A Multi-model Aid for Interface Design (MAID): Helping Designers Reason about Information Match

Christopher A. Miller, Peggy Wu, Eric Engstrom, Jeff Rye, Kimberly Ferguson-Walter
Smart Information Flow Technologies, LLC
Minneapolis, MN

Debra Schreckenghost
TRACLabs, Inc.
Houston, TX

We previously developed a core representation for describing the information a human needs to perform a task and the information provided by a user interface. This representation is highly abstract and is based on information theoretic properties, thus it can be applied to a wide variety of work domains and information and display types. Since information need and information conveyed are described in the same numerical scales, it is straightforward to compute a degree of match between them. In prior work, we used this capability to dynamically and automatically reconfigure cockpit displays for military cockpits. In recent work, however, we adapted this approach to the task of evaluating and critiquing display format designs to support procedure execution in the context of NASA's space operations. The representation and reasoning approach generalizes well to describing information types in procedural domains. The resulting tool can be used to (a) analyze a proposed display format for a given task, (b) propose a format for a given task, (c) project how changes to a procedure will affect the suitability of a previous format, and (d) project how changes to a format will improve or reduce its suitability for a given procedure.

INTRODUCTION

User interface (UI) development is an art and science that requires substantial training and experience. Even for experts, each new design requires much effort and, frequently, long periods of "iterative prototyping:" uncomfortably close to trial and error. As systems become more complicated, developing interfaces for them becomes more complex as well.

NASA's extraordinarily complex and highly critical work domains, especially for space, represent extreme examples. Hence, NASA is seeking tools to aid analysis of interfaces via quantitative measures of effectiveness. These imply a core representation that enables a comparison of task-based information needs and the degree to which a candidate, multi-modal information presentation method meets those needs.

One particularly high-need area for aiding is displays to present procedures and support their execution. While this need exists in many domains, both the need and the opportunity to use a procedure presentation design aid is greater for NASA's space applications than most others. NASA makes greater use of procedures than most other fields, preplanning most activities and maintaining a large library of procedures and then revising those procedures prior to each use. Of course, NASA's need for accurate procedures is greater due to the degree of complexity of the systems involved and the consequences of failure. The fact that there is such a robust "culture of procedures" (Jamieson & Miller, 2000) at NASA provides both the opportunity and the motivation to augment procedure presentation through a display design aid.

Over 15 years, we have developed, designed, used and validated a representation for reasoning about information required for task performance, information conveyed by a candidate display and the degree of match between the two. We have not previously used the representation to assist in the more interactive task of aiding a human designer to create UIs, but rather in Adaptive Information Management (AIM) systems: the automated configuring of UIs at run time.

We leveraged our prior work on AIMs to provide a core reasoning capability for review, design, evaluation and management of UIs to support procedure viewing and execution. To date, our work has focused on adapting the prior AIM approach to both reason about procedural information and to operate as a semi-autonomous design aid. For convenience, we call this product a Multi-Modal Advisor for Interface Design or, MAID. We have recently completed implementation of a working prototype of MAID and have demonstrated it's utility in performing a variety of analyses on a different NASA procedure and display types—for the International Space Station, for the Cockpit Avionics Upgrade to the Shuttle and for potential design concepts for the Orion or Crew Exploration Vehicle. In the remainder of this paper, we will first describe MAID reasoning approach and then illustrate its use in a specific example.

REASONING ABOUT INFORMATION MATCH

Our approach to reasoning about information match has been described extensively elsewhere (Miller, 1999) and its use in AIM systems such as the Pilot's Associate (Miller, Shalin, Geddes & Hoshstrasser, 1992) and Rotorcraft Pilot's Associate (Miller & Hannen, 1999) have been reported. Thus, we will only briefly summarize the approach here.

The basic reasoning of MAID is shown in Figure 1 and can be summarized as follows: tasks give rise to information

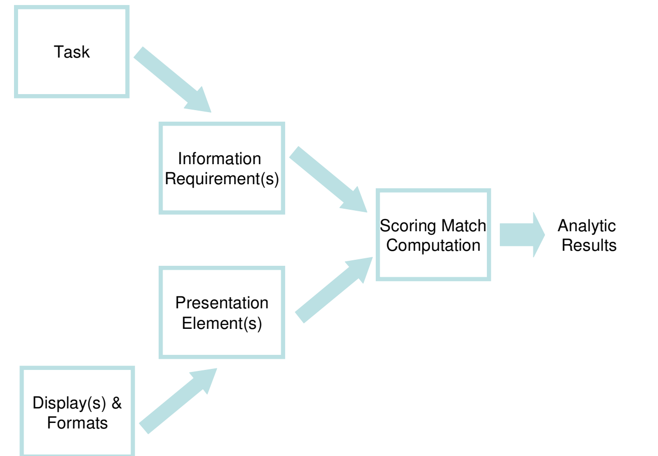

Figure 1. Summary of information match reasoning.

requirements (IRs), thus task knowledge can yield knowledge about the information required to perform those tasks. IRs can be met by Presentation Elements (PEs). PEs can be grouped only in predefined, "legal" and familiar combinations known as Formats. Formats can only be presented on certain Devices in the workstation environment. Insofar as the underlying "vocabulary" for describing the information referred to by IRs and PEs is the same, reasoning about the degree of match or satisfaction between the information needs of a context or task and the information provided by one or more candidate displays is possible.

The knowledge implied by Figure 1 can be divided into the four types: (1) Task knowledge, (2) Information Requirements knowledge, (3) Information Presentation knowledge (combining displays and PEs), and (4) Scoring Components for overall evaluation. Each of these classes of reasoning will be discussed in separate subsections below.

Task Knowledge

Tasks are a powerful, human-centered means of structuring and thinking about human activities. Greeno (1983) viewed information needs as the arguments to a problem solving process represented by the task to be performed, and Rouse, et. al.(1987) and Mitchell and Saisi (1987) pioneered concepts for automated, task-based information management.

In previous work (e.g., Miller & Hannen, 1999), we used a dynamically tracked task model of possible mission tasks combined with "intent inferencing" software to estimate currently active pilot tasks. Here, however, we are treating procedures as their own task models. Since a procedure is already decomposed into steps, the hierarchical and sequential decomposition of a task model is preserved. Better still, procedure execution (especially in NASA's highly instrumented environments) requires accessing a procedure step to obtain both information about performing that step and supporting displays for it. This largely eliminates the need to "infer intent." Instead, we treat step selection as a form of "intent declaration" and make the reasonable assumption that when an operator calls up a given step, s/he needs information required to perform that step.

Information Requirements (IR) Knowledge

Tasks organize an operator's activities into discrete, goal-directed chunks. Thus, the set of operator tasks identify and organize the information and interaction capabilities the operator will need. This is, fundamentally, the insight behind task analysis to derive display and training requirements in human factors (e.g., Kirwan & Ainsworth, 1992). Sets of task x IR clusters can be analyzed globally (over the complete set of tasks the operator may encounter) or in various subsets (e.g., over currently active tasks or some other subset the designer is concerned with). Even within the analysis of a single procedure, we make the distinction between persistent IRs (those needed throughout the duration of the procedure) and step-specific IRs (those needed only for the single step).

Developing computational methods for reasoning about information 'fit' mandates developing a computationally tractable representation for the information required by a task and that provided by a display. The representation we developed for Information Requirements (IRs) was a simple data structure that listed the IRs for each task, along with some parameters describing how the information was needed for that task. Thus, the task *Vector_for_ Landing_Approach* might require IRs like *Heading, Bearing, Altitude, Gear_Status*, etc. IRs represent abstract information needs, independent of any specific display method. Including an IR for a task means that the information is needed, but says nothing (yet) about how it is provided. Because this is an abstract representation of need, it is inherently multi-modal insofar as alternate presentation modalities might be used to satisfy the need.

Simply listing the IRs for a task is not sufficient for selecting a good presentation method. We need to describe how the information is needed for the performance of the task. We accomplish this using a set of descriptive parameters created by Geddes and Hammer (1991) and refined and formalized by us (Miller et al., 1992). Each IR in a task is described in terms of five parameter/value pairs that characterize how that information is needed for that task. We refer to these as the SRBIC parameters—after the first letter in their names. Values for each parameter range from 0-10 and represent the 'proportion' of that parameter which is needed for this task. Definitions of the parameter terms are provided in Table 1.

It should be noted that the scalar values assigned to SRBICs for both IRs and PEs are not simply opinions or individual judgments, but are instead based on the information theoretic properties of the information type (e.g., heading) and how it behaves in the context of the task or display of interest. We have worked extensively to refine this process and have achieved both computational approaches to deducing SRBIC values from task descriptions and a detailed training manual for assigning them (Miller et al., 1992).

While we have illustrated these parameters for a continuous, numerical values, they have also been demonstrated to work for symbolic values. An example using these parameters to evaluate a presentation is included in the next section.

Information Presentation Knowledge

Most modern cockpit display devices (including auditory and tactile ones) are capable of presenting many pieces of in-

Parameter	Example
Scope is the extent to which simultaneous access to the total range of values for the information element is needed by the task (IR) or provided by a display (PE).	Tasks in which heading changes substantially, rapidly and/or uncontrollably may need all 360° to be visible simultaneously, while a precise maneuvering task (e.g., refueling) might need only 5% of values.
Resolution is the need (IR) or ability (PE) to make fine distinctions in the values of the information.	High resolution corresponds to the need to know that altitude is 6533' rather than "somewhere between 6000' and 7000'or that radar status is in narrow field of view mode rather than that it is simply "on". Resolution need is dependent on the number of significantly different states of the IR for the task.
Bandwidth is the need (IR) or ability (PE) of the observer to maintain timely awareness of the information value by frequent sampling and/or rapid uptake. High values of bandwidth imply the need to maintain high currency or be updated frequently. Bandwidth values for IRs in tasks are based on the rate of change between significantly different states.	The presence of a missile lock on one's aircraft needs to be known rapidly (without significant time delays) to provide maximal evasion time. Alarm buzzers and flashing lights to satisfy this need have rapid uptake rates. Both are instances of high bandwidth.
Importance—"Importance" represents the relative necessity of this information for successful task performance—as distinct from the relative importance of the task itself. Importance values approximate a probability of correct, accurate task performance "on the first try" with vs. without the IR.	Status and position of landing gear is highly important to the task of landing, while the status and ability to dim cabin lights is much less so.
Control—"Control" is the need (IR) or ability (PE) to affect the information's value in addition to monitoring it. Higher values for Control indicate more need to control the information value. For many tasks, the user simply needs to know an IR value but does not need to control it, so the control value will be 0.	The distance of an enemy fighter is an important IR, but there is no way a pilot can directly control it. By contrast, throttle position is also important for many tasks and is almost entirely under the pilot's control.

Table 1. Information parameter definitions and examples.

formation either sequentially or simultaneously. To represent and reason about this variety, we have found it convenient to describe the information that a "Presentation Element" (PE) can convey, and then define acceptable, 'legal' ways in which PEs can be aggregated into higher level constructs such as a 'page' or 'format' which can, in turn, be presented on specific types of available devices or channels.

A PE is the smallest controllable graphical (or other modal) element which can be selectively turned on and off in a workstation—roughly analogous to a graphical "widget". Collections of graphical elements that make up, say, a compass or altitude tape display could be individually selected and were, therefore, treated as PEs collectively.

PEs and IRs are represented in the same vocabulary. For example, some PEs can present "heading" information and some cannot. The IR(s) which a PE satisfies are listed in the PE knowledge structure along with parameter/value pairs that represent how that PE conveys that information—with what degree of scope, resolution, etc. As for IRs, because the PE is described in terms of the information it provides, reasoning about multi-modal presentations is simple.

The definition and scaling of the descriptive parameters is similar for PEs as for IRs. Thus, while a scope of 6 (on our scales—cf. Miller, 1992) for an IR means that 16-30% of the possible values need to be presented simultaneously, a scope of 6 for a PE means that 16-30% of the values *are* presented. Importance values are not included for PEs because the PE "inherits" its importance from the IR(s) it satisfies. Figure 2 illustrates the data structures for the IR Heading used in a Vectoring task, with several candidate PEs. It illustrates how the same representation of information and its attributes can describe both information need and presentation. The boxed scores at the bottom of each PE illustrate a match computation between the task's information need and information presented by each PE to be discussed next.

Scoring Information Fit

The core and simplest form of scoring information fit involves simply computing mismatch between the IRs needed by a set of tasks and the PEs conveyed by a set of displays. Because we use the same formal representation for both IRs and PEs and because the scales are linearized and normalized across the different parameters, the task of identifying acceptable PEs to meet an IR and computing the "fit" between them is greatly facilitated. Any PE presenting an IR needed by an active task is a candidate for presentation, but some PEs will be better than others. We can determine the fit of each PE by determining how closely the SRBIC values of the IRs match those supplied the candidate PE. The Importance value is used to prioritize the IRs within a task so that IRs that are more important receive better PE matches than those that are less important. Alternate formulae for calculating the match between an IR and a candidate PE are possible, but a typical one we have used is to simply take the absolute value of the difference of each parameter and sum the results. This penalizes each candidate PE for over or under providing the information in the way it is needed.

One way of selecting a PE is by comparing all candidates by means of this function. The PE whose score is lowest will have the least deviation from what was needed (see the scores in Figure 2 indicating that the manipulable dial is the best sin-

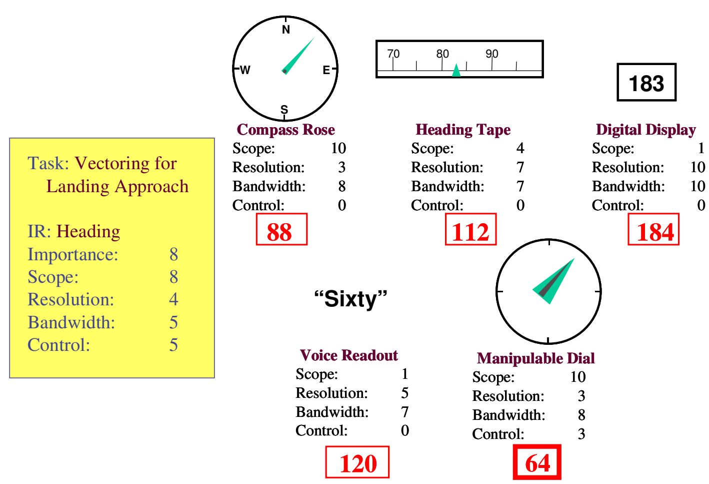

Figure 2. IRs for a task, candidate PEs and match scores.

gle PE to meet the heading need for the Vectoring task—primarily because the needed Control capability). This 'raw' fit computation can be used over sets of IRs (for a task or task set) and PEs (from different display concepts) to provide a simple analysis of fit and to answer questions such as:

- Adequacy of candidate display(s) for given procedure
- Need for display modifications given procedure mods
- Tradeoffs between alternate candidate displays for a given procedure
- Identification of low- or no-value PEs
- Identification of un- or poorly-covered IRs
- Recommendations of alternate display configurations

This core match computation, based on the information "fit" between described needs and the information provided in a candidate display or format, is at the core of the MAID tool that we have implemented—though, as will be seen, this comparatively simple computation allows us to draw some powerful conclusions when taken over even small analytic sets. We will provide an illustration of the MAID tool that we have implemented and its use to address questions in a NASA procedure domain next. More extensive and interactive demonstrations will be available on site at the conference.

MAID FUNCTIONS AND EXAMPLES

The main MAID prototype screen is illustrated in Figure 3. MAID is implemented as a Java application using Eclipse Rich Client Platform, and thus runs in Mac, PC and Linux environments. MAID takes as input a data file describing information needs and candidate displays in .csv format to provide (1) a library of elements that the analyst can manipulate. MAID then provides a number of analysis tools to compute and visualize the comparison including: (2) a table view (of information need x presentation matches and their individual scores), (3) a sequential, timeline-like graph showing both match score by procedure step and a Gantt chart view illustrate context switching between alternate display formats, and (4) summary statistics to characterize the match overall. Display element visualizations are also available (not shown in Figure 3 except via the covered tab under the sequence graph) to convey illustrations of the displays being considered. (5) Controls are provided for configuring analyses, altering the visualizations, exploring alternate match possibilities and adding, deleting and editing existing libraries.

MAID analyses, as we have prototyped and demonstrated them in the development of this tool, begin with a procedure of interest. MAID can support access to libraries of procedures in text or pdf form. Then the procedure must be characterized in terms of its information requirements, each with its individual SRBIC scores, using the guidelines we have provided. MAID supports inputting or editing these on a one by one basis or, more conveniently for larger datasets, creating them in a spreadsheet tool such as Excel and importing them to MAID as a .csv file. A similar approach must be taken to characterizing the information provided by the presentation elements contained in the candidate displays. In addition, MAID supports associating presentation elements with the displays or display pages/formats they can be presented on, as well as associating a graphical image of the display format for reference if the

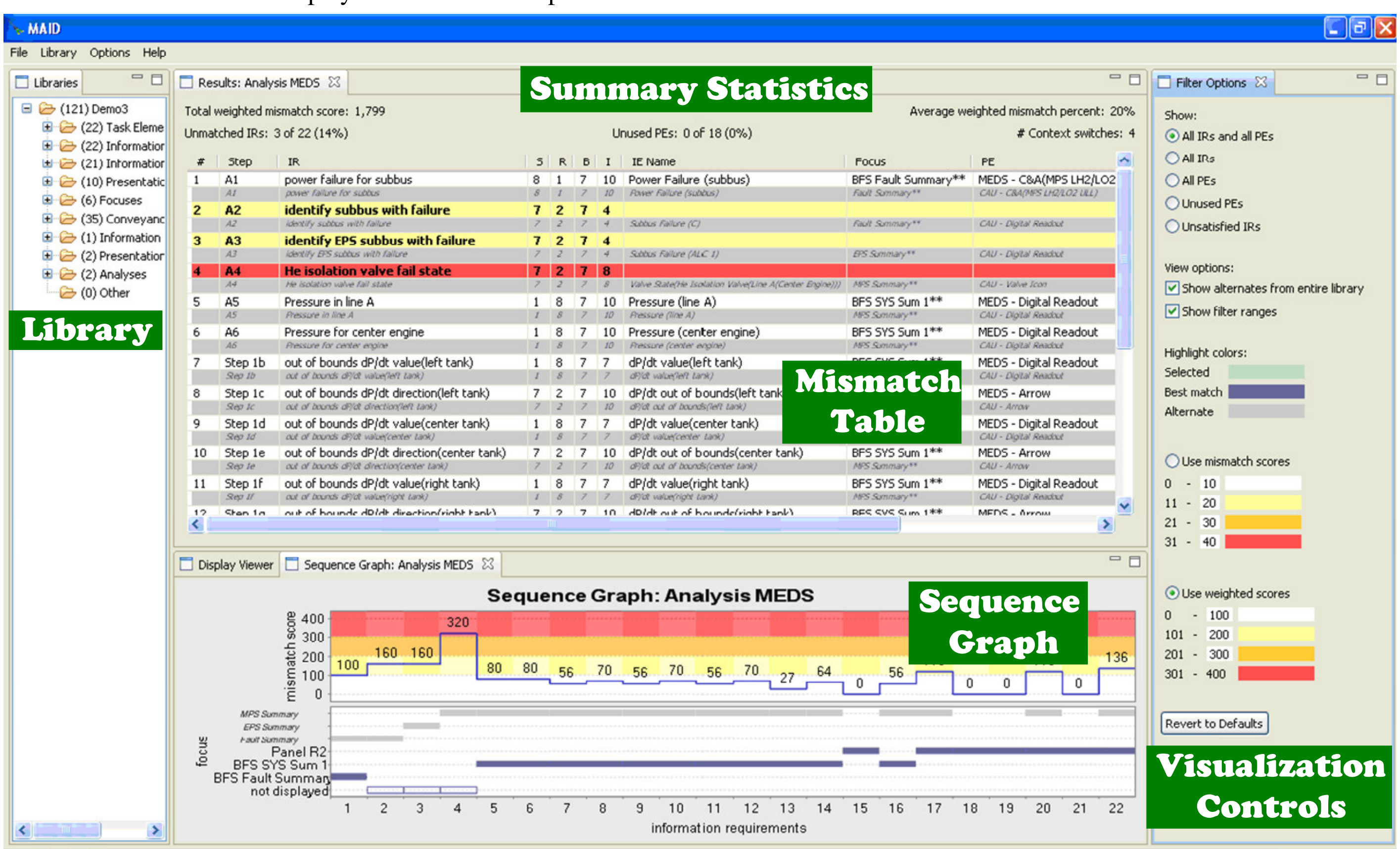

Figure 3. MAID prototype showing analysis of CAU Helium Valve display with labeled regions.

MAID-user desires.

Once all of the component knowledge is in place, a library can be opened in the MAID tool and analyses constructed. To construct an analysis the user selects a defined set of IRs (corresponding to the tasks of interest) and a set of PEs (corresponding to one or more candidate displays). Of course, additional PEs, IRs, PE sets and IR sets may be defined and then used in alternate analyses.

Once an analysis is configured, the designer asks MAID to compute the mismatch table for the comparison between the two sets. The result is as depicted in Figure 3 above. The mismatch table shows, on separate lines, each IR appearing in the set, along with the best matching PE for that IR from the set the designer specified. For each IR x PE match, then, a computed mismatch score is shown, along with a version of that score weighted by the importance of the IR. This information is also shown in graphical form in the upper portion of the sequence graph at the bottom of Figure 3—where numbers along the x-axis correspond to the sequence of IRs (first one needed, then second, etc.) The SRBIC parameter scales and the scoring method outlined above are such that a perfect match scores a mismatch of zero and a complete mismatch scores 40 "raw" points or, weighted by an importance score that can range from 1-10, 400 weighted mismatch points. Thus, we can see in Figure 3 that the average weighted mismatch score for each IR is about 80 points, but the mismatch for IR #4 is 320—a much more serious failing.

The Gantt chart at the bottom of Figure 3 shows the number of time the operator, using this set of PEs, would be required to shift focus of attention from one display surface to another. Summary statistics at the top of the display show:

- *The total mismatch score*. This figure can be used to compare this analysis to alternatives—for example, how well an alternate PE set would satisfy this IR set, or how well this PE set would satisfy the different IR set that arises if the procedure were changed.
- *The number and percentage of unmet IRs and unused PEs*—good indicators of whether the IR set is well covered and the PE set is used efficiently.
- *The average weighted mismatch percent*—the average degree to which each PE "misses" completely satisfying its corresponding IR.
- *The number of context switches* across different display pages.

The MAID user, typically a display designer or analyst, can use the controls on the right of the screen to adjust what is shown and highlighted (via alternate color schemes) and can interactively choose alternate PEs to satisfy specific IRs to see the effect of these changes on the scoring components.

One case (illustrated in Figure 3) we analyzed during our project came from the design of NASA's Cockpit Avionics Upgrade (CAU) program—a redesign of the Shuttle's display suite (McCandless, et al, 2005). A case had been identified during the course of this effort in which dual failures in the helium delivery system for the main engines during ascent could yield conditions in which following a diagnostic procedure without full awareness of the context would result in unintended engine shutdown, mission abort, and emergency landing—all very costly and risky activities. Specifically, the failed status of a helium isolation valve was had to be inferred from indirect indicators in the baseline shuttle displays, but was made explicit in the CAU displays. Under conditions where a power failure causes the He valve to fail closed before the helium leak is detected, 7 of 8 astronaut crews using the baseline Shuttle displays failed to realize that following the helium leak procedure exactly led to inadvertent shut down of an engine, while 0 of 8 crews made this mistake when using the more explicit CAU displays. In the MAID analysis illustrated in Figure 3, the need for status information about the He valve corresponds to IR #4 which is highlighted in red in both the table and the sequence graph, is shown as having no matching PE, and receives the highest mismatch score (320) in this analysis.

We will illustrate the use of the MAID tool, and the various analyses and findings it can support, with a variety of examples from NASA domains including, CAU and baseline shuttle displays, current procedures and displays from the International Space Station, and candidate display concepts from the Crew Exploration Vehicle—Orion.

ACKNOWLEDGEMENTS

This work was sponsored by a NASA Small Business Innovation Research grant, contract # NNJ06JD65C. We would like to thank Doug Wong, our contract monitor.

REFERENCES

Geddes, N. & Hammer, J. (1991). Automatic display management using dynamic plans and events. In: *Proceedings of the 6th Aviation Psychology Symposium* (Columbus OH, May).

Greeno, J. (1983). Conceptual entities. In (D. Gentner & A. Stevens, (Eds.) *Mental Models* Lawrence Erlbaum, Hills-dale NJ.

Jamieson, G. and Miller, C. (2000). Exploring the 'culture of procedures'. In *Proceedings of the Conference on Human Interaction with Complex Systems.* Urbana-Champaign, Ill. May.

Kirwan, B. and Ainsworth, L. (1992). *A Guide to Task Analysis*. London; Taylor and Francis.

McCandless, J., McCann, R., Berumen, K., Gauvain, S., Palmer, V., Stahl, W., and Hamilton, A. (2004). Evaluation of the Space Shuttle Cockpit Avionics Upgrade (CAU) Displays. In *Proceedings of the HFES 49th Annual Meeting*, Sep. 26-30, Orlando, FL, 10-14.

Miller, C., Levi, K., and Shalin, V. (1992). Learning System for Pilot Aiding: Final Report. Technical Report WL-TR-92-6002, Wright Patterson AFB. 22 May.

Miller, C. A., Shalin, V., Geddes, N. D. & Hoshstrasser, B. (1992). "Plan-Based Information Requirements: Automated knowledge compilation to support information management in an intelligent pilot-vehicle interface." In *Proceedings of the AIAA/IEEE Digital Avionics Systems Conference*. Seattle, WA; October 5-9.

Miller, C. and Hannen, M. (1999). The Rotorcraft Pilot's Associate: Design and Evaluation of an Intelligent User Interface for a Cockpit Information Manager. *Knowledge Based Systems, 12*. pp. 443-456.

Miller, C. (1999). Bridging the information transfer gap: Measuring goodness of information "fit". *Journal of Visual Languages and Computing, 10*(5). pp 523-558.

Mitchell, C. & Saisi, D. (1987) Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. *IEEE Transactions on Systems, Man and Cybernetics 17*, 594-607.

Rouse, W., Geddes, N. & Curry, R. (1987) An architecture for intelligent interfaces: outline of an approach to supporting operators of complex systems. *Human-Computer Interaction 3*, 87-122.