each other and provide different data, we identified additional dimensions of

trustworthiness based on operational environments. Overall we had two conclusions: 1) that the

trust process is a multi-stage process consistent with several of the previous theories, and 2) current theories of trust focus on two-three dimensions, but we found an additional nine themes of trustworthiness that emerged from our interviews. Future work in the next phase will focus on developing experiment tests and measures based on these dimensions, testing and validating some of these dimensions using experimental platforms, and further development, refinement, and validation of questionnaire items from these themes.

REFERENCES

- Handy, C. (1995). Trust and the virtual organizations. *Harvard Business Review*, 73(3), 40-50.
- Klein, G., Phillips, J. K., Rall, E., & Peluso, D. A. (2007). 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* (pp. 113-158). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1), 50-80.
- Lyons, J. B., & Stokes, C. K. (2012). Human-human reliance in the context of automation. *Human Factors*, 54, 112-121.
- Mueller, S. T., & Veinott, E. S. (2008). *Cultural Mixture Modeling: Identifying cultural consensus (and Disagreement) using Finite Mixture Modeling*. Paper presented at the 2008 Cognitive Science Society Meeting, Washington, D. C.
- Muir, B. M. (1987). Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies*, 27(5-6), 527-539.
- Muir, B. M. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated system. *Ergonomics*, 37(11), 1905-1922.
- Pirolli, P. L., Lee, T., & Card, S. K. (2004). Leverage points for analyst technology identified through cognitive task analysis. Palo Alto, CA: Palo Alto Research Center.
- Rempel, J. K., Holmes, J. G., & Zanna, M. P. (1985). Trust in close relationships. *Journal of Personality and Social Psychology*, 49(1), 95-112.
- Rotter, J. B. (1980). Interpersonal trust, trustworthiness, and gullibility. *American Psychologist*, 35, 1-7.
- Zand, D. E. (1972). Trust and managerial problem solving. *Administrative Science Quarterly*, 17(2), 229-239.

APPENDIX A: TRUST QUESTIONNAIRE ITEMS

Trust items

Accessibility of Methods - Do you recognize, understand the methods that were used? The ability to see process/components for generating information and evaluate them for oneself; transparency. Is the scientific methodology for generating the data described in the document?

Questionnaire Items:

[Note: “ the source” = system, individual, or team]

- The process used to generate the information is clear to me. (+)
- The methods for collecting and analyzing the data have been well-described. (+)
- The way in which [the source¹] came to its conclusions is clear to me. (+)
- I understand the process by which [the source] developed this information. (+)
- I have access to the methods by which the data were collected and analyzed. (+)
- I have insufficient insight into how [the source] reached its conclusions (-)
- I have inadequate information about how this information was generated. (-)
- The approach used to generate the conclusions is difficult to understand. (-)
- It is easy for me to trace the process used to reach these conclusions. (+)
- I understand the basis for the conclusions that have been drawn. (+)
- I have sufficient information about how [the source] generated its conclusions. (+)
- The methods used to generate this information are unclear to me. (-)
- The way the data were collected and analyzed is transparent . (+)
- I have insufficient information about the process by which [the source] came to its conclusion.(-)

Data Aggregation – Degree to which data is already processed or prefiltered vs raw.
Is the raw data available?

Questionnaire Items:

- The information provided by [the source] is close to its original form. (+)
- The information provided by [the source] has been modified significantly from its original form. (-)
- I believe the integrity of the data has been compromised. (-)
- The data that fed into the conclusions is readily available. (+)
- There is a substantial difference between this information and the raw data. (-)
- The data provided to me has been filtered significantly. (+)
- The data has been modified substantially from its original form. (-)
- The information given to me by [the source] has been filtered very little. (+)
- I believe the data has been edited significantly. (-)
- I can access the data that provided the basis for the conclusions drawn. (+)
- The original data has been filtered substantially. (-)

¹“the source” is a placeholder for a system, a person, or team.

Source Agenda - Source's motives, reasons, incentives for providing information/data. These could be the source's funding sources or political affiliation.

Questionnaire Items:

- I think [the source] has a hidden agenda with this information. (-)
- I believe [the source] has good motives for providing this information. (+)
- [The source] has ulterior motives (e.g., financial, religious, political, or other) in providing the information to me. (-)
- It is clear that [the source] is motivated by a hidden agenda. (-)
- The reasons [the source] has for reporting these data are honorable. (+)
- I believe this information is intended to promote a particular agenda. (-)
- I believe there is an incentive for [the source] in providing this particular information. (-)
- [the source] has honorable intent in providing this information. (+)
- I question [the source's] motivation for providing this information. (-)
- [the source] is using this information to promote a hidden agenda. (-)
- The information reported has been influenced by either financial, religious, political, or other ulterior reasons. (+)
- I do not believe [the source] has a hidden agenda for providing this information. (+)

Appearance– Data Packaging and presentation. If it is a written document does it have typos? Are there charts, graphs, and colorful images? is consistent with expectations.

Questionnaire Items:

- The information's format is consistent with my expectations (+)
- The presentation of this information is well-polished. (+)
- The quality of the graphics in this document is appropriate for this type of source. (+)
- The overall quality of the [document] does not match my expectations for this type of source. (+)
- The images used to communicate the information are appropriate (+).
- The use of color in [this document] is appropriate. (+)
- The tone of the [document] is appropriate for the subject matter. (+)
- The images used to communicate the information are inconsistent with my expectations(-).
- The use of color used to present this information is inconsistent from what I would expect. (-)

Logic of Argument – Predictability based on logic of the argument. Does the argument make sense coming from the source it is coming from? Does it fit with someone of this stature or organizational or political affiliation?

Questionnaire Items:

- The information provided by [the source] presents a logical argument. (+)
- I can easily follow the reasoning of the argument presented. (+)
- The logic of the argument makes sense to me. (+)
- I can anticipate what might happen next based on the logic of the argument. (+)
- The argument [the source] makes is inconsistent with what I know about [the source]. (+)

- The argument provided by [the source] seems irrational. (-)
- It is easy to follow the logic of [the source's] argument. (+)
- [The source's] argument is consistent with what I know about [the source]. (+)

Credibility – Source/data's pedigree, reputation, level of education and/or training. What is it? Does their degree fit with what they are writing about (e.g., are the geologist writing about politics (less credible) or geology (highly credible)).

- The source of this information is credible. (+)
- I have high confidence in the reputation of [the source]. (+)
- I find the information believable based on what I know about [the source's] credentials. (+)
- I have doubts about the credibility of these findings. (-)
- I believe [the source] is well-versed in the topic he/she has written about. (+)
- The information provided by [the source] has a high degree of credibility. (+)
- The [source] lacks credibility. (-)
- [The source's] argument is believable. (+)
- The data provided by [the source] is not credible. (-)
- I am confident in the credibility of [the source].
- I am skeptical about the credibility of these findings (-)

Reliability – Replication across time and/or people (findings, opinions).

- The findings on this topic have been replicated over time. (+)
- The opinions offered on this topic have been fairly consistent over time. (+)
- The information provided by [the source] has been inconsistent over time. (-)
- The views expressed by [the source] have varied over time (-)
- The findings have changed over time. (-)
- There has been a significant shift in opinion on this topic. (-)
- [The source's] argument has been inconsistent over time (-)
- The views on this topic have changed significantly over time. (-)
- The views expressed on this topic have been consistent across sources (+)
- The information provided by [the source] is consistent with information I have seen from other sources. (+)

Validity- Different data sources are saying similar things

- The conclusions drawn by [the source] are similar to those drawn by others. (+)
- This information matches information provided by other sources. (+)
- The findings correspond to those offered by other sources. (+)
- The information provided by [the source] is inconsistent with what I've seen elsewhere. (-)
- The findings have been replicated by multiple sources. (+)
- The information presented here is quite different from other reports I've read. (-)
- The opinions expressed on this topic differ significantly across sources. (-)

- There is little correspondence between this information and the information provided by other sources. (-)
- The findings on this topic have been replicated across different people. (+)

Data Recency – The time/date when the data originated. More recent data is better than older data was what was implied in the interview. “Latest findings”

- The information presented by [the source] is very recent.
- The information provided by [the source] is outdated.
- This information is up-to-date.
- I believe this information is obsolete.
- The data offered by the source originated a long time ago. (-)
- The data provided by [the source] is current.
- The data provided by [the source] appear to be out-of-date.
- I believe this information reflects the latest findings. (+)

Traceability – Extent to which analysis can be traced back to the raw data ---

- I can trace [the source’s] conclusions back to the raw data.
- I can see how [the source] arrived at its/his/her conclusions from the raw data.
- There is a clear link between [the source’s] findings and the raw data.

Supervision – Extent to which the data/source is monitored by a trusted entity

- [The source] is well-monitored. (+)
- I believe there is inadequate supervision of [the source’s] work. (-)

Plausibility – Extent to which the information from the data/source is believable, probable and/or conceivable.

- The information provided by [the source] seems plausible to me. (+)
- The findings offered by [the source] seem probable. (+)
- [The source’s] argument seems implausible. (-)
- I have a hard time believing the information provided by [the source]. (-)
- The information presented by [the source] is not plausible. (-)
- The information described by [the source] is believable. (+)
- I think the conclusions drawn by the source seem reasonable. (+)
- The findings provided by [the source] seem inconceivable to me. (-)

**Appendix G. Remotely Piloted Aircraft (RPA) Testbed for
Research on Robust Asset Management and Decision
Making**

***In Support of the Robust Decision Making (RDM) for Improved
Mission Assurance Strategic Technology Thrust (STT)***

Mark Derriso, AFRL/RQQI
Thierry Pamphile, AFRL/RQQI
Tim Halverson, RHCP
Kevin Gluck, RHAC
Roman Ilin, RYMT
Igor Ternovskiy, RYMH
Jeremy Knopp, RXLP
Michael Grimaila, AFIT/ENV

October 28, 2013

**Distribution authorized to Department of Defense and U.S. DoD contractors only;
Critical Technology; December 2012. Refer other requests for this document to
AFRL/RQQD, Wright-Patterson Air Force Base, OH 45433-7542.**

**WARNING – This document contains technical data whose export is restricted by
the Arms Export Control Act (Title 22, U.S.C., Sec. 2751, et seq.) or the Export
Administration Act of 1979, as amended, Title 50, U.S.C., App. 2401, et seq.
Violations of these export laws are subject to severe criminal penalties. Disseminate
in accordance with the provisions of DoD Directive 5230.25. *(Include this statement
with any reproduced portions.)***

**DESTRUCTION NOTICE – Destroy by any method that will prevent disclosure of
contents or reconstruction of the document.**

See additional restrictions described on inside pages

**AIR FORCE RESEARCH LABORATORY
AEROSPACE SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

Table of Contents

Table of Contents	165
List of Figures	167
List of Tables	168
ACRONYMS	169
1. INTRODUCTION	170
1.1. Purpose	170
1.2. Background	170
1.3. Scope	171
2. Simulation Environment	171
2.1. Simulation Scenario	171
2.2. Simulation Tools	172
3. RQ RPA Virtual Experiment	173
3.1. RPA Virtual Experiment Objective	173
3.2. RPA Virtual Experimental Methods	173
3.2.1. Experimental Task	173
3.2.2. Design of Experiments	173
3.2.3. Participants	176
3.2.4. Apparatus	176
3.3. Results	177
3.3.1. Time Target in View	177
3.4. Discussion	179
4. RH RPA Human Experiment	179
4.1. Objective	180
4.2. Methods	180
4.2.1. Task	180
4.2.2. Design	181
4.2.3. Participants	182
4.2.4. Apparatus	183
4.3. Results	183
4.3.1. In-Task Subjective Workload Rating	183
4.3.2. Heart Rate Variability (HRV)	186
4.4. Discussion	187

5. Future Research	187
5.1. Predictive Displays	187
5.2. Classification of Faults	188
6. References	188

List of Figures

Figure 1. VSCS Control Station	172
Figure 2. Target Vehicle at Starting Compound	174
Figure 3. Target Vehicle in Urban Environment	175
Figure 4. Simulation Architecture	177
Figure 5. Participant Work Station	177
Figure 6. Time the Target was in Sensor Footprint under the Different Fault Conditions	178
Figure 7. Time Target was in Sensor Field-of-View under the Different Indication Conditions	178
Figure 8. Time Target was in the Sensor Field-of-View under Both Independent Variables	179
Figure 9. RPA Operator Display Interface	180
Figure 10. RPA Supervisor Display Interface	181
Figure 11. Mean subjective workload as a function of task load and vehicle fault.	184
Figure 12. Scatter plot of subjective workload measures	185
Figure 13. Mean subjective workload as a function of task load and participant role	186
Figure 14. Mean heart rate variability (HRV) by epoch	187
Figure 15. Spark-Line Type Predictive Display	188

List of Tables

Table 1. Fault Condition Independent Variable	175
Table 2. Dependent Variables	176
Table 3. Datasets recorded for the operator and supervisor participant roles ...	182

ACRONYMS

1.1	711HPW/RH 711 th Human Performance Wing Human Effectiveness Directorate
1.2	AFRL Air Force Research Laboratory
1.3	AFRL/RQ AFRL Aerospace Systems Directorate
1.4	AFRL/RY AFRL Sensors Directorate
1.5	AFSOC Air Force Special Operations Command
1.6	CCA Cooperative Control Algorithm
1.7	DOE Design of Experiments
1.8	EO/IR Electro-Optical/Infrared
1.9	HUMAN Human Universal measurement & Assessment Network
1.10	ISHM Integrated Systems Health Management
1.11	SME Subject Matter Expert
1.12	RDM Robust Decision Making
1.13	RPA Remotely Piloted Aircraft
1.14	RSTA Reconnaissance, Surveillance, and Target Acquisition
1.15	STT Strategic Technology Thrust
1.16	UAS Unmanned Aerial System
1.17	VSCS Vigilant Spirit Control Station
1.18	VSSim Vigilant Spirit Simulation
1.19	WPAFB Wright Patterson Air Force Base

INTRODUCTION

Purpose

The purpose of this report is to present the results of the research conducted in support of the AFRL project entitled "Remotely Piloted Aircraft (RPA) Testbed for Research on Robust Asset Management and Decision Making". This project was part of the overarching Robust Decision Making (RDM) Strategic Technology Thrust (STT) Initiative funded by AFRL headquarters. The objective of this effort was to detect and identify degradations in human and aircraft performances due to increases in task load on RPA operators and systems failures. In addition to presenting experimental results, this report identifies areas for future investigation.

Background

Remotely Piloted Aircraft (RPA) are being employed in an increasing number of missions by all branches of the military. The USAF Unmanned Aerial System (UAS) Flight Plan states that total UAS flight hours more than doubled from 2006 to 2009. Manpower available for command and control of these assets has not increased at the same pace. The report also identifies increased use of system automation and human supervisory control as key assumptions that enable planned increases in future UAS/RPA operations. The growing number of UAS/RPA conducting a single mission carries with it attendant consequences for decision makers ultimately responsible for managing these assets. Limitations in communication bandwidth, task overloading, and reduced situation awareness of UAS/RPA are all likely problems with the potential to substantially degrade and limit operational performance.

The purpose of the AFRL multi-directorate Strategic Technology Thrust (STT) initiative on Robust Decision Making (RDM) was to investigate the benefits of providing enhanced situation awareness (SA) information to RPA operators and mission-level decision makers. To achieve this, the RDM researchers planned to develop a testbed for discovery research on robust asset management and decision making. The primary objective of this effort was to provide quantitative SA information regarding the operator and aircraft conditions to improve tactical and operational decision making for mission assurance. The project intended to exploit previous and on-going work in the areas of Integrated Systems Health Management (ISHM), Cooperative Controls and Cognitive Modeling. Furthermore, the project planned to leverage existing facilities including the Human Universal Measurement & Assessment Network (HUMAN) and ISHM labs.

The RDM project intended to fill a critical gap in the RPA domain which is the lack of actionable, comprehensive system state information available to RPA operators and mission-level decision makers. Currently, there is only basic system state information displayed to the operator and the information that is presented is not integrated to provide an overall aircraft state or projected remaining capability. In addition, there is no operator state information (i.e., cognitive state evaluation) currently displayed to the RPA mission commander. This gap impacts long-term mission planning which requires a significant increase in the number of RPA operational hours without a comparable increase in manpower. It is anticipated that operator and mission commander workload may increase with this increased operational tempo. The RDM project aimed to alleviate some of the

workload with decision aid tools, cooperative control algorithms (CCAs), and system state information.

Scope

The RDM project involved collaboration across multiple AFRL technical directorates including Aerospace Systems Directorate (RQ), Human Effectiveness Directorate (RH), Sensors Directorate (RY), and Materials, Manufacturing Directorate (RX). In addition, the Air Force Institute of Technology (AFIT) Department of Systems Engineering & Management participated as well. The original objective of this project was to develop a testbed to evaluate the integration of aircraft and operator conditions, CCAs and decision aids designed to help the operator and mission commanders make more informed decisions. RH's task was to provide instrumented workstations that could monitor operator readiness using physiological measurements and RQ's task was to estimate aircraft condition using vehicle state data by leveraging the ISHM testbed currently under development. The RH and RQ experimental facilities were to be networked to provide an operator-in-the-loop simulation capability that integrated both operator and vehicle state information. Additionally, RQ was tasked to provide CCAs that would consider operator and vehicle state information before calculating a flight trajectory for an aircraft. Using data acquired from physiological measurements, RY was tasked to develop regression models for RPA operators to predict cognitive stress and fatigue. Finally, RH was to develop decision aids to assist the RPA operators and supervisors to enhance operational decisions in a dynamic environment. The cognitive model developed by RY would allow a significant number of constructive simulation runs to be executed in order to evaluate the decision aids.

The original scope of this effort was reduced due to limited resource availability. CCAs were removed from consideration when the STT Subject Matter Expert (SME) for that area relocated. Development of the decision aids was postponed to a future phase of the STT pending funding and resource availability. The ISHM testbed was not mature enough to leverage for the current effort. Lastly, it was decided that RH and RQ would work independently to evaluate specific areas of the problem space. RH would concentrate on collecting physiological data from operators and RQ would assess the value of displaying vehicle state information to operators for enhanced decision making. In order to integrate these two efforts in the future, a common virtual environment was selected. The virtual environment selected was the Vigilant Spirit Simulation (VSSim) and the Multi-UAV Supervisory Control Interface Technology (MUSCIT) scenario, leveraged from previous work within RH. The Vigilant Spirit Control Station (VSCS) provided the operator interface and the MUSCIT scenario was modified to incorporate the effects of vehicle degradation.

Simulation Environment

Simulation Scenario

The RDM study leveraged the VSCS simulation architecture and the MUSCIT scenario previously developed by RH. The MUSCIT scenario was developed with input from Air Force Special Operations Command (AFSOC) and thus provided a representative operational scenario in which to study RPA operator workload, decision making and overall system performance. The scenario represented a typical AFSOC mission involving a range of Reconnaissance, Surveillance, and Target Acquisition (RSTA) tasks to include point

surveillance, route surveillance, area search and target tracking within an urban environment.

The original MUSCIT scenario was modified to include vehicle degradation (e.g., camera gimbal freeze) in the RQ experiment. For the RH portion of the experiment the scenario was modified to include, experimental control over the level of task load experienced within each vignette in the scenario, and RPA team expansion to include a multi-aircraft monitor-like supervisor (MAM; Stilson, 2008), in addition to two operators. Each operator served as sensor operator and RPA “pilot”, and had control of two RPAs. The supervisor assigned tasks to the operators, in addition to monitoring operator’s performance and vehicle’s condition. The operator tasks included following unfriendly vehicles, escorting friendly vehicles, searching for unfriendly units, and monitoring unfriendly units. Vehicle failures included intermittent communications (i.e., vehicle control) and loss of sensor zoom.

Simulation Tools

Vigilant Spirit Control Station (VSCS) / Vigilant Spirit Simulation (VSSim). VSCS is an advanced ground control station for coordinating and controlling multiple RPAs which can be simulated or live assets. VSCS was used as the simulation architecture framework for the current study. The VSCS interface for the current study is shown in Figure 26. The displays on the right side of the control station can display video for each of the two RPAs under control. VSSim is the overarching simulation and provides simulated vehicles as well as subsystems including various sensor and weapon payloads. VSSim controls the simulation as well as the movement of simulated assets. Simulated assets for the current study included RPAs, and a gimballed electro-optical (EO)/infrared (IR) sensor.

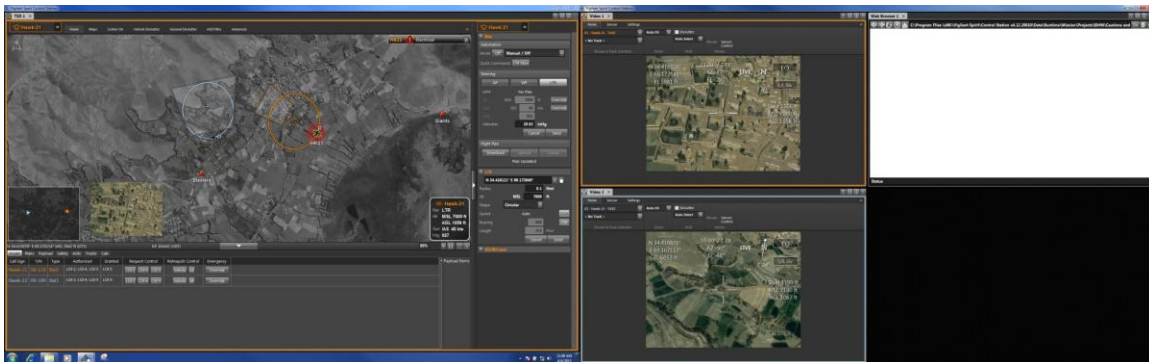


Figure 26. VSCS Control Station

Virtual Real Scene Generator (VRSG™) by MetaVR, Inc. VRSG™ is a real-time computer image generator designed to visualize geographically expansive and detailed worlds on PCs. It was used to generate the simulated EO/IR sensor images displayed to the RPA operator in the AFRL/RQ experiment.

SubrScene simulation visualization toolkit. Subrscene is a Government Off-the-Shelf product compatible with both Linux and Windows operating systems. It is an Open Scene Environment (OSE) application that provides real-time scene rendering for pilot vehicle

applications. It was used to generate the out-the-window scene in the AFRL/RH experiment.

RQ RPA Virtual Experiment

RPA Virtual Experiment Objective

The objective of this experiment was to investigate the impact of a fault warning system on mission performance for a variety of different system faults. For this experiment, the operator served as both pilot and sensor operator. Since this was the initial look at this problem space, emphasis was placed on obtaining a top level understanding of the effects of different fault types and their associated warnings. One of the main impacts of taking this approach is that the amount of time spent on each treatment condition was reduced. Instead of running a complete mission from start to end, we ran shorter mission vignettes that were a subset of a mission. The main drawback to this approach is that no concrete data is collected about how the various treatment conditions impact the mission. The advantage of this approach is that the experimenters are able to explore a larger set of treatment conditions. Given that this was the initial study in this project, this broader approach was considered more appropriate.

RPA Virtual Experimental Methods

Experimental Task

The participants were required to maintain line of sight of a truck. The trial would start with the truck parked near a compound (Figure 27), and at approximately 1.5 minutes into the trial the truck would begin to move into a more urban environment (Figure 28). The participants were responsible for slewing and zooming the sensor and moving the vehicle loiter point as needed. The target vehicle never stopped moving throughout the trial. There were distractor vehicles within the virtual environment, but the target vehicle was the only red vehicle in the scene. The target path went through generic Middle Eastern rural and urban terrain. There were 6 unique paths that the target vehicles could take. There was a second RPA available to the operators if they felt employing it was the best method to mitigate a fault (i.e., they could essentially replace the degraded RPA with the second RPA).

Design of Experiments

The experiment was a 4x2 full factorial design with a repeated baseline condition. From pilot experimentation the experimenters did not believe there would be enough knowledge transfer between treatment conditions to justify a between subject design. Also, a within subject design provides more statistical power when compared to a between subject design for the same number of participants. The full factorial design was used to help mitigate individual differences and to help balance the run matrix to compensate for training effects.

Before the data collection began all the participants were trained on all the different fault conditions and fault mitigation strategies. The participants could either try to compensate for the different faults by modifying their behavior or bring in another RPA. An example of behavior modification for the data link fault would be to delay sending inputs to the aircraft. If a behavior modification didn't work the participants could move the other RPA in the area to complete the mission. Each participant ran through 12 trials. The trials were as follows: 6 trials with warnings, and 6 trials without warnings. Of the 6 trials that had warnings, 4 had faults 2 did not. The participants were told before the trial started that they had warnings. For the 6 trials that did not have warnings, 4 had faults 2 did not. The

participants were told before the trial started that they did not have warnings. Each participant was exposed to each of the 4 unique fault conditions twice (once with warnings and once without). Once a fault occurred it did not clear for the rest of the trial. Only one fault occurred per trial. Each trial was 6-8 minutes long. Breaks were provided as necessary.

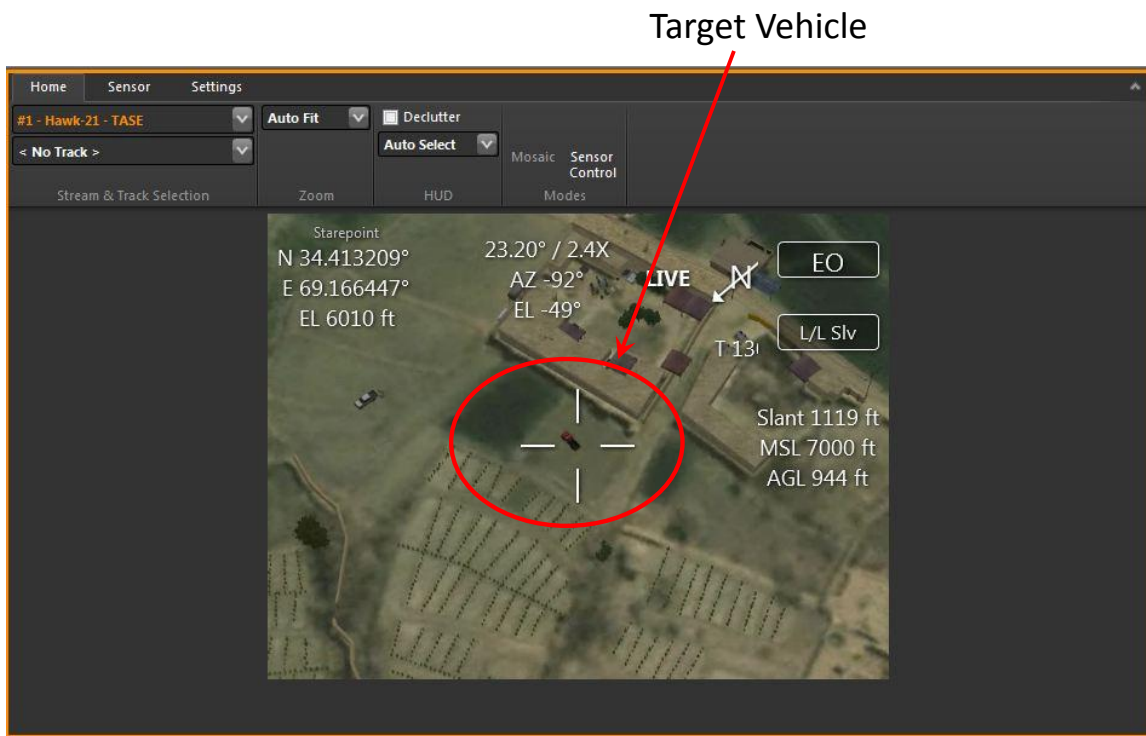


Figure 27. Target Vehicle at Starting Compound

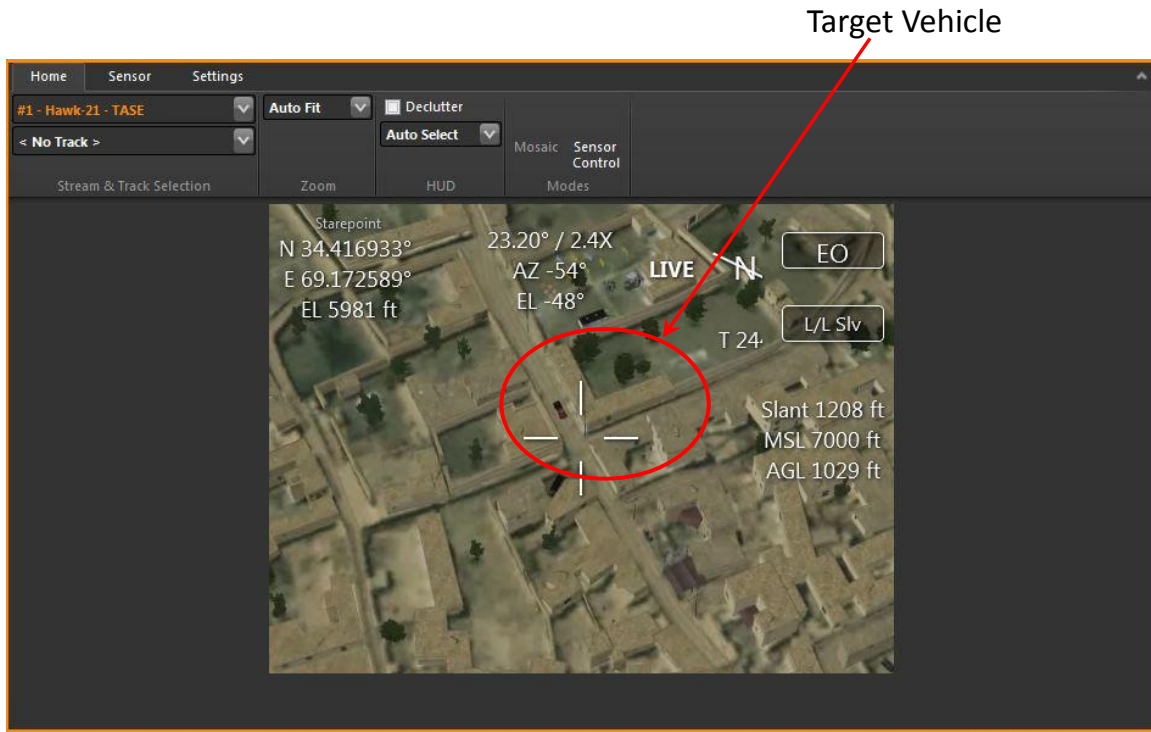


Figure 28. Target Vehicle in Urban Environment

Independent Variables

There were two independent variables used in this experiment. The experimenters were constrained by the time and resources available to implement specific faults within the simulation environment. These limitations were a critical factor in the selection of the independent variables. . The first independent variable was fault type and had five levels; data link, avionics, electrical, mechanical, and none (baseline – no fault present). Table 1 describes the different fault conditions. The faults were chosen to represent a variety of potential fault conditions that could also be implemented in a simulation environment. The second independent variable was fault indication with two levels; on or off. At the beginning of each trial the participants were told if they were going to be alerted to any faults during the trial.

Table 25. Fault Condition Independent Variable

Fault Name	Fault Description
Data link	Increased the delay of inputs from 1 seconds to 2 seconds
Avionics	No longer able to slew the sensor
Electrical	No longer able to zoom the sensor
Mechanical	Increased the turn radius by 50%
None	No fault occurred – baseline condition

Dependent Variables

As mentioned above, this experiment was an initial investigation, and was not meant to be the only experiment that addressed all issues, rather it was meant to help guide future research. The dependent variables were selected with that goal in mind. The most important of the metrics chosen was the performance metric of the time the target was in the sensor view. This reflected the main goal given to the participants which was to keep the target in the sensor view. All the other metrics looked at quality of the work, the participant's workload, and the participant's situation awareness (SA).

Even though several dependent measures were collected, not all were analyzed due to complications and time constraints. The planned data was collected on two different machines. Target data was collected on one machine and sensor data was collected on a different machine. In order for the data to be analyzed the data had to go through post processing to put it in a form that the analysis software can understand. A problem occurred due to the fact the two machines collected data at different rates, which caused the post processing to become very labor intensive and could not be automated. Unfortunately, again due to lack of resources and time, not all of the data was able to go through the post processing. This lack of resources also forced the prioritization of which variables to analyze. Table 2 shows the dependent measures that were collected and what information the metric provided.

Table 26. Dependent Variables

Metric Name	Information Provided
Time Target in Sensor View (Footprint)	Performance – more time in view, the better
Zoom Level	Quality – more zoomed in the better
Distance of Target to Center of Sensor Footprint	Quality/SA – the closer to the center the better
Number of Slews	Workload – fewer slews is better
Number of Zooms	Workload – fewer number of zooms is better
Number of Loiter Changes	Workload – fewer changes to loiter point is better

Participants

Thirteen participants (8 males, 5 females) were used for this experiment. Since only two of the participants were considered subject matter experts (SME), no analysis was done comparing SME's and non-SME's.

Apparatus

This study used four computers running the Windows 7 operating system that were connected to a local network through a 24-port Gigabit switch. Two computers were responsible for running MetaVR's Virtual Real Scene Generator (VRSG) with a custom version of the Afghanistan database. These computers were used as sensor feeds for the Vigilant Spirit Control Station via streaming video over the network. The third computer (Dell 6300) ran the Vigilant Spirit Simulation software. This software controlled the

simulation as well as the sending of distributed interactive simulation (DIS) packets of the entities in the scene. The fourth computer (Dell 6500) ran the Vigilant Spirit Control Station in conjunction with two 24-inch monitors set to a resolution of 1920x1280 pixels. Figure 29 shows the architecture of the simulation and Figure 30 shows the participant work station.

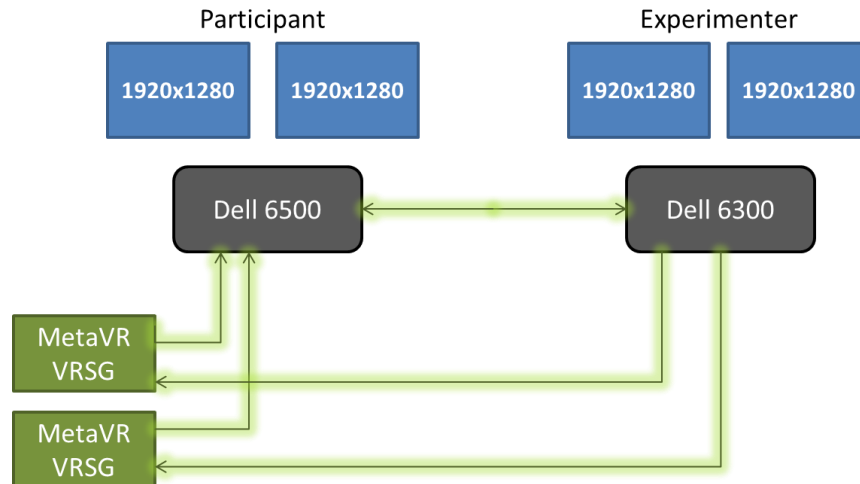


Figure 29. Simulation Architecture



Figure 30. Participant Work Station

Results

As mentioned previously, not all the data collected was able to be analyzed. The results discussed below are for the time the target was in the sensor footprint. This is the main performance metric and as such did provide a reasonable amount of insight about the impact of the different fault conditions. The analysis of the data takes into account the different target paths.

Time Target in View

The first analysis looked at the main effect of the different fault configurations (shown in Figure 31). Of the five different fault conditions only the avionics fault performed statistically significantly worse than the others ($p < .01$).

The second analysis looked at the impact that the presence of the warning had on operator performance. No significant difference was seen between the two conditions (see Figure 32).

A third analysis was done to look for interactions between the two independent variables. No statistically significant effects were seen (Figure 33).

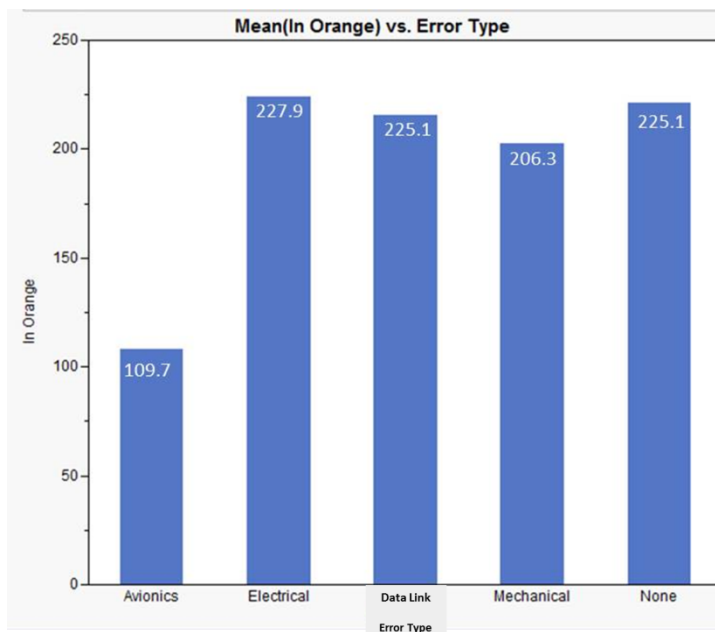


Figure 31. Time the Target was in Sensor Footprint under the Different Fault Conditions

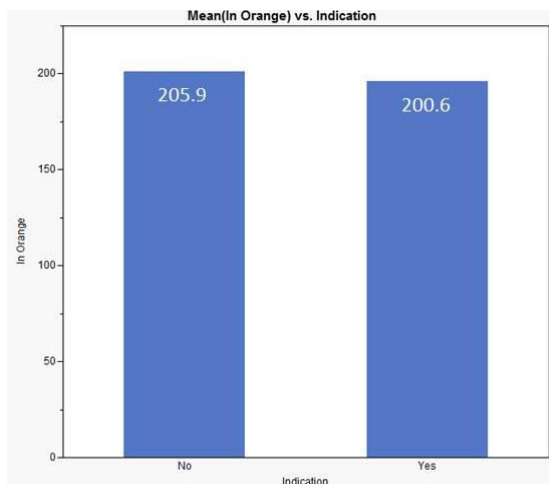


Figure 32. Time Target was in Sensor Field-of-View under the Different Indication Conditions

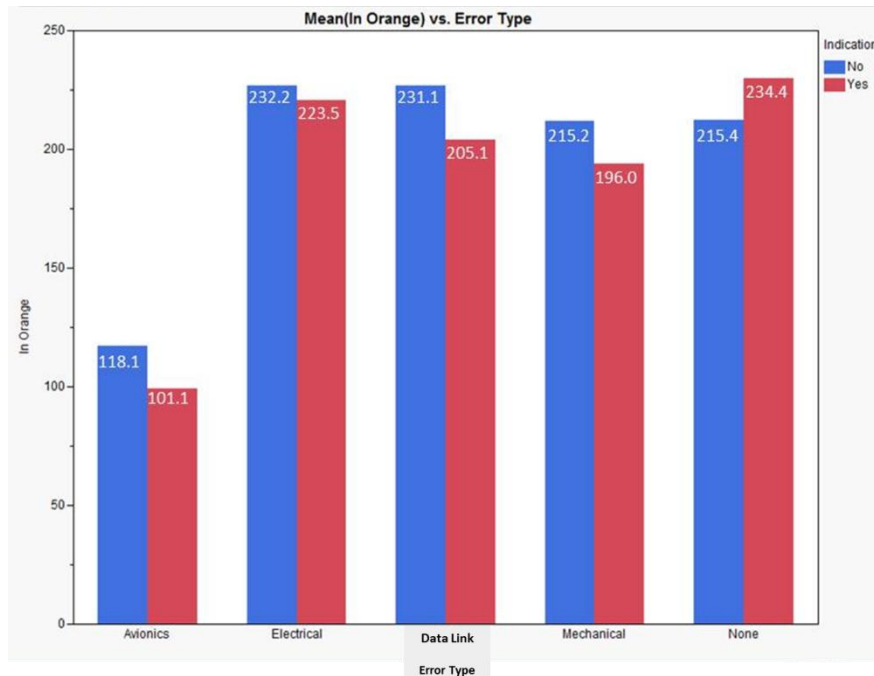


Figure 33. Time Target was in the Sensor Field-of-View under Both Independent Variables

Discussion

The objective data that was able to be analyzed was the most informative. The amount of time the target stayed in the sensor field-of-view is the main objective performance metric we collected. The data suggests that the presence of an indication of a fault alert was not very helpful. This is somewhat counter intuitive at first glance. However, when looking at the performance on the different types of faults, some significant conclusions can be drawn. Under faults that have less immediate impact (electrical, data link, mechanical) the participants performed about the same as the baseline conditions because of they were able to modify their behavior to compensate for the fault. In the avionics fault condition (which had a very drastic and immediate impact) the participants performed much worse. This would suggest that only providing an operator with a fault indication after the fault has occurred is less than optimal, especially in the case of severe faults like the avionics fault.¹ From experimenter observation, participants that would shadow the primary RPA with the second RPA didn't lose the target under the avionics fault, unlike those that did not shadow. This strongly suggests the need for the participant to have knowledge of the system state over time to better predict when a fault may occur. Having prognostic information could help operators better prepare for vehicle faults and the resultant degradation. An example of a predictive display is the spark line concept discussed in the following section.

RH RPA Human Experiment

¹ This may only be the case with type of RPA's being simulated for this experiment, because the options the participants had to mitigate the fault was to modify their behavior or bring in the other aircraft. If the operator had more options (e.g., rebooting the RPA to fix the avionics fault) the fault indication may have been more beneficial.

Objective

The objective of this experiment was to investigate individual and team performance and physiology when vehicle health degrades and task load increases. This experiment was intended to provide a better understanding of the effects of vehicle degradation and task load to support the development and validation of aids for Remotely Piloted Aircraft (RPA) mission decision makers. The aids would sense and assess the states of RPA assets (operators and RPAs), and help mission leaders decide which assets to allocate to which tasks.

Methods

Task

A team of three participants controlled four RPAs to complete a 69-minute scenario. The scenario was divided into nine epochs. Each epoch lasted approximately seven minutes. In the first six minutes of each epoch, the team was presented with a set of two to four tasks to accomplish. Task load and vehicle failures were manipulated in eight of these epochs. In the ninth epoch, participants dealt with a “catastrophic” event. At the end of each epoch, the participants were queried (through text chat) about their workload during the previous tasks.

Two team members were assigned to operator roles and one was assigned to a supervisor role. Each operator controlled two RPAs, operating as both sensor operator and “pilot” (route planner). The supervisor assigned tasks to the operators, as well as replied to situational awareness questions (through text chat) related to both operator activity and vehicle status.

Figure 34 shows the VSCS display interface for the operators. On the far left of the interface is the tactical situation display (TSD), which shows a satellite image of the region, major roads, and the locations of friendly assets. To the right of the TSD are the RPA controls. To the right of the controls are text chat windows used for communication among team members (top) and instructions/questions from the experimenter (bottom). The two video displays on the far right side of the interface show video from the two RPAs under the control of that operator.



Figure 34. RPA Operator Display Interface

Figure 35 the VSCS display interface for the supervisor. On the far left of the interface is detailed information on each vehicle. To the right of the vehicle information is the TSD. To the right of the TSD are text chat windows used receiving task assignments (top), for communication among team members (middle), and instructions/questions from the experimenter (bottom). The four video displays on the far right side of the interface show video from all four RPAs.

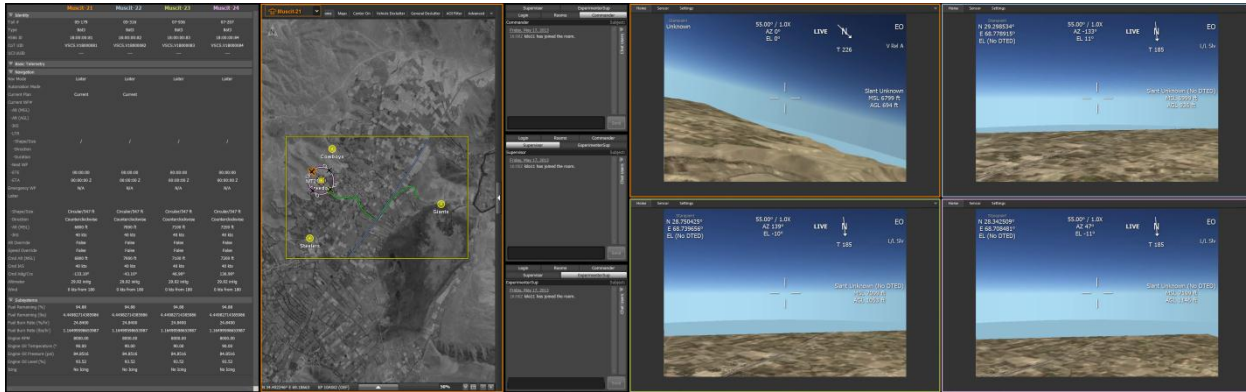


Figure 35. RPA Supervisor Display Interface

The scenario contained four types of tasks: search, monitor, follow unfriendly, and follow friendly. In the search task, participants were given the coordinates of a region in which armed personnel would exit from buildings one at a time. The task was to find those units and report their final coordinates. In the monitor task, participants were given the coordinates of a region in which a number of armed units were moving around. These units entered nearby buildings one at a time, and the task was to report which building each unit entered. In the follow unfriendly task, participants were given the coordinates of quickly moving vehicles that must be followed using the RPA sensors. In the follow friendly task, participants were given the coordinates of slowly moving vehicles that could be followed using a combination of the RPA cameras and TSD.

Design

Each participant engaged in the experiment for four days over two weeks, for approximately two hours each day. The participants were trained for the first three days. During the first two days, participants were trained individually. The first day of training focused on familiarization with the VSCS interface, vehicle control, sensor control, and terrain familiarization. The second day of training continued the previous day's training, and introduced chat communication, identification of friendly and unfriendly units, vehicle faults, and task types (search, monitor, follow unfriendly, and follow friendly). Participants were required to successfully complete 80% or greater of the assignments in a series of five-minute missions before continuing to the third day of training. The first four missions focused on one task type. The final two required performing two tasks concurrently, with the final mission including a vehicle fault. On the third day of training, participants were introduced to their teammates and trained on their randomly assigned roles. The team was required to successfully complete 80% or greater of the assignments in a 17-minute scenario before continuing to the fourth day of data collection. The team had three attempts to reach

criterion performance, with additional training given after a failure. Each evaluation was given with a different scenario.

On the fourth day, the participants completed the 69-minute mission described in the previous section. Task load and vehicle faults were manipulated across nine epochs in the scenario. Task load was manipulated by varying the number of concurrent tasks in an epoch, with two tasks in the low task load condition and four tasks in the high task load condition. Supervisors were encouraged to balance the task load evenly across the two operators. Vehicle faults were either present or not. When vehicle faults were present, they occurred in two vehicles, one assigned to each operator. This resulted in four types of epochs: low task load without vehicle fault, low task load with vehicle fault, high task load without vehicle fault, and high task load with vehicle fault. Two epochs of each type were presented in the scenario.

Two types of simulated, electronic vehicle faults were used: intermittent communications and loss of sensor zoom. The intermittent vehicle control results in frequent loss of RPA routing control and sensor control for the duration of the epoch. The loss of sensor zoom results in a complete loss for the whole epoch. The type of vehicle fault, as well as the type of task (search, monitor, follow unfriendly, and follow unfriendly), was randomly assigned to each epoch.

Several physiological and behavioral data sets were recorded. Eye movements were recorded with SmartEye Pros, remote, 60 Hz, six-camera eye pupil-center/corneal-reflection eye trackers. Heart and brain activity were recorded with BioSemi ActiveTwos, 128-channel, 2,048 Hz electroencephalograph (EEG) plus electrocardiograph (ECG) systems. Keystrokes, mouse clicks, and mouse movements were recorded using RUI (Kukreja, Stevenson, & Ritter, 2006). The VSCS interface was recorded using Morae® usability software. Text chat was recorded using Openfire XMPP server (Ignite Realtime, 2013). Not all data was recorded for each team role (operator or supervisor). **Table 27** shows which datasets were recorded for each role.

Table 27. Datasets recorded for the operator and supervisor participant roles

		Data type					
		Eye movements	EEG	ECG	Key & mouse	Interface recording	Chat
Role	Operator	X	X	X	X		X
	Supervisor			X	X	X	X

In addition to the objective data listed above, subjective data was collected with a number of questionnaires. These included: An in-task workload measure based on Van Orden (2001) was collected at the end of each epoch in the Day 4 mission. This task load measure used a 0 to 100 scale, which was anchored with text related to level of effort. The Dundee Stress State Questionnaire (Matthews et al., 2002) and Stanford Sleepiness Scale (Hoddes, Zarcone, Smythe, Phillips, & Dement, 1973) were presented before and after the Day 4 mission. The Pittsburgh Sleep Quality Index (Buysse, Reynolds, Monk, Berman, & Kupfer, 1989) was administered before the Day 4 mission.

Participants

Twenty one people (seven teams), five female and sixteen male, ranging in the age from 18 to 54 (mean = 27.8) from Wright-Patterson Air Force Base and the surrounding community participated in the experiment. Participants were screened as follows: 18 to 65 years of age; no known motor, perceptual, or cognitive conditions that preclude them from operating a computer, reading small characters on a computer monitor, hearing and comprehending verbal commands presented through headphones, or learning complex, computer-based tasks; the ability to fluently communicate in written and spoken English.

Nine additional participants did not complete the experiment, five female and four male. Of these nine, six did not pass training criteria, two voluntarily withdrew during training, and one had scheduling conflicts midway through training. The two participants who withdrew during training stated that they “were not good at video games” and felt the task was too difficult to be worth continuing.

Participants were paid \$15/hour to participate. Additionally, each participant was awarded a \$2 bonus for each task successfully completed by the team during data collection on day four, with a maximum bonus of \$45 per participant.

Apparatus

All three participant stations were configured the same. VSCS, version 4.12, was displayed on two 24” LCD monitors with resolution of 1920 x 1200. VSCS ran on a Windows 7 workstation with an Intel 3.2GHz Xeon processor, 12GB of RAM, and an ATI FirePro V4800 video card. The participant stations were visually isolated from each other and the participants wore in-ear headphones (Sennheiser CX300B MK II) to discourage verbal communication and enhance the simulation of a distributed RPA team.

The Vigilant Spirit simulator, version 4.12, ran on Windows 7 workstation with an Intel 3.2GHz Xeon processor, 12GB of RAM, and an ATI FirePro V4800 video card. The Openfire text chat server (Ignite Realtime, 2013) ran on the same system. SubrScene, version 12.12.08w, provided the simulated video for the RPA sensor feeds. Each RPA’s video was generated using a dedicated Windows 7 workstation with an Intel 3.2GHz Xeon processor, 12GB of RAM, and an ATI FirePro V4800 video card.

All of the systems listed above, including SmartEye and BioSemi systems, were networked through an isolated, 1000BASE-T network. The clocks of all computers on the network were synchronized using the Network Time Protocol (NTP). The BioSemi hardware for each participant were daisy-chained together and connected to single computer.

Results

Only a subset of the data collected in the study have been analyzed. Key data that investigate whether the task load and vehicle failure manipulations used did affect participants’ perceived workload and physiology have been analyzed. Ongoing efforts are attempting to uncover details in the unanalyzed data to make recommendations for assessing and mitigating the workload and other consequences of vehicle faults that the participants experienced.

In-Task Subjective Workload Rating

The in-task subjective workload rating was analyzed as a function of task load, vehicle faults, and participant role. Participant was used as a repeated-measure grouping factor. Linear mixed-effect models were used for the analysis, using the R environment

(Pinheiro, Bates, DebRoy, Sarkar, R Development Core Team, 2011; R Development Core Team, 2011).

Figure 36 shows the participants' mean subjective workload rating as a function of task load and vehicle faults. The subjective workload was higher in the participants were performing two task concurrently, $F(1,112)=48.7$, $p < .001$. The subjective workload was not higher when vehicle faults were present, $F(1,112)=3.5$, $p=.065$. However, there was a significant interaction between task load and vehicle fault, $F(1,9)=5.1$, $p=.026$; vehicle faults had a larger effect on subjective workload when the participants were only performing one task. This may be the result of a ceiling effect, but the distribution of subjective workload ratings shown in Figure 37 do not support that conclusion. Regardless, further investigation will be needed to determine if the greater effect of vehicle faults on single taskings is meaningful.

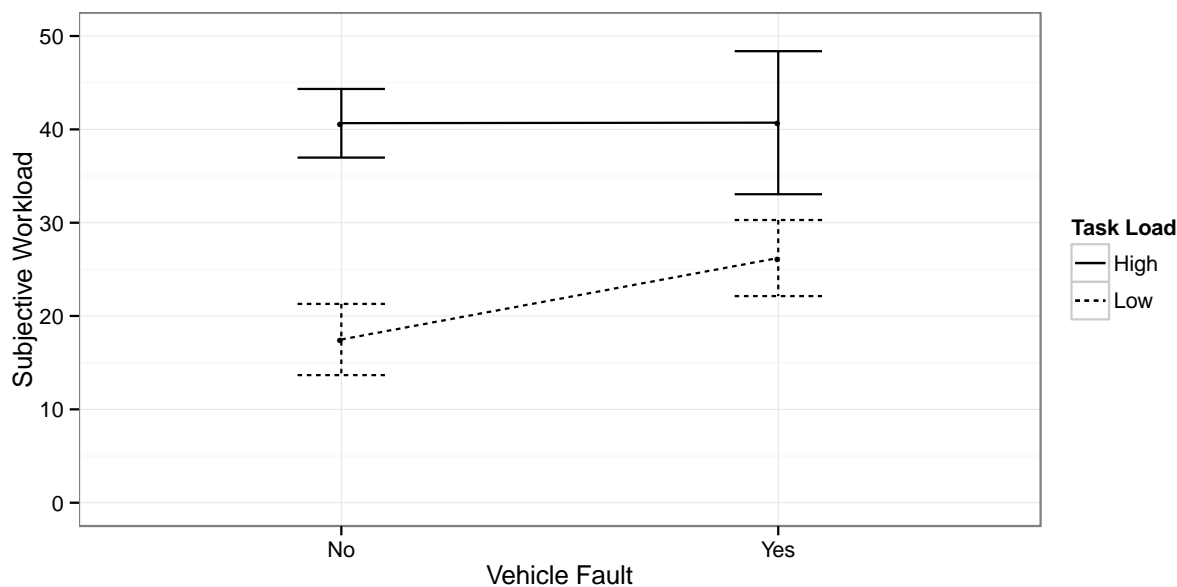


Figure 36. Mean subjective workload as a function of task load and vehicle fault. (Error bars show ± 1 standard error of the means.)

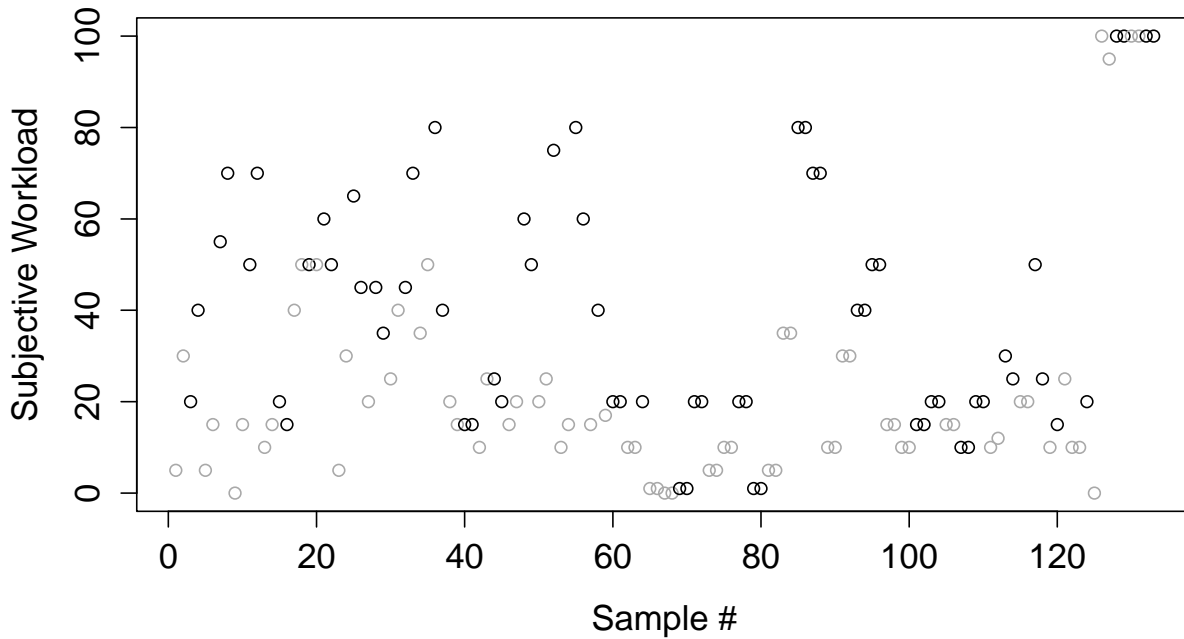


Figure 37. Scatter plot of subjective workload measures

(Black circles show the high task load values and the grey show the low task load values. The lack of clustering near the top or bottom of the distribution suggests there were no ceiling or floor effects.)

Figure 38 shows participants' mean subjective workload rating as a function of task load and participant role. The overall subjective workload did not vary by participant role (operator or supervisor), $F(1,112)=0.35$, $p=.571$. However, there was a significant interaction between task load and role, $F(1,112)=7.9$, $p=.006$; the operators were affected more by the task load manipulation than the supervisors, while the supervisors were at a consistently higher workload.

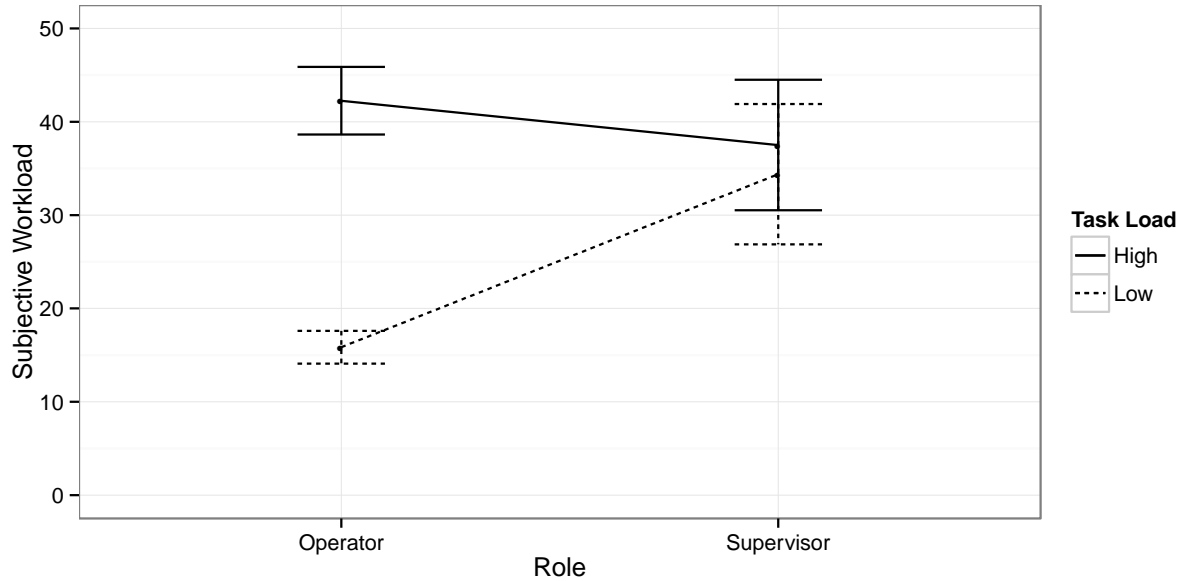


Figure 38. Mean subjective workload as a function of task load and participant role
(Error bars show ± 1 standard error of the means.)

Heart Rate Variability (HRV)

The continuous physiology measurements collected in this study provided many ways in which to measure the effects of the task load and vehicle failures on physiology. For this initial analysis, heart rate variability (HRV) is analyzed, as it has been shown to be a very reliable physiological indicator of workload (Wilson & Eggemeier, 1991). The mean HRV, measured as the standard deviation of normal-to-normal in the ECG data, was analyzed as a function of task load, vehicle faults, and participant role. Participant was used as a repeated-measure grouping factor. Linear mixed-effect models were used for the analysis, using the R environment (Pinheiro et al., 2011; R Development Core Team, 2011).

Figure 39 shows participants' mean HRV as a function of task load and vehicle fault. A high task load resulted in a lower HRV, $F(1,145)=9.80$, $p=.002$, as is seen in the literature (Wilson & Eggemeier, 1991). Vehicle faults also resulted in a lower HRV, $F(1,145)=6.27$, $p=.013$. However, HRV did not vary meaningfully with participant role, $F(1,11)=1.18$, $p=.300$. No interactions were significant.

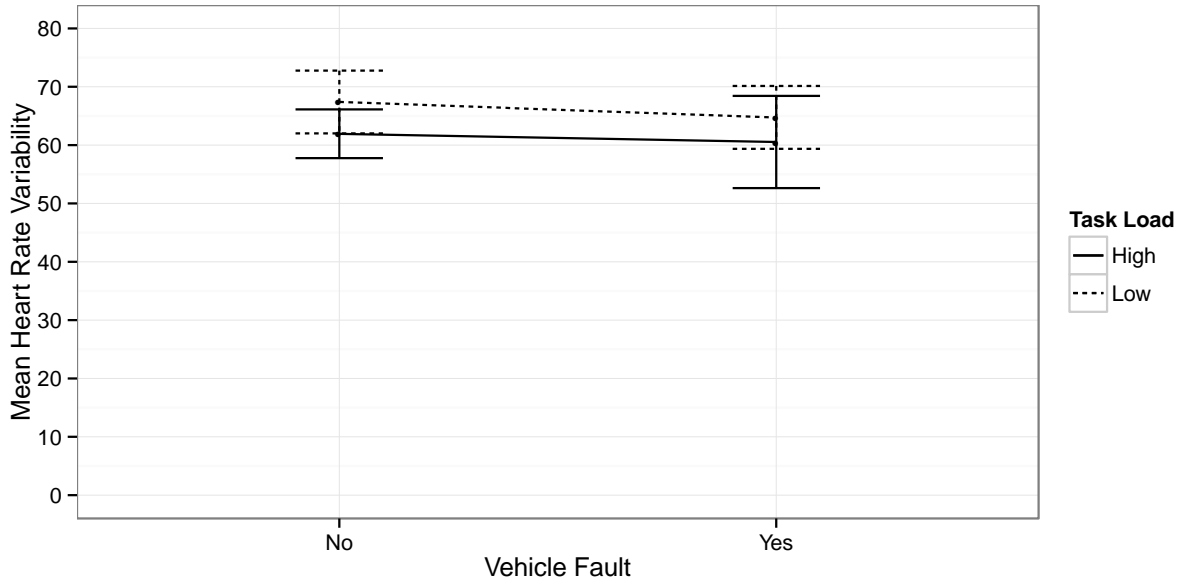


Figure 39. Mean heart rate variability (HRV) by epoch
(Error bars show ± 1 standard error of the means.)

Discussion

These preliminary results support the need for tools that help decision makers in RPA teams; the design of such tools was one of the original goals of this STT. While the supervisors did not directly take part in the tasks, simply maintaining awareness of team and asset performance resulted in a substantial increase in workload that was greater than or equal to that of the operators. This increase in workload was observed in both the subjective and physiological indicators. Additional analysis of the data from this study, and future studies, will be needed to inform the design of useful decision support tools.

The results also support the need for integrated system health status information. Even though the participants performed adequately in the presence of vehicle faults during training, the presence of vehicle faults increased both subjective workload and the physiological indicator of workload. The subjective data suggests that even in situations in which the operators have additional, healthy assets to deploy (i.e. in the low task load condition), the impact of unpredictable vehicle health substantially increases workload. Short-term, advanced warning of vehicle degradation, perhaps achieved through more effective system health monitoring, may have allowed the operators to utilize the second vehicle, or allowed the supervisor to reassign tasks, in a timely manner.

Future Research

Overall the initial study was successful in guiding future research. There are two areas for consideration for future research.

Predictive Displays

Future research should investigate the use of predictive displays, such as the spark line² (Tufte, 2006) shown in **Figure 40**. Data that is collected from the various subsystems

² A spark line is a very small line chart, typically drawn without axes or coordinates. It presents the general shape of the variation (typically over time) in some measurement, such as temperature or stock market price, in a simple and highly condensed way

most likely has a normal range. Providing the operator with a visual representation of the normal range and a compressed time history of the systems behavior may allow the operator to anticipate a failure. More research is required to better understand the specific subsystem faults and how they might fail to determine if the spark line concept has merit. The classification of faults is the start to this additional research.



Figure 40. Spark-Line Type Predictive Display

Classification of Faults

Future research should further investigate the fault types and the associated mitigation of those faults, for various classes of RPA. The research should be designed to allow extrapolation to untested faults and ensure different fault conditions are addressed. For example, one way to classify the faults would be by how salient it is to the operator. By classifying different faults and testing them under different conditions it may be possible to predict the impact of untested faults.

Clearly the current study was very preliminary and there is a great deal of research yet to be done before any conclusive statements can be made regarding the value or impact to the operator of vehicle health status information. However, providing health status information to the operator will undoubtedly become more important as RPAs become more and more autonomous. What information should be provided and how it should be displayed to the operator definitely warrants more in-depth research.

References

- Buyse, D. J., Reynolds, C. F., III, Monk, T. H., Berman, S. R., & Kupfer, D. J. (1989). The Pittsburgh Sleep Quality Index: A new instrument for psychiatric practice and research. *Psychiatry Research*, 28, 193–213.
- Hoddes, E., Zarcone, V., Smythe, H., Phillips, R., & Dement, W. C. (1973). Quantification of sleepiness: a new approach. *Psychophysiology*, 10(4), 431–436.
- Ignite Realtime (2013). Openfire (Version 3.8.2) [Computer software]. Palo Alto, CA: Jive Software.
- Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI: Recording user input from interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4), 656–659. doi:10.3758/BF03193898
- Matthews, G., Campbell, S. E., Falconer, S., Joyner, L. A., Huggins, J., Gilliland, K., et al. (2002). Fundamental dimensions of subjective state in performance settings: Task engagement, distress, and worry. *Emotion*, 2(4), 315–340. doi:10.1037//1528-3542.2.4.315
- Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., R Development Core Team. (2011). *nlme: Linear and Nonlinear Mixed Effects Models* (3rd ed.).
- R Development Core Team. (2011). *R: A Language and Environment for Statistical Computing* (2nd ed.). Vienna, Austria: R Foundation for Statistical Computing.

- Stilson, M. T. (2008). *Multi-UAV Control: An Envisioned World Design Problem*. Doctoral dissertation, Wright State University, Dayton, OH.
- Tufte, Edward., *Beautiful Evidence*, Graphic Press, 2006.
- Van Orden, K. (2001). *Monitoring moment-to-moment operator workload using task load and system-state information* (No. 1864). SSC San Diego.
- Wilson, G. F., & Eggemeier, F. T. (1991). Psychophysiological assessment of workload in multitask environments. In D. Damos (Ed.), *Multiple-task performance* (pp. 329–360).