U.S. Department of
Transportation

**Federal Railroad
Administration**

Human-Centered Technologies Tool

Office of Research
and Development
Washington, DC 20590

DOT/FRA/ORD-07/10

Final Report
April 2007

This document is available to the U.S. public through the National Technical Information Service, Springfield, VA 22161.

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE April 2007	3. REPORT TYPE AND DATES COVERED Final Report September 2003–March 2007

4. TITLE AND SUBTITLE Human-Centered Technologies Tool	5. FUNDING NUMBERS
6. AUTHOR(S) Victor Riley,[1] Stephen Reinach,[2] and Donald Green[3]	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

1. User Interaction Research & Design
   2125 Whalen Dr.
   Point Roberts, WA 98281
2. Foster-Miller, Inc.
   350 Second Avenue
   Waltham, MA 02451-1196
3. Consultant

8. PERFORMING ORGANIZATION REPORT NUMBER

DFRA.040038

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Department of Transportation
Federal Railroad Administration
Office of Research and Development
1120 Vermont Ave. NW MS-20
Washington, DC 20590

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

DOT/FRA/ORD-07/10

11. SUPPLEMENTARY NOTES

COTR: Dr. Thomas Raslear

12a. DISTRIBUTION/AVAILABILITY STATEMENT This document is available to the public through the National Technical Information Service, Springfield, VA 22161, and at www.fra.dot.gov.	12b. DISTRIBUTION CODE

13. ABSTRACT

Operator error is a leading cause of incidents and accidents, and equipment design has been shown to facilitate such errors. Human factors expertise is often used to evaluate the design of equipment and user interfaces for their potential to cause, facilitate, or permit operator error. Developing broad and deep expertise in human factors, however, requires many years of formal training, and Federal Railroad Administration (FRA) regulators are often asked to evaluate equipment and user interface design without the benefit of such training. This report describes the development and evaluation of a computer-based analysis tool to help FRA Office of Safety personnel determine whether specific railroad technology designs may cause operator errors and, if so, what types of errors. The tool allows an analyst to model the layout, physical appearance, and functionality of controls and displays. The tool then applies standard human factors principles to analyze the design for errors that may be caused by factors, such as violation of cultural conventions, inconsistencies within the interface, hidden functions, and inadequate feedback. The tool is designed to evaluate a single interface or compare two interfaces for interoperability. This report describes the software and summarizes the development and evaluation effort. It also contains a users' guide to assist tool users.

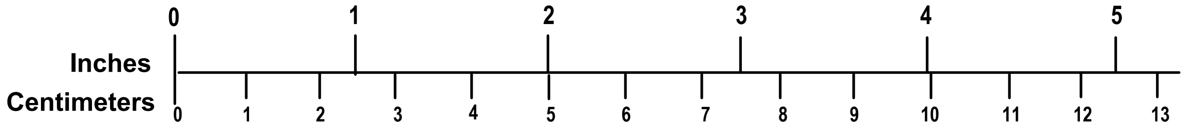
14. SUBJECT TERMS Human factors, human error, operator error, interface analysis tool, interoperability	15. NUMBER OF PAGES 89
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

METRIC/ENGLISH CONVERSION FACTORS

ENGLISH TO METRIC			METRIC TO ENGLISH		
LENGTH (APPROXIMATE)			**LENGTH** (APPROXIMATE)		
1 inch (in)	=	2.5 centimeters (cm)	1 millimeter (mm)	=	0.04 inch (in)
1 foot (ft)	=	30 centimeters (cm)	1 centimeter (cm)	=	0.4 inch (in)
1 yard (yd)	=	0.9 meter (m)	1 meter (m)	=	3.3 feet (ft)
1 mile (mi)	=	1.6 kilometers (km)	1 meter (m)	=	1.1 yards (yd)
			1 kilometer (km)	=	0.6 mile (mi)
AREA (APPROXIMATE)			**AREA** (APPROXIMATE)		
1 square inch (sq in, in^2)	=	6.5 square centimeters (cm^2)	1 square centimeter (cm^2)	=	0.16 square inch (sq in, in^2)
1 square foot (sq ft, ft^2)	=	0.09 square meter (m^2)	1 square meter (m^2)	=	1.2 square yards (sq yd, yd^2)
1 square yard (sq yd, yd^2)	=	0.8 square meter (m^2)	1 square kilometer (km^2)	=	0.4 square mile (sq mi, mi^2)
1 square mile (sq mi, mi^2)	=	2.6 square kilometers (km^2)	10,000 square meters (m^2)	=	1 hectare (ha) = 2.5 acres
1 acre = 0.4 hectare (he)	=	4,000 square meters (m^2)			
MASS - WEIGHT (APPROXIMATE)			**MASS - WEIGHT** (APPROXIMATE)		
1 ounce (oz)	=	28 grams (gm)	1 gram (gm)	=	0.036 ounce (oz)
1 pound (lb)	=	0.45 kilogram (kg)	1 kilogram (kg)	=	2.2 pounds (lb)
1 short ton = 2,000 pounds (lb)	=	0.9 tonne (t)	1 tonne (t)	=	1,000 kilograms (kg)
				=	1.1 short tons
VOLUME (APPROXIMATE)			**VOLUME** (APPROXIMATE)		
1 teaspoon (tsp)	=	5 milliliters (ml)	1 milliliter (ml)	=	0.03 fluid ounce (fl oz)
1 tablespoon (tbsp)	=	15 milliliters (ml)	1 liter (l)	=	2.1 pints (pt)
1 fluid ounce (fl oz)	=	30 milliliters (ml)	1 liter (l)	=	1.06 quarts (qt)
1 cup (c)	=	0.24 liter (l)	1 liter (l)	=	0.26 gallon (gal)
1 pint (pt)	=	0.47 liter (l)			
1 quart (qt)	=	0.96 liter (l)			
1 gallon (gal)	=	3.8 liters (l)			
1 cubic foot (cu ft, ft^3)	=	0.03 cubic meter (m^3)	1 cubic meter (m^3)	=	36 cubic feet (cu ft, ft^3)
1 cubic yard (cu yd, yd^3)	=	0.76 cubic meter (m^3)	1 cubic meter (m^3)	=	1.3 cubic yards (cu yd, yd^3)
TEMPERATURE (EXACT)			**TEMPERATURE** (EXACT)		
[(x-32)(5/9)] °F	=	y °C	[(9/5) y + 32] °C	=	x °F

QUICK INCH - CENTIMETER LENGTH CONVERSION



QUICK FAHRENHEIT - CELSIUS TEMPERATURE CONVERSION

°F	-40°	-22°	-4°	14°	32°	50°	68°	86°	104°	122°	140°	158°	176°	194°	212°
°C	-40°	-30°	-20°	-10°	0°	10°	20°	30°	40°	50°	60°	70°	80°	90°	100°

For more exact and or other conversion factors, see NIST Miscellaneous Publication 286, Units of Weights and Measures. Price $2.50 SD Catalog No. C13 10286

Updated 6/17/98

Contents

Illustrations

Tables

Acknowledgments

Human factors issues are a leading cause of train accidents and incidents in the United States. The purpose of this project was to develop and evaluate a software tool to help Federal Railroad Administration (FRA) personnel identify potential human factors problems associated with new technologies, equipment, and user interfaces, in order to help eliminate these problems before these technologies are deployed on a large scale. This research was performed under the FRA's Office of Research and Development Contract DTFR53-01-D-00029.

The authors would like to thank a number of individuals who assisted in the development or support of this tool. First, the authors would like to express thanks in particular to Dr. Thomas Raslear, FRA Office of Research and Development Human Factors Program, for sponsoring this research and providing programmatic support. The authors would also like to thank Mr. Mark Hartong, FRA Office of Safety, for championing the development of the tool, providing critical feedback during the testing and evaluation phases, and aiding in setting up the user tests and field evaluation. The authors also wish to thank Mr. Kurt Bruck, Foster-Miller, for developing early versions of the software tool.

Executive Summary

Federal Railroad Administration (FRA) Office of Safety personnel are often asked to evaluate new technologies, equipment, and user interfaces to determine if their designs may pose any safety hazards. One of the principal hazards that may arise from equipment design is operator errors facilitated by the arrangement or functioning of controls and displays. It may take years of training in human factors, however, before an evaluator can effectively identify the range of potential problems that may exist in a design.

The purpose of the human-centered technologies tool program was to develop and test a computer-based tool to help FRA personnel identify potential human factors problems associated with new technologies, equipment, and user interfaces. In particular, the tool allows an analyst to model the layout, physical form and appearance, and functionality of the displays and controls within an interface. The tool then applies standard human factors principles to determine if the design may facilitate particular types of operator error. These potential error causes include:

- Violation of cultural conventions or function mapping that may cause reversal errors
- Inconsistencies within the interface
- Unprotected controls with critical functions that may cause inadvertent activation
- Confusable controls with conflicting functions
- Hidden functions
- Mode errors
- Unlabeled control positions or display elements
- Controls with too many positions for the control type
- Inadequate feedback
- Inappropriate color use
- Inconsistent placement of information across screen pages
- Inconsistent assignment of function keys across screens
- Potential situation awareness lapses due to automation

In addition to identifying potential error-inducing attributes of a design, the tool also allows an analyst to compare two different interfaces for inconsistencies that may lead to negative transfer-induced errors. This is an important consideration in the approval of interoperability programs, where train crews may be asked to use different types of equipment for the same purpose. These inconsistencies include:

- Cases where the same function is operated by different controls or indicated by different displays.
- Cases where the same control or indicator is located in different places.
- Cases where the same control, in the same location in both interfaces, operates different functions.

- Cases where a control functions differently than the analogous control in the other interface (such as a switch that activates a function when pushed up in one system and down in the other).
- Cases where the same control or display has different labels in the different systems.

The tool functions were designed to support evaluation of the human factors issues contained in 49 CFR, Part 236, Subpart H. These include feedback, situation awareness, input errors, protections against inadvertent activation, and control/display relationships. Furthermore, because emerging technologies, such as positive train control (PTC) systems, sometimes use multipurpose graphical user interfaces and function keys with variable function assignments, the tool supports evaluation of such interfaces. It also supports a variety of workstation configurations, including conventional control stands and desktop layouts in train cabs and portable devices, such as remote control locomotive (RCL) operator control units.

Tool capabilities were developed incrementally and presented in four tests/demonstrations at FRA. At each test, participants filled out evaluation forms with ratings of perceived tool usefulness and usability, along with comments and recommendations. User ratings have been consistently and highly favorable throughout the program.

Outputs provided by the tool are messages describing the particular human factors concerns associated with each control and display. These messages are intended to be advisory only; the appropriateness of a particular design depends not only on the attributes of that control or display but also the context of the whole interface design, cultural conventions and expectations that may be uniquely associated with that function, and other factors that the evaluator may be aware of. Therefore, the tool is intended to inform the regulator’s decision, not to replace it.

1. Introduction

This report describes the development and operation of the FRA human-centered technologies tool, a computer-based tool to assist FRA regulators by analyzing user interfaces for their potential to induce operator error. A regulator would use the tool when evaluating a new railroad technology, equipment design, or user interface for potential operator input errors, display reading errors, and negative transfer between two interfaces that may raise interoperability concerns. The tool provides advice about potential operator input and display reading errors based on the attributes of the design and standard principles of human factors. The tool, however, is intended to inform, rather than replace, regulator judgment.

1.1 Background

Railroad properties and operations are becoming increasingly technological, with RCL and PTC systems and other forms of control and automation being introduced rapidly. Federal regulators face growing challenges in trying to ensure the safety of new technologies. When little or no operational experience with a new technology is available to draw from, regulators must apply their own experience and judgment to analyze new systems for their potential to create hazards.

One of the most important categories of hazards is human factors issues. Human error is the leading cause of accidents in many areas of transportation and industrial control, and railroad systems are no exception (FRA, 2006). Ideally, human factors experts would be available to perform detailed evaluations of all new technologies, equipment, and user interfaces put into operation by railroad operators. The pace of technology change and the introduction of new equipment, however, coupled with the shortage of available experts, prevent this kind of broad and deep analysis of potential human factors hazards.

Performing thorough, detailed evaluations of new designs raises several challenges. First, the evaluation process is relatively unsystematic, so the identification of issues is wholly dependent on the expertise of the evaluator and how much time he/she has to perform the analysis. Second, if the evaluation is not thorough, the regulator may not have much confidence in any related enforcement action, and the applicant may have strong grounds on which to challenge it. Third, the lack of a systematic evaluation process makes it difficult for applicants to anticipate the issues that may arise in an evaluation and the outcome of that evaluation. Consequently, applicants may not be well prepared for regulatory review.

The use of a structured tool to assist human factors evaluations, therefore, has several advantages. It can extend the regulator's expertise to cover the identification of design attributes that may facilitate human error before such errors occur in operations and lead to accidents, and can help produce a more comprehensive, yet more efficient, human factors evaluation. It can also give the regulator and applicant more confidence in the comprehensiveness of the evaluation and back up regulator recommendations with specific analysis results. Furthermore, if applicants also use the tool during system development, they are more likely to have already identified and addressed important aspects of the design that may result in human error. Consequently, designs would be more mature, and their underlying human factors issues already addressed, by the time they are presented for FRA review.

1.2 Objectives

The major objective of this program was to design, build, and test a computer-based tool to help FRA Office of Safety personnel identify potential human factors problems associated with new technologies, equipment, and user interface designs. Within this overall objective were sub-goals and requirements defined during the initial program kickoff meeting at FRA. These were to:

- Make the tool useful to, and usable by, users with no formal training in human factors.
- Make the tool capable of addressing the human factors issues associated with PTC and RCL systems, as these were the technologies of greatest concern to FRA at the time, due to their newness, sophistication, and potential for unforeseen problems.
- Make the tool capable of addressing interoperability issues by comparing two different interfaces and identifying inconsistencies between the two that may lead to human errors.

1.3 Overall Approach

Starting with these initial objectives, a functional requirements document was developed that defined the functions and capabilities of the tool and the logic of the user's interactions with it. Appendix A includes this document. In addition to these high level objectives, the tool was designed to provide assistance in the human factors areas identified in CFR § 236 Appendix E, "Human-Machine Interface (HMI) Design" (2005). Table 1 and Table 2 show how the tool's proposed functions mapped to these topic areas.

Table 1. Relationship of tool functions to high level PTC rule issues

Proposed Tool Analyses	Situation Awareness	Over-Reliance	Active Role	Feedback	Workload	Predictability/Consistency	Memory/Processing	Anticipate Errors
Errors arising from co-located controls or selection options								x
Inappropriate use of color								x
Compromised readability due to formatting problems					x		x	x
Inconsistent representation of the same function on different panels or displays	x					x		x
Inconsistent operation of the same function or interface element						x	x	x
Errors arising from similar representation of different functions						x	x	x
Mode errors		x	x			x		x
Inadequate feedback	x	x	x	x				x
Information gaps due to automation	x	x	x	x		x		x
Errors and delays arising from:	---	---	---	---	---	---	---	---
- different locations for same functions					x	x	x	x
- different terms or labels for same functions					x	x	x	x
- different graphical representations for same functions					x	x	x	x
- different control types for same functions					x	x	x	x
- different functional logic for same functions					x	x	x	x
- different automation levels for same functions	x	x	x	x		x	x	x

Table 2. Relationship of tool functions to low level PTC rule issues

Proposed Tool Analyses	Display/Control Location	Arrangement and Grouping	Stimulus-Response Compatibility	Protect Safety Critical Controls	Display Ergonomics	Text Formatting	Information Formatting	Color	Warnings	Data Entry
Errors arising from co-located controls or selection options	x	x		x						x
Inappropriate use of color								x		
Compromised readability due to formatting problems					x	x				
Inconsistent representation of the same function on different panels or displays	x	x		x						
Inconsistent operation of the same function or interface element			x							x
Errors arising from similar representation of different functions			x				x			x
Mode errors				x						x
Inadequate feedback					x	x	x	x	x	
Information gaps due to automation										
Errors and delays arising from:	---	---	---	---	---	---	---	---	---	---
- different locations for same functions	x	x		x						
- different terms or labels for same functions			x							x
- different graphical representations for same functions			x				x			
- different control types for same functions			x							x
- different functional logic for same functions			x							x
- different automation levels for same functions										

Following approval from FRA for the proposed list of tool functions, development began on the first version of the tool, which the authors tested with potential FRA Office of Safety users in October 2004. This version of the tool provided the basic workspace and interaction paradigm for the entire tool and supported a limited set of control-related analysis functions. User ratings were highly favorable so the authors continued with an expanded analysis of control functions and limited analysis of graphical display functions. Development continued through four iterations with testing at the end of each phase, culminating in a field test by FRA users.

The tool supports the following types of equipment and user interface technologies:

- Hard controls, such as pushbuttons, switches, rotary selectors, levers, knobs, etc.
- Discrete displays, such as indicator lights, meters, gauges, etc.
- Graphical displays and their elements, including diagrams, icons, text, etc.
- Function keys with variable function labels.

The tool supports analysis and identification of the following types of design problems that may facilitate user error:

- Violation of cultural conventions or function mapping that may cause reversal errors
- Inconsistencies within the interface
- Unprotected controls with critical functions that may cause inadvertent activation
- Confusable controls with conflicting functions
- Hidden functions
- Mode errors
- Unlabeled control positions or display elements
- Controls with too many positions for the control type
- Inadequate feedback
- Inappropriate color use
- Inconsistent placement of information across screen pages
- Inconsistent assignment of function keys across screens

The tool also supports comparisons between two interfaces to identify inconsistencies that may lead to user errors due to:

- Cases where the same function is operated by different controls or indicated by different displays.
- Cases where the same control or indicator is located in different places.
- Cases where the same control, in the same location in both interfaces, operates different functions.

- Cases where a control functions differently than the analogous control in the other interface (such as a switch that activates a function when pushed up in one system and down in the other).
- Cases where the same control or display has different labels in the different systems.

The tool is implemented in Microsoft's .NET framework. The .NET framework is Microsoft's preferred development environment for Windows-based applications, and it includes built-in exception handling, security features, and standard Windows user interface widgets. .NET applications run in all modern versions of Windows so the tool should run equally well in all of the operating systems employed by FRA. Initial tests were run on obsolete Windows 2000 laptops being replaced by FRA, and the tool ran quickly even in that environment.

1.4 Organization of the Report

This report describes the design of the human-centered technologies tool, including its objectives, theoretical basis, interaction design, analysis products, and testing. Section 2 discusses the tool design and use, while Section 3 describes the evaluation of the tool. Section 4 presents some recommended enhancements to the tool based on user evaluation feedback and experience. Section 5 contains a brief list of the references used in the report. The report also includes two appendices. Appendix A presents the tool's functional requirements. These requirements served as the tool's building blocks and as a communication aid between software developers. Appendix B contains a user's manual to provide a complete description of the tool's operation. Lastly, a list of abbreviations and acronyms used in the report is provided.

2. Human-Centered Technology Tool Design

This section describes the theoretical basis of the tool and the user-tool interaction design.

2.1 Theoretical Basis

The theoretical basis for the tool is drawn largely from error taxonomies developed by Fitts and Jones (1947a, 1947b). Based on extensive investigations of aircraft accidents during World War II, most of which happened in U.S. territory rather than in combat, Fitts and Jones (1947a) identified several common types of control operation errors: substitution errors (where the operator intended to use one control but used another one instead), adjustment errors (such as moving a switch to the wrong position), forgetting errors, reversal errors (operating a control in the wrong direction), unintentional activation, and inability to reach a control. An example of a substitution error is found in a design of some aircraft, in which the flap and gear handles were co-located and had similar shapes, causing some pilots to raise the gear after landing instead of the flaps.

The tool is able to identify design attributes that may lead to several control operation errors, including substitution errors, reversal errors, and unintentional activation. Fitts and Jones (1947b) also identified several types of display reading errors. The tool is able to identify several potential display reading errors, including reversal errors, interpretation errors, and substitution errors. The research team also used Norman's (1981) taxonomy of errors and slips to further inform the tool capabilities and explanations, and FRA's *Human Factors Guidelines for Locomotive Cabs* (Multer, Rudich, & Yearwood, 1998) for cultural conventions and other attributes unique to the train cab.

Based on these, the tool uses the following logic to diagnose the following problems:

- Reversal errors: Control or display movement violates cultural conventions contained in the tool database (for example, operating a control up or to the right decreases the value of the associated function), or control or display movement is inconsistent with the associated real-world function mapping (such as moving a lever forward to move the train in reverse).
- Reversal errors: Inconsistencies within the interface (for example, one knob increases a value when turned clockwise and another when turned counterclockwise).
- Unprotected controls: Controls with critical functions that may cause hazardous conditions if inadvertently activated and are infrequently used (a brake control, for example, should not be protected because it is frequently used and must be quickly accessed, but a control shutting off PTC enforcement probably should be protected because inadvertent activation could cause a hazardous condition, and the control is very infrequently used).
- Substitution errors: Adjacent or symmetrically arranged controls with the same physical form and appearance but with conflicting functions (for example, brake and throttle controls; if the consequences of a substitution error are benign, the tool will not flag adjacent controls of the same type).

- Delayed response: Hidden functions (controls or displays without appropriate labeling).
- Mode errors: Operating a control produces different consequences depending on control or system mode.
- Adjustment errors: Controls with too many positions for the control type.
- Double entry errors or other errors associated with inadequate feedback: No display associated with a control.
- Display reading errors: Colors not used consistently with cultural conventions regarding safe, caution, and warning conditions, and poor color contrast (such as blue text on a black background).
- Display reading errors: Inconsistent placement of information across screen pages.
- Function key selection errors: Inconsistent assignment of function keys across screens (for example, a particular key is used for ENTER on most screens but CANCEL on one screen).

2.2 Interaction Design

Figure 1 shows the tool workspace. The Menu Bar contains the usual Windows File, Window, and Help menus. The Dialog Area is where the user selects options and responds to tool queries. The Work Area is where the user indicates the locations of the controls and displays in the system interface under consideration. As with most Microsoft Windows applications, there is also a toolbar with commonly used functions, such as Save, Copy, Paste, Help, and Delete.

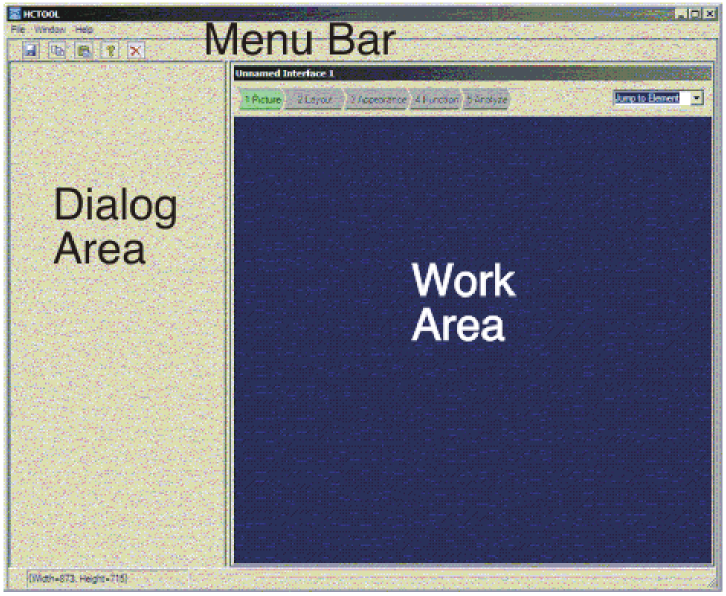


Figure 1. Tool workspace

Using the tool involves several steps that are guided by the tool interface. Because the tool uses information about the physical layout of the interface and the proximity of displays and controls to each other, the user must inform the tool about this layout. To help with this, the user can load a picture of the system interface into the Work Area to act as a guide for indicating control and display type and location. In order to support analysis of designs from the earliest stages through fielded equipment, the tool supports a variety of common graphic formats, including jpeg, gif, and bmp. Conceivably, the picture loaded by the user can be anything from a conceptual drawing on a napkin to a photograph of the finished product. Figure 2 shows the tool workspace with a photograph of the Acela cab controls to guide an analyst's placement of the various controls and displays used in that cab.

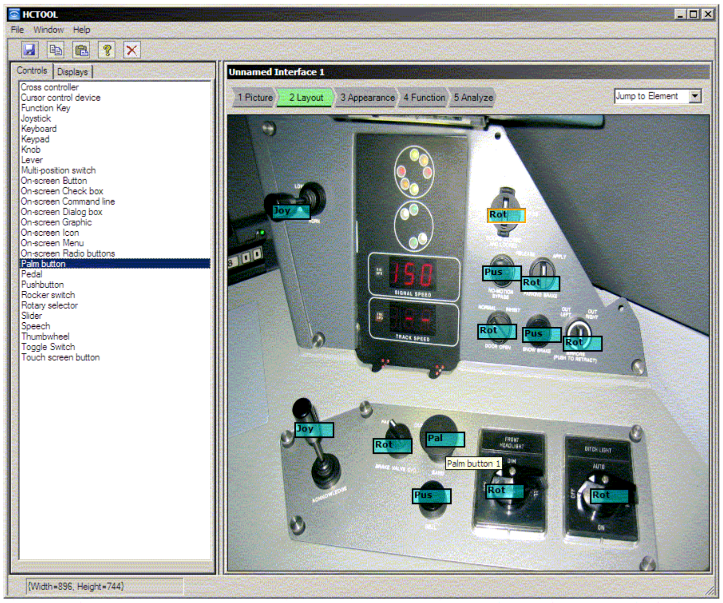


Figure 2. Tool workspace with photograph

The tool does not know anything about the picture being used; it is intended merely as a guide for defining the interface layout by placing the appropriate tags over the objects in the interface. The next section describes that process.

2.2.1 Define Interface Layout

Once the user has loaded a guide picture into the workspace, he/she can begin the process of tagging controls and displays. The resulting tags will tell the tool what types of controls and displays are present in the interface and where they are located in relation to each other. To do this, the user selects the Layout button on the top of the Work Area. A window appears in the Dialog Area giving the user access to the list of controls (in one tab) and displays (in the other) that the tool recognizes.

To tell the tool that the interface contains a joystick control in a particular location, the user selects joystick from the list of available controls types in the Dialog Area, then clicks over any joysticks he/she sees in the interface picture. In the example shown in Figure 2, two joysticks are to be tagged. The user then does the same for all other controls seen in the picture. Figure 2 shows the complete set of control tags for the Acela right forward control panel.

As stated before, the tool does not know anything about the content of the picture, but it does know about the tags. Based on the tags shown in Figure 2, the tool now knows that there are two joysticks, seven rotary selector switches, three pushbuttons, and one palm button. It also knows that each of the pushbuttons on the upper panel is surrounded on three sides by rotary selectors. This knowledge of adjacency starts to give the tool some of the information it needs to diagnose substitution errors (a potential posed by two adjacent or symmetrically arranged controls of the same type). However, whether such a danger exists or not also depends on whether the relevant controls have identical physical appearances or not, and whether operating the function of one control when the other was intended might create a hazardous condition. In order to complete this reasoning then, the user must tell the tool about what each control and display looks like and what each does. That will be accomplished in the subsequent steps.

To continue with the discussion of the interface layout definition, the user performs the same procedure to tag displays. In the example, an indicator cluster and two numeric indicators would be tagged. The displays tab, however, also provides an option for defining the layout of a graphical user interface. Selecting this option provides a set of specialized tools for defining such displays because the various elements on a graphical display can be of virtually any size, and they appear on a variable background.

Defining the layout of a graphical display is accomplished on two levels. First, the user defines the major background areas of the display as graphic regions. These regions serve as containers for the icons, text boxes, diagrams, function key labels, and other graphical elements that the user interacts with. Next, the user defines the graphical elements that appear in each region by selecting the type of element (diagram, icon, text, etc.) and then clicking and dragging to form a box over the element. This procedure allows the tool to handle graphic elements of various sizes and recognizes that some graphical elements may appear within other graphical elements (such as the icon of a train appearing on the diagram of a track on a graphical PTC map display). Figure 3 shows an example of this. Figure 3 also demonstrates the use of the tool on a conceptual design sketch. Once the interface layout has been defined, the user can move on to defining the physical form and appearance of each control and display.

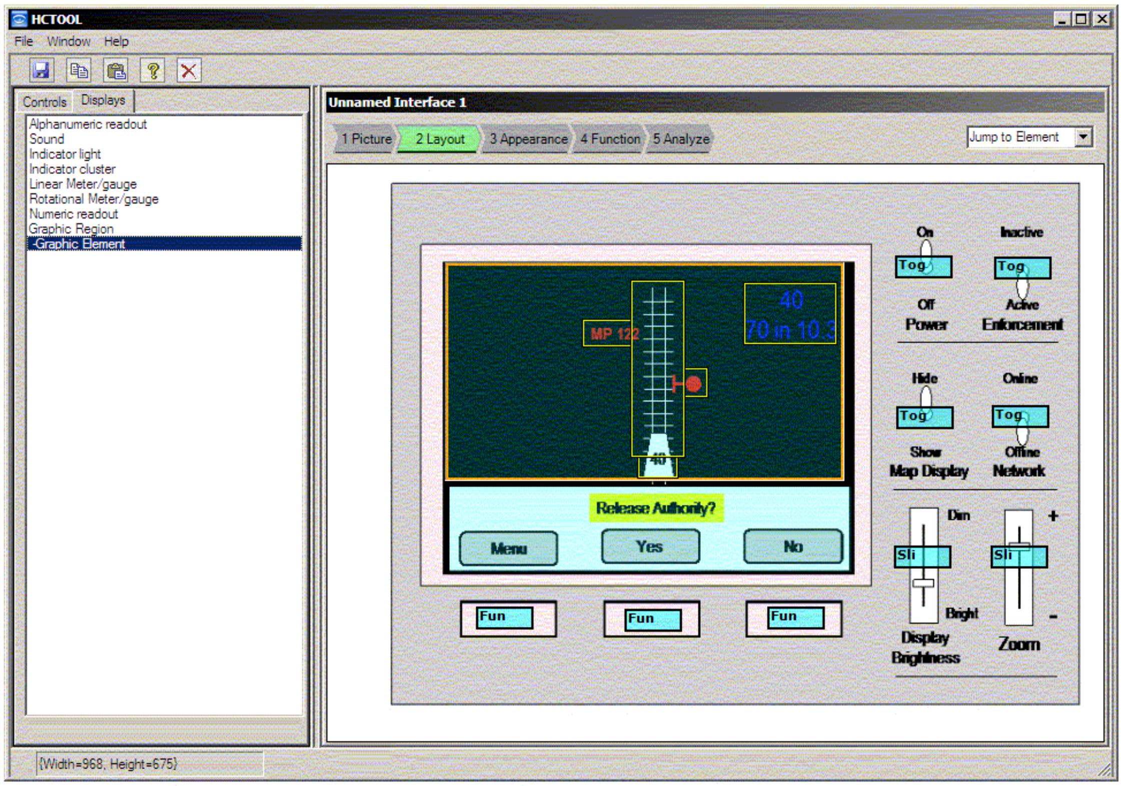
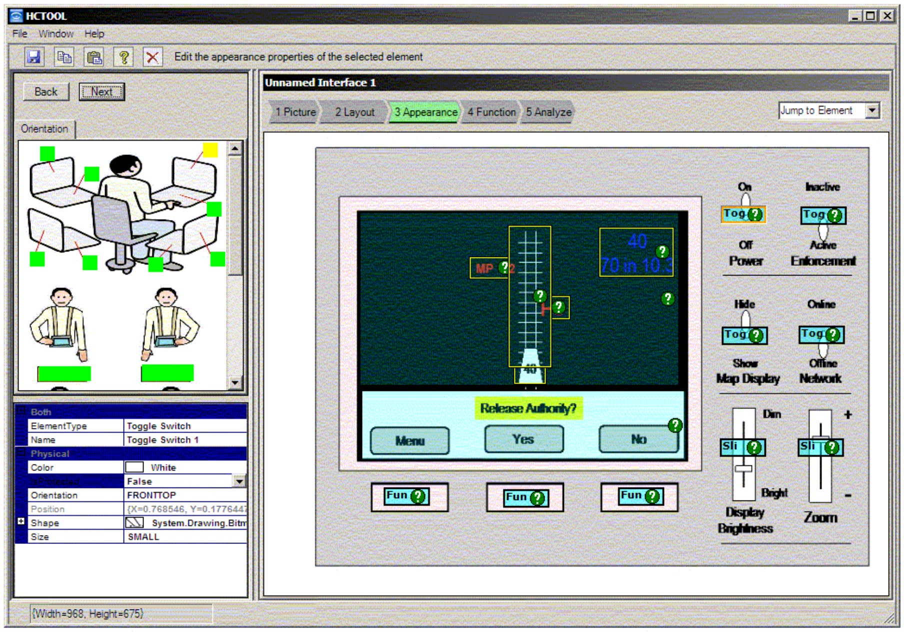


Figure 3. Defining the positions and types of graphical regions and elements

2.2.2 Define Control and Display Physical Forms

To start the process of defining the physical form and appearance of each control and display, the user selects the Appearance button above the Work Area. This brings up a dialog in the Dialog Area that guides the user through the appropriate appearance dialog for the selected control or display. The user selects a control or display for definition by clicking on its associated tag, and the dialog asks the user to select the shape, color, size, and orientation most appropriate for that control or display. The dialog is customized for each control and display type. For example, if the user selects a toggle switch, the dialog offers a selection of possible toggle switch shapes to choose from. If the user selects a graphic region, no shape options are presented because background regions do not have meaningful shapes.

As the user selects the options appropriate for a given control or display, the tool populates a properties box containing the properties the user is defining for that object. This properties box appears in the lower portion of the Dialog Area. After completing the definition of an object's properties, the user can go back and edit the properties directly within the properties box or use a series of tabs within the Dialog Area to select the specific property he/she wishes to change. Figure 4 shows an example of a properties box.



Both	
ElementType	Toggle Switch
Name	Toggle Switch 1
Physical	
Color	□ White
IsProtected	False
Orientation	FRONTTOP
Position	{X=0.768546, Y=0.1776447
Shape	System.Drawing.Bitm
Size	SMALL

Figure 4. Orientation dialog and properties box

Figure 4 also shows the orientation step for defining how a control or display is oriented in relation to the user. The tool must know whether a control or display is oriented horizontally or vertically, as well as whether it is on a fixed workstation panel or on a control box. The tool uses this information to determine if the movement directions of controls and displays are consistent and appropriate for the functions they control or indicate, and to identify any direction of movement inconsistencies between controls of the same type in the same orientation.

The tool also recognizes that many products and systems use the same part numbers for multiple controls and indicators, for reasons of cost, simplicity, unified appearance, logistics, and so forth. Because many controls and displays in an interface are likely to be identical, the tool allows the user to copy the properties from a control or display whose properties have been defined and paste them onto all identical controls or displays. This makes it possible for a user to assign physical appearance properties to large numbers of identical controls or displays very quickly.

In this step, the user also has the opportunity to define the graphic regions, on a graphic user interface, that contain function key labels. When defining such a region, the user is asked how many different screens there are, where a screen is defined as a unique arrangement of function key labels. Defining this property in this step allows the tool to guide the user through the actual assignment of labels to function keys in the next step, which is to define the functions of all the controls and displays in the interface.

2.2.3 Define Control and Display Functions

To start this process, the user selects the Function button at the top of the Work Area. The tool then prompts the user to select a control or display tag in order to start the dialog that defines the operation of that control or display.

When the user selects a control, the tool asks the user to enter a name for the function operated by that control, then guides the user through a dialog in which the relevant units, importance, frequency of use, direction of movement, and associated displays are all defined. When the user selects a display, the tool again guides the user through a dialog to define the display's units, importance, and direction of movement. When the user selects a graphic region containing function key labels, the tool provides a dialog pane with pull-down menus to facilitate the assignment of functions to keys for each of the screens defined in the previous step.

2.2.4 Analyze

The user asks the tool to analyze the interface by selecting the Analyze button at the top of the Work Area. The tool then presents a window with a text report of the potential problems found with the design in a form that the user can print out. The user can also switch to a graphic view of the reported problems. In this view, yellow caution symbols are superimposed over each of the control and display tags where the tool detects a potential problem. The user can click on each symbol to see a list of the potential problems associated with that control or display. The problems themselves are listed in the Dialog Area in a tabbed interface, in which each tab title describes the type of problem and the body of the tab pane describes the specific problem and the possible human performance consequences of the problem.

The text report, which is suitable for printing, provides a summary of the analysis in which the number of instances of potential problems is listed. The potential problems include:

- Incomplete physical properties specification: Lists the number of controls and/or displays whose physical properties have not been fully defined.
- Incomplete functional specification: Lists the number of controls and/or displays whose functions and associated behaviors (movement directions, functional logic, etc.) have not been fully defined.
- Controls using critical function color but not marked as hazardous: Lists the number of yellow- or red-colored controls that were not defined to have critical functions.
- Adjacency issues: Lists the number of controls and displays that are adjacent to identical controls or displays; the functions assigned to the controls and displays in question are such that operating a control, or reading a display, when intending to use the adjacent control or display could result in a hazardous condition.
- Poor color contrast choices: Lists the number of graphical display elements (icons, text, etc.) whose colors result in low contrast with the graphic region background color they are on (such as purple text on a black background).
- Instances of poor color choices for critical controls: Lists the number of controls that operate critical functions but are not color coded to indicate that the functions are critical.
- Instances of confusing movement directions: Lists the number of controls or displays that violate cultural conventions in their movement directions (such as increasing a value by moving a control down instead of up or left instead of right).
- Instances of reversed movements: Lists the number of instances in which two controls or display elements of the same type operate in opposite directions (such as a toggle switch that turns its function on when pushed up and another that turns its function on when pushed down).

- Inconsistent function key/display screen pairs: Lists the number of instances in which potentially incompatible functions (such as CANCEL and OK) are assigned to the same function key on different screens.
- Instances of no display feedback for controls: Lists the number of controls that do not have associated displays and therefore lack appropriate feedback concerning the functions operated by those controls.
- Instances of an unusual number of positions: Lists the number of controls that have more than the recommended number of positions for the control type (such as a rocker switch with more than three positions).
- Instances of critical controls not being protected: Lists the number of controls with critical functions, whose inadvertent operation may lead to a hazardous condition and that are not frequently used and should therefore be protected from inadvertent operation.
- Instances of display movement that is inconsistent with the motion of associated controls.

After this quantitative summary, the specific issues are listed by the control or display element associated with the issue, the function associated with that control or display element, and a description of the problem and, where appropriate, a recommended solution. For example, a reversed movement report may read, "Slide 2: brake: Slider 2 has inconsistent movement with nearby control Slide 1 --- Inconsistent directions of movement between controls may cause the user to operate a control in the wrong direction."

In addition to the written, printable report, the tool provides alert tags in the Work Area next to the control or display element generating the potential issue. When the user clicks on an alert tag, which is a yellow caution triangle with an exclamation point in the middle, the Dialog Area displays a tabbed interface with the potential issues listed. Each tab contains one issue, and the tabs are titled to indicate the type of issue (such as protection) contained in the tab. At the top of each tabbed region, the specific control or display element that the tab refers to is listed along with the function associated with the control or display element. Below that is a short statement of the issue (such as, "The control is unprotected and has a critical function.") At the bottom of the tabbed region is a more complete description of the issue and a recommended design change. (For example, "Critical functions should be protected. Since it is used infrequently and has a critical function, it is advisable to protect this control to avoid accidental activation.")

Once the user has defined and saved two interfaces, he/she can also use the tool to compare the two interfaces for inconsistencies that may lead to negative transfer, which in turn can cause errors when operators accustomed to using one interface try to operate the other. These problems can include:

- Cases where identical controls in identical locations operate different functions between the two interfaces.
- Cases where the same function is operated by controls in different locations between the two interfaces.
- Cases where the same control operates in different directions between the two interfaces.
- Cases where the same control uses different functional logic between the two interfaces.

The user accomplishes this analysis by selecting Compare Two Interfaces from either the File menu or from the application splash screen, and then he/she opens the two interfaces in sequence. The tool then provides the analysis of inconsistencies in a printable report. This capability is anticipated to be useful for evaluating the interoperability of multiple systems and establishing training and operating requirements for transitioning operators from one system to another.

2.3 System Requirements

The tool runs in Microsoft Windows, on the .NET framework. .NET is included with every version of Windows from Windows XP Service Pack 2 (SP2) on and is the recommended development environment for Windows. The .NET framework can be installed on Windows platforms before Windows XP SP2 by downloading the framework from Microsoft's internet site. However, the installation package for the tool contains the .NET framework and automatically detects whether the target machine already runs the framework or not; if it does not, the installation package automatically installs it along with the tool. The tool can run on any version of Windows that supports .NET, including Windows 2000, Windows ME, and Windows 98. The host computer should have at least a 600 MHz Pentium class processor, 256 MB of RAM, and 24 MB of hard disk space for .NET. The tool itself is much smaller; if the host computer already has the .NET framework installed, the tool itself only requires about 1.5 MB. The tool is installed by double-clicking on an installer wizard, which guides the user through the installation process. The installation process is a standard Windows process, in which the user can select a target directory or accept the default directory suggested by the wizard.

3. Testing and Evaluation

Three test and evaluation sessions at FRA headquarters provided critical feedback during the tool development process. The research team conducted an alpha test in October 2004 to evaluate the tool's overall structure, approach, and interaction design, with just a few control-related analysis functions supported. A beta test was held in July 2005 with an expanded set of control- and display-related functions. A third test took place in September 2006 with a nearly complete set of analysis capabilities, including graphical user interface analysis and function key analysis. Each session was attended by potential users of the tool from FRA Office of Safety. During each session, the research team first presented the users with an overview of the tool, then showed them an example analysis. Lastly, users received an interface to analyze on their own using the tool. Researchers watched as users tried to use the tool, provided help where needed, and took notes on bugs, suggestions, and requested new features. Finally, attendees filled out an evaluation form. This form asked users to rate the potential usefulness of the tool on a three-point scale (very useful, moderately useful, not useful) and the usability of the tool on a six-point scale (very easy to use, mostly easy to use, moderately easy to use, moderately hard to use, mostly hard to use, or very hard to use). Researchers also asked attendees to share any general feedback they had about the tool and to note any features they would like future versions of the tool to include.

Table 3, Table 4, and Table 5 show user ratings for the two rated questions for each of the three evaluation sessions. Seven FRA staff participated in the first and third sessions while four participants took part in the second (beta) session.

Table 3. Alpha test user ratings

	Number of respondents
Potential tool usefulness	
Very useful	6
Moderately useful	1
Not useful	0
Tool usability	
Very easy to use	2
Mostly easy to use	3
Moderately easy to use	2
Moderately hard to use	0
Mostly hard to use	0
Very hard to use	0

Table 4. Beta test user ratings

	Number of respondents
Potential tool usefulness	
Very useful	4
Moderately useful	0
Not useful	0
Tool usability	
Very easy to use	0
Mostly easy to use	3
Moderately easy to use	1
Moderately hard to use	0
Mostly hard to use	0
Very hard to use	0

Table 5. Third test user ratings

	Number of respondents
Potential tool usefulness	
Very useful	4
Moderately useful	3
Not useful	0
Tool usability	
Very easy to use	0
Mostly easy to use	3
Moderately easy to use	2
Moderately hard to use	2
Mostly hard to use	0
Very hard to use	0

Comments made by test participants suggest that the somewhat lower scores for usefulness and usability can be attributed to the increased level of tool complexity as more features were added in the later versions of the tool. With more analysis capabilities provided, it took more time to train participants how to use the tool, and they had to learn some new, unique dialogs for defining graphic user interfaces (in the second test) and function key function assignments (in the third test). A few bugs, particularly in the third test, also affected participant use of the tool and have since been resolved.

Some participant comments include:

- "The design is user-friendly. The potential for this tool is very powerful." (Test 1.)
- "To add features such as the proximity of controls and how clustered they are. Perhaps include the hand movement, body physical movement, visual movement (movement of head to see something)." (Test 1.)
- "(Add) display screen analysis capability." (Test 1. Feature added before Test 2.)
- "(Add) print out reports of exceptions." (Test 1. Final version provides printable report.)
- "I feel that the tool has the potential to greatly improve the design of HMI (Human Machine Interface)." (Test 1.)
- "I think it is a very useful tool." (Test 2.)
- "More on-screen dialog (instructions) would be helpful or a users manual containing instructions." (Test 2. Enhanced prompting provided for Test 3, and user's manual provided for field test.)
- "The option for importing photos of equipment to analyze appears to be very helpful." (Test 2.)
- "Generally, I think the tool is fairly intuitive once people have received a demonstration and corresponding opportunity for practice. I also think there is a risk of people forgetting how to use some functions if they aren't required to practice in the field once they've received an initial briefing." (Test 3.)
- "Needs: being able to go back." And, "Should be able to correct individual entries without affecting everything." (Test 3. Final version allows a user to go back and edit any step in a tabbed dialog pane that provides direct access to every analysis step.)
- "(Needs) being able to print out a report." (Test 3. Final version of the tool provides printable report.)
- "Generally very good, remove bugs as discussed." (Test 3.)

After completing the third test in September 2006, the tool was made available to FRA regulators for individual testing in a field test.

4. Future Enhancements

As part of each of the three user tests, user comments and requests for additional features users would find useful were collected. A number of these suggestions have been incorporated into the existing tool. Other requests may be met in a future version of the tool. The following discusses some of the more significant recommendations or suggestions.

4.1 Analysis of Control and Display Proximities

One test participant suggesting adding analysis of "…features such as the proximity of controls and how clustered they are. Perhaps include the hand movement, body physical movement, visual movement (movement of head to see something)." Another participant suggested that "switch reaching and location" information should be included in the tool.

Several possible means of addressing these requests exist. First, an ability could be developed within the tool to allow users to select two adjacent controls and ask the tool to report the minimum edge-to-edge separation for that control pair based on human factors guidelines, such as those by the Department of Defense (1999). For example, two pushbuttons may be closer together than two knobs because of the thumb's positioning between the two knobs during rotation. Second, control orientation data already entered by the user may be used to support analysis of allowable head and body movement (Multer, Rudich, & Yearwood, 1998). Third, heuristics may be added to the tool to support analysis of control and display groupings, based on individual control and display proximities compared to the average proximities across the interface. After the user has defined the controls and displays, the tool could guide the user through a set of dialogs in which the he/she reports meaningful relationships between control and display functions. The tool could compare these relationships to the actual interface groupings to determine if the groupings are appropriate. Finally, the relationship between hand location necessary to operate a control and the associated display(s) could be analyzed to determine if the hand position might obscure needed display information or feedback.

4.2 Analysis of Separate Interface Parts

Another question that came up in testing was whether or not the tool could handle multiple displays within a single interface, such as a dispatcher's workstation where he/she may have three to four displays around a workstation. One way to implement this feature would be to take a single picture of the entire workstation and treat it as a single interface. The individual elements of the interface, however, may be too small to be identified and/or labeled by the user. Alternatively, the tool could allow the user to define separate panels or displays within a single interface, and the tool could stitch or link the different displays/panels through an orientation dialog or other means. For example, when the user loads an interface picture, he/she would indicate, on the orientation graphic, which orientation panels apply to the picture currently shown. If the picture is of a single panel with a single orientation, the tool would assume that all the controls and displays defined for that picture would have the same orientation and that orientation question could be dropped from the individual control and display dialogs for that picture. This would streamline the physical form dialog. If the picture contains two or three orientations, the orientation step for the individual controls and displays would still be needed, but the choices could be limited to the ones initially specified. This might make it possible to support keyboard shortcuts for this step.

4.3 Addition of an Undo Function

Two users in the third test suggested providing an undo function. To address this need, the tool could save the state of the properties database with each user entry and allow the user to revert back to the immediately preceding state to undo an undesired change.

4.4 Ability to Update Tool Capabilities

A user in the first test suggested that the tool should "update controls with new control types as [they are] developed." This is important since technology is constantly changing, and potential exists for new control and display types to be introduced into railroad equipment and systems. One way this could be done is to implement a dialog where a user defines a new control or display–a switch that is not in the current tool data base, for example. The user could take a picture or screenshot of the new switch, define it as a type of toggle switch, and upload it into the tool so that it appears in the list of available switch types.

Beyond this, the tool currently has hooks built into the software to handle control types not currently used in most railroad systems, such as modal controls (those whose effect varies in different system, device, or control modes). While modal controls have caused significant problems in aviation, including several accidents (Parasuraman & Riley, 1997), they have not been widely used (yet) in railroad systems. As new systems such as PTC grow in features and capabilities, however, and the number of functions outgrows the user interface, it is possible that modal controls will find their way into railroad technologies. While the tool dialogs currently support the definition of modal controls, the analysis of modal control effects is not yet fleshed out because how these controls will be used in railroad systems is currently unknown. In this way, the tool analysis capabilities can grow with evolving technology without necessarily making the user's interaction with the tool more complex.

Finally, the tool might also support user defined control and display types. For example, if a completely new type of control is developed and it cannot be classified as a switch, knob, button, or any of the other existing control types, the tool might provide a dialog to build a new class of control type.

4.5 Multifunction Controls

Some RCL devices have a pushbutton that resets the alerter when pressed and released, and it deploys sand when pressed and held for more than 2 seconds. Additionally, in some RCL devices, a toggle switch may have one surface-level function and another function when pulled out of its normal plane. To allow multifunction controls, the tool would provide the ability to select multiple function logic options for a control and would analyze the relevant controls for each function they perform.

4.6 Incorporation of Industry-Specific Cultural Conventions

Some controls and displays used in the railroad industry are unique in their expected behaviors, such as the throttle control that moves back for more power and forward for less. Currently, the tool would identify this as a cultural convention violation, even though it is an accepted design within the industry. Defining a set of industry-specific controls and displays with unique conventions would enable the tool to avoid flagging controls and displays that conform to industry conventions as violating conventions; more importantly, it would enable the tool to correctly identify cases where a control or display does not conform to the industry standard.

4.7 Scanning Sequences

Certain displays should be scanned periodically, and controls and displays that are closely related should be placed in the same groups. The analysis of groupings was discussed above. To support reasoning about scanning sequences, a checkbox could be provided for each display or display element for the user to indicate whether the display requires periodic scan. In the analysis stage, the tool would include these displays among the list of critical controls and displays that should be located within the central region of the interface, and it would alert the user if any scanning items are outside this area.

4.8 Control/Display Position in Relation to the Operator

Guidelines (Multer, Rudich, and Yearwood, 1998) exist regarding where displays and controls should be located in relation to the operator based on importance and what the sequence of operations should be. In addition, some test participants from the Office of Safety have asked for guidance on reach distances and angles. Stitching together multiple pictures into a single interface representation, as described in Section 4.2, would enable the tool to support reasoning about reach angles and whether controls and displays are located appropriately for their importance.

4.9 General Ease of Use Improvements

Additional ease of use improvements include:

- Building more intelligence into the tool so that it recognizes that the ON label corresponds to the on function state, FORWARD corresponds to forward, and so forth. The tool could also recognize synonyms, such as activate, start, and enable, for on. This would help streamline the function dialog and remove or automate what is probably the most confusing step.
- Implementing the help function. A Help menu item is currently in the menu bar, but the help files have not yet been built.
- Improving screen-by-screen prompting. The tool currently provides some prompting, but it is currently just a first pass, and the wording has not been refined.
- Improving the analysis reports. Again, the wording in these reports is a first pass and should be refined.
- Replacing the control and display graphics with photographs to make the intended control and display choices more clear.

- Providing definitions of the various control and display types, as well as other pop-up kinds of help. The tool currently provides tool-tips for some items, such as the function logic choices, but other lists are implemented in a way that does not allow for tool-tips.
- Adding keyboard shortcuts so more choices could be made from the keyboard and less mouse manipulation is required.
- Adding color name labels in the color palettes for users with color deficient vision.
- Improving the tool’s overall graphic presentation so it looks and feels more like a finished product.

4.10 Web Version

One other area that some users have mentioned for improvement is to make a Web-based version of the tool so multiple users can collaborate on a joint analysis. This might also make it more efficient for a manufacturer to perform an analysis of its own product and share the resulting analysis file with FRA. The .NET framework in which the tool is implemented is intended to be Web friendly, so this should not require a whole-scale rewriting of the software. In addition, it might make the tool more portable between operating systems.

4.11 Availability to Industry

One of the goals for the tool has been to make it available to industry. This would allow equipment designers and manufacturers to evaluate their own concepts during development and, hopefully, identify potential issues and concerns before the equipment reaches production. It would also allow railroad operators to evaluate equipment they procure from system vendors. It is hoped that, in this way, the tool can help ensure that railroad equipment is safer and less error-prone before FRA gets involved. This would lead to safer equipment overall and reduce the workload of FRA personnel when evaluating new systems, equipment, and technologies.

5. References

Code of Federal Regulations. (2005). *49 CFR § 236 Appendix E to Part 236—Human-Machine Interface (HMI) Design.* Washington, DC: U.S. Government Printing Office.

Department of Defense (1999). *Department of Defense design criteria standard: Human engineering.* MIL-STD-1472F.

Federal Railroad Administration. (2006). *Safety of remote control locomotive (RCL) operations: Final report*. Washington, DC: Author. Retrieved February 8, 2007, from www.fra.dot.gov/downloads/safety/05_007775finalreport_RCL.pdf.

Fitts, P., & Jones, R. (1947b). *Psychological aspects of instrument display. I: Analysis of 270 "pilot-error" experiences in reading and interpreting aircraft instruments.* Memorandum report TSEA4-694-12A, Aero Medical Laboratory, Dayton, OH.

Fitts, P., & Jones, R. (1947a). *Analysis of factors contributing to 460 "pilot error" experiences in operating aircraft controls.* Memorandum report TSEA4-694-12, Aero Medical Laboratory, Dayton, OH.

Multer, J., Rudich, R., & Yearwood, K. (1998). *Human factors guidelines for locomotive cabs*. Washington, DC: Federal Railroad Administration Technical Report No. DOT/FRA/ORD-98/03.

Norman, D. (1981). Categorization of action slips. *Psychological Review*, 88, 1-15.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors, 39(2),* 230-253.

Appendix A.
Functional Requirements

Appendix A presents the functional requirements generated as the basis for the tool design. The requirements also served as a primary means of communication between the human factors engineers and software developers in this program. At its most basic level, the tool contains four primary functions:

1. Define new interface
2. Edit existing interface
3. Analyze interface
4. Compare two interfaces

In addition, the functional requirements specification contains a set of situation awareness parameters (5) used to develop the tool's situation awareness analysis capability. This Appendix provides these parameters for the reader's reference. Each function is expanded below in the functional requirements specification, along with the situation awareness parameters.

Functional requirements

1. Define new interface

    1.1. Name interface/system (set up file or database)

    1.2. Insert/remove underlying layout graphic (optional) (always available)

    1.3. Define location in relation to user

        1.3.1. Graphic of user

        1.3.2. Analyst positions thumbnail of interface in relation to user

    1.4. Add display or display element

        1.4.1. Define type

            1.4.1.1. Alphanumeric readout

            1.4.1.2. Aural

            1.4.1.3. Digital

            1.4.1.4. Graphic

            1.4.1.5. Indicator light

            1.4.1.6. Indicator cluster

            1.4.1.7. Meter/gauge

        1.4.2. Define graphic representation of hardware displays

            1.4.2.1. Meter/gauge only:

1.4.2.1.1. Shape (shape chart based on type)

1.4.2.2. All except aural

1.4.2.2.1. Color (color chart)

1.4.2.2.2. Size (small, medium, large)

1.4.2.2.3. Mounting orientation (horizontal, vertical)

1.4.3. Define graphic representation of electronic display elements

1.4.3.1. Full time or part time?

1.4.3.1.1. If part time, name associated file(s)

1.4.3.2. Designate graphic regions

1.4.3.2.1. Ask if multiple regions

1.4.3.2.2. If no, define background color for whole display (color chart)

1.4.3.2.3. If yes, click and drag to designate regions

1.4.3.2.4. Define background color (color chart or pick color?)

1.4.3.2.5. Continue until click OK

1.4.3.3. Designate graphic objects

1.4.3.3.1. Click and drag to box over large objects

1.4.3.3.2. Define color (color chart)

1.4.3.3.3. Click to designate small objects

1.4.3.3.4. Define color (color chart)

1.4.4. Add display function

1.4.4.1. Select display or graphic object

1.4.4.2. Define name or select from list of prior functions

1.4.4.3. Add label (enter name or select hidden)

1.4.4.4. Add units (enter names or accept default of none)

1.4.4.5. Define logic (radio buttons)

1.4.4.5.1. Permanent

1.4.4.5.2. Multistate

1.4.4.5.3. Modal

1.4.4.5.4. Pop-up

1.4.4.6. Define criticality (radio buttons)

1.4.4.6.1. Low

1.4.4.6.2. Medium

1.4.4.6.3. High

1.4.4.6.4. Critical

1.4.4.7. For graphic objects

1.4.4.7.1. Define object states

1.4.4.7.1.1. Enter names (list)

1.4.4.8. Define dynamic behavior

1.4.4.8.1. Select type (radio buttons)

1.4.4.8.1.1. on/off

1.4.4.8.1.2. less/more

1.4.4.8.1.3. forward/back

1.4.4.8.1.4. faster/slower

1.4.4.8.1.5. up/down

1.4.4.8.1.6. right/left

1.4.4.8.1.7. other

1.4.4.8.2. Meter/gauge only

1.4.4.8.2.1. Select motion relationships

1.4.4.8.3. Graphic, indicator light, indicator cluster, meter/gauge

1.4.4.8.3.1. Identify display position or state corresponding to each type state (radio buttons)

1.4.4.8.4. Graphic, meter/gauge

1.4.4.8.4.1. Identify display position layout (diagram)

1.4.4.9. Select SA parameters (check boxes: see list)

1.4.4.10. Identify related control(s)

1.4.4.10.1. Click on control(s) or select "none" or "out of view"

1.4.4.11. Identify related display(s)

1.4.4.11.1. Click on display(s) or select "none," "aural," or "out of view"

1.5. Add control

1.5.1. Define type

1.5.1.1. Control pad

1.5.1.2. Cursor control device

1.5.1.3. Joystick

1.5.1.4. Keyboard

1.5.1.5. Keypad

1.5.1.6. Knob

1.5.1.7. Lever

1.5.1.8. Palm button

1.5.1.9. Pedal

1.5.1.10. Pushbutton

1.5.1.11. Rocker switch

1.5.1.12. Rotary selector

1.5.1.13. Slider

1.5.1.14. Soft function key

1.5.1.15. Speech

1.5.1.16. Switch

1.5.1.17. Thumbwheel

1.5.1.18. Touch screen button

1.5.1.19. On-screen

1.5.1.19.1. Button

1.5.1.19.2. Check box

1.5.1.19.3. Command line

1.5.1.19.4. Dialog box

1.5.1.19.5. Graphic

1.5.1.19.6. Icon

1.5.1.19.7. Menu

1.5.1.19.8. Radio buttons

1.5.2. Define graphic representation

1.5.2.1. Shape (shape chart based on type)

1.5.2.2. Color (color chart)

1.5.2.3. Size (small, medium, large)

1.5.2.4. Orientation (horizontal, vertical)

1.5.2.5. Protected? (yes/no)

1.5.3. Add control function

1.5.3.1. Define name or select from list of prior functions

1.5.3.2. Add label (enter name or select hidden)

1.5.3.3. Add units or positions (enter names or select none)

1.5.3.4. Define logic

1.5.3.4.1. Surface

1.5.3.4.2. Multistep

1.5.3.4.3. Press and hold

1.5.3.4.4. Press and release

1.5.3.4.5. Modal

1.5.3.5.Define criticality (low, medium, high, critical)

1.5.3.6.Define dynamic behavior

1.5.3.6.1. Select smooth/stepped (radio buttons)

1.5.3.6.2. Select type (radio buttons)

1.5.3.6.2.1.on/off

1.5.3.6.2.2.less/more

1.5.3.6.2.3.forward/back

1.5.3.6.2.4.faster/slower

1.5.3.6.2.5.up/down

1.5.3.6.2.6.right/left

1.5.3.6.2.7.other

1.5.3.6.2.7.1. Define states (list)

1.5.3.7.Identify relationships between type state and control position (radio buttons)

1.5.3.8.Identify control label positions (diagram)

1.5.3.9.Select SA parameter(s) (check boxes) (see list)

1.5.3.10.Identify related control(s)

1.5.3.10.1. Click on control(s) or select “none” or “out of view”

1.5.3.11.Identify related display(s)

1.5.3.11.1. Click on display(s) or select “none,” “aural,” or “out of view”

1.5.3.11.2. Define whether display provides direct feedback of control input

1.6. Define destructive interrelationships

1.6.1. For every instance of adjacent controls of same type and physical form:

1.6.1.1.“Could it cause a problem if you intend to select (control A function) but select (control B function) instead?”

1.6.1.2.“Could it cause a problem if you intend to select (control B function) but select (control A function) instead?”

1.7. Add automation

1.7.1. Define name

1.7.2. Define function(s) (new or select from list of prior functions)

1.7.2.1.Select SA parameter (see list)

1.7.2.2.Select automation level (see list)

1.7.2.3.Select related display(s)

1.7.2.4.Select related control(s)

1.7.2.4.1. Used when automation is on?

1.7.2.5.Define states

1.7.2.5.1. Feedback?

1.7.2.5.2. Define state transitions

1.7.2.5.2.1. Feedback?

2. Edit existing interface

2.1. Rename interface/system (set up new file or database based on existing one)

2.2. Redefine location in cab

2.3. Insert/remove underlying layout graphic (optional) (always available)

2.4. Add display or display element

2.4.1. Define type

2.4.1.1.Alphanumeric readout

2.4.1.2.Aural

2.4.1.3.Graphic

2.4.1.3.1. Edit graphic regions or objects

2.4.1.3.2. Edit functional regions or objects

2.4.1.4.Indicator light

2.4.1.5.Indicator cluster

2.4.1.6.Meter/gauge

2.4.1.7.Numeric readout

2.4.2. Define graphic representation

2.4.2.1.Shape (shape chart based on type)

2.4.2.2.Color (color chart)

2.4.2.3.Size (small, medium, large)

2.4.2.4.Orientation (horizontal, vertical)

2.4.2.5.Position in relation to user

2.4.3. Add display function

2.4.3.1.Define name or select from list of prior functions

2.4.3.2.Add label (name or hidden)

2.4.3.3.Add units or positions (names or hidden)

2.4.3.4.Define logic

2.4.3.4.1. Surface

2.4.3.4.2. Multistep

2.4.3.4.3. Modal

2.4.3.4.4. Pop-up

2.4.3.5.Define criticality (low, medium, high)

2.4.3.6.Define dynamic behavior

2.4.3.6.1. Select continuous/discrete

2.4.3.6.2. Select motion relationships (increase/decrease, left-right/up-down)

2.4.3.7.Select SA parameter (see list)

2.4.3.8.Identify related control(s)

2.4.3.9.Identify related displays(s)

2.4.3.9.1. Identify where feedback is provided

2.5. Modify display or display element

2.5.1. Select display element to be modified

2.5.2. Redefine type

2.5.2.1. Alphanumeric readout

2.5.2.2.Aural

2.5.2.3.Graphic

2.5.2.3.1. Redefine graphic regions or objects

2.5.2.3.2. Redefine functional regions or objects

2.5.2.4.Indicator light

2.5.2.5.Indicator cluster

2.5.2.6.Meter/gauge

2.5.2.7.Numeric readout

2.5.3. Redefine graphic representation

2.5.3.1.Shape (shape chart based on type)

2.5.3.2.Color (color chart)

2.5.3.3.Size (small, medium, large)

2.5.3.4.Orientation (horizontal, vertical)

2.5.3.5.Position in relation to user

2.5.4. Modify display function

2.5.4.1.Redefine name or select from list of prior functions

2.5.4.2.Modify label (name or hidden)

2.5.4.3.Modify units or positions (names or hidden)

2.5.4.4.Redefine logic

2.5.4.4.1. Surface

2.5.4.4.2. Multistep

2.5.4.4.3. Modal

2.5.4.4.4. Pop-up

2.5.4.5.Redefine criticality (low, medium, high)

2.5.4.6.Redefine dynamic behavior

2.5.4.6.1. Select continuous/discrete

2.5.4.6.2. Select motion relationships (increase/decrease, left-right/up-down)

2.5.4.7.Modify SA parameter (see list)

2.5.4.8.Modify related control(s)

2.5.4.9.Modify related display(s)

2.6. Add control

2.6.1. Define type

2.6.1.1.Control pad

2.6.1.2.Cursor control device

2.6.1.3.Joystick

2.6.1.4.Keyboard

2.6.1.5.Keypad

2.6.1.6.Knob

2.6.1.7.Lever

2.6.1.8.Multi-position switch

2.6.1.9.Palm button

2.6.1.10.Pedal

2.6.1.11.Pushbutton

2.6.1.12.Rocker switch

2.6.1.13.Rotary selector

2.6.1.14.Slider

2.6.1.15.Soft function key

2.6.1.16.Speech

2.6.1.17. Thumbwheel

2.6.1.18. Toggle switch

2.6.1.19. Touch screen button

2.6.1.20. On-screen

2.6.1.20.1. Button

2.6.1.20.2. Check box

2.6.1.20.3. Command line

2.6.1.20.4. Dialog box

2.6.1.20.5. Graphic

2.6.1.20.6. Icon

2.6.1.20.7. Menu

2.6.1.20.8. Radio buttons

2.6.2. Define graphic representation

2.6.2.1. Shape (shape chart based on type)

2.6.2.2. Color (color chart)

2.6.2.3. Size (small, medium, large)

2.6.2.4. Orientation (horizontal, vertical)

2.6.2.5. Position in relation to user

2.6.2.6. Protected?

2.6.3. Add control function

2.6.3.1. Define name or select from list of prior functions

2.6.3.2. Add label (name or hidden)

2.6.3.3. Add units or positions (names or hidden)

2.6.3.4. Define logic

2.6.3.4.1. Surface

2.6.3.4.2. Multistep

2.6.3.4.3. Press and hold

2.6.3.4.4. Modal

2.6.3.5. Define criticality (low, medium, high)

2.6.3.6. Define dynamic behavior

2.6.3.6.1. Select continuous/discrete

2.6.3.6.2. Select motion relationships

2.6.3.6.2.1. Knob

2.6.3.6.2.1.1. Increase/decrease, clockwise/counterclockwise

2.6.3.6.2.2. Lever

2.6.3.6.2.2.1. Increase/decrease, forward/back, up/down, left/right

2.6.3.6.2.3. Pedal

2.6.3.6.2.3.1. Increase/decrease, up/down

2.6.3.6.2.4. Pushbutton

2.6.3.6.2.4.1. Active/inactive, in/out

2.6.3.6.2.5. Rocker switch

2.6.3.6.2.5.1. Side, state

2.6.3.6.2.6. Slider

2.6.3.6.2.6.1. Increase/decrease, forward/back, up/down, left/right

2.6.3.6.2.7. Thumbwheel

2.6.3.6.2.7.1. Increase/decrease, forward/back, up/down, left/right

2.6.3.6.2.8. Toggle switch

2.6.3.6.2.8.1. Active/inactive, forward/back, up/down, left/right

2.6.3.7.Select SA parameter (see list)

2.6.3.8.Identify related display(s)

2.6.3.9.Identify related control(s)

2.7. Modify control

2.7.1. Select control to be modified

2.7.2. Redefine type

2.7.2.1.Control pad

2.7.2.2.Cursor control device

2.7.2.3.Joystick

2.7.2.4.Keyboard

2.7.2.5.Keypad

2.7.2.6.Knob

2.7.2.7.Lever

2.7.2.8.Multi-position switch

2.7.2.9.Palm button

2.7.2.10.Pedal

2.7.2.11.Pushbutton

2.7.2.12.Rocker switch

2.7.2.13. Rotary selector

2.7.2.14. Slider

2.7.2.15. Soft function key

2.7.2.16. Speech

2.7.2.17. Thumbwheel

2.7.2.18. Toggle switch

2.7.2.19. Touch screen button

2.7.2.20. On-screen

2.7.2.20.1. Button

2.7.2.20.2. Check box

2.7.2.20.3. Command line

2.7.2.20.4. Dialog box

2.7.2.20.5. Graphic

2.7.2.20.6. Icon

2.7.2.20.7. Menu

2.7.2.20.8. Radio buttons

2.7.3. Redefine graphic representation

2.7.3.1. Shape (shape chart based on type)

2.7.3.2. Color (color chart)

2.7.3.3. Size (small, medium, large)

2.7.3.4. Orientation (horizontal, vertical)

2.7.3.5. Position in relation to user

2.7.3.6. Protected?

2.7.4. Modify control function

2.7.4.1. Redefine name or select from list of prior functions

2.7.4.2. Modify label (name or hidden)

2.7.4.3. Modify units or positions (names or hidden)

2.7.4.4. Redefine logic

2.7.4.4.1. Surface

2.7.4.4.2. Multistep

2.7.4.4.3. Press and hold

2.7.4.4.4. Modal

2.7.4.5. Redefine criticality (low, medium, high)

2.7.4.6.Redefine dynamic behavior

2.7.4.6.1. Select continuous/discrete

2.7.4.6.2. Select motion relationships

2.7.4.6.2.1. Knob

2.7.4.6.2.1.1. Increase/decrease, clockwise/counterclockwise

2.7.4.6.2.2. Lever

2.7.4.6.2.2.1. Increase/decrease, forward/back, up/down, left/right

2.7.4.6.2.3. Pedal

2.7.4.6.2.3.1. Increase/decrease, up/down

2.7.4.6.2.4. Pushbutton

2.7.4.6.2.4.1. Active/inactive, in/out

2.7.4.6.2.5. Rocker switch

2.7.4.6.2.5.1. Side, state

2.7.4.6.2.6. Slider

2.7.4.6.2.6.1. Increase/decrease, forward/back, up/down, left/right

2.7.4.6.2.7. Thumbwheel

2.7.4.6.2.7.1. Increase/decrease, forward/back, up/down, left/right

2.7.4.6.2.8. Toggle switch

2.7.4.6.2.8.1. Active/inactive, forward/back, up/down, left/right

2.7.4.7.Modify SA parameter (see list)

2.7.4.8.Redefine related display(s)

2.7.4.8.1. Identify where feedback is provided

2.7.4.9.Redefine related control(s)

2.8. Define or modify destructive interrelationships (function matrix)

2.9. Add or modify automation

2.9.1. Define name

2.9.2. Define function(s) (new or select from list of prior functions)

2.9.2.1.Select SA parameter (see list)

2.9.2.2.Select automation level (see list)

2.9.2.3.Select related display(s)

2.9.2.4.Select related control(s)

2.9.2.4.1. Used when automation is on?

2.9.2.5.Redefine states

2.9.2.5.1. Feedback

2.9.2.5.2. Redefine state transitions

2.9.2.5.2.1. Feedback

3. Analyze single interface

3.1. Select interface to be analyzed

3.2. Select analysis type or full report

3.2.1. Potential control selection errors

3.2.1.1.Adjacent controls with destructive functions

3.2.1.1.1. Adjacent controls (any direction), same group, same type, same shape, same color, same size, same orientation, potentially destructive functions (either direction)

3.2.1.2.Symmetrical controls with destructive functions

3.2.1.2.1. Symmetrical controls, same type, same shape, same color, same size, same orientation, potentially destructive functions

3.2.2. Potential control movement errors

3.2.2.1.Movement direction violates cultural convention

3.2.2.1.1. Up/forward/clockwise to increase/high/turn on, down/back/counterclockwise to decrease/low/turn off

3.2.2.2.Movement direction inconsistent with spatial mapping in relation to operator

3.2.2.2.1. Forward, back, up, down, right, left, early, late

3.2.2.3.Movement direction inconsistent with other controls for same function type

3.2.3. Hidden functions (press-and-hold logic)

3.2.4. Unlabeled control positions

3.2.5. Inadvertent activation

3.2.5.1.Critical functions with unprotected controls

3.2.6. Number of function states inappropriate for control type

3.2.6.1.Control pad: 4

3.2.6.2.Knob: maximum of 3 ganged without gloves, 1 with

3.2.6.3.Palm button: 2

3.2.6.4.Pushbutton: 2

3.2.6.5.Rocker switch: 3

3.2.6.6.Rotary selector: 12

3.2.6.7.Toggle switch: 3

3.2.6.8.On-screen

3.2.6.8.1. Check box: 2

3.2.6.8.2. Radio buttons: 2

3.2.7. Appropriateness of control design and placement for importance

3.2.7.1.Control location

3.2.7.1.1. Location within group appropriate for importance

3.2.7.1.1.1.Most important most accessible, first in group

3.2.7.2.Color

3.2.7.2.1. Red: critical/warning

3.2.7.2.2. Yellow: caution

3.2.7.2.3. Green: safe

3.2.7.3.Size

3.2.7.3.1. Critical or highly important: large (no critical or important control should be small)

3.2.8. Mode errors

3.2.8.1.Potentially destructive functions assigned to same control based on mode

3.2.9. Control distances

3.2.9.1.Select controls distances function from menu

3.2.9.2.Click on control box, box highlights

3.2.9.3.Click on second control, read minimum distance

3.2.9.4.Click on third and more controls, read distance from first control

3.2.9.5.Click on first control again to deselect as the source control

3.2.10. Adjacent display elements with destructive functions

3.2.10.1.Adjacent displays (any direction), same group, same type, same shape, same color, same size, same orientation, potentially destructive functions (either direction)

3.2.11. Symmetrical display elements with destructive functions

3.2.11.1.Symmetrical displays, same type, same shape, same color, same size, same orientation, potentially destructive functions

3.2.12. Display direction coding

3.2.12.1. Direction coding violates cultural conventions

3.2.12.1.1. Up/forward/clockwise to increase/high/turn on, down/back/counterclockwise to decrease/low/turn off

3.2.12.2. Direction inconsistent with spatial mapping in relation to operator

3.2.12.2.1. Forward, back, up, down, right, left, early, late

3.2.12.3. Direction inconsistent with related controls

3.2.12.4. Direction inconsistent with related displays

3.2.13. Units not displayed

3.2.14. No label for displayed information

3.2.15. Display color coding

3.2.15.1. Red: stop, critical

3.2.15.2. Yellow: caution

3.2.15.3. Green: go, safe

3.2.16. Other display color issues

3.2.16.1. Contrast (blue on black, yellow on white)

3.2.16.2. Blue text

3.2.16.3. Color 3D effects (red on blue, blue on red)

3.2.17. Size of display elements

3.2.17.1. Important displays or elements should not be small

3.2.18. Functional logic of displays inappropriate for importance

3.2.18.1. Important elements always displayed (not only on certain pages)

3.2.18.2. Pop-up information adequate

3.2.18.3. Critical information available within one step

3.2.19. Display and display element position inappropriate for importance

3.2.19.1. Critical information within 15 degrees from center

3.2.19.2. Secondary information within 35 degrees from center

3.2.19.3. Most important options located first in lists, menus, fields, or groups

3.2.20. Display element groups

3.2.20.1. Related display elements collected together in same group (note violations where display elements are separated from related elements by unrelated elements)

3.2.21. Inconsistent position mapping of displays or display elements with related displays and controls

3.2.22. Inconsistent placement of same information on different pages

3.2.23. Display/control integration

3.2.23.1. Feedback

3.2.23.2. Grouping

3.2.23.2.1. Functionally related controls and displays grouped

3.2.23.3. Inconsistent position mapping for same function between controls and displays

3.2.23.4. Inconsistent mapping of function keys to display options

3.2.23.5. Hand may obscure display when operating control

3.2.23.5.1. Possible where controls exist above displays

3.2.24. Analyze potential situation awareness gaps

3.2.24.1. Control functions with no related display elements

3.2.24.2. Unassigned SA parameters

3.2.25. Analyze potential automation effects

3.2.25.1. SA parameters assigned to automation with no related controls

3.2.25.2. SA parameters assigned to automation with no related displays

3.2.25.3. SA parameters assigned to automation and related displays with no engineer involvement when automation is on

4. Analyze two interfaces

4.1. Selected interfaces to be analyzed

4.2. Select analysis type or full report

4.2.1. Analyze potential selection errors

4.2.1.1.Same function, different locations

4.2.1.2.Same location and control type, different functions

4.2.1.3.Same function and control type, different activation logic

4.2.1.3.1. Movement direction

4.2.1.3.2. Modes

4.2.1.3.3. Steps

4.2.1.3.4. Press and hold

4.2.2. Analyze delay potentials

4.2.2.1.Same function, different control types

4.2.2.2.Same function, different labels

4.2.2.3.Same function, different graphical representation

4.2.3. Analyze automation differences

4.2.3.1.Same function, different levels (over-reliance on lower automation system)

5. Situation awareness parameters

5.1. Actual speed

5.2. Allowed speed

5.2.1. Wayside signal

5.2.2. Cab signal

5.2.2.1.On/off

5.2.2.2.In/out of territory

5.2.2.3.Failure conditions

5.2.3. Civil speed

5.2.3.1.Timetable

5.2.4. PTC

5.2.4.1.In/out of territory

5.2.4.2.Failure conditions

5.2.4.3.Modes?

5.2.5. Temporary restrictions

5.2.5.1.Work areas

5.2.5.2.Track condition

5.2.5.3.Other?

5.2.6. Dispatcher permissions

5.2.7. Fusees

5.2.8. Torpedoes

5.2.9. Equipment limits

5.3. Brakes

5.3.1. Dynamic brake status and capacity

5.3.2. Air brake status and capacity

5.3.2.1.Propagation

5.3.2.2.Pressure setting

5.3.2.3.Mode (gradual, full release)

5.3.3. Hand brake status

5.3.4. Parking brake status

5.3.5. Snow brake status

5.3.6. Tests done and status

5.4. Consist

5.5. Power

5.5.1. Locomotive horsepower

5.5.2. Overall power level

- 5.5.3. Distributed power levels

5.6. Weight/trailing tonnage

5.7. Track conditions

- 5.7.1. Current location
- 5.7.2. Grades
- 5.7.3. Curves
- 5.7.4. Configuration
  - 5.7.4.1.Crossovers
  - 5.7.4.2.Sidings
  - 5.7.4.3.Spurs
  - 5.7.4.4.Grade crossings
  - 5.7.4.5.Track junctions
- 5.7.5. Temperature
- 5.7.6. Precipitation

5.8. Slack

5.9. Schedule

- 5.9.1. Station stops
- 5.9.2. Time

5.10.Leave station authorization

5.11.Radio frequency

5.12.End of train (EOT) device status

- 5.12.1. Two-way communications status
- 5.12.2. Brake pressure
- 5.12.3. Mode
- 5.12.4. EOT device emergency braking availability

5.13. Doors opened/closed

5.14. Lights

5.15. Tilting mechanism

5.16. Upcoming automation actions and behaviors

5.17. Train systems integrity

- 5.17.1. Faults
- 5.17.2. Failures

Appendix B.
Human-Centered Technology Tool Users' Manual

Contents

B1. Introduction

Welcome to the human-centered technologies tool (the tool). This version of the tool (1.0) is designed to help you identify potential human factors problems associated with equipment and user interface designs. Such problems may include the potential for:

- Operating a control in the wrong direction or misreading a display due to violations of user expectations or inconsistencies within the interface.
- Operating the wrong control or using the wrong display due to similarities between adjacent controls and displays.
- Inappropriate use of color.
- Unintentional activation of a control that may lead to hazardous consequences.
- Confusion due to lack of feedback from control input.
- Errors due to inconsistent assignment of functions to function keys on different screens.

The tool may be used to analyze a design at any stage of development, from the earliest concept drawings to a finished product. It may be used for workstations with panels that the user sits or stands in front of, such as in a locomotive cab or dispatcher station, or it may be used for devices that the user holds or wears, such as RCL devices.

B2. General Operation

To start the tool, double-click on the tool icon, or click on Start, All Programs, HC Tool. A screen will come up with several options:

- To start the analysis for a new interface, select Define New Interface.
- To continue the analysis of an interface you have already started and saved to a file, select Edit Existing Interface.
- To analyze an interface that you have completed defining and have saved to a file, select Analyze Interface.

The other two options are grayed out. They will be available in future versions of the tool.

B2.1 Starting Out

An interface is defined by telling the tool what kinds of displays and controls are used, where they are located in relation to each other, what functions they perform, and how they perform those functions. You do this by following a sequence of five **steps**. You select the step you want to perform through the pointed buttons at the top of the screen (1. Picture, 2. Layout, etc.). The tool guides you through each step for every control and display in the design. The steps should be completed roughly in order:

1. Picture: Allows you to upload a picture of the interface you wish to analyze to serve as a guide for defining the controls and displays.
2. Layout: Allows you to define the types and locations of controls and displays.
3. Appearance: Allows you to define what each control and display looks like and how it is oriented.
4. Function: Allows you to define what each control and display does and how.
5. Analyze: Allows the tool to analyze the interface and report potential problems to you.

These steps are arranged in the general order you should use. You must define the type and position (layout) of a control or display before you can define its appearance or function.

Figure B-1 shows the tool interface. The area to the left is the user **dialog area** where the tool places options for you to select from and guides you through the process of performing each step. To the right is the **work area**, where you define the interface layout and function. At the top of the work area is a **prompt line**; if you get stuck and are unsure about what to do next, this line should tell you what kind of input the tool is expecting. The **menu bar** with the basic application selections is at the top, and some function buttons are below that (save, copy, paste, help, and delete).

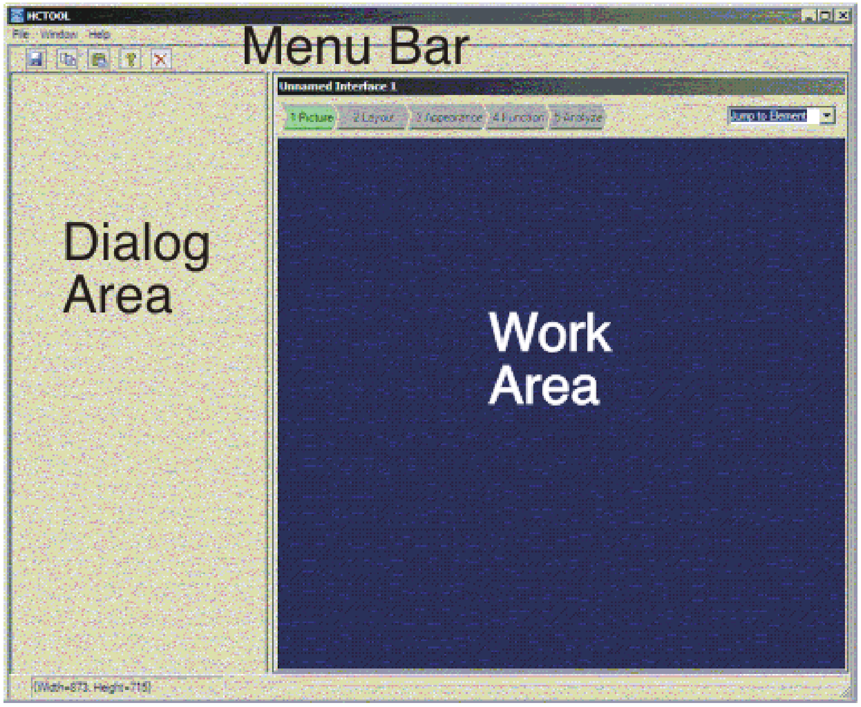


Figure B-1. HC TOOL screen layout

B2.2 Importing a Picture

To help you define the controls and displays layout, you may import a picture of the interface into the work area. This picture can be in any common computer graphic format (jpeg, gif, tiff, etc.). The tool does not "know" anything about this picture; it is only there as a guide for you. Therefore, the picture can be anything from a simple line drawing to a photograph. If you are evaluating a new design before the product or system has been built, the easiest way to import the picture of the design might be to take a screen shot of a manufacturer's prototype (if it is on a computer) or scan a drawing of the interface to a file. You can also use a digital camera to take a photograph of a drawing, a model, or the finished product.

Once you have the image of the interface, in whatever format (scanned, photographed, etc.), on your computer, you may import it into the work area. Select "1. Picture" from the buttons at the top of the work area. This will bring up a Windows file dialog box; find the file you want to use, highlight it, and select "Open" or double-click on it. The tool will import the file into the work area.

Once the image is open in the work area, you can move on to the next step of defining the interface layout.

Example: Importing an image and defining the controls and displays layout

1. Start the tool, and select Define New Interface.
2. Click on the "1. Picture" button at the top of the work area.
3. Double-click on the file named Acela.jpg.

4. A photograph of the right forward panel of the Acela cab should come up in the work area.

B2.3 Controls Layout

Now that you have a picture to work with, you can tell the tool what the various controls on the panels are and where they are located. You will do this by selecting the appropriate control type from a list in the dialog area, then clicking on each control of that type that you see in the picture. To continue with the example above:

1. Select Layout. This brings up a list of available display or control types in the dialog area on the left. At the top of the dialog area are two tabs, one for controls and one for displays. Make sure the "controls" tab is selected so you see the list of controls in the dialog area. (See Figure B-2.) If you look at the Acela picture in the work area, you should see several of the listed controls: two joysticks (for the horn and the alerter), seven rotary switches, three pushbuttons, and a palm button (the large, mushroom-shaped control on the lower part of the panel).
2. To define the positions of the joysticks, select "Joystick" from the list on the left.
3. Click on either of the joysticks you see in the photograph, then the other, in any order.
4. To define the positions of the rotary switches, select "Rotary switch" from the list on the left.
5. Click on each of the rotary switches you see, again in any order.
6. Do the same for the pushbuttons and the palm button.

When you are done with this step, the screen should look like Figure B-2.

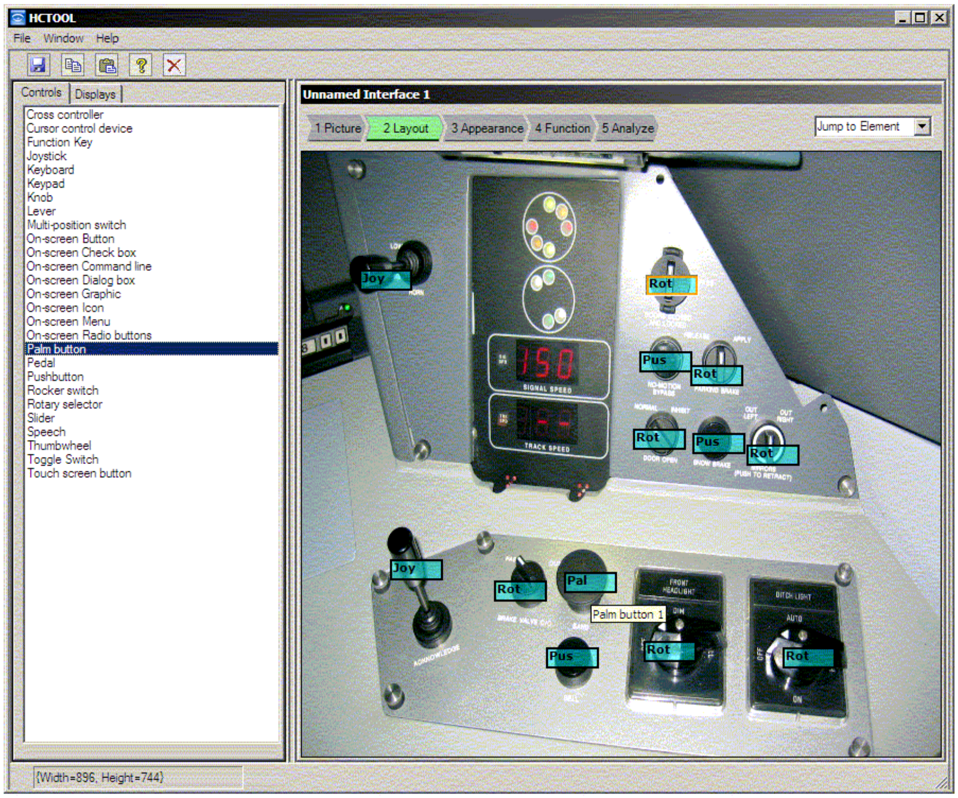


Figure B-2. The Acela right forward cab with the various control types tagged

B2.4 Displays Layout

You can define the layout of displays in the same way. Select the Displays tab at the top of the dialog area to show the list of available display types. Then:

1. From the list on the left, select "Indicator cluster," and click on the in-cab signal display. Since this display consists of two parts but is really a single indicator, click just once between the two circles.
2. Finally, from the list on the left, select "Numeric indicator," and click on both the upper and lower speed indicators (signal speed and track speed). These are two different indicators, so each should get a "tag."

When you are done with this step, the screen should look like Figure B-3.

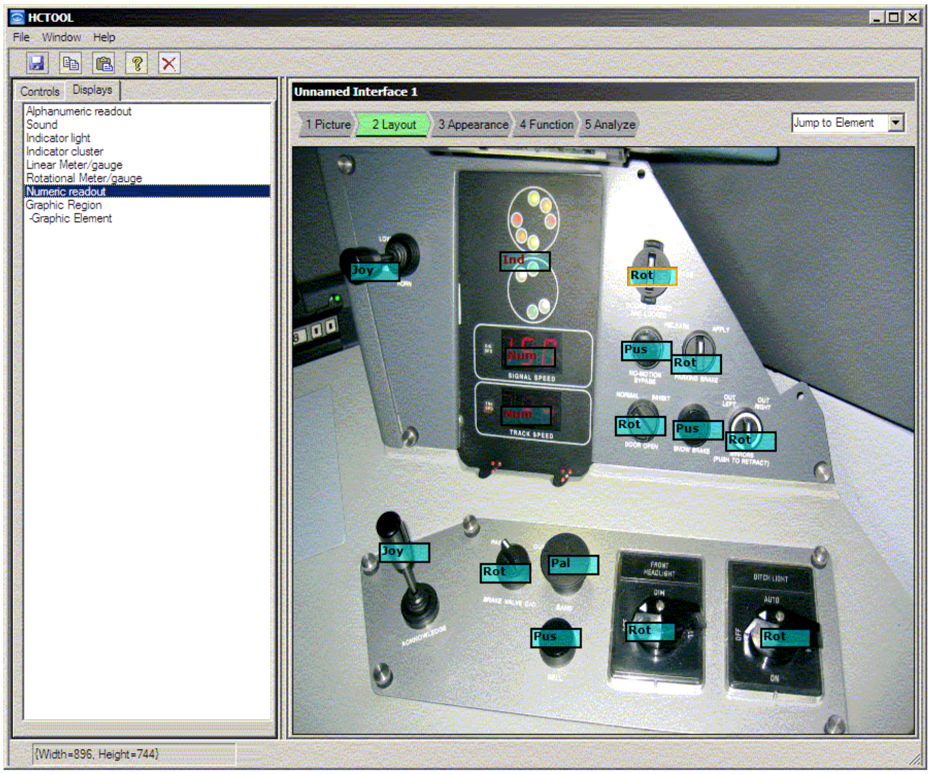


Figure B-3. The Acela panel with controls and displays tagged

This example illustrates the process of defining a new interface, importing a picture to help you define the layout of the interface, and defining the types and positions of controls and displays within the interface. At this point, the tool knows that there are two joysticks to the left of all the other controls and displays, a group of rotary switches and pushbuttons to the right of the displays, and so forth. However, the tool does NOT "know" anything about the picture itself; the "**tags**" that you have placed on the picture, indicating the types and relative positions of the displays and controls, are all the tool knows about the interface at this point.

B2.5 Saving Your Work

If you want, you can now save the work that you have done by selecting the save icon (the disc icon, just like in Microsoft Word) under the menu bar. However, we will not work with this interface any more in this manual, so you can close it out. You can do this by clicking on the black "x" in the upper right of the tool window (just as you would to close any window in Microsoft Windows). You can also select File, Exit. Or, you can select File, Close, which will close out the current analysis and bring you back to the start screen.

B3. Example: Defining a Complete Interface with Controls, Displays, and Function Keys

In this example, we illustrate the process of defining a complete interface. This example will take up most of the rest of this manual, so we provide headers for the various steps for reference.

B3.1 Import the Interface Picture

First, click on the Picture button at the top of the work area. From the file list, select "third test.gif." This is a hypothetical interface layout for a hypothetical PTC system (see Figure B-4). On the left side of the work area, you will see a large, graphic screen with a simple map of the track in front of the train and an icon at the bottom representing the train itself with a number indicating the current speed. To the left of the track is a milepost marker ("MP 122"). To the right of the track is a signal icon. In the upper right is an area of blue text with, again, the current train speed on the top and, below, the next authorized speed ("70 in 10.3," meaning 70 miles per hour will be allowed in 10.3 miles). At the bottom of the screen are three function keys and their associated labels, embedded in a white prompt area.

To the right of the screen, you will see four toggle switches and two sliders. The functions of these controls will be explained later, in the section on the "Controls Functions" step.

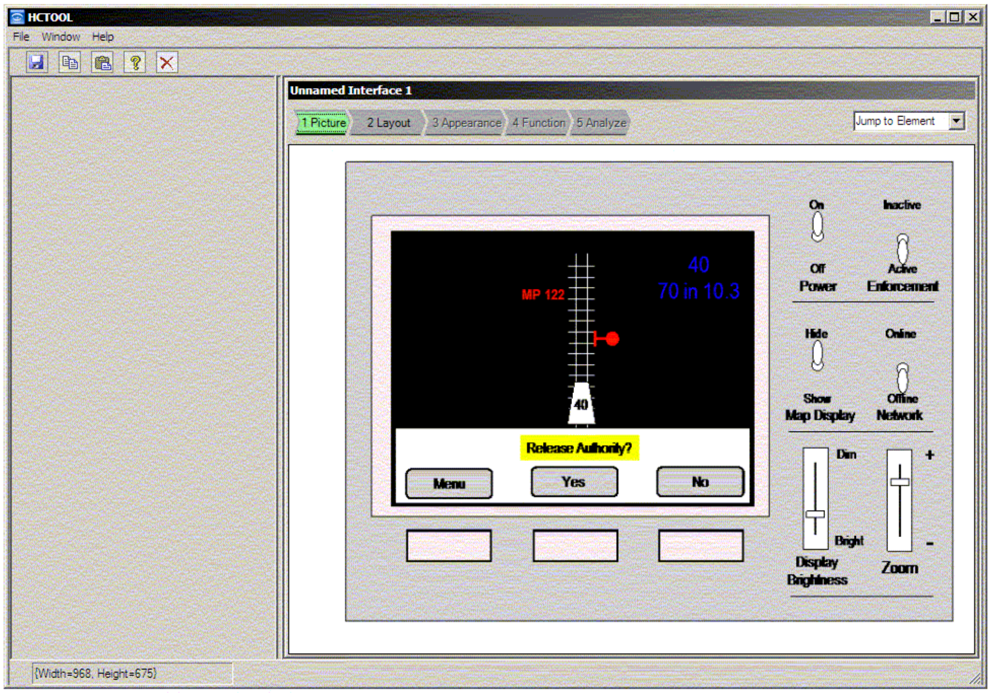


Figure B-4. The toy PTC system interface for this example

B3.2 Define the Layout

In this step, you will tell the tool where the various controls and displays on the panel are located and what types they are.

B3.2.1 Defining Control Type and Placement

To tell the tool where the controls are and what types they are, select “2. Layout,” and make sure the Controls tab is selected. Then, from the list in the dialog area, click on “Toggle switch,” and then click on the four toggle switches you see to the right of the display. This should put tags saying “Togg” over each of the four switches. Then, from the list in the dialog area, click on “Slider” and then click on the two slide controls (for “Display Brightness” and “Zoom”). Tags saying “Slid” should appear over the two sliders. Finally, select “Function key” from the list in the dialog area, then click on the three function keys below the display.

B3.2.2 Placement of Control Tags

It does not matter where on a control you put the tag. However, the tool can reason best if the relationship between the tags is consistent with the relationship between the controls themselves. So, for example, if two controls are adjacent and are even with each other (that is, one is not higher or lower on the panel than the other), the tags should also be even with each other. This will help the tool determine which controls are candidates for confusion based on their proximity. This can be seen in Figure B-5.

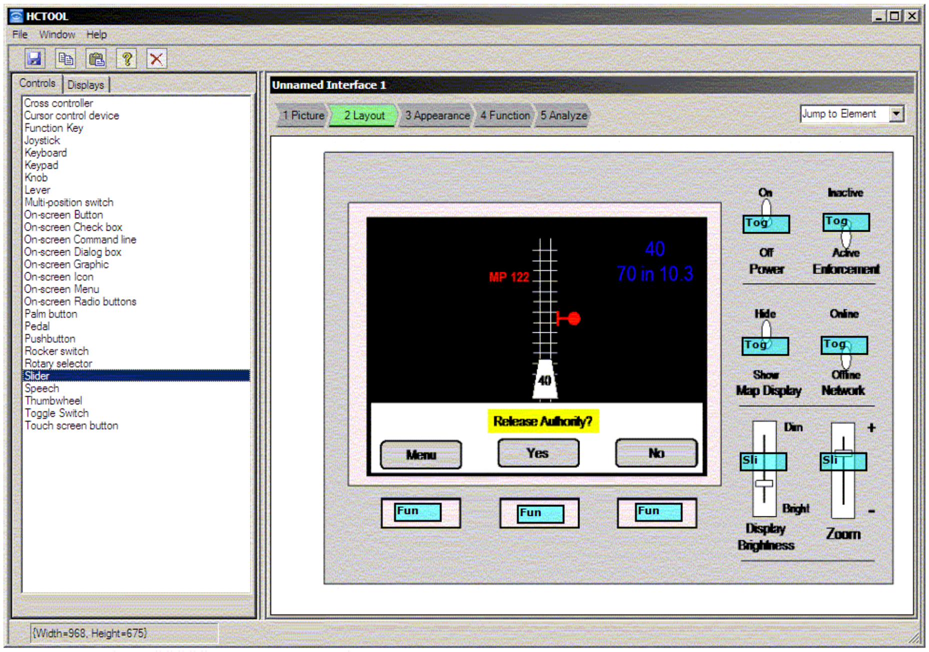


Figure B-5. Control and display tags should be reasonably level with each other

B3.2.3 Delete Control Tags

If you make a mistake, you can correct it by selecting the control tag you want to delete, then selecting the button with the red “X” under the menu bar. This is the same delete button as is used in Windows Explorer to delete files and folders, and it works the same way. If you want to delete other tags, you will have to select each tag and re-select the “delete” for each one you want to delete. This will prevent you from forgetting that the delete function has been selected and deleting other tags by mistake.

When finished defining control layouts, select the Displays tab in the dialog area to show the available display types.

B3.2.4 Defining Display Type and Placement for a Graphical Display

Conventional displays, such as gauges, are tagged just like the controls were in the previous section. This was demonstrated in the prior example of the signal cluster and numeric readouts in the Acela cab. However, graphical displays (such as graphical user interfaces, maps, system diagrams, and so forth) are much more flexible and must be defined in terms of their various **regions** (the broad areas of background color) and **graphic elements** (the individual objects that appear on the display). Because these can vary greatly in size, the tool uses boxes to define their positions. To define the position of either a graphic region or a graphic element, you select the appropriate display type from the dialog area, position the cursor at one corner of the object that you are trying to define, then click and drag over the object to form a box that surrounds it.

If you have not done so already, select the Displays tab in the dialog area to see the available display types.

B3.2.5 To Define a Background area or "Graphic Region"

In the example, there are two **graphic regions** (see Figure B-6). Graphic regions are the background areas of a display, and they typically contain the graphic elements or objects on the display. In this example, there are two graphic regions: a black background taking up most of the display and a white background containing the function key labels.

Select "Graphic region" from the list in the dialog area, position the cursor over one corner of the black background area behind the track map, click the left mouse key, and drag the cursor over the region to the diagonally opposite corner (just to the top of the white region) to enclose the black region in a box. Then click and drag over the white region. You should now have two boxes, one covering each of the major regions of the display.

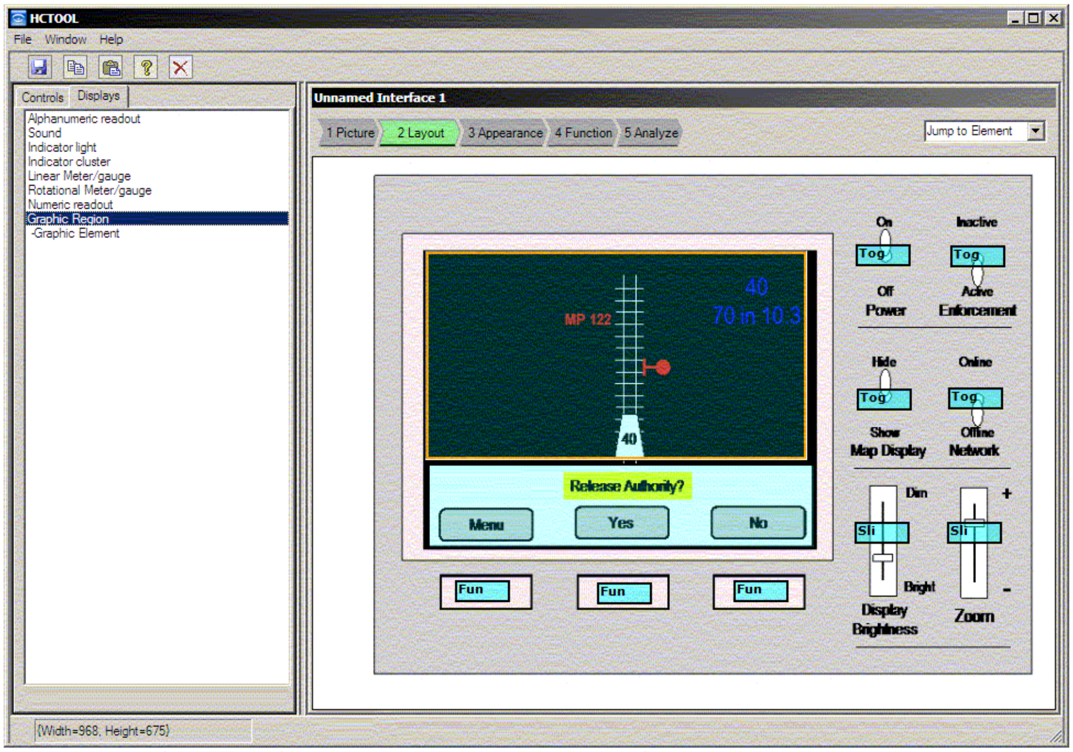


Figure B-6. The first graphic region is shown outlined in orange (because it is selected in this picture), and the second is shown as a turquoise box superimposed over the function key labels (because it is unselected). If you select (click anywhere in) the second graphic region, it will be outlined in orange to show that it is the selected region

B3.2.6 To Define Graphic Elements

The **graphic elements** are the objects on the screen (with the exception of the function key labels –you do not need to define these because they will get handled another way). In the example, these include the milepost marker, the track, the train, the signal, and the speed readout. You will click and drag over each graphic element the same way you did for each graphic region. However, if one graphic element contains another graphic element, it is better to define the smaller one first, then the larger one. This is because you cannot click inside of a graphic element to define another one inside it.

To demonstrate, select "Graphic Element" from the dialog area and position the cursor over one corner of the "MP 122" object. Click the left mouse button and drag the cursor to the other corner to form a box over the object. Then do the same for the signal, then the white train symbol at the bottom (with the "40" inside it). Then position the cursor over one of the upper corners of the track symbol (not the lower, to avoid the train symbol box), and click and drag to the other corner. Finally, do the same for the blue text at the upper right of the display. The display should now appear something like Figure B-7.

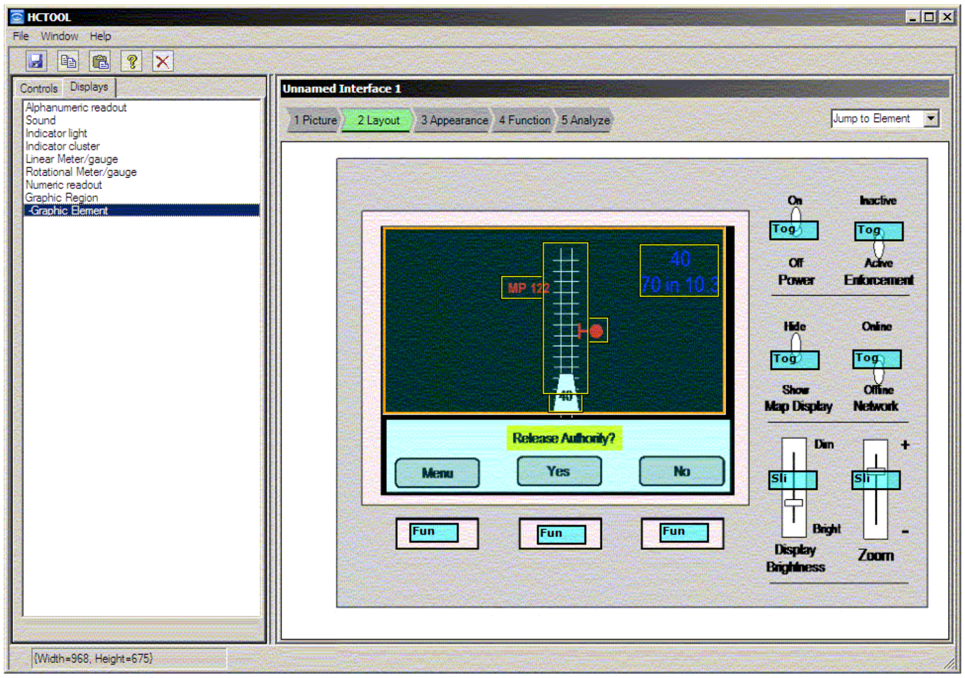


Figure B-7. The graphic elements have now been defined, outlined in yellow boxes over the black background of the main graphic region

B3.2.7 Save Your Work

To save your work so far, click on the Save icon (with the disk icon). If you have not saved this analysis yet, it will ask you to enter a name for the file. Name the file whatever you like, but do not include a file extension. The tool will automatically append the appropriate file extension (.OIF). If you select another file name or enter one with an extension other than .OIF, the system will advise you that the extension is wrong. If this happens, simply delete the file extension from the file name, or enter .oif at the end of the file name.

When finished defining displays layout, select the Appearance button over the work area.

B3.3 Define Appearance

In this step, you will tell the tool what the various controls and displays that you have defined look like and how they are oriented in relation to the user.

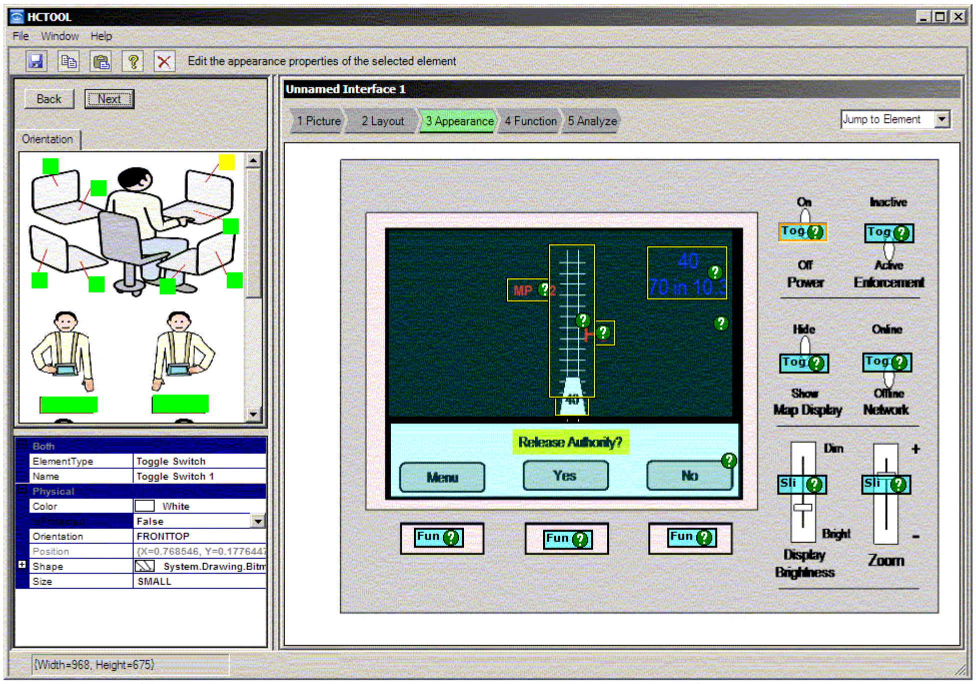
To start this step, if you have not already done so, select the Appearance button. The tool will bring up the first step that is appropriate for whatever kind of object is currently selected. So, for example, if the last object you worked on was a graphic element, the tool would now show a color palette so you can choose the appropriate color for that object. However, let us instead start with controls. To demonstrate, click on the toggle switch labeled “Power.” The dialog area now contains a selection of shapes for toggle switches, and the prompt line instructs you to select the shape that most closely matches the shape of the control you selected. Click on the shape you think is the best match, then select the Next button at the top of the dialog area.

Next, the dialog area will ask you to choose the color of the control. Click in the box with the closest color and select Next. If you have difficulty distinguishing the colors, you can select a color that looks close and read the color name in the **properties box** below the dialog area. The properties box lists all the attributes of the control as you select them. If you want, you can edit these properties directly within the properties box. For now, though, finish selecting the appropriate color and select Next. Answer the question about size and select Next.

The tool bases its reasoning about the confusability of controls on their similarity, as defined by your responses to this dialog. Therefore, it does not matter whether your responses are perfectly accurate, but rather that they are consistent. For example, if you are unsure whether a control should be marked as "small" or "medium," pick one and be consistent from then on; that is, use the same size rating for other controls of similar size. What is important to the tool is not whether a control is necessarily small, medium, or large on its own, but rather whether it is smaller or larger than, or the same size as, other controls that it might be confused with. This is most useful when there are multiple sizes of the same type of control in the interface.

Next, the tool will ask you to tell it how the control is mounted. In this particular case, assume the interface is for a control panel used by a locomotive engineer and that the engineer sits or stands in front of the panel. Assume, also, that the panel is mounted vertically in front of the engineer.

The dialog for indicating orientation to the tool uses diagrams of panel faces in relation to the user, as shown in Figure B-8. A green box is used to indicate the panel face that the box points to. Click in the green box associated with the vertical panel in front of the operator, and select Next.



Edit the appearance properties of the selected element

Both	
ElementType	Toggle Switch
Name	Toggle Switch 1
Physical	
Color	□ White
IsProtected	False
Orientation	FRONTTOP
Position	{X=0.768546, Y=0.177644
Shape	System.Drawing.Bitm
Size	SMALL

{Width=968, Height=675}

Figure B-8. To tell the tool that the panel is located in front of the operator and oriented vertically, select the green box corresponding to that location (now shown in its selected yellow state)

Next, the tool will ask if the control is "protected" or "unprotected." A control may be protected (from inadvertent activation) by a guard, a barrier, or some other means. In the case of the Power switch in this example, the control is unprotected. Choose the appropriate response and select Next.

B3.3.1 To Change a Control's Appearance Properties

At this point, the tool will indicate that the physical specification has been defined for the selected control. (A check mark is on the tag corresponding to the control.) You can see these properties listed in a properties box below the dialog area. If you change your mind about any of the responses you have made, you can edit them directly in this properties box. Alternatively, you can click on the back arrow at the top of the dialog area to go back to the response you want to change, change the response, and then click on Next again until you return to the end of the dialog. Finally, you can select another control, then reselect the current control. If the properties have already been defined for this control (as we just did), the dialog box will present these properties in a series of tabs, so you can select the tab corresponding to the change you want to make. For example, if you want to change the size you assigned to the control, you can select the "size" tab, make the change, and then either select another tab to make more changes or another control. Either way, your change is recorded as soon as you make it.

B3.3.2 To Copy One Control's Physical Appearance Properties to Another Control

Notice that the Power toggle switch that you have just defined looks exactly the same as the other three toggle switches on the interface. Instead of going through all of those steps for each control, you can copy the appearance properties from the control you have just defined to the other controls that look the same.

To do this:

1. Select the Power switch (its properties show up in the properties box below the dialog area).
2. Select the "copy" icon (like in Microsoft Word™, this looks like two sheets of paper) in the buttons under the menu bar.
3. Select a control you wish to copy the properties to (such as the "Enforcement" toggle switch—note that its properties box is empty).
4. Select the "paste" icon (as in Microsoft Word™, this looks like a clipboard with a sheet of paper in front of it), next to the copy icon.
5. Note that the properties box for this control is now populated and that there is a checkmark on the control indicating that its properties have been defined.
6. Repeat steps 3 through 5 for the other controls that look the same. In this example, these would be the "Map Display" and "Network" switches.

B3.3.3 To Define the Appearance Properties for the Remaining Controls

Select the next control whose appearance properties you wish to define. Again, note that its properties box is empty. As before, select the shape, size, color, and protection that you think is appropriate for this control.

B3.3.4 To Define the Appearance Properties for Graphical Displays

When finished defining the appearance properties of all the controls, simply select a display, graphic region, or graphic element to assign appearance properties to. Because the displays in this example consist of graphic regions and graphic elements, that is what we will focus on. For conventional displays, such as gauges, indicator lights, and so forth, the process is the same as for controls.

For the purposes of this example, click on the box outlining the black graphic region (the black background) behind the track map. The color palette should come up in the dialog area. Click on the black square in the color palette to tell the tool that the graphic region is black in color, and select Next.

Next, the tool will ask you how many viewable screens the graphic region has. This is where the function key logic starts to apply. If a graphic region contains function key labels, and there are 20 screens with different labels (so a given function key may have 20 different functions across all of the screens), then you would tell the tool that there are 20 screens. In this case, the black graphic region with the map does not contain any function key labels, so accept the default value of 1 for this step and select Next. In the next step, the tool will ask you to name the screen. You can give it a name, or accept the default name of ”screen1,” and select Next. This completes the appearance specification for this graphic region.

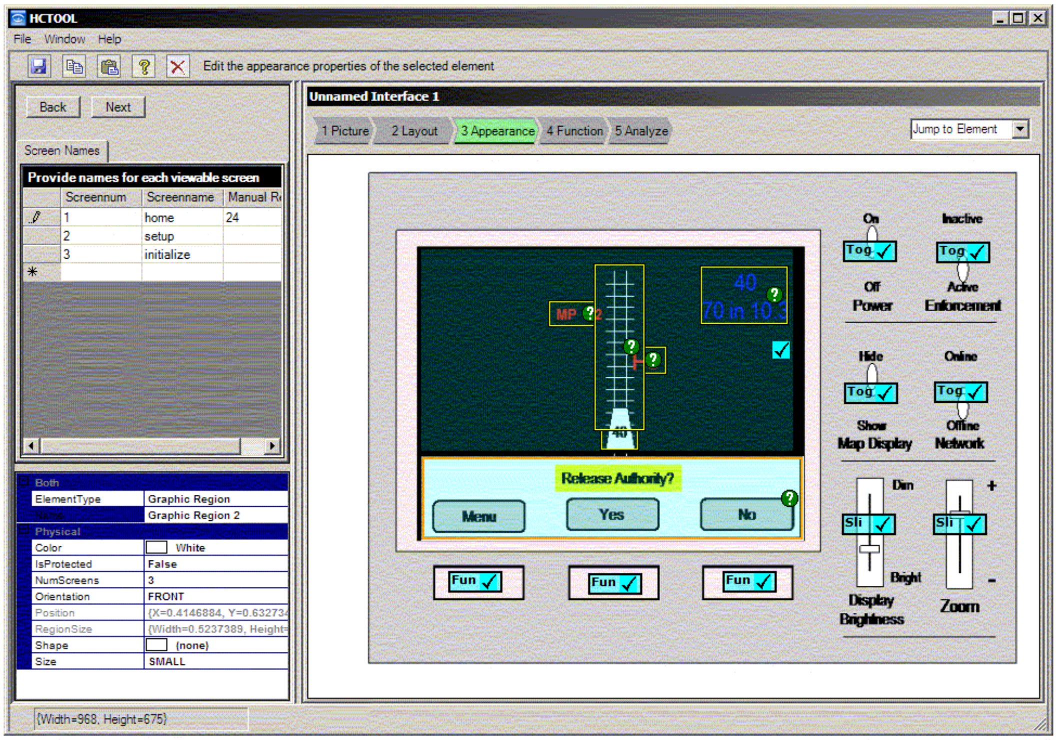
B3.3.5 To Define Multiple Screens for Function Key Labels

Now, let us do the same for the white graphic region below, containing the function key labels. In this case, however, assume that there are three different viewable screens. On the first screen, as shown in the example, function key 1 is labeled MENU, 2 is YES, and 3 is NO. Suppose that the manufacturer of this system gave you a Human Computer Interface specification that listed all the screens and described their operation, and that this specification listed the functions for these function keys in Table B-1.

Table B-1. Function key assignments for the three screens in our example

	Function key 1	Function key 2	Function key 3
Screen 1 (home)	MENU	YES	NO
Screen 2 (setup)	BACK	CANCEL	DONE
Screen 3 (initialize)	BACK	ENTER	CANCEL

In this case, then, when you get to the step where the tool asks you how many viewable screens the system has, enter or select 3. In the next step, you will see a table, like an Excel™ spreadsheet, where you can enter the names for the screens (if you wish). In the last column, there is also a place for you to enter in the page number that the screen appears on in the specification (see Figure B-9). So, for example, if screen 1 is the "home" screen and it is documented on page 24 of the HCI spec, you can enter "home" as the name for the first screen and in the next column enter 24 for the page number. This should help you keep track of all the different screens and their contents when working with a complex system that may have 50 or 100 different screens and 8 or 10 function keys.



Provide names for each viewable screen

	Screennum	Screenname	Manual R
✎	1	home	24
	2	setup	
	3	initialize	
*			

Both	
ElementType	Graphic Region
Name	Graphic Region 2
Physical	
Color	□ White
IsProtected	False
NumScreens	3
Orientation	FRONT
Position	{X=0.4146884, Y=0.63273
RegionSize	{Width=0.5237389, Height=
Shape	□ (none)
Size	SMALL

{Width=968, Height=675}

Figure B-9. The table in the dialog area can be used to specify the various screen names and where they are described in the manufacturer's documentation for a graphic region containing function key labels.

B3.3.6 To Define Appearance Properties of Graphic Elements

Next, we will finish defining the appearance properties of the interface by doing so for the graphic elements. To start this process, click on one of them to highlight it. For the purposes of this example, select the milepost element (the red letters to the left of the track reading "MP 122").

In the dialog region, you will be asked to enter a name for this object. Click in the text box and type "milepost," then select Next. Then you will be asked to select the appropriate object type. There are various types of graphic objects:

- A "Button" in this context is an on-screen button that you would click on in order to activate a function.
- A "Diagram/Graphic" is a potentially complex object intended to represent a real-world counterpart. The track map in this example would be such an object because its shape would correspond to that of the actual track.

- A “Dialog Box” is a text box with a place for the user to respond.
- “Digital” means a digital indicator or display.
- An “Icon” is a simplified representation of a real-world object. The signal object in this example would be an icon because its shape does not correspond to a real signal aspect.
- A “Menu” can be a pull-down menu or a selectable list of options.
- “Text” refers to an indicator that is only alphanumeric.

For the example, for the “milepost” object, select “Text” and then Next.

Finally, you will see a list of possible colors for the object. Where a control is presumed to have a single dominant color, graphic display elements may have several colors, and they may change from screen to screen. For example, the signal icon on this display might be red, yellow, or green depending on the underlying aspect (although the icon shape would not be capable of displaying the entire aspect, the color might provide a useful summary of the signal status). Because a graphic object might have several possible colors, the list of colors shown here is accompanied by check boxes so you can check multiple colors as applicable to a given object. For the example, assume the milepost labels are always red. Click in the checkbox for “red” and select Next.

At this point, the tool will tell you that the physical properties for this graphic element are complete. Select one of the other graphic elements in the example, and repeat the process of defining its physical form. Some hints for this example:

- The track object should be a “Diagram/Graphic,” and it is always white.
- The train symbol should be an icon, and it is always white (you can ignore the “40” text on the train symbol—it is not necessary to define the text separately).
- The signal icon can be red, yellow, or green.
- The speed and future speed indication in the upper right corner is “Text” and is always blue.

When finished defining the physical forms for all of the display elements, select “4 Function” to define the functions for all of the controls and displays.

B3.4 Define Functions

The purpose of the functions dialog is to tell the tool what function each control activates and each display represents, and how the possible states of those functions are mapped to the positions of the controls and displays. For example, consider a ceiling light controlled by a wall switch. The name of the function might be “light,” and it would have two possible states: on and off. The switch would have two positions: up and down. If the “on” state were mapped to the “down” switch position (that is, the light comes on when the switch is pushed down and turns off when the switch is pushed up), people from the United States may find the switch confusing because it operates in the opposite way than they expect (while people from the United Kingdom might find it completely natural). This is the type of counterintuitive design that the tool is intended to identify for you. In order for it to do so, you have to tell it what the functions and their possible states are, and how these states are mapped to the controls' various positions. Fortunately, the tool will guide you through this process.

To start this process, if you have not already done so, select "4. Function," and select a control or display tag to define the functions for that control or display. For the example, click on the control tag for the Enforcement toggle switch. Then, in the dialog area, enter in the name of the function. Usually, you can get this directly from the control label itself. In this case, the label is "Enforcement," so simply enter enforcement into the text box and select Next.

B3.4.1 To Define Control Units

Next, the tool will ask you to enter in or choose function units. "Units" in this context means feet, miles, pounds, etc. This is important because if a control is controlling a function in terms of ounces but the display is showing the same function in terms of pounds, there would be a mismatch between the two. Later, when you define the displays functions, you will select "miles" as the units for the milepost indication on the display. But for now, we are dealing with the enforcement switch, and it has no units, so accept the default entry of "none" by simply selecting Next.

B3.4.2 To Define Control Function Logic

Next, the tool will ask you to select the **function logic** for the control. The "function logic" describes how the control activates its function. "**Surface**" means that the function activates immediately when the control is operated (the function isn't buried under layers of menus or behind other steps). "**Multi-step**" means that the function only activates after the user has performed several actions in sequence (such as pulling a control out to unlock it, then pushing it to activate it). "**Press and Hold**" means that the function only activates after the control has been engaged for a period of time. The power buttons of some cell phones work this way; the user must hold the button for a second or so to avoid accidentally turning the phone off by inadvertently pressing the button momentarily. "**Press and release**" means the function only activates when the user releases the button after pressing it. "**Modal**" means that whether the function activates or not depends on the mode of the control or of the system. For example, the power switch on a multi-function remote control may turn the TV on in one mode and the DVD player on in another. This is becoming increasingly important in some systems because as technology becomes more complex, controls are increasingly being used for multiple functions, and mode errors have become an important source of hazard in some systems.

(While we have tried to avoid using any human factors jargon in this tool, this is one case where such jargon is necessary. Fortunately, you can see our definitions of these options by hovering the cursor, without clicking, over each one. A small yellow message will appear explaining what the term means.)

For this example, however, all of the controls will be **surface** level controls (the function activates immediately when the control is operated). Select Surface (or accept the default if it is already selected), and select Next.

B3.4.3 To Define Potential Hazard of Inadvertent Operation

Next, the tool will ask if accidentally operating the control could cause a hazard. To answer this question, think about what the possible consequences of operating the control inadvertently might be. In this case, for the “Enforcement” switch, accidental operation could shut off enforcement. If this were to happen without the engineer noticing (perhaps because he or she was looking out the window then the switch was bumped), the engineer might continue to rely on enforcement that is no longer in effect. Therefore, this question should probably be answered “yes” for this particular control. Select “Yes,” then Next.

Next, the tool will ask if the control is used frequently during normal operation of the system. If a control is operated frequently (say several times an hour), it might not be a good candidate for being protected because protections (such as a barrier the engineer has to lift in order to operate the control) would be very inconvenient for such a control (and might itself pose a hazard if the control required immediate, safety critical access, such as a brake control). However, if the control might cause a hazard if operated inadvertently AND it is NOT operated frequently (say, once a shift), then it probably should be protected. In this case, the “Enforcement” switch is not likely to be used frequently, so answer “No” and select Next. (Later, in the Analysis step, the tool will determine that the combination of infrequent use and potential hazard would make this a good candidate to be protected from inadvertent operation.)

B3.4.4 To Define Type of Motion

Next, the tool will ask whether the motion of the control is discrete (stepped) or continuous (smooth). An example of discrete (with fixed positions) would be a light switch with hard on and off positions. An example of smooth would be a continuously variable dimmer control for a light, such as a slide control or a knob. The “Enforcement” toggle switch in the example only has two, hard positions, so select Stepped and Next.

B3.4.5 To Map Control Positions to Function States

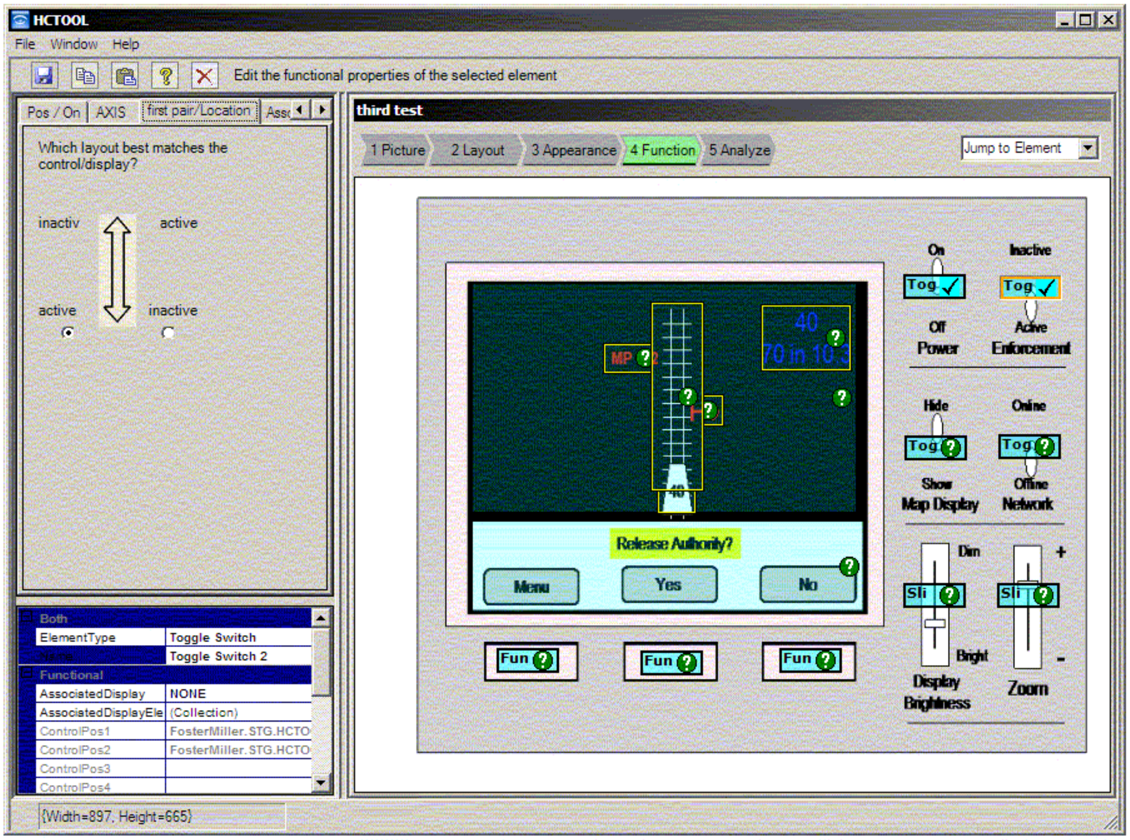
Next, the tool will ask you to enter in the control positions. As with the function name, you can probably take the names of the control positions directly from the panel. In this case, the positions labeled for this switch are “Inactive” and “Active,” so just enter those names into the two text boxes in the dialog area. One tip on entering information into these boxes: you can position the cursor in the top box, type “inactive,” then press the Tab key to send the cursor to the next box, and type “active.” As always, select Next when done.

Next, the tool will ask you which type of **function states** best describes these positions. Function states are general classes of function types. This is necessary in order for the tool to determine whether the layout of the control is likely to fit user expectations. The options are “Off/On,” “Less/More,” “Forward/Back,” etc. To return to our example of the ceiling light, a simple power switch with two positions would be “Off/On,” while a dimmer control would be “Less/More.” The accelerator of a car would be “Slower/Faster,” while the gear shift would be “Forward/Back.” (Even though the gear shift would also be “Less/More,” this would not capture the important role of the lever in directional control. If the driver had to push the gear shift lever forward to put the car in reverse, assigning it the “Forward/Back” attribute would enable the tool to determine that the control positions would likely be opposite what the user would expect.) At any rate, for the current PTC example, the “Enforcement” switch is most like “Off/On” because it determines whether enforcement is active or not, so select that option and then Next.

Next, the tool asks you to indicate which control position (based on the positions of the labels on the panel) corresponds to each of the two states selected in the prior step. This is necessary because the control labels may not correspond exactly to the function states that the tool knows about. For example, the tool “knows” about “Forward” and “Back,” but the “back” position of a gear shift lever might be labeled “reverse,” and the tool does not know about “reverse.” By having you tell the tool how the states it knows (“forward” and “back”) about are mapped to the control positions you defined (such as “front” and “reverse”), the tool can determine whether the control is laid out as a user would expect. So, for this step, the tool asks you to indicate which control position corresponds to the “off” state. Select Inactive (one of the labels you provided earlier) and then Next. Then, the tool will ask the same for the other function state (“On”). Select Active and Next.

Next, the tool asks you whether the switch moves vertically in relation to the user (up/down on a vertical surface or forward/back on a horizontal one) or horizontally (left/right). In this case, the switch is oriented up and down, so select the up/down arrow and Next.

Next, the tool asks you to indicate where each control position is in relation to the control itself (that is, which position is up and which is down). To do this, it presents you with the same vertical diagram of the control movement that you selected in the last step, with the two positions (active and inactive) mapped to each of the switch positions. See Figure B-10. Essentially, it is asking if the “inactive” position is on the top, or if the “active” position is on the top. Since the “inactive” position is on the top and “active” is on the bottom for this control, select the corresponding option and then Next.



Which layout best matches the control/display?

inactiv / active ● — active / inactive ○

Both	
ElementType	Toggle Switch
Name	Toggle Switch 2
Functional	
AssociatedDisplay	NONE
AssociatedDisplayEle	(Collection)
ControlPos1	FosterMiller.STG.HCTO
ControlPos2	FosterMiller.STG.HCTO
ControlPos3	
ControlPos4	

{Width=897, Height=665}

Figure B-10. In this step, select the option that best matches how the "Enforcement" switch is laid out. Notice that the tool places the control positions ("inactive" and "active") that you gave it in the two different possible arrangements. Select the one that matches the actual arrangement of the control. In this case, "inactive" is in the up position and "active" is in the down position

B3.4.6 To Map Control to Related Display(s)

Having finished with the mapping of function states to control positions, the tool now asks you to indicate whether there is a display associated with the control and, if so, what it is. The selections are "None," "Aural" (meaning that the control activates something that you hear, such as a beep; the horn and bell controls would also qualify for this), "Other" (which could be used for a display that is out of view), and "On Screen." If you can see the display or indicator that relates to the control that you are specifying, select "On-Screen," and then click on the display tag or region. In this example, suppose the "Enforcement" switch also turned the speed display on and off; select "On Screen," then click on the box surrounding the blue speed text in the upper right part of the display. The "graphic element" identifier for the speed box should appear in the text area beneath the "On Screen" selection. If so, select Next.

B3.4.7 To Remove an Associated Display

If you make an error and select the wrong display, you can remove it by selecting it in the text box (it should then have a blue highlight) and selecting the Remove button below the text box. Once it disappears, you can then select another display or one of the other options and move on.

B3.4.8 To Map Function Keys to the Function Key Labels

Recall that the white graphic region contains the function key labels, and we earlier told the tool that there were three different viewable screens with different labels associated with this graphic region. Because the functions of the function keys can change from screen to screen, we need to associate each key with the graphic region in which the label for that key appears. That will associate the key with all the possible functions that key can have on any screen.

To do this, select the first function key. The tool will present the "Associated Displays" dialog. Select "On Screen," then select the white graphic region. The graphic region identifier for that region should appear in the text box below the options. Select Next. (The next step will be the situation awareness parameters for future use; since they are not active now, simply select Next again to pass over it.) Then do the same for the other two function keys.

B3.4.9 Define Situation Awareness Parameters (Future)

Finally, the tool asks you to indicate what factors the engineer thinks about or considers when using the control. This step is provided to support the future analysis of automation functions. For now, disregard this step and select OK.

This completes the definition of functions for the "Enforcement" switch. At this point, select the next control to define the functions for and repeat the process. It might be helpful to select the "Map Display" or "Network" switch because they are similar to the "Enforcement" switch (they are also "Off/On" types of function), so if you select this control to do next, you can follow along through the previous steps with few changes. This should help you become accustomed to the dialogs and familiarize you with the process. Although the process has a lot of steps and may seem complex, it should start to go quickly once you become familiar with it.

B3.4.10 To Define Display Functions

To start this process, select a display element to define by clicking anywhere inside it. For the example, start with the milepost marker by clicking in the square surrounding the "MP 122" text on the display.

B3.4.11 To Define Display Units

First, the tool will ask you, as it did for the controls, to tell it whether the display element has units and, if so, what they are. Since mileposts are associated with miles, select Miles and then Next.

B3.4.12 To Define Display Function Logic

Next, the tool will ask you to tell it what the function logic of the display is. In this case, the "function logic" means under what logical conditions the display element appears. "**Permanent**" means that it is always on the display. "**Only this screen**" means that there are other screens available that it may not appear on. For example, the same PTC screen may be used for system schematics, and the milepost marker would not appear on these screens. "**Only some screens**" means that the display element does not appear on most, only on a few. "**Modal**" means that the display element only appears when the system or display is in certain modes. For example, if the PTC were in initialization mode, a progress indicator showing the time remaining until the initialization process was complete might appear only when the system was in that mode. "**Automatic pop-up**" means that the display element appears automatically when needed and disappears again when not. A dialog box or prompt is an example of this. "**User initiated pop-up**" means that the display element appears when the user calls for it. A clock that appears when the user presses a button or clicks on an icon to see the time might be an example of this. If the display element does not seem to fit well in any of these categories, select "**Other.**" For this example, however, the milepost marker appears only on the map display, so select "Only this screen," and then Next.

B3.4.13 To Map Function States to Display Element Positions

Next, the tool will ask if the specific positions or directions on the graphic are meaningful. This is necessary so the tool can determine if the display operates the way users would expect. For example, the track map is laid out to correspond to the direction of the track in relation to the train; the track ahead of the train on the map is also the track ahead of the train in reality. In this case then, the answer would be yes: the directions on the graphic are meaningful. For the milepost marker, the answer is a little more ambiguous; the orientation of the text itself is not meaningful, but its position in relation to the train is. To continue with the example then, answer Yes and Next.

Next, as with the control functions, the tool will ask which set of function states best describes these positions. The options here are the same as for the controls functions. In this case, since the position of the milepost marker is in front of the train, the best option would be "Forward/Back." Select that and then Next.

Next, as with the control functions, the tool will ask you to indicate how the display moves in relation to the user. In this case, the milepost symbol moves toward the user as the train moves along the map, so select the vertical arrow and Next. Then, again as with the control, the tool will ask if the "forward" position is on top, or the "back" position is on top. Select the option that corresponds to the design ("forward" on top and "back" on the bottom) and Next.

This completes the process of defining the functions for this particular display element. As before, select another display element and repeat the process. One hint: all of the display elements have the same position relationships to the train except for the speed indicator at the upper right corner of the display. Alternatively, you could say that only the track itself has meaningful directions and that the other graphic elements are only directionally meaningful because of their relation to the track. In this case, you could answer "no" to the question of whether the positions or directions of the graphics are meaningful for all of the graphic elements except the track. The result should be the same because all of the other graphics are tied to the track position.

B3.4.14 To Define the Graphic Region Functions

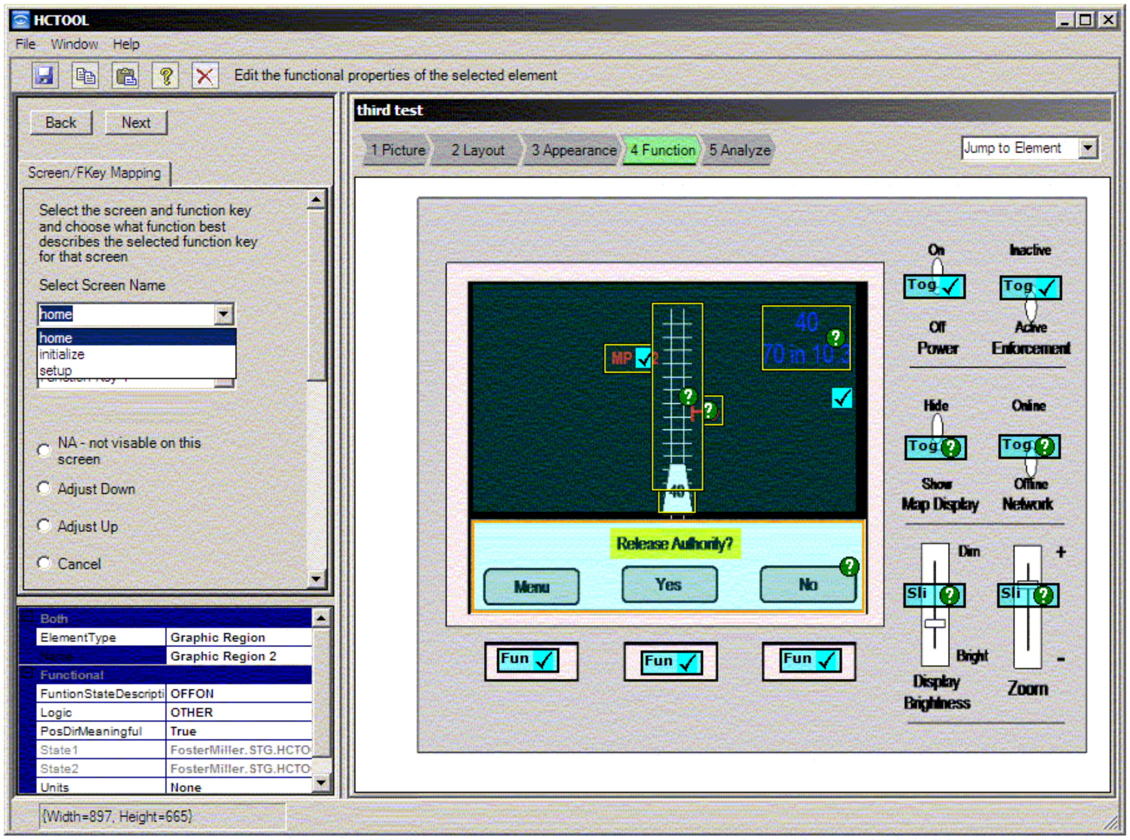
Recall that we defined two separate graphic regions (and recall that these are the backgrounds of the display): one large black region behind the track map and the small white region containing the function key labels. Because graphic regions can contain function key labels and otherwise only serve as backgrounds, defining their functions requires defining the function key operations.

First, let us select the large black graphic region. The tool will present a dialog to define the function key operations for this region but, since it does not contain any function key labels (and we did not associate the function keys to this region earlier), these options are grayed out. There is nothing to do here, so simply select Next. The tool will then tell you that the functional specification for this region is complete.

B3.4.15 To Define the Function Key Functions

Next, select the small white graphic region containing the function keys. Since we did associate the function keys to this region in an earlier step, the tool will bring up the function key assignment dialog. All of the assignments of each key to each screen take place within this screen.

Two drop-down list boxes say “Select Screen Name” and “Select Function Key.” As shown in Figure B-11, clicking on the down arrow on the right side of the “Select Screen Name” key displays the names of the three screens we identified earlier. Using the combination of these boxes, you can select the screen name and function key (1, 2, or 3) combination that you want to define. Let us look at an example.



Edit the functional properties of the selected element

Select the screen and function key and choose what function best describes the selected function key for that screen

Select Screen Name

home

home
initialize
setup

○ NA - not visable on this screen
○ Adjust Down
○ Adjust Up
○ Cancel

Both	
ElementType	Graphic Region
Name	Graphic Region 2
Functional	
FuntionStateDescripti	OFFON
Logic	OTHER
PosDirMeaningful	True
State1	FosterMiller.STG.HCTO
State2	FosterMiller.STG.HCTO
Units	None

{Width=897, Height=665}

Figure B-11. The drop down list of screen names used to associate function key functions with screens

Let us say that we want to define the three function key functions for the “Home” screen, the one depicted in the example. (These functions are also listed in Table 1.) With “Home” selected in the top box and “Function Key 1”selected in the second box, select the option below that seems to best describe the function for this key on this screen. The label for function key 1 on the “Home” screen says “MENU,” so scroll down the list of options and select “Open Menu” to indicate that this function key leads to a menu from the “Home” screen. Then scroll back up and select Function Key 2 in the lower box to assign that key’s function on this same screen.

In this case, the label for function key 2 is “YES,” so scroll down the list of options again and select “Yes.” Then scroll back up and select the third function key in the lower box. That key’s label is “NO,” so scroll down and select “No.”

You have now defined the functions for all three keys on the “Home” screen. To move on to the second screen (“Initialize”), scroll back up and select “initialize” in the Select Screen Name box, then select Function Key 1 in the Select Function Key box. On this screen, Function Key 1 is labeled BACK, Function Key 2 is labeled CANCEL, and Function Key 3 is labeled DONE. Repeat this process for this screen by selecting the appropriate function type for Function Key 1, then 2, and then 3. Finally, select the third screen (“Setup”), select Function Key 1, and repeat the process for the three function keys. On this screen, Function Key 1 is labeled BACK, Function Key 2 is ENTER, and Function Key 3 is CANCEL.

Once you have defined the functions for all three function keys for all three screens, select Next. The tool will tell you that the functional specification for the graphic region is complete.

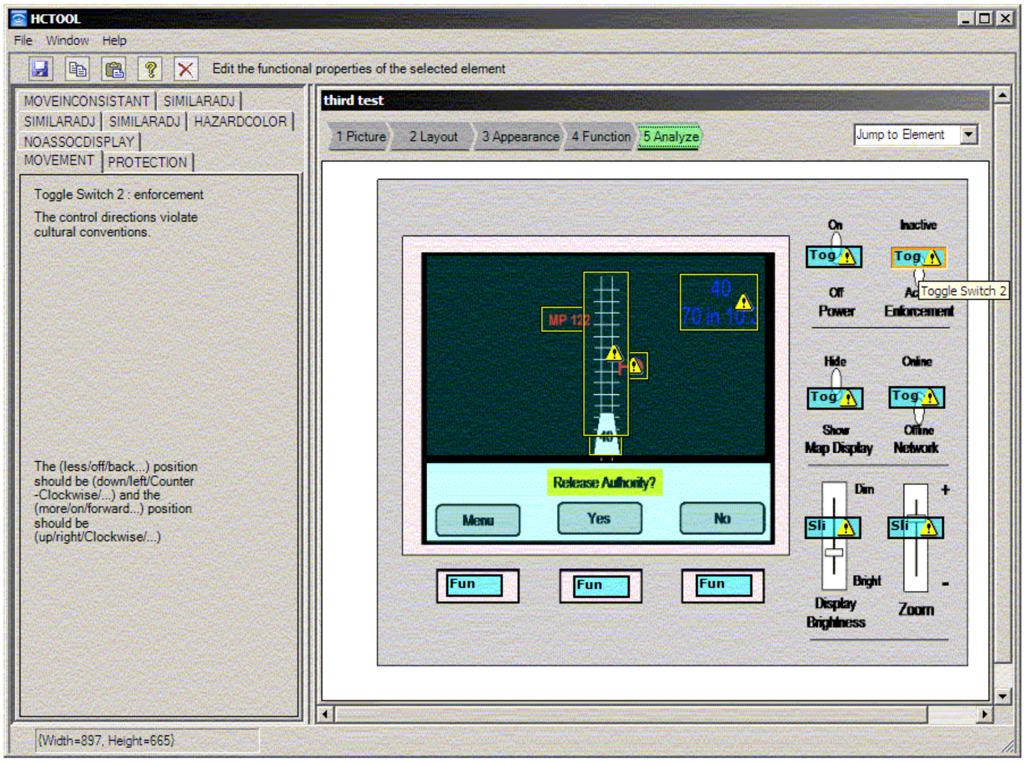
At this point, it may help to explain some of the rationale for this particular exercise. In a complex system with many function keys and many different screens (for example, a real-world PTC system may have 8 function keys and around 50 different screens), it may be difficult to determine whether functions have been assigned to function keys consistently across all of the screens. Inconsistent assignments may cause errors. For example, if the operator is accustomed to using a screen in which the far right function key is ENTER and is therefore accustomed to using that key to confirm an entry, he or she may make an error on a screen where that same key is assigned CANCEL or NO. The tool contains logic to identify cases where the same key may be assigned incompatible functions, such as ENTER/YES/DONE/CONFIRM on one screen and CANCEL/NO/ESCAPE on another.

The dialog for entering this information into the tool is designed to be as flexible as possible. By providing the two drop-down list boxes (one for the screen and one for the function keys), the tool enables you to define the functions for all of the keys screen-by-screen or key-by-key. Depending on how the manufacturer's specification is laid out, it may be more convenient to define the functions for all the keys on the first screen, then for all the keys for the second screen, then the third, etc. Alternatively, it may be easier to define all of the functions for Function Key 1 for all the screens, then go back and do the same for Function Key 2. The use of the two list boxes will support either method equally well.

B3.5 Analyze Interface

To analyze the interface design based on the information you have entered, select "5. Analyze." If the tool finds any problems, it will put up an alert indicator for each control and display element for which it finds an associated problem. It will also provide a written report listing all of the issues that it finds. It presents this report first in a window that pops up over the work area. If you click on the work area behind it, you can see the contents of the report on a control-by-control or display-by-display basis.

The interface for this is shown in Figure B-12. The tab headers indicate what kind of potential problem is detected (inconsistent movement, protection, etc.). Selecting each tab provides a more detailed description of the potential problem or issue. This interface provides an easy way of exploring the issues by control and display, but all of the identified issues are listed in the text report. You can switch back and forth between the two through the Window menu in the menu bar at the top of the screen, and you can print the written list of issues to document the analysis results.



Toggle Switch 2 : enforcement

The control directions violate cultural conventions.

The (less/off/back...) position should be (down/left/Counter -Clockwise/...) and the (more/on/forward...) position should be (up/right/Clockwise/...)

Figure B-12. The analysis report for the enforcement switch, contained in the tabs in the dialog area

B3.6 Compare Two Interfaces

You can also use the tool to compare two interfaces for problems that may arise from inconsistencies between them. These problems may lead to negative transfer, in which users make errors on one interface because of expectations formed by using the other. These may include:

- Cases where identical controls in identical locations operate different functions between the two interfaces.
- Cases where the same function is operated by controls in different locations between the two interfaces.
- Cases where the same control operates in different directions between the two interfaces.
- Cases where the same control uses different functional logic between the two interfaces.

You may compare two interfaces any time after you have defined both of the interfaces and saved each one as a separate .oif file. To start the comparison, you can either return to the main screen by closing any open files, or select File–Compare Two Interfaces. However, remember to save your current work first.

When you select either of these options, the tool will present two open-file dialogs. Select the first interface you want to compare in the first dialog and the second in the second dialog. The tool will then ask you what type of report you want. For the most complete report, select All. The tool will then provide a printable report of the various inconsistencies that it finds between the two interfaces being compared.

B4. Save

As always, do not forget to save the analysis before you close the tool.

B5. Contact

Foster-Miller, Inc., User Interaction Research and Design, Inc., and Mekinesis are committed to helping FRA users apply the tool. If you want or need help with any aspect of using it, or have any questions or feedback, please contact:

Dr. Victor Riley
User Interaction Research and Design, Inc.
360-945-0206
vic@uird.com

Abbreviations and Acronyms

EOT	end of train
FRA	Federal Railroad Administration
PTC	positive train control
RCL	remote control locomotive