Value Focused Thinking: Guiding C2 System Interface Design

Dr. Janet E. Miller
Air Force Research Laboratory/HECS
2255 H Street
Wright-Patterson AFB, OH 45433
Voice: 937-656-4847; Fax: 937-255-9198
Janet.Miller3@wpafb.af.mil

Value Focused Thinking: Guiding C2 System Interface Design

Dr. Janet E. Miller
AFRL/HECS
2255 H Street
Wright-Patterson AFB, OH 45433
Voice: 937-656-4847; Fax: 937-255-9198
Janet.Miller3@wpafb.af.mil

ABSTRACT

Interfaces, from displays for command and control to screens for military intelligence analysts, are the gateways to the underlying software technology. Despite how state-of-the-art and sophisticated the underlying algorithms are, if the interface does not address the user's critical functions, the software capabilities will be underutilized. Therefore, design of the interface is critical to ensure the user has the right information displayed at the right time in the right way. Meeting this goal within project constraints is a challenge. Therefore, a framework is needed to guide development so that resources can be focused on the most relevant aspects of the interface development. Value Focused Thinking (VFT) provides an objective methodology that is well suited for handling multi-objective problems such as interface design. In this paper, the VFT methodology will be described and a specific VFT hierarchy that was developed with the Air Force Research Laboratory will be explained. Discussions show that VFT can be more generally applied to gain understanding for Command and Control (C2)-specific software development and that the described research can be used for guiding the development of software tools for C2 functions.

Introduction

Interfaces are the gateways to the underpinnings of any software. This is especially true in the fast-paced, extremely dynamic world of a commander and his staff on the battlefield. Despite how state-of-the-art and sophisticated the algorithms underlying a capability are, if the interface does not address the functions or show the information that the commander considers vital, the software will not be used to its fullest capacity. Evaluating the usability of the interface is necessary but not sufficient in determining the success of the interface. Rather the usability *and* the usefulness must be evaluated. Many methods exist that try to tickle out a user's requirements so that the requirements can be fed into the development process. Many other methods exist for evaluating how well an interface satisfies a user. However, an objective methodology that is well suited for disclosing and addressing multiple, competing requirements, such as those encountered in interface design for Command and Control (C2) systems, and that provides an unbiased evaluation artifact would be helpful. Value Focused Thinking (VFT) provides the type of objective methodology that is well suited for handling such multi-objective problems.

Background

Value-focused thinking (VFT) (Keeney, 1992, 1994) is a methodology that can help identify what is needed in an interface for a particular application and can be used to compare different potential interface solutions or can be used to judge how well an interface currently meets the customer's needs. The methodology provides a means to reveal and address the multiple objectives of an interface design effort. Considering that all development efforts have resource constraints, such a methodology would help drive a project in the right direction. Value focused thinking (VFT) is a proven decision analysis methodology that can be applied to a variety of multi-criteria situations (Kirkwood, 1997). The primary benefit that VFT provides is its ability to identify and convert the goals of a project or values of an organization into an objective realm. 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. 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.

The VFT methodology concentrates on determining the values at the core of a decision. Keeney (1992) writes that, "Values are principles used for evaluation. We use them to evaluate the actual or potential consequences of action and inaction, of proposed alternatives, and of decisions." To think of these values in a decision process, the decision must have the following properties: the decision should be a real problem, it should be of great importance, and it should be complex and have no absolute solution. The decision maker should be able to answer the "why is this important" test. If the decision has no real importance the input to the decision will not carry the necessary relevance to make a true decision. The question that surfaces at this point is how to determine what the decision-maker values. It is important that only values are being pursued and that the decision-maker has no alternatives in mind. Having alternatives already in mind limits the thought process.

During discussions and value elicitations, the values and measures pertinent to the decision are developed and placed into a hierarchical structure. They are then weighted by the decision-maker. The weighing process allows the general process to be customized to the particular instantiation. The decision to be made, such as determining which interface design provides more effective support, is evaluated, then scored and ranked using an additive value function, producing a measure 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). The purpose of this paper is to explain the theoretical framework of VFT and not to delve deeply into the mathematics. For the interested reader, an explanation of the mathematics can be found in Keeney (1992). The below application of VFT to the interface design problem further illustrates the method.

Example Application of VFT

The interface evaluated in the research (McGee, 2003) with the Air Force Research Laboratory was for automation trying to relieve some of the cognitive burdens caused by data overload of intelligence analysts. With the advent of the Internet and the military's effort to web-enable most capabilities, the intelligence domains have quickly become inundated with data. This particular field of military intelligence is no exception and so an inferencing capability was being developed to point the analysts to the highest pay-off information. Although the development of the inferencing capability of the software had expended substantial resources, the interface had been given very little attention.

Two members of the intended user community, military intelligence analysts, met with the researchers during this study. At the initial meeting of the group, the first task at hand was to clearly identify the problem. This is an extremely critical step as this frames the rest of the investigation. After discussion, the research question was framed as “What is valued in a software interface for a complex analytical domain?” The intelligence analysts identified three main components, or capabilities, to consider in software interfaces that they use for analytic purposes: input, the underlying processing, and output. While valuing the input and output processes were expected, including a view of the underlying processing was unexpected. The analysts were adamant about including a view into the processing as this is how they judge and understand the actual software technology. Their reputation as analysts depends upon their accepting or rejecting the software’s results and to make that decision requires knowing how the results were determined. Using these three components as the main elements seemed intuitive to the subject matter experts (SMEs) as a way to capture the important aspects of software interface. These components became the first, or top, tier of the hierarchy. The top hierarchy was then further investigated to develop the second tier of the hierarchy as shown in Figure 1.

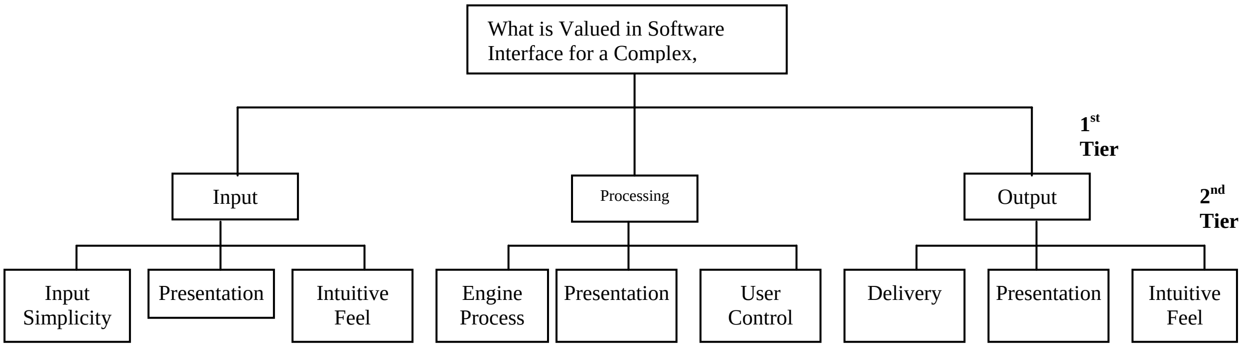

Figure 1. Top tiers weighting with Local and Global Weights

Values, such as presentation and intuitive feel, appear in more than one branch and it may appear that this violates the need for mutual independence. However, since the Input, Processing, and Output components are mutually independent, the same value terminology can be present in each branch and still remain independent. The complete second tier was further defined into three lower tiers to make a total of five levels. For illustrative purposes, Table 1 shows the third level breakdown (italics) and fourth tier (indented under the italics) for the Engine Process, a second tier level.

The fourth tier attributes were given associated measures by the SMEs and then assigned a lower and upper bound (Table 2). A lower bound would indicate when the attribute has zero value and an upper bound is when the attribute would have 100% value. With the hierarchy built and measures and bounds agreed upon, weights were assigned. Weights were assigned to a generic concept of a software interface and not to any particular instantiation of an interface. There are two types of weights, local and global. A local weight identifies the total proportion that a value has of the value above it. The sum of the local weights in a *branch* across a tier must sum to one. A global weight identifies the overall total proportion a value has towards the goal. In global weighting, the sum of all the global weights across a *tier* must sum to one. The global

Engine Process	To be able to display what the engine is doing.
Visibility (3rd tier)	The ability to show the algorithms that the engine is using, to see them, and to be able to step through each step of the algorithm.
Traceability (4th tier)	The ability to trace where the algorithm comes from and see where it is used in the processing.
Comprehendible (4th tier)	The ability to explain the algorithm and its uses in an understandable way.
Confidence (3rd tier)	The ability to see that the algorithm is being used correctly in the software to provide confidence in the software’s processing ability.
Appropriate (4th tier)	To show the engine is using the appropriate or the correct algorithm
Verification (4th tier)	The ability to show that the algorithm is working at the right level, being done correctly, and using the right data and calculating it the right way.

Table 1. Engine Process Breakdown

Fourth-Tier Hierarchy Value	**Associated Measure**	**Lower Bound**	**Upper Bound**
Traceability	Ease of tracing algorithms	Can’t Trace	Easy to Trace
Comprehendible	Ease of comprehending algorithm	No Explanation	Highly Specific
Appropriate	Are algorithms appropriate	No	Yes
Verification	Can data be verified	No	Yes
	Can calculations be verified	No	Yes

Table 2. Engine Process Measures and Bounds

weight of a value can be calculated by multiplying its local weight by all of the local weight of the values that are connected to it from above. Figure 2 shows the top two hierarchy tiers with their local and global weights.

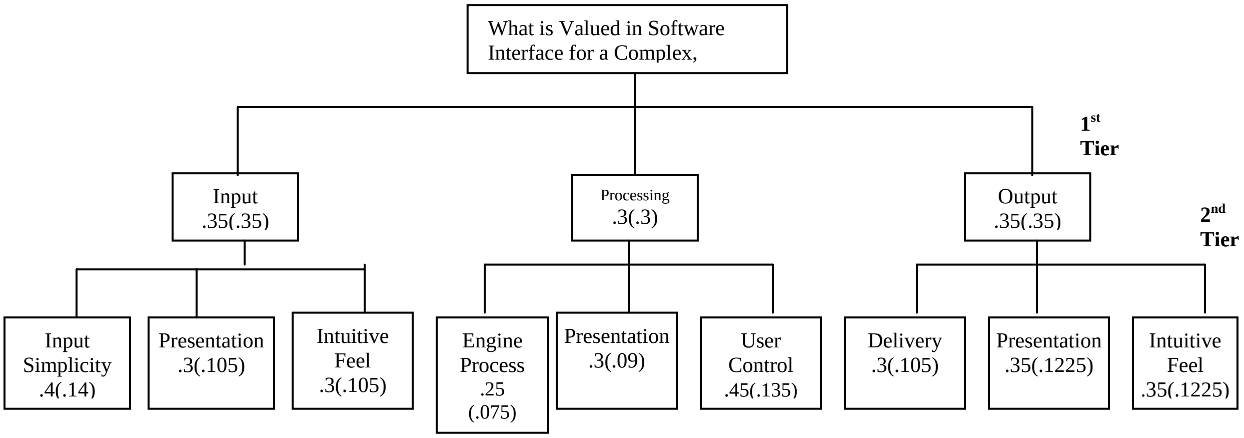

Figure 2. Top tiers weighting with Local and Global Weights (in parentheses)

The final step in applying VFT in the research was to apply the model to a specific decision, in this case the interface. The prototype software interface was scored and the deterministic analysis was performed by taking the value of the score of each measure and multiplying it by its global weight. The results were very revealing. Although development of the prototype had expended substantial resources, the interface had been given very little attention. This deficiency became readily apparent in the scoring of the interface, scoring only 0.13 out of a possible 1 (Table 3). It became even more apparent in the Output component because the SMEs determined there was no Output component so the value received was 0. Further, the developers

	Total possible value component	**Value receive component**
Input	.35	0.05
Processing	.3	0.08
Output	.35	0.0
Total value of baseline	**1**	**0.13**

Table 3. Total Prototype Scoring

did not anticipate that the analysts would want to input their own data. However, the analysts stated that they would want to additionally use the inferencing network to do ‘what-if’ scenarios and would like to be able to input data into the network but off-line from the validated model which was used for actual prediction.

In summary, the above described application of the VFT methodology showed how a VFT hierarchy is developed. In addition, this gave an example of the benefit of the method by revealing both shortfalls in the software interface (no output component) and two unexpected customer requirements (to view the engine process and to use the software to do ‘what-if’ scenarios). The next section discusses how VFT in general and the above research in particular can be applied to C2 system interface design.

Discussion

One particular problem that C2 experiences, and that many software systems are trying to address, is data overload. Billions have been and are being spent on data gathering to ensure needed data is available, but merely having data available through software does not guarantee that the correct data is displayed in a manner that contributes to timely understanding. Although everyone agrees that data overload is a commonplace challenge that is extremely difficult to address, the precise definition of data overload is far from obvious. Common to most definitions of data overload is the notion that somehow an excessive amount of data creates additional cognitive burdens for the human operator. In many a developer’s mind, this immediately conjures up ideas of automation solving the problem behind the scenes and presenting a neat and tidy solution to the problem at hand. VFT can be applied to C2 system interface design to gain understanding of what data, or type of data, needs to be displayed to the decision maker.

The VFT methodology systematically reveals what is important to the decision maker for a particular decision or problem. Focused, hard thinking about the problem provides a deep and thorough understanding of the issues intrinsic to the decision at hand. All multi-attribute decisions, such as is encountered in C2 system interface design, require making trade-offs and revealing what is important while exposing the issues is necessary to make the most informed decisions on the trade-offs. Keeney (1992) lists some of the benefits as guiding information collection; evaluating alternatives; interconnecting decisions; improving communications; facilitating involvement in multiple-stakeholder decisions, and guiding strategic thinking. These are all benefits required for developing usable and useful software interfaces for C2 and the structured, systematic VFT method helps ensure that the entire problem space is included.

The output of the VFT process includes the actual hierarchy. Developing the hierarchy provides the insight and communication, but once the hierarchy is built, it can be used to evaluate alternative interface designs without having to again fully engage the decision maker in the process.

The described research with military intelligence analysts is relevant to developing systems for C2. Similar to the analysts, those performing C2 are staking their decisions and reputation on the underlying technology. While many of the machine-to-machine ideas are indeed useful, the ultimate responsibility for the problem being resolved undoubtedly still rests with a person and not the automation. This would definitely hold true for C2 where life and death decisions are made. Rather than merely accept the automation's response, the commander and his staff would want to understand *how* the answer was derived. Since many data feeds are automated using the machine-to-machine paradigm, the user can only sort through and make judgments based on what can be accessed through the interface. If the interface does not allow access to what the user needs or represents the data improperly, the cognitive burden is increased and not reduced.

Conclusion

Interfaces, from displays for command and control to screens for military intelligence analysts, are the gateways to the underlying technology. Despite how state-of-the-art and sophisticated the underlying algorithms are, if the interface does not address the user's critical functions, the software capabilities will be underutilized. Therefore, design of the interface is critical to ensure the user has the right information displayed at the right time in the right way. Value Focused Thinking (VFT) provides an objective methodology that is well suited for handling multi-objective problems such as interface design and can be used to guide development so that development can be focused on the most relevant aspects of the interface development.

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. *Value-Focused Thinking: A Path to Creative Decisionmaking*. Cambridge, MA: Harvard University Press, 1992.

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

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

McGee, Christopher M. "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.