International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.372

(04/2005)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

Man-machine language – Human-machine interfaces for the management of telecommunications networks

Templates for telecommunications human-machine interfaces

ITU-T Recommendation Z.372

ITU-T Z-SERIES RECOMMENDATIONS

LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

FORMAL DESCRIPTION TECHNIQUES (FDT)	
Specification and Description Language (SDL)	Z.100–Z.109
Application of formal description techniques	Z.110–Z.119
Message Sequence Chart (MSC)	Z.120–Z.129
Extended Object Definition Language (eODL)	Z.130–Z.139
Testing and Test Control Notation (TTCN)	Z.140–Z.149
User Requirements Notation (URN)	Z.150–Z.159
PROGRAMMING LANGUAGES	
CHILL: The ITU-T high level language	Z.200–Z.209
MAN-MACHINE LANGUAGE	
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.349
Data-oriented human-machine interfaces	Z.350–Z.359
Human-machine interfaces for the management of telecommunications networks	**Z.360–Z.379**
QUALITY	
Quality of telecommunication software	Z.400–Z.409
Quality aspects of protocol-related Recommendations	Z.450–Z.459
METHODS	
Methods for validation and testing	Z.500–Z.519
MIDDLEWARE	
Distributed processing environment	Z.600–Z.609

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Z.372

Templates for telecommunications human-machine interfaces

Summary

This Recommendation provides a specification for the design of the Human-Machine Interface (HMI) for object browsers. An object browser allows a user to select an object from a group of objects and perform some required functions on that object. This specification is called a pattern. A software pattern is used by designers to generate templates and specific instantiations for products. Object browsing is one of a few generic tasks performed by operators of telecommunications networks across a wide range of network and service management applications.

Source

ITU-T Recommendation Z.372 was approved on 13 April 2005 by ITU-T Study Group 4 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

Human-computer interface, human-machine interface, software patterns, user interface.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction

There are two types of interfaces in the telecommunications network management business:

- Between management applications and equipment; and
- Between the management applications and the human operators that use them.

The first type of interface is referred to as machine-machine interface, and the second type is referred to as the human-machine interface (HMI). This Recommendation deals with the HMI. In the ITU-T M.3000-series Recommendations, this interface is identified as the "G" interface.

It has long been recognized by the industry that machine-machine interfaces are key components of network management applications. Allowing these applications to exchange data and invoke management functions automatically using standard protocols has been the primary focus of the ITU-T and the TMF.

Typically, the role played by the HMI has not been as well understood. Although substantial investments in HMIs have been made, there has not been a clearly articulated strategy that could be implemented across the required range of application interfaces. This has resulted in substantial duplication of effort, high development costs, and a lack of consistency in the HMIs produced. Improper or inadequate design at this interface can result in significant problems in the network in terms of reliability and operational efficiency.

The impact of this situation should not be underestimated. Between 60% and 70% of the software in typical network management applications is related to the HMI. In addition, the functional requirements that HMI components need to satisfy are at least as complex as those involved in machine-machine interfaces. In fact, for many service providers, the HMI is the application, since it is the only component with which the operator interacts. Telecommunications network operators manage their infrastructure and services via HMIs to operations systems (OSs) and management applications.

A number of standards bodies have developed some general standards for this vital interface; however, this work has been limited in scope to very generic issues, or to the high-level design principles used to guide the development of HMIs. Other HMI standards include T1.232 (ANSI), ISO 9241 and ITU-T Rec. Z.361 [3]. In addition, the TMF released TMF-046 in 2002 that deals with telecom objects provides information requirements for graphic displays. TMF-046 has been submitted to the ITU-T and has been given the number ITU-T Rec. Z.371.

This Recommendation provides standard HMI patterns and templates for telecommunications network operations support systems (OSS). These application patterns and templates specify the high-level HMI components, structure, and behaviour for the most common tasks encountered by operators in the performance of their day-to-day duties. An example is the frequent set of actions involved in retrieving, viewing and selecting a specific object from a large set of objects. We have identified a HMI pattern called object browsing that facilitates this series of frequent actions. Isolating a customer record that requires some attention from a large set of records is a good example of a task that is facilitated by the object browsing pattern. Another example would be the selection of a particular network element from the set of network elements within the area of responsibility of the operator.

A very large percentage of tasks associated with managing a network are facilitated by a small number of frequently occurring HMI patterns such as Object Browsing. These patterns are used as the basis for specific task templates such as a customer record viewer.

Standardized task templates will benefit both vendor and customer communities equally. Development costs will be significantly reduced as toolkits evolve to include these standard templates. This cost reduction will be passed onto the customers. Service providers will benefit through products that are easier to learn, understand and less likely to produce errors on the part of the operator.

ITU-T Recommendation Z.372

Templates for telecommunications human-machine interfaces

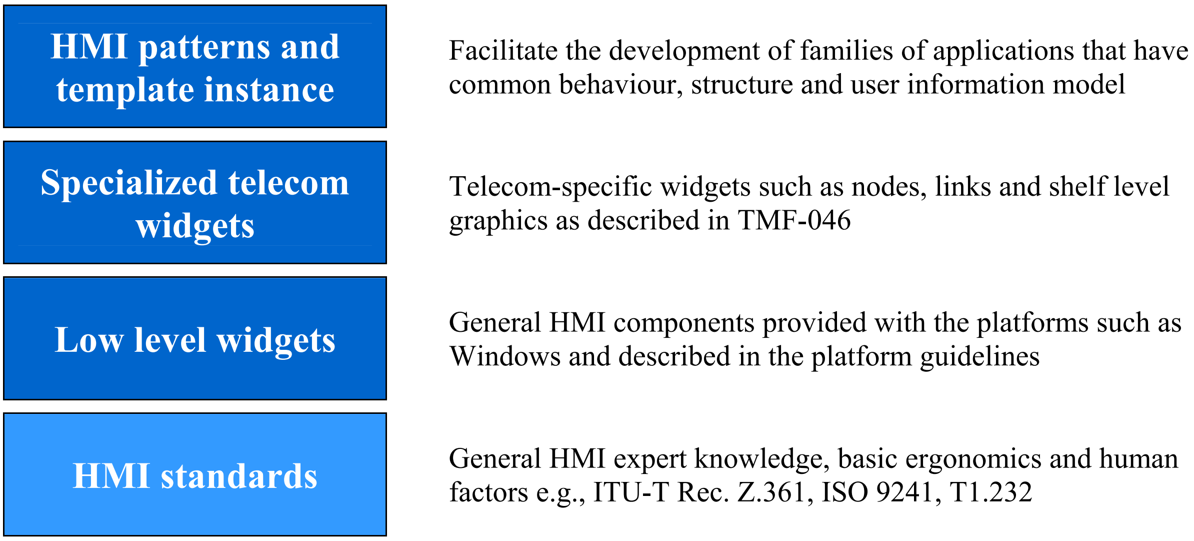
1 Scope

This Recommendation describes requirements for the "G" interface of the telecom operations environment. This "G" interface, identified in the ITU-T M.3000-series Recommendations of requirements, specifically ITU-T Rec. M. 3010, defines the interface between the user and the workstation or HMI.

The software that instantiates a HMI or a "G" interface (the "G" interface is an M.3000-series reference) can be described as shown in Figure 1-1. This architectural description provides an improved means of specifying and understanding the complex relationship between the system and the user.

Figure 1-1 shows three levels of complexity of user interface presentation elements, namely, low-level widgets, specialized widgets, and templates, all underpinned by HMI standards. The lowest levels of HMI elements are the most thoroughly standardized. These low level elements are provided by the platform used, for example, the Windows platform. They are described in the guidelines provided for these platforms by the manufacturer.

The foundation for all HMI presentation levels is generally HMI or user interface standards. In ITU-T Rec. Z.361 [3] the ITU-T has standardized a set of high-level design principles to assist designers as they assemble task managers using the objects discussed in these three sections. There are also sources of good HMI design in the field, in the platform guidelines and in other standards documents.



HMI patterns and template instance

Facilitate the development of families of applications that have common behaviour, structure and user information model

Specialized telecom widgets

Telecom-specific widgets such as nodes, links and shelf level graphics as described in TMF-046

Low level widgets

General HMI components provided with the platforms such as Windows and described in the platform guidelines

HMI standards

General HMI expert knowledge, basic ergonomics and human factors e.g., ITU-T Rec. Z.361, ISO 9241, T1.232

Figure 1-1/Z.372 – HMI architecture

HMI standards: General human factors knowledge, for example, ITU-T Rec. Z.361 [3], ANSI T1.232 [7], ISO 9241, and the platform guidelines provided by the manufacturer.

Low level widgets: Part of any HMI is a set of low-level elements such as buttons, windows, and scroll bars. These are defined by the specific user interface style or toolkit supported at the workstation (OSF Motif, Windows, JAVA, etc.). These platform styles are aptly described in style guides provided by the manufacturer. These HMI elements are relatively stable and common across the range of systems or products.

Specialized widgets: TMF-046/ITU-T Rec. Z.371 provides requirements for the second layer of the architecture, namely the specialized widgets.

The patterns and templates: The top level patterns in the diagram in Figure 1-1 are created when HMI architects and designers analyze a wide range of applications or task managers and discern a repeated set of small steps or actions. These steps or sub-tasks, when coupled with a normalized workflow, form a pattern. Application patterns and templates are the focus of this Recommendation.

Note that in TMF-046/ITU-T Rec. Z.371 the top-level class was called task managers, and herein is called templates or patterns, a pattern being a generic source for the specific template designs required for a task or task manager. Task managers are the most complex aspect of the HMI to design and are where the most benefit will be derived from standardization.

A vast majority of the tasks in network operation centres would benefit from a standard way of presentation and behaviour. Much of the costly work required to design and learn these interfaces is eliminated when the use of a standard becomes the norm, and when this standard is available in the GUI builders or toolkits.

Standardization of these interfaces produces three key values for service providers, which are:

- The HMIs delivered by vendors will be more easily learned and more easily remembered by the network operators. This will reduce training times and reduce scheduling problems.
- As operators learn these interfaces and as they understand the underlying patterns, they will make fewer mistakes.
- As learning is eased and mistakes are reduced, the overall efficiency of the work force will increase.

Standardization of these interfaces also produces significant value for the product developer:

- Reduced development time and lower cost, allowing more time for feature development and code quality.
- As new system integration occurs it becomes increasingly important that the multiple products that make up the support systems have standard HMIs.

These HCI requirements are based on standard available information technology. They strive to maximize usability and interoperability of applications or products from different vendors.

This Recommendation was written in co-operation with the Telemanagement Forum. The Telemanagement Forum retains ownership of TR129 and copyright of the underlying intellectual property. The ITU-T owns the copyright of ITU-T Rec. Z.372.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[1] ITU-T Recommendation M.3100 (2005), *Generic network information model.*

[2] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network.*

[3] ITU-T Recommendation Z.361 (1999), *Design guidelines for Human-Computer Interfaces (HCI) for the management of telecommunications networks.*

[4] Telemanagement Forum TMF-046/ITU-T Recommendation Z.371 (2005), *Graphic information for telecommunication management objects*.

[5] Telemanagement Forum TR129 Version 2/ITU-T Recommendation Z.372 (2005), *Templates for telecommunications human machine interfaces*.

[6] ITU-T Recommendation M.3050.0 (2004), *Enhanced Telecom Operations Map (eTOM) – Introduction*.

[7] ANSI T1.232-1996 (R2001), *OAM&P – G Interface Specification for Use with the Telecommunications Management Network (TMN)*.

3 Definitions

This Recommendation defines the following terms:

3.1 pattern: A pattern is a recurring solution to a standard problem. When related patterns are woven together they form a "language" that provides a process for the orderly resolution of software development problems. Pattern languages are not formal languages, but rather a collection of interrelated patterns, though they do provide a vocabulary for talking about a particular problem. Both patterns and pattern languages help developers communicate architectural knowledge, help people learn a new design paradigm or architectural style, and help new developers ignore traps and pitfalls that have traditionally been learned only by costly experience.

Source: *Special Issue on Patterns and Pattern Languages*, Vol. 39, No. 10, October 1996.

3.2 template: A template is a Telecom specialized version of a generic HMI pattern. A template will have those HMI components required for the task for which it is designed. An example of a template derived from the browser pattern is a fault manager. The template described in this Recommendation is the Network Component Manager (NCM).

3.3 HMI component: A component[1] is a high-level user interface mechanism such as a tree or a set of buttons that have a specific function. Examples include: filters cluster – setting the criteria for retrievals from a database; details area – a region of the screen reserved for the detailed information related to a specific object of interest. We use component generally to describe an area of the screen with distinct functions that may be required or may be optional in a given pattern or template.

3.4 task manager: Task managers are created when specific controls, information elements, and widgets are assembled by developers to implement a specific application's functionality. They are the final HMI as experienced by the user.

4 Abbreviations

This Recommendation uses the following abbreviations:

ASIC	Application-Specific Integrated Circuit
CASE	Computer-Aided Software Engineering
eTOM	enhanced Telecom Operations Map
GUI	Graphical User Interface
HCI	Human-Computer Interface
NCM	Network Component Manager

[1] The use of the term "component" is limited to this Recommendation and does not imply or refer to any other use or meaning.

QoS	Quality of Service
SLA	Service Level Agreement
TMF	Telemanagement Forum
UI	User Interface

5 Conventions

This Recommendation details the requirements for patterns and templates that may be useful in any application user interface but have been specifically related for use in the telecom industry. Compliance with this Recommendation is voluntary. For an application to be considered compliant, the user interface must provide the minimum properties described when using any of the constructs defined in this Recommendation.

(R) designates those aspects which must be present if the application is to be considered compliant with this Recommendation.

(O) designates optional aspects of the Recommendation. However, it should be noted that where appropriate given the context, it is desirable for the HCI to use these aspects as well. Also note that there are cases where a component or area of a template may be optional but, if it is included, then there are required **(R)** aspects to the included component.

6 Framework

6.1 HMI patterns

In the past, vendors have thought that the HMIs for their products provided strategically important product differentiation. However, the truth is that, in general, there is no strategic value in designing different interfaces for many of the normal day-to-day tasks people are faced with. In fact, there is negative value for the following important reasons:

- The cost of rebuilding HMIs from scratch every time a new project comes on line. Recall that 60% or more of code costs relate to the HMI.
- Passing on this unnecessary cost to customers.
- Requiring that the operators learn the new HMI and make mistakes because each HMI is different in task flow and behaviour.
- Hindering UI integration. HMIs are almost never used in isolation and, as the industry relies more and more on multi-vendor solutions, the differences in the HMIs will exacerbate the problems for the user.

This practice has led vendors to guard the IPR of their HMIs, even when they are poorly designed. The Telecom industry, through the ITU-T and the TMF, is creating the required mechanisms to communicate and share the expertise, wisdom, experience and knowledge of their respective HMI designers and architects. The result will be significant added value for the customer.

Generic HMI patterns and telecom task templates, along with TMF-046/ITU-T Rec. Z.371 are the first significant steps taken to make the design of HMIs in the Telecom space more efficient and effective. Our intent is not to limit design creativity, but to liberate the time wasted redesigning so that it may be applied to more complex and challenging aspects of the product.

The use of software patterns is providing a powerful tool to the software designer. Hardware designers have used similar techniques for years and no one would dream of starting from scratch to design an ASIC or a power supply. HMI designers will benefit as well from pattern use. The use of patterns is motivated by the following values (see [B13]).

6.1.1 Success is more important than novelty

The longer a pattern has been used successfully, the more valuable it tends to be. In fact, novelty can be a liability because new techniques are often untested. Finding a pattern is a matter of discovery and experience, not invention. A new technique can be documented as a pattern, but its value is known only after it has been tried. This is why most patterns describe several uses. Thus, HMIs, like other patterns, should only be changed if clear value is provided which is greater than the value of reduced training and greater user familiarity with a well-understood pattern).

6.1.2 Emphasis on writing and clarity of communication

Most pattern descriptions document recurring solutions using a standard format. In many cases, projects failed because developers were unable to communicate good software designs, architectures, and programming practices to each other. Well-written pattern descriptions improve communication by naming and concisely articulating the structure and behaviour of solutions to common software problems.

6.1.3 Qualitative validation of knowledge

Another aspect is to qualitatively describe concrete solutions to software problems, instead of quantifying or theorizing about them. There is a place for theoretical and quantitative work, but such activities are more appropriate in a context separate from discovering and documenting patterns. The goal is to appreciate and reward the creative process that expert developers use to build high quality software systems.

6.1.4 Good patterns arise from practical experience

Every experienced developer has valuable patterns that can be shared. The experience of HMI architects and software developers can be engaged. The combined experience of both vendor and service provider HMI experts will produce high quality HMI patterns.

6.1.5 Recognize the importance of human dimensions in software development

The purpose of patterns is not to replace developer creativity with rote application of rigid design rules. Neither is the intent to replace programmers with automated CASE tools. Instead, the intent is to recognize the importance of human factors in developing HMI software. Applying these software engineering concepts to HMI patterns provides the same advantages to the HMI architects. Analysis of the work that people do in operations centres using task analysis shows that although there are many tasks, most tasks are comprised of a relatively small set of recurring generic actions. A small number of HMI patterns can be defined which enable these recurring actions. It is surprising how significant this simple work flow analysis requirement is, and how often it is ignored or never considered.

The HMI pattern in clause 7 and the templates in clause 8 will provide a standard way of addressing a key recurring action. Operators can learn standard patterns and, when faced with familiar tasks but applied to a different product or service, will already know generally how to proceed. Their expectations will be reinforced, learning will progress rapidly, and error production will drop. Over time, these patterns will become embedded in the GUI toolkits, further reducing costs and optimizing good design practices.

This Recommendation provides specifications for the most prevalent pattern – the object browser.

6.2 Normalized interaction flow for HMI patterns

Research shows that in Western culture, the workflow across any given screen is most efficient and effective when arranged from top left to bottom right. This assignment is the result of much learned behaviour that flows from the way books and documents are read in Western culture.

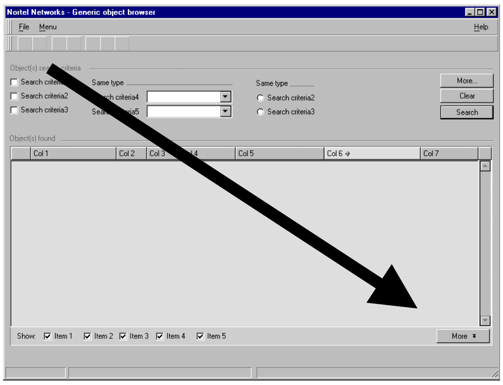


Figure 6-1/Z.372 – Normal workflow on a screen

This workflow will be the norm for all patterns and templates described in this and subsequent versions of the Recommendation. It is fundamental to the standardization of these HMIs. This workflow will seem familiar to a user and if reinforced through a number of tasks, it will promote learning, remembering and ease of use.

NOTE – Other cultures may be familiar with different workflows, and it is expected that the toolkit vendors will build in the required options to accommodate these situations.

6.3 Example of HMI pattern development

The first step is to understand how the high level task, fault management, for example, is divided into a number of sub-tasks. This process starts with experience and analysis of the tasks and subtasks required, knowledge of the work practices of the responsible departments, and experience with the models from ITU-T Rec. M.3400 and the Telemanagement Forum's eTOM documents. Analysis of how these sub-tasks are done in the field permits the HMI architects and designers to discern a pattern or patterns. Using the above fault management case, a number of subtasks emerges, for example:

Surveillance: First alert;

Localization;

Check affected services;

View affected elements;

Determine a repair strategy;

Issue repair orders;

Verification.

HMI architects should also look at related management domains, such as Configuration, Performance and Security, to find cases where the same pattern applies. Examples include checking inventory lists to support planning and installation in configuration, reporting statistics in performance, or viewing logs in security.

Having analyzed these sub-tasks, the HMI expert then extracts the common pattern or patterns. As mentioned in the introduction, the recurring set of actions involved in selecting a single object from a large set of objects is supported by an object browsing pattern. This task has four steps:

1) Set the retrieval criteria, for example the "North East region";
2) View the list of retrieved objects; all network elements in that region;
3) Filter the retrieved list and select a single object; those with severe alarms set against them;
4) View the details pertaining to that object; parameters and alarms.

The final step is to create a template which enables the task steps to be followed using the top left to bottom right normalized interaction flow.

The result is shown in Figure 6-2. This defines the required HMI components for the task. Each component will have a behaviour and a set of attributes that, together with the other HMI components, will allow the user to complete the task for which the pattern is designed.

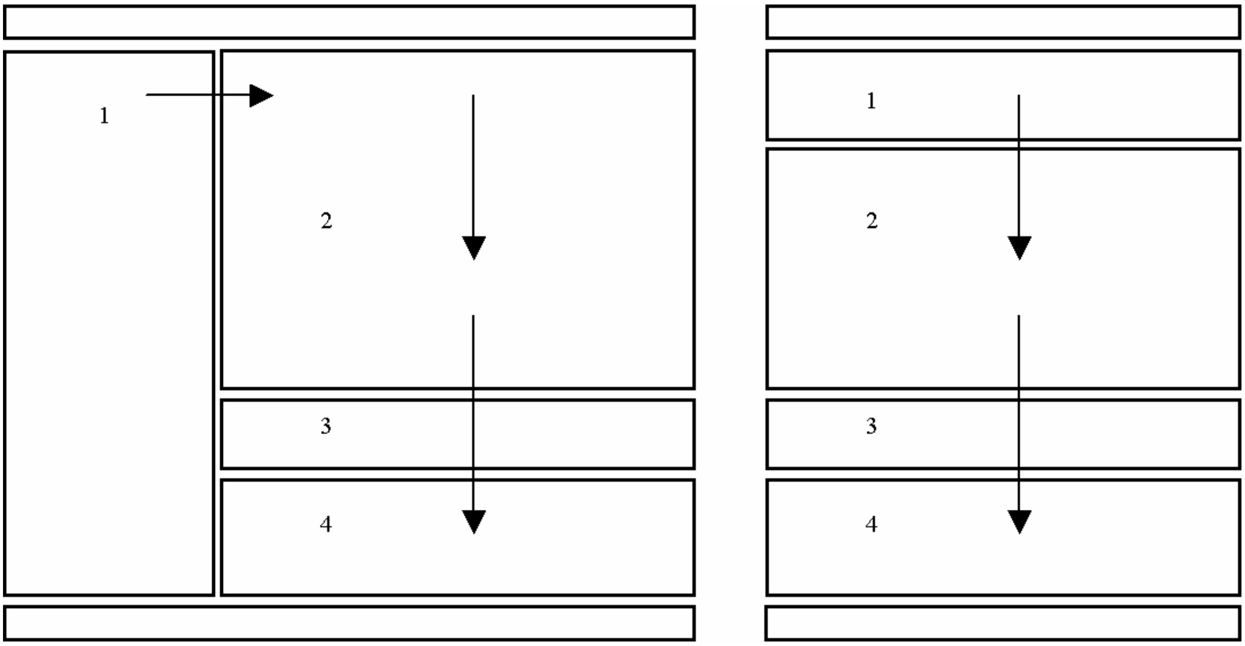


Figure 6-2/Z.372 – The object browsing pattern

Area 1: The selection criteria areas allow the user to specify a subset of data to be retrieved. Either a text-based selection criteria area or a tree is used.

Area 2: The results of the search are displayed in this region.

Area 3: Sets of filters are sometimes provided to further select specific data from the retrieved data. The data on which filters operate is now local to the workstation and no data retrieval is processed. The result for the user is typically one of speed of response.

Area 4: Details regarding a selected object are displayed in this area.

Standardizing such patterns constitutes a radical departure from traditional GUI design in the Telecom industry and represents a significant value proposition to both the vendor and service provider communities.

This new innovation will provide HMI designers with the main building blocks for most interfaces and will dramatically reduce the cost and churn related to the HMI. These templates will be incorporated in the GUI toolkits and enable "standardized" user interfaces to be developed.

6.4 Relationship to the eTOM

The eTOM business process framework is used in this Recommendation to provide the framework for identifying the business processes, models, tasks and sub-tasks which can be analyzed to determine recurring sets of actions that could discern a common pattern.

Note that the eTOM is now available from the ITU-T as ITU-T Rec. M.3050.0 [6].

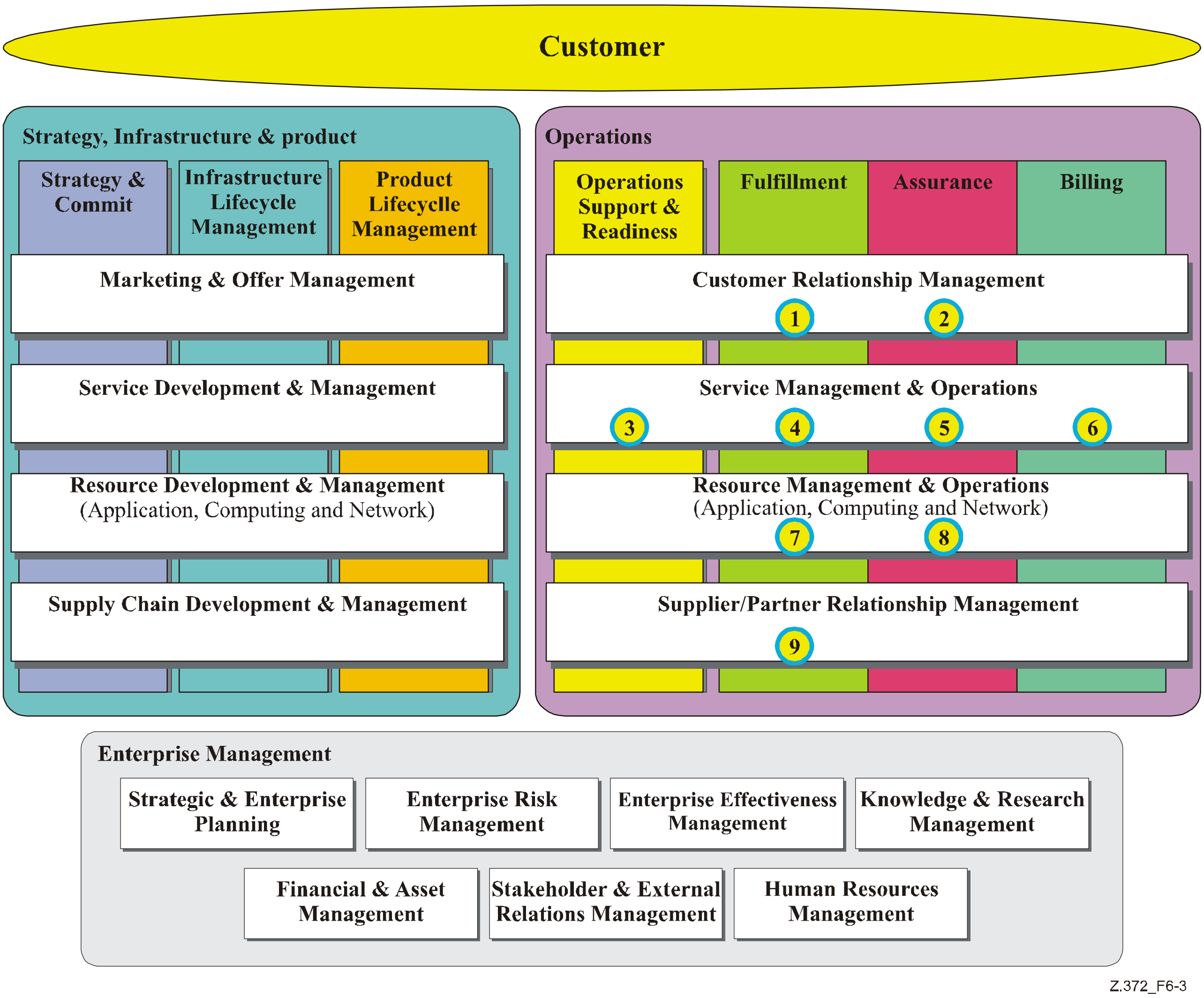

Customer
Strategy, Infrastructure & product
Strategy & Commit
Infrastructure Lifecycle Management
Product Lifecyclle Management
Marketing & Offer Management
Service Development & Management
Resource Development & Management
(Application, Computing and Network)
Supply Chain Development & Management
Operations
Operations Support & Readiness
Fulfillment
Assurance
Billing
Customer Relationship Management
1
2
Service Management & Operations
3
4
5
6
Resource Management & Operations
(Application, Computing and Network)
7
8
Supplier/Partner Relationship Management
9
Enterprise Management
Strategic & Enterprise Planning
Enterprise Risk Management
Enterprise Effectiveness Management
Knowledge & Research Management
Financial & Asset Management
Stakeholder & External Relations Management
Human Resources Management
Z.372_F6-3

Figure 6-3/Z.372 – eTOM level 1 processes

Figure 6-3[2] shows the level 1 processes involved in the eTOM with the major process groupings consisting of Strategy, Infrastructure and Product, Operations, and Enterprise Management. Looking for recurring instances of the object browsing pattern is not difficult. Almost any modern HMI has to accommodate the user's need to select a single object from a large set of objects. To specifically identify cases, which might be unique to the Telecom space, the business processes in the operations section are detailed here. These examples facilitate making a robustly defined pattern definition and help to identify the most useful candidates to provide detailed template examples.

6.4.1 Order handling

In Figure 6-3, note the circled number 1.

Order handling HMIs need to be able to browse the current product offerings and easily drill down to check availability, feasibility or special solutions that may be applicable to a particular customer. They need to browse and filter for customer records and the status of particular orders.

[2] The actual eTOM process diagram may have evolved from this version; however, the point is that HMI patterns apply to all business areas.

6.4.2 Customer QoS/SLA management

In Figure 6-3, note the circled number 2.

A QoS service HMI needs to be able to browse SLAs by service descriptors, customer contracts or product catalogues. Drill down and filtering operations involve finding the applicable SLAs for a particular customer or service and determining if any violations have occurred. Filtering needs to be done based on severity of violation. Once an SLA has been identified, further drill down is necessary to determine components of the SLA-like network or resource availability and other regulatory parameters like %completion on time for order requests, time to repair commitments or customer contact performance.

6.4.3 Manage service inventory, configuration & activation

In Figure 6-3, note the circled number 3.

A service inventory HMI needs to be able to browse and update the service inventory database. These service instances must be able to drill down to resources related to supporting each service and current capacity levels to determine if system configuration, activation and provisioning are prepared to support the offered services.

6.4.4 Service configuration & activation

In Figure 6-3, note the circled number 4.

The service configuration & activation HMI must be able to browse the service inventory database, and the available resources for fulfilling the service. Filtering by service, customer or alerts, should enable drill down to specific work orders to derive information about the status of implementation, service testing, and service activation.

6.4.5 Service problem management

In Figure 6-3, note the circled number 5.

The service problem management HMI must be able to browse services and associated resources. These browsers may be network topology related so that physical and logical views of the network can be displayed and drill down to related sub-networks to identify problem areas. Browsing by services, customers or related network element will allow drill down and filtering to related alarms and trouble ticket status.

6.4.6 Service & specific instance rating

In Figure 6-3, note the circled number 6.

Service and instance rating HMIs must allow browsing by service or customer and drill down and filter rate usage records from the resource layer. Usage records are then often filtered by other parameters, which might indicate the need for applying tariffs or identifying abnormalities such as fraudulent activity.

6.4.7 Resource provisioning

In Figure 6-3, note the circled number 7.

Resource provisioning HMIs must be able to browse for specific service instances and for available resources. Service instances can then drill down to specific order requests to display status of requested network capacity, equipment, software, shipment, or installation.

6.4.8 Resource trouble management

In Figure 6-3, note the circled number 8.

Resource trouble management HMIs provide browsing of resources and reported alarms or other events. Resources might be browsed with physical and logical views of networks, which can drill down to subsystems, shelves, racks, and individual cards, which may be the cause of an event. Events and alarms can be filtered by a number of criteria to suppress redundant, transient or implied events. Events may drill down to an associated trouble report. Trouble reports can then be further filtered by status and priority.

6.4.9 S/P purchase order management

In Figure 6-3, note the circled number 9.

Purchase order management need to browse by partner or other purchasing channels, product type or services type to gather purchase orders and filter by fulfilment date to ensure on-time and correct delivery requested by the enterprise.

The next clause, clause 7, presents the first of a series of HMI patterns. The object list browser pattern is perhaps the most frequently encountered pattern and will be the basis for a large number of task and subtask templates.

Subsequent Recommendations will add other patterns and template examples. It is expected that vendors will come forward with pattern concepts in other domains such as service management. The HMI team of the Telemanagement Forum will review these patterns and, in consultation with other HMI experts, select the best patterns offered for standardization. This process will be refined in consultation with ITU-T, the Telemanagement Forum and with the companies who offer their patterns for standardization.

Legend for diagrams

Colour-coding has been adopted to help distinguish different HMI component areas in the diagrams used throughout this Recommendation. This coding is not to be used in the creation of actual product implementations. Note the examples of real screens such as indicated in Figure 7-2.

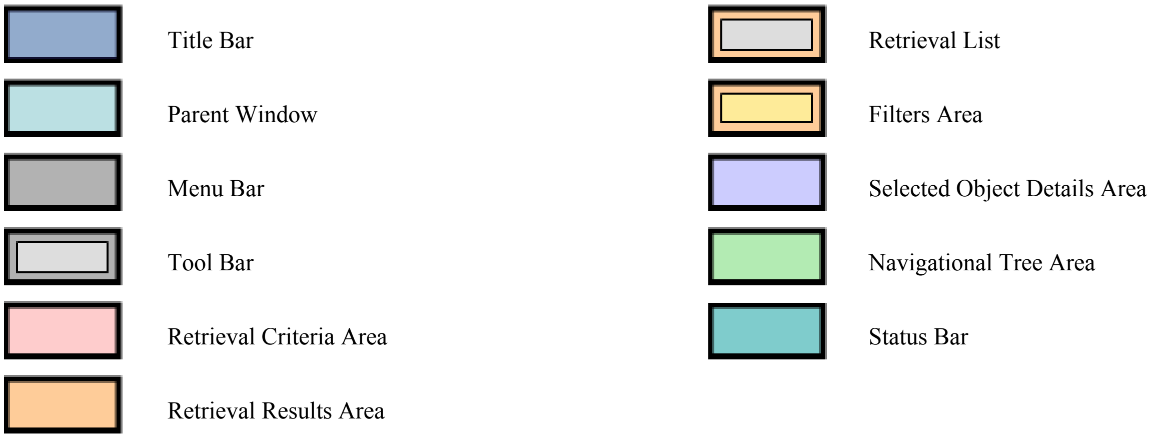


Figure 6-4/Z.372 – Colour legend

The diagrams in this Recommendation are not intended to imply any particular toolkit or platform.

7 Object browser pattern

This Recommendation applies to object browsers. Subsequent releases of the Recommendation will deal with other patterns.

An object browser allows users to display and manage, or edit, sets of similar objects. Object browsers support a standard set of tasks enabling the user to:

- Retrieve a particular set of objects based on a set of attributes;

- View the list of objects and a subset of attributes; filter the list of instances;
- Select a specific object from the list;
- View attribute details for a specific object;
- Edit an object's attributes; in the case of a simple object (for example, trouble ticket);
- Open additional tools to perform complex tasks (for example, advanced editor).

There are essentially two main types of browsers: Browsers that have a tree component and browsers that do not. Browsers with trees do not have selection criteria areas, and browsers without trees have the text-based retrieval criteria area. Other HMI components are the same for both types.

7.1 Object browsers – using text-based retrieval criteria

Object browsers can be launched the way all applications can be launched, e.g., from a menu item, a command line interface or icon on desktop.

7.1.1 When to use

Use a text-based browser when the data, and thus the search criteria, are not particularly hierarchical and when the data can easily be described by text and numeric fields.

Object browsers can be launched in the same way that any application can be launched, e.g., from a menu item, a command line interface or icon on desktop.

7.1.2 Template

The layout is designed to represent the flow of steps or subtasks for object management, that is:

Step 1: Retrieval of a set of objects is accomplished through the use of the retrieval criteria area. The user fills in the appropriate fields and initiates the retrieval. A key issue is the feedback to the user dependent on the number of objects retrieved and the time it will take to retrieve them from the database.

Step 2: The user can now review the retrieved objects, scroll through the list if needed, and decide if further filtering is required.

Step 3: A common example might be to filter an alarms list based on the severity of alarm or on the date of the alarm.

Step 4: Once a specific object has been selected from the retrieved data set, the attributes of that object can be displayed in the object details area. The user may be able to edit these attributes in order to correct information about the object.

Further tools may be opened from the menu area to assist the user in the completion of the task. For example, it may be necessary to initiate the creation of a report on a particular card or service in order to determine the correct action.

These activities are dependent on the nature of the object being observed and nature of the task the user is performing. However, the organization of the object browser and its behaviour can be consistent regardless of the specific task details.

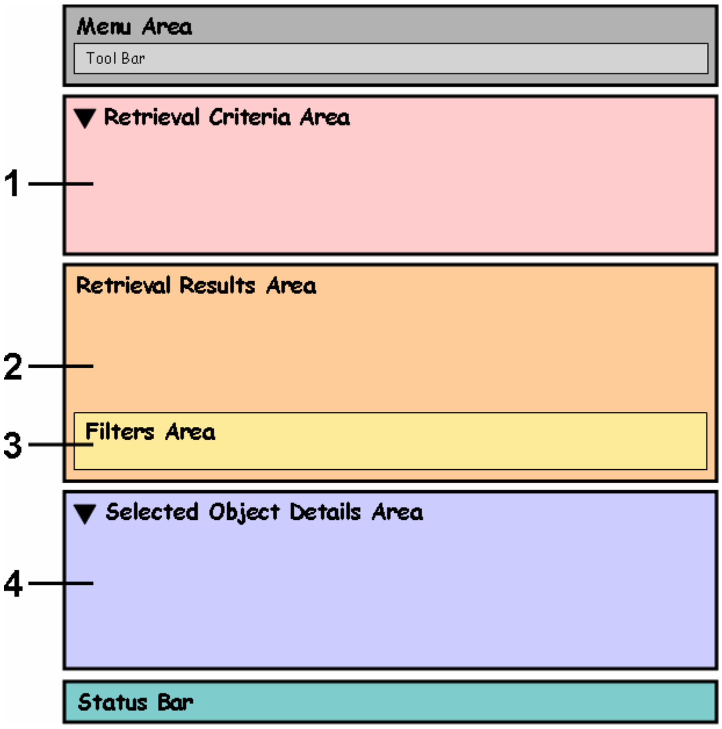
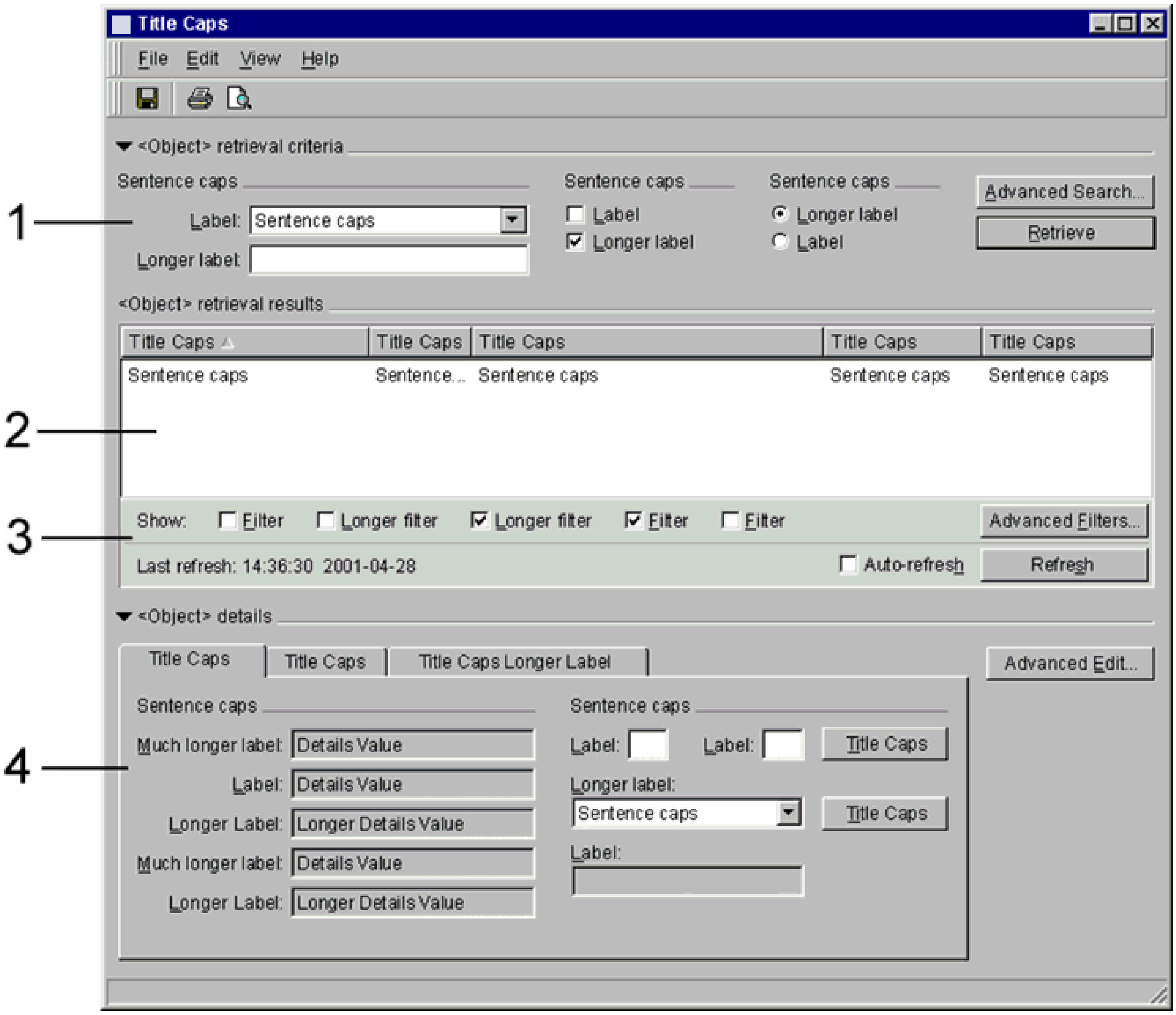


Figure 7-1/Z.372 – Flow of activities through the text-based object browser



Label: Sentence caps

Longer label:

□ Label

☑ Longer label

● Longer label

○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps	Sentence...	Sentence caps	Sentence caps	Sentence caps

□ Filter □ Longer filter ☑ Longer filter ☑ Filter □ Filter

Last refresh: 14:36:30 2001-04-28

□ Auto-refresh

Much longer label: Details Value

Label: Details Value

Longer Label: Longer Details Value

Much longer label: Details Value

Longer Label: Longer Details Value

Label:

Label:

Longer label: Sentence caps

Label:

Figure 7-2/Z.372 – Generic text-based browser example

7.1.3 HMI components

The list-based instance browser has these HMI components:

- (R) Title bar: displays the name of the window or dialog and optionally instance information;
- (R) Menu bar: provides menus for the application;
- (O) Toolbar: provides access to frequently used commands using icons;
- (O) Retrieval criteria area: user enters attributes of instances to be retrieved;
- (R) Retrieval results (instance list) area: lists the instances matching criteria;
- (O) List filters area: filters the list of retrieved instances;
- (O) Selected object details area: displays the instance details;
- (O) Status bar allows the system to report the status of activity initiated by the user.

7.1.3.1 Title bar (R)

(R) The title bar is mandatory in all windows and dialog boxes. It provides the name of the window or dialogue box.

The title bar must follow the guidelines set out in the platform guidelines. It will look like the diagram in Figure 7-3.

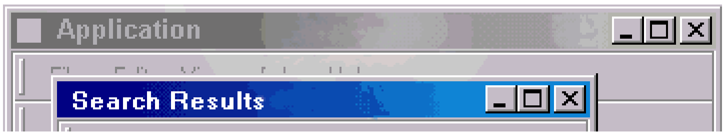


Figure 7-3/Z.372 – Window title bar

(O) Text in the title bar should use this format, which follows the Windows' recommendation:

<NE> – <application name> – <management system>

- the name of the NE/object being managed (SP, NP, NE);
- the application name – if applicable (Map Topology, Performance Manager, etc.);
- the management system.

7.1.3.2 Menu bar (R)

(R) The menu bar is mandatory.

(R) The menu bar contains menu items that open applications and commands that affect the application currently in use. The menu bar is located directly below the title bar. The left to right order of items in the menu bar is mandatory.

A typical menu in a browser window is shown in Figure 7-4.

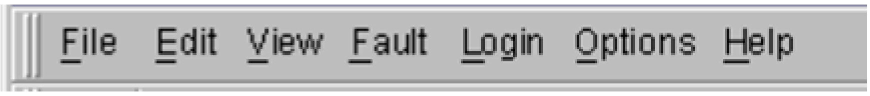


Figure 7-4/Z.372 – Typical menu

- (R) **File** menu is mandatory. It is first in the menu bar. It contains menu items that affect the entire application, such as Save or Print. The last menu item closes the window and any applications running within it.
- (O) **Edit** menu is optional. It may contain the standard Cut, Copy, and Paste actions. Include additional actions as needed.
- (O) **View** menu is optional. It contains all commands that affect the appearance of the window (for example, Show/Hide Retrieval pane and Show/Hide Filters).
- (O) **<functional menus>** may be containers for the most useful or frequently used menu items for the specific application. Alternatively, a menu may be named after the main object being managed (for example, alarms or trails), and contain all menu items that perform actions on this object.
- **Help**:

  (R) is required if the system provides on-line help.

  (O) is optional if the system does not provide on-line help.

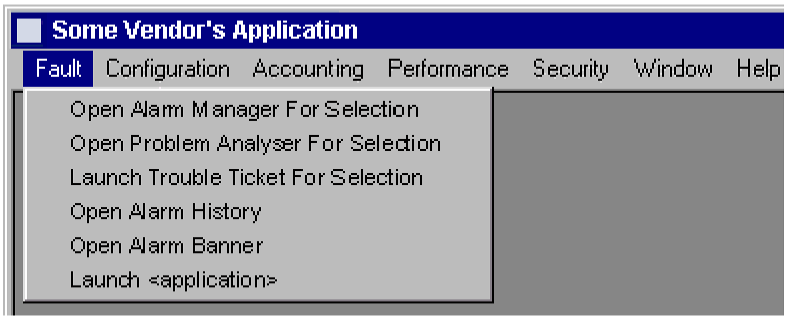

Some Vendor's Application
Fault
Configuration
Accounting
Performance
Security
Window
Help
Open Alarm Manager For Selection
Open Problem Analyser For Selection
Launch Trouble Ticket For Selection
Open Alarm History
Open Alarm Banner
Launch <application>

Figure 7-5/Z.372 – Example of an opened menu item

(R) A general rule applies to the menu bar item lists. Any command that applies to an object and appears on buttons or object menus must also appear in the main menu item lists. Exceptions are buttons such as "Okay" or "Cancel".

7.1.3.3 Tool bar (O)

(R) Toolbars provide access to frequently used commands. Toolbars use icons as command buttons (see Figure 7-6). Toolbar icons are never unique within the interface. That is, commands that appear in the toolbar must also appear as a menu bar item, or object pop-up menu item.

Use the toolbar to provide quick access to frequently used commands, and standard menu items such as Save and Print.

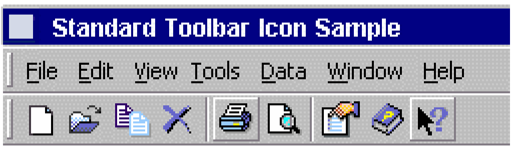


Figure 7-6/Z.372 – Toolbar example

(R) Do not re-invent icons, reuse existing icons.

(R) Toolbars appear directly below the menu bar. Use these guidelines for layout:

- Place commonly used toolbar icons such as Print and Save on the left of the toolbar;
- Place application-specific icons to the right;
- Place the Help toolbar icon furthest to the right.

7.1.3.4 Retrieval criteria area (O)

(O) The retrieval criteria area is optional. It is optional as the results area may already be populated when the window is opened as a result of previous application activity or the context of the browser launch.

(R) Users enter criteria in this area and instances matching the criteria populate the object results area.

(R) Key retrieval criteria are shown in the retrieval criteria area of the object browser. Secondary retrieval criteria can be shown in modal dialogs. The modal dialog should be accessed via a button in the retrieval criteria area.

(R) The retrieve operation is performed when the user clicks the "Retrieve" button located in the Retrieval criteria area. In all cases, clear indication of the scope of the retrieval and clear progress indication is necessary.

(O) Two radio button options for populating instance lists may optionally be provided as follows:

- **Append to List** adds newly retrieved instances to the existing list.
- **Replace List** clears the existing list and replaces it with newly retrieved instances.

(O) Optionally, selection sets of advanced filters can be named and saved. The saved filters should be available from a pull-down selector in the filters area. The selected saved filter can be enabled or disabled using a check box beside the pull-down selector.

7.1.3.5 Retrieval results (R)

(R) The retrieval results area is mandatory. It lists the instances that matched the retrieval criteria.

(R) Label the instance list area "<object> retrieval results."

(O) It is also recommended to show the count of retrieved items out of the total records searched (if known) and the retrieval completion time.

(R) When a text-based browser is launched in context of an object, the results area should be populated with content relative to the launched context.

(R) When the user clicks on an item in the list, the item is highlighted, and its details are displayed in the Object details area. Through the standard shift-click, control-click, and drag mechanisms, users can select multiple items. When more than one item is selected, the details area shows no specific attribute information, and commands that logically apply to only one object (such as "Show Equipment") are disabled. Commands applying to a number of objects remain enabled.

(R) Actions that apply to an object in the list should be available in a pop-up menu that is activated by the right-mouse click.

(R) Horizontal scroll bars must not be used in the default retrieval results area. The information required can be organized and displayed using one of the techniques mentioned elsewhere in this Recommendation, for example, tabs. Note that horizontal scroll bars, in general, are not acceptable as they significantly diminish an operator's effectiveness. However, if the user resizes a window, it may be necessary to add scroll bars to enable access to data in the resized window.

(R) When the items represented in the list carry a hierarchical or containment semantic, the user should be allowed to navigate, i.e., drill down to browse the contained items. This is typically done by double-clicking the item or via a pop-up menu item.

(O) Below the list there may be a check box to select "Auto-refresh" and "Refresh" modes. A time stamp, labelled "Last refresh", may show the last refresh time and date. The default state for the "Auto-refresh" check box is *on* if the list auto fills and *off* if the list does not auto-fill.

(O) When appropriate, alternative views may be made available using view toggle buttons. This is particularly useful when items can be represented graphically with a different paradigm than the table view. For instance, the nodes contained in a regional network may be represented as a list in a table view as well as a geographic view showing the topology of the network. When provided with various views, the user can understand various aspects of the items presented in the Retrieval results areas. In this example, the list view gives a textual view of some attributes of the network nodes, but does not help in understanding the topology. The geographic view makes the topology explicit but may not represent all of the attributes values. For consistency, both Retrieval results areas must show exactly the same objects. For examples refer to TMF-046/ITU-T Rec. Z.371.

(R) When browsing very large datasets, it is possible that the retrieval operation may take more than 2 seconds. In this case, it is necessary to display a wait cursor. When the retrieval operation may reach 10 seconds, it is also recommended to use acceleration techniques like lazy retrieval of the items. This means that only the visible portion of the list is actually loaded, while the rest is loaded only when the user scrolls the list. This technique generally makes the initial retrieval operation and display faster than 10 seconds, while slowing down the scrolling operation. Retrieval is performed in the background and should not impact the interactivity of the UI. When the user scrolls over an area of the list where data is not yet retrieved, ellipsis (…) should be displayed in the list and wait cursor should indicate to the user that the missing data is being retrieved.

7.1.3.6 Filters area (O)

(R) The filters area enables users to narrow down the retrieval results list using key attributes. The filters only operate on retrieved data. This allows faster manipulation of the data already retrieved.

(R) Key filters should be shown in the Filters area of the object browser. Secondary filters can be shown in a modal dialog if they do not fit in the Filters area of the object browser.

(O) Sets of advanced filters can be named and saved and would be available from a pull-down selector in the filters area. These sets would be enabled or disabled by check boxes in the selector.

7.1.3.7 Selected object details area (O)

(R) The Object details area displays the properties of a selected object in the list, and provides command buttons that apply to the selected object. The details area may be a tabbed area to accommodate a number of categories of information.

(R) Label the Object details area "<object> details." For example, a Shelf view title may be "Equipment details".

(R) Depending upon the data, the fields displayed in the selected object details area may be editable by the user. Editable fields are distinguished from non-editable fields following the platform guidelines.

(R) Key object properties should be shown in the details area of the object browser. Secondary editable object details can be shown in the advanced editing tool if they do not fit in the details area of the object browser. The advanced editing tool should be accessed via a button in the details area.

(R) Controls such as Print, Copy, Paste, are platform specific. However, if a copy/paste function is available it should copy the complete data record, not just the visible fields.

(O) Other command buttons may also be placed in the object details area. For example, in the case of an alarm manager, actions such as "Acknowledge" and "Clear" may be included.

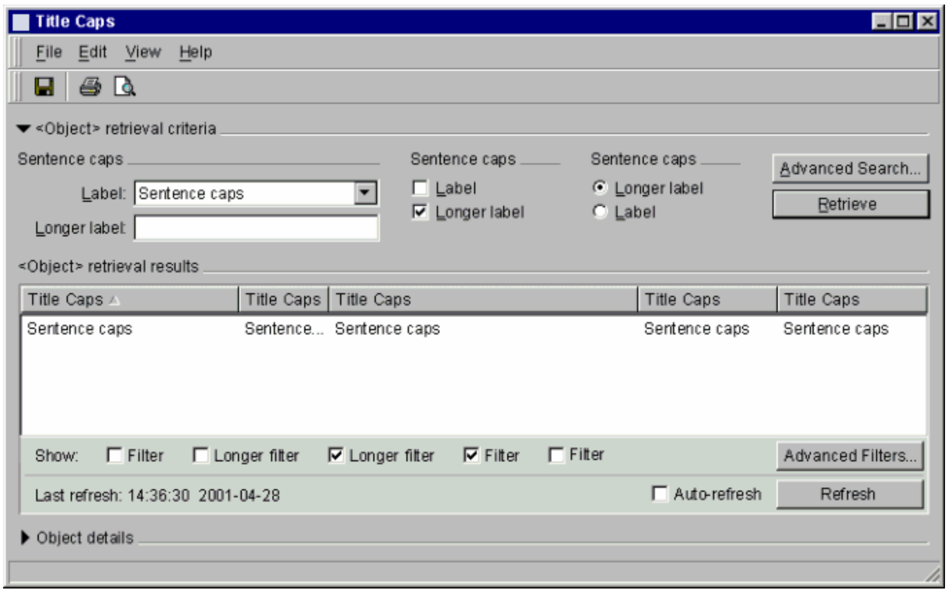
(O) When an object has many attributes in various data types, these details should be organized by categories. There are several main ways of displaying this data.

1) Fixed landscape plus tabs if required with data grouped by category. Most often required data is provided in the main area with tabs used for the rest.

2) Fixed portrait or table. This is displayed on the right of the Retrieval results area with a vertical shape. In this case, the details are not represented with a form but with a table containing columns. The first column represents the list of attributes, i.e., names and subsequent columns represent the attribute values. When allowed, editing of the values is performed in the table cells. This type of details area is sometimes named Property Sheet. It is very common in Integrated Development Environment (IDE) user interfaces. It offers a very compact view of the item details. This type of details area is recommended when the item has a limited number of detailed information and when edition is not allowed or allowed but possible via simple dropdown lists or checkboxes in the table cells.

3) The standard Windows alternative; Right Mouse menu – Properties, which is available to display details about a selected object. This mechanism should only be used when details or properties need to be viewed occasionally, depending on the task. Opening windows can obscure the object of interest and window proliferation must be avoided.

(O) Users can display or conceal the Object details area using a toggling View menu item. Also, the "bird beak" widget in the Object details title area collapses or expands this area.



Sentence caps

Label: Sentence caps

Longer label:

Sentence caps

☐ Label

☑ Longer label

Sentence caps

● Longer label

○ Label

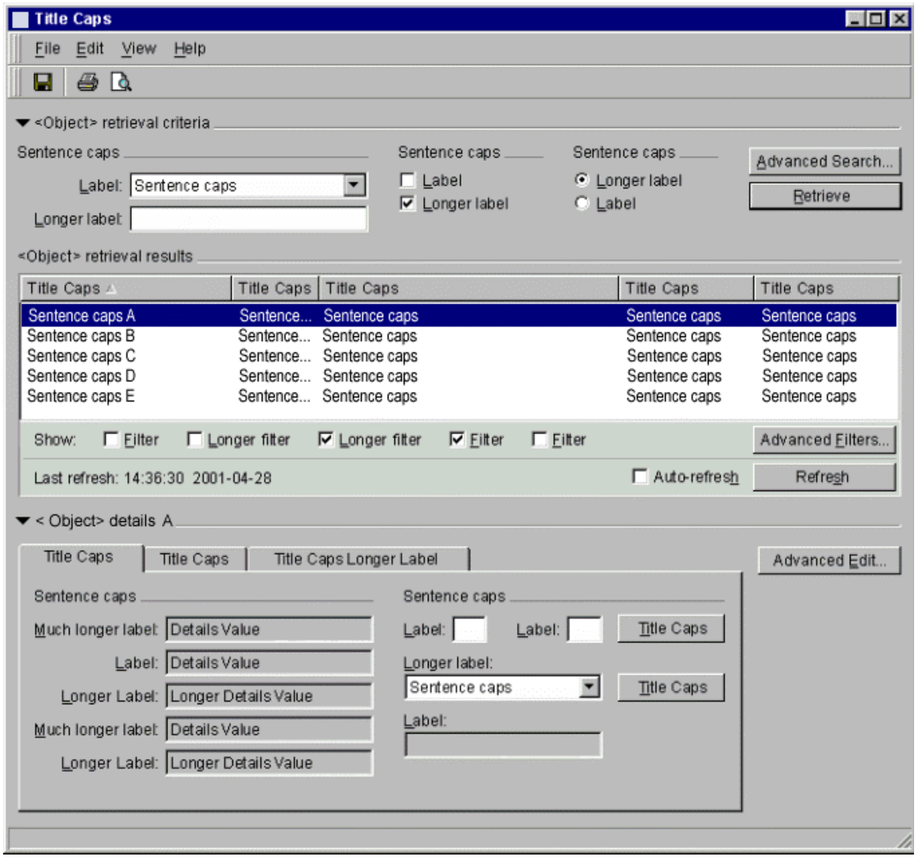
Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps	Sentence...	Sentence caps	Sentence caps	Sentence caps

Show: ☐ Filter ☐ Longer filter ☑ Longer filter ☑ Filter ☐ Filter

Last refresh: 14:36:30 2001-04-28

☐ Auto-refresh

Figure 7-7/Z.372 – Generic list-based browser with object details closed



Sentence caps

Label: Sentence caps

Longer label:

Sentence caps

☐ Label

☑ Longer label

Sentence caps

● Longer label

○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps A	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps B	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps C	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps D	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps E	Sentence...	Sentence caps	Sentence caps	Sentence caps

Show: ☐ Filter ☐ Longer filter ☑ Longer filter ☑ Filter ☐ Filter

Last refresh: 14:36:30 2001-04-28

☐ Auto-refresh

Sentence caps

Much longer label: Details Value

Label: Details Value

Longer Label: Longer Details Value

Much longer label: Details Value

Longer Label: Longer Details Value

Sentence caps

Label: Label:

Longer label: Sentence caps

Label:

Figure 7-8/Z.372 – Generic list-based browser with object details open

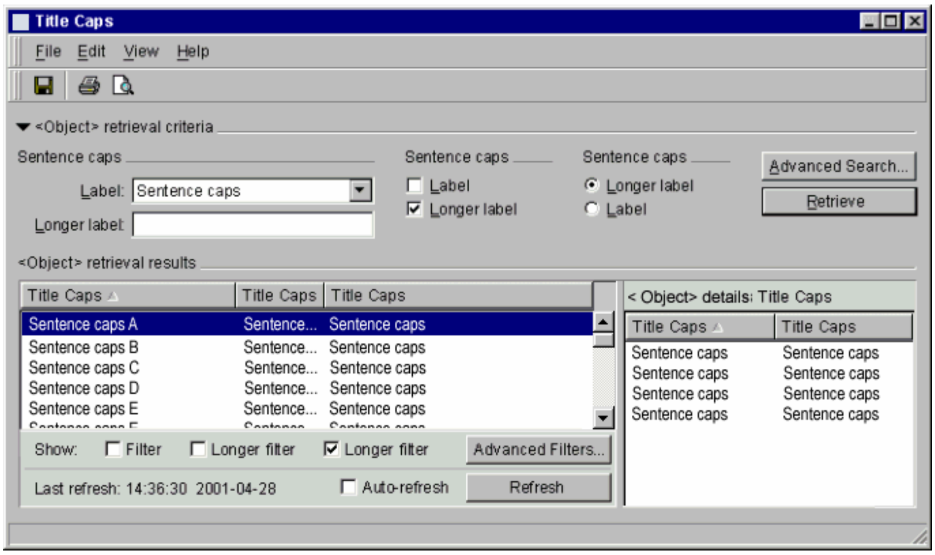


Figure 7-9/Z.372 – Generic list-based browser with object details table

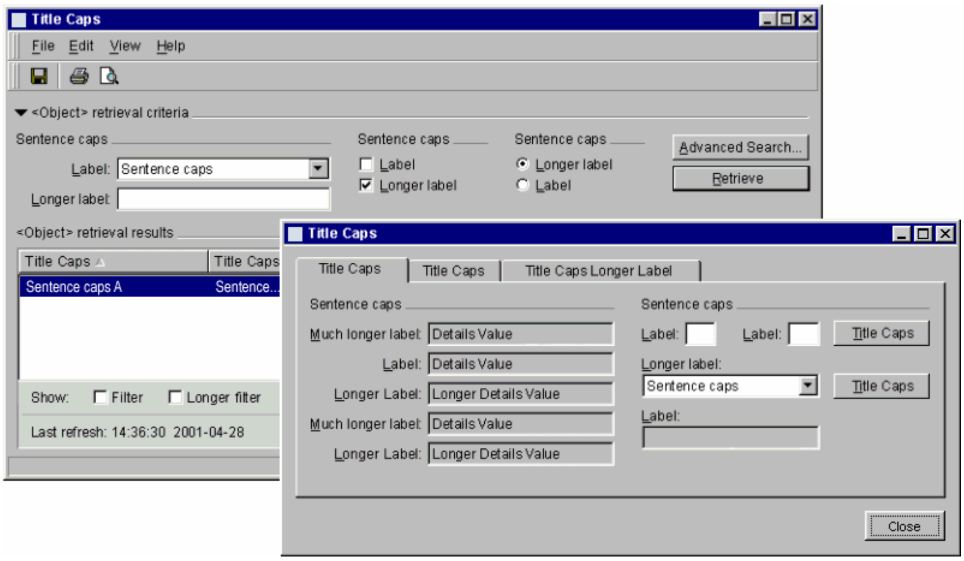
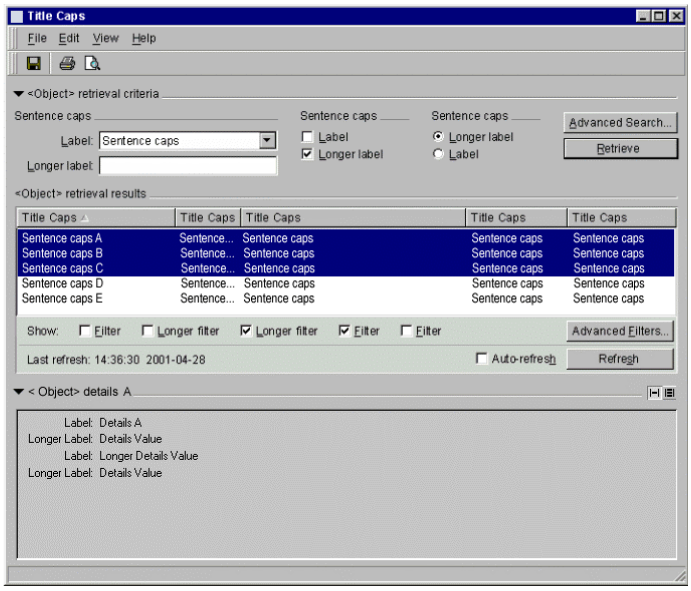


Figure 7-10/Z.372 – Generic list-based browser with object details properties

It is possible to show details from more than one selected object. The results are shown in Figures 7-11 and 7-12.



Sentence caps

Label: Sentence caps

Longer label:

Sentence caps: ☐ Label ☑ Longer label

Sentence caps: ● Longer label ○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps A	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps B	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps C	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps D	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps E	Sentence...	Sentence caps	Sentence caps	Sentence caps

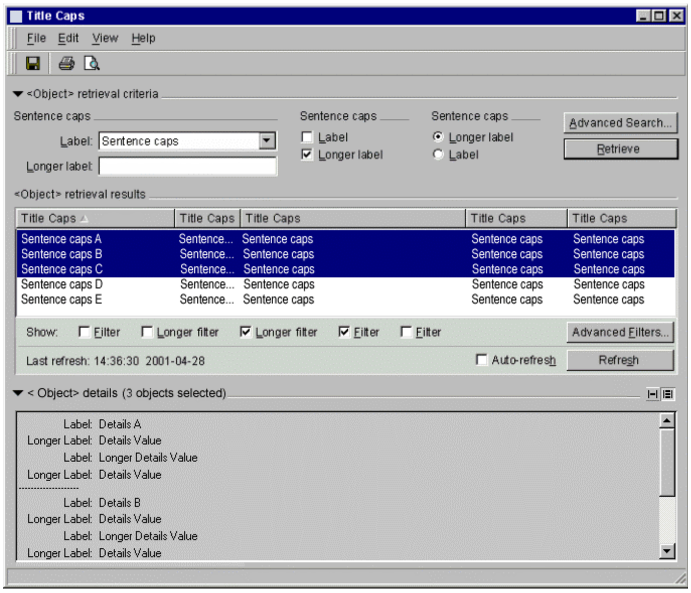
Show: ☐ Filter ☐ Longer filter ☑ Longer filter ☑ Filter ☐ Filter

Last refresh: 14:36:30 2001-04-28 ☐ Auto-refresh

Label: Details A
Longer Label: Details Value
Label: Longer Details Value
Longer Label: Details Value

Figure 7-11/Z.372 – Multiple objects selected, only one set of details shown

In this case, it is the details of the first object of the selected set, which are displayed.


Title Caps
File Edit View Help
<Object> retrieval criteria
Advanced Search...
Retrieve
<Object> retrieval results
Advanced Filters...
Refresh
< Object> details (3 objects selected)

Sentence caps

Label: Sentence caps

Longer label:

Sentence caps: ☐ Label ☑ Longer label

Sentence caps: ● Longer label ○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps A	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps B	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps C	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps D	Sentence...	Sentence caps	Sentence caps	Sentence caps
Sentence caps E	Sentence...	Sentence caps	Sentence caps	Sentence caps

Show: ☐ Filter ☐ Longer filter ☑ Longer filter ☑ Filter ☐ Filter

Last refresh: 14:36:30 2001-04-28 ☐ Auto-refresh

Label: Details A
Longer Label: Details Value
Label: Longer Details Value
Longer Label: Details Value

Label: Details B
Longer Label: Details Value
Label: Longer Details Value
Longer Label: Details Value

Figure 7-12/Z.372 – Multiple sets of details shown

The choice is up to the user. The selection is made through a view toggle icon as shown.

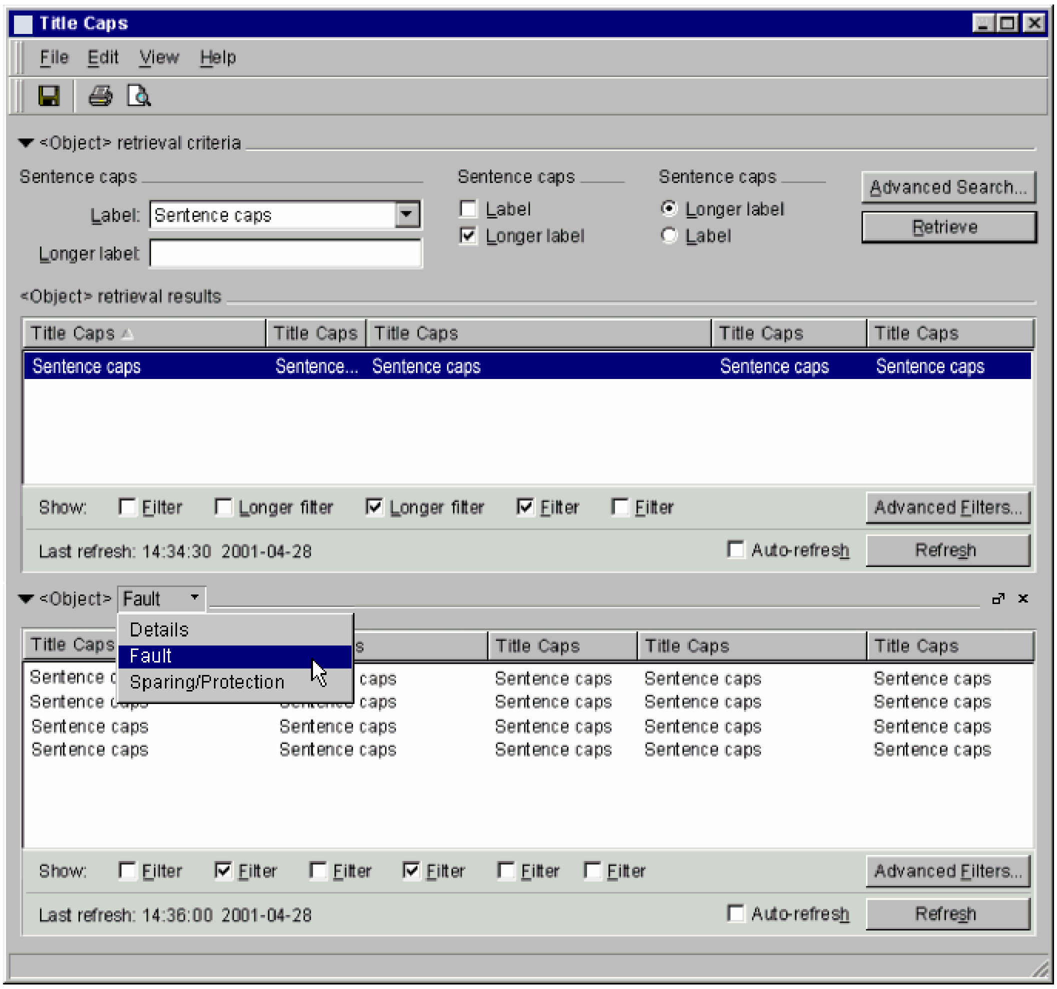
Another mechanism for the display of object attributes makes use of different ways of displaying the panes within the Object detail area. There are several ways in which the panes may be displayed.

Object detail panes (O)

The Object details area displays the 'general' Details pane by default. As well as 'general' Details, there may be more specialized types of panes such as Fault (Alarm), Lateral Navigation, Sparing/Protection mapping, and others. These specialized panes may be displayed one at a time or in combination: shown one at a time, only the single selected pane appears in the Object details area; shown in combination, the selected panes are all displayed within the Object details area.

Configurable panes (O)

Specialized detail panes can be selected from the View menu or a pull-down incorporated into any of the pane headers, and as many or few may be loaded at any given time. The user has the choice of replacing the current info pane or adding a new one to the details area. The type or category of data is selected via the dropdown menu as shown. This mechanism can be valuable when there may be too many data categories for tabs.



Sentence caps

Label: Sentence caps

Longer label:

Sentence caps

□ Label

☑ Longer label

Sentence caps

● Longer label

○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps	Sentence...	Sentence caps	Sentence caps	Sentence caps

Show: □ Filter □ Longer filter ☑ Longer filter ☑ Filter □ Filter

Last refresh: 14:34:30 2001-04-28 □ Auto-refresh

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps	Sentence caps	Sentence caps	Sentence caps	Sentence caps
Sentence caps	Sentence caps	Sentence caps	Sentence caps	Sentence caps
Sentence caps	Sentence caps	Sentence caps	Sentence caps	Sentence caps
Sentence caps	Sentence caps	Sentence caps	Sentence caps	Sentence caps

Show: □ Filter ☑ Filter □ Filter ☑ Filter □ Filter □ Filter

Last refresh: 14:36:00 2001-04-28 □ Auto-refresh

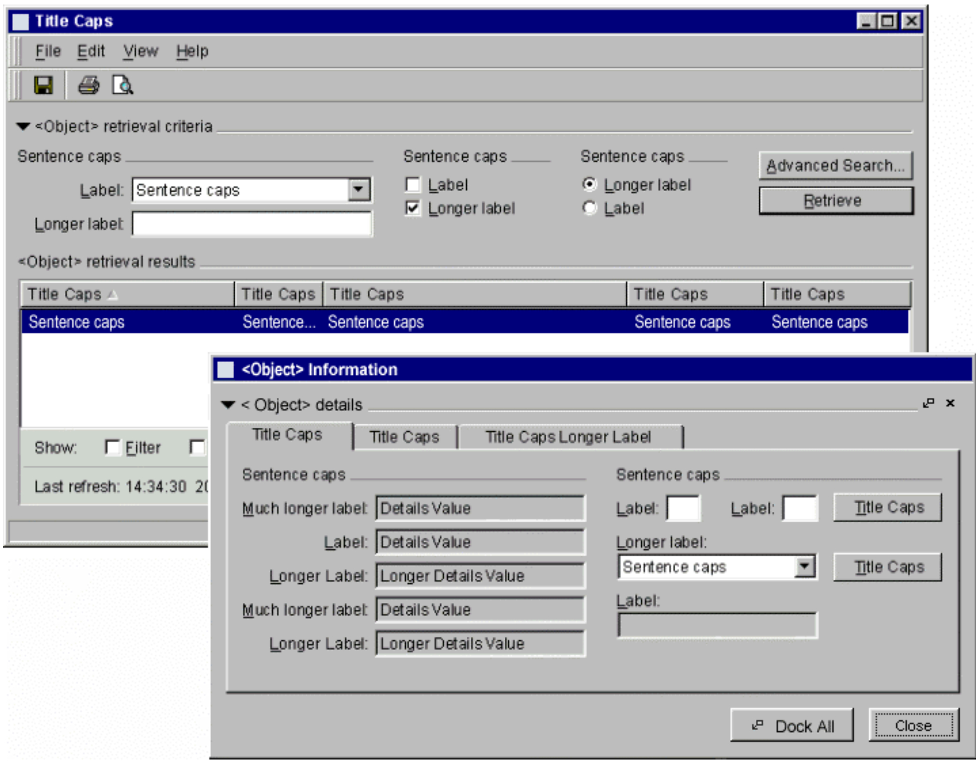
Figure 7-13/Z.372 – Configurable panes

Detachable panes (O)

(R) A further enhancement allows detaching panes from the main object browser window into a separate 'Information Palette'. Detached panes can subsequently be returned to the main window one at a time, or all at once through a single action (by closing the whole palette and returning its panes to the object details area of the object browser window). Panes may also be 'dismissed' by clicking their 'close' buttons or deselecting them from the View menu.

(R) The information palette is floated in front of its parent window. There should only be one information palette per object browser instance; bringing an object browser into focus also pulls its associated information palette to the foreground.

(R) Only one information pane of each available type can be loaded at any one time, and the content of the pane always remains tightly coupled to the current selection context in its associated object browser window. The user is given the option of detaching or 'floating' detail panes and placing them in a separate information palette in the interest of freeing up as much space as possible when necessary within the object browser.



Sentence caps

Label: Sentence caps

Longer label:

Sentence caps

☐ Label

☑ Longer label

Sentence caps

● Longer label

○ Label

Title Caps	Title Caps	Title Caps	Title Caps	Title Caps
Sentence caps	Sentence...	Sentence caps	Sentence caps	Sentence caps

☐ Filter ☐

Last refresh: 14:34:30 2(

Sentence caps

Much longer label: Details Value

Label: Details Value

Longer Label: Longer Details Value

Much longer label: Details Value

Longer Label: Longer Details Value

Sentence caps

Label: Label:

Longer label: Sentence caps

Label:

Figure 7-14/Z.372 – Detachable panes

7.1.3.8 Status bar (O)

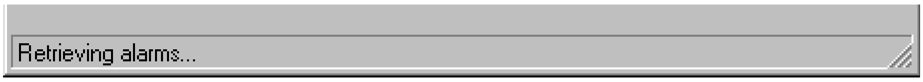
The status bar is located at the bottom of the window. It can be subdivided into different sections that provide "at a glance" status information on the product, current applications, etc. The status bar is optional because other mechanisms may be used, when more suitable, such as an hour glass.

One of the most significant factors in promoting ease of use and satisfaction with an application is providing timely and accurate status information on the activity of the system. There are few things more frustrating to a user than to initiate a system operation, and then be left wondering whether the command was accepted, how the operation is progressing, how long it may last, when it has finished, and whether or not it was successful.

(R) Status information must include the following:

- An acknowledgement that the operation was accepted and initiated;
- How long the operation is expected to take in the form of hourglasses for short operations, or progress bars for longer operations;
- Any new status information during the operation; and
- An indication that the operation has concluded.

(R) Status information should be placed in the status area of the window from which the operation was initiated. See Figure 7-15.

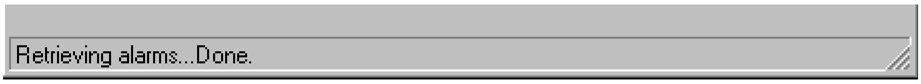


Retrieving alarms...

Figure 7-15/Z.372 – Status message in status bar

(R) If user interaction with the system should become blocked during the operation (e.g., when requests are being sent to the server, a response from the server is required before the operation can proceed, or where the client is processing data), the cursor should be changed to the standard platform "Busy cursor"; typically this is an hourglass. When the operation has been completed (either successful or failed), return the cursor to the normal state.

(R) If the operation is successful, the word "Done" should be appended to the message in the Status area; for example, "Retrieving <object>…Done". See Figure 7-16.



Retrieving alarms...Done.

Figure 7-16/Z.372 – Successful operation message

7.1.4 Object browser variations using text retrieval criteria

The object-based browser has many options varying its appearance and the information it provides, for example:

- the retrieval criteria area, the filters area, and the details area are optional;
- from the View menu, users can show or hide the retrieval, list filter, and details areas;
- the user can open or close the "bird beak" widget in the retrieval criteria and object details areas.

Following are a number of base browser examples.

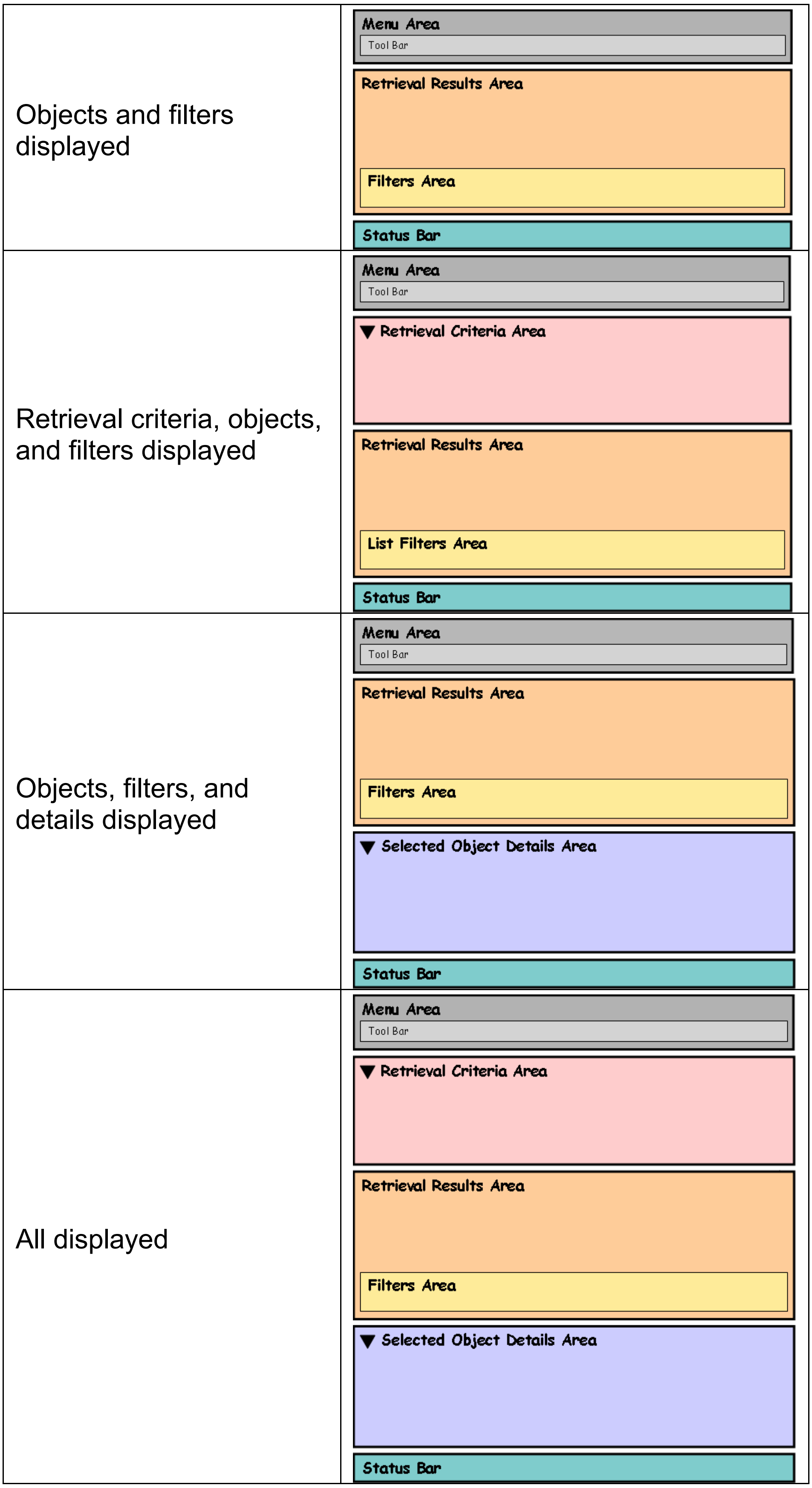


Figure 7-17/Z.372 – Browser variations

7.1.5 Examples of browsers based on different view types

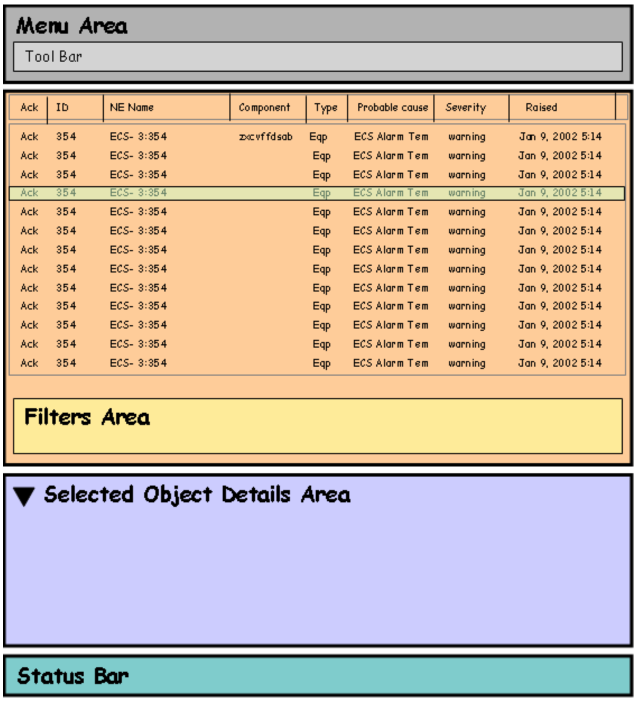
Typically, the results area is a textual view, but the object browser also supports other view types, for example, a process control view or a graphical view (see TMF-046/ITU-T Rec. Z.371).

For network elements, graphical views (for example, shelf-level graphics) are often preferred. Other views may provide additional or other forms of information and should be evaluated based upon specific tasks.

Different views of the retrieval results area may be available within a single application. These can be selected via the view menu or view toggle buttons in the results area.

Filters, when used in lists, typically hide instances associated with the filter. In graphic views, the filters typically remove overlays of information on the objects associated with the view.

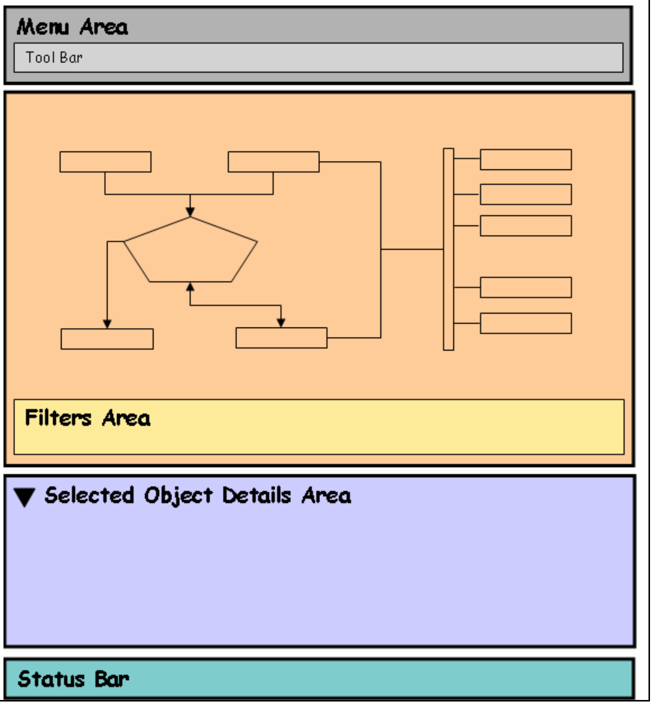
Textual view



Ack	ID	NE Name	Component	Type	Probable cause	Severity	Raised
Ack	354	ECS- 3:354	zxcvffdsab	Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14
Ack	354	ECS- 3:354		Eqp	ECS Alarm Tem	warning	Jan 9, 2002 5:14

Process control view

In this case the filter for example might gray out a specific class of activities, or it may expand or collapse certain regions of the process.



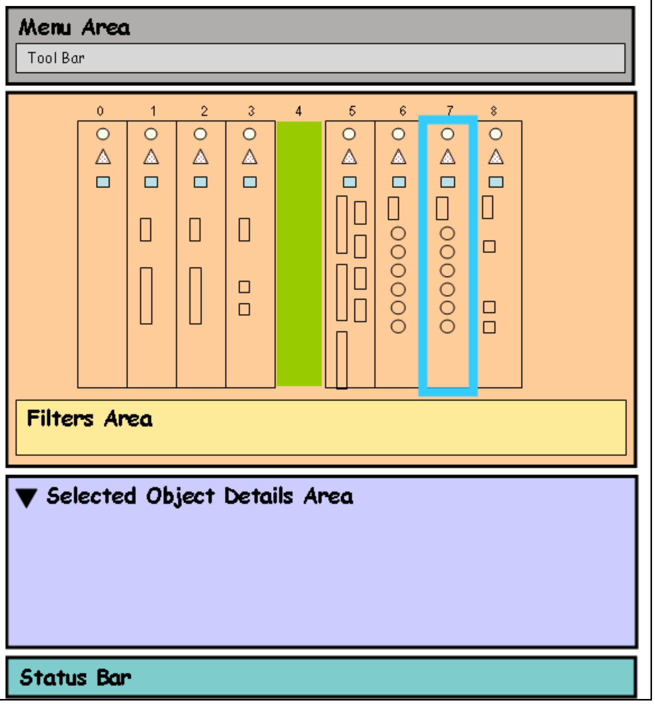
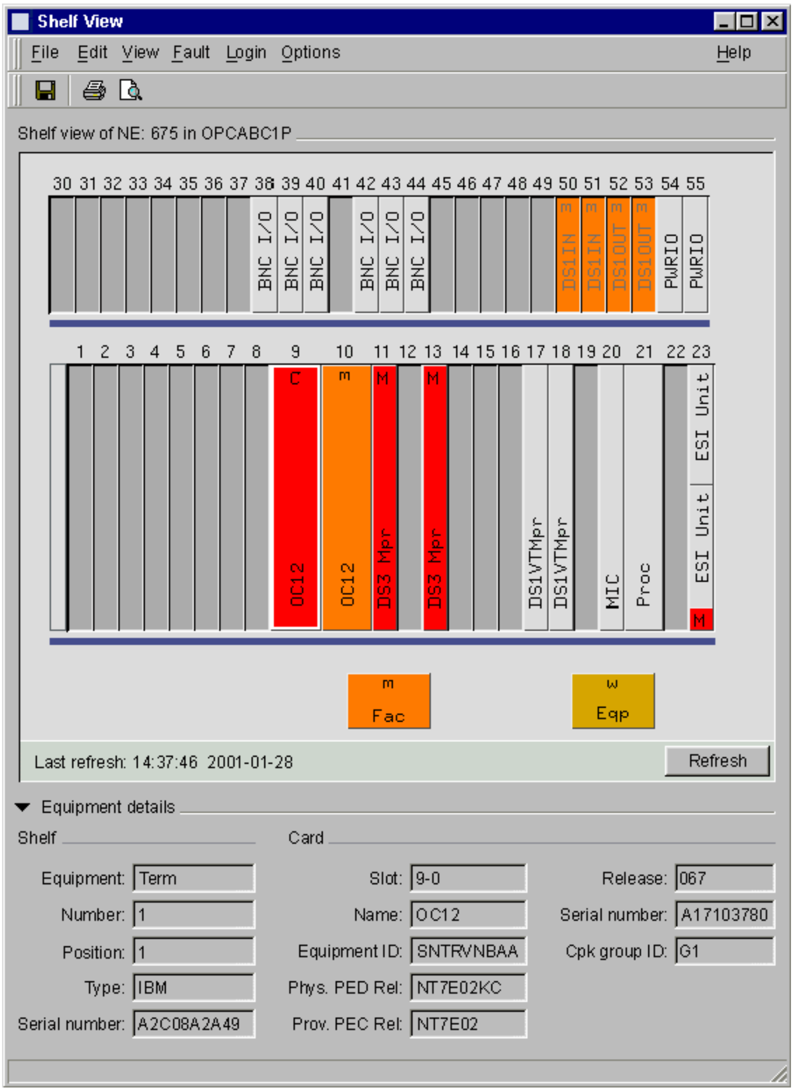
Graphic view	

Figure 7-18/Z.372 – Browser list views



Shelf view of NE: 675 in OPCABC1P

Last refresh: 14:37:46 2001-01-28

Shelf

Equipment: Term

Number: 1

Position: 1

Type: IBM

Serial number: A2C08A2A49

Card

Slot: 9-0

Name: OC12

Equipment ID: SNTRVNBAA

Phys. PED Rel: NT7E02KC

Prov. PEC Rel: NT7E02

Release: 067

Serial number: A17103780

Cpk group ID: G1

Figure 7-19/Z.372 – Shelf-view example

7.2 Object browsers using trees for retrieval

7.2.1 When to use

Use an object browser with a tree when showing a true hierarchy of a set of instances (for example, bay, shelf, card, port, interface).

The object browser using a tree builds on the text-based browser format and adds navigational capabilities of the tree.

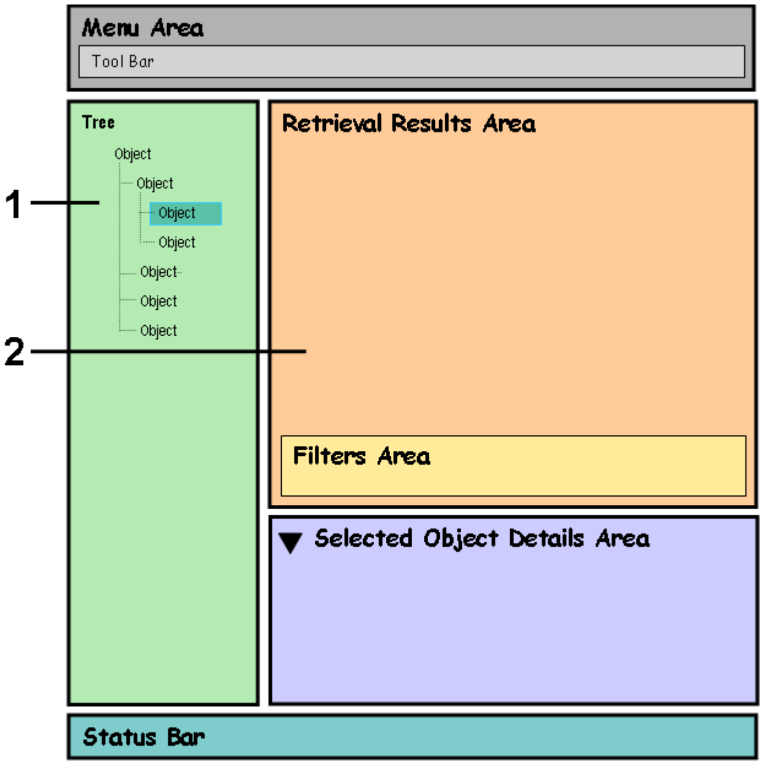
If the data set is large, either broad with many instances or deep with a complex hierarchy, trees should not show the end leaves.

7.2.2 Browser templates using trees

(R) Follow the Windows Internet Explorer model for trees, that is, a selection in the tree changes the details on the right-hand side. The information shown on the right gets context from the tree. Trees may not always show the end leaves. If they do not show the end leaves, a selection in the tree will populate the retrieval results area. If the end leaves are shown, then a selection in the tree will fill the details area and no retrieval results area is shown.

7.2.2.1 Object class browser

In a class tree browser, a set of classes is displayed in the tree. Instances and details areas are displayed on the right.

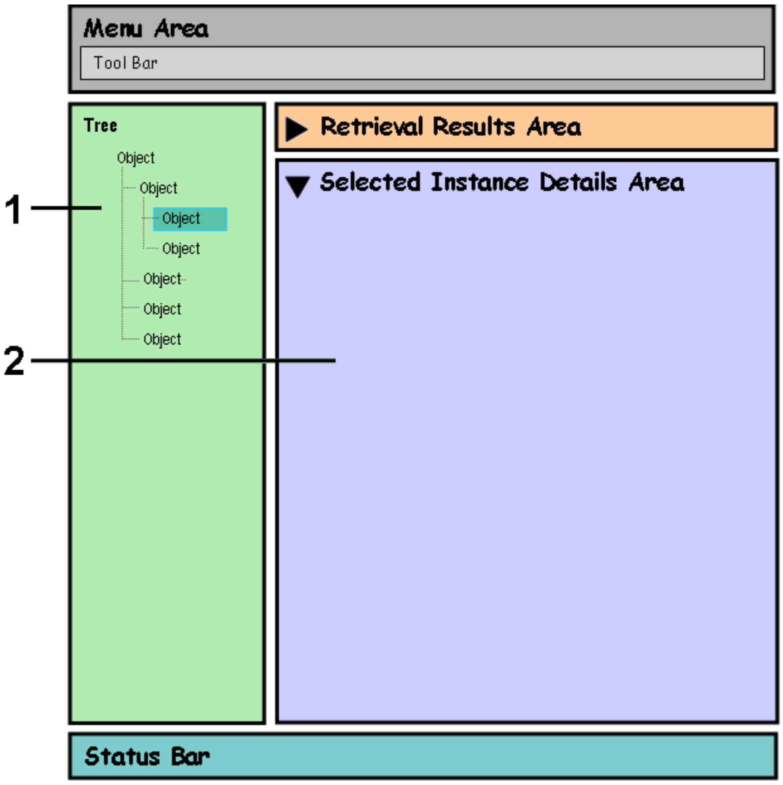


1 Class tree
2 Instance results and details

Figure 7-20/Z.372 – Class tree browser format

7.2.2.2 Leaf tree browser

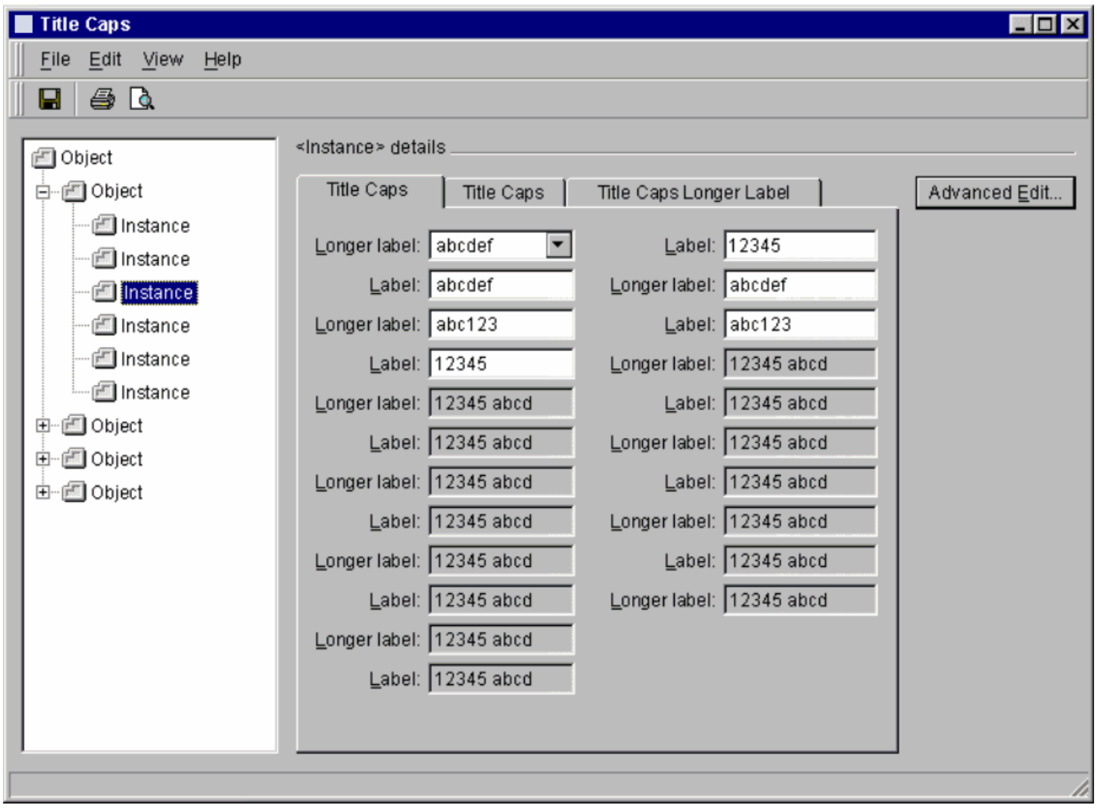
In a leaf tree browser, a set of end leaves is displayed in the tree. The details area is displayed on the right.



1 Instance tree

2 Details

Figure 7-21/Z.372 – Leaf tree browser



<Instance> details

Longer label: abcdef

Label: abcdef

Longer label: abc123

Label: 12345

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Label: 12345 abcd

Label: 12345

Longer label: abcdef

Label: abc123

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Label: 12345 abcd

Longer label: 12345 abcd

Figure 7-22/Z.372 – Instance tree with details area example

7.2.3 HMI components

7.2.3.1 Tree

(R) The tree allows the user to navigate a hierarchy of objects in a way analogous to the Windows Explorer model, expanding or collapsing branches and scrolling if necessary to the object of interest.

(O) Additional information presented in a separate column relating to objects in the tree may be included (to the left of the tree), with items updated as the tree is expanded and collapsed. A typical example of this is the display of summarized Alarm Counts beside items in a tree representing nodes in different areas of a network, as the groups are collapsed (hiding the view of the contained nodes), the summarized alarm count for the entire group is shown.

(R) Single selection of an object in the tree updates the context of the results and details areas. Selection of an object in the results area resets the view context and changes the selection in the tree to reflect the object currently selected in the view. There is a one-to-one relationship between the object selected in the tree and the object selected in the results area.

7.2.3.2 Other HMI components

With the exception of the tree itself, the HMI components are the same as in the text-based browser pattern.

7.2.4 Tree-based object browser variations

The variations in the tree-based browser, as far as the other HMI components are concerned, are the same as the list-based browser.

8 Template examples based on the object browser pattern

8.1 Network Component Manager (NCM) template

This clause provides a template for the development of a network component manager. This template is an integral part of this Recommendation and thus follows the convention of (R) Requirements, and (O) Optionality.

A network component is a logical resource in the network that provides a specific function. Components can be implemented either in hardware or software. A component can be a single card or shelf, or it may be a collection of cards or shelves co-located or distributed geographically. A component can also be a software function that runs in a standard computing engine. Network components are pre-engineered by the network equipment provider. For administrative purposes and efficiency, components generally need to be graphically displayed on workstations for purposes of maintenance. Note that this definition applies to this Recommendation and other HMI-related Recommendations.

8.1.1 Characteristics

The scope of component managers can include the management of entire network components, not just one contained component. In addition, NCMs need to be flexible enough to support a full range of network components, from very large multi-frame/bay configurations to simple ones.

8.1.1.1 Users

The primary key users are 1st, 2nd and 3rd level fault support staff; Secondary key users are craft in the field and nodal provisioning personnel; potential Tertiary users include Planning and Engineering (assessing current inventory), Installation staff (post-commissioning adds) and Security Administration. There is no direct mapping of tasks to users; primary users will have a need to perform primary, secondary and tertiary tasks at one time or the other. However, the bulk of their use of the NCM will probably be for primary tasks.

8.1.1.2 Tasks

The primary tasks fall in fault troubleshooting, response and verification. The NCM should support easy navigation from external systems/applications and internally through simple and complex network components. It should facilitate detailed locate and diagnose activities, with summary status info at each level to enable an effective drill down. It should also accurately represent logical and physical relationships and provide the basis for diagnosis and testing (in place or through the launch of other applications). Once a diagnosis has been determined, the NCM should support the operator response, by supporting NOC/craft communication, allowing the editing of key provisioning parameters and supporting component state/status change and traffic control. For effective troubleshooting and response, the operator needs to verify and monitor the condition of the components that he/she is working on: the NCM should provide summary status info at each component level and provide some level of auto update (to show when a problem is cleared).

Secondary tasks include component configuration and access to key tools in context. For example, a view of current inventory supports communication with craft in the field during component adds. Such a task may include post commissioning hardware installation. Providing a provisioning tool in place, such as a property sheet in the details area, simplifies the task. When it is not feasible or appropriate to provide specialized tools in place, the NCM should provide access to key tools in context such as a fault log tool, diagnostics and testing tools, inventory, component provisioning, circuit provisioning and real-time performance stats. These tools would be accessible through object menu launch of the appropriate tool in the context of a component selected in the NCM.

The NCM supports tertiary tasks through the launch of tools that support tasks that may not be NCM specific, such as performing S/W inventory downloads, displaying historical performance data and setting access control. These are links to other applications that are not necessarily launched in context of selections in the NCM.

8.1.2 Mapping network CM task flow to the object browser pattern

The network component manager template is derived from the standard object browser pattern. The template supports the typical shelf level troubleshooting task pattern: from location and selection of a component to viewing the selected component and associated details.

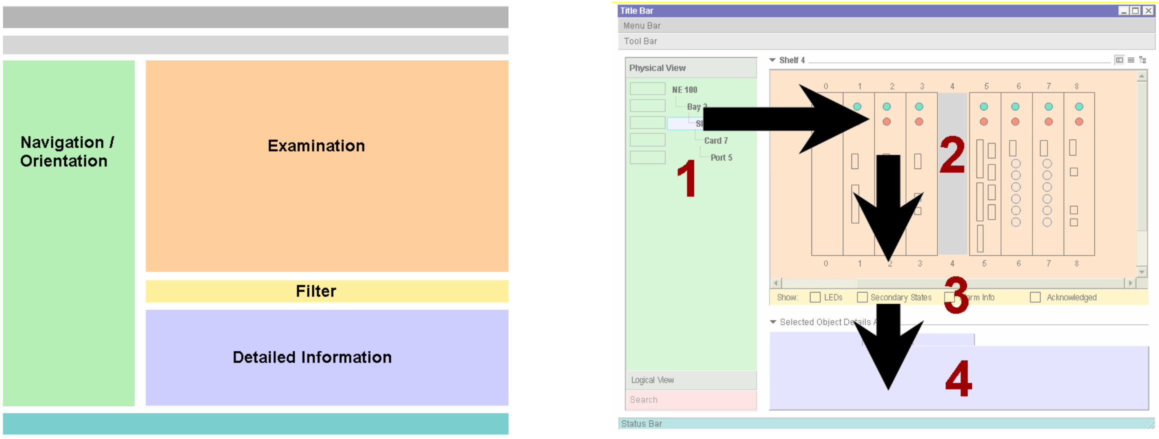


Figure 8-1/Z.372 – Generic NCM template

Area 1: This area allows the user to specify a component to examine through navigation in a tree.

Area 2: The specified component is displayed in the examination area.

Area 3: The use of filters further determines what is displayed in the examination area.

Area 4: Detailed information for a selected object is displayed in this area.

Functionality of the NCM can be divided between three core activities: Orientation/Navigation, Examination, and Detailed Information. In the course of moving through the archetypical NCM task of troubleshooting, the user alternatively focuses on orienting themselves/locating a troubled component, examining the component in context and retrieving additional details about the component to arrive at a diagnosis and a course of action.

The **Orientation/Navigation Area** provides basic browsing tools such as physical and logical trees, as well as specialized tools such as context markers (bookmarks) and a network component level find mechanism. Selection of a component from any of these areas will set the context of the other two areas of the NCM. In addition, in the course of a 'drill-down' in the examination area, users can refer back to a tree to help orient them to the larger context.

(R) The orientation/navigation is required if there is appreciable hierarchy to represent (e.g., 3 or more).

(O) The orientation/navigation is optional if there is very little or no hierarchy to represent.

(R) **The Examination Area** provides graphical and list views of particular levels within the hierarchy of the NCM such as Bay, Shelf, Card, Port, sub-component. Depending on the situation, several levels of the containment hierarchy could be viewed. The graphical view represents components in an accurate manner, with attention to size, orientation, location and key functional features; it does not detail the appearance of the component in a photo-realistic way. By simplifying the graphical representation, it can be used in conjunction with other graphical devices such as alarm balloons and state indicators with a minimum of visual interference.

The Detailed Information Area offers additional detail for components selected in the examination area: details in general, or fault-specific details. Additionally, technology-specific functionality can be added: protection/sparing, connectivity: lateral (port-to-port) navigation, break out panels and logical/physical navigation.

(R) This area is required if all of the required attributes of the object cannot be shown in the retrieval result area.

(O) This area is optional if all required attributes can be effectively displayed in the retrieval results.

The three areas support each other and aid the user to set up and maintain context for the duration of the NCM session.

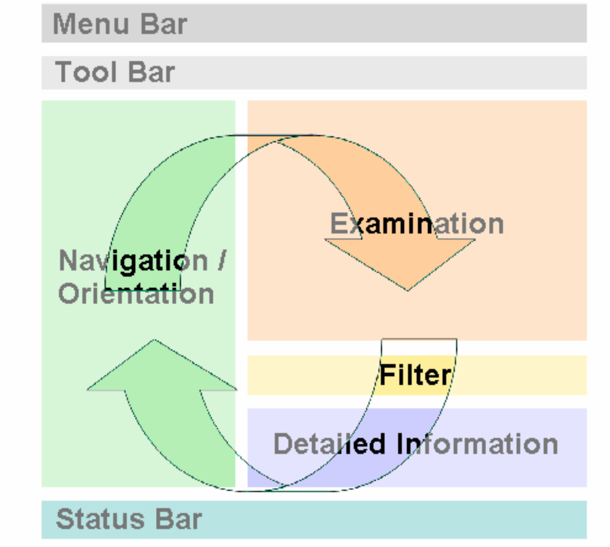


Figure 8-2/Z.372 – NCM area interdependence

The Menu Bar (R), Tool Bar (O) and Status Bar (O) provide standard functionality in support of the principal task areas.

8.1.3 (O) Collapsing areas of the NCM

As the NCM template draws on the flexibility of the list browser pattern, the display of several different combinations of areas is possible. A user may wish to collapse or expand specific areas depending on the task at hand: this is accomplished through selection in a View Menu or direct expand/collapse controls associated with each pane.

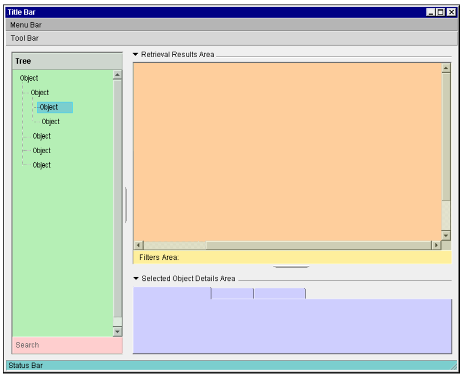


Figure 8-3/Z.372 – NCM default layout

(O) Once a particular component has been localized, the user may want to collapse the navigation area:

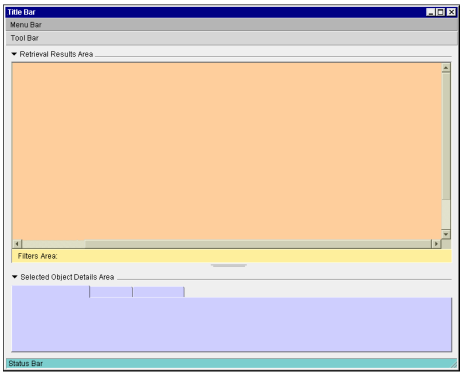


Figure 8-4/Z.372 – Navigation area removed

(O) Focus can be brought to the examination area:

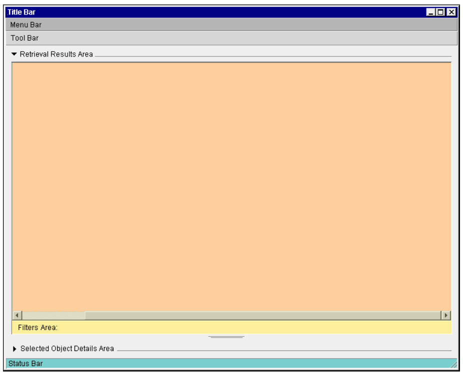


Figure 8-5/Z.372 – Details area collapsed

(O) Or the user can collapse the examination area to concentrate on details:

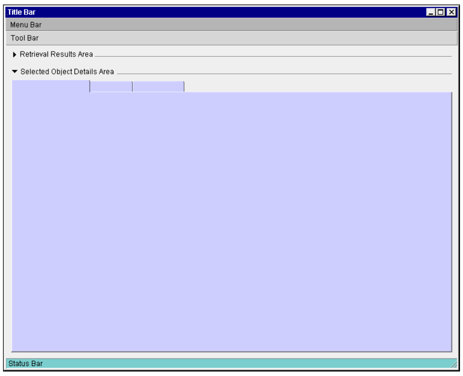


Figure 8-6/Z.372 – Details area expanded

8.1.4 Logical and physical views

(O) The component manager supports the ability to adjust an operator's view as he/she explores a particular network component. These views control what the operator sees and how the data is presented. For example, certain components are best represented as physical entities such as shelves and cards; while others are more easily represented as logical entities such as logical processors and processes.

(R) The logical/physical nature of the components is reflected in the navigation and in the examination areas. A simple means to toggle views is to select the physical tree or the logical tree when both are available. Because the tree selection affects the other areas of the NCM, only one or the other may be displayed at any one time, resulting in a logical or a physical view of the components.

8.1.5 Navigation

8.1.5.1 (R) Selection in a tree

Navigation in a tree follows standard tree conventions for expanding and collapsing branches of the tree and selection within the tree. Selection in the tree sets the context of the NCM, determining the content of the examination area.

8.1.5.2 (R) Telescoping views

An alternative to navigating through component layers through their representation in the Navigation tree is to drill down through selection in the examination area. A user can move through the containment hierarchy by selecting components displayed within successive examination representations. In this way, by selecting a component within a component, the context of the NCM can be reset to reflect its selection in the tree, its representation in the examination area, and display of its details in the details area.

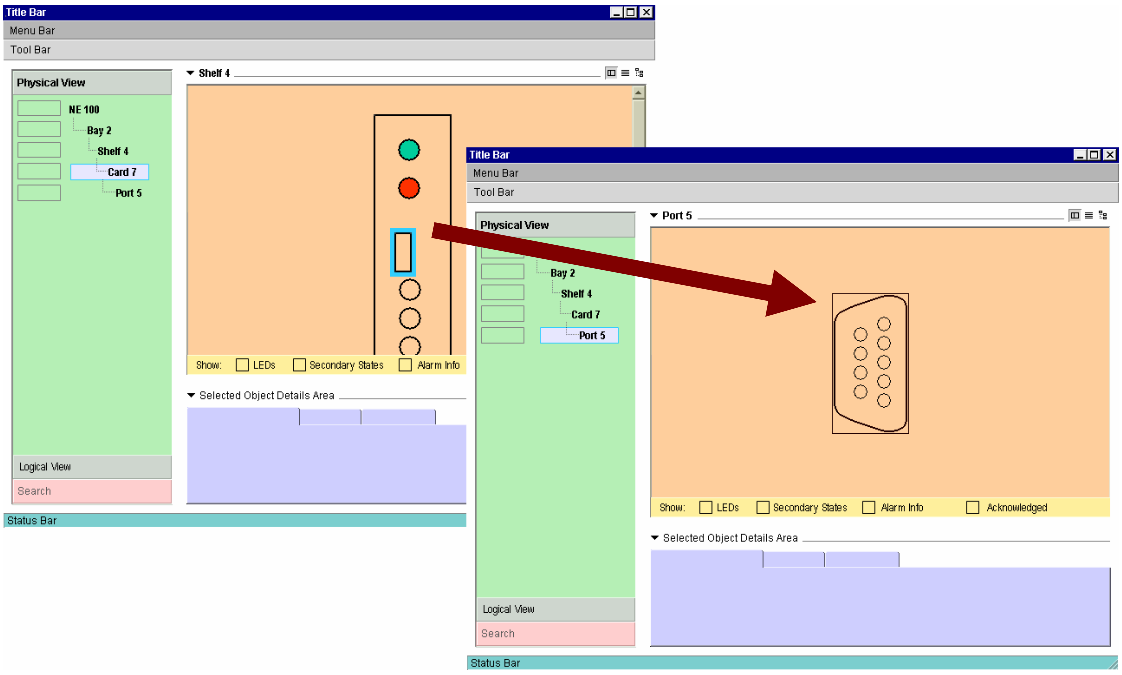


Figure 8-7/Z.372 – Telescoping views

8.1.5.3 (O) Base view context tracking

By using an historical pull-down selector that stores examination area contexts, the user can move backwards and forwards to recently viewed components.

8.1.5.4 (O) Context markers

Context markers are a convenient way to 'bookmark' a context that the user has navigated to so that he may easily return to the same view after navigating away in the course of an investigation. These markers can be saved within and across NCM sessions.

8.1.5.5 (O) Component level find

This is a secondary navigation tool that allows the user to find a particular component or component type that may be distributed throughout the network component. It works like a typical find tool, allowing the user to specify a distinguishing characteristic, then the entire context of the NCM can be set to reflect the located component. The tree, examination and details are updated to the new context.

8.1.6 (R) Data representation in the examination area

(R) Representation of the object of interest and its most critical attributes in the examination area is required.

(O) Once data is loaded into the examination area, the operator can select how it should be represented. Options can include a graphical presentation, a textual (or table) presentation or a relational (tree) presentation.

Only those views which are appropriate for a particular network component need be included in the NCM.

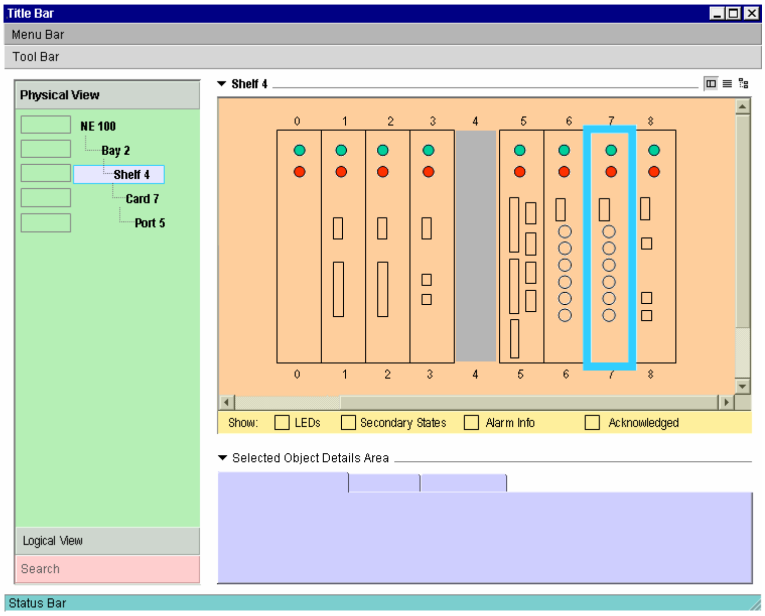
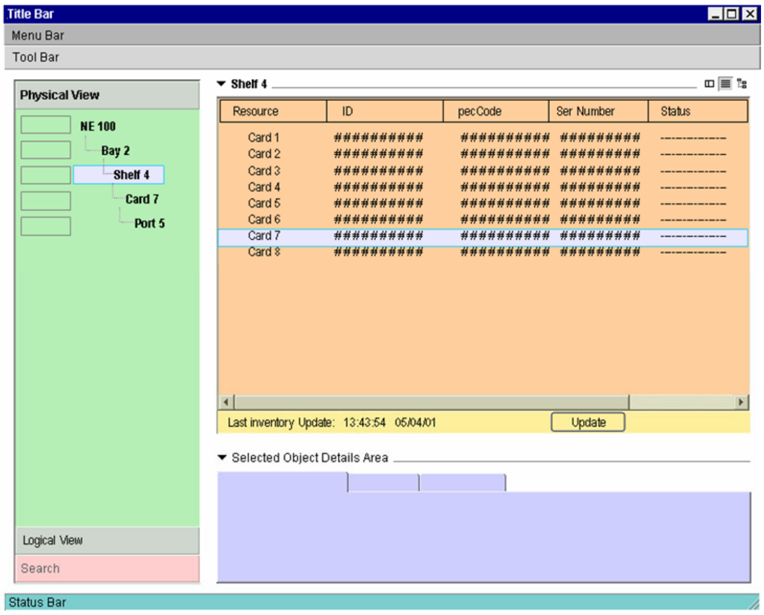


Figure 8-8/Z.372 – Physical view – Graphical representation



Resource	ID	pecCode	Ser Number	Status
Card 1	##########	##########	#########	-----------
Card 2	##########	##########	#########	-----------
Card 3	##########	##########	#########	-----------
Card 4	##########	##########	#########	-----------
Card 5	##########	##########	#########	-----------
Card 6	##########	##########	#########	-----------
Card 7	##########	##########	#########	-----------
Card 8	##########	##########	#########	-----------

Last inventory Update: 13:43:54 05/04/01

Figure 8-9/Z.372 – Physical view – List (inventory) representation

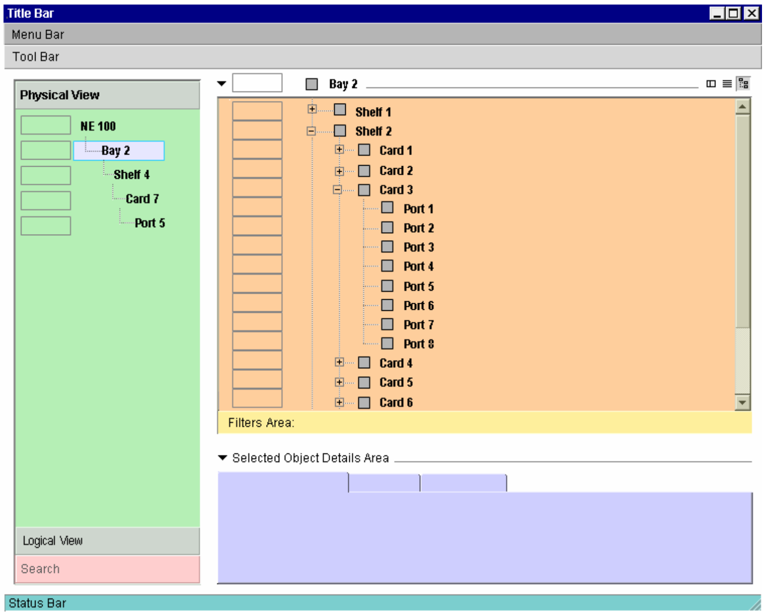


Figure 8-10/Z.372 – Physical view – Relational (tree) representation

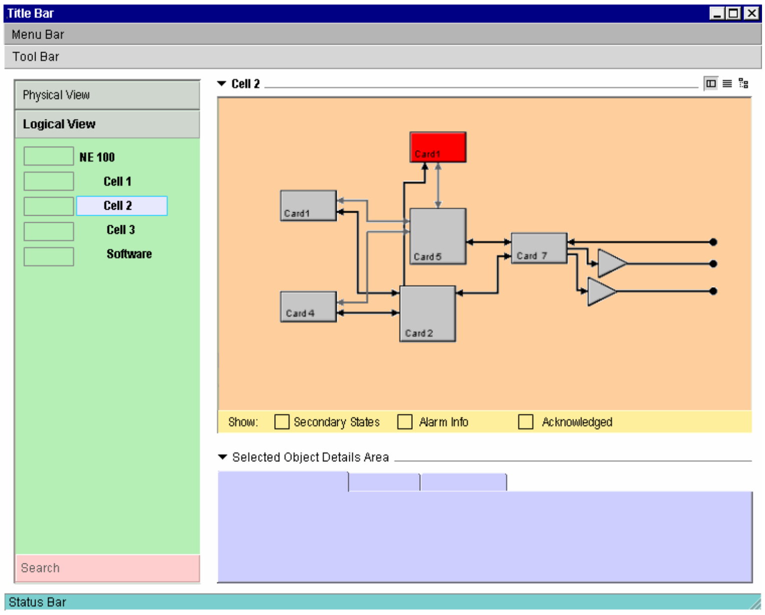
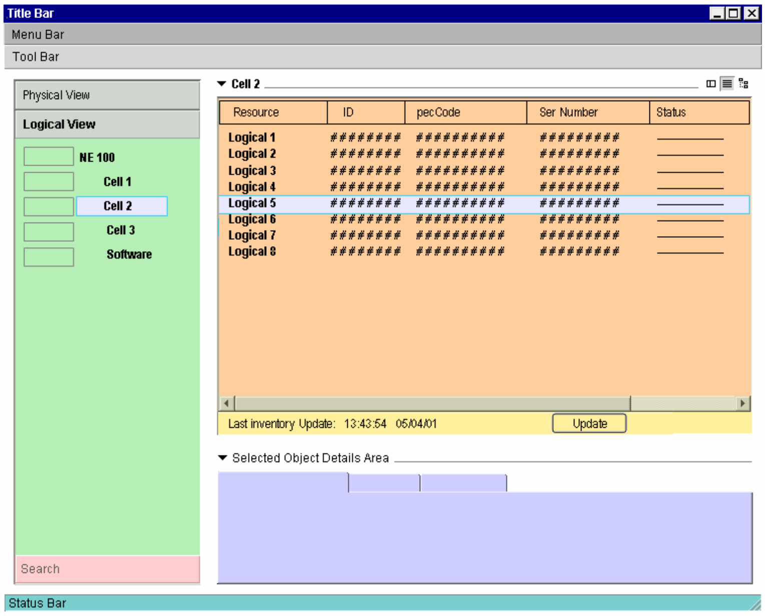


Figure 8-11/Z.372 – Logical view – Graphical representation



Resource	ID	pecCode	Ser Number	Status
Logical 1	########	##########	#########	———
Logical 2	########	##########	#########	———
Logical 3	########	##########	#########	———
Logical 4	########	##########	#########	———
Logical 5	########	##########	#########	———
Logical 6	########	##########	#########	———
Logical 7	########	##########	#########	———
Logical 8	########	##########	#########	———

Last inventory Update: 13:43:54 05/04/01

Figure 8-12/Z.372 – Logical view – List (inventory) representation

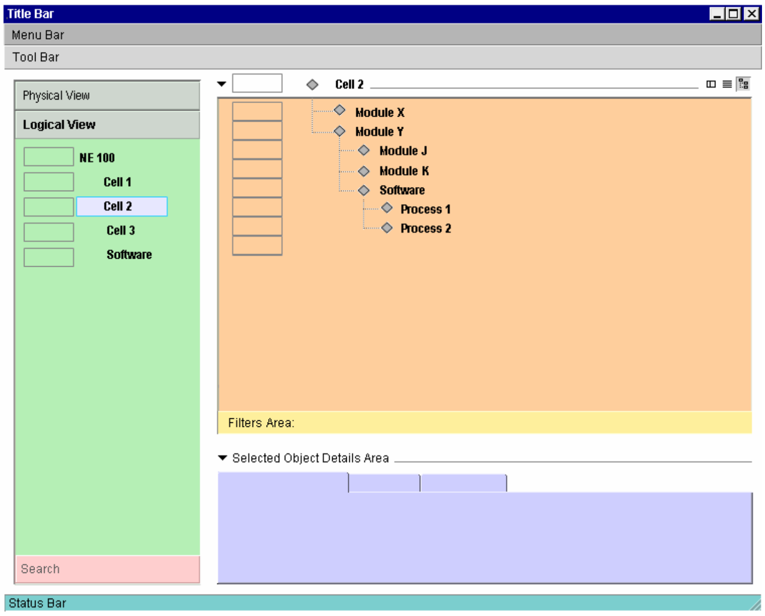


Figure 8-13/Z.372 – Logical view – Relational (tree) representation

8.1.7 Details

(R) The details area is required if all of the required attributes of the object cannot be shown in the retrieval result area.

(O) The details area is optional if all required attributes can be effectively displayed in the retrieval results.

(R) When a component is selected in the examination area, additional information specific to that component shall be displayed in the details. If nothing is selected, then the details relate to the base object represented in the examination area, and highlighted in the navigation area tree.

(R) Details apply to objects selected in a graphical, list or relational representation.

(O) As well as general details and fault summary type details, the details area can also contain technology-specific functionality: e.g., the display of protection/sparing details; connectivity: lateral (port-to-port) navigation from a selected component; break out panels and logical/physical navigation.

8.1.7.1 (O) User configurable detail display

Several types of detail panes may be supported by adhering to two main strategies: the first is providing a 'Detail type' pane selection mechanism in the details area; the second is to support limited 'tear-off functionality'.

(O) Detail pane selection

Detail pane selection provides the user with the means to 'load' and expand only those detail panes of interest.

(R) The panes can be arranged as tabbed panes, or individually viewable panes selected with a pull-down selector.

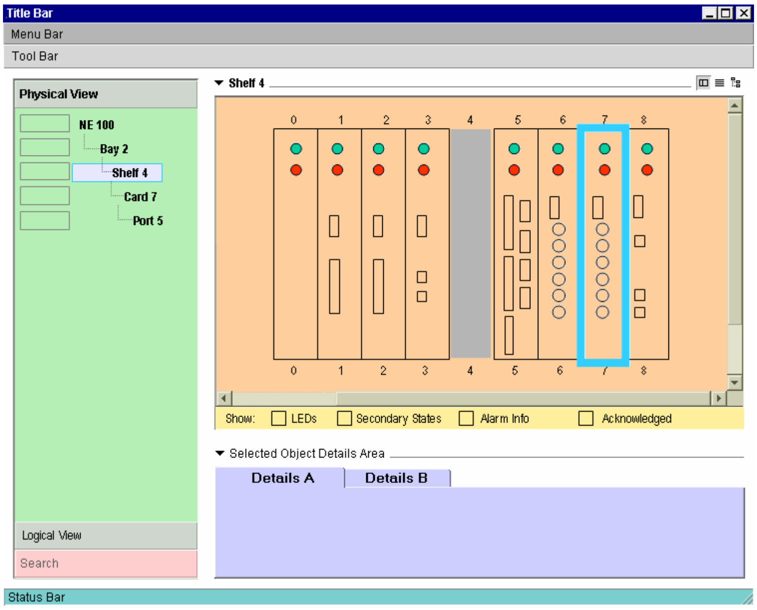


Figure 8-14/Z.372 – Tabbed details

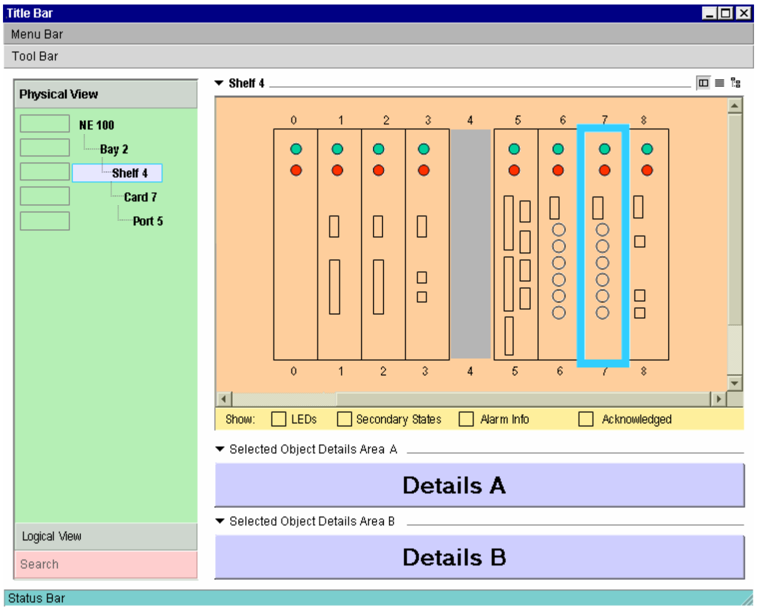


Figure 8-15/Z.372 – Multi-pane details

8.1.7.2 (O) Floating details window

The user also has the option to display the panes individually, or to tear off a pane for display in a separate details floating window.

(R) Each of the panes that have been 'torn off' appears in the floating window.

(R) There is only one floating details window per NCM instance. In this way, propagation of windows is kept to a minimum while allowing the user to make full use of the examination and details areas. Details that appear in the floating window continue to be determined by selection in the examination area. Floating panes can be returned to the main NCM details area at any time, collapsed or closed altogether.

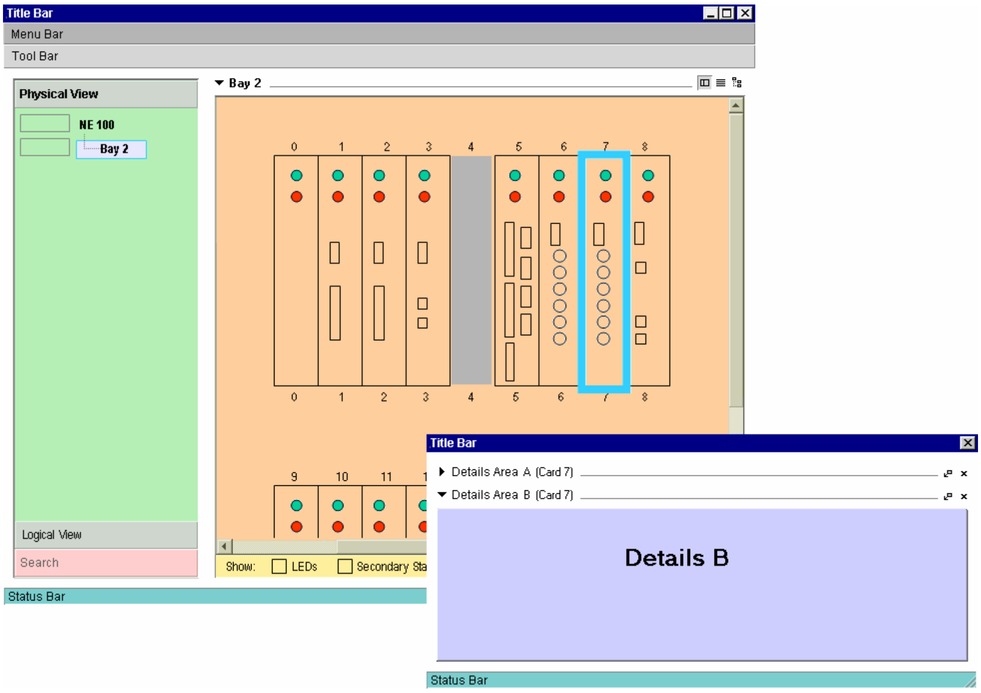
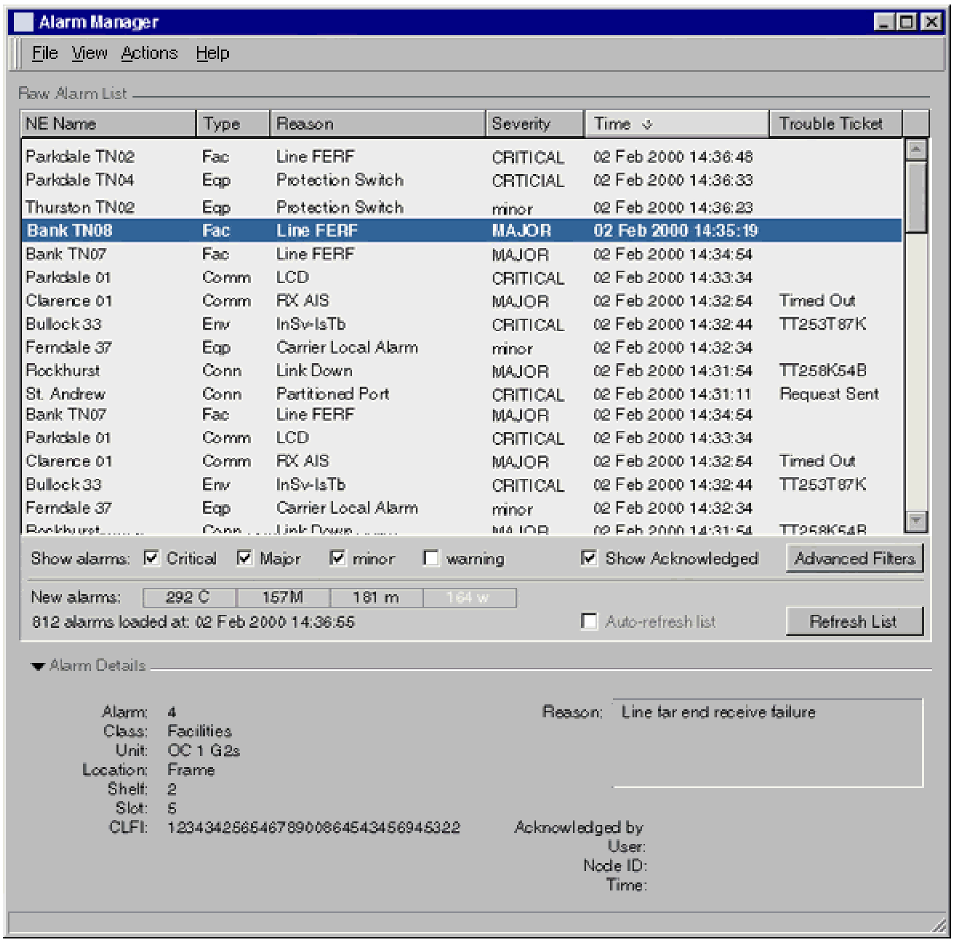


Figure 8-16/Z.372 – Floating details window

8.2 Other template examples

This clause provides a number of other template examples that show how the pattern may be used. These examples are not binding and are presented only to indicate the value of the pattern as it applies across different task areas.

First is a list of alarms, warnings, and other messages, that can be sorted, filtered, and updated. It alerts maintenance users to problems, and provides management with a view of outstanding problems.



NE Name	Type	Reason	Severity	Time	Trouble Ticket
Parkdale TN02	Fac	Line FERF	CRITICAL	02 Feb 2000 14:36:48	
Parkdale TN04	Eqp	Protection Switch	CRITICAL	02 Feb 2000 14:36:33	
Thurston TN02	Eqp	Protection Switch	minor	02 Feb 2000 14:36:23	
Bank TN08	Fac	Line FERF	MAJOR	02 Feb 2000 14:35:19	
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	
Parkdale 01	Comm	LCD	CRITICAL	02 Feb 2000 14:33:34	
Clarence 01	Comm	RX AIS	MAJOR	02 Feb 2000 14:32:54	Timed Out
Bullock 33	Env	InSv-IsTb	CRITICAL	02 Feb 2000 14:32:44	TT253T87K
Ferndale 37	Eqp	Carrier Local Alarm	minor	02 Feb 2000 14:32:34	
Rockhurst	Conn	Link Down	MAJOR	02 Feb 2000 14:31:54	TT258K54B
St. Andrew	Conn	Partitioned Port	CRITICAL	02 Feb 2000 14:31:11	Request Sent
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	
Parkdale 01	Comm	LCD	CRITICAL	02 Feb 2000 14:33:34	
Clarence 01	Comm	RX AIS	MAJOR	02 Feb 2000 14:32:54	Timed Out
Bullock 33	Env	InSv-IsTb	CRITICAL	02 Feb 2000 14:32:44	TT253T87K
Ferndale 37	Eqp	Carrier Local Alarm	minor	02 Feb 2000 14:32:34	
Rockhurst	Conn	Link Down	MAJOR	02 Feb 2000 14:31:54	TT258K54B

Show alarms: ☑ Critical ☑ Major ☑ minor □ warning ☑ Show Acknowledged

New alarms: 292 C | 157M | 181 m | 164 w

812 alarms loaded at: 02 Feb 2000 14:36:55 □ Auto-refresh list

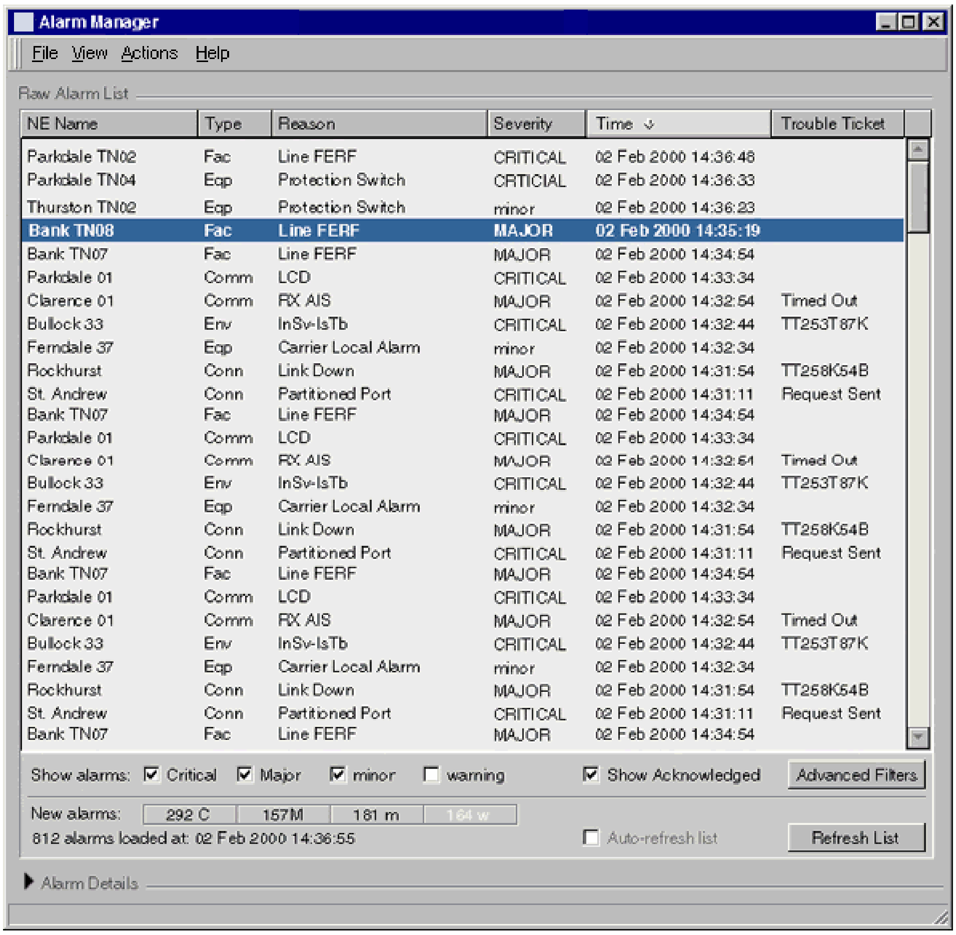
Alarm: 4
Class: Facilities
Unit: OC 1 G2s
Location: Frame
Shelf: 2
Slot: 5
CLFI: 12343425654678900864543456945322

Reason: Line far end receive failure

Acknowledged by
User:
Node ID:
Time:

Figure 8-17/Z.372 – Alarm manager windows example – Details expanded

Figure 8-18 shows that when the details area is collapsed, there is more screen real estate for the list and the number of visible lines is increased.



NE Name	Type	Reason	Severity	Time ↓	Trouble Ticket
Parkdale TN02	Fac	Line FERF	CRITICAL	02 Feb 2000 14:36:48	
Parkdale TN04	Eqp	Protection Switch	CRITICIAL	02 Feb 2000 14:36:33	
Thurston TN02	Eqp	Protection Switch	minor	02 Feb 2000 14:36:23	
Bank TN08	Fac	Line FERF	MAJOR	02 Feb 2000 14:35:19	
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	
Parkdale 01	Comm	LCD	CRITICAL	02 Feb 2000 14:33:34	
Clarence 01	Comm	RX AIS	MAJOR	02 Feb 2000 14:32:54	Timed Out
Bullock 33	Env	InSv-IsTb	CRITICAL	02 Feb 2000 14:32:44	TT253T87K
Ferndale 37	Eqp	Carrier Local Alarm	minor	02 Feb 2000 14:32:34	
Rockhurst	Conn	Link Down	MAJOR	02 Feb 2000 14:31:54	TT258K54B
St. Andrew	Conn	Partitioned Port	CRITICAL	02 Feb 2000 14:31:11	Request Sent
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	
Parkdale 01	Comm	LCD	CRITICAL	02 Feb 2000 14:33:34	
Clarence 01	Comm	RX AIS	MAJOR	02 Feb 2000 14:32:54	Timed Out
Bullock 33	Env	InSv-IsTb	CRITICAL	02 Feb 2000 14:32:44	TT253T87K
Ferndale 37	Eqp	Carrier Local Alarm	minor	02 Feb 2000 14:32:34	
Rockhurst	Conn	Link Down	MAJOR	02 Feb 2000 14:31:54	TT258K54B
St. Andrew	Conn	Partitioned Port	CRITICAL	02 Feb 2000 14:31:11	Request Sent
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	
Parkdale 01	Comm	LCD	CRITICAL	02 Feb 2000 14:33:34	
Clarence 01	Comm	RX AIS	MAJOR	02 Feb 2000 14:32:54	Timed Out
Bullock 33	Env	InSv-IsTb	CRITICAL	02 Feb 2000 14:32:44	TT253T87K
Ferndale 37	Eqp	Carrier Local Alarm	minor	02 Feb 2000 14:32:34	
Rockhurst	Conn	Link Down	MAJOR	02 Feb 2000 14:31:54	TT258K54B
St. Andrew	Conn	Partitioned Port	CRITICAL	02 Feb 2000 14:31:11	Request Sent
Bank TN07	Fac	Line FERF	MAJOR	02 Feb 2000 14:34:54	

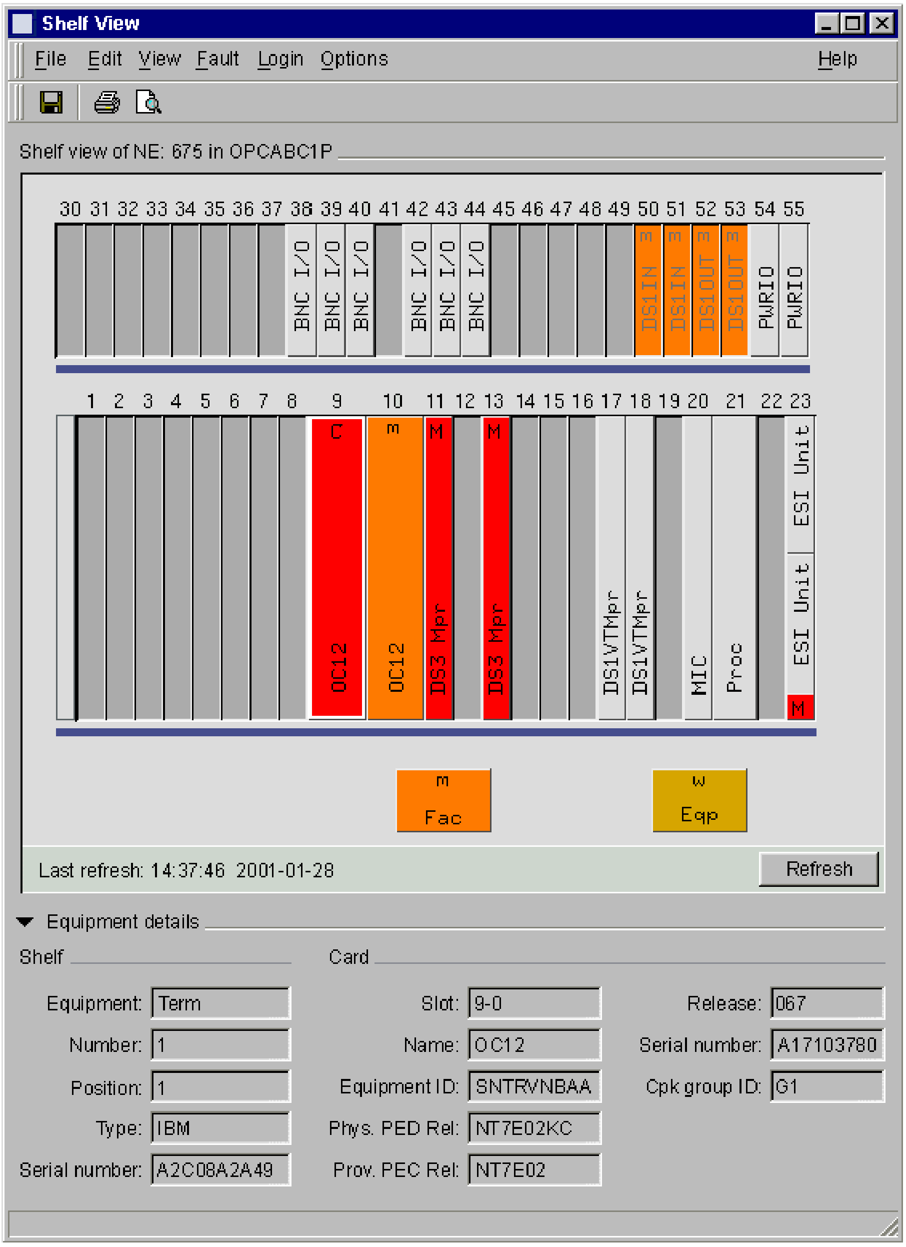
Show alarms: ☑ Critical ☑ Major ☑ minor □ warning ☑ Show Acknowledged

New alarms: 292 C 157M 181 m 164 w

812 alarms loaded at: 02 Feb 2000 14:36:55 □ Auto-refresh list

Figure 8-18/Z.372 – Alarm manager example – Details collapsed

The next example shows all the cards in one or more shelves of equipment within an element. It can provide an overview of the configuration for surveillance and faultfinding and as a launch point for Fault Management, Configuration, and Performance tasks.



Shelf view of NE: 675 in OPCABC1P

Last refresh: 14:37:46 2001-01-28

Shelf

Equipment: Term

Number: 1

Position: 1

Type: IBM

Serial number: A2C08A2A49

Card

Slot: 9-0

Name: OC12

Equipment ID: SNTRVNBAA

Phys. PED Rel: NT7E02KC

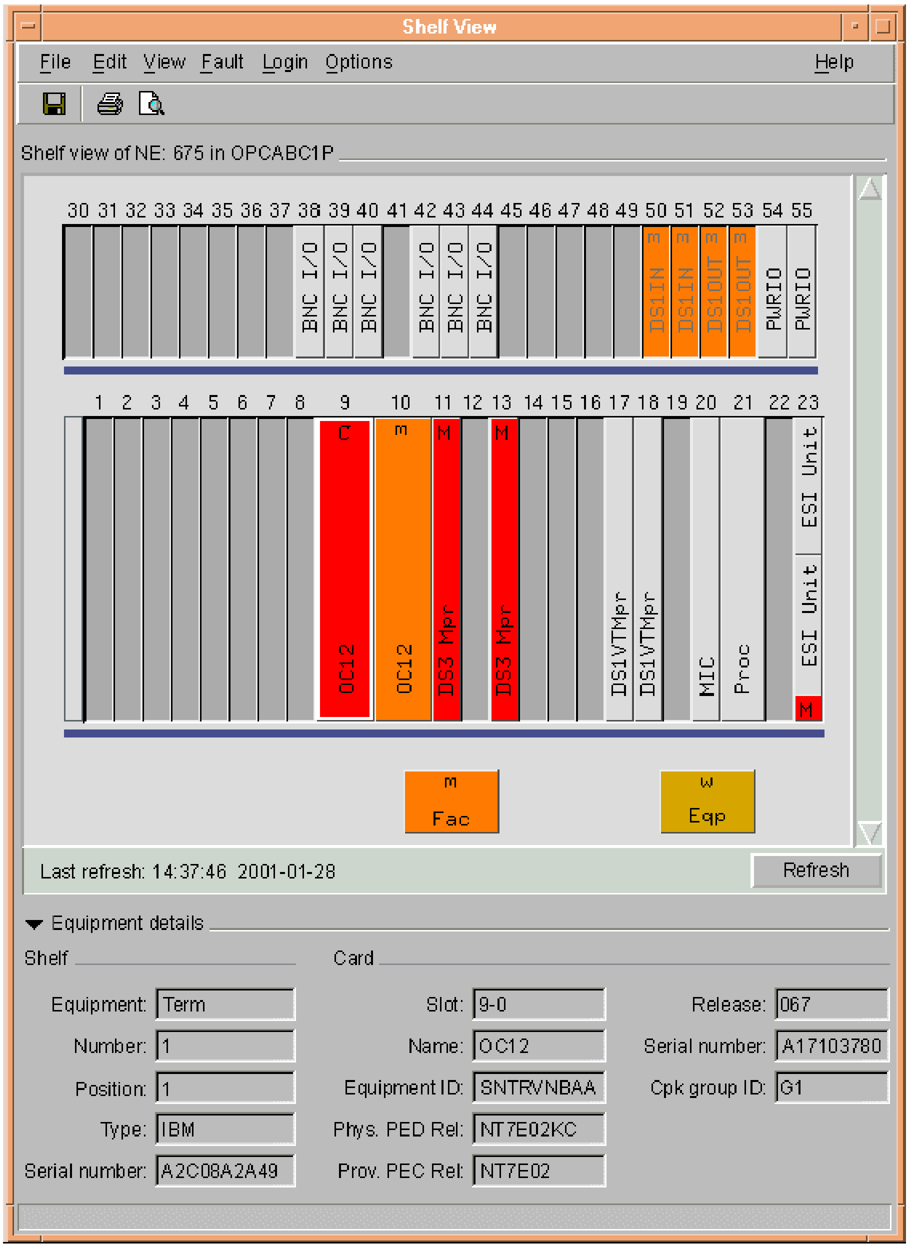
Prov. PEC Rel: NT7E02

Release: 067

Serial number: A17103780

Cpk group ID: G1

Figure 8-19/Z.372 – Shelf view template for Windows



Shelf view of NE: 675 in OPCABC1P

Last refresh: 14:37:46 2001-01-28

Equipment details

Shelf

Equipment: Term

Number: 1

Position: 1

Type: IBM

Serial number: A2C08A2A49

Card

Slot: 9-0

Name: OC12

Equipment ID: SNTRVNBAA

Phys. PED Rel: NT7E02KC

Prov. PEC Rel: NT7E02

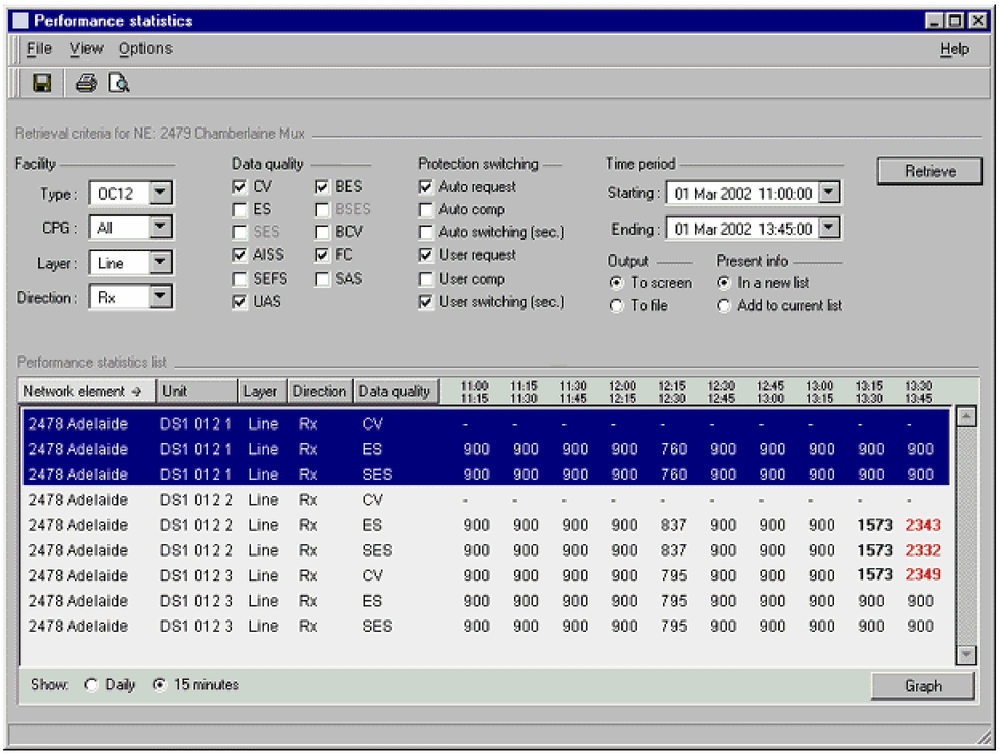
Release: 067

Serial number: A17103780

Cpk group ID: G1

Figure 8-20/Z.372 – Shelf view template for UNIX

Performance management provides functions that evaluate and report on the behaviour of equipment and the effectiveness of the network, or network element. It provides statistics for monitoring, evaluating, and correcting network behaviour.



Performance statistics

File View Options Help

Retrieval criteria for NE: 2479 Chamberlaine Mux

Facility
Type : OC12
CPG : All
Layer : Line
Direction : Rx

Data quality
☑ CV ☑ BES
□ ES □ BSES
□ SES □ BCV
☑ AISS ☑ FC
□ SEFS □ SAS
☑ UAS

Protection switching
☑ Auto request
□ Auto comp
□ Auto switching (sec.)
☑ User request
□ User comp
☑ User switching (sec.)

Time period
Starting : 01 Mar 2002 11:00:00
Ending : 01 Mar 2002 13:45:00

Output
● To screen
○ To file

Present info
● In a new list
○ Add to current list

Retrieve

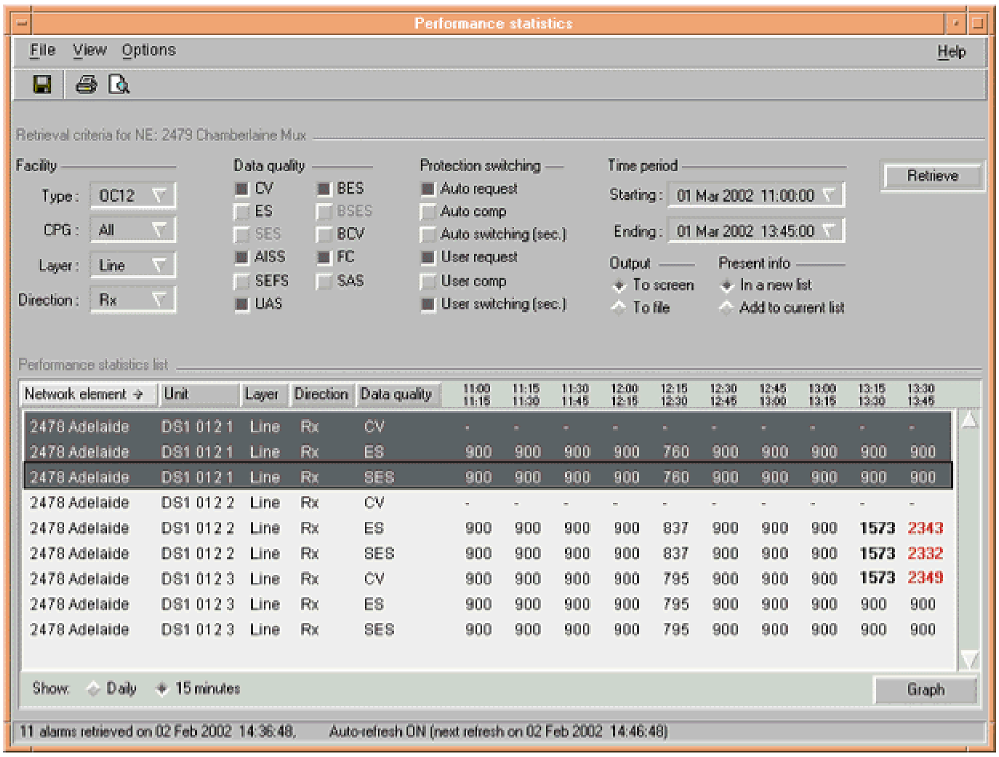
Performance statistics list

Network element	Unit	Layer	Direction	Data quality	11:00 11:15	11:15 11:30	11:30 11:45	12:00 12:15	12:15 12:30	12:30 12:45	12:45 13:00	13:00 13:15	13:15 13:30	13:30 13:45
2478 Adelaide	DS1 012 1	Line	Rx	CV	-	-	-	-	-	-	-	-	-	-
2478 Adelaide	DS1 012 1	Line	Rx	ES	900	900	900	900	760	900	900	900	900	900
2478 Adelaide	DS1 012 1	Line	Rx	SES	900	900	900	900	760	900	900	900	900	900
2478 Adelaide	DS1 012 2	Line	Rx	CV	-	-	-	-	-	-	-	-	-	-
2478 Adelaide	DS1 012 2	Line	Rx	ES	900	900	900	900	837	900	900	900	1573	2343
2478 Adelaide	DS1 012 2	Line	Rx	SES	900	900	900	900	837	900	900	900	1573	2332
2478 Adelaide	DS1 012 3	Line	Rx	CV	900	900	900	900	795	900	900	900	1573	2349
2478 Adelaide	DS1 012 3	Line	Rx	ES	900	900	900	900	795	900	900	900	900	900
2478 Adelaide	DS1 012 3	Line	Rx	SES	900	900	900	900	795	900	900	900	900	900

Show: ○ Daily ● 15 minutes

Graph

Figure 8-21/Z.372 – Performance statistics example for Windows



Performance statistics

File View Options Help

Retrieval criteria for NE: 2479 Chamberlaine Mux

Facility
Type : OC12
CPG : All
Layer : Line
Direction : Rx

Data quality
■ CV ■ BES
□ ES □ BSES
□ SES □ BCV
■ AISS ■ FC
□ SEFS □ SAS
■ UAS

Protection switching
■ Auto request
□ Auto comp
□ Auto switching (sec.)
■ User request
□ User comp
■ User switching (sec.)

Time period
Starting : 01 Mar 2002 11:00:00
Ending : 01 Mar 2002 13:45:00

Output
◆ To screen
◇ To file

Present info
◆ In a new list
◇ Add to current list

Retrieve

Performance statistics list

Network element	Unit	Layer	Direction	Data quality	11:00 11:15	11:15 11:30	11:30 11:45	12:00 12:15	12:15 12:30	12:30 12:45	12:45 13:00	13:00 13:15	13:15 13:30	13:30 13:45
2478 Adelaide	DS1 012 1	Line	Rx	CV	-	-	-	-	-	-	-	-	-	-
2478 Adelaide	DS1 012 1	Line	Rx	ES	900	900	900	900	760	900	900	900	900	900
2478 Adelaide	DS1 012 1	Line	Rx	SES	900	900	900	900	760	900	900	900	900	900
2478 Adelaide	DS1 012 2	Line	Rx	CV	-	-	-	-	-	-	-	-	-	-
2478 Adelaide	DS1 012 2	Line	Rx	ES	900	900	900	900	837	900	900	900	1573	2343
2478 Adelaide	DS1 012 2	Line	Rx	SES	900	900	900	900	837	900	900	900	1573	2332
2478 Adelaide	DS1 012 3	Line	Rx	CV	900	900	900	900	795	900	900	900	1573	2349
2478 Adelaide	DS1 012 3	Line	Rx	ES	900	900	900	900	795	900	900	900	900	900
2478 Adelaide	DS1 012 3	Line	Rx	SES	900	900	900	900	795	900	900	900	900	900

Show: ◇ Daily ◆ 15 minutes

Graph

11 alarms retrieved on 02 Feb 2002 14:36:48, Auto-refresh ON (next refresh on 02 Feb 2002 14:46:48)

Figure 8-22/Z.372 – Performance statistics sample for UNIX

BIBLIOGRAPHY

[B1] ISO 9241-1:1997, *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 1: General introduction*.

[B2] ETSI EG 201 024 V1.1.1 (1997), *Human Factors (HF); User interface design principles for the Telecommunications Management Network (TMN) applicable to the "G" Interface*.

[B3] SONET Interoperability Forum, SIF-007-1996, *Design Principles for the development of OAM Graphical User Interfaces*.

[B4] ITU-T Recommendation M.3020 (2000), *TMN Interface Specification Methodology*.

[B5] ITU-T Recommendation M.3200 (1997), *TMN management services and telecommunications managed areas: Overview*.

[B6] ITU-T Recommendation M.3300 (1998), *TMN F interface requirements*.

[B7] ITU-T Recommendation M.3400 (2000), *TMN management functions*.

[B8] ITU-T Recommendations M.3208.x series, *TMN management services for dedicated and reconfigurable circuits nework*.

[B9] X.700 series of ITU-T Recommendations, *OSI management standards*.

[B10] T1M1.5 T1.200 Series, *TMN Architecture, Models, Functions and Protocols*.

[B11] ITU-T Recommendation E.134 (1993), *Human factors aspects of public terminals: Generic operating procedures*.

[B12] ITU-T Recommendation E.135 (1995), *Human factors aspects of public telecommunication terminals for people with disabilities*.

[B13] *Special Issue on Patterns and Pattern Languages*, Vol. 39, No. 10, October 1996.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	**Languages and general software aspects for telecommunication systems**

Printed in Switzerland
Geneva, 2005