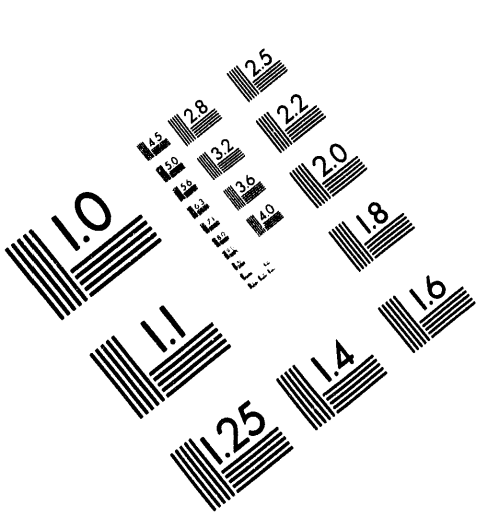
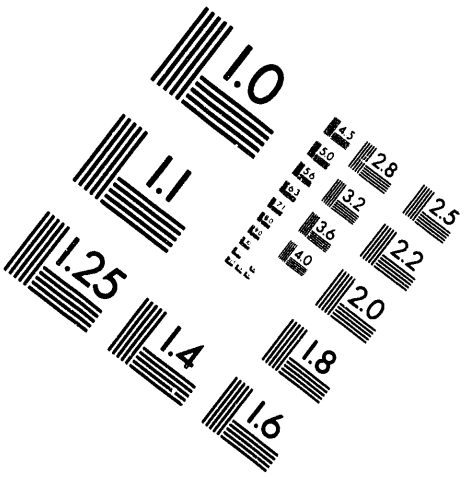
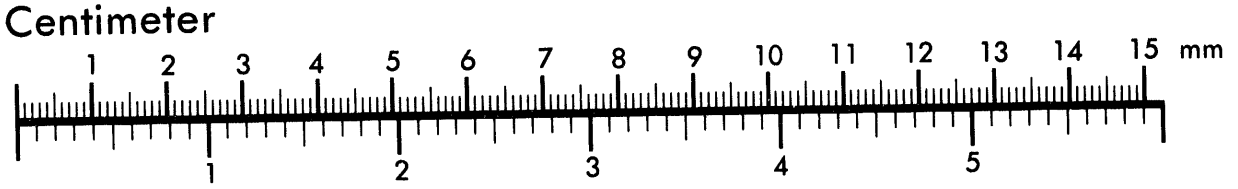


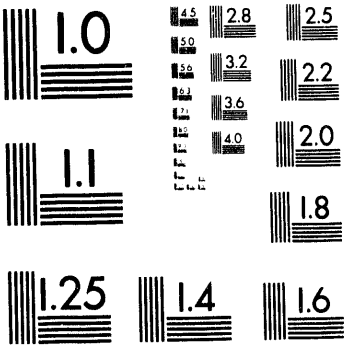
Association for Information and Image Management

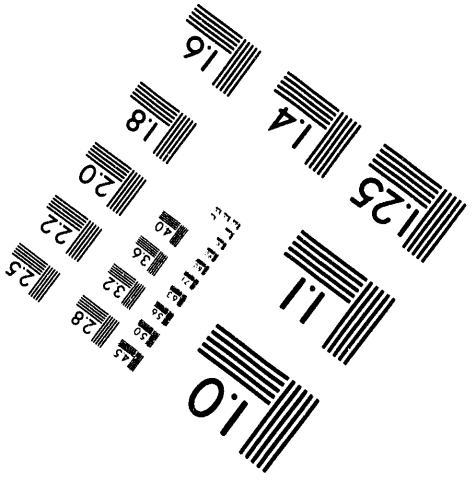
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

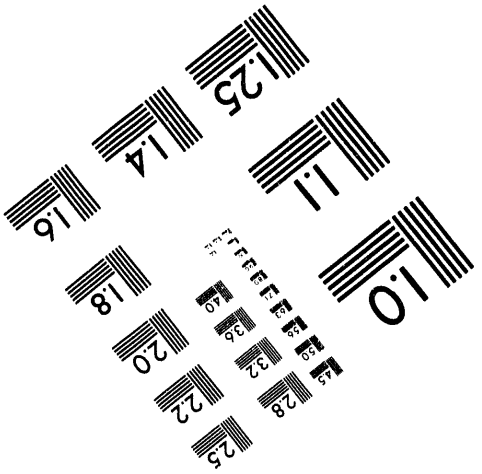
301/587-8202











1 of 1

Conf-940748--14

SAND94-1156C

Alarm Annunciation in a Graphical Environment

Douglas G. Adams
Sandia National Laboratories
Security Technology Department
Albuquerque, NM 87185-0780

ABSTRACT

Well-designed graphical user interfaces, such as Microsoft® Windows™ or UNIX™--based X-Windows®, provide a capability for enhanced display of security alarm information. Conversely, a poorly designed interface can quickly overwhelm an operator. This paper describes types of graphical information that can be displayed and offers guidance on how to best display that information. Limits are proposed for the complexity of the user interface, and guidelines are suggested for the display of maps and sensors.

INTRODUCTION

A physical security system includes these elements: Sensors that can detect an intruder; communications to transmit intrusion alarm information; and a reporting system to display alarm data to a human operator. An annunciator performs sensor reporting and control. When an intruder is detected, an alarm indication is sent to the annunciator. The annunciator should notify the operator with audible and visual indications. This can be as simple as a bell and light or as complex as a graphical user interface. Historically sensor alarms have been reported on light panels or on text-based computer displays.

Given the popularity of graphical user interfaces such as Microsoft® Windows™, or the UNIX™ based X-windows®, it was inevitable that alarm annunciators would be created using these interfaces. This paper describes the attributes of a well-designed graphical annunciator.

Foremost, a good graphical annunciator does not have an unlimited number of features. The current graphical user interfaces (GUIs) provide a wealth of features for displaying information. For annunciators, a good display limits the ways information is displayed and places constraints on which operations are allowed.

The following graphical features or concepts are described, and feature limits are proposed.

WINDOWS

The window, a rectangular region on the display screen, is the primary method of displaying information in today's GUI. A window can be any size up to and including the entire display screen, and multiple windows can be visible at any one time. A window can contain text, graphics, or controls.

Multiple windows of various sizes allow maximum flexibility when displaying information. A good alarm display, however, should limit the size and number of windows. No more than three windows should be visible at any one time. One of these should be the full size of the screen and should contain an overview of the system status. A smaller window containing subordinate information can be displayed as needed. Subordinate windows should never be larger than one-half of the screen. A third window may be displayed which contains menus or other operational controls.

Limits on the number and size of windows allow operators to quickly find important information. Windows do not have to be resized or moved to view information.

MENUS

A menu is a list of available commands. When a command is selected, a function is performed. Menus are usually displayed along the top of a window and can be nested. That is, selecting an item causes a subordinate menu to be displayed with additional items. Menus provide a clear and concise method of organizing system commands. It is important, however, that a menu structure not be too large or over nested. A good menu should have no more than nine items and should not be nested more than three levels. Users tend to get lost in deeply nested menus.

Limiting the number of items in a menu reduces the time required to find a particular item. Limiting the number of nested levels makes a menu structure easier to use. Complicated menu structures are intimidating to new users; experienced operators find them annoying.

BUTTONS

Although menus can display system commands in an easy-to-use structure, common commands should not be placed in menus. Common commands should be available as buttons. A button simulates the action of a push-button switch. An operator can "push" a button to initiate system action. Buttons are usually activated by depressing a mouse button or key on the keyboard. Only the most important commands should be placed on buttons, and only those commands that are valid in the current context should be available.

Buttons can be very flexible. Sensor or map icons can be made to act as buttons. Buttons can be grouped into button bars. A button bar organizes buttons into a single area on the screen for ease of access. Buttons can also be context sensitive, although this should be done in a consistent manner. Button flexibility must not be overdone. Limit visible buttons to a maximum of nine. Buttons should have good descriptive text labels which indicate their function.

MAPS

The primary advantage of GUIs is the capability to display maps or graphics of the secured area. Maps allow the user to quickly relate a security alarm to its location. Several map sources are possible:

Source	Scanned	Electronic
Topographic maps	✓	
Other government maps	✓	
Commercial maps	✓	
Aerial photographs	✓	
Site drawings	✓	✓
Sketches	✓	✓

All these sources fall into two groups: scanned copies of paper media, or electronically created graphics. Any source mentioned provides a useful graphic for alarm annunciation.

Of all the possible graphics sources, the best is a stylized sketch based on a topographic or other hard copy map. Maps usually have too much detail for effective use in security applications. Effective displays require small-scale maps of about 1:5000. An operator can create a sketch based on a larger-scale map and can eliminate unnecessary detail while providing the necessary scale.

Any maps provided for annunciation should be

interactive. The system should represent sensors on the map. The system should provide mechanisms to display and control sensors by performing operations on the graphic. The user should perform major operations by interaction with maps.

SENSORS

To support an interactive map, sensors should display on the graphic. The most effective representation for sensors or sensor groups is a small graphic. All sensor graphics (icons) should use the same graphic, be the same size, and use consistent colors. When feasible, display sensors together as a single icon. This can reduce screen clutter. No map should contain more than 50 sensor or group icons, although the total number of sensors displayed can vary based on the complexity of the map graphic.

A sensor icon should represent the status of the associated sensor. Display sensor states using unique colors and shapes. Grouped sensor icons should indicate the state of the worst-case sensor associated with the group. If any sensor in the group is in alarm, for example, the group icon should indicate an alarm. Consider making an alarming sensor the worst-case sensor state. Other sensor states are possible. Display these states appropriately.

Sensor State	Description
Secure	Sensor is clear, no intrusion detected
Alarming	Sensor is active, an alarm has occurred
Accessed	Sensor status reporting is disabled.
Unacknowledged	Sensor is active, with a new intrusi alarm.

TEXT

Graphically displaying information on a map does not eliminate the need for textual display of information. Provide dedicated areas of the display for descriptions of sensors. A good system will also provide some type of on-line or quick help. Limit text to vital information only -- place details in subordinate windows.

COLOR

Although color can be an effective aid to highlight important information, use it sparingly. A user should not be dependent on colors to operate a system. Remember that about 10 percent of the population has some form of color blindness.

Keep the number of colors to seven or fewer. Every additional color visible on the screen adds to the perceived complexity of the display. Menus, buttons, and backgrounds should be in shades of gray. Maps should be black and white or use low-saturation colors. Reserve the primary colors to indicate sensor status: red for alarms, yellow for access, and green for secure.

OPERATIONS

The overriding design philosophy for any security system must be "operator first." Operators must always be in command of the system. To achieve "operator first," follow these design rules:

1. Minimize the number of actions required to perform any command. An operator should only have to click the mouse once, or depress a single key for any major operation.

2. Only valid operations, based on context, should be available. For example, the operator should not be able to access a sensor if it is already accessed.

3. The system should use prompts to guide the operator through complex operations. A context-based command selection (see item 2 above) could be used to direct operators' actions without removing their control.

4. Annunciator systems should never override an operation in progress. If the user is assessing an alarm, the system must never abort the assessment to notify of a new alarm. The assessment should continue, and a non intrusive notification of the new event should occur. The operator can then choose to abort the current operation at his discretion. This principle applies in all situations.

5. Systems should not annoy the user. Avoid

using loud, continuous alarms or bright, flashing displays. The user is the most important factor to successful system operation. Keep the user happy.

6. Options should be available for performing any single command. What is simple for one user may be complex for another. Commands available as menu items, buttons, and keystrokes result in a friendlier system; the user could select his or her preferred method.

CONCLUSIONS

The mission of any annunciator system is to enhance a site's security. If a system fails its security mission, it is a failure as a system. Fancy graphics cannot salvage an ineffective system.

A simple-to-use system is much more likely to succeed than an unnecessarily complex one. Consider a simple user interface. Limit the total number of maps, sensors per map, buttons, menus, dialog boxes, and colors.

ACKNOWLEDGMENTS

This work was sponsored by the US Air Force, Electronic Systems Command, Electronic Security and Communications Center of Excellence (ESCCE); and the US Department of Energy under contract DE-AC04-94AL85000. Special thanks to the staff of ESCCE for their comments and assistance with this research.

References

1. Microsoft® Corp., "The Windows Interface: An Application Design Guide," Microsoft, 1992.

2. Anthony T. Bonina, "Guide to the selection and design of operator interface panels," *Instrumentation and Control System*, May 1994, Vol. 67, Number 5, pp. 23 - 25.

3. Douglas G. Adams, "DISS Annunciator Technical Performance Specification Inputs," *Dispersed Integrated Security System Evaluation, Development, and Integration of DISS Components*, January 1992, section 3.2.

DISCLAIMER

DATE
FILMED
8/31/94
END