INTEGRATED COMMUNICATIONS MANAGEMENT OF BROADBAND NETWORKS

Crete University Press, Heraklio, Greece
ISBN 960 524 006 8

Edited by David Griffin

Copyright © The ICM consortium, Crete University Press 1996

First published in 1996 by
CRETE UNIVERSITY PRESS
Foundation for Research and Technology
P.O. Box 1527, Heraklio, Crete, Greece 711 10
Tel: +30 81 394235, Fax: +30 81 394236
email: pek@iesl.forth.gr

Copyright © The ICM consortium, CUP 1996

The ICM consortium consists of the following companies:

Alcatel ISR, France
Alpha SAI, Greece
Ascom Monetel, France
Ascom Tech, Switzerland
Centro de Estudos de Telecommunicações, Portugal
Cray Communications Ltd., United Kingdom (Prime contractor)
Danish Electronics, Light & Acoustics, Denmark
De Nouvelles Architectures pour les Communications, France
Foundation for Research and Technology - Hellas, Institute of Computer Science, Greece
GN Nettest AS, Denmark
National Technical University of Athens, Greece
Nokia Corporation, Finland
Queen Mary and Westfield College, United Kingdom
Unipro Ltd., United Kingdom
University College London, United Kingdom
University of Durham, United Kingdom
VTT - Technical Research Centre of Finland

Chapter 8

Human Computer Interface issues

Editors: Cyril Autant, Richard Lewis
Authors: Pekka Jussila, Kyriakos Varvaressos, Xavier Pandolfi,
Petri Niska, Panos Georgatsos, Patrick Legand, Cyril Autant

When designing complex systems, care must be taken to hide complexity from the end-user. All the functions offered should be accessible in a straightforward and friendly way, without the need for detailed knowledge of the system architecture. This requires special attention in the context of TMN owing to the complexity of both managed and managing systems.

This chapter describes some of the issues which the project encountered and addressed during the design and implementation of three TMN Human Computer Interfaces (HCI) which form part of the ICM Testbed.

8.1 Introduction

8.1.1 The Human Computer Interfaces designed within ICM

The HCI is the information exchange point between the system and the user of a TMN system. Within ICM, three HCIs have been designed:

- The Public Network management system HCI, as used in the VPCM system (see Chapter 5).

- The Value Added Service management system HCI, as used in the VASP TMN for VPN management (see Chapter 6).
- The ICM ATM Simulator HCI (see Chapter 11). For the sake of clarity, this is generally referred to as the “MMI” or “SimMMI” in order to differentiate it from the TMN HCIs.

These HCIs support network and service Performance and Configuration Management, as well as the specification and execution of simulation experiments. Throughout the development of these three human-computer interfaces it has been important to keep the same rules and the same approaches to represent data. Therefore common rules have been used to design and implement the HCIs and the same logic has been followed in the way information and actions are presented to the end-user.

The remainder of this section provides an overview of the role of these HCIs within the ICM Testbed. The next section introduces the general and specific requirements, that serve as a basis to design the HCIs. This is followed by a description of how these requirements are met, including descriptions of the design approaches, and a summary of the three implementations.

8.1.1.1 Role of VPCM HCI

As part of the ICM system, the VPCM HCI interprets the information which must be presented to the user; it translates the information held in the information models into a displayable format for the human operators, and vice-versa. The HCI presents the TMN management capabilities for human information and/or intervention. Given that different users may have different needs (depending on their level of experience and role in the organisation), a customised view of data is provided and adapted management capabilities are offered. This HCI presents the user with details of connection acceptance and rejection, in order to allow the operator to be aware of traffic levels and exception conditions.

8.1.1.2 Role of VASP HCI

The VASP Workstation HCI is used by human operators employed by the VASP. It is used to graphically represent customer iVPNs and end-to-end VPCs used to create iVPNs. The operators can create views of new customers or Public Networks once appropriate contracts with these parties have been created with the VASP. For existing customers, the VASP HCI can be used to create new end-to-end VPCs between CPN endpoints. The HCI helps to select the most appropriate route for new end-to-end VPCs, based on the customer's requirements (e.g. bandwidth, CDV, CoS, cost, etc.) and available Public Network connectivity.

8.1.1.3 SimMMI network configuration role

The role of the SimMMI is twofold: support of stand-alone simulation runs and support of management experiments. In both contexts, the simulator’s man-machine interface, the SimMMI, acts as a network configuration and monitoring tool. Through its facilities users can configure the network and also initiate, control and monitor simulation runs. One role, therefore is to support the physical and logical configuration of experimental networks and their characteristics, including user and traffic profiles, as well as to present these configurations back to the user.

8.1.1.4 SimMMI experiment support role

A significant aspect of the role of the SimMMI is that it can act as a testing and validation tool for the underlying simulator. Through it, users can see and analyse the results of experiments. In this context the SimMMI also acts as a meta-management tool, whereby users can initialise and launch the required TMN processes for the management of the network. In this role, the SimMMI offers facilities enabling the users to:

- configure the actual TMN, by distributing the TMN processes to TMN hosts as appropriate,
- initialise the TMN processes, by downloading appropriate network and operational parameters information, and,
- instantiate the TMN processes, in a user-selected order.

The network configuration information passed to the TMN processes concerns network topology, the connection types supported by the network and network traffic information, as specified by the simulator user. Management experiments in a simulated network environment should employ a certain degree of flexibility in terms of network configurations and actual TMN configurations. It should be stressed that, in this context, the SimMMI also plays the role of a testing and debugging tool for the initialisation and activation of the TMN processes.

The design and the implementation of the SimMMI have been influenced from the fact that such a tool must be able to be used by a variety of users within or outside of the context of the ICM project. The potential users of the SimMMI are:

- ATM simulator developers

  Apart from its obvious role as a configuration tool, the SimMMI has been extensively used as a validation and debugging tool for the simulator itself. Through its monitoring and result analysis facilities the output of the simulator can be analysed and therefore be verified.
- TMN developers

  The SimMMI acts as a start up tool for the TMN processes managing the network. Through its start-up facilities, the TMN processes can be instantiated in a consistent way and the configuration of the TMN can be tuned. Problems related with initialisation and activation can be identified.
- ATM network designers

  The SimMMI offers statistical analysis and logging facilities per simulation run, which in the context of management experiments with the simulator can aid optimal ATM network designs.
- ATM network novice users

  The SimMMI has been designed following standard ATM networking concepts and terms, providing a visual representation of them; therefore it offers an opportunity to novice users to get acquainted with the ATM networking framework.

The basic features of the SimMMI are presented in Chapter 11 (Section 11.4.1). Design and implementation aspects of the SimMMI are described in this chapter (Section 8.5).

8.2 HCI requirements

Modern technical systems are becoming increasingly complex, which can make them difficult for their users to understand, especially in the case of unusual situations or malfunctions. Different kinds of complexity can be distinguished:

- Complexity coming from the need to integrate in the same application various paradigms and data which are organised within different schemes.
- Complexity arising when the user has to face complex situations where decisions must be taken quickly, while the amount of data to analyse is very large.
- Integration complexity, concerning requirements for systems to be able to interoperate with other sub-systems.
- Representation complexity, dealing with the need to provide support for knowledge representation of strongly structured data (cartography, multi-media).
- Finally, the "Man Machine Interface complexity" coming from an increasing need from the users to develop the interactivity and look at the same situation from different viewpoints. The HCI must give the user the possibility to take quick decisions.

Traditionally, the designers of complex systems have invested much care and effort into HCI design. In the past decade, the potential of new information processing techniques has been exploited in order to improve the presentation of information and guidance to the operator in a safer and more flexible way. This tendency exploits advances in two major fields: artificial intelligence and graphical interfaces. Intelligent user interfaces [8.25] help support the user by suggesting appropriate actions or by adapting the interface according to context; graphics help present complex information to the user in a more easily understood way.

Research into the subject has resulted in successful HCI prototypes and confirmed many of the findings of psychologists and cognitive scientists. It has also revealed a number of additional problems that had not been previously imagined and has produced new insight into the complex interrelations between human and technical systems.

The requirements for Network Management (NM) HCI are divided into General HCI and TMN HCI requirements. These requirements are presented below.

8.2.1 General HCI requirements

General HCI requirements are those which apply to any system, including TMN systems. These are described below, classified according to usability attributes [8.20].

8.2.1.1 Learnability

Learnability is important for novice users, i.e. those using the HCI for the first time. Every user starts in this group. Graphical user interfaces (GUIs) have been found useful for them and thus should be used for the WS-OS realisation (R1).[1] At present, GUIs

1. See Table 8.1 on page 216.

have become a standard in the user interface (UI) field. A GUI contains windows, a mouse or some alternative pointing device and a high-resolution video display.

The windowing system in a GUI acts as a front end to the operating system [8.17, pp. 137-183]. It manages objects like the screen, windows and input devices. The windowing system must fulfil some basic conditions for applications such as the WS-OS to be learnable:

- The UI should be mode-less [8.19, pp. 451-452] (R2).
- On-line help should be available [8.20] (R3).
- Dialogue and messages should be expressed using natural language with specialised vocabulary (R4), in spite of keeping them short and simple [8.20, pp. 115-116, 123-126]. Abbreviations and specialised vocabulary should be explained using hyperlinks in on-line help (R5), but at most to three levels in depth, as required by the memorability requirement.

  There is a possible problem with terminology: if the expressions in the UI are kept short, specialised language and abbreviations will be used. Different users, however, may want to use different terminology. This contradiction can be solved by allowing the users to define their own vocabulary (R6).
- Graphical network presentation should be available as a tool in the UI (R7) providing a strong metaphor or mapping [8.20, pp. 126-129] between the user model of a network and the WS-OS managing it.

The operator interface is not assumed to be an “immediately usable” system. Learnability is considered less important if conflicting with other usability issues. This is due to the expected user categories being expert and casual users, not novice users. In spite of this, the user should be able to use the interface at least to some extent from the very beginning, if familiar with windowing UIs (R8).

8.2.1.2 Memorability

Casual users are those having used a WS-OS before, but rather more intermittently than expert users. Network configuration tasks are rather infrequently used, unlike fault management and performance monitoring. For casual users, the system memorability is the most important property of the WS-OS. The keys to memorability are familiarity, visual attraction and simplicity.

- Familiarity is achieved through homogenous presentation. This is partly achieved through the use of guidelines [8.21] and (de facto) standards. Guidelines give a rational, tested basis for UI development at the level of simple components. De facto standards may be followed in order to integrate the tools with those that the users are already familiar [8.18]. The Motif guidelines [8.21] are followed in the WS-OSs, but use of other (de facto) standards is optional (R9). A common HCI for all applications in a TMN workstation is one way to achieve this familiarity. Inside a WS-OS, familiarity should be achieved by internal commonality of the interface components (R10).
- The objects in the UI should be visually attractive and clear, since objects with these properties imprint themselves in the minds of users. For visual attractiveness, special attractors should be used for most important objects with multiple representations (R11). For clarity, the main domain for the network manage-

ment functions, i.e. the basic network presentation, should be a figure of a network and as a composition of subnetworks when needed (R12). This provides the user with a clear mental model of the managed domain and may be modified as needed for the user commands.

- Interactions should be kept simple, allowing at most three window or menu levels in depth (R13). This enables the user to keep all things associated with interactions in short-term memory.
- The simpler the UI, the easier it is to use. Cognitive overload [8.11, pp. 72-83], should be avoided, by managing and reducing the incoming information (R14). The computer should be left to do the work and bring the information to the user. In tasks such as performance monitoring or fault management, this is achieved through filtering. In other object management tasks, the focus should rather be on rational semantics, simplicity and consistency of the interface. Combined this with the requirement for the depth of interactions (R13), no task without saveable results should be more complicated than three steps in the task model (R15).

8.2.1.3 Satisfaction

For its high information management bandwidth [8.11, pp. 72-83], a GUI should be used for the HCI (R1). In this, the network presentation should be paid special attention.

Usually part of the management tasks involve other OSs than the WS-OS. Therefore, there are two kinds of system feedback (R16):

- *Immediate feedback*, delivering the result of the invoked operation to the user. This is used in tasks local to the OS.
- *Delay feedback*, indicating when an operation has started and been completed, delivering the result (or error status) to the user. This is used in distributed tasks.

User perspective should be used in the interactions in the UI. This means:

- viewing dialogues from the user's perspective (R17), and,
- viewing user functions and utilities from the user perspective instead of the machine’s perspective (R18).

8.2.1.4 Efficiency

Efficiency is the most important usability attribute in the user interfaces of software applications and platforms used to manage broadband telecommunications networks [8.15]. Efficiency is especially important for:

- tasks usually not considered as services supported for everyday use, such as design, planning and installation, and,
- everyday technical tasks, such as maintenance and performance management.

Some requirements to be used for achieving efficiency are listed below.

- The presentation through the HCI should be homogenous (R19) with a framework sufficiently simple but at the same time attractive.
- The incoming information to the user should be filtered (R14), to reduce the cognitive overload recognised as a threat in TMN UIs [8.11, pp. 78-83].
- The HCI should be common for all tasks done (R20).

- The dialogue should be kept simple (R21).
- The general behaviour should be modifiable (R22) to users' preferences.
- There should always be a discrete way to input data from pre-generated choices as selection input (e.g. menus, selection lists and figures) instead of using free-form input (e.g. typing from the keyboard) (R23).
- The user should be offered power commands to apply on groups of intrinsic objects (R24) to avoid executing tasks in too much detail [8.24]. Intrinsic objects are application objects in the HCI, i.e. objects not used by the user to control the program [8.19, p. 449].

The users probably have a clear mental model about the managed domain. In addition to this, they need a mental model about the tasks they must perform. Goal or task hierarchies should be presented explicitly to the users as supported by the management platform or applications (R25). The users should understand how to accomplish their goals, and how different tools and actions help them to achieve their goals.

8.2.1.5 Error prevention and tolerance

Fault tolerance is one of the key issues both in operator tasks and GUIs. An unreliable GUI also needs fault prevention and tolerance, especially when considerable damage can be made using it.

- Event correlation should be exercised (R26), to cope with both faults received from the managed domain, and, information about the management system.
- Common HCI (R20) between management applications and functionalities reduces the possibility of errors.
- Selection input (R23) reduces the possibility of erroneous input.
- Undo and error recovery possibilities should be available (R27). It must be considered to what degree these are provided by the HCI, and to what extent they are provided by the applications themselves. Also the feedback model, whether continuous or discrete, is affected by these facilities [8.24].

8.2.1.6 Summary

Many WS-OS requirements impact usability attributes other than those against which they have been listed above. Table 8.1 presents the usability attributes [8.20, pp. 26-37] in columns and the WS-OS general requirements related to them in rows. The ticks in the cells mark the WS-OS usability attributes positively affected by the WS-OS requirements.

8.2.2 Non-functional requirements

A number of non-functional HCI requirements can be identified, which are not necessarily specific to the TMN. These are described below.

8.2.2.1 Distribution

The software developed for the HCI should be distributable (R28) among hosts. This requirement is satisfied by the platform when using multiple OSs [8.23]. The HCI soft-

	Learnability	Memorability	Satisfaction	Efficiency	Faultlessness
GUI (R1)	✓		✓		
Modelessness (R2)	✓	✓			
On-line help (R3, 25)	✓			✓	✓
Vocabulary (R4, 5)	✓	✓		✓	
Terms (R6)	✓	✓	✓		✓
Net. presentation (R7, 12)	✓	✓	✓		
Standards (R8, 9)	✓	✓		✓	
Commonality (R10)	✓	✓			
Attractors (R11)	✓		✓		
Simplicity (R13, 15, 21)	✓	✓		✓	✓
Filtering (R14, 26)	✓			✓	✓
Feedback (R16)			✓		✓
User perspective (R17, 18)	✓		✓		
Homogeneity (R19)	✓	✓		✓	
Common HCI (R20)	✓	✓		✓	✓
Modifiability (R22)				✓	✓
Selection input (R23)		✓		✓	✓
Power command (R24)				✓	✓
Undo (R27)			✓		✓

Table 8.1 Usability attributes and the general WS-OS requirements related to them

ware should be separated from the application code (R29). This modularity is accomplished by the distributability requirement (R28).

8.2.2.2 Control

When considering HCI design it is important that the user should be in control. This is achieved as follows:

- The user functions should be able to be started and initialised through the HCI, setting arguments as provided by the user function (R30).
- It should be possible to observe the status of a user function through the HCI (R31).
- The user functions should be able to be configured through the HCI (R32).
- The user should get appropriate feedback about any situations in user functions from the HCI (R34). In normal situations this means the results of a user function and in abnormal situations this means error information.

8.2.2.3 Reliability

Software reliability is especially important in network management systems, as they are complex continuous processes. The HCI is no exception to this.

- The HCI software should be fault tolerant and crash recoverable (R33) for errors in the agents in the management system.

8.2.2.4 Performance

Performance is very important in real time systems, in order to be usable and reduce errors. The HCI should be engineered to respond quickly to the user. Where this is not possible, the HCI should indicate that processing is in progress, with some indication of the time needed to complete the task.

8.2.2.5 Security

The security requirements are especially important for a manager operating on agents that provide a restricted managed domain and restricted operations. The following basic requirements can be identified.

- An operation on a managed object is not allowed to affect managed objects belonging to other management domains (R35).
- An operation by the user of the HCI on some object should only be allowed given the appropriate permission level (R36).

There are optional means and policies for restricting access on managed objects. The restrictions may take place at the process level, application-entity level, HCI level, or operator level. There may be different access restriction policies, according to individual users, user groups, operator or user types. The HCI should be able to specify the access restriction level and policy and to use them (R37).

8.2.3 TMN HCI requirements

The networking environment in which the HCI operates, imposes a number of network-specific requirements which need to be taken into account during design and implementation phases. These requirements are discussed in this section.

Requirement set 1: Data entry and presentation requirements
These requirements arise from the huge amount of data to be handled.

Considering the resources that make up an ATM network, the amount of data that need to be input for their configuration and the amount of data that need to be displayed as results of a simulation run is enormous. The amount of data becomes even larger when considering that the HCI must be able to support multiple network domains.

This requires a HCI design and ergonomy that will minimise confusion with respect to the data that needs to be input or displayed. The user's mental images of the network configuration must be as much as possible reflected in the way the system prompts the user for data operations. With respect to network configuration, the HCI should guide and facilitate the users to be able to understand: the data that needs to be input; the data that has already been input; and the data that remains to be input. With respect to presentation of management information, the HCI should direct the user to view information in an organised way. In addition, the data handling system of the HCI should be able to cope with the huge amount of data, without distorting its display facilities. The time required for the user to input all the required data is another critical issue.

Requirement set 2: Data consistency requirements

These requirements arise from dependencies among configuration data.

Network resource configuration may depend on the configuration of other resources. For instance, the configuration of route selection entries (definition of priority) depends on the number of connection types defined to use the routes; or the definition of network traffic depends on the defined access nodes and on the defined network services which in turn depend on the defined connection types. Therefore, changes in resource configuration must always be presented to or at least be easily realised by the user.

In addition, consistency checking facilities must be provided so that the users are able to cross-check network configuration information that it is not directly presented at configuration time. For instance, after the definition of VPCs, it is helpful for the user to know how many VPLs have been defined on a given link or how many routes are using each VPC. The system should never allow incomplete or inconsistent information that might cause malfunction.

Requirement set 3: Graphical User Interface requirements

These requirements are related to the creation and maintenance of large networks.

Scaling both in terms of network size (number of network resources) and networks (number of network domains) were posed by ICM. Therefore, the HCI should be able to cope with large networks. Handling of large network is meant from the view of network display and supported operations rather then from the view of handling the related information (considered in the first set of requirements). The interface, therefore, needs to provide enhanced zooming and network presentation facilities to allow users to operate efficiently even when dealing with large networks.

Requirement set 4: Human factor requirements

These requirements are related to the usability and the 'look and feel' of the system.

The general HCI requirements described in Section 8.2.1 can be applied to any computer system from a single-user word processor to the most complex real-time system. The TMN is in the latter category and the complexity of the TMN, the consequences of network failure, and the technical nature of the system render the importance of these requirements sufficiently different from those implied by the simpler case to merit further definition.

The HCI is to be used both by novice and expert users either as a configuration or monitoring tool. Although the operations offered by the system are presented in a way forcing the novice user to use them in a certain order, an expert user must be able to use these operations in a fluent way, concurrently and repeatedly. Therefore, the ergonomy of the HCI should provide an optimum trade-off in the usability between novice and expert users. On one hand the HCI should guide novice users to the correct sequence of operations and on the other hand the interface should not appear too overprotective, therefore boring and slow, to the expert users.

The ‘easy-to-learn’ aspect of the HCI usability is a serious aspect which becomes more critical considering the number of operations that a user can apply to managing networks and services.

The ‘look and feel’ of the interface, apart from the usual standards (buttons, window style, menus etc.), must reflect the type and the style of the user work. The HCI should be designed in a way to emphasise the important aspects of the users’ current work, hiding all other unrelated aspects. Moreover, since the user is expected to use the interface for a considerable amount of time, tools for customising network resource or result presentations should be provided.

8.2.4 Fulfilling the requirements

The following paragraphs describe how some of the specific TMN requirements detailed in Section 8.2.3 have been addressed by the project. This description is based on the Simulator MMI (SimMMI) as this is a superset of a TMN HCI, supporting configuration and monitoring as well as network simulation. However, the same approaches are relevant to TMN HCIs.

Fulfilling requirement set 1: Data entry and presentation

To fulfil the first set of requirements, the SimMMI offers an explicit separation of activities and a clear decomposition of network related information. The user realises, and furthermore is directed to follow, this decomposition by the ergonomy of the interface. In particular, the HCI works in four modes, each encompassing its own activities. The HCI presents to the user only the operations that are related to the current working mode. The working modes of the HCI are:

- network configuration mode,
- running mode,
- play-back mode,
- result mode.

A description of the SimMMI features in each of its modes can be found in Chapter 11.

The network configuration information that needs to be input by the user is decomposed into the following sets of information:

- *network connection types information*, which includes specification of:
  - bandwidth characteristics,
  - performance characteristics,
  - burst generation model parameters (see Chapter 11, Section 11.3.3.3),
- *network services information,* which includes specification of constituent connection types and their direction,
- *topological information*, which includes definition of network domains and placement of nodes and links inside each domain,
- *physical resource configuration information,* which includes specification of:
  - node type (CPN, access, transit, gateway),
  - node switch architecture (one buffer per link, shared buffer),
  - node buffers’ size,
  - link capacity, delay and maximum number of VPLs to be defined.

- *network traffic information*, which includes specification of:
  - network services,
  - service patterns (see Chapter 11, Section 11.3.1),
  - user groups (see Chapter 11, Section 11.3.1),
- *VPC information*, which includes specification of VPC topology and bandwidth.
- *routing information*, which includes definition of:
  - routes per connection type, and,
  - definition of appropriate route selection entries.

The order of the above information categories proposes a clear sequence of the configuration activities that need to be undertaken for the complete network configuration. This order is enforced by the SimMMI. Restrictions applied to the user actions must not surprise the user and must be a logical consequence of some missing actions. For example, in order to create network traffic the user must first define the services supported by the network. It is thought to be very helpful if users would know the exact sequence of the activities that must perform in order to configure properly a network. For this reason, apart from the user's ability to rely on the HCI ergonomy to find out the required configuration activities, the SimMMI offers an *activity-oriented network configuration* facility that completely guides the users of the necessary configuration activities. This facility operates per either a single or multiple network domains.

A further segregation of the above sets of information is provided by considering the connection and service type information independent of the specific network configuration information. The users can define connection and service types independently of the networks; this information may be even supplied by other users. The configuration of a specific network in terms of the connection types that supports and the services that offers to its users is done by simply selecting a specific set of predefined connection and service types.

In order to minimise network configuration time, the SimMMI provides tools facilitating quick network configuration (e.g. copy, setting a common configuration to all network resources of the same type etc.), tools for re-using existing configurations and tools for tailoring the configuration of specific group of resources.

To better organise the user's work in a systematic and structured way, the SimMMI offers various administration and file management tools. These tools allow the users to store their work independently from other simulator users as well as to copy networks to/from other users.

To further enhance decomposition of information and separation of operations, the SimMMI offers two explicit views of each network domain:

- the physical view, and,
- the logical view

In the physical view the network physical resources (nodes, links) are displayed whereas in the logical view the logical network entities (VPCs, routes) are displayed. The physical and logical views of the network are available not only during configuration but also in the other modes; therefore not only configuration but also network simulation result information decomposition is achieved. Apart from the obvious benefits from the display viewpoint, the physical and logical views make the user think along this distinction, proposing to them that configuration of logical resources cannot be made unless the physical resources have been configured.

For handling network result information, the SimMMI offers a clear distinction between *networks* and *experiments*. The notion of an experiment is used to denote a network that has completed a simulation run; an experiment is made up from a network, run-time conditions specifications and the results of a simulation run. Different experiments may be defined per network. An experiment is treated as an entity in its own right. Users can load experiments to view results. Different experiments can be viewed and their results may be compared. Result information is requested and displayed in the appropriate network view and is available on a resource or network-wide basis. In particular, the following result information can be requested and displayed:

- In the physical view:
    - Per node (e.g., buffer cell loss rate),
    - Per link (e.g., utilisation),
    - Network-wide (e.g., link utilisation).
- In the logical view:
    - Per node (e.g., node connection rejection ratio),
    - Per VPC (e.g., connection rejection ratio),
    - Network-wide (e.g., connection rejection ratio).

In the result mode, the configuration operations disappear, leaving to the user the operations and facilities required for the request and display of the results.

By explicitly considering the configuration and running modes, the users' options are restricted. While in the configuration mode, the user is concerned only with the work of configuring the network. Only when it enters the running mode, does the SimMMI reveal the options related to the dynamic aspects of the network simulation. The configuration aspects are then hidden.

For handling information in the running and play-back mode, the SimMMI uses a similar decomposition of information as in the result mode, that is, it decomposes information to be displayed per resource type (physical or logical) and per actual resource. However, the presentation of the information in these modes is according to the nature of the modes. The information is presented to give the feeling of a running environment to the user. The information displayed is in the form of events associated with a particular resource. In the running mode, this information is displayed on-line, as it is received from the simulator, whereas in the play-back mode this information is played-back, in slow-motion, at a user defined pace. For system performance reasons, the information to be displayed is user selectable. The following information (events) can be displayed:

- In the physical view:
    - Per node (e.g. buffer cell loss events),
    - Per link (e.g. full capacity transmission events).
- In the logical view:
    - Per node (e.g. connection request related events),
    - Per VPC (e.g. connection request related events).

Fulfilling requirement set 2: Data consistency

To satisfy the requirements coming from the dependency among different configuration data, the SimMMI prompts users of required configuration changes. The effect of changing the configuration of a resource on the configuration of other resources is assessed automatically by the SimMMI and in response it either warns the user of the consequences in the configuration of other resources or prevents the user from undertaking the specific action.

In addition, it offers powerful *resource configuration queries* with the purpose of helping users to cross-check their network configuration data or identify missing information. The queries relate resource configuration information in higher forms that are not directly presented to the user at resource configuration times. The queries are offered per resource or network-wide basis and they are the following:

- In the physical view:
  - Per node (e.g. node traffic),
  - Per link (e.g. VPLs defined).
- In the logical view:
  - Per node (e.g. node to node route connection related queries),
  - Per VPC (e.g. routes defines using the VPC),
  - Network-wide (e.g. class route network views).

Fulfilling requirement set 3: Graphical User Interface

For handling large networks and multiple network domains, the SimMMI offers extended zooming and scrolling facilities. Zooming is done by appropriately adjusting the scaling of network presentations. Two options for scrolling are provided: (a) by moving the standard scroll-bars, and (b) by moving an area on a miniature network representation, whereby the part of the network included in the area is displayed in the user viewable area.

In addition, the SimMMI offers various presentation views of the networks, allowing the users to adjust network presentation to the needs of their current work activity. The following *network presentation options* are offered to the users:

- *'closed network presentation'*, whereby whole networks, or sub-networks, are viewed as single visual objects,
- *'gateway network presentation'*, whereby only the gateway nodes of the networks are displayed with the international links between them,
- *'open network presentation'*, whereby networks are viewed with all of their nodes and links,
- *'local view of the network around a node'*, whereby the local part of the network around a node is displayed; this part includes the node incoming, outgoing links and their capacities, and its buffers, all arranged according to the selected switching architecture (one buffer per outgoing link or shared buffer).

The above presentation views are available both in the physical and logical views of the networks (apart from the last one which is available only in the physical view).

Fulfilling requirement set 4: Human factors

To satisfy the requirements about usability, the SimMMI has been designed following standard ATM networking notions and terms. Therefore, it is believed that it offers a fair trade-off between novice and expert users. The minimum requirement posed to novice users is a basic knowledge of ATM networks. The expert users, on the other hand, will find themselves immediately in a familiar environment, ready to take on the required activities for network configuration and running.

The easy-to-learn aspect of the HCI is based on the fact that the SimMMI has been designed following the usual GUI standards. Apart from this, the SimMMI tries to unify operations, utilising similar display customs for the ATM entities. For example, in the SimMMI the activity of configuring VPCs - selection of a sequence of links in the physical view - is similar to the route definition activity - selection of a sequence of VPCs in the logical view. Display consistency is another issue that has been pursued. In all of its working modes, the SimMMI uses similar display entities for displaying and acquiring information.

To satisfy the 'look and feel' requirements, apart from the fact that it has been built following the usual GUI standardisation, the SimMMI offers options to adjust the network presentation according to the user's current activity. In particular, the SimMMI, besides the network presentation options described previously, it offers the following *resource presentation* options:

- For nodes:
  - by name,
  - by type (CPN, access, transit, gateway),
  - by switch architecture.
- For links or VPC:
  - by name,
  - by icon.

Furthermore, the SimMMI offers its facilities and displays information in a way that suits the current working mode. For instance, the information displayed in the running mode gives the feeling to the user that something is happening, while at the same time the user is informed about the network state.

In addition, the SimMMI allows users to customise their environment, by offering *dynamic colour adjustment* facilities. This way, the users may highlight visual objects on the screen (nodes, links, VPCs, routes), putting emphasis as desired.

8.3 Implementation issues

8.3.1 Introduction

Intelligent HCIs and advanced methods to develop such interfaces are two forms of Advanced Information Processing technologies considered in the ICM project. HCI and HCI development are still relatively new fields in TMN development, although of growing interest, because of increasingly complicated TMN environments and management applications [8.16]. The HCI is clearly a key component in any TMN system, as it is the information exchange point between the user and the system. In complex systems where the number of managed elements might be large, the HCI has to provide to the end-user the means to handle the volume of information in an efficient and reliable way.

8.3.2 HCI functionality and architecture in the ICM testbed

In this section, we introduce the Workstation (WS) and focus on the Workstation architecture. In this context, the Human Computer Interface (HCI) is what the user actually sees and uses, while the WS is the software which achieves this interaction.

The TMN Workstation function (WSF) model [8.12] specifies the conceptual boundaries of the ICM HCI. Figure 8.1 shows the WS in the ICM system architecture.

The HCI of a software application is represented by the set of functions that allow an operator to interact with the application by using the computer resources - keyboard, screen, mouse etc. - for the purpose of operating, maintaining and supervising the system being managed. In the context of TMN, the HCI should display information in a friendly way, in order to give the operator a synthesised view of the current status, to ease the decision process, and support successful management operations.

The ICM WS model incorporates the Seeheim user interface model [8.2]. The ICM WS model both supports user function specific components and provides general support services aiding the interaction between these components and the TMN. The ICM WS architecture is shown in Figure 8.2 and its decomposition in terms of support services in Figure 8.3.

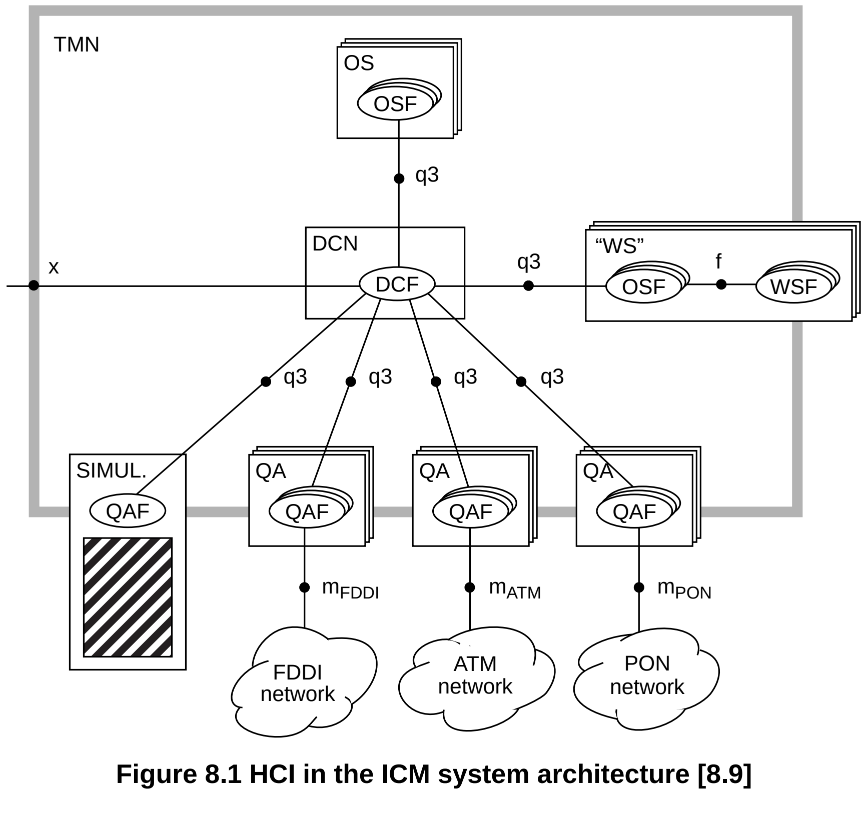


Figure 8.1 HCI in the ICM system architecture [8.9]

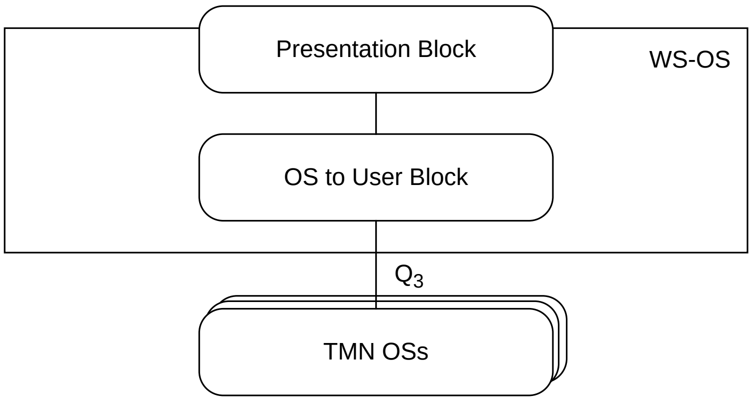


Figure 8.2 HCI architecture

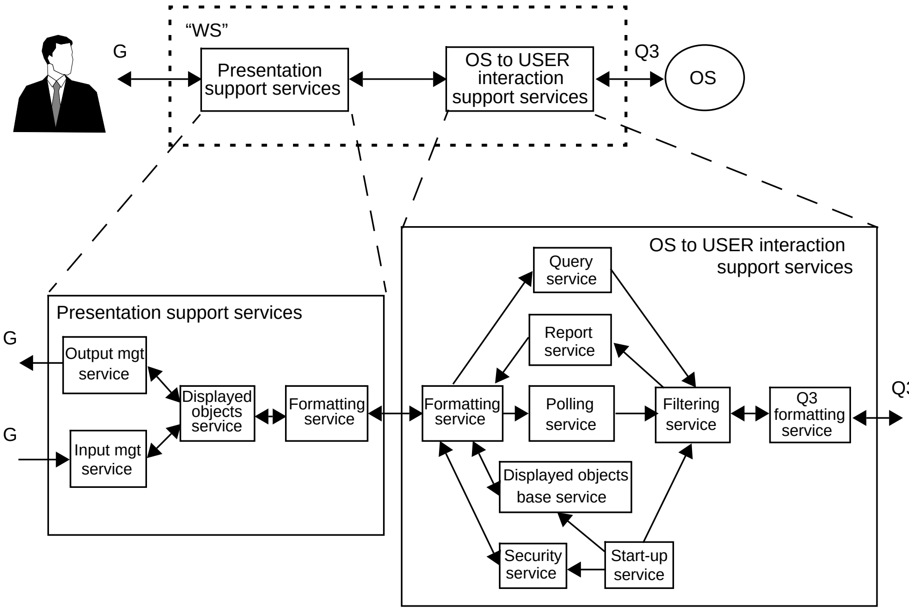


Figure 8.3 HCI support service decomposition [8.8]

As shown in Figure 8.3, the WS is divided into the following two functional blocks:

- Presentation block: encompassing the functions for interacting with the user and presenting the information held in the WS information model;
- OS to user block: encompassing the functions for interacting with TMN to retrieve the required information and the functions for processing this information in order to present it to the users.

8.3.3 Design method

Since the developed TMN is a large system, the developed WS also becomes quite large and complicated. In response to this complexity, an advanced object-oriented (OO) development methodology is used. This is of great help in development and maintenance. Apart from the modelling methodology, rational design decisions, aiming at encapsulation, code reusability and modularity, are required to make the complex system usable and likely to be developed in the future.

The point of departure of the proposed OO WS design is the set of user (operator or experimenter) requirements with respect to TMN applications and the set of general graphical requirements concerning the GUI overview, e.g., information presentation and UI look & feel. Emphasis is put on the design, so that the defined WS object classes do not impose any restructuring on the MIB object classes while being sufficiently generic to support:

- a variety of TMN functions (incorporated into the ICM testbed),
- independence between the GUI and the underlying functionality,
- the integration of new functionality or the modification of existing one.

The reason for choosing an OO viewpoint for the WS design lies with the merits of the OO methodology [8.1]. Such a viewpoint provides the means for a comprehensive decomposition of a complex system, enabling maintainability, a key issue in a TMN system's life cycle. Furthermore, OO concepts map easily on the implementation of graphical user interfaces, and most importantly, OO methodology is consistent with the way the management information is modelled within the TMN operations systems.

A characteristic of the proposed design is the clear distinction between the MOs contained in the MIB and the WS objects (needed by the HCI). Specifically, the proposed design considers the following categories of object classes:

- Managed objects (MOs): contained in the Management Information Base (MIB), representing the information which the TMN handles. These objects may include not only the objects providing a management representation of the resources (physical and logical) of the network but also the objects that are created in the various OSs in the TMN for representing the management functions.
- Displayed objects (DOs): contained in the so called Displayed Object Base (DOB); with the purpose of handling the information exchange between the users and the TMN.
- Graphical objects (GOs): representing the means through which the information will be presented to the user.

For example, the user function for viewing and browsing objects uses DOs to create GOs representing MOs in one view of the network, a sub-network or a switch, for example. A performance monitoring user function uses query and report DOs as persistent application objects, that create GOs for their set-up and for displaying the results whenever needed. Different user functions have different requirements for their view to the MOs. It must be considered for all user functions, which GOs and DOs can be reused in the appropriate WS design and implementation.

The object-oriented approach to graphical interfaces is straightforward, since the majority of today's user interface toolkits are built based on object-oriented concepts. GUIs are implemented by various presentation objects (widgets) which are concrete

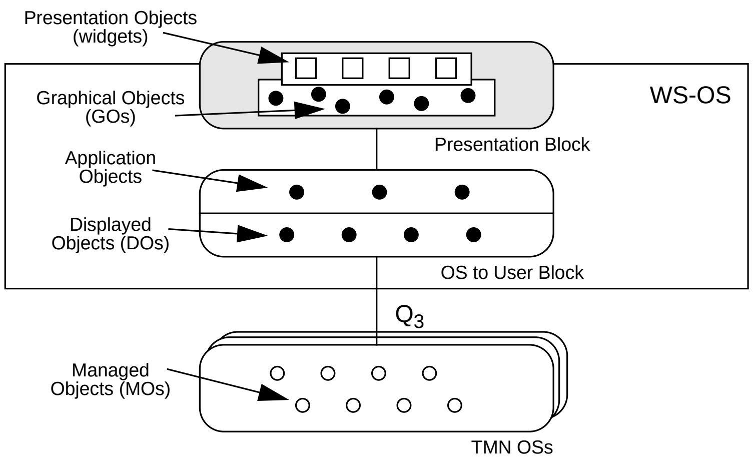


Figure 8.4 The HCI information model

objects with fixed sets of properties, which can be manipulated by the user. These objects make explicitly visible the abstractions that exist in minds of the users. The users can develop their mental models quicker via objects visible on the screen than via invisible ones.

8.3.3.1 The object model

There is either a one-to-one or one-to-many association at the conceptual level between DOs and MOs. DOs have their interface for MO information retrieval and control via the support services that embed the CMIS interface to the TMN OSs of the system. The DOs form the interaction logic front end needed by the managed objects in different user functions. The behaviour of the DOs classes encompasses:

- interaction with the appropriate MOs for retrieving the required information from the TMN and for controlling it,
- processing the information received from TMN in order to transform it to the user format, and,
- storing the transformed information.

The graphical objects (GOs) provide the means for displaying the information held in the DOs. The GOs do not contain explicit storage of information about the MOs; the information to be displayed is kept at the DOs.

In the Seeheim model for user interface [8.24], the interaction objects, that the users use to communicate with the applications, are called standard interaction primitives (SIP). The GOs defined above, correspond to the display components of the SIPs.

The GOs may use several presentation objects (widgets) through which the GUI is implemented. These objects are specific to the used presentation system.

The GOs implement the logical GUI, not the physical one, and need a consistent interface to the presentation. Therefore the GOs are associated to a Presentation Server. The Presentation Server is a special class with only one instance per windowing system. It represents the windowing (GUI management) system used, embedding its features. This way, the design becomes independent of the windowing system, and different windowing systems can be used for the interface development.

The GOs connected to a particular Presentation Server delegate the windowing operations through this server. Delegation can be implemented using varying signatures, possible in C++ programming language, for each Presentation Server member function used to manipulate different presentation objects. Thus, syntactic advantages of an OO language is used here. The UI events are received directly from the presentation objects (X Window System widgets) by the appropriate GO using the mechanism of the used windowing system.

The TMN interaction functionality which may be associated with a GO, i.e. a graphic object for starting, quitting and managing applications, is not carried out by the GOs but by specially designed intelligent objects, called application objects.

Application objects are responsible for creating the appropriate DOs – which, as said previously, interact with the MOs and format the received information. Moreover, application objects may create other GOs. Therefore, each application object corresponds to a user function and is associated with a number of DOs (keeping the required information formatted), and with a number of GOs (needed for application object's presentation).

The following paragraphs outline the basic DO and GO subclasses. Their exact definition depends on the management applications that the HCI will offer to the user.

The object model can be seen in Figure 8.5. The Booch OOD notation has been used to represent the relationships [8.1]. Main classes and their function in the model are presented in Table 8.2. The "Purpose" column describes the requirements met by the object class.

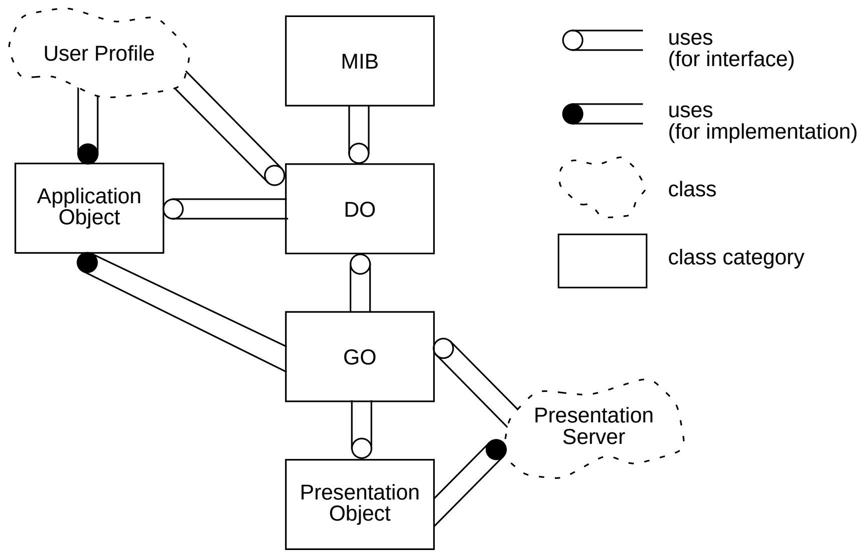


Figure 8.5 The basic HCI object model

Class or class category	Tasks	Purpose
MIB (MOs)	• CMIP/S attribute/event operations	• representation of the managed resources
DO	• interacting with MOs • information formatting • storing formatted information	• user task representation and storage, for consistent information representation
GO	• embedding GUI information and operations presentation-system-independently	• logical, consistent GUI object interface, for code reuse with different presentation systems
User profile	• start-up with correct environment	• user customisation
Application object	• initialising and quitting user functions	• start-up object, for adding new user functions consistently
Presentation server	• operating all presentation dispatching events from MOs to user functions (centralised)	• presentation-system specific HCI operating conditions, for information hiding
Presentation object	• realising the visible user interface	• look & feel consistency

Table 8.2 Basic HCI classes

DO and GO class categories are class hierarchies having each a root class inherited by subclasses specified by different user functions implemented in the system. The DO subclasses are specified by the services provided for different applications. For example, if a monitoring application makes scheduled queries to MOs, or a view application queries network elements contained in some sub-network, a query DO is needed to carry out these tasks. This way, a group of DO classes is specified according to the needs of user functions, as in Figure 8.6.

The GO class category contains more classes than the DO category. This is due to the fact that each DO needs GOs:

- for displaying its information to the user,
- for its configuration and control, and,
- for collectively representing selected information about selected or all instances of one DO class, and providing the user with a control interface on all of them.

The purpose of the GO object of the last kind is to reduce cognitive overload by combining information about many instances of one user task into one place.

The exact definition of the GO subclasses depends on the DO classes and management applications in the system. The amount of GO classes is to some extent reduced by inheriting from common super-classes.

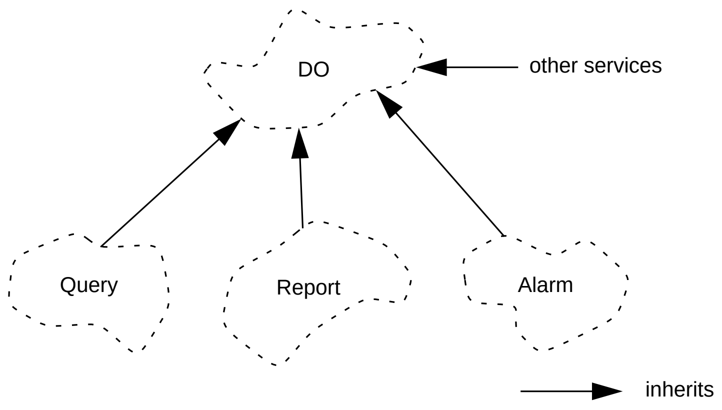


Figure 8.6 Structure of the DO class category

8.3.3.2 Features of the proposed design

The concept of Displayed Object Base contains the information from the TMN formatted to the user requirements. This way, the information available in the TMN becomes visible on the screen without associating the MOs in the MIB with any user function. The WS becomes generic in the sense that changes in the HCI do not affect the MOs in the MIB but only the DOs. Changes in the MIB affect only the DOs and not the presentation part of the WS. Considering explicitly the notion of GOs, the presentation part of the HCI may become independent of the presentation windowing system used.

The methodology used in the ICM WS design offers:

- modularity enabling modifications: by introducing clear boundaries in the behaviour of object classes in the GO, and DO categories;
- independence between the interface and the application part of the code: MOs independent of user functions and therefore changes in MIB do not affect the interface and vice versa;
- information hiding: by embedding the windowing system and making the major part of the WS independent of it;
- common infrastructure for different management applications: by refining the GO, DO classes new functionality can be introduced easily or re-used by using inheritance mechanisms;
- code reuse and reduced complexity, through class hierarchies, in sub-classing GOs into finer forms, and by using multipurpose objects in instance relations, in MO - DO - GO connections.

8.3.3.3 Workstation global architecture overview

When we try to define the WS design, it quickly becomes clear that a “global architecture” which enables the whole WS to be divided into “structured independent layers” is needed. Once this is done, we have to see which functions are performed by which layers. Finally, by listing and defining all the facilities (see Appendix A), we are able to affect each function to each layer and to specify the inter-relations between these layers.

Results of this are shown in Figure 8.7 in which applications, support services, resources and network-user are represented in horizontal layers. Facilities are (conceptual) vertical stripes that make the inter-relation between all layers.

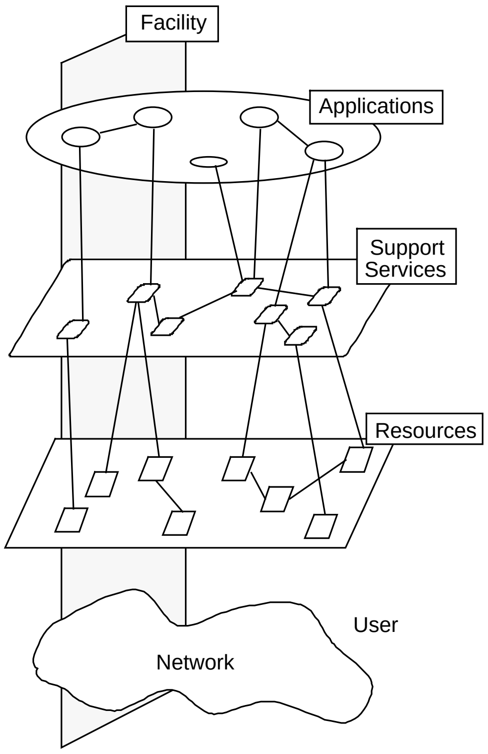


Figure 8.7 Facilities and layers

It now becomes easy to represent a specific facility and how all involved elements fit together from the user to the applications in an orderly fashion within this facility.

One facility can involve several applications, each of which can call for several support services, and so on. On the other hand, one resource can be used by several support services, one support service by several applications, etc.

The platform support services are basic components that can be manipulated by management applications to use resources. For instance, the storage service allows an application to record data on a resource.

Resources are a complete set of hardware/firmware components. For instance, keyboard is a resource to generate ASCII code. Its associated implicit service is an ASCII table (corresponding to the driver) which is bridged to an equal display service.

Figure 8.8 describes the approach. In Figure 8.9, call arrows correspond to:

- a user can call an application, or be called by an application (user authentication),
- applications can call other applications or multi-media services,
- multi-media services can call resources (using drivers).

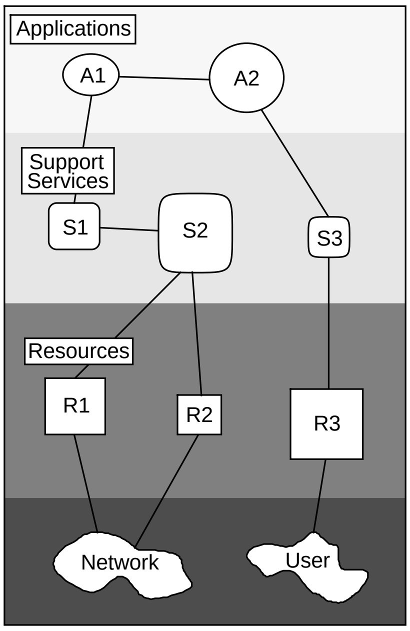


Figure 8.8 Internal structure of a facility

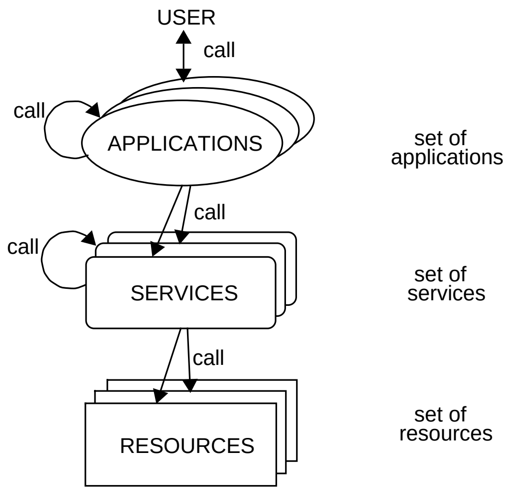


Figure 8.9 Software platform relations

In order to build a software platform where it is possible to easily and quickly implement new management applications, the software platform architecture must have the following characteristics:

- It must be easy to add new resources, with a minimum of constraint on the application level. For instance, changing a storage resource (e.g. floppy disk to hard disk) must be transparent to the application.
- The application level does not care how the service is performed. For instance, it is very easy to have new version of services (upgrade), without impacting the application level.

A correct definition of service access (APIs - Application Programming Interfaces) must provide a high level of independence between applications and services (and resources). So, services must provide APIs to the Application level to guarantee independence.

Figure 8.10 describes the functional specification of a service:

Service are composed of APIs, functions and drivers. Applications access the service through the API. Services access to resources through driver. The way services perform functions is totally invisible to the application.

Appendix A lists the set of support services that have been defined and implemented for the ICM WS.

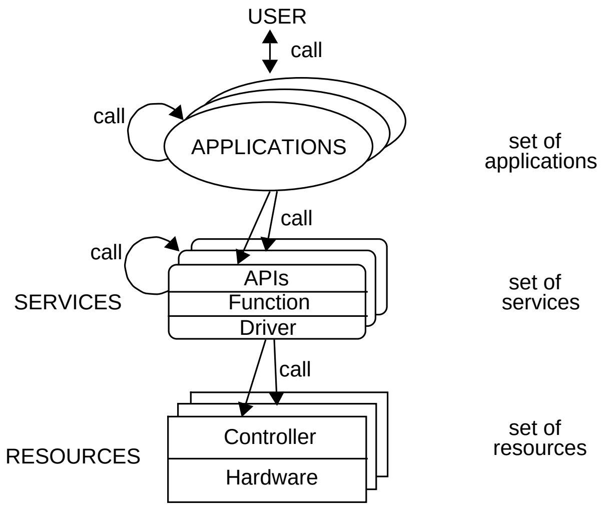


Figure 8.10 Service functional relations

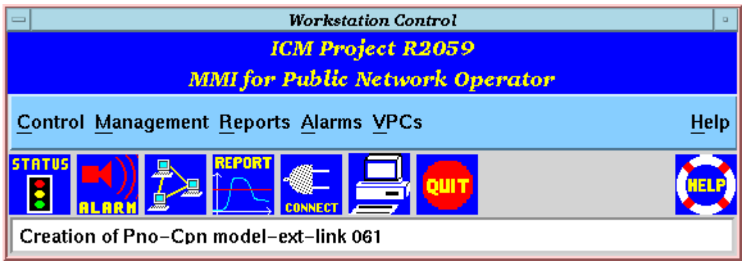
8.4 Experience in designing and implementing HCIs for IBC network management

8.4.1 The Public Network Operator workstation

The Public Network Operator (PNO) Workstation is the interface between the operator and the TMN through which the public network is managed. This interface is very critical as it is the only exchange point between the human network manager and the complex TMN system managing the network. It is thus very important to offer the operator all the necessary facilities to access information on the configuration and status of the network. The following sections show examples of the screens designed to provide the operator with these facilities.

8.4.1.1 The workstation control screen

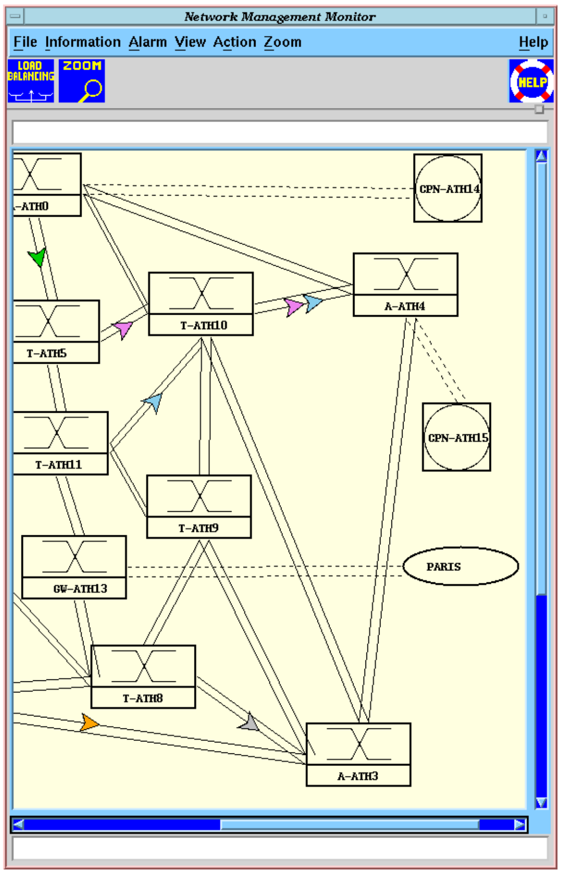
This application (Figure 8.11) gives an access to all the other applications available to the user according to his profile. From the menus and/or icons of this screen, the operator has an access to all the windows of the Workstation. The windows have been classified according to the kind of services offered:



Creation of Pno-Cpn model-ext-link 061

Figure 8.11 Workstation control screen

- Control: used to manage the Workstation facilities (screen, mouse, etc.),
- Management: network management facilities. This menu provides an access to the connection manager application, to the management monitor application and to the status monitor application,
- Reports: reports management facilities. Access to the report manager, to the query manager and to the query editor,
- Alarms: alarms management facilities for access to the monitoring and reporting applications,
- VPCs (available only with VPC management applications): VPC management facilities. Class of service model application and VPC required bandwidth manager application.



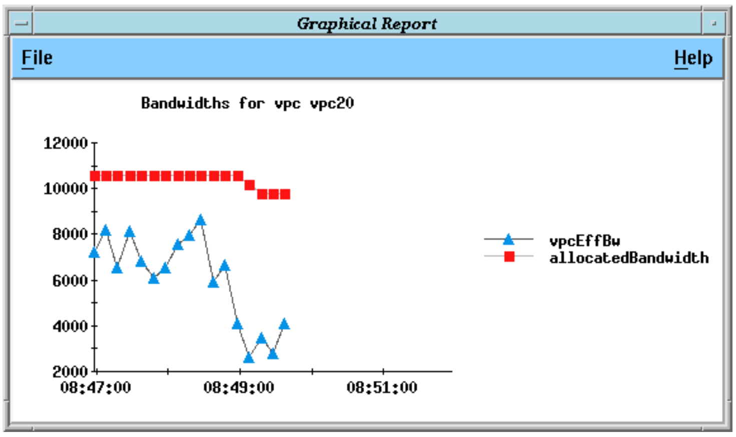


Figure 8.12 Network management monitor screen and graphical report screen

8.4.1.2 The network management monitoring screen

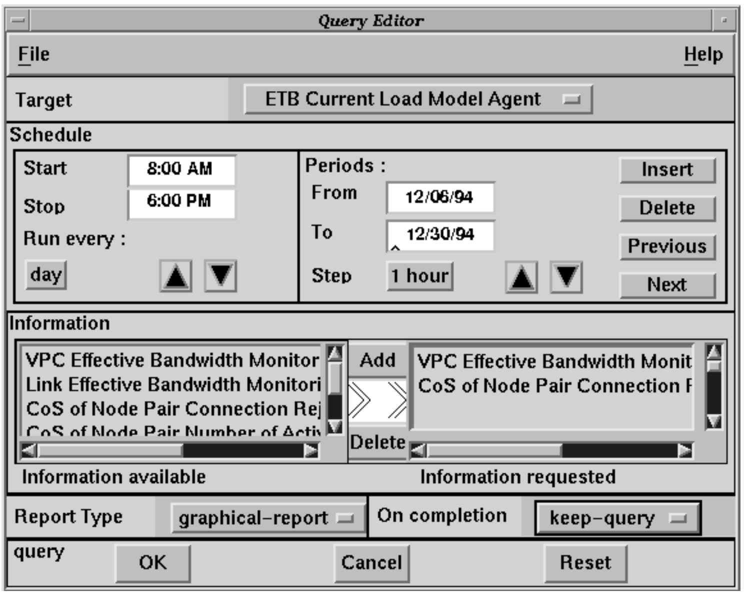
This screen (Figure 8.12) is the most important one in the PNO Workstation. It has been designed to be the main window of the operator Workstation. Most static and dynamic information on the network are either represented or can be obtained from this window. It offers a view of the network with switches, links and VPCs (when running the VPC management application). Dynamic information can be obtained on all displayed items by clicking on them (switches, links, VPCs). Those information can be obtained as arrays, with all the information on the selected items, or as graphs where the evolution of some selected parameters is displayed. Such a graph is represented in Figure 8.12.

The menus available on this window are the following:

- File: print the current view, or close the window,
- Information: to select the types of alarms to be displayed or to obtain information on bandwidths,
- View: to select the displayed VPCs,
- Zoom: to select the zoom factor.

8.4.1.3 The query editor screen

This screen (Figure 8.13) allows the user to build queries on the network. A query consists in the measure of one or several parameters on a given period of time. Once the query has been designed, it can be run by the operator. A report is then built with the


Query Editor

File | Help

Target: ETB Current Load Model Agent

Schedule
Start: 8:00 AM
Stop: 6:00 PM
Run every : day
Periods :
From: 12/06/94
To: 12/30/94
Step: 1 hour

Information
Information available:
VPC Effective Bandwidth Monitor
Link Effective Bandwidth Monitori
CoS of Node Pair Connection Rej
CoS of Node Pair Number of Activ

Information requested:
VPC Effective Bandwidth Monit
CoS of Node Pair Connection F

Report Type: graphical-report
On completion: keep-query

Figure 8.13 The query editor screen

values of the parameters requested in the query. The figure shows a query with the following characteristics:

- two parameters are examined: VPC Effective bandwidth and CoS node pair connection rejection,
- the first measure is on the 6th December, the last on the 30th December, and the query is run every day,
- the measures are done every hour between 8:00 am and 6:00 pm,
- the report built is a graphical report (but may optionally be textual).

Figure 8.14 shows an example of graphical report of the measures obtained from the runs of a query.

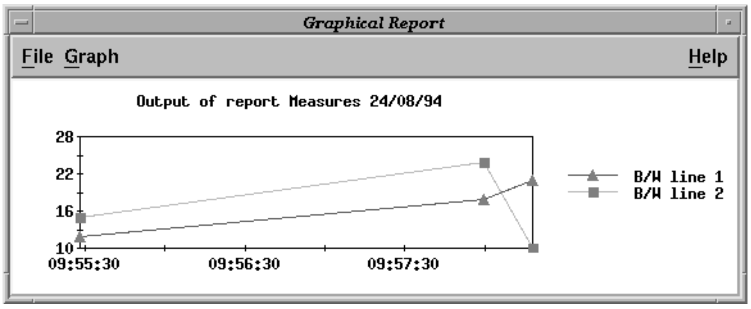


Figure 8.14 Graphical report of measures

8.4.2 The VASP WS-OS

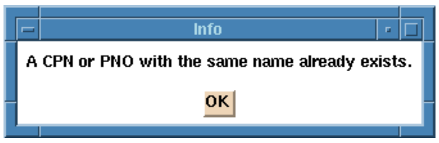
The intermediate Virtual Private Network (iVPN) Value-Added Service Provider (VASP) management system, described in Chapter 6, has some specific properties in its Workstation Operations System (WS-OS). The iVPN VASP tasks are characterised by their infrequency and by a direct connection to customers. Memorability and error prevention and tolerance are emphasised in the WS-OS for the first reason, efficiency and again error prevention and tolerance for the second reason. Since one purpose of the VASP WS-OS implementation is to demonstrate VPN creation, maintenance and manipulation, learnability is also of great importance. Task-specific requirements are distributability, security requirements and feedback from the OSs managed recursively by the WS-OS.

The usability attributes learnability, memorability, efficiency, error prevention and tolerance are realised by meeting a number of the requirements presented in Section 8.2.1.

Learnability is important for demonstrability, to give the first impression to the audience. In the iVPN VASP WS-OS, it is enabled by the means below.

- The UI is modeless, which is natural to a GUI. The iVPN VASP WS-OS is to some extent also easy to use for a beginner without domain expertise. An example of the WS-OS use is presented in Section 6.3.9.

- Dialogue and messages use natural language. The vocabulary is specialised, due to the audience expertise. An example pop-up is shown in Figure 8.15.



A CPN or PNO with the same name already exists.

Figure 8.15 An error message for the VASP operator

- There is an on-line help system with embedded hyperlinks, as in Figure 8.16, to enable quick information acquisition.
- The user organisation has the possibility to affect the terminology. This is done by editing a configuration file with a simple dictionary for the essential terms, or a usual X Window System application configuration file for any texts presented in the HCI.
- Different networks are presented as networks in the user interface. Backgrounds, nodes, edges, selections, node and edge labelling and topology editing capabilities are available to provide the user the required control over the presentation. Examples of networks are in Figure 6.18 - Figure 6.20.

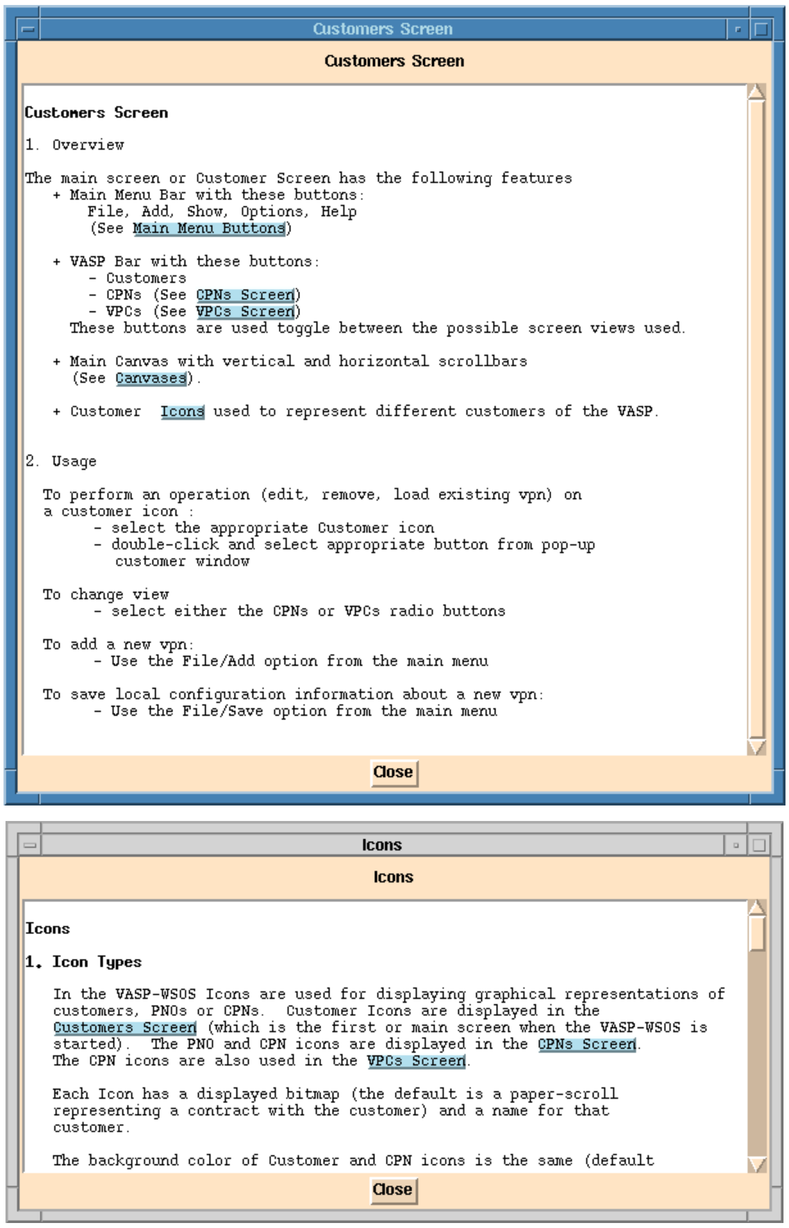
The tasks performed using iVPN VASP WS-OS are infrequent and thus the HCI must help the user to memorise at least their initial steps. There are two features enabling memorability:

- Standardisation is achieved using the Motif look and feel [8.21] implemented by the Tk toolkit [8.22]. The result is described in Section 6.3.9.
- Attractors are those used to represent VASP objects in the HCI. They are easy to remember and aesthetic. In the iVPN VASP WS-OS, these are limited to iconic representations on the networks (see Chapter 6: Figure 6.19 and Figure 6.20).

Efficiency is important for the iVPN VASP tasks, because configuration tasks, e.g., VPN formation, give the customers of the VASP operator the first impression of the VASP. Two features are implemented solely to facilitate efficiency:

- Common HCI and internal commonality of the interface components enable routine in iVPN management. An example of this in the iVPN VASP WS-OS is common terminology, accelerators and commands used in the forms of Figure 8.17.
- Selection input, as in Figure 8.18, allows the user to avoid typing in information on keyboard. The information about available choices is retrieved from and must be validated by the OSs in an agent role to the iVPN VASP WS-OS.

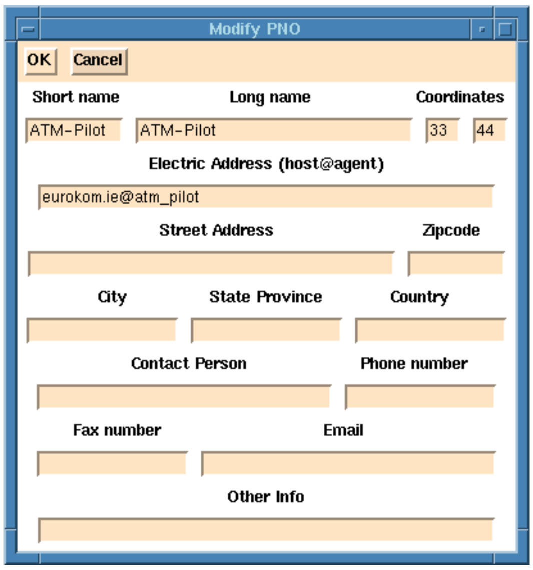
The error tolerance of the WS-OS is acceptable in the sense that problems in other OSs will not develop into fatal errors in the WS-OS. Since no fault or performance monitoring is done in the iVPN, event correlation is not needed in the WS-OS. The Tk toolkit [8.22] provides a good shelter against usage errors. However, due to the inter-OS communication required, it is not easy to implement means for user-initiated undo or error recovery.



```
Customers Screen

1. Overview

The main screen or Customer Screen has the following features
   + Main Menu Bar with these buttons:
       File, Add, Show, Options, Help
       (See Main Menu Buttons)

   + VASP Bar with these buttons:
       - Customers
       - CPNs (See CPNs Screen)
       - VPCs (See VPCs Screen)
     These buttons are used toggle between the possible screen views used.

   + Main Canvas with vertical and horizontal scrollbars
     (See Canvases).

   + Customer  Icons used to represent different customers of the VASP.


2. Usage

  To perform an operation (edit, remove, load existing vpn) on
  a customer icon :
        - select the appropriate Customer icon
        - double-click and select appropriate button from pop-up
          customer window

  To change view
        - select either the CPNs or VPCs radio buttons

  To add a new vpn:
        - Use the File/Add option from the main menu

  To save local configuration information about a new vpn:
        - Use the File/Save option from the main menu
```

```
Icons

1. Icon Types

   In the VASP-WSOS Icons are used for displaying graphical representations of
   customers, PNOs or CPNs.  Customer Icons are displayed in the
   Customers Screen (which is the first or main screen when the VASP-WSOS is
   started).  The PNO and CPN icons are displayed in the CPNs Screen.
   The CPN icons are also used in the VPCs Screen.

   Each Icon has a displayed bitmap (the default is a paper-scroll
   representing a contract with the customer) and a name for that
   customer.

   The background color of Customer and CPN icons is the same (default
```

Figure 8.16 Windows in the iVPN VASP WS-OS on-line help system



Short name	Long name	Coordinates	
ATM-Pilot	ATM-Pilot	33	44

Electric Address (host@agent)
eurokom.ie@atm_pilot

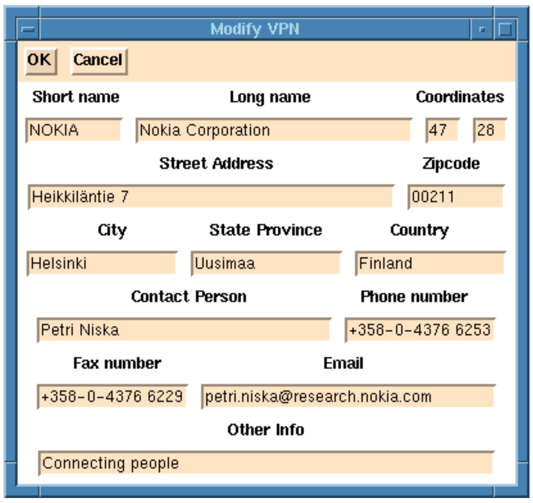
Street Address	Zipcode

City	State Province	Country

Contact Person	Phone number

Fax number	Email

Other Info



Short name	Long name	Coordinates	
NOKIA	Nokia Corporation	47	28

Street Address	Zipcode
Heikkiläntie 7	00211

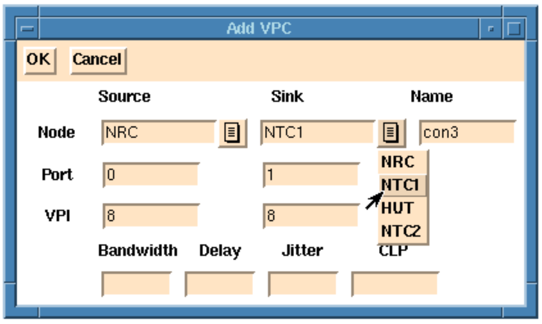
City	State Province	Country
Helsinki	Uusimaa	Finland

Contact Person	Phone number
Petri Niska	+358-0-4376 6253

Fax number	Email
+358-0-4376 6229	petri.niska@research.nokia.com

Other Info
Connecting people

Figure 8.17 Modification forms with common terminology, accelerators and commands



	Source	Sink	Name
Node	NRC	NTC1	con3
Port	0	1	
VPI	8	8	

Figure 8.18 Selection input when adding a VPC

8.5 Implementation of the SimMMI

8.5.1 Design approach

The SimMMI has been designed and implemented following a functional approach. However, the use of GUI tools that are implemented by various presentation objects (widgets) and various triggering mechanisms (callback procedure calls and event handlers) has forced to use object-oriented principles.

Widgets are concrete, user manipulated objects with fixed number of properties. ATM network physical resources (nodes, links) and virtual resources (VPCs) have been visualised using such objects. User defined widget resources and widget resources access mechanisms have been used to model the main application objects (e.g. networks, nodes, links etc.).

A client-server model was adopted for the design of the running mode of the SimMMI. The SimMMI acts both as a client requesting process execution (simulator and OSs) and as a server accepting requests from clients (simulator or other process).

The design of the system has been influenced from the different styles of its usage. Four different usage modes have been identified. Figure 8.19 shows the main system design activities.

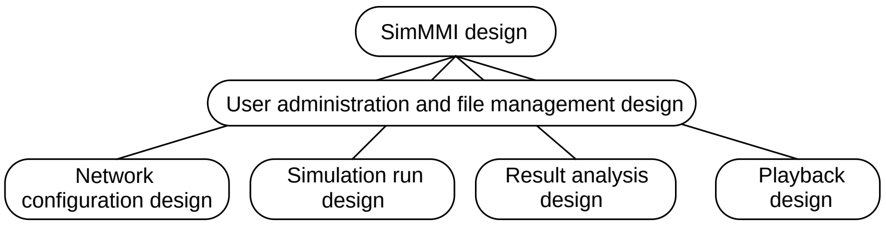

SimMMI design
User administration and file management design
Network configuration design
Simulation run design
Result analysis design
Playback design

Figure 8.19 SimMMI main design activities

8.5.2 SimMMI tool screens

Figure 8.20 is a screen dump of the SimMMI configuration mode window. In this window there is a clear distinction of the network views:

- The *topological view* where the user creates and configures the physical resources of the network (nodes and links).
- The *logical view* where the user creates and configures virtual resources of the network (VPCs and routes).

The SimMMI supports both single and multiple domain network creation and configuration. The user can switch to the following network representations:

- *Network representation*. All the networks on the screen are represented as single objects interconnected by links in the topological view and with virtual paths in logical view.
- *Node representation*. All the networks on the screen are represented as a set of nodes interconnected by links in the topological view and with virtual paths in logical view.
- *Gateway representation*. This representation is the same with the 'node representation' with the difference that only the gateway nodes are displayed.

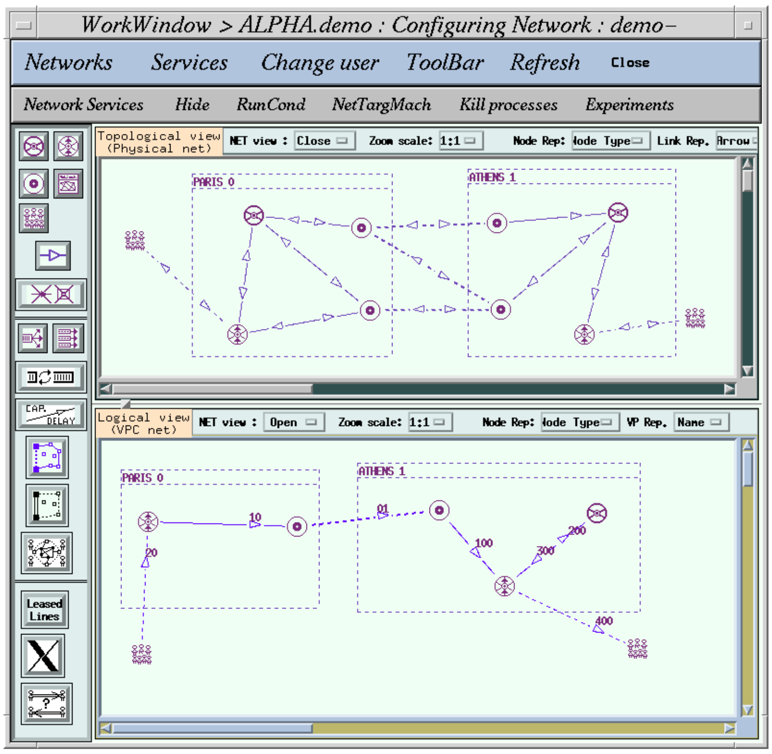


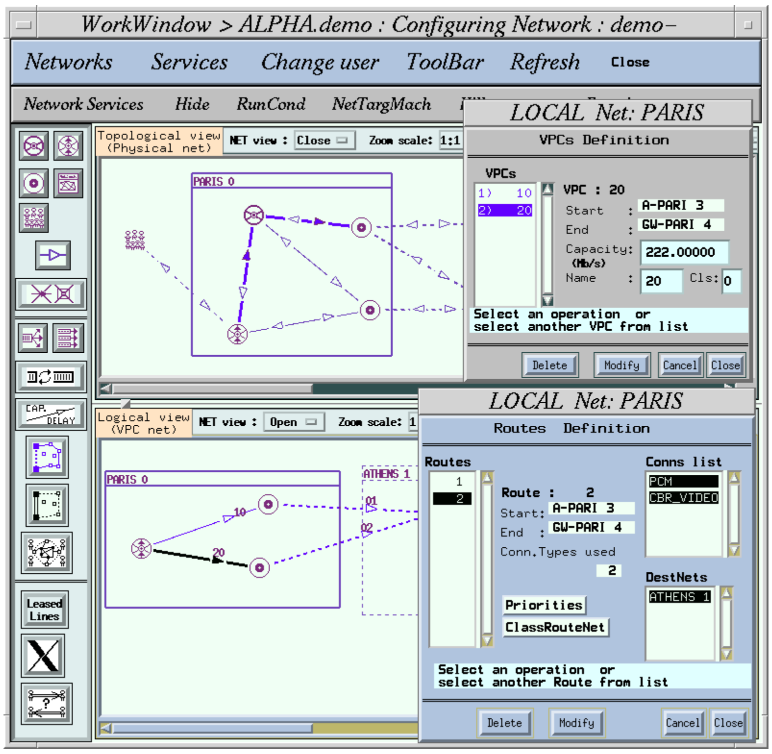
Figure 8.20 Configuration mode of the SimMMI

In each of the above network representations, the user can switch the node representation to:

- *Node type* representation,
- *Switch architecture* representation, showing the internal structure of the switch,
- *Name* representation, showing user defined node and link /VP names.

Figure 8.21 shows how the user is prompted to create VPCs and routes for a specific network domain. The appropriate tools have been called for the 'PARIS' network domain.

- *VPCs definition dialogue*. To define a VPC the user repeatedly clicks on arrows representing the links in the topological view. Each link selected is added to the VPC definition and is painted with the colour used to display VPCs. Using the displayed dialogue, the user completes the VPC definition. When the user finishes a VPC definition, the source node, the destination node and a line with an arrow representing the VPC is created in the logical view of the network.
- *Route definition dialogue*. To define a Route the user repeatedly clicks on arrows representing the VPCs in the logical view. Each VPC selected is added to the route definition and is painted with the colour used to display routes. Using the displayed dialogue the user completes the route definition. In the



1) 10
2) 20

VPC : 20
Start : A-PARI 3
End : GW-PARI 4
Capacity: (Mb/s) 222.00000
Name : 20 Cls: 0

Select an operation or
select another VPC from list

1
2

Route : 2
Start: A-PARI 3
End : GW-PARI 4
Conn.Types used 2

PCM
CBR_VIDEO

ATHENS 1

Select an operation or
select another Route from list

Figure 8.21 Configuration mode of the SimMMI for VPC and route creation

example shown below, the user has specified that the route under definition is used to serve services requiring PCM and CBR_VIDEO connection types and that this route has destination the 'ATHENS' network domain.

The button ClassRouteNet, shown in the 'Route definition' dialogue, gives to the user the Class Route Network view. That is, for each connection type and for each source - destination pair all the routes between them that are used by the connection type.

Figure 8.22 is a screen dump of the SimMMI running mode window. The SimMMI receives from the simulator and displays to the user:

- *the buffers in switches that have started losing and continuously loose cells*. This is indicated to the user by colouring in red a link that is attached to the buffer;

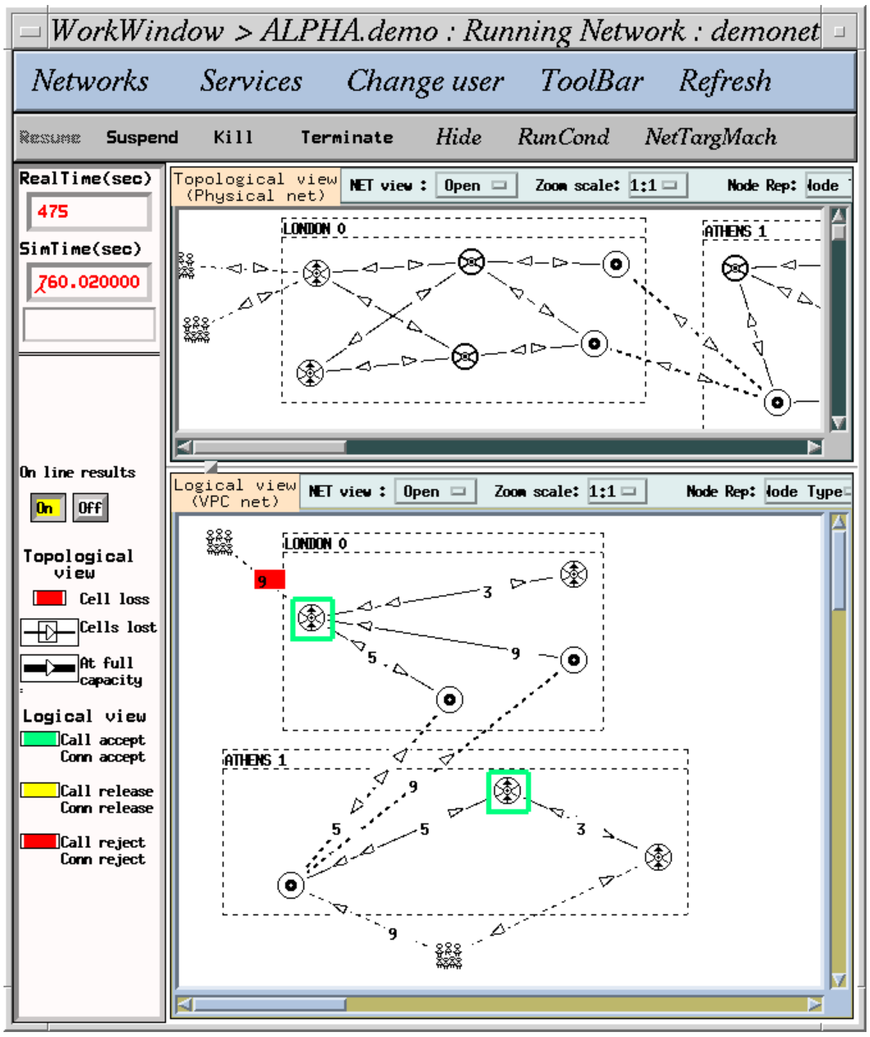


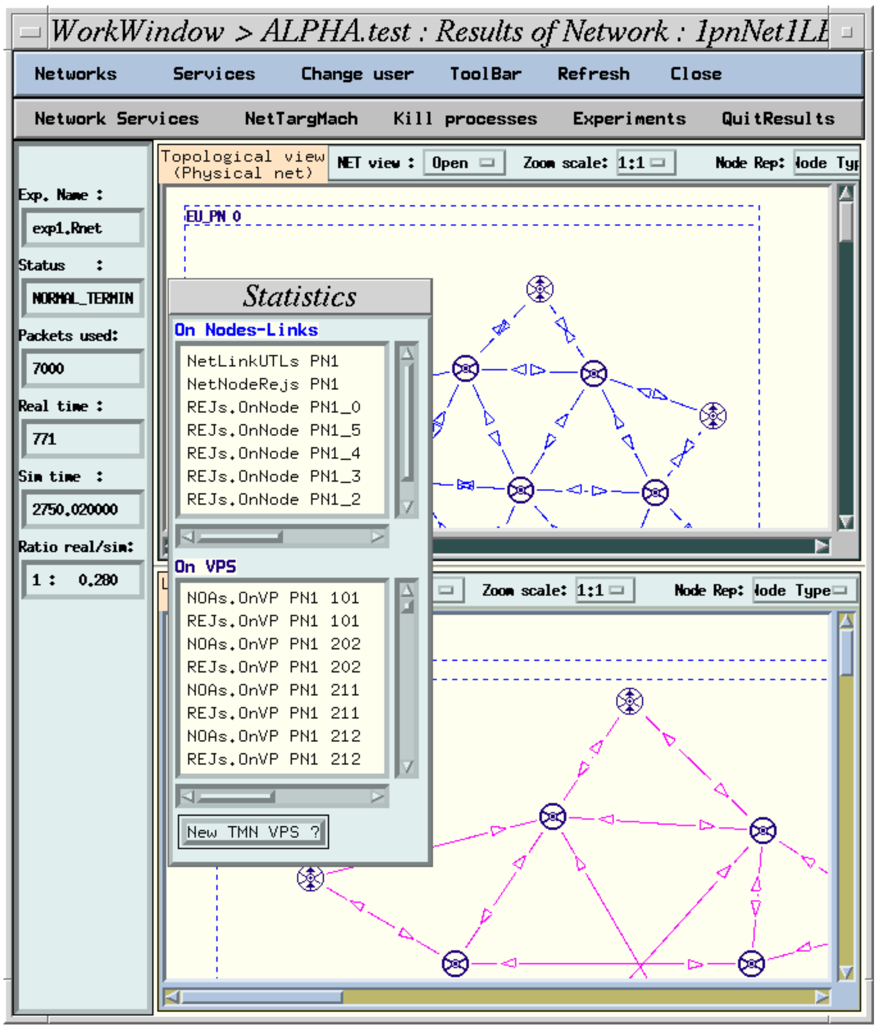
Figure 8.22 Running mode of the SimMMI

- *the buffers in switches that had lost cells at least once in the past*. This is indicated to the user by a small box around the arrow of a link attached to the buffer;
- the links that transmit at full capacity. This is indicated to the user by changing the width of the line representing the link;
- *Node call accept/release/ reject indications*. This is indicated to the user by painting in green/yellow/red the source and destination nodes;
- *VPC connection accept/release/reject indications*. This is indicated to the user by painting in green/yellow/red the arrows representing the VPCs. In addition the number of active connections on each VPC is displayed on the VPCs arrow position.

Figure 8.23 is a screen dump of the SimMMI result mode window. The user can make many simulation runs of the same network storing in files the results of each simulation run. At any time, the results of a specific simulation run can be loaded.

In this window, the system displays some simulation run specific information and offers to the user a list of network resources for which there are available statistics.

The user selects items from the list of the statistics and gets a graph representation of them. In Figure 8.24, examples of such graphs are presented.



Exp. Name : exp1.Rnet
Status : NORMAL_TERMIN
Packets used: 7000
Real time : 771
Sim time : 2750.020000
Ratio real/sim: 1 : 0.280

NET view : Open
Zoom scale: 1:1
Node Rep: Node Type

```
NetLinkUTLs PN1
NetNodeRejs PN1
REJs.OnNode PN1_0
REJs.OnNode PN1_5
REJs.OnNode PN1_4
REJs.OnNode PN1_3
REJs.OnNode PN1_2
```

```
NOAs.OnVP PN1 101
REJs.OnVP PN1 101
NOAs.OnVP PN1 202
REJs.OnVP PN1 202
NOAs.OnVP PN1 211
REJs.OnVP PN1 211
NOAs.OnVP PN1 212
REJs.OnVP PN1 212
```

Figure 8.23 Result mode of the SimMMI

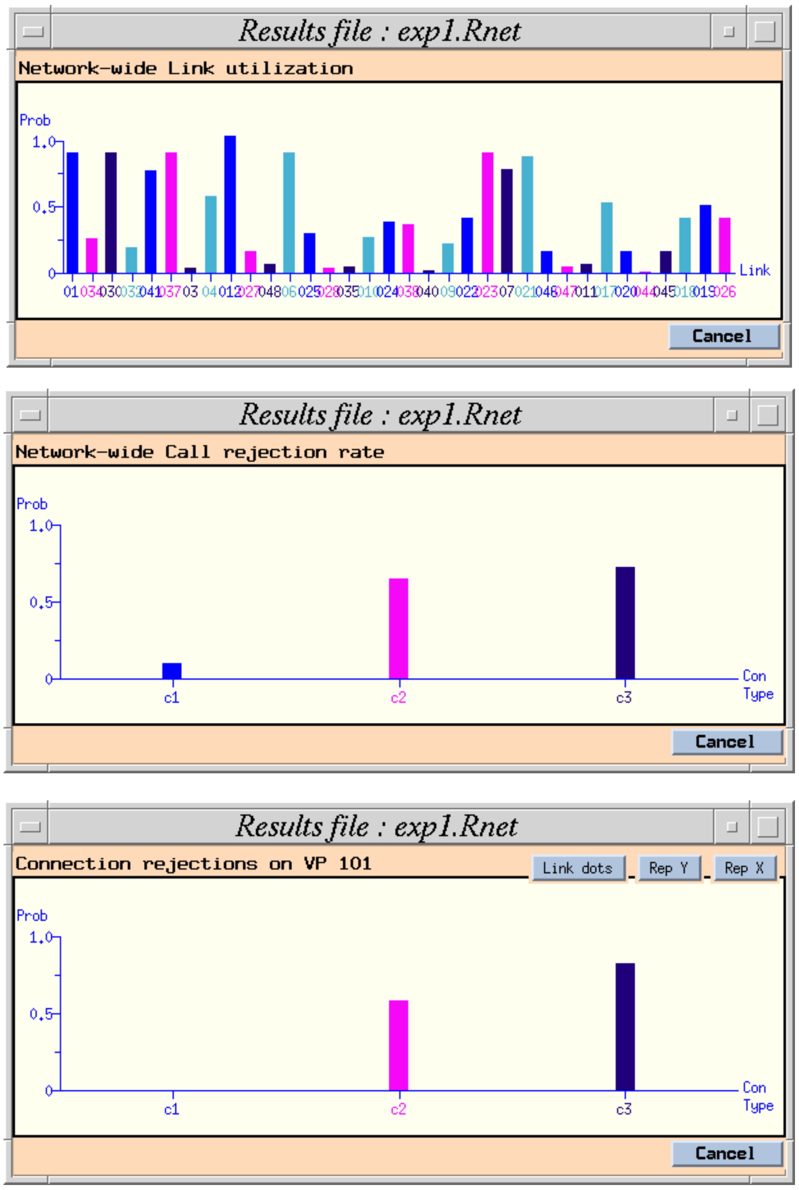


Figure 8.24 Graphical representation of network resource utilisation

8.6 References

[8.1] G. Booch, “Object-Oriented Design with Applications,” the Benjamin/Cummings Publishing Company, 1991.

[8.2] RACE R2059 ICM Deliverable 2, “Initial TMN Architecture, Functions and Design Approaches,” R2059/CRA/ATG/DS/R/003/b1, Andy Carr, editor, December 1992.

[8.3] CCITT Draft Recommendation M.3300, “TMN Management capabilities presented at the F interface,” November 1991.

[8.4] CCITT - Draft Recommendation M.3010, “Principles for a Telecommunications Management Network,” November 1991.

[8.5] O. Chambon, X. Pandolfi, “ICM Workstation - User Management Capabilities for the VPC Management Case Study,” Version 1.0, April 1994.

[8.6] ISO/IS 9595, “Information Technology - Open Systems Interconnection - Common Management Information Protocol Specification,” Version 2. July 1991.

[8.7] RACE R2059 ICM Deliverable 14, “ICM Case Studies,” R2059/QMW/BM1/DS/P/014/b1, Babul Miah, editor, September 1994.

[8.8] RACE R2059 ICM Deliverable 3, “Selection of Networks, Network Interfaces and Definition of Testbed Laboratory,” R2059/VTT/TEL/DS/R/004/b1, Jim Reilly, Juha Koivisto, editors, December 1992.

[8.9] RACE R2059 ICM Deliverable 5, “Revised TMN Architecture, Functions and Case Studies,” R2059/ICS/DPG/DS/P/007/b1, David Griffin, editor, September 1993.

[8.10] CCITT Recommendation X.722 | ISO/IEC 10165-4:1992, “Information technology - Open Systems Interconnection - Structure of management information: Guidelines for the definition of managed objects,” 1992.

[8.11] RACE R1003 Guideline Main Event 6, “The Application and Integration of AIP Techniques within the RACE TMN,” 1991.

[8.12] ACE R1003 Guideline Main Event 8, “TMN implementation architecture,” March 1992.

[8.13] ISO/IS 10164-4, “Information Technology - Open Systems Interconnection - Systems Management - Part 4: Alarm Reporting Function,” August 1991.

[8.14] ISO/IS 10164-6, “Information Technology - Open Systems Interconnection - Systems Management - Part 6: Log Control Function,” August 1991.

[8.15] P. Jussila, “Efficient management of broadband telecommunications network,” in proceedings of Broadband Islands ‘95, Sept. 4-6, 1995.

[8.16] N. Mandich, T. Belleli, “HCI considerations in TMN systems,” in The Management of Telecommunications Networks, Smith, Mamdani, Callaghan (editors), Ellis Horwood, 1992, pp. 251-260.

[8.17] A. Marcus, “Graphic Design for Electronic Documents and User Interfaces,” ACM Press. 1992.

[8.18] David E. Meyer, Sylvian Kornblum (editors), “Attention and Performance XIV (Synergies in Experimental Psychology, Artificial Intelligence, and Cognitive Neuroscience),” MIT Press, 1993.

[8.19] W. M. Newman, R. E. Sproull, “Principles of Interactive Computer Graphics,” McGraw-Hill, 1981.

[8.20] J. Nielsen, “Usability Engineering,” Academic Press, 1993.

[8.21] Open Software Foundation, “OSF/Motif Style Guide,” Revision 1.2, Prentice Hall. 1993.

[8.22] J. K. Ousterhout, “Tcl and the Tk Toolkit,” Addison-Wesley, 1994.

[8.23] G. Pavlou, “The OSI Management Information Service,” User's Manual, Version 1.0, for system version 3.0, 1993.

[8.24] G. E. Pfaff, “User Interface Management Systems,” in proceedings of the Workshop on UIMS held in Seeheim, Federal Republic of Germany, November 1-3 1983, Springer-Verlag. 1983.

[8.25] “Intelligent User Interfaces,” by Joseph W. Sullivan and Sherman W. Tyler, ACM, Press, Addison-Wesley, 1991.