

N 9 3 - 1 7 3 1 5

1992

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

A PROCESS FOR PROTOTYPING ONBOARD PAYLOAD DISPLAYS FOR
SPACE STATION FREEDOM

Prepared By: Loretta A. Moore, Ph.D.
Academic Rank: Assistant Professor
Institution and Department: Auburn University
Department of Computer Science and Engineering
NASA/MSFC:
Laboratory: Mission Operations
Division: Mission Engineering
Branch: Man/Systems Integration
MSFC Colleague: Joseph P. Hale

Introduction

Significant advances have been made in the area of Human-Computer Interface design. However, there is no well-defined process for going from user interface requirements to user interface design. Developing and designing a clear and consistent user interface for medium to large scale systems is a very challenging and complex task. The task becomes increasingly difficult when there is very little guidance and procedures on how the development process should flow from one stage to the next. Without a specific sequence of development steps each design becomes difficult to repeat, to evaluate, to improve, and to articulate to others.

This research contributes a process which identifies the phases of development and products produced as a result of each phase for a rapid prototyping process to be used to develop requirements for the onboard payload displays for Space Station *Freedom*. The functional components of a dynamic prototyping environment in which this process can be carried out is also discussed. Some of the central questions which are answered here include: How does one go from specifications to an actual prototype? How is a prototype evaluated? How is usability defined and thus measured? How do we use the information from evaluation in redesign of an interface? and Are there techniques which allow for convergence on a design?

Benefits of Rapid Prototyping

Rapid prototyping allows a designer to generate effective displays in a short amount of time, to experiment with different approaches to the interface, and to evaluate the displays with end users. Benefits of a prototyping process include the ability to keep development costs down and to maintain a consistent interface for the end user. Rapid prototyping keeps development costs down by iteratively refining the interface during the requirements definition phase, thus minimizing changes that will need to be made during and after flight code development. Rapid prototyping helps to maintain a consistent interface by allowing the designer use a User Interface Management System for the design of displays.

Rapid Prototyping Process

The prototyping process which has been proposed for onboard payload displays consists of five major phases: identification of known requirements, analysis of the requirements, development of a formal design representation and specification, development of the prototype, and evaluation of the prototype. Two additional phases, which are not a part of the prototyping process, are design of an implementation model and programming of the interface. The first five phases are discussed below.

Identify Know Requirements. During this phase the designer gathers all known requirements for the development of the payload displays. There are two categories of requirements. The first category is payload specific and the second category includes those requirements which contain information about the interface design. The list of requirements are as follows:

- Flight Human-Computer Interaction (HCI) Standards
- User Interface Requirements Document
- User Support Environment/Data Management System (USE/DMS) Capabilities/Constraints
- Payload System Capabilities/Constraints
- Functional Objectives
- Functional Flows
- Functional Requirements
- Operations Display Requirements
- Preliminary Procedures
- Displays and Controls Flight Systems Software Requirements (Level B D&C FSSR)

Integrate/Analyze the Requirements. The designer then integrates and analyzes the requirements of the system and ascertains the functions that are needed by the user. The products produced as output of this phase include an User-Centered Functional Analysis, a Task Analysis, an User Analysis, and Operational Flows. During the user-centered functional analysis a functional specification is derived. This specification should not only define the functions of the system and their inputs and outputs, but include human performance requirements for the functions allocated to humans. Human performance requirements should include statements concerning errors, performance speed, training time necessary to ensure the minimum skills,

and job satisfaction, at a minimum. Specific criteria for measuring these requirements should be developed in the next phase.

The goal of the task analysis is to identify what tasks are to be performed while considering the user and the system. The result is a set of specific tasks and concepts the operator will perform and manipulate with the aid of a computer. The task analysis breaks down and evaluates a human function in terms of the abilities, skills, knowledge, and attitudes required to perform these functions.

During the user analysis the designer must attempt to understand the user's mental activities or how the system being designed will influence the user's thoughts, that is, the user's views of a task and the user's view of the functionality and operation of the system. The designer should consider human cognitive processing, such as, limitations on short term memory or processing capability, and the user's model of the system. After analyzing the user the designer must be able to decide training requirements, what form of dialogue is most appropriate, and what level of support the user will require, both semantically and syntactically. The operational flows will specify the simulator requirements.

Develop Formal Design Representation/Specification. During this phase the designer should map concepts to formal representations and specifications. Two representations created should include a behavioral representation and a constructional representation. The behavioral representation is conveyed in the form of a storyboard. This is when a series of sketches (preliminary screen designs) depict the significant changes of actions which will take place during the interaction. It is a design of the appearance and behavior of the user interface. The human interface objects and their relationships augmented with preliminary sketches of screen graphics are depicted here. The constructional representation will consist of a state transition diagram. The state diagram developed at this phase can act as a generator for the system or a mockup to be evaluated. The human interface states with their transitions, triggering conditions for transitions, and side effects are shown here. The usability specification will contain specific measurable criteria for user performance and satisfaction. Convergence on an interface design is achieved by correcting components of the user interface that do not meet the usability specification.

Develop Rapid Prototype. The actual development of the prototype involves prototyping of the displays using the HCI prototyping tool, development of a low fidelity simulator, building of an interface between the displays and the simulator, integration of these components, and testing to ensure that the interface does what the designer expects.

In order to prototype the displays, the look and feel of the window(s) must be defined first. This includes answering questions such as: What will be the initial size and position for the window? What color should the background be? How will the final window work? Should it automatically page to other windows? Should there be help information? Should there be a scroll bar? Once the windows and their characteristics have been developed the types of data to be displayed in each window must be identified. For each dynamic data field the following questions must be answered: How should the dynamic field work? Should it provide online help, menus, pop-up option list, push buttons, or logging? How should the dynamic data be accessed and updated? Should the user be allowed to change the data? Should changes be confirmed and/or logged in an event file? How should the dynamic field look? What is the best way to display this data (e.g., integer, text, bar, gauge, meter, etc.). What should be its size and position in the format? What color should it be? What font should be used for text?

Once the designer has completed the analysis phase, and the content has been designed, the visual design can proceed. This involves determining the best screen layout, the grouping and structuring of information relevant to a task in blocks, the choice of terms and titles, which text symbols and graphics will be used, abbreviations and mnemonics, the distance between blocks, dividing lines, whether there will be left or right justification, design of lists and menus and their placement, and highlighting and labeling. There are guidelines to be followed for all of the areas discussed above. The disadvantage is that there are numerous and sometimes conflicting requirements. In the next section an expert system is proposed which will aid the designer by providing an online and user friendly way to access these guidelines so that they can be followed.

Evaluate Prototype. Evaluation provides a means of objectively assessing a design. It allows the designer/developer to verify user and system performance against requirements, to assess the performance of the User Interface dialogue, and provides data to the iterative design process. During the formal design representation and specification phase, the designer created a usability specification, after each evaluation iteration the user and system evaluation characteristics will be compared against pre-established criteria to determine the degree of correspondence. This allows convergence on a design, by correcting those components of the User Interface that do not meet the pre-established criteria.

During evaluation, the prototype will be shown to the principal investigators of the experiment and typical end users who will provide feedback to the designer and/or developer. The evaluation will consist of a static and dynamic evaluation component. Evaluation of the static displays will involve assessing the displays to determine whether or not they are in compliance with standards such as the Flight Human-Computer (HCI) standards. The dynamic component will consist of evaluation of the interaction between the user and the system.

Evaluation (of the Static Displays). Displays are currently measured against standards by manually inspecting the design and comparing it against what is contained in the requirements document. This process is time consuming and produces errors because of the large volume of requirement standards that exist. An Expert System for Evaluating Static Displays of the HCI is proposed which will evaluate the design, state any problems which are found, provide an explanation for the problem, and offer a suggestion on how to correct it. A good model is Tullis' Display Analysis Program [Tullis, 1988] which takes alphanumeric screen designs (no color, highlighting, separator lines, or graphics) and produces Tullis' display-complexity metrics plus some advice. The only problem with this program is that it only evaluates alphanumeric displays (no color, highlighting, separator lines, or graphics).

Evaluation (of the Human-Computer Interaction). Evaluation of the dynamic component should be accomplished by conducting usability studies. Given a functional prototype and some tasks that can be accomplished on that prototype, the designer should observe how users interact with the prototype to accomplish those tasks in order to identify improvements for the next design iteration. There are two problems that designers should be aware of when conducting these studies: the performance of the users is usually elevated under test conditions and problems created by the artificial environment of testing. Once the evaluation is complete the results should be measured against the pre-established usability criteria and the iteration should continue until there is a match between the output of the evaluation and the usability specification.

Measurable evaluation parameters should include: time to learn to use the system (i.e., training time measurement - how much time it takes to reach a particular level of proficiency), speed of task performance (or time to complete representative tasks), rates and types of errors by users, retention over time, and subjective satisfaction. The load demands of the work situation might also need to be assessed, as well as, whether or not there was effective operator planning. For example, whether or not the user carried out the operation, carried out the operation as efficiently as possible, used wrong commands, used too many keystrokes, or received several help and error messages. The system should also be evaluated to determine which features of the system were used or not used effectively. For example, the number of times a help or explanation screen was requested. This will give the designer or developer some indication of which features of the system should be enhanced and which should be eliminated. Analysis of the errors and types of errors encountered will assist in redesign of the screens and dynamic data objects.

There are several techniques proposed for the collection of this data. They include embedded evaluation techniques, observation, and subjective satisfaction measures. The embedded evaluation techniques includes a capture/playback component and an analysis component. The Capture features captures a user's session and saves this information to a log. This log can later be "played" back or analyzed. Simple reports such as the frequency of each error message, menu-item selection, dialog-box appearance, help invocation, form-field usage, etc., are of benefit in order to redesign the interface. Observation allows a designer to sit down with the users of the system, in order to monitor the interaction, and observe any problems that the user may encounter. This information is used for the next iteration. Subjective satisfaction measures a user's acceptance of a system. This can be done through the use of surveys, questionnaires, and interviews. Shneiderman [1992] developed the Questionnaire for User Interface Satisfaction (QUIS) as a generic user-evaluation questionnaire for interactive systems. This survey could be used in conjunction with an interview.

Interactive Prototyping Environment.

There is no current tool which allows for screen design, simulation, and evaluation. The most advanced tools for rapid prototyping which do not require programming experience are called UIMS (User Interface Management Systems). This term is used to describe software tools that enable designers to create a complete and working user interface without having to program in a traditional programming language. However, the users have to use a programming language to implement additional functions such as database search, network communication, or scientific computation. An environment is presented here which will allow for development, simulation, and evaluation of designs.

There are four components of the HCI prototyping environment: 1. HCI format development tool, 2. Test and evaluation simulator development tool, 3. Dynamic, interactive interface between the HCI prototype and simulator, and 4. Embedded capability to evaluate the adequacy of an HCI based on a user's performance. The HCI format development tool allows the designer to dynamically develop displays. The test and evaluation simulator development tool will allow the functionality of the system to be implemented and will act as a driver for the displays. The dynamic, interactive interface will handle communication between the HCI prototyping environment and the simulation environment. The embedded evaluation tool will perform the evaluation of the Human-Computer Interaction in terms of the measures discussed in the previous sections.

Conclusions

The process presented above provides a means of producing an efficient and effective prototype. Methodologies for gathering and refining operations and end user requirements, approaches to display design, and measures and methods for evaluating designs were presented. An architecture which will enable this process to be carried out was also presented.

References

- Bass, Len and Coutaz, Joelle (1991). *Developing Software for the User Interface*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- Karat, John (Ed.) (1991). *Taking Software Design Seriously: Practical Techniques for Human-Computer Interaction Design*. Boston, MA: Academic Press.
- Shneiderman, Ben. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (Second Edition)*. Reading, Massachusetts: Addison-Wesley Publishing Company.
- Tullis, T. S. (1988). *Screen Design*. In Helander, Martin (Editor), *Handbook of Human-Computer Interaction* (pp. 377-411), Amsterdam, The Netherlands: Elsevier Science Publishers.