User-Centered Design and Experimentation to Develop Effective Software for Evidence-Based Reasoning in the Intelligence Community: The TRACE Project

Improving reasoning in intelligence analysis is of vital national importance. The trackable reasoning and analysis for crowdsourcing and evaluation project takes a user-centered design approach combined with rigorous experimentation to ascertain effective structured techniques to support high-quality reasoning. Results suggest a light structure is more effective than a rigid structure.

Intelligence analysts have long encountered issues with avoiding cognitive biases—shortcuts that can lead to errors in human judgments, while engaging in evidence-based reasoning. Studies of the intelligence community have documented several intelligence failures over the years, yet identified solutions are rarely implemented or systematically tested for efficacy. Tools and techniques that are intended to help analysts engage in more effective evidence-based reasoning have been offered over the years.
These structured techniques (STs) aim to improve analysts’ reasoning, but they often do not give analysts sufficiently concrete solutions for analytic challenges. Instead, current STs provide fairly general process checks, and many are inefficient, ineffectual, and cognitively burdensome—negatively impacting fluid reasoning, particularly analogical reasoning. Analysts tend not to use them regularly (for example, a recent examination identified ST use in only 6 of 29 CIA reports) and instead fall back on less cumbersome, instinctive processes.

Based on interviews with intelligence analysts, we know that several types of analytic work are constrained by several constraints. First, information available to analysts, which is the basis for their reasoning, is not directly relevant and credible. Some information is deceptive by intent, and much else is otherwise misleading or incomplete. Second, analysts often face time constraints to conduct their evidence-based reasoning and report writing. Third, analysts have the same cognitive biases as everyone else. Their training in analysis and reasoning does not completely prevent them from being subject to the normal functioning of human cognition when processing complex and contradictory information under time constraints. All of these factors make analysis challenging even for the most highly trained and skilled intelligence analyst.

The trackable reasoning and analysis for crowdsourcing and evaluation (TRACE) project, based at Syracuse University, aims to fill a critical gap in empirical testing of STs to improve evidence-based reasoning. The project is supported by the Intelligence Advanced Research Projects Activity (IARPA) through the Crowdsourcing Evidence, Argumentation, Thinking and Evaluation (CREATE) program. The goal of CREATE is to evaluate STs in conjunction with crowdsourcing methods to ascertain if crowd-generated analysis can support more effective evidence-based reasoning.

The TRACE project proceeds in three phases. In the first phase, as there is no consensus around the effectiveness of different STs, we decided to take an exploratory approach and to rigorously experiment with a variety of STs that might aid in reasoning, with the goal of using results from the experiments to drive our own software development. To do so, we created the TRACE web-based application to experiment at and internal test techniques that might improve the quality of reasoning. Ultimately, independent evaluation by a team specifically contracted by IARPA to validate the results will provide further evidence of the reliability of our application. The goal is to understand how different techniques may support evidence-based reasoning. We decided to focus our software development in this first phase on individual reasoning to better understand the mechanisms and techniques that are effective in improving the quality of reasoning at the individual level. During the second and third phases, we start with the best STs and designs from individual-level reasoning and add crowdsourcing features to them, continuing with our rigorous experimental approach to determine effectiveness. Across the three phases, increasingly more complex reasoning problems will challenge our techniques. The TRACE project and CREATE program are currently in the first phase. This paper describes our designs and testing up to this point and outlines how specific techniques seem to improve overall reasoning quality as well as specific aspects of reasoning in the context of intelligence analysis. Results so far from testing in the first phase suggest that rigid STs commonly used in the IC are no more effective than unaided reasoning. Instead, providing tools and nudging to flexibly structure and support reasoning is proving more effective.

EVIDENCE-BASED REASONING CHALLENGES AND APPROACHES IN THE INTELLIGENCE COMMUNITY

STs were developed in the intelligence community to help overcome cognitive shortcomings by encouraging intelligence analysts to systematically identify and evaluate relevant information, weigh contradictory facts, and consider alternative explanations. Current STs include the key assumptions check (KAC), which promotes reflection on the assumptions that are driving reasoning, and red-team analysis, which asks analysts to consider imaginatively the behavior of adversaries from another’s cultural and organizational viewpoint. In practice, however, interviews we conducted with intelligence analysts suggest that some of these approaches rely on laborious reasoning steps or require managers to bring in cultural experts to review analysis from the adversary’s point-of-view.
Some of these techniques and similar approaches are inscribed in software. Argument mapping software, for example, provides a visual representation of key argument components in place of complex and confusing text. Mapping breaks an argument down to its key parts (e.g., premises, warrants, and evidence) and displays them in a visual form so that users learn how to properly construct arguments and identify their weaknesses with existing ones. Although argument mapping systems can improve reasoning skills, they are laborious and place heavy cognitive demands on users. Studies suggest that people find them difficult to use and unappealing.

In addition to usability issues associated with current reasoning tools, analytic reports tend to lack clarity and defensibility. Analytic reports, including the underlying reasoning and knowledge limitations, need to be easily comprehensible and accessible to policymakers and readily support unbiased decision-making by customers, such as the President, congressional committees, and military or civilian officials. Recipients of the reports also need sufficient information to feel assured that they understand their options and the limitations of the analysis without being overwhelmed with detail. Standalone, hard copy documents and highly condensed briefings meet some reporting requirements. Complex analyses in such formats, however, do not easily permit readers to drill down into the reasoning process or see the evidence or assumptions that are at its core.

Crowdsourcing offers some promise to improve evidence-based reasoning. Existing crowdsourcing strategies recruit participants from larger groups or populations to develop and identify high-quality solutions to problems. One typical application of crowdsourcing within an organization or business is to invite a large group of people to provide assistance on a task that might otherwise have been done by an individual. Crowdsourced work is often done by volunteers who are passionate or highly knowledgeable about the task, and it is increasingly recognized as a “legitimate, complex, problem-solving model.”

Crowdsourcing processes can range from writing whole Wikipedia articles where people contribute all phases of the project to microtasks where participants get assigned to small pieces of a larger project, such as in EyeWire, in which workers map neurons in a game-like setting or Zooniverse.org, in which workers can engage in a variety of citizen-science projects, such as classifying images of plants. The size and quantity of tasks being crowdsourced vary greatly from Amazon Mechanical Turk, with millions of small tasks, to open source software development projects (e.g., Mozilla) where crowd participants work on fixing bugs or adding features (e.g., open source software development).

While crowdsourcing has already been used in a variety of contexts inside and outside the IC ranging from medicine to astronomy to encyclopedic knowledge to prediction markets, the approach is associated with considerable challenges. Identifying and retaining skilled workers requires matching project needs with motivations that attract participants. Ensuring quality of the crowdsourced work products requires integrating checks and monitoring systems that flag poor quality work or possible efforts to sabotage the process. Combining piecemeal work into larger products requires planning and experimentation to identify the right aggregation approach.

Despite these potential challenges, crowdsourcing offers several advantages that may prove useful for easy intelligence problems such as identifying objects in visual evidence and more complex ones that involve understanding past events or producing long-term scenarios. One advantage of crowdsourced projects involving crowds is the ability to aggregate individual work in ways that neutralize the impact of inaccurate, weak, or unreliable submissions (perhaps from sabotage). Another advantage deeply intertwined with the nature of crowdsourcing is modularized work, in that individuals can choose to work on the tasks in which they are most interested or for which they have the best skills. Research suggests that crowdsourcing projects designed with intrinsic motivational factors in mind such as project needs and tasks that people find meaningful, progress tracking, and feedback can help promote engagement and maximize work quality.

In addition to crowdsourcing and STs, the TRACE team is investigating other techniques for changing behavior that can be applied in the context of evidence-based reasoning. Nudging is the use of positive and indirect suggestions to influence behavior. Nudging is increasingly used in the software system design to promote desirable behaviors based on user’s actions. Past research has found nudging to be an effective way to promote behavioral change, including in the areas of personal information disclosure, and civic engagement. Given that people make mistakes and are swayed by cognitive biases in decision-making, nudging is another technique TRACE can apply to support more effective reasoning. Moreover, nudging that notifies users of relevant materials, data, or analysis available from
other crowd workers may reduce cognitive load while improving speed and accuracy of analysis. Little is known, however, about the efficacy of nudging in intelligence analysis, and TRACE will contribute to understanding the usefulness of the approach in this context.

Similarly, although checklists have had limited success as a key element in the STs currently employed in the IC, strategically embedded checklists may be a valuable asset in a reasoning system. Research suggests that checklists can significantly improve attention to critical process steps, team coordination, and team decision-making. Checklists have been developed and examined in a variety of contexts, including policing and crime prevention and in surgery. These types of checklists have been systematically developed based on close observation of tasks and process experimentation to ensure correct behaviors at critical junctures. We will integrate checklists and nudges (e.g., reminders and guidance) on process steps to determine the benefits to reasoning and analytical writing.

Software design itself can be an important dimension helping to support evidence-based reasoning. Design science is concerned with “how things ought to be, with devising artifacts to attain goals.” Building effective software requires maximizing utility by choosing the optimum design from a range of possibilities. User-centered design (UCD) approaches can help designers to more actively focus on the needs of the users by positioning them at the center of all design activities.

**THE TRACE APPROACH**

TRACE is designed to enhance the analyst’s natural ability to reason and explain his or her judgments (see Figure 1) by implementing a variety of different approaches. In the first design and research phase, we conducted two formal experiments, three rounds of pilot testing, and several smaller mini-experiments with a variety of STs aimed at improving the overall quality of evidence-based reasoning as embodied in analytic reports. This section outlines the STs we designed and tested and what our results suggest so far. First, though, we explain the analytic process and the evaluations that drive our assessment of the effectiveness of our application.

The focus of TRACE user testing and experimentation is primarily on examining the outcome of the analysis process—the analytic report. The two main goals for the analytic reports produced by the users of the TRACE application are that (1) products be well reasoned and (2) products will be well understood by consumer surrogates based on indicators of quality of reasoning outlined by Intelligence Community Directive 203: a document specifying what quality of reasoning should include in the context of intelligence reasoning products. Good reasoning includes factors such as detailing the uncertainties associated with the findings, assumptions that informed the reasoning, and how findings might change if assumptions were wrong. It also provides logical argumentation to justify the findings and details alternative hypotheses and why they do not hold. The analytic product is, thus, a written report of the findings of a given problem. In the first phase of the project, we are experimenting with problems based on hypothetical, constrained situations inspired by real-world events and fictionalized so that reasoners cannot look up additional information. For example, one of our problems fictionalized an actual violent altercation at a political protest in a European city, which is transposed to a different location, and asks users to establish the motives for the violence. Users are provided several manufactured documents that are constructed to resemble real documents, such as news reports, government briefings, and even a Facebook page of one of the key actors as sources of information to evaluate and as the basis to render a judgment.

We adopt a UCD approach to building and testing the TRACE software. In other words, we make the needs and abilities of end users the central focus of our development efforts. This approach entails systematically sampling members of the population of potential users, inviting them to test and provide feedback on working prototypes of the software, as well as suggest ways to improve the software. To ensure that we consider usability issues throughout the software development and evaluation processes, we have designated members of the research team to oversee user interface (UI) design and evaluation. The tasks of the design and user-testing team include storyboarding, rapid UI prototyping, designing formative evaluation tasks for internal testing, and collecting positive and negative feedback on UI design. The development team, including subject matter experts and intelligence analysts, will continue to identify issues and improve features in parallel to the iterative software development process.
Within the first year, we developed several STs and systematically tested them. Among the techniques that we focused on in this first phase of the project is structured debate (SD). Debate allows for the careful examination of positions and evidence and is effective at promoting critical thinking and behavior change. Psychological research concerning debates and argumentation suggests that humans are more effective at evaluating arguments and reasons in the context of a debate and that individuals who reason together in a group are more effective at reasoning than if doing so alone. Our design of SD started as a self-debate environment where users created arguments and were asked to imagine a worthy opponent and channel that opponent’s arguments. In testing, we found that SD held promise at helping people to better justify their findings in some of the reasoning problems we created. In a second iteration of the design, we shifted the framing from a “debate” to a consideration of “pros and cons.” Similar to the debate design, reasoners had to advance multiple hypotheses and then outlining the pros and cons for those hypotheses. Our testing of the pros and cons suggested the technique held promise in improving particular aspects of reasoning, such as the quality of justification.

The TRACE team also examined the analysis of competing hypotheses (ACH) ST. Significant cognitive science research and analytical experience suggests the value of ACH as an analytic method used within the IC. The method, however, is not uniformly and frequently used due to the comparatively large investment of time and effort required for complex problems. ACH has inherent difficulties that researchers have not yet sufficiently explored. Participants struggle to formulate multiple mutually exclusive hypotheses for other than relatively simple and not necessarily realistic binary judgments. Uncertain and incomplete information further can render an unmanageable number of potential hypotheses. The notion of diagnostic evidence—data that conclusively prove that particular reasoning paths can be ignored—is tantalizing, but such evidence is difficult to define and discover in fluid situations where human planning and motivations are involved or when multiple interdependent forces may be operating. Finally, the cognitive and practical difficulties of defining and collecting evidence to refute hypotheses are significant. Advocates of this method hoped that computer support for ACH would help increase its usability, but so far, it has not done so. This may be due to the ways in which the method has been implemented in software.

We first designed ACH by adhering to the structure outlined by Heuer. After multiple rounds of testing, we found that it was not a significant aid in improving reasoning as compared with unaided reasoning. There was no substantial improvement in the accuracy of the findings or the quality of...
reasoning. Consequently, in later versions, we pulled aspects of this multistep STs apart. One of our key design investigations is examining how best to support a critical first step of ACH: hypothesis generation. A core challenge of ACH is that the technique requires reasoners to generate multiple hypotheses, and those need to be critically evaluated against the evidence. The quality of the hypotheses generated may significantly impact the process, however, as multiple hypotheses might be misleading or distracting when they are of poor quality. Thus, we focused our efforts on designing mechanisms to better support the generation of high-quality, testable hypotheses. Internal testing suggests that reasoners who develop more hypotheses that are of high quality—that is, significantly different from each other and testable—tend to produce a more thoughtful analysis.

We also designed a version of KAC: an ST that aims at improving evidence-based reasoning through the process of critically evaluating and examining assumptions that may undermine a judgment. The process of scrutinizing assumptions—those the reasoner may hold, those that are in the information available, as well as the assumptions embedded in the request for information from the customer—helps analysts by highlighting the circumstances in which assumptions could be challenged and ultimately helps them evaluate the impact of assumptions in the final judgment. Our internal testing with KAC suggested that the technique is effective. Moreover, our testing suggests that reasoners who are not assigned to the KAC technique will often overlook their assumptions and not fully recognize the potential impact of assumptions in their analysis.

TRACE also adopts nudges as just-in-time feedback mechanisms to guide users through the reasoning and reporting steps, as well as to provide correction, alternative frames, and recommendations for next steps. The nudging features in TRACE at this stage are based on simple behaviors that are observed by the software. For example, when users have opened a source to review, if they have not used the tagging tool—which is a memory and synthesis aid—the software will pop-up a note that encourages the use of that tool. The user can disregard the suggestion, which is why it is a nudge and not a rigid requirement. In the future, nudging will be augmented with additional computational techniques, such as finding patterns or anomalous behavior programmatically to support reasoning tasks and the generation of analytic reports. In our experimentation so far, nudges effectively guide users toward more quality reasoning. For instance, nudging users to review hypotheses to make sure they are testable can influence them to edit and improve their hypotheses.

A key task of evidence-based reasoning is evaluating the information relevant to the analysis. Intelligence analysis is a three-step, often nonsequential, process of (a) sense making, (b) critical evaluation, and (c) synthesis of the available information. Based on user feedback, we developed a set of tools to aid source evaluation, organization, and sense making. To help reasoners process and understand information from multiple sources, we have included tools in TRACE to enable highlighting, commenting, and tagging. The tagging feature is designed to help reasoners identify and categorize key pieces of information. TRACE nudges users to think about actors, events, facts, and assumptions as categories to tag information, but users can also create their own tags. The commenting feature is also designed to aid sense making, helping reasoners connect their own opinions to the available information. To facilitate user analysis and report writing, all tags and comments are visible in the main workspace of TRACE. Users may also take other kinds of notes in a space designed to be flexible and to foster different styles of analysis—from lists to elaborated paragraphs. Taken together, these three tools aid processes of critical evaluation and synthesis of information.

Finally, the software’s analytic product design includes a checklist of attributes of a completed report. This checklist synopsizes the features of their evidence-based reasoning that needs to be made explicit for an external audience to understand the final judgment, its justifications, as well as the alternative hypotheses considered, as well as assumptions made that could change the findings. The checklist features change depending on some of the tools users’ employ, as well as key features of the problem they aim to solve. The checklist serves as a final review for the reasoner, and its implementation has improved the overall quality of reasoning of the analytical products.

TRACE GOING FORWARD

TRACE’s current design utilizes STs as a collection of tools that improve evidence-based reasoning as ascertained from user testing and rigorous experimentation. The next step in TRACE’s design is to use
crowdsourcing to address shortcomings in intelligence work by breaking up tasks and examining the division of labor. As previously discussed, crowdsourcing provides a means of collective innovation and content creation from large groups of practitioners. TRACE crowdsourcing aims to improve analysis results by reducing the systematic and random errors individuals may generate. Because switching tasks can increase cognitive load and interrupt working memory required for complex reasoning, the TRACE software design will provide context-sensitive searches and aggregate relevant information generated by others to aid analysis. TRACE will capture insights that users generate as they analyze and identify linkages among facts, people, and events. TRACE will indicate relevance or accuracy of information through aggregated rankings. As we move from individual-level reasoning to crowds, we will continue to rigorously test the techniques implemented and iteratively develop our software to support intelligence analysis. Our goal is to identify how crowdsourcing techniques can be used for effectively mitigating cognitive load, distributing critical evidence-based reasoning tasks, and harnessing the collective intelligence of people with diverse skills and knowledge.

Building upon user behavior patterns gleaned from log data obtained during the first phase of this project, we aim at developing a context-aware, smart-nudging system for TRACE to guide users and provide feedback when needed. Background software processes, using advanced natural language processing and machine learning techniques, will be used to mitigate extrinsic cognitive load. TRACE’s automated support will allow users to generate and assess multiple hypotheses, render well-supported judgments, and overcome common cognitive biases when addressing the constrained and unconstrained problems that face our nation.

Although TRACE is still in its first phase of development, internal testing results indicate that the software is effective at improving the quality and accuracy of reasoning. By taking a UCD approach, combined with rigorous experimentation, we are developing effective techniques and software to aid intelligence reasoning.

ACKNOWLEDGMENTS

This project was funded by the Intelligence Advanced Research Projects Activity (Contract number: 2017-16121900004). The TRACE project is indebted to SRC, Inc., our project’s software developers and integral team members: Deborah Plochocki (Product Owner), Louis Nau (Scrum Master), Andrew Whelan, Laura Simonetta, Ryan Conner, and Timothy Bower. Special thanks to Dr. Nancy McCracken, who helped wrangle log data, and to the doctoral students who made this project go: Corey Pavlich, Eric Tsetsi, Jerry Robinson, Olga Boichak, Jordan Canzonetta, Feifei Zhang, Rhema Zlaten, Greg Russell, Niraj Sitaula, and our subject matter experts Dr. Sarah Taylor and Roc Myers.

REFERENCES


**ABOUT THE AUTHORS**

**Jennifer Stromer-Galley** is with Syracuse University. Contact her at jstromer@syr.edu.

**Patricia G. C. Rossini** is with Syracuse University. Contact her at patyrossini@gmail.com.

**Kate Kenski** is with the University of Arizona. Contact her at kkenski@email.arizona.edu.

**James Folkestad** is with Colorado State University. Contact him at james.folkestad@colostate.edu.

**Brian McKernan** is with Syracuse University. Contact him at bmckerna@syr.edu.

**Rosa Mikeal Martey** is with Colorado State University. Contact her at rosa.martey@colostate.edu.

**Benjamin Clegg** is with Colorado State University. Contact him at benjamin.clegg@colostate.edu.

**Carsten Østerlund** is with Syracuse University. Contact him at costerlu@syr.edu.

**Lael Schooler** is with Syracuse University. Contact him at lschoole@syr.edu.