Application of in-house software to improve the design process of multifunctional aircraft indicators and control panels

M. A. Glukhov, E. D. Glukhova, P. A. Marunkov, A. S. Barulin
State Research Institute of Aviation Systems, Moscow, Russia
edgluhova@gosniias.ru

Abstract. The process of frame development for Multipurpose Control and Display Unit (MCDU), Multi-Function Display (MFD), and Primary Flight Display (PFD) involves a significant documentation effort and must produce consistent interface at the early design stages. We developed a formal description of the interface, including the interaction vocabulary, as well as custom software to simplify these tasks and improve the overall design quality. The software builds a model of the human-machine interface and uses it to automatically generate frame specification/documentation. This methodology and the software are currently used in several commercial projects.

1. Introduction.
Glass cockpit is the dominating platform for all major aircraft manufacturers in Russia and throughout the world. The process of frame development for MCDU, MFD, and PFD play an important role both in modernization of the existing aircraft models and in development of advanced piloted aircraft.

State Research Institute of Aviation Systems has acquired extensive experience in the development and prototyping of frames for MCDU, MFD and PFD. We have observed the following characteristics of this process:

- Each frame has a large number of elements, the appearance and behavior of which must be documented in detail.
- The frames are subject to frequent updates due to modifications of flight control software and crew workflow.
- The specification/documentation must be kept up to date at each such development cycle.
- Manual verification of multiple interrelated tables describing the frames is error-prone and time consuming.
- Multiple developers work on the project independently.
- The sheer number of frames makes it close to impossible to create a high quality holistic interface without automation.

This paper describes the improved process of frame design utilizing in-house software CAD MCDU and CAD MFD, developed by State Research Institute of Aviation Systems.

2. The process of frame development.
Streamlining of frameset design has become an important problem since a number of important factors must be considered during development of complex and efficient aircraft human-machine systems, including:

- development of formal user interface description capabilities (metamodel);
- consistency and unification of human-machine interface;
- automated development of user interface frames and the accompanying documentation.

Our main contribution was the application of interaction vocabulary methodology [1] to the design and development of frames for MCDU, MFD, and PFD. We defined and refined the vocabulary pyramid and developed our frame design software implementing that metamodel.

A carefully designed interaction vocabulary, which is essentially a human-computer interaction metamodel, facilitates user interface consistency and has other advantages, including:
- hardware independence;
- reusable user interface components;
- readily available reuse capability for other equipment;
- enabled automation of user interface development.

The end user (cabin crew) benefits from such vocabulary due to:
- consistency of behavior and semantics of basic vocabulary elements on a variety of aircraft models;
- short learning curve since the set of vocabulary elements is small;
- reduced human error rate and overall safety improvement.

3. The interaction vocabulary

We systematically studied and categorized human-computer interaction patterns in MCDU and developed the interaction vocabulary illustrated in Table 1. This vocabulary was the key in successful development and deployment of our in-house user interface design software. The vocabulary comprises three levels.

| Vocabulary Levels                      | Examples of user interface components |
|----------------------------------------|---------------------------------------|
| Application-specific level - frames    | frameset                              |
|                                        | frame                                 |
| Generic Input/Output actions/functions | parameter display (integer, string, etc.) |
|                                        | parameter entry                       |
|                                        | switch (go) to a different frame      |
|                                        | multiple choice                       |
|                                        | execute [command]                     |
| Primitive level - atomic actions and feedback | character display |
|                                        | button press, release                 |
|                                        | physical button illumination          |

The upper level

MCDU, by definition, is flexible hardware designed to accommodate a variety of tasks. An individual task is mapped to a frameset. For example, MANEUVER frameset supports data entry for execution of maneuvers, while CORRECTION frameset is used for coordinate correction. A frameset is either a single frame or a tree (hierarchy) of frames.

Intermediate level

At this level, I/O functionality is defined. An input or output function implements entry or display of a parameter according to a specified formatting algorithm.

Primitive level

This level includes atomic operations like individual character display or visual feedback.
4. Overview of the software.

Following this formal description, we designed databases with human-machine interaction models. A model for each frame includes the tree of interface groups and elements along with their properties. Then we developed graphic editors to populate these databases.

An interactive simulator was also developed for MCDU. It enables tuning of the crew’s interaction with the control panel right after populating the corresponding database entry (Fig. 1).

To simplify development of the documentation and to ensure its consistency, we developed a capability to export the frame functional/behavioral description. This capability eliminated the time-consuming and error-prone manual entry and verification of the tables describing the functionality.

In cooperation with onboard interface engineers, we developed a module for generating data exchange protocols with the onboard interface station and header files for the corresponding aircraft control software.

5. Frame development flow.

Determine general frame requirements, specifically, the set of tasks the crew will perform for each frame.

Define the frame structure and enter the detailed description of each frame using the GUI (Fig. 3).
Right after entering a frame model with CAD MCDU, a built-in MFC emulator can be used for testing and evaluating the crew workflow, without leaving the development environment.

6. Generation of frame documentation.
Upon entering the frame description with «CAD MCDU», several types of specification/documentation can be generated:
- frame functional descriptions;
- data exchange protocols;
- header files for the aircraft control software;
- graphic images of frames, when applicable;
- other required tables and reports.

Text documents are generated in LaTeX format, which is subsequently converted into other appropriate formats to comply with the Unified Software Documentation Standards of the Russian Federation. Graphic files are generated in PNG format chosen for its convenience in development of the crew workflows.

7. Conclusions.
Our CAD MCDU and CAD MFD software significantly improves the quality and efficiency of frame development. This is achieved by utilizing a well-defined human-machine interaction description language and automated generation of documentation. The number of errors and the complexity of the development are significantly reduced. Onboard interface engineers no longer need to be involved in the development process since data exchange protocols are generated automatically. The capability to evaluate and tune the crew workflows at an early design stage significantly improves the ergonomic quality of the frames since it facilitates the synergy of the inseparable human and machine components of the system where the fidelity of the machine and the efficiency and reliability of the person complement each other [2].

We collected development time statistics by surveying the personnel involved in frame development supplementing the data with our personal experience.

1. Time to develop Frame Functional Description (FFD), a.k.a. frame documentation (Table 2).

| Table 2 The time to develop Frame Functional Description | Original methodology | With our framework |
|---------------------------------------------------------|-----------------------|--------------------|
| Development of MCDU FFD (new crew workflow)            | 4 weeks               | 1 week             |
| Changes in MCDU FFD (crew workflow change)             | 2 weeks               | ½ week             |
| Development of FFD for MFD/PFD (new crew workflow)    | 1 day                 | 1 hour             |

2. Automatic error detection
Average number of errors detected by our software in FFD developed with the original methodology is described in the Table 3.

| Table 3. The number of errors in FFD | Original methodology |
|-------------------------------------|----------------------|
| In a new FFD                        | 30 errors            |
| In FFD after a preliminary review by an expert | 10 errors |

3. C/C++ header file development
Average time to develop the C/C++ header file is described in the Table 4.
Table 4. The time to develop C/C++ header file

|                      | Original methodology | With our framework      |
|----------------------|-----------------------|-------------------------|
| One frame of MCDU    | 3 hours               | generated automatically |
| One frame of PFD or MFD | 1 hour               | generated automatically |

References

[1] Alan Cooper, Robert Reimann, David Conin, Christopher Noessel, Jason Csizmadi, Doug LeMoine, About Face: The Essentials of User Interface Design.

[2] Silvestrov, M.M., Begichev, Yu.I., Varochko, A.G., Koziorov, L.M., Lukanichev, V.Yu., Naumov, A.I., and Chernyshov, V.A., Ergaticheskie integrirovanny kompleksy letatelnykh apparatov (Aircraft Ergatic Integrated Systems), Silvestrov, M.M., Ed., Moscow: Branch of Voenizdat, 2007.