Beyond Requirements: Improving Software Tools for Intelligence Analysts

Sarah Geitz, SAIC; *Organizer and Chair*
Brian Moon, Klein Associates; *Co-Chair and Panelist*
Anita D'Amico, Secure Decisions; *Panelist*
Robert R. Hoffman, Institute for Human and Machine Cognition; *Panelist*
Rob Page, Zope Corporation; *Panelist*

Abstract

The goal of this panel is to discuss critical human factors concerns in the development of software for intelligence analysts. The panel presentations are designed to provide a high level overview of the software development process, the intelligence analysis process, and the challenges encountered in both obtaining user feedback. Presentations will examine a variety of issues including the analysis of imagery, text, information assurance, data fusion, visualization models and in establishing situational awareness, as well as empowering analysts with open source software.

The panel discussion will focus on extracting generic processes that can be applied to obtaining more accurate software metrics, requirements and solutions from a world where certain topics cannot be discussed. Methods and metaphors for better describing to individuals working outside of the classified world the context a within which a tool will be used, may be touched upon, as well as identifying ways of overcoming both internal and external politics. Human factors concerns may also be addressed, such as evaluating how trust affects the feedback received from individual analysts and communication and interaction within and between groups of analysts. Identifying and overcoming potential perceptual problems in the software development process will also be discussed.

The anticipated outcome of the panel will be to target individual processes, techniques and technologies that can be applied to obtaining requirements to support cognitive processes, which can in turn be applied developing software tools that better fit the needs intelligence analysts.

Introduction

The creation of software tools to aid intelligence analysts is an iterative process. Observing and interviewing users, collecting requirements, identifying data sources, creating user interaction and visualization models and insuring technical solutions are scheduled to fit timeline and budget constraints are challenges faced in all software development. However, software development for intelligence analysis differs from developing commercial software in several major respects.

First, users are rarely the consumer. Software contract procurement decisions are made at the management, agency or command level, not by the analyst/users. The analyst/users may be asked for their opinions but decisions on what is purchased or developed is rarely theirs.

Second, the classified nature of intelligence analysis does not always allow developers to fully understand the context their products and tools will be used within. Thus products and tools can be developed under inaccurate premises, potentially endangering national security.

Third, working with classified materials often means working on multiple networks. Transferring information between networks can present cognitive, technical and security problems not existing in the outside world.

Finally, political constraints are placed upon analysts from within and between agencies and employers. Even though analysts rarely have veto power, they may become victims of corporate intrigue. Ultimately, internal and external politics can make it difficult to obtain accurate feedback regarding the metrics of success or failure in software products.

Software tools for intelligence analysts are built to support the analytic process. The tools are generally developed outside the world of the analyst, sometimes using a list of requirements and sometimes not. These requirements are often derived by individuals working within the classified world with no background in human factors, cognitive sciences, visual or user interface design. Individual analytic approaches and conclusions vary with subject matter expertise, knowledge of source, experience, motivation, access to information, tools used and organizational processes and procedures. The analyst's product is ultimately inference driven and therefore difficult to quantify. Each step along the way the recurrent human factors of trust, decision making, semantics and data overload emerge. The accuracy of the requirements derived from within the classified environment can be a determining factor in the success or failure of a tool. Obtaining accurate feedback of the true impact of the proposed intervention is essential.

Panel Overview

The panelists discuss how these issues apply to the areas of interest:

Anita D'Amico, Secure Decisions

Secure Decisions has been developing visualization software for Information Assurance (IA) analysis, primarily for the DOD and intelligence community, for six years. The IA domain is related to Intelligence Analysis, but is distinct from it. IA analysts protect electronic information from unwanted access or unwanted distribution. Such activities include monitoring network traffic, analyzing traffic for suspicious or unexpected behavior, and predicting future attacks. A particular type of IA analyst, referred to as a threat analyst, does perform intelligence analysis functions in order to find potential network attackers, identify the attacker's modus operandi, and predict his next actions.

There are, of course, differences between the domains. IA analysts have the benefit, in some cases, of direct feedback from their actions: if they succeed in detecting an attack, they have tangible evidence of that success. The dynamics of the IA space are also somewhat more hectic than in the intelligence domain: things happen, at times, as frequently as minute-by- minute. Nevertheless, with these differences in mind we can apply the lessons of one domain to the other.

The most significant lesson learned from our experience with IA that applies directly to the success of software in the intelligence analysis domain is the issue of successful transition of software into operational use. The most important metric for software is how it transitions from the laboratory into operational use; without successful transition, the software development effort is wasted. Under a recent contract to DARPA, Secure Decisions has identified the key characteristics of IA situational awareness software that makes a software system a good candidate for operational transition. Among these are:

Operational Need – More than just solving a problem or two for the analyst, the software must address a key operational need for the organization.

Suitability to Task – This is the basic characteristic of addressing the needs of the analyst: the focus of most requirements analysis efforts. However, it is important that the software not only solve the analyst's problems, but does so in a manner consistent with how those problems fit into the analyst's work. Proper CTA analysis can help to identify this.

Disruptiveness – The software cannot have an overly disruptive effect on current operations.

Task Focus – The software must maintain a strong focus on its key mission, and not attempt to solve every problem the analyst might have in a single package.

Usability – This does not refer strictly to the user-friendliness of the HCI, although that is always important, but to the broader issue of whether the software is usable to the analyst *and* to the organization within which the analyst works. Added to the traditional user-centric usability metrics of effectiveness, efficiency, and satisfaction are the operational metrics of cost, equipment and facility requirements, support requirements, training needs, operational staffing, and much more.

Adaptability – The software must be capable of adapting to changes in the operational context, to avoid becoming shelfware as things change.

Functional Stability and Robustness – The software must do what it does well, without failing or acting abnormally, and must not change constantly.

Scalability – The software must keep up with changes in the quantity and rates of information presented to it.

Transition and Post-Deployment Support – All of the support issues, including Documentation, Training, help desk support and more, will have a significant impact on how successfully analysts will adopt the technology.

Robert R. Hoffman, Institute for Human and Machine Cognition

Making Good Decisions about Decision-Aiding

One cannot help but be struck by the fact that many current scholarly books on judgment and decision making do not include any straight-out attempts at defining this thing called "decision" (Arkes & Hammond, 1986; Juslin & Montgomery, 1999; Kahneman & Tversky, 2000; Plous, 2993; Smith, Shanteau, & Johnson, 2004; Sternberg, & Frensch, 1991). Only in one edited volume (Svenson & Maule, 1993) do we find the word "decision" as an entry in the subject index. Only in two do we find a discussion of the difference between "decision" and judgment" (Hammond, McClelland, & Mumpower, 1980; Yates, 2003).

A decision is generally understood as a mental event that is a singular point or circumscribed span of time—a "moment of choice"—that leads immediately or directly to some form of commitment to action (e.g., push the button). These basic elements of the received notion of "decision" include three stage-like elements: 1. Entry point for information, 2. Span of apprehension, and 3. Commitment to actions or action sequences. This scheme has been the received view in the literature on the psychology of judgment and decision making, the literature on economic judgment and decision-making, the literature on medical decision making, the literature on geographical decision making, to name just a few. This scheme and elaborations of it can be found lying at the core of a great many descriptive and computational models of decision making (cf. Wright & Bolger, 1992), including models that are being used to inform the creation of decision aids for the intelligence community (Moon & Hoffman, 2005).

The finalistic point notion—allowing us to say that decisions are things that are "made"—is a form of deciding that has been taken to be the prototype, but may be the exception rather than the rule. While deciding involves entry points for information, the entry points may be other decisions, or mental events that themselves have an element of deciding. While the traditional view regards decisions as culminations following individual acts of judgment, they are, in fact, expressions of contingencies.

> Decisions are assumed to be discrete events, clearly distinguishable from other group activities… Decision makers often can identify discrete decision points and feel a sense of completion at making a decision. These boundaries are not always as clear as they seem at first, however, and there is not always agreement on what events are involved in a given decision. Definitions of decision-making episodes are ambiguous in several respects (Hirokawa and Poole, 1996, p. 9).

A decider's decision processes derive from a host of "deep" contributors that are important to understand in their own right, and are largely neglected in judgment and decision-making research and notions of what decision-aiding is all about. Examples of the ten "cardinal issues" (Hoffman and Yates, 2005; Yates, 2003) are:

Mode: *Who (or what) will make this decision, and how will they approach that task?*
Investment: *What kinds and amounts of resources will be invested in the process of making this decision?*
Tradeoffs: *All of our prospective actions have both strengths and weaknesses. So how should we make the tradeoffs that are required to settle on the action we will actually pursue?*
Judgment: *Which of the things that they care about actually* would *happen if we took that action?*
Acceptability: *How can we get the other stakeholders to agree to this decision and this decision procedure?*

Research has resulted in numerous useful conclusions about decision making expertise (Yates, 2006), yet there are also enormous shortcomings in current understanding. At one level, this is disappointing. But, in our view, the component analysis afforded by the cardinal decision issue perspective has been fruitful in its ability to identify specific questions that need to and can be answered. The next step is actually finding the answers.

Current tools for aiding in the intelligence analysis process are, on the surface of it, motivated by Herbert Simon's notion of bounded rationality. However, they primarily leverage the literature of biases and human limitations, and rely on models of dozens of bias to create tools that are intended to mitigate bias. In fact, the core notion is one of unbounded irrationality rather than bounded rationality (Hoffman, 2005), and it is not clear from the cognitive task analysis work conducted to date that this is really what intelligence analysts actually need. In fact, the tools that have been created in recent years create make work for the analyst, they turn the analyst into a probability juggler, they have graphics displays that are colorful and fancy but that are largely devoid of directly-perceivable meanings, and they offer limited value-added from the analyst's perspective.

Those who would create intelligent decision architectures might benefit from considering the macrocognitive view of deciding (Klein, et al., 2003), one that is significantly richer than the received view of Entry Point→Apprehension→Commitment to Action. The macrocognition view would see deciding as a distributed activity that is parallel and highly interactive with other macrocognitive processes, rather than seeing decisions as things that are made. The macrocognitive view would advocate the creation of what are called Janus systems (Hoffman, Lintern, & Eitelman, 2004), which are systems that merge training and performance support. These kinds of approaches may lead to more useful decision aids and other forms of intelligent systems.

Brian Moon, Klein Associates

More than just the 'human factors guy': Playing Roles in a Multi-Dimensional Assessment of a Horizontal Integration Solution

In March of 2005, the United States Joint Forces Command conducted a joint utility assessment of a proposed horizontal integration solution presented by the United States Army's Intelligence and Security Command. The assessment took place across multiple sites – in South Korea and Hampton Roads, Virginia – and included the participation of a number of intelligence community member agencies. The task of organizing and executing an overarching assessment plan that got at, among other things, the operational and analytic impacts of the proposed solution was no small order. Complicating it all was an incredibly short timeframe – just over four months from project initiation to final report.

My comments will reflect my experience as a member of the USJFCOM assessment team. Initially brought in as the "human factors guy," my role was quickly extended when it became apparent that an overarching assessment plan was necessary. I shall speak to the various roles that I played, the 'politics' of the assessment, the products I produced, and how such assessments fall outside the traditional 'requirements-based development process." Most importantly, I shall also comment on the human factors role I eventually did play and the data I discovered under challenging collection and analysis conditions. I suggest that, given the common backgrounds of most human factors professionals, the roles necessary to conduct large-scale assessments might rightly be filled by the human factors community.

Rob Page, Zope Corporation

The open source development methodology is *not* most importantly about free software. It is fundamentally about evolving a software development and business paradigm that eliminates vendor lock-in and islands of innovation that result.

Most software is built on top of a development platform. Examples include .NET, J2EE, LAMP, and Zope. These platforms allow software developers to benefit from myriad infrastructure services, components and related engineering aids (e.g., testing frameworks) when they develop applications.

The adoption of open source platforms provides the intelligence community with platform-, organization- and requirements-portable foundations on which sustainable community-wide collaboration, innovation and development occur.

Collaboration within communities-of-interest is not a new idea. The ICML working group is an example of a community of interest that collaboratively developed a product – in this case an XML specification. Given the maturity of open source software platforms, their reliability and substantial deployment, they are the clear and obvious choice for sustainable, directed innovation in the software tools needed by Intelligence Analysts.

References

Arkes, H. R., & Hammond, K. R. (Eds.) (1986). *Judgment and decision making: An interdisciplinary reader*. Cambridge: Cambridge University Press.

Hammond, K. R., McClelland, G. H., & Mumpower, J. (1980). *Human judgment and decision making: theories, methods, and procedures.* New York: Praeger.

Hirokawa, R. Y., & Poole, M. S. (Eds.) (1996). *Communication and group decision making*. Thousand Oaks, CA: Sage Publishing.

Hoffman, R. R. (2005). "*The Psychology of Intelligence Analysis* Revisited: An update from cognitive psychology post-1980." Report, Institute for Human and Machine Cognition. Pensacola, FL.

Hoffman, R. R., Lintern, G., & Eitelman, S. (March/April 2004). The Janus Principle. *IEEE: Intelligent Systems*, pp. 78-80.

Hoffman, R. R., & Yates, J. F. (2005). Decision(?)-Making(?). *IEEE Intelligent Systems*, in press.

Juslin, P., & Montgomery, H. (1999). *Judgment and decision making*. Mahwah, NJ: Erlbaum.