Participatory Design Methods for Command and Control Systems

Track: C2 Analysis

Dr. Janet E. Miller
Air Force Research Laboratory/HECS
Wright-Patterson AFB, OH 45433
Voice: 937-656-4401; Fax: 937-255-6555
Janet.Miller3@wpafb.af.mil

Participatory Design Methods for Command and Control Systems

Dr. Janet E. Miller
AFRL/HECS
Wright-Patterson AFB, OH 45433
Voice: 937-656-4401; Fax: 937-255-6555
Janet.Miller3@wpafb.af.mil

ABSTRACT

To ensure network-centric and other systems provide relevant capability to the C2 user, effort needs to be expended in understanding the requirements, designing the appropriate solution, developing the capability and implementing for acceptance. The cognitive systems engineering and the software systems engineering communities struggle with the difficulties of understanding a domain and its challenges and then handing research results to designers and developers so that shared understanding of the problem and possible solutions exists. They also struggle with the more frustrating challenge of a developed system being implemented but not enthusiastically embraced by the end-user. Participatory Design (PD) has a goal of engaging researchers, designers, developers, practitioners and end-users in all of the various activities leading to the successful development and implementation of systems. PD is an umbrella methodology which includes studies, theories, conferences and practices (Muller and Kuhn, 1993; Kensing and Blomberg, 1998; Madsen, 1999). This paper will discuss two methods which embrace participatory concepts. The first is Elicitation by Critiquing (EBC) and the second is Value Elicitation. Both techniques provide a way to engage all of the stakeholders at the requirements discovery stage of development, which is the first critical step of system development.

Introduction

Cognitive systems engineers (CSE) and software systems engineers (SSE) endeavor to develop useful, usable systems to support cognitive work such as is accomplished in a C2 network-centric environment. Cognitive task analysis (CTA) methods are used to discover expertise that domain practitioners utilize to perform tasks so that better support, such as automation or training, for these cognitive activities can be developed. Specifically, CTA's identify ineffective strategies that lead to poor performance (i.e., a model of mistakes that "novices" make), as well as adaptive strategies that have been developed by highly skilled practitioners to cope with task demands (i.e., a model of "expert" skill). The complexity of understanding the environment and the tasks, combined with the fact that experts performing cognitive tasks have difficulty reliably articulating about the task when asked, contribute to making discovering expertise hard. There is a myriad of other challenges which is why a variety of cognitive task analysis methodologies exist (Schraagen *et al.*, 2000). However, many of these methods are skeptically viewed by a domain's practitioners as they perceive an outsider cannot fully understand their particular challenges.

In addition, despite having established methods of gaining understanding about a domain, the cognitive systems engineering community struggles with the difficulties of handing the research results to software designers and also of handing designs to developers so that shared understanding of the problem and possible solutions exists. Software systems engineers (SSE) are also busy trying to support user's needs but come from a somewhat different angle as CSE focuses on the cognitive aspects as revealed by cognitive and domain explorations while SSE focuses on issues such as data and technology requirements and concerns itself less with the cognitive challenges that the software may present. While software systems are becoming increasingly critical and vital to our existence, as evidenced by the trend toward net-centric environments, these two communities that aim to support the user's needs often still miss the mark. The frustration culminates when a developed system is implemented but is not enthusiastically embraced by the end-user.

Participatory Design (PD) is an established, diverse research and practice area which has a goal of engaging researchers, designers, developers, practitioners and end-users in all of the various activities leading to the successful development and implementation of systems. It is an approach to the assessment, design, and development of technological and organizational systems that places a premium on the active involvement of workplace practitioners (usually potential or current users of the system) in all of the steps in the research, design and decision-making processes. PD is an umbrella methodology which includes studies, theories, conferences and practices (Muller and Kuhn, 1993; Kensing and Blomberg, 1998; Madsen, 1999). When utilized, PD can help address the above issues that the CSE and SSE communities face when solving difficult problems as the entire set of stakeholders' viewpoints are brought in from the beginning. Elicitation by Critiquing (EBC) and Value Elicitation are two participatory methods that provide ways of engaging the stakeholders in the requirements discovery stage of development which is the first critical step of system development.

Elicitation by Critiquing

Elicitation by Critiquing (EBC) is a methodology that engages subject matter experts themselves as the ones to reveal strategies of another domain practitioner to the CSE, or SSE, investigator. First, one domain practitioner completes a task and then that completed task is presented to yet another domain practitioner for critiquing (Figure 1). The CSE takes note of the critiques to gain and document understanding of the domain for later reflection by the domain expert.

EBC introduces different roles for the CSE practitioner and the expert than other cognitive task analysis methods (Figure 2). For example, with direct observation, the CSE watches the expert as a domain practitioner performing actual or simulated tasks. The value of the results for direct observation is mainly a function of how realistic the scenarios and performances are and how well the observed events stress the cognitive system in order to reveal leverage points for improvement. With interviewing, the expert tells information to the CSE investigator, often stories based on past cases and experiences. While being questioned by the CSE, the expert as storyteller might reveal not only how he handled a particular case, but also may have further comments on how that case changed his later work practices. The value of the results from interviews is mainly a function of the probing skill of the investigator and how well the interviewee understands what type of data is sought. With critiquing, however, the domain expert

evaluates the performance of another domain practitioner thereby participating in the process of revealing knowledge of the domain's practices and strategies. This has the added benefit of setting up a situation where the domain expert is not concerned about his performance evaluation. In addition, the technique relies less on the investigator's probing skills and domain knowledge. The value of the results from EBC is mainly from the participation of the domain practitioner. EBC also has the benefit of combining many of the advantages from performance simulations with the practical advantages of interviewing techniques.

A study was performed to determine the usefulness of EBC (Miller *et al*, 2005). The goal of the elicitation was to identify military intelligence analysts' strategies and challenges encountered in doing their work. The method involved two stages. First, an actual trainee performed a typical domain task in order to capture his performance process. Then, a representation (transcript, screen shots and note-taking artifact) of the trainee's performance on the task was presented to six experienced domain practitioners for critiquing. The experienced practitioners' comments on the trainee's performance were captured and iteratively analyzed for patterns. To investigate how much the critiquer's comments were influenced by the performance of the trainee, a second trainee performed the same task and was critiqued in the same manner by two of the original expert critiquers.

This initial investigation into critiquing suggested that CTA challenges of gaining access to tasks and experts, efficiency, and repeatability can be addressed with critiquing. Task accessibility issues were addressed as once each trainee's process was captured, the re-created performances which consisted of the transcript, the screen shots of the querying capability and slides of the note artifact, were used multiple times. They were used six times for the first trainee's process and twice for the second trainee's process. Hence, as the task was packaged, it was accessible for multiple sessions. Physical access to experts was overcome because the critiquing sessions were held outside of normal work areas, which have security access issues. Additional access challenges because of the existence of only a few experts could be overcome with this method if several trainees or novices were used to create different presentations on the same or different problems and the few available experts critiqued the different trainee or novice processes, as was done with two of the experts critiquing both trainees. Another access issue that arises in some domains is that experts are reluctant to participate due to repercussions, such as erroneously performing an act while being observed that could have dire consequences. Those performing command and control functions fall within this area of sensitivity. For Elicitation by Critiquing, as the expert is in a role other than performing work, he may see the elicitation as less of a threat and therefore be more willing to participate.

While the above listed CTA challenges are addressed by EBC, challenges still exist in fully understanding certain types of problems in domains. For example, repetitive acts or processes, such as building an air tasking order, can be investigated with EBC but uncovering what instigates an 'aha moment' for creative problem solving, such as a commander determining an innovative defense strategy, is still not simple. For EBC, such a moment would have to be created, and, hence, what made that moment would already have to be understood. In addition, if the domain is dynamic so that new methods are constantly evolving, such as in basic research, there will still remain some mystery to the domain. In these cases, another participatory method,

value elicitation, can be used to help identify requirements for what might be helpful especially in terms of training and software tools.

Value Elicitation (VE)

Value Elicitation (VE) centers around value focused thinking (VFT), a proven decision analysis methodology that can be applied to a variety of multi-criteria situations (Kirkwood, 1997). The methodology concentrates on eliciting the values of the stakeholders at the core of a decision before any particular solution to the decision is considered. As an objective methodology, VFT is well suited for disclosing and handling multiple, competing requirements of the stakeholders, such as those encountered in interface design for C2 and other network-centric systems, and provides an unbiased evaluation artifact. The primary benefit that VFT provides is its ability to convert the goals of a project or values of an organization into an objective realm and its structure lends itself to handling multi-objective problems even if the objectives are of a subjective nature. Using VFT, high level objectives are broken down into smaller values during facilitated sessions with the stakeholders. Once articulated, the values can be measured and put to a common scale, allowing their contribution to the overall objective to be evaluated. By assigning quantifiable measurements to the components, the multi-objective goal can be evaluated. Once developed the hierarchy can be consistently and constantly applied to different system developments and can allow a fair comparison between potential software solutions.

During discussions and value elicitations with the stakeholders, the values and measures are developed and placed into a hierarchical structure. They are then weighted by the decision-makers. The weighing process allows the general process to be customized to the particular instantiation. The interface, or other object, to be evaluated is then scored and ranked using an additive value function, producing a measure of derived directly from the decision-maker's values. The additive function, $v(x) = \sum\lambda_i v_i(x_i)$ for all i measures, associated with VFT methodology, is used and mutual preferential independence is assumed (Kirkwood, 1997).

As VFT concentrates on determining the values at the core of the decision, the choice is not between varieties of alternatives—each with its own benefits and drawbacks—but rather between a selection of the alternatives that give the greatest utility with regard to what has been determined valuable. VFT emphasizes that values should be the focus for making a good decision. However, most people try to look at all the alternatives and compare them against each other without defining a similar basis. This presents difficulty if one alternative is extremely better at one aspect of the decision while the other alternative is extremely better at another aspect. This type of decision is called a multi-objective decision, where multiple objectives are desired in the decision. VFT provides a structure to compare these objectives against each other based on the decision-maker's values. VFT, however, takes more time and requires the decision-maker to give his mind to the exercise, but the benefit of this structure makes it worth the effort. (Keeney, 1992) In addition, as it is participatory, the stakeholders are the ones driving the development of the hierarchy.

In previous work, McGee (2003) developed a VFT hierarchy (Figure 3) during discussions with military intelligence analyst participants. This hierarchy was then used to evaluate a system

which was still under development for that military intelligence analyst community. As expected, when this hierarchy was applied to the newly developed system, the system under evaluation did not fair well, scoring only 0.13 out of a possible 1.0 with the analysts agreeing with this scoring. One relevant outcome of this exercise was that the evaluation revealed user concepts and ideas that the developers had not even entertained.

That work, presented at the 2004 Command and Control Research and Technology Symposium, on VFT (Miller, 2004) has been extended by applying the hierarchy to an additional system for evaluation. The VFT hierarchy, which has been put into an Excel spreadsheet for computational ease, was applied to a text exploitation capability being developed for the same intelligence analyst community. The program manager for the system development had become sensitive to human factors issues once the foundations of the technology were well underway and was interested in knowing how the tool's interface would fare. She had worked closely with the intended user community during the majority of the development to avoid erroneous assumptions and wanted to verify that she was on target. When the capability was scored by the intended user community using McGee's VFT model, a value of .588 was derived. While this showed need for further usability development, the text exploitation tool was still in early enough stages of development to make adjustments. The weak areas which the results highlighted were not totally surprising as the program manager had held meetings where some of the tool's interface shortfalls had been discussed. However, this score raised questions on the limited value of some aspects of the existing hierarchy. Discussions with the military analysts indicated agreement that there was more refinement to be done on the hierarchy. To address the issues, the hierarchy concept was further developed to include what could be considered a 'gold standard' (Figure 4) using information from The Windows Interface Guidelines for Software Design. A full, weighted hierarchy was not developed, but discussions gave the group understanding on how the values related. For example, in the complete, original hierarchy for this investigation, the "Output Presentation" under sub-category "Aesthetics" is "Readability" and was broken into "Ease of reading color" and "Ease of reading fonts" (Table 1). In the 'gold standard', the concept of "Professional Look" is included which the group agreed added another dimension. The capability was then rescored with the augmented information and received a higher score, this time a .658, which along with the discussions involved with deriving the score, helped the program manager better focus in on requirements for improvement.

The above usages of VE indicated that there is value in using the hierarchy to gain understanding of a domain and its requirements. Nevertheless, it is understood that any model must necessarily be based on some type of particular problem set which causes a natural bounding of the resulting model and those boundary conditions must be appreciated when applying the model. This does not negate the value of this multi-attribute theory-based model for C2 or other domains but rather the method can be added to the tricks of the CSE and SSE trades for identifying requirements for development of tools that fully support the human's role to prevent human-error-to-blame mishaps.

Discussion

While software systems are becoming increasingly critical, complicated and vital, the two communities that aim to support the user's needs struggle with the difficulties of understanding

and articulating about domains and their challenges so that excellent systems are implemented. CSE focuses on the cognitive aspects as revealed by cognitive and domain explorations. SSE focuses on issues such as data and technology requirements and concerns itself less with the cognitive challenges that the software may present. A well-accomplished CTA, a critical step in system development, indeed reveals critical sources of cognitive complexity that must be addressed in new systems, but the hand-off to the software designers and developers is often in the form of staid documents and reports. While both disciplines are necessary to create systems, the partnering of the two has been irregular and awkward leaving a gap. In addition, the domain practitioners are often not brought into the decision making process during system development. As a result, both communities and the domain practitioners experience frustration when an implemented system misses the mark either in its capabilities or its ability to be embraced for other reasons.

Domain practitioners want systems that address their needs as much as the CSE and SSE communities want to develop such systems. PD is an established category of methods which aim to inform and engage all stakeholders in the process of system development. Involving the expected users and various stakeholders during requirement definition, as is advocated by PD, is important for developing useful, usable systems which gain acceptance when implemented. Both EBC and Value Elicitation leverage the knowledge and understanding of potential users of a system by having them participate in understanding the requirements for support before a system is built. These methods contribute to the pool of potential PD methods and should help the CSE and SSE communities work together for the benefit of the domain practitioners to overcome the gaps between task analysis, design and development which are arise in building C2 and other systems.

One challenge area concerns using PD for the development of future-focused systems, that is, systems which are to be developed not for today's practitioners' particular problems, but for anticipated problems of the future. The United States' Air Force Research Laboratory looks to address today's warfighter's problems, but also considers what will be the challenges to the warfighter in 2015 and beyond. While there are some in the operational community who may be forward thinking, the majority is intensely tied to the problems at hand and so getting a practitioner to think a decade into the future on what the work may be like let alone what challenges that work might bring is difficult. Both EBC and VE seem to be likely candidates for this research area. For each, a future conflict scenario would be the framing for the task performance or elicitation. For EBC, the domain practitioner would reflect on how he would address a facet of the scenario and that could be captured and presented to gain other's comments. Questions for using VE would help the practitioner extrapolate current practices to the future and help guide the development of values for the future-focused system. However, this is an area for future research.

Conclusion

To ensure network-centric and other systems provide relevant capability to the C2 user, effort needs to be expended in understanding the requirements, designing the appropriate solution, developing the capability and implementing for acceptance. Using Participatory Design (PD), which has a goal of engaging researchers, designers, developers, practitioners and end-users in

all of the various activities leading to the successful development and implementation of systems, can support reaching those goals. EBC and VE are two methods which can be added to the list of PD techniques as both techniques provide a way to engage all of the stakeholders at the requirements discovery stage of development, which is the first critical step of C2 or any system development.

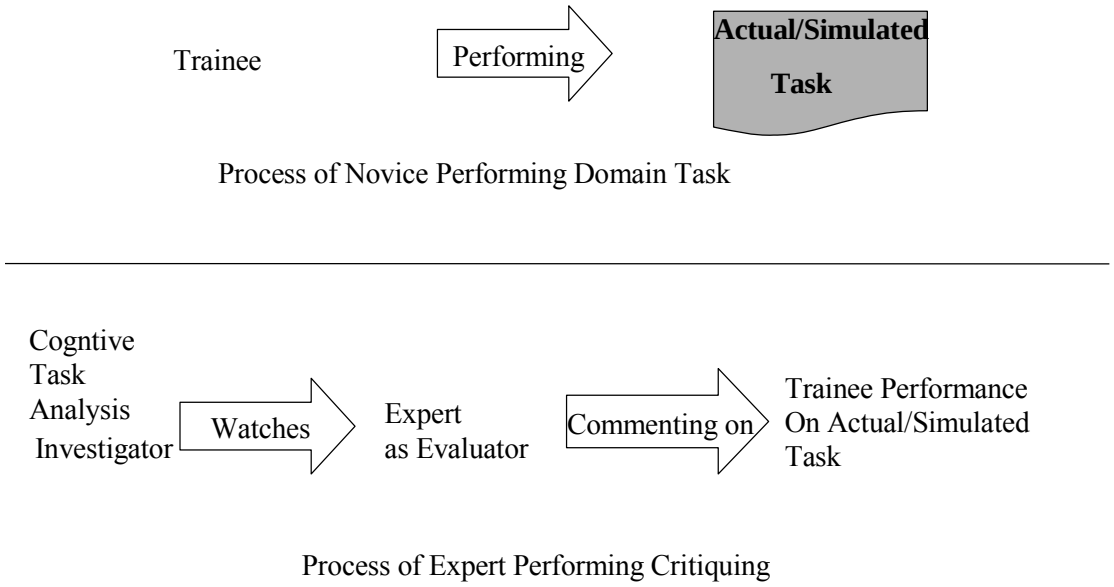

Figure 1 Two Processes of Elicitation by Critiquing

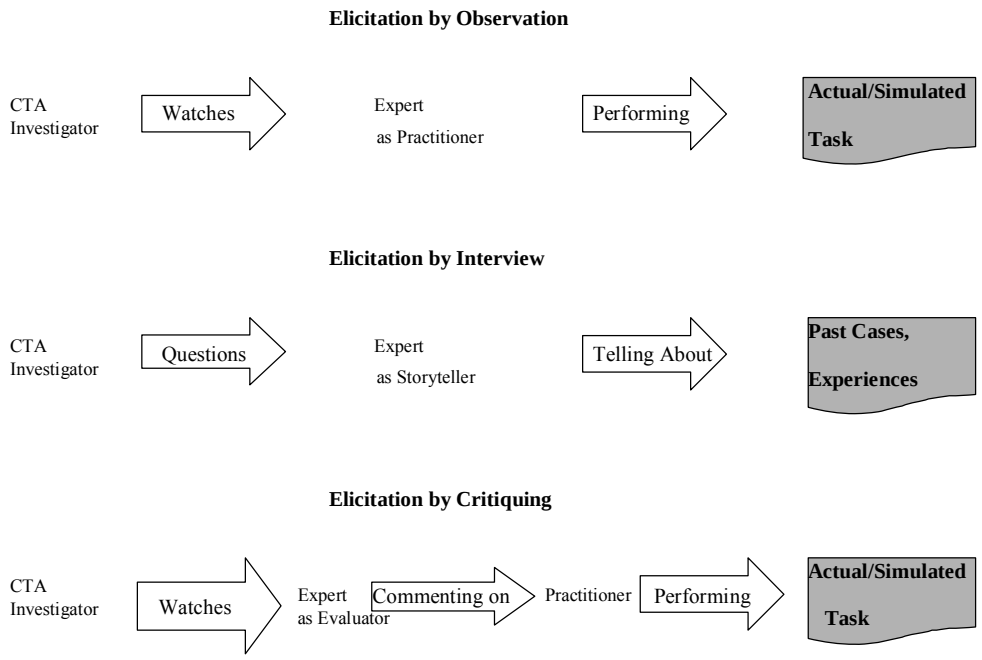

Figure 2. Comparison of Cognitive Task Analysis (CTA) Methods

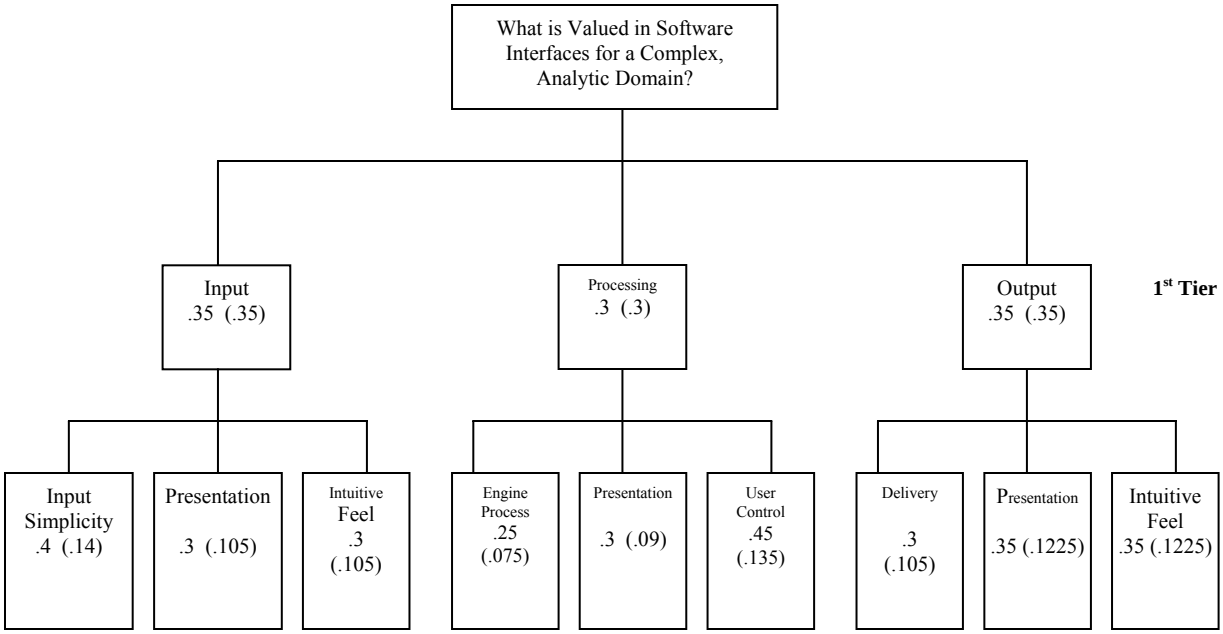

Figure 3. Value Hierarchy from McGee (2003)

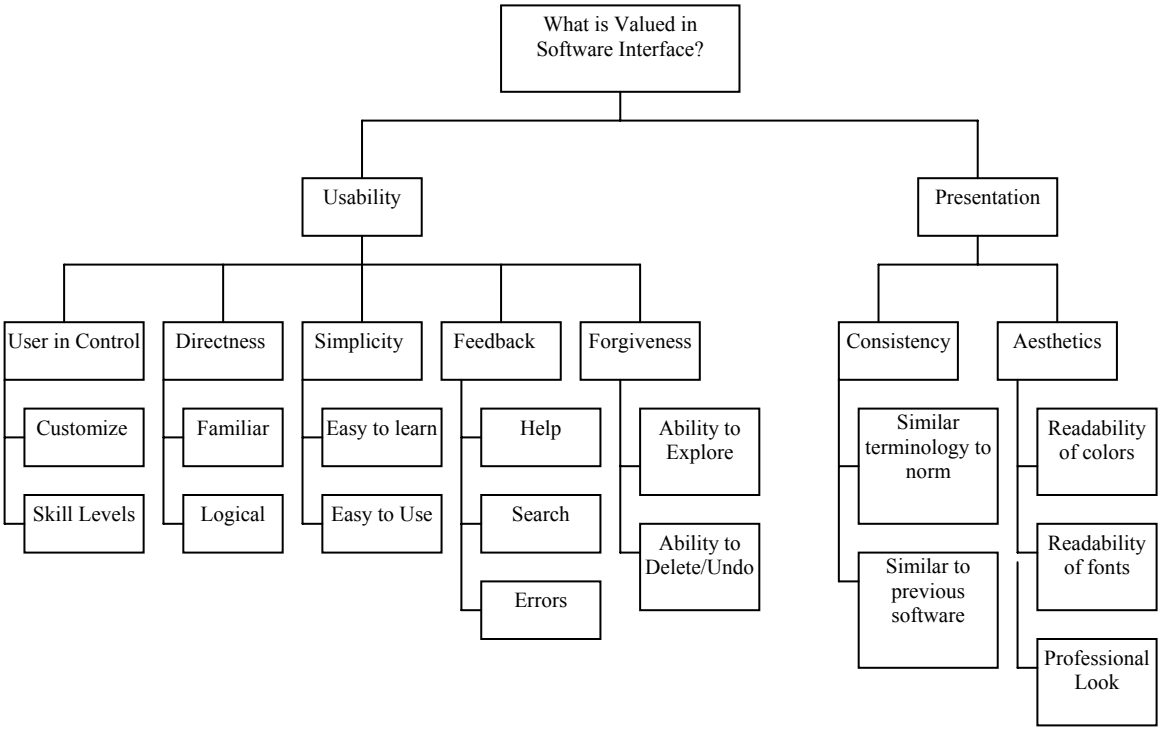

Figure 4. Gold Standard for Software Interfaces

Output Presentation	Defined as the ability of the interface to display the data in way that makes it easier for the user to present the data to the customer.
Aesthetics	To present the data in a way that is pleasing to the eye to the customer or end user **PLEASE NOTE: This value corresponds with the literature found in <u>The Windows Interface Guidelines for Software Design.</u>**
Readability	The ability to present the data in a way that enables the customer to read and process data with ease. This may include the colors, fonts, format and overall look of the interface.
Attention	To allow the user to inform the customer of aspects of the processing and analysis that may be of importance to the customer, e.g.. where the output is, highlight important data
Feedback	The ability to provide information that would aid the user and give more information where needed, e.g. a help option
Flexibility	Defined as the ability of the interface to be adapted by the user to enhance the appeal and efficiency for the customer.
Customize	To be modified by the user so that the look and feel of the software allows them to output data in an easier, more comfortable fashion for the customer, e.g.. to be able to change the visual options of the interface.

Table 1. Defintions for Output Presentation Values

DISCLAIMER

The views expressed in this paper are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense or the U. S. Government.

REFERENCES

Keeney, Ralph L.(1992) *Value-Focused Thinking: A Path to Creative Decision-making*. Cambridge, MA: Harvard University Press, 1992.

Keeney, Ralph L. (1994) "Creativity in Decision Making with Value-Focused Thinking," *Sloan Management Review*, 35: 33-41 (Summer 1994)

Kirkwood, Craig W. (1997) *Strategic Decision Making, Multiobjective Decision Analysis with Spreadsheets*. Belmont: Wadsworth Publishing Company, 1997.

Kensing, F., and Blomberg, J. (Eds.) (1998) *Computer Supported Cooperative Work* special issue on participatory design, 7 (3-4)

Madsen, K. H. (1999) *Communications of the ACM* special issue on usability in Scandinavia and the US. 42(5), May 1999

McGee, Christopher M. (2003) "A Value Focused Thinking Approach to Software Interface in a Complex Analytical Domain." MS Thesis. AFIT/GOR/ENS/03-16, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. March 2003.

Microsoft Press. *The Windows Interface Guidelines for Software Designs*. Redmon WA: Microsoft Press, 1995

Miller, Janet E. (2004) "Value Focused Thinking: Guiding C3 System Development", 2004 Command and Control Research and Technology Symposium, San Diego, CA 15-17 June 2004

Miller, Janet E., Patterson, E. S. and Woods, D. D. (2005) "Elicitation by Critiquing as a Cognitive Task Analysis Methodology" *Cognition, Technology and Work,* 15 December 2005, Online First

Muller, M. J. and Kuhn, S. (1993) *Communications of the ACM* special issue on participatory design 36 (6), June 1993

Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.), (2000), *Cognitive Task Analysis*. Mahwah, NJ: Lawrence Erlbaum Associates

AUTHOR BIOGRAPHY

Dr. JANET E. MILLER is a Research Scientist with an emphasis in decision sciences at the Air Force Research Laboratory, Human Effectiveness Directorate, Wright Patterson AFB, Dayton, OH. Her experience includes gathering user requirements and implementing automated systems at the Pacific Air Forces and the National Air and Space Intelligence Center. Her degrees include a BS in Electrical Engineering from Louisiana Technical University (1984), an MS in Management of Information Systems from the University of Arizona (1988) and a Ph. D. in Cognitive Systems Engineering (2002) from the Ohio State University. Additionally, she is holds the rank of Lieutenant Colonel in the US Air Force Reserves (Active).