KNOWLEDGE-BASED CRITIQUING OF GRAPHICAL USER INTERFACES WITH CHIMES*

Jianping Jiang
Elizabeth D. Murphy
Leslie E. Carter

CTA INCORPORATED
6116 Executive Blvd. Suite 800
Rockville, MD 20852, USA

Walter F. Truszkowski

NASA-Goddard Space Flight Center
Code 522.3
Greenbelt, MD 20771, USA

ABSTRACT

CHIMES is a critiquing tool that automates the process of checking graphical user interface (GUI) designs for compliance with human factors design guidelines and toolkit style guides. The current prototype identifies instances of non-compliance and presents problem statements, advice, and tips to the GUI designer. Changes requested by the designer are made automatically, and the revised GUI is re-evaluated. A case study conducted at NASA-Goddard showed that CHIMES has the potential for dramatically reducing the time formerly spent in hands-on consistency checking. Capabilities recently added to CHIMES include exception handling and rule building. CHIMES is intended for use prior to usability testing as a means, for example, of catching and correcting syntactic inconsistencies in a large user interface.

1. INTRODUCTION

With continuing support from the National Aeronautics and Space Administration (NASA, Code O), the evolution of the CHIMES methodology and toolset has taken place in a series of research and prototyping cycles. The goal has always been to improve the usability of user interfaces developed at the NASA-Goddard Space Flight Center (GSFC) by providing user-interface designers with an automated design-evaluation capability. Recent prototypes have focused on implementing the CHIMES concept of knowledge-based compliance checking.

Available user-interface design software provides designers with many useful capabilities, with the notable exception of any capability to evaluate the "look and feel" of a graphical user interface. Such interfaces are often evaluated for compliance with human factors guidelines or corporate style-guide requirements. Evaluation is typically done by time-consuming, manual review and usability testing. Taking steps to speed up the evaluation process, the present CHIMES prototype is capable of evaluating the look of single and multiple display screens that include alphanumerics, color, and graphics. The full CHIMES concept encompasses rule-based evaluation of user-interface behavior.

CHIMES is intended for use by GUI designers prior to formal usability testing, as a means of cleaning up a GUI and improving consistency from screen to screen. Rules in the knowledge base critique the design, and an advice generator offers advice, warnings, and tips to the designer. Explication of the CHIMES knowledge base and critiquing process is the primary purpose of this paper.

2. OVERVIEW OF CHIMES DATA FLOW

Figure 1 provides a conceptual overview of the flow of data during a CHIMES evaluation. Moving from left to right on the figure, the resource file representing a GUI design is acquired by CHIMES and transformed into an intermediate representation, which is transferred to the knowledge base. The acquired GUI design is then submitted to analysis and evaluation by the user-selected rule set. Products of the analysis include problem statements ("critiques"), advice, and suggested modifications. User-selected modifications

*For further information contact: Walter F. Truszkowski, Code 522.3, NASA-Goddard Space Flight Center, Greenbelt, MD 20771 U.S.A. (301)286-8821, FAX: (301)286-1768, Email: truszkow@kong.gsfc.nasa.gov.

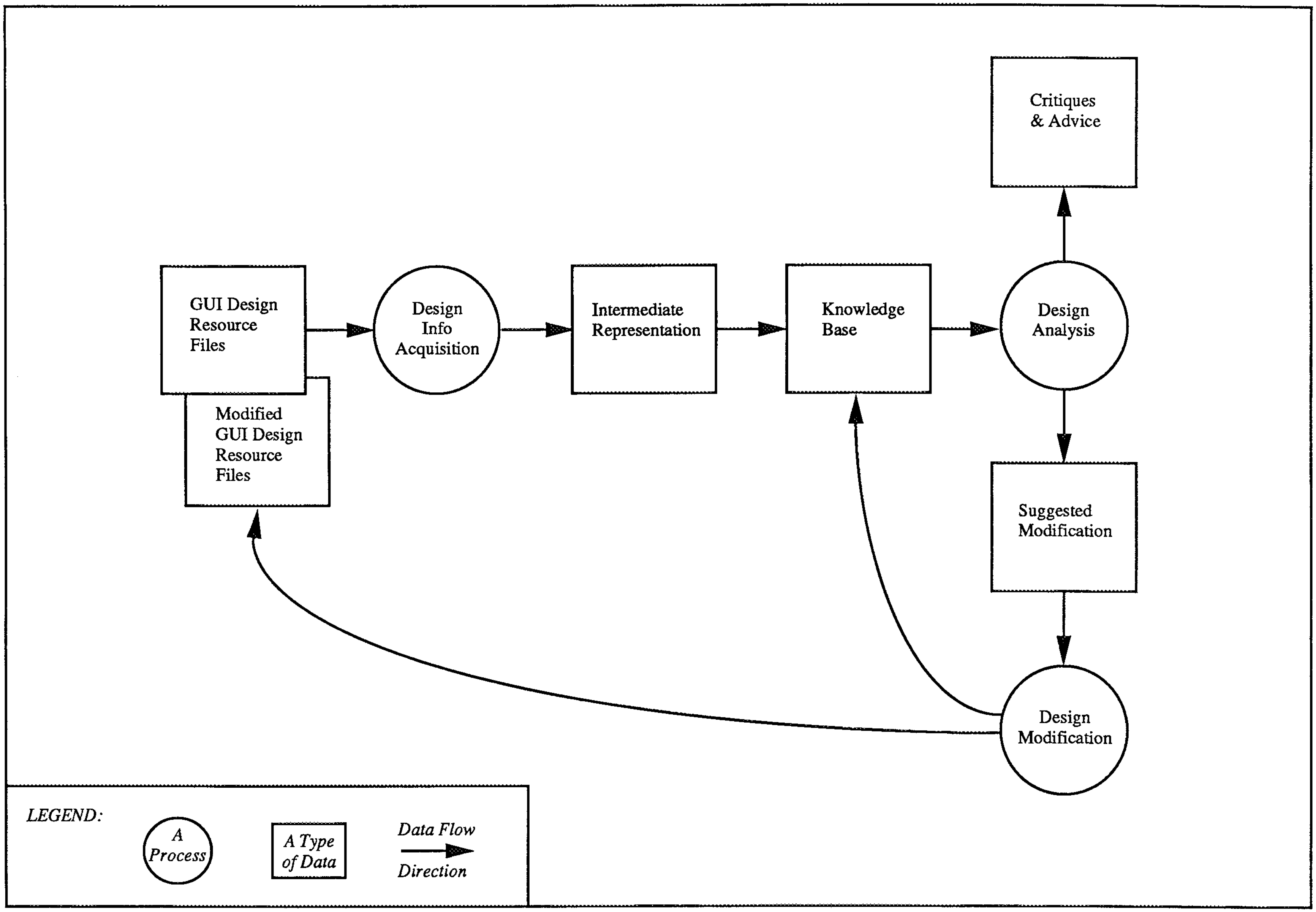

Figure 1: CHIMES Data Flow

are made automatically by CHIMES and sent to the knowledge base for re-evaluation. The resource file representing the GUI design is also automatically updated.

The remainder of the paper focuses on the contents of the knowledge base, describes the critiquing process, presents a case study, and discusses plans for enhancing CHIMES.

3. KNOWLEDGE BASE

The knowledge base stores a representation of the design to be evaluated as well as the rules that encode the heuristics for design evaluation. Each rule in the CHIMES knowledge base can be considered a critic[1]. Key components of the knowledge base include the qualitative and quantitative heuristics for evaluating the graphical design and use of color in a single display screen and, for multiple panels, heuristics on design consistency. The knowledge base is implemented in CLIPS[2].

Graphical Display Heuristics. CHIMES uses guidelines from the OSF/Motif[1] Level One Certification Checklist[8] and from the human factors literature to evaluate the "look" of single and multiple display panels. The CHIMES approach allows compliance checking of requirements and guidelines not included in the OSF/Motif defaults. For example, the number of type sizes and number of fonts per screen, as well as text justification and use of highlighting, can be checked for compliance with human factors recommendations.

Color Heuristics. The key human factors recommendation on color is that it should be used for functional purposes, not simply to decorate the screen. Functional purposes include attracting attention to critical data objects, communicating organization, indicating status, and establishing relationships between distant items[6]. To assist GUI designers in the effective use of color, CHIMES not only suggests appropriate colors but also incorporates its suggestions with the designer's functional purposes for using color and provides remedies for misuse of color.

Color heuristics implemented in the most recent prototype permit CHIMES to evaluate the consistency of color usage across multiple panels. The tool checks the consistency of both foreground and background colors; offers alternatives to the original color combination; allows the designer to preview different color combinations; and permits automatic modification of colors when the user finishes making changes.

The following are a few of the color heuristics applied in a CHIMES evaluation[5]:

- Pale foreground colors should not be used on a very bright saturated green background because of the resulting very low contrast.

- The same background color should be used for both a panel and its items unless there is a functional, user-task related reason for using different colors.

- Some background colors are not recommended for use with certain foreground colors because of the resulting color interference.

These heuristics are implemented in dozens of highly specific rules. Once the detected colors have been evaluated, CHIMES gives specific advice to improve color contrast and legibility.

Consistency-Checking Heuristics. Consistency is one of the primary human factors principles of screen design. Consistency of object location and screen behavior supports the end user's development of expectations about where to find common controls and of how the GUI will respond to user input. In general, an interface that reliably meets end-user expectations supports more efficient human performance as compared to an interface that violates end-user expectations.

As a basis for checking the internal design consistency of multiple panels or screens, the CHIMES knowledge base contains a set of rules on which there is general agreement in the human factors literature. When departures from consistency are warranted in the context of user's tasks[3], CHIMES is capable of handling exceptions.

The following are a few of the consistency-checking heuristics implemented in the CHIMES knowledge base[5]:

- The typographic elements of data items which serve the same type of function in a design are consistent within and across panels, unless there is a functional or

[1] Motif is a trademark of the Open Software Foundation, Inc.

user-task related reason for using different typographic elements.

- The background color of panels in a design is consistent across panels, unless there is a functional or user-task related reason for using different colors.
- The shadowing of pushbuttons is consistent within and across panels unless there is a functional or user-task related reason for using different shadowing thicknesses.

Although checking the consistency of location of displayed objects presents difficult technical problems, CHIMES is capable of checking the placement of the menubar. The current criterion for menubar placement is that recommended by the OSF/Motif guidelines[8]: "at the top edge of the application, just below the title area of the window frame." In the full CHIMES concept, other GUI style guides can be encoded as sets of rules in the knowledge base and applied upon user selection.

4. CRITIQUING PROCESS

The CHIMES heuristics are represented as CLIPS rules. A CLIPS rule has two parts: a conditional part and an action part. The conditional part describes the CLIPS data-memory configuration for which the rule is appropriate. (The GUI design to be evaluated is represented as facts in the CLIPS data memory.) The action part of a rule specifies the instructions to be executed when the conditional part of the rule is satisfied.

The CLIPS inference engine is the executor that determines which heuristics should be used by selecting and then executing the appropriate rule. Three steps are involved in selecting and executing rules: 1) match rules; 2) select-rules; and 3) execute rules. In the first step, match-rules, the inference engine finds all of the rules that are satisfied by the current contents of data memory according to the inference engine's comparison algorithms. The matched rules are potential candidates for execution. The second step, select-rules, applies some selection strategy to determine which rules will actually be executed. The last step, execute-rules, fires the rules previously selected.

Using the CLIPS inference engine and representing the GUI design as CLIPS facts allows the representation of heuristics as rules to match specific design patterns. For example, the rule "check-background-color-accord-pnl" represents a way to check item background inconsistency. Once the heuristics are modeled as rules, the CLIPS inference engine uses the rules to critique the GUI design that has been acquired by CHIMES.

4. CASE STUDY

As a preliminary test of CHIMES' ability to detect human factors problems in a user-interface design, we applied CHIMES to a real- world software application known as the Request Oriented Scheduling Engine (ROSE). Developed by NASA-Goddard, ROSE was designed to meet the needs of mission planners and spacecraft operators in a satellite ground-control environment[10].

The evaluation of the ROSE user interface was designed to meet two goals: 1) to identify human factors issues in need of resolution by the ROSE developers; and 2) to study how CHIMES can assist a GUI designer in catching and correcting human factors problems. For comparative purposes. we conducted both a CHIMES evaluation and a heuristic (manual) evaluation[4].

CHIMES Evaluation of the ROSE User Interface. The CHIMES evaluation took less than 10 minutes and detected three problems related to the use of fonts and typographic elements. ROSE used more than the three fonts permitted by a conservative rule in the CHIMES knowledge base. Contrary to the convention of using normal style fonts for menu options, ROSE used an italic font for options in pull-down menus. This use of italics made ROSE inconsistent with other OSF/Motif applications. CHIMES also detected typographic inconsistencies across widgets in ROSE. Several labels for the same kind of button had been implemented in mixed case, while others were in all upper case.

Heuristic Evaluation of the ROSE User Interface Three evaluators conducted the heuristic evaluation. (Two were human factors professionals who specialize in user-interface design; the third was an experienced designer of GUIs.) They spent a total of 12 person hours reviewing the ROSE documentation and on-line demonstrations. The heuristic evaluation found additional problems that CHIMES was not able to detect because of current limitations in its knowledge base.

To detect some of the problems found by the evaluators, CHIMES would need knowledge of user-interface behavior. For example, any attempt to access the ROSE help facility caused the system to crash because this facility had not yet been implemented, although a help icon was displayed on some screens. CHIMES did not detect this problem because its current knowledge base encompasses only the look, but not the behavior of buttons. The full CHIMES concept includes evaluation of user-interface behavior.

The human evaluators found problems in screen layout that CHIMES was not able to detect. In some instances, interface elements were not grouped to aid the user's understanding of their interrelationships. Further, the heuristic evaluation found that certain panel overlays obscured useful information. To detect problems of this kind, CHIMES would need semantic capabilities beyond its current scope. For example, CHIMES would need knowledge of user goals and information requirements in order to suggest alternative layouts.

A particularly difficult issue for an automated evaluation is the absence of information that should be, but is not, displayed. For example, the human evaluators noted a general lack of user guidance (i.e., instructions displayed on the screen to aid the user in navigating through the ROSE user interface). Fairly sophisticated capabilities would be needed for CHIMES to detect the absence of user guidance or other missing information.

Similarly, advanced semantic capabilities would be needed to detect redundant information. The heuristic evaluation found, for example, a redundancy in panel titles, and the evaluator recommended simplifying the user interface by removing the redundancy.

Problems of appropriate widget selection, identified by a human evaluator, pose a significant challenge to CHIMES or any automated user-interface evaluation tool. For example, five pull-down menus were lined up horizontally to perform a task that should be performed by a menubar. Although CHIMES can detect the misplacement of a menubar, it cannot currently assess the appropriateness of the widgets selected by the user-interface designer.

As highlighted in the case study, the capabilities and limitations of CHIMES make it a useful tool to aid the user-interface designer, but not one that will replace usability testing. In the realm of user-interface syntax, CHIMES can reliably detect both inconsistent design elements and non-compliance with style guidelines. With syntactic issues cleared up prior to usability testing, such testing can then concentrate on semantic issues that affect end-user performance and satisfaction.

5. CURRENT AND FUTURE DIRECTIONS

The existing CHIMES prototype reads and evaluates GUIs created in TAE Plus[9]. Although TAE Plus supports CHIMES development, it limits the designs that CHIMES can evaluate. To make CHIMES a useful tool to GUI designers who do not use TAE Plus, we are developing an interface to OSF/Motif's user interface language (UIL)[7], which will allow CHIMES to evaluate any Motif-based design.

We are also currently developing a capability to allow CHIMES users to customize the knowledge base. We have demonstrated that CHIMES can work with a knowledge base containing several sets of rules. Switching from one set of rules to another does not require recompiling. Further, we have demonstrated that a rule can be modified through the CHIMES user interface and that the modified rule can be sent back to the knowledge base for execution. Now we are developing a capability to allow CHIMES users to set up new guidelines by customizing existing guidelines. A new guideline can later be loaded into the CHIMES knowledge base for evaluating GUI designs.

Other plans for the future call for research into possible uses for CHIMES as an intelligent agent and for experimental evaluation of the effects of CHIMES capabilities on the performance of user-interface designers.

References

[1] G. Fischer, A. C. Lemke, T. Mastaglio, and A. I. Morch. Using critics to empower users. In *Proceedings of CHI'90 Human Factors in Computing Systems*, pages 337–347, N.Y., 1990. ACM/SIGCHI.

[2] J. C. Giarrantano. *CLIPS User's Guide (Version 5.0)*. NASA Lyndon B. Johnson Space Center, Houston, TX, January 1991.

[3] J. Grudin. The case against user interface consistency. *Communications of the ACM*, 32:1164–1173, 1989.

[4] J. Jiang, L. E. Carter, E. D. Murphy, S. C. Bailin, and W. F. Truszkowski. Automated evaluation of graphical user interfaces (GUIs) using CHIMES. In *2nd Mid-Atlantic Human Factors Conference*, pages 57–63, Fairfax, VA, 1994. George Mason University.

[5] J. Jiang, E. D. Murphy, and L. E. Carter. Computer-human interaction models (CHIMES), revision 3. Technical Report DTSL-94-008, NASA-Goddard Space Flight Center, Greenbelt, MD, 1994.

[6] D. J. Mayhew. *Principles and guidelines in software user interface design.* Prentice-Hall, Englewood Cliffs, NJ, 1992.

[7] Open Software Foundation. *OSF/Motif Programmer's guide (Revision 1.1).* Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[8] Open Software Foundation. *OSF/Motif style guide (Revision 1.2).* Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[9] M. R. Szczur. Transportable applications environment (TAE) plus user interface designer workbench. In *Proceedings of CHI'92 Human Factors in Computing Systems*, pages 231–232, New York, May 1992. ACM/SIGCHI.

[10] T. Weldon and J. Odubiyi. Users guide for the request oriented scheduling engine (ROSE). Technical Report DSTL-89-021; CSC/TM-90/6108, NASA-Goddard Space Flight Center, 1990.