SIF-Approved Document #: SIF-007-1996
(SIF Contribution #: SIF-NM-9602-018R2)

SIF APPROVED DOCUMENT

WORK GROUP: **Distributed Network Management Environment**

TITLE: **Design Principles for the Development of OAM Graphical User Interfaces**

DATE: **August 22, 1996**

EDITOR: **Name: Tom Winlow** **Miguel Planas**

Voice: 613-763-3642 **613-763-2736**

email:

ABSTRACT: The performance of telecommunications user interfaces can be enhanced through the application of specific design principles. The principles in this contribution build on those discussed in ANSI T1.232-1993/1995.

This document has received the approval of the SONET Interoperability Forum (SIF).

August 22, 1996

TABLE OF CONTENTS

1. SCOPE, PURPOSE, AND FIELD OF APPLICATION

Telecommunication networks are invariably populated by a variety of elements and systems supplied by numerous vendors, many of which use their own proprietary approach for the network management user interfaces. This lack of consistency not only creates a problem in accessing information in these very complex systems but also increases operational and training costs to the users.

While a generic user centered design approach is applicable to the design of the human to machine interface for telecommunication network management applications, its application will be limited due to the specialized and complex nature of the operational environment, and will not provide a complete and optimum solution. This document presents specific design principles that can significantly enhance the impact, usability and power of such user interfaces, as well as providing a degree of across vendor consistency in their operation.

The most recent work in industry is focused on graphical user interfaces, commonly called GUIs. The current standard reference on the G interface, ANSI T1.232-1995 deals briefly with the relatively new and exciting potential of these interfaces in the section on direct manipulation.

There is a set of design principles articulated in T1.232 which are intended to guide product designers as they create user interfaces. These design principles are summarized below.

1 - Consistency: Consistency of User Interface conventions between and among applications helps users transfer skills in new and unfamiliar situations.

2 - Empowering the user: Applications are tools and users should have control over their tools.

3 - Feedback: Feedback lets the user know what is happening in an application

4 - Efficiency: Efficient user interfaces enable users of all skill levels to perform their work with a minimum number of steps.

These principles are generic in that they apply to most if not all user interfaces. For example, an application designed for word processing benefits when the above principles guide the designer.

This contribution extends these principles with a complementary set which are intended to optimize benefits specifically for the management (operations, administration, maintenance, and provisioning - OAM&P) of telecommunications networks and network elements.

The principles build on those articulated in T1.232, providing design recommendations which will make the work of the craftsperson more efficient, more fulfilling and less error prone. Given that these design principles apply specifically to telecommunications systems, then it follows that the user interfaces for telecommunications products should be different from those designed with other environments in mind, for example, office applications. Section 3 elaborates on some of the reasons why this is true.

2. REFERENCES

The following standards contain provisions which have been considered during the process of development of this contribution. It is believed that the proposals in this recommendation do not contradict or otherwise invalidate any of the recommendations in these standards.

ANSI T1.203-1988, American National Standards for Telecommunications - Operations, Administration, Maintenance, and Provisioning (OAM&P), Human Machine Language

ANSI T1.232-1995, American National Standards for Telecommunications - Operations, Administration, Maintenance, and Provisioning (OAM&P), G Interface Specification for use with the Telecommunication Management Network (TMN) (Note: This refers to release 2 of this standard.)

ANSI T1.210-1992, American National Standards for Telecommunications - Operations, Administration, Maintenance, and Provisioning (OAM&P), Principles of Functions, Architecture, and Protocols for Interfaces Between Operations Systems and Network Elements.

ANSI T1.214-1990, American National Standards for Telecommunications - Operations, Administration, Maintenance, and Provisioning (OAM&P), Generic Network Model for Interfaces Between Operations Systems and Network Elements

ANSI T1.215-1990, American National Standards for Telecommunications - Operations, Administration, Maintenance, and Provisioning (OAM&P), Fault Management Messages for interfaces

ITU-T Recommendations Z.301- .341 (1988) Man Machine Language (MML)

ITU -T Recommendations Z.352 (03/93) Man Machine Language, Data Oriented Human-Machine Interface Specification Technique - Scope, Approach, and Reference Model.

ITU-T Recommendations M.3000 series- Telecommunications Management Network

Bellcore GR-2869-CORE, Generic Requirements for Operations Based on the TMN Architecture, October 1995

Bellcore, GR-1093-CORE, Generic State Requirements for Network Elements, Dec.1995

3. THE "G" INTERFACE MODEL

Systems architects have partitioned the functionality of complex telecommunications systems into specific sub-groups or layers. This process reduces the complexity of the design within each layer and allows the interfaces between these groups of functions or layers to be specified. Network management has been similarly partitioned in the Telecommunications Management Network (TMN) specifications. Figure 1 provides a high level view of this partitioning and identifies the interfaces between the layers, including the "G" interface.

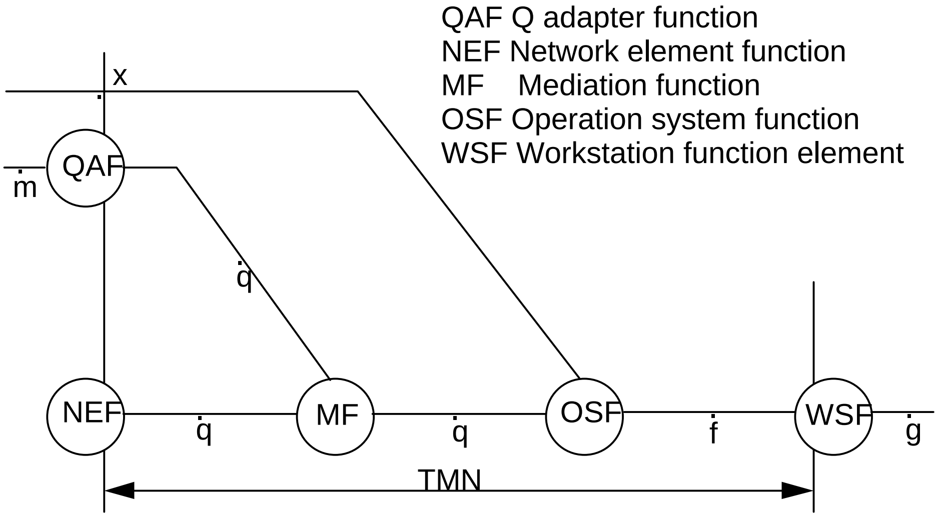

QAF Q adapter function
NEF Network element function
MF   Mediation function
OSF Operation system function
WSF Workstation function element
x
ṁ
QAF
q̇
NEF
q̇
MF
q̇
OSF
ḟ
WSF
ġ
TMN

Figure 1: Classes of reference points in the TMN

The interface between the actual user or craftsperson and the workstation is the "G" interface. While this interface is technically outside the boundaries of the TMN architecture, ANSI has created T1.232 to provide a degree of standardization at this vital interface. The "G" interface is critically important in modern telecommunications networks. It is the only view of the system that the craftsperson and the organization sees. Improper or inadequate design at this interface can result in significant problems in the network in terms of reliability and operational efficiency.

This contribution argues that the "G" interface can be appropriately divided into four conceptual components to provide an improved means of specifying the complex relationship between the system and the user. It also enables a better match between the users understanding of the information required for the job and the understanding required of the system designers who will be crafting the interface. Finally dividing the "G" interface into components makes the job of the system designers much easier for the same reasons that motivate partitioning the system, in particular, mapping the widgets to the user model (see 3.1).

One of the most challenging aspects of user interface design is making certain that the user model is understood and reflected in the design. The user model should achieve the following:

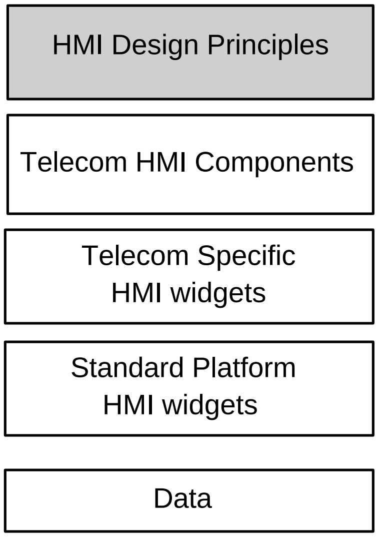
- bring the disciplines of psychology, user needs assessment, and UI design to bear on the presentation of the particular product functionality through the interface
- promote the distillation of design principles to be used as guides and metrics
- match the internal conceptual user model to the external world

A useful model should promote high quality, consistent UI design across a wide range of product environments.

3.1 Modular Approach to UI Design

There are many advantages to partitioning the G interface into a number of modules. Most importantly, the recommended partition will allow de-coupling the "look and feel" or UI operating system elements from those aspects which are application specific. For example, while the rules for the shape and visual characteristics of buttons or menus are contained in the particular system or platform, the assemblage of these and other elements into applications is a higher level design domain. Also, a given set of user design principles could be used for a number of different applications, each instantiation resulting in a different interface design for the actual screens and transitions.

There are four distinct modules within the proposed model, each which has a well formed set of requirements.



The HMI design Principles constrain how designers use the HMI widgets to construct the Telecommunications HMI components

The data module contains information about the specific systems, network elements and components for which the craftsperson has responsibility. For example, the name or location of a network element, the directory number of a subscriber, the number of DS-0 channels available across a specific link or the description of a permanent virtual circuit will have data specifying their attributes and class characteristics. The specific structure of this information is specified in the ITU M.3000 series - Maintenance: Telecommunications Management Model.

The HMI widgets component is comprised of two principal groups. First is a generic set of elements such as buttons, windows, and scroll bars. These are defined by the specific user interface style supported at the workstation. These HMI elements are relatively stable and common across the range of systems or products currently available in the market place. Second is a telecommunications specific set of widgets such as network element symbols, link or connection graphics, and shelf level graphics. While there are differences in the specific details of these systems, and while in the interests of UI design they should converge, this is not the subject of this question.

This contribution argues that these HMI widgets are not sufficient to ensure adequate user interfaces for telecommunications products or applications. Many of these building blocks or widgets and the ways of assembling them into functional frames were developed in and for the business office environment for which some of the most successful applications were developed. Examples are **Lotus 123** and **WordPerfect**. This environment strongly adhered to and developed the desktop model for the interfaces deployed on work stations and to date has been perhaps best implemented in the Macintosh workstation. However, the telecommunications management environment is significantly different in a number of critical areas. The following aspects differentiate a telecommunications applications.

1 - Telecommunications network management UIs are time critical

There is no concept of "alarm" in an office word processing application. In the management of a telecommunications network, and specifically the fault management application, faults may be critical and demand attention immediately. Thus the notion of asynchronous interrupts and the need to access specific information coupled with specific actions is vital for the user.

2 - Errors can be much greater in their potential effects

The capability of taking down significant sections of a large telecommunications network carrying large volumes of traffic demands a much more rigorous approach to the means of ensuring that errors do not go undetected. Some mechanisms used in generic interfaces are appropriate, but more are required.

3 - Users handle much larger volumes of data in real time

Network control centers can be likened to traffic control centers where from time to time large numbers of apparently isolated events may occur. Providing the user with assistance in prioritizing, filtering and managing these inputs is paramount in the design of effective network control applications.

4 - The objects with which the applications deal are much more complex

In a typical office application a user is confronted with files, documents, and possibly spread sheets. A typical network control application for example for configuration, deals with central office switches, Sonet transport equipment, plus a host of other equipment types each of which has different capabilities and requirements.

5 - Product developers are typically not experts in the operations domain

Everyone uses word processors, and spread sheets and drawing applications. Not many product developers have ever sat behind an operations desk when a number of mishaps are causing mayhem in the network .

Each of these has direct and important implications for the way the user interface is developed, and therefore requires careful consideration of those aspects that can be standardized. Section 4 presents a set of design principles which can guide the development community to produce user interfaces that meet the specific requirements of the telecommunications craftsperson. The principles do not constrain the developers to one specific design solution. They do provide general guidelines so that design solutions are better adapted to the needs of the telecommunication user and provide solutions which enhance the performance of the user.

The following principles are presented in the general context of resource management. The principles apply to all five functional domains in the TMN model, namely Faults, Configuration, Accounting, Performance and Security. They are also intended to apply to service management. This means, for example, that objects presented at the HMI could be either physical or logical .

Also important is the fact that these principles should be applied together in a balanced way.

4. DESIGN PRINCIPLES

Each principle discussed in this section will be presented using the following format:

Principle
This section formally states the principle

Rationale
This section presents the reasoning behind the principle. It attempts to present in a concise manner the distillation of knowledge in the following areas:

– psycho physical and psychological data
– previous design experience
– knowledge of the requirements of the user of systems which perform OAM for telecommunications equipment.

Design Implications
This section details specific implications for UI designs for telecommunications OAM applications.

Example
The screen example will present an **example** of how the principle couild be realized.

Note: These are examples only and are not intended in any way to represent requirements or design solutions.

Note: The appropriate way to view the images is to display them on a workstation. The paper document should be used only as a reference to the soft copy.

4.1 Group Functionality to Match the Users Conceptual Model

It is essential that (a) the right set of tools be defined to allow users to efficiently manage telecommunications networks, that (b) functionality is **grouped** correctly among the tools , and (c) that the functionality to perform specific tasks is available to the user without unnecessary intermediate steps.

Rationale

Until recently, OAM systems have been developed using either command-driven or menu-driven systems. These systems essentially use modifications of the verbal or written command model to organize and access functionality. The typical grammar for this type of interface is <command> <object>; that is, the user first states the action to be performed, and then defines the object(s) to be manipulated. This is an effective method of interaction in these types of systems, since it mirrors how commands are verbally expressed.

It is essential to note that these traditional methods of functional control do not map effectively to Graphical User Interfaces (GUI) styles, which operate best through object manipulation . This model hinges on the notion that the user manipulates graphic representations of objects using various **tools** that provide bundles of functionality.

In order to achieve the above, it is necessary to define a coherent model that matches the user's view for any given task. For instance, fault management technicians do not work in a clerical environment, and hence models based on clerical concepts such as "desktops" "documents" and "file folders" will be of limited use for them. Similarly, data-entry clerks who perform bulk provisioning tasks work in environments where "forms" and "audit" models are commonplace.

The primary reason for the success of current GUI-based products is not because they are more attractive, but rather because they allow the user to perform tasks in a more natural manner than traditional systems. GUIs are successful because they rely on the direct manipulation of objects and the associated commands, rather than relying on the recall of commands from long-term memory. GUI-based systems that do not take this fact into account are typically unsuccessful. Graphics, buttons, scroll bars etc. will not make up for a system that fails to partition and organize functionality to match the user's cognitive model of the task.

Design Implications

1. OAM tools and products should be defined to match **functional** units of specific tasks, rather than the organization of hardware or software components of the system.

2. GUI-based systems should operate using object-manipulation models which support the notion of object-oriented direct manipulation techniques. The typical grammar for these UI styles id of the form <object> <action>. That is, the user first specifies the object to be manipulated, and then defines the action to be performed on it.

3. The basic functional blocks upon which OAM tools and products are defined should match the functional breakdown defined in the basic TMN model. According to this model, OAM functions can be broken down into Fault Management, Configuration Management, Performance Management, Accounting Management and Security.

As an example, a set of **Fault Management** tools may be specified as follows:

Alarm Surveillance

Fault Verification

Trouble Reports

Testing

These tools can be launched via standard menus from an anchor tool that contains a representation of the target network, or as individual standalone tools. Tools to perform all other defined OAM functions may be defined, organized and accessed in similar manner.

Graphic Example

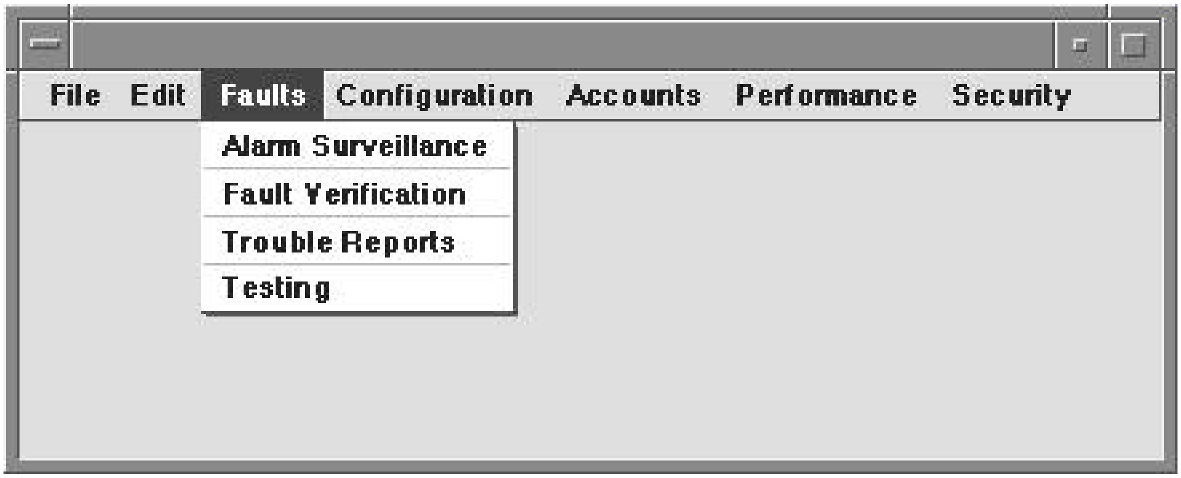


Figure 1 - Functional Partitioning

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.2 Design with the Minimum Frame of Reference necessary

Screen displays should contain **only** the information required by the user in order to (a) detect important signals, (b) perform the correct actions, and (c) maintain a stable frame of reference in any given situation. Superfluous decorations and data which do not add information relevant to the task will decrease the effectiveness of the displays.

Rationale

Graphic displays should communicate their current state to the user clearly, unambiguously, and directly. It is crucial that the user detects all relevant information, does not misplace or lose important information or functional blocks, or lose their way around the interface.

There are two basic contributors to human ability to detect signals in any environment: The strength and quality of the signal, and the noise or irrelevant data in which signals are always imbedded. For example it is much easier to detect a hand-clap in a quiet room than in a crowded one. This notion carries across all perceptual dimensions. Noise occurs in all the senses, and affects our abilities to detect, judge and act upon all situations in a detrimental fashion.

The effort required to work in noisy environments also has negative physical and emotional effects. For example, displays noisy with many and / or bright colors, and containing too much irrelevant information will cause visual fatigue, irritations of the eyes, muscle tension, and may lead to chronic headache and other stress-related complaints. On way to achieve the reduction of noise in the display is by grouping elements into objects with higher levels of abstraction as shown in figure 2.

Design Implications

1. Muted colors should be used for steady state displays in surveillance tasks. Background objects should use low levels of contrast and low levels of saturation while remaining visible to the user. Objects which contain relevant information should be displayed in brighter colors.

2. In general, there should be no more than three levels of information within a given display:

   – Background (the level containing the frame of reference or context; e.g. geographical maps or schematic diagrams)

   – Middle Ground (the level which contains the objects of interest to the user, e.g. nodes in the network)

   – Foreground (the level which contains the most important signals to the user; e.g. display elements used to differentiate alarmed objects in the network)

These levels should be differentiated in terms of brightness and / or saturation variations, with the highest brightness and / or saturation given to the foreground.

3. Displays should limit the number of colors that are presented simultaneously in the foreground. As a rule of thumb, 4 different hues could be displayed simultaneously without overloading the display.
4. Avoid details that do not add information germane to the functional task at hand.
5. Use simple graphical shapes. In the graphic design of screen objects, avoid highlights and excessive detail.

Graphic Example

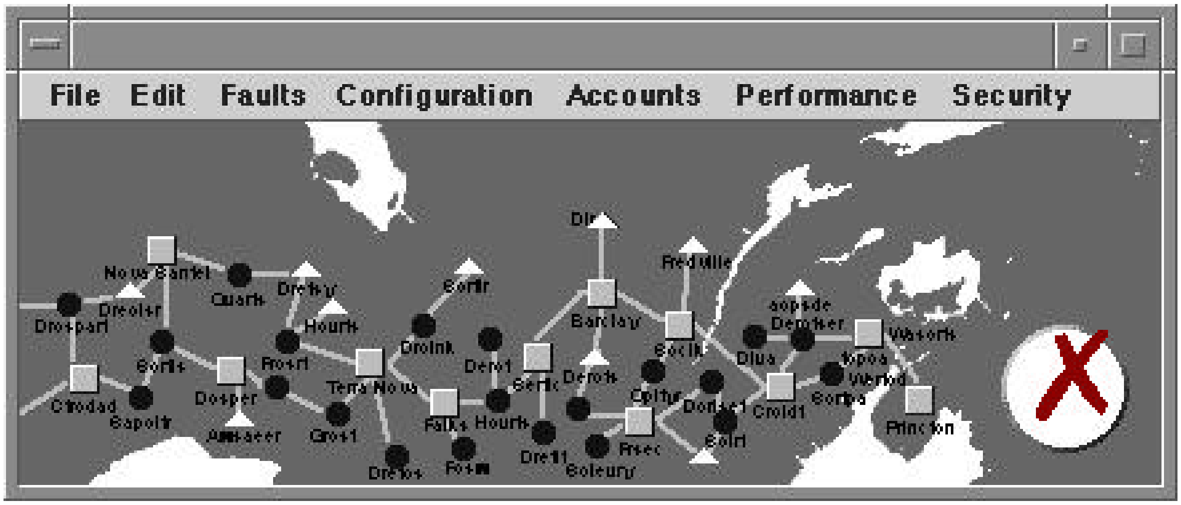


Figure 2a

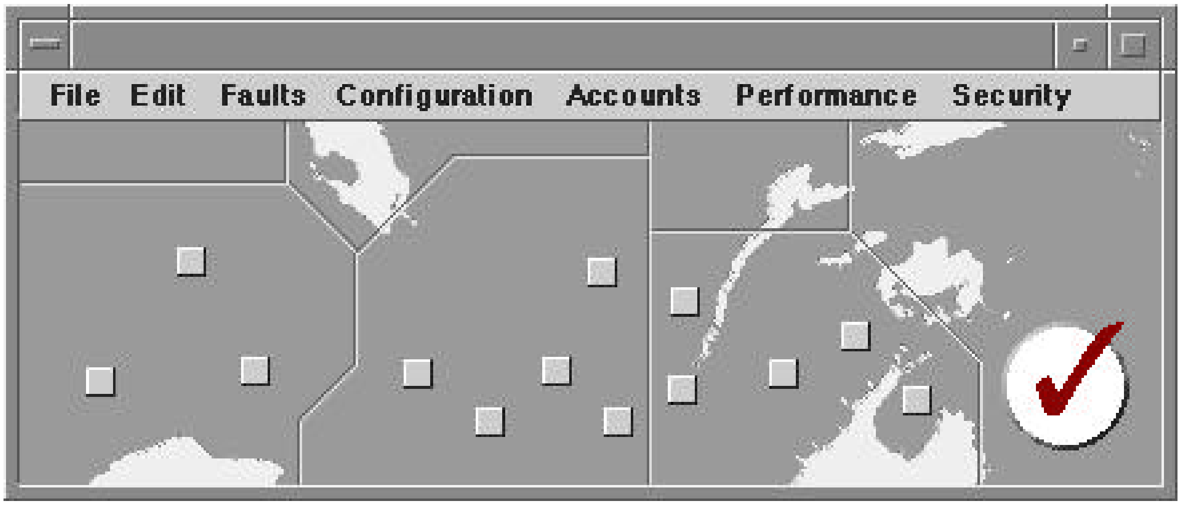


Figure 2b - Minimum Frame of Reference

Note: The above images are examples only and are not intended in any way to represent requirements or design solutions.

4.3 Provide Redundant Coding for Critical or New Information

Note: Coding in the above principle means the way or ways in which the information is presented to the user. Colour, for example, is one technique to code information.

When important or time-critical information regarding an object needs to be presented, vary a minimum of two visual cue dimensions (e.g. shape, size, color, position, etc.) simultaneously. This principle is especially important when presenting alarms or performance information which may lead to service degradation if left unattended.

Rationale

Objects in our everyday world exist in multidimensional space, and have many attributes which make them distinguishable from other objects and from the environment they inhabit. Our perceptual mechanisms are specifically geared to function in this multidimensional world. For instance, it is easier to tell one object from another if both their colors and shapes are different than if they were identical in all but one of these dimensions.

This multidimensional perceptual strategy allows us to overcome physical limitations such as color blindness, and environmental adversities such as poor lighting, atmospheric disturbances etc. in our everyday lives.

It should also be noted that complex telecommunications equipment is often in a combination of states at any given time. For instance, a given node may have any number of active alarms of various severities, some of them acknowledged, others not. Communicating this complexity of states to the user by varying only one dimension (e.g. color) will result in extremely complex and extensive coding schemes which will be extremely difficult to interpret.

Design Implications

1. Color must **never** be used by itself as a mechanism to communicate important or time critical changes of state. Color enhances the effectiveness and attractiveness of carefully-designed displays. However, note the following:
   - a significant proportion of the population is color-blind. Of this population, the largest proportion has difficulty in discriminating between red and green.
   - color displays are notoriously difficult to control and calibrate properly. Improper settings not only cause stress and physical discomfort, but can also mask information that is coded in color.
   - displays should effectively convey important information to the user even if the display mechanism is degraded.

   This notion has long been recognized, some times implicitly, in applications such as displays of traffic control signs, where a missed or misinterpreted signal can lead to disastrous consequences.

2. It is desirable that one of the dimensions included in the coding mechanism for display of object state be textual. Text information, used intelligently in conjunction with other graphic displays can enhance the effectiveness of the display by providing an explicit verbal label of the state for the user. This will help users communicate among themselves, and will facilitate access to support documentation.

3. When manipulating changes in color displays, it is desirable to vary both hue and brightness of the display in any given change of state. Manipulating only one of these dimensions significantly reduces the discriminability of the signal to the user.

Graphic Example

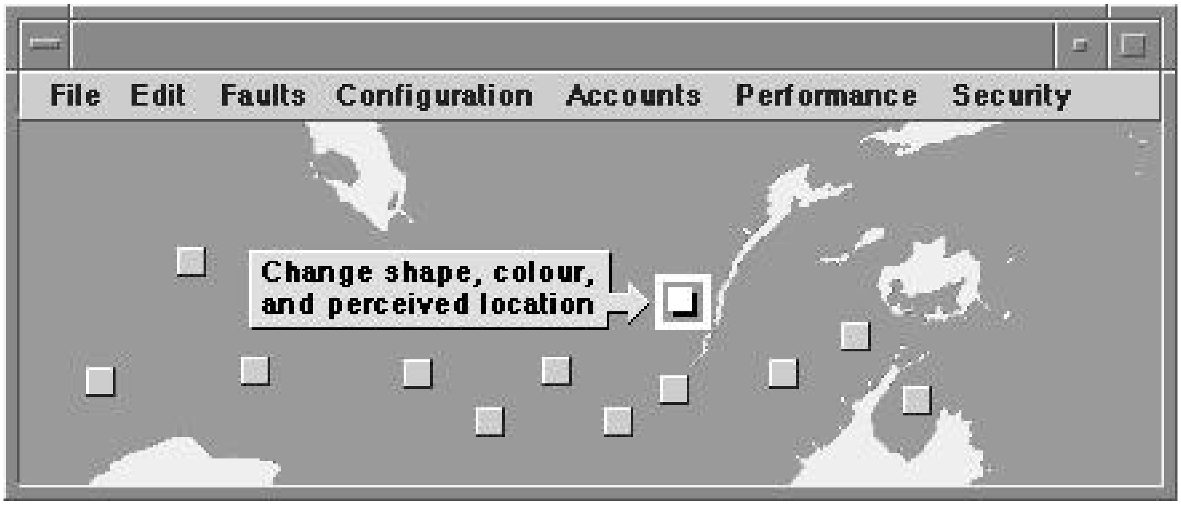


Figure 3 - Redundant Coding

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.4 Flatten the Hierarchy

Most people have problems with traversing hierarchical systems because they loose their frame of reference and can quickly become lost in the hierarchy. Requiring these extra steps adds both time and effort to the task, increases the cognitive load, and decreases the user's satisfaction with the product. Presenting the hierarchy also consumes precious screen real estate.

Rationale

Objects and functions in complex systems are typically hierarchically organized and implemented. Often this hierarchy is reflected in the user interface of the systems via hierarchical menus and modes. For example, in dealing with an alarm, users often are given an indication, graphical or otherwise, that an object is alarmed somewhere in the hierarchy. Typically, the user is then required to (a) traverse the hierarchy until the object is located, (b) determine the state of the object, and (c) act upon the situation. Also users are often required to open a number of separate windows in order to find required information for task resolution. Besides slowing down the analysis and understanding of the problem, users may also get lost when a large number of windows is open for each task.

This hierarchy is required for implementation purposes, but should **not** be forced on the user.

Design Implications

1. All of the primary information required to understand the context, scope and significance of the situation should be directly available to the user within the current display.

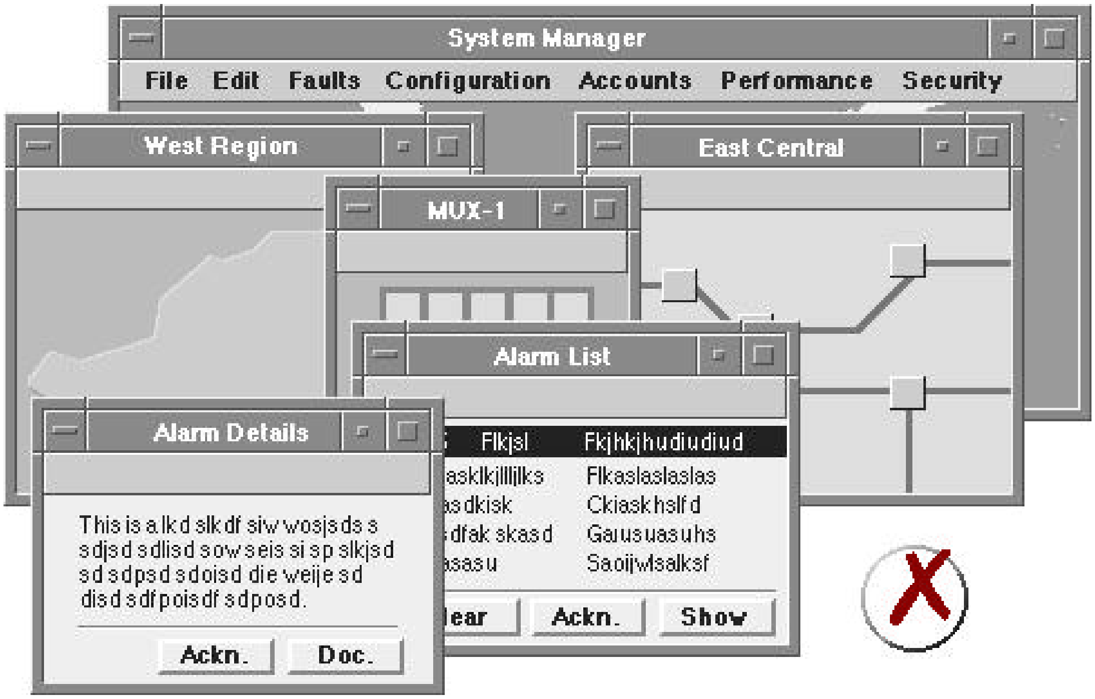
2. Supplementary information needed in a given situation should be made available to the user within one step of the current display.

3. The user should have direct access to all of the primary functionality required to resolve a given situation directly within the current display.

4. Supplementary functionality needed to resolve a given situation such as lists or tables should be made available to the user within one step of the current display.

Note: The following images are examples only and are not intended in any way to represent requirements or design solutions.

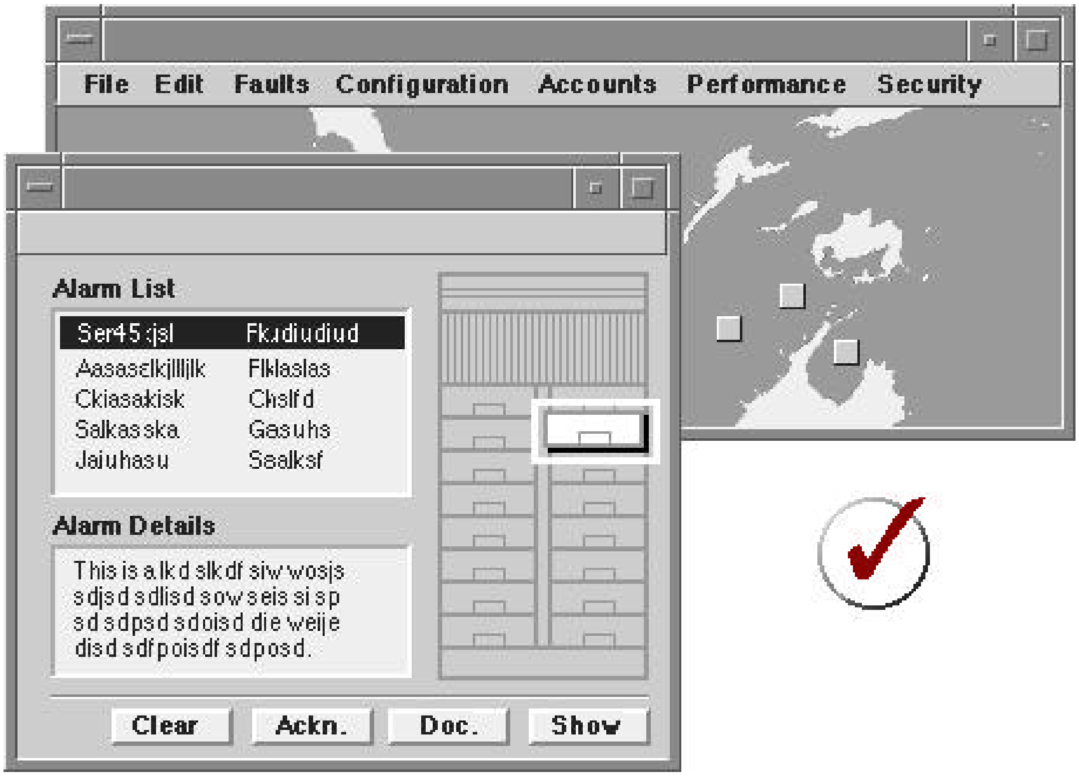
Graphic Example



Flkjsl Fkjhkjhudiudiud
asklkjllljlks Flkaslaslaslas
asdkisk Ckiaskhslfd
dfak skasd Gauusuasuhs
asasu Saoijwlsalksf

This is a lkd slkdf siw wosjsds s
sdjsd sdlisd sowseis si sp slkjsd
sd sdpsd sdoisd die weije sd
disd sdfpoisdf sdposd.

Figure 4a



Ser45:jsl Fkudiudiud
Aasasslkjllljlk Flklaslas
Ckiasakisk Chslfd
Salkasska Gasuhs
Jaiuhasu Sealksf

This is a lkd slkdf siw wosjs
sdjsd sdlisd sowseis si sp
sd sdpsd sdoisd die weije
disd sdfpoisdf sdposd.

Figure 4b - Flattening the Hierarchy

4.5 Telescoping in Place

Users require fast, efficient means to assess a situation and act upon the objects involved in a given task. Requiring extra steps adds both time and effort to the task, and decreases the user's satisfaction with the product.

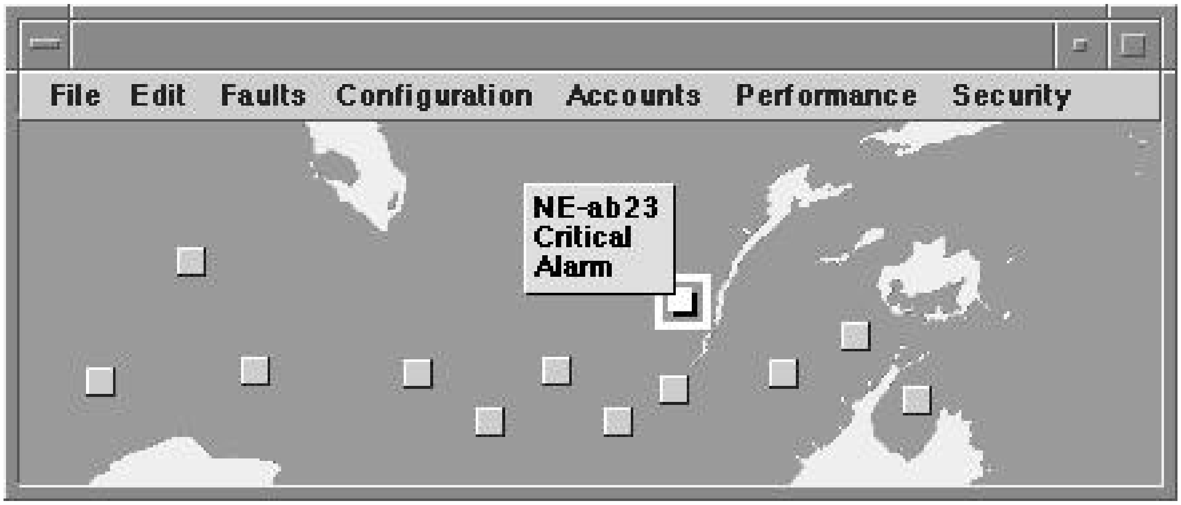
Rationale

Information regarding network events, whether these are urgent alarms or simply information displays often require the user to respond to an event presented on the screen. Often there is pertinent and limited information which could be presented with no user action required, and which would improve the overall picture the user has of the network. The system should provide key or relevant information when the context is known with no required user action.

Design Implications

1. Summary information available to the system regarding alarmed network elements should be displayed directly on presentation of the alarm to the user
2. Management systems should provide this summary information directly to the user with no actions required, and should present related information directly by accessing the alarm object.

Graphic Example



NE-ab23
Critical
Alarm

Figure 5 - Telescoping in Place

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.6 Support Parallelism Through Concurrent Views

Users must be allowed to generate multiple concurrent views of different aspects of the network. These views may be either logical or physical views. This ensures that users can maintain a suitable context within which tasks can be accomplished.

Rationale

Humans are parallel processors, especially when solving complex problems such as those often found in large telecommunications networks. For instance, when troubleshooting a fault, a user may need to view details of several related network elements, and at the same time, maintain an overall view of the network.

Multiple views minimize the load on the user's short-term memory, thus reducing commission of memory-related errors. In addition, they often help the user maintain a stable overall frame of reference as compared with traditional systems, which force the user to constantly traverse a maze of modal hierarchical levels.

Design Implications

1. Default parameters for windows locations and sizes should be designed such that newly opened windows should not completely cover the previous frame.
2. Users should be able to replace the current frame of reference with a new one.
3. A user should be able to have several different views of the network at the same time.
4. Specific context information should be displayed in all windows.

Graphic Example

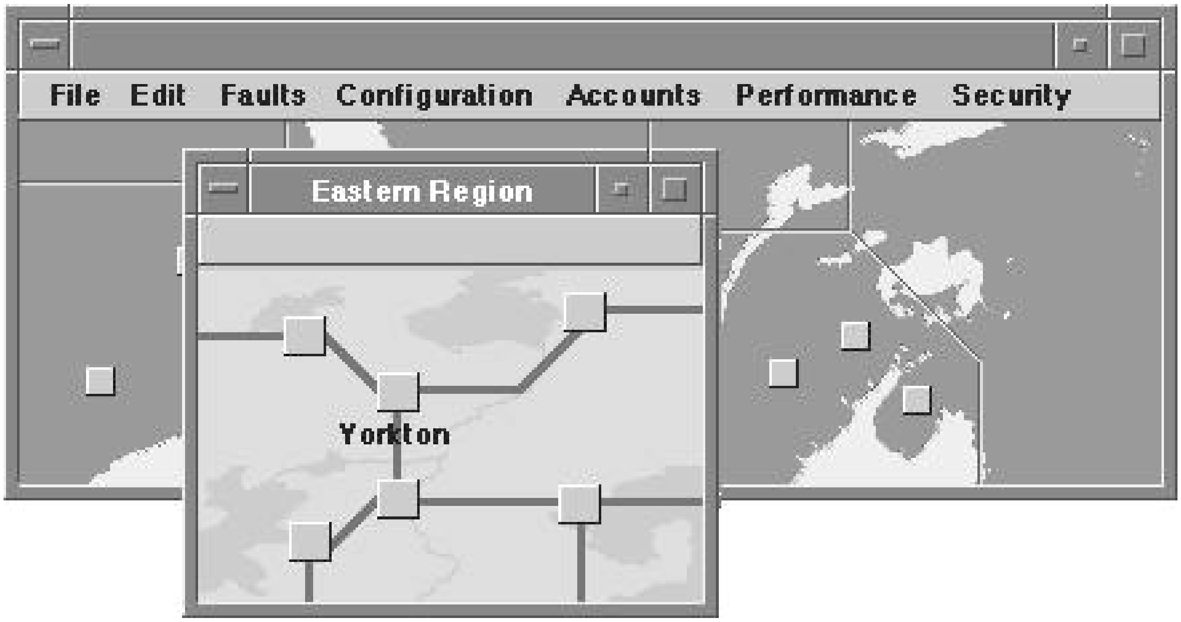


Figure 6 - Concurrent Views

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.7 Allow for Customization

OAM products must allow several types of customization of their user interfaces. Administrations and users require the ability to optimize the interface for specific or unique business requirements. At the same time, it should not be possible for users to so modify their displays that the efficiency or effectiveness of task performance is negatively affected. A simple example would be for a user to change the color of critical alarms displayed on the workstation.

Rationale

Regardless of how a given interface is constructed, there will be requirements to add or delete specific elements, choices, or options. These requirements result from differences in the operating environments in which the system is imbedded, differences in the vintage of equipment which is connected to the management platform, or differences in current operating practice and experience. Also, certain jurisdictions may allocate the tasks differently among their work forces. In addition, users may enhance their performance by customizing specific tools to accommodate task or individual requirements.

Design Implications

1. A system administrator or equivalent should be able to divide functionality among various user groups according to job type and jurisdiction.

2. Users should be able to customize their individual displays to reflect their level of expertise and preference of access to frequently performed tasks.

3. Users should **not** be able to alter the basic layout of their displays in ways that make them fundamentally different to other users within the organization that perform the same or similar tasks.

4. The following aspects are examples of user interface customization requirements:

    - Accommodation of native languages

    - Support for natural writing symbols and punctuation

    - Support native conventions for date, currency, weights, numbers and addresses

    - Provide for specific work habits and the environment

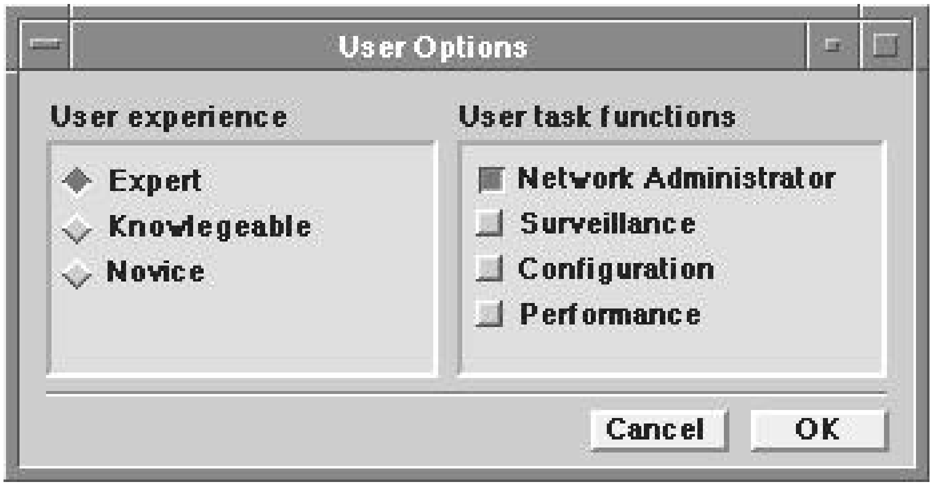
    - Communicate to users in natural and inoffensive ways

    - Be sensitive to the customers culture

    - Specific color meaning requirements

    - Ability to filter out non-essential or not-wanted information

Graphic Example



User Options

User experience

● Expert
○ Knowlegeable
○ Novice

User task functions

☑ Network Administrator
□ Surveillance
□ Configuration
□ Performance

Figure 7 - Allow Customization

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.8 Make Everything Concrete

Give all abstract events and constructs attributes that can be addressed, surveyed and modified, thus making them concrete.

Rationale

Network elements or events do not exist in isolation and it is the relationship of an object or event to other related objects which provide the necessary context for people to effectively deal with them.

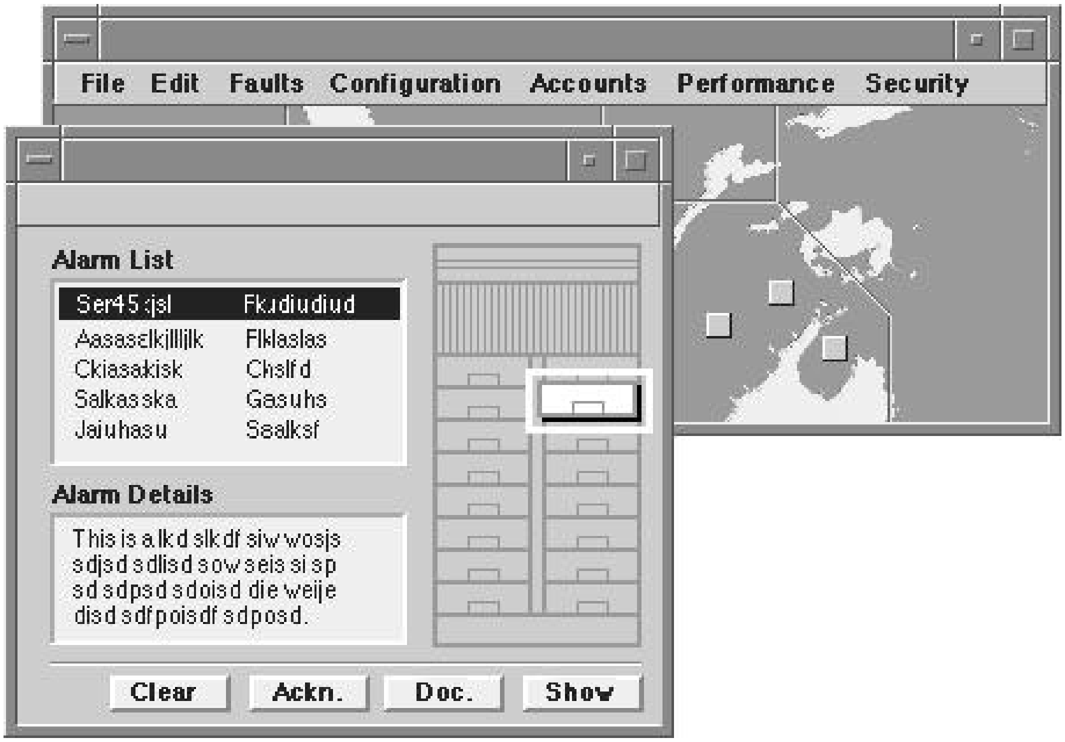
People understand the world around them by remembering relationships between objects. They even do this in the case of abstract concepts such as network events. In order to deal with these, people first create internal representations which make the concepts concrete. In this way, they can explore interrelationships and create meaning based on the information they perceive.

Design Implications

1. Every system object and every system event that is displayed should be treated as an object.

2. Abstract concepts such as logical groupings, alarms, or performance thresholds should also be treated as objects

Graphic Example



Ser45:jsl	Fkudiudiud
Aasaselkjllljlk	Flklaslas
Ckiasakisk	Chslfd
Salkasska	Gasuhs
Jaiuhasu	Saalksf

This is alkd slkdf siw wosjs sdjsd sdlisd sowseis si sp sd sdpsd sdoisd die weije disd sdfpoisdf sdposd.

Figure 8 - Object oriented perspectives

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.9 Establish a consistent and complete visual language

A visual language is akin to a spoken one in that it empowers users to comprehend abstract events and objects through a clear and unambiguous taxonomy and lexicon.

Rationale

The visual language presented by the user interface is critical to the effectiveness and efficiency of the users. Language systems consist of two components: vocabulary and grammar. Vocabulary is the collection of elementary symbols, for example, letters and words. Grammar is the collection of rules used to combine symbols.

People will respond to good visual design in specific and measurable ways. In particular the system and tasks will be easier to understand, easier to learn, and will be more engaging.

Design Implications

1. Contrast: If elements are the same, make them look the same. If elements are different, make them look different.

2. Repetition: Create a series of common visual threads to tie the interface together by repeating one or more elements throughout.

3. Alignment: Every element should have a visual connection to another element. Do not position elements arbitrarily

4. Proximity: Physically group related elements.

5. Apply the above - Contrast, Repetition, Alignment, Proximity, in a consistent way to the following visual attributes: position, size, shape, color, texture, composition, viewpoint, depth, and style.

Graphic Example

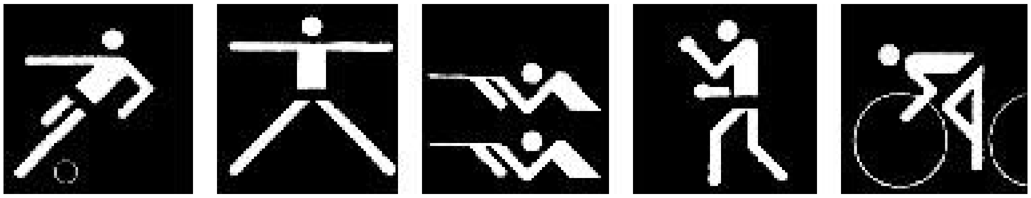


Figure 9 - Complete and consistent visual systems

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.

4.10 Use animation only to communicate transitions

When people see movement, they infer that something has changed, and further that there has been some event which has caused the change.

Rationale

In many situations our perception of causality is linked to our perception of movement. Motion usually results in a direct perception of causality. Animation when applied to transitions assures the user that the system is responding to a command and is not simply idle.

Design Implications

1. The use of animation on screen should be reserved for the representation of a transition and NOT for steady state information.
2. The presentation of information or its retrieval that will not be completed in less than a few seconds should be accompanied by a transition indicator.
3. Steady state conditions, as they will persist until an action is taken, are not candidates for animation.

Graphic Example

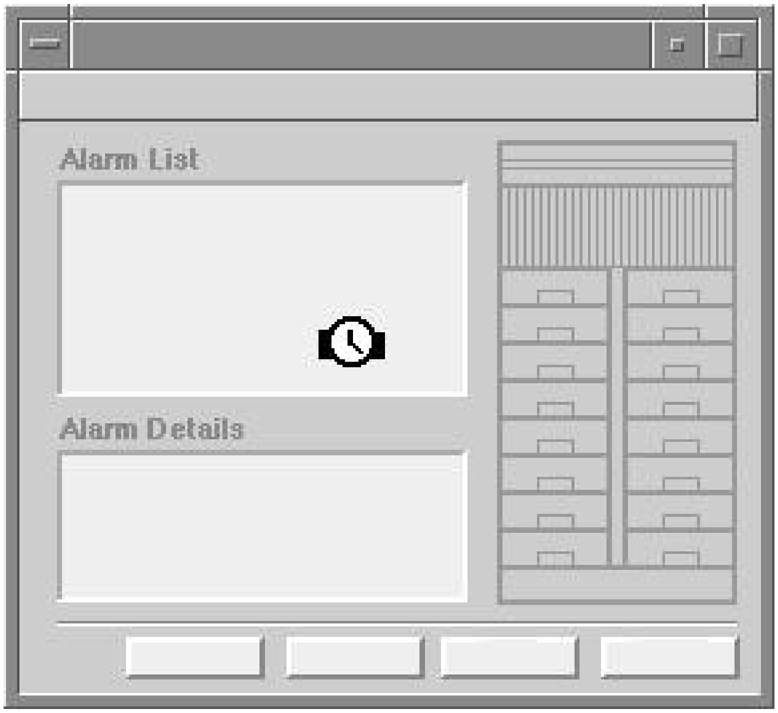


Figure 10 - Transitions use motion coding

Note: The above image is an example only and is not intended in any way to represent requirements or design solutions.