Modeling Information "Fit": A Tool for Interface Design

Christopher A. Miller
Jeffrey M. Rye
Peggy Wu
Smart Information Flow Technologies
Minneapolis, MN USA
{cmiller, jrye, pwu}@sift.info

ABSTRACT

This demonstration illustrates a computational analysis method using information theoretic attributes to quantitatively characterize information need for task performance, as well as information conveyed by a candidate display. Once represented in this way, various computational algorithms can provide a "mismatch" analysis of the "fit" between the two. This approach has been implemented in a prototype user interface analysis tool: MAID (Multi-modal Aid for Interface Design). MAID supports user interface design and redesign in response to procedure revisions for NASA applications. MAID has been demonstrated on examples of designs for the International Space Station, Space Shuttle (both current interfaces and proposed upgrades) and for hypothetical designs for the Crew Exploration Vehicle Orion. At least one of these examples will be presented.

Author Keywords
User Interface Design, Computational Analysis, Formal Methods, Model-based UI, Information Theory, Evaluation.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces: Evaluation/Methodology.

General Terms
Design, Human Factors, Verification, Theory.

INTRODUCTION
User interface (UI) design remains as much art as science, and usability analysis is expensive and time consuming. Rounds of "iterative prototyping" followed by usability analysis are uncomfortably close to trial and error. Highly useable UI analysis tools, ideally providing accurate, *quantitative* assessments of quality, are needed.

NASA's development and evaluation of UIs for human space missions, perhaps more than any other domain, has need of such tools. Not only are NASA's systems arguably more complex than most, with consequences of even small failures resulting in catastrophes, but there is rarely a chance to learn from trial and error and the opportunity to field a new systems happens once a decade at best. UI designs must be "right" when fielded.

Computational approaches to user interface design and analysis date to at least the compositional work of Mackinlay [1] in the mid 1980s. This conference has served as a venue for presentation and discussion of extensive work in model-based UI approaches [2]. Computational modeling of the effects of UIs (e.g., workload, reaction time, errors, etc.) is also common (e.g., [5]). Extensive work has been performed on formal methods for analyzing user interfaces (e.g., [4]) and our own former work in this area [3], for application in automated adaptation of helicopter user interfaces, forms the basis for the tool developed here.

In the tool we will demonstrate, we have taken a somewhat different approach. First, we are not attempting to automate the task of user interface design—although our approach has its roots in that goal [3]. Instead, we are aiding well-trained, professional human UI designers. This both allows and demands very fine-grained reasoning about the strengths and weaknesses of a design. Second, our approach is rooted in information theory and is not, at least directly, attempting to model human performance. This has both pros and cons, but it does remove us one step from potentially inaccurate models and predictions of UI effects and, instead, focuses directly and objectively the information contents of a UI. Finally, our approach currently offers little focus on UI layout and instead focuses primarily on content. While this may well be regarded as a weakness (and one we intend to address in future work), we believe that focusing on layout issues too early, or with too coarse-grained a reasoning approach, tends to lock designers and designs into a "compositional" design—facilitating the design or analysis of UIs that obey a core layout architecture with compositional elements. While this can be effective within the constraints of core layout templates, it proves difficult to break out of those constraints and either develop or analyze novel display concepts.

A MULTI-MODAL AID FOR INTERFACE DESIGN
We designed, implemented and tested MAID (a Multi-modal Aid for Interface Design) under a NASA Small Business Innovation Research grant. The challenge problem focused on evaluating whether a UI contained the information required to execute a NASA procedure, whether changes to the procedure necessitated changes to the UI

Copyright is held by the author/owner(s).
IUI 2011, February 13–16, 2011, Palo Alto, California, USA.
ACM 978-1-4503-0419-1/11/02.

and/or whether any of various alternate UIs did a better or worse job of meeting the needs of a given procedure.

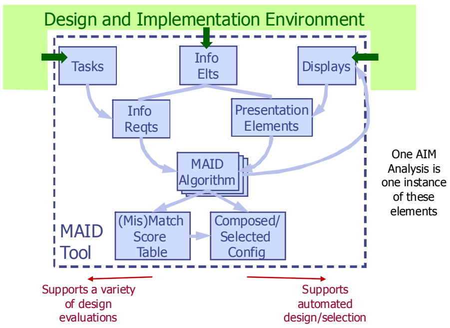

Figure 1. MAID reasoning architecture.

The core elements of the MAID representation are presented in Figure 1. MAID adopts the philosophy that information is only required because it serves as the arguments to a task-based process that would either fail, or succeed less readily if the information were not present (in an appropriate form). Thus, MAID begins with a task, or in this case, a NASA mission procedure, decomposed into its sequential and hierarchical steps. Tasks require information, which can be expressed in an implementation-independent format. For example, a given task may require heading information, but we have not yet said anything about how that information should be conveyed. Instead, we represent the way in which information is needed in that task (which, in turn, is based on the behavior and predictability of the information for that task) via five information-theoretic parameters abbreviated as SRBICs:

- *Scope*-- proportion of the total range for this information required simultaneously for the task
- *Resolution*-- level of detail required for task performance
- *Bandwidth*-- frequency user must be updated on the data
- *Importance*-- necessity of information for task success
- *Control*-- need to affect info value vs. monitoring only.

Quantitative scales for these parameters, along with guidelines for assigning them, have been developed and are presented in [3]. A key aspect of our approach is that the information is conveyed by a candidate display element (of whatever modality) can be represented in the same quantitative formalism. For example, a given task may require low scope (what proportion of the total range of 360 degrees should be shown simultaneously) but high resolution (to the degree or tenth of a degree) for heading information because heading behaves with reasonable stability during task performance contexts, but it is very important to achieve high heading accuracy. Candidate displays can then be evaluated in terms of the way in which they convey heading: e.g., a compass rose shows all 360 degrees (high scope) but it is difficult to tell a ship's current heading with accuracy greater than a quadrant (low resolution).

By representing both information need and information conveyed in the same quantitative format, we are in a good position to apply various computational algorithms to evaluate the degree of match between them. This is, in a nutshell, what the MAID tool does—providing as its core output a "mismatch table" along with graphical presentations of mismatch score, summary statistics, etc. (cf. Figure 2). A developer can then inspect these results to tell not just how good a fit the candidate display provided, but also where it failed to match and therefore might be improved. Aggregate scores (and defined mismatch thresholds) can be used to determine whether a procedure's tasks have changed sufficiently to mandate a change in the displays used to support them, and can compare the degree of fit across different candidate displays.

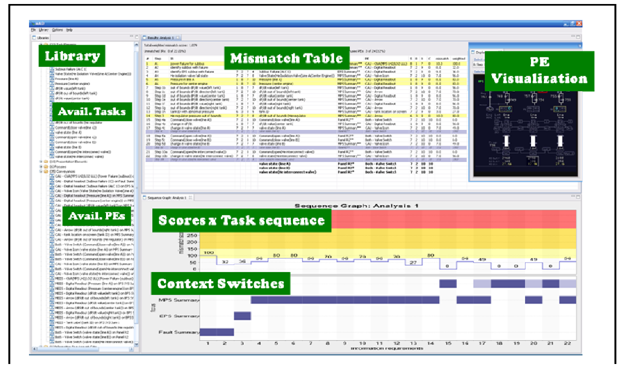

Figure 2. Output of a MAID Analysis.

These capabilities will be illustrated in an interactive demonstration of MAID to be presented at IUI'11. During the course of the MAID development effort, MAID was demonstrated on UI designs for the International Space Station, Space Shuttle (both current screen designs and a proposed upgrade) and for hypothetical designs for the Crew Exploration Vehicle Orion. At least one of these examples will be presented and explained at our demonstration.

ACKNOWLEDGMENTS

This work was supported by a NASA SBIR grant # NNJ07JA17C. We thank Doug Wong and Barbara Wolford for their program management assistance and advice.

REFERENCES

1. MacKinlay, J. Automating the design of graphical presentations of relational information. *ACM Trans. Graphics, 5*(2), (1986), 110-141.
2. Maybury, M. & Wahlster. W. *Intelligent User Interfaces*. Morgan Kaufman; San Fransisco, 1998.
3. Miller, C. Measuring Goodness of Information Fit. *J. Visual Languages & Computing, 10*, (2000), 523-558.
4. Palanque, P. & Paterno, F. *Formal Methods in Human-Computer Interaction.* Springer-Verlag, London, 1998.
5. Teo, L. and John, B. (2008). Cogtool-Explorer: towards a tool for predicting user interaction. In *Proc. CHI '08*. 2793-2798.