Optimisation of information presentation on multifunction displays

Gregory Francis[1] and Clarence E. Rash[2]
[1]Purdue University West Lafayette, Indiana
[2]U.S. Army Aeromedical Research Laboratory
Fort Rucker, Alabama
USA

Abstract

A difficult part of creating a multifunction display (MFD) system is mapping the MFD information to the physical hardware. A new software tool, MFDTool, that applies an optimisation algorithm to designer-specified constraints, creates the optimum layout of MFD information on MFD hardware. MFDTool converts the constraints into mathematical cost functions and utilises standard optimisation algorithms to determine the MFD label assignment design that maximally satisfies the constraints. The software can accommodate multiple types of interactions, multiple user abilities, and multiple constraints.

Introduction

Display clutter has been a classic problem in the cockpit, where available space is limited and information volume and need are high. One solution to this problem is the use of multifunction displays (MFDs). MFDs are capable of presenting a variety of information from diverse sources, thereby freeing up space in the cockpit. They increase the total amount of information available, but with the limitation that only some of it can be presented at any given time. These displays involve fixed hardware (or simulated software) switches and flexible software pages. The information presented on these displays generally is arranged hierarchically so that the user starts at a top level and moves down the hierarchy by selecting appropriate MFD pages. Other uses of MFDs include automated teller machines (cash dispensers), medical devices, electric typewriters, retail registers, and fax machines.

Designing a MFD is a challenging task. The human-computer interactions involved in accessing information from an MFD are complicated and not entirely understood. At some point in the design of an MFD, decisions must be made about how to map the various parts of the information hierarchy to user actions (e.g., button pushes). This subtask is difficult because the ways of mapping even a small hierarchy database to hardware buttons lead to a combinatorial explosion that precludes an exhaustive search of all possible mappings. Therefore, MFD designers generally rely on experience and guidelines. Francis and Reardon (1997) summarised many of these guidelines, which, while helpful, are of limited utility because the complexity of the task makes it difficult to insure that a set of