ISBN 0-315-56463-6

Image Organization and Navigation Strategies for a Radiological Workstation

by
Etienne Roger

A THESIS
submitted to the School of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
M.A.Sc.
in
Electrical Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
OTTAWA , ONTARIO K1N 6N5

November, 1988

ABSTRACT

An important component of a radiologist's workload is the "reading" of radiographs and the accompanying reporting process which normally involves dictation and verification of the subsequent transcription. A subset of the patient radiographic file is chosen by the radiologist and arranged on viewboxes (typically a set of 8 viewboxes arranged in two rows of 4). The radiologist reads this subset of films and reports his findings in the context of the accompanying requisition statement. One can distinguish two distinct preliminary phases in the "analog" reading process. The first phase deals with the film selection and placement on the viewboxes (image organization), and the second with finding specific images (involving image navigation).

The object of the present thesis is to investigate *image organization* and *image navigation* issues of the user interface of a given multimedia workstation. The image organization issue concerns the classification, sorting and presentation of the radiographic image content of a patient file. The image navigation issue refers to the means of accessing these images.

ACKNOWLEDGMENT

I wish to express my gratitude and appreciation to my thesis supervisor, Dr. Morris Goldberg. No amount of words would suffice to thank him for his constant guidance, encouragement and support which he has generously given me during my research program. Dr. David Goodenough was my co-supervisor and also provided financial support.

I would like to thank the members of the MUSIC project of the Medical Communications Research Centre for their help and support. Special thanks are given to the psychologists from Carleton University, Dr. Tombaugh, Dr. Dillon, Marjorie Coristine and Cynthia Houston for their considerable assistance in the preparation, realization and evaluation of the field tests.

Finally, I would like to acknowledge the financial supports from OGS, NSERC and FCAR grants.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Hospital operations generate, for each patient, a large number of data sets of different types and formats, including handwritten requests, typed forms, computer based data, voice messages/reports, and diagnostic information such as electrocardiograms and images. The need exists for both individual and shared access to these distributed data sets from different locations in the hospital. Examples of the former are a radiologist making a voice report and a nurse or physician reading and updating portions of a patient file. Examples of the latter are consultations between a nurse and a physician, and between an attending physician and a radiologist; in both cases the need arises for shared access and communication of distributed patient data. Of particular importance is the timely transfer of diagnostic images and radiological reports, together with the patient files, throughout the hospital. These multimedia documents containing voice, image and text segments need to be shared during consultation sessions between an attending physician and a radiologist.

The rapid communication of diagnostic information together with the patient files will lead to improved patient care and can reduce the length of time spent in the hospital by the patient. Both factors are important incentives for the introduction of a multimedia communication system.

The multimedia workstations are the end-points of the multimedia communication system. The workstations provide means for creating, editing and consulting multimedia documents. In the context of radiology, the image aspect of the workstation is of extreme importance. Indeed, the radiographic films reading process is the basis of the report produced thereafter by radiologists. Hence the workstation must provide, at reasonable cost, adequate means for viewing the images, both in terms of architecture and user interface. The adequacy of the architecture is measured by factors, such as display resolution and image access time. The adequacy of the user interface is more difficult to measure as it is subject to numerous human factors [FOLEY, 84] [SHNEIDERMAN, 87].

The object of the present thesis is to investigate *image organization* and *image navigation* issues of the user interface of a given multimedia workstation. The image organization issue concerns the classification, sorting and presentation of the radiographic image content of a patient file. The image navigation issue refers to the means of accessing these images.

The current research was carried out in the context of the MUSIC (MUltimedia System for Integrated Communications) project, a multi-disciplinary (network, multimedia database, image processing, user interface) research effort of the University of Ottawa Medical Communications Centre.

The thesis is organized as follows. We first review in chapter 2 some user interface concepts; *interaction styles, architectures, models and design tools.* Then, in chapter 3, we present a review of Picture Archiving and Communication Systems (PACSs), followed in chapter 4 by a presentation of object-oriented programming languages in general and Smalltalk/V in particular. Next, the design process leading to our image organization and navigation strategies is elaborated in chapter 5. The actual implementation of these user interface strategies, using the Smalltalk/V language, is presented in chapter 6. Finally in Chapter 7 we describe the field tests which were conducted at the Ottawa Civic Hospital and present the results thereof.

Bibliography

[FOLEY, 84] J.D. Foley, V.L. Wallace and P. Chan, "The Human Factors of Computer Graphics Interaction Techniques", *IEEE Computer Graphics and Applications,* pp. 13-48, Nov. 1984.

[SHNEIDERMAN, 87] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Adison-Wesley, 448p., 1987.

Chapter 2

User Interfaces

2.1 Introduction

The term "user interface", also known as *man-machine interface* or *end-user facility*, refers to the software and hardware which mediates [FISCHER, 87] between the user and the applications provided by a computer system (Fig. 2.1). User interfaces are interactive systems which support two-way communications in terms of dialogue translation and data presentation.

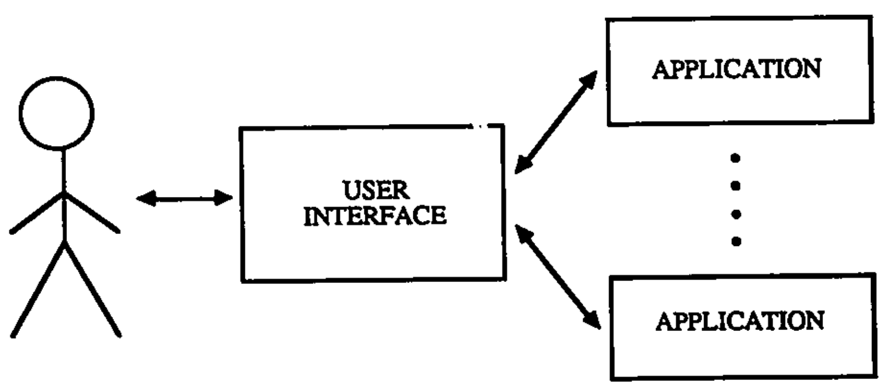


Figure 2.1: The User Interface.

In this chapter we first describe some of the methods or styles in which to carry the human-computer interaction. Then, we review some techniques for designing user interfaces. Finally, we present an architecture for user interfaces.

2.2 Interaction Styles

There exists a variety of methods or styles in which to carry on a human-computer interaction. The primary interaction styles [SHNEIDERMAN, 87] are: command language, form fill-in, menu selection, direct manipulation and natural language. With command languages, users instruct the computer by typing sentences in a well defined and specific vocabulary and grammar (e.g. MS-DOS on IBM's Personal Computer). With form fill-in, the user sees a display of fields, moves a cursor among the fields, and enters data where desired (e.g. the control panel on Apple's Macintosh). With a menu-oriented style of interaction the user instructs the computer by making selections from lists of items: at any instant of the human-computer dialogue all alternatives are presented to the user (e.g. pull-down menus on the Macintosh). Direct manipulation refers to a user interface where the world of action is represented and manipulated visually rather than syntactically (e.g. file icons on the Macintosh). Natural language interaction enables users to instruct the computer using human language (e.g. French, English, etc.).

User interfaces are not restricted to single interaction style: complex applications' needs may be better served by a combination of a number of styles. The appropriateness of any given interaction style depends on a number of factors and the selection of a style versus another is part of the user interface design process.

2.3 User Interface Design

User interface design begins with a careful study of the intended users of the computer system (user model) and of the way they currently (before the introduction of the computer system) perform their tasks (user task model). The purpose of this study is to obtain information about the objectives, knowledge, representations and working methodology of the users [SCAPIN, 86]. The user model attempts to profile variables such as education, training, goals, motivation and previous computer experience. These factors come into play in the selection of an interaction style. The user task model identifies the objects and actions involved in the accomplishment of the users' tasks. The set of tasks to be supported by the computer system is selected from the user task model. Once the user and task modelling is accomplished and the set of computer-supported tasks selected, choices are made regarding input devices, interaction styles, screen layouts, terminology, etc. These issues are not independent and latter choices often require revision of previous ones.

The entire user interface design process is iterative in nature [SCAPIN, 86] (Fig. 2.2). Rapid prototyping is advocated [BETTS, 87] in order to assert the validity of the design. The user is, therefore, an active element of the iterative chain. Design issues must be revised and assumptions reevaluated in light of changes dictated by results of user trials.

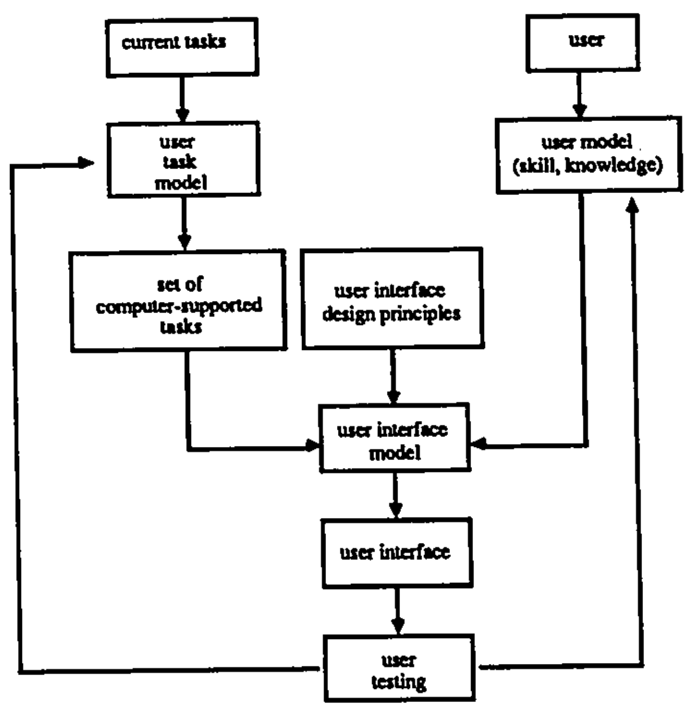


Figure 2.2: User Interface Design Process.

2.3.1 User Modelling

The profile of a user may include such diverse characteristics as age, sex, physical abilities, education, cultural or ethnic background, training, computer experience background, motivation, goals, and personality [SHNEIDERMAN, 87]. A user should also be characterized on the basis of the expected evolution of his knowledge and attitude towards the computer system. This evolution is a function of the initial profile of a given user and of the frequency of exposure to the computer system. Novice users, for instance, have no or very little a priori knowledge about the computer system. Intermittent users know their "way around" the system in general (retain semantic knowledge) but tend to forget the specific details of the operation of the system (lose syntactic knowledge). Expert users are expected to be thoroughly familiar with all aspects of the system (both semantic and syntactic aspects).

The user issue of user interface design also includes the so called *human factors*. "Human factors bears most on the perceptual and motor" human processes, "concerned as it is with the application to human performance, much as engineering is concerned with the application of physics and mathematics to mechanical and electrical performance" [FOLEY, 84]. Before addressing the human factors as such, we shall review the three types of basic human processes involved in a human-computer interaction as described by Foley, Wallace and Chan in [FOLEY, 84].

There are three types of basic human processes involved in a human-computer interaction: perception, cognition and motor activity. "Perception is the process whereby unintelligible physical stimuli (generated in this case by the computer) are received by the receptor organs, transmitted to the brain, and are there recognized by a process theorized to be akin to pattern recognition... Cognitive psychology deals with how we acquire, organize, and retrieve information", it "provides a framework for studying and simplifying the information structures that new users must develop... The motor process comes into play when the user, having received, recognized, and decided how to respond to the stimuli, performs a response in physical actions." This "process almost always depends on continuous perception and cognition to close the feedback loop." These basic human processes are at the root of the human factors. Shneiderman [SHNEIDERMAN, 87] lists five key measurable human factors:

1. Time to learn.

2. Speed of execution.

3. Rate of errors by users.

4. Subjective satisfaction.

5. Retention over time (ease of regaining competence after disuse period).

Foley, Wallace and Chan [FOLEY, 84] classify the human factors in a two level hierarchy. At the primary level we find:

1. Speed (time to accomplish a task).

2. Accuracy in accomplishing the task.

3. Pleasure derived from the process by the user.

And at the secondary level, influencing the primary level (speed, accuracy and pleasurability), we find:

1. Learning time.

2. Recall time (retention over time).

3. Short-term memory load (pertains to knowledge about the task underway).

4. Long-term memory load (pertains to the usage of a technique).

5. Fatigue and error susceptibility caused by insufficiently diversified tasks (routine), displeasing stimuli, uncertainty, heavy memory load, visual and auditory stimuli overload and motor fatigue.

6. Naturalness (measure of the system's ability to reflect everyday activities).

7. Boundedness (measure of the size of the user's tri-dimensional workspace: perception, cognition and physical space).

2.3.2 User Task Modelling

Modelling the user's tasks as performed before the introduction of a new computer system is an important early phase of the design of a user interface. Such a process is essential for the designer who seeks a thorough understanding of the application area. This understanding can be gained by studying the way the application is currently treated [FOLEY, 84].

Johnson [JOHNSON, 85] describes user interface design "as both a top-down and bottom-up set of processes with the goals and tasks requirements of the interaction being mapped onto appropriate interaction techniques". The top-down approach starts by an analysis of the users' tasks and ends with the production of the specifications of the user interface. The bottom-up approach starts with some creative design (e.g. mouse and icon based interaction style) which is then refined to meet the tasks' requirements. The relationship between the tasks' analysis and the user interface is shown in figure 2.3.

The task description used to form the GTM (generalized task model) has to be sufficiently abstract and general in order to allow enough latitude in the design for novel

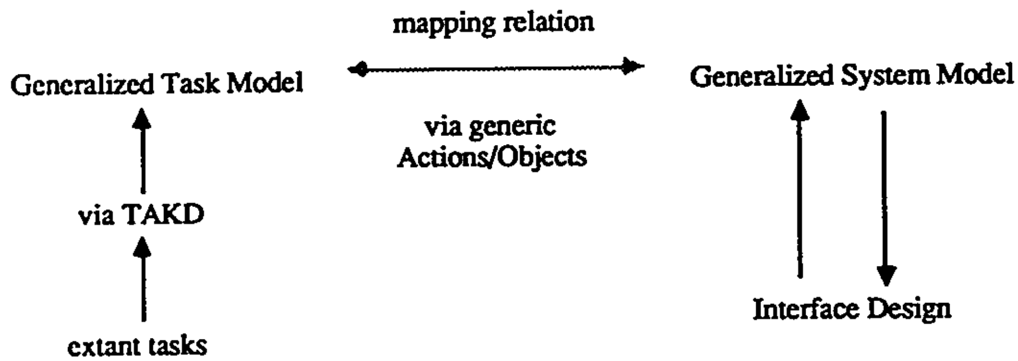


Figure 2.3: Task Description and Interface Design.

ideas. The GSM (generalized system model) is a high-level abstract description of the system; it is the conceptual design model.

Johnson [JOHNSON, 85] proposed a method for analyzing and describing tasks known as TAKD (task analysis for knowledge descriptions). The TAKD method essentially is a process for generating generic actions and generic objects from a description of the users' tasks. The procedure involves the following main steps:

1. Identifying a representative set of the tasks involved.

2. Carrying out task analyses of the identified tasks.

3. Identifying the complete set of actions and objects used in all the tasks.

4. Defining the set of generic actions and objects from the individual actions and objects.

5. Re-expressing the tasks in terms of generic actions and objects.

2.3.3 User Interface Modelling

Many techniques have been developed for modelling the user interface. These techniques involve the modelling of various aspects of a user interface such as its functionality, semantic, objects, actions, etc.

Foley, Wallace and Chan [FOLEY, 84] proposed a top-down, four-level approach to user interface modelling. The first level is the conceptual model, sometimes called the user's "mental model" of the system. For example, a user's mental model of the Apple's Macintosh microcomputer is of a desk-top. The second level, the semantic level, describes the meanings conveyed by the user's command input and by the computer's output display. The third level, the syntax level, defines how the units (words) which convey semantics are assembled into a complete sentence which instructs the computer to perform a certain task. Whereas all three previous levels are device independent, the fourth and lowest level, the lexical level, deals with device dependencies and with the precise mechanism by which a user specifies the syntax.

Shneiderman's syntactic/semantic model [SHNEIDERMAN, 87] attempts to model the knowledge of the user as two main components (Fig. 2.4): the *semantic knowledge* (about concepts), and the *syntactic knowledge* (about device-dependent details). The semantic knowledge is further divided into task concepts (objects and actions) and computer concepts (objects and actions).

The syntactic knowledge refers to the device-dependent details of a computer system. For examples, usage of the EXIT command to save a file and terminate a text editing session; and manipulation of a scroll-bar on a screen to scroll a portion of text. This knowledge is often arbitrary, ill-structured, and, being system dependent, difficult to conciliate across various computer systems. For example, on some systems, terminating a text editing session by issuing the QUIT command will also save the text, on some others

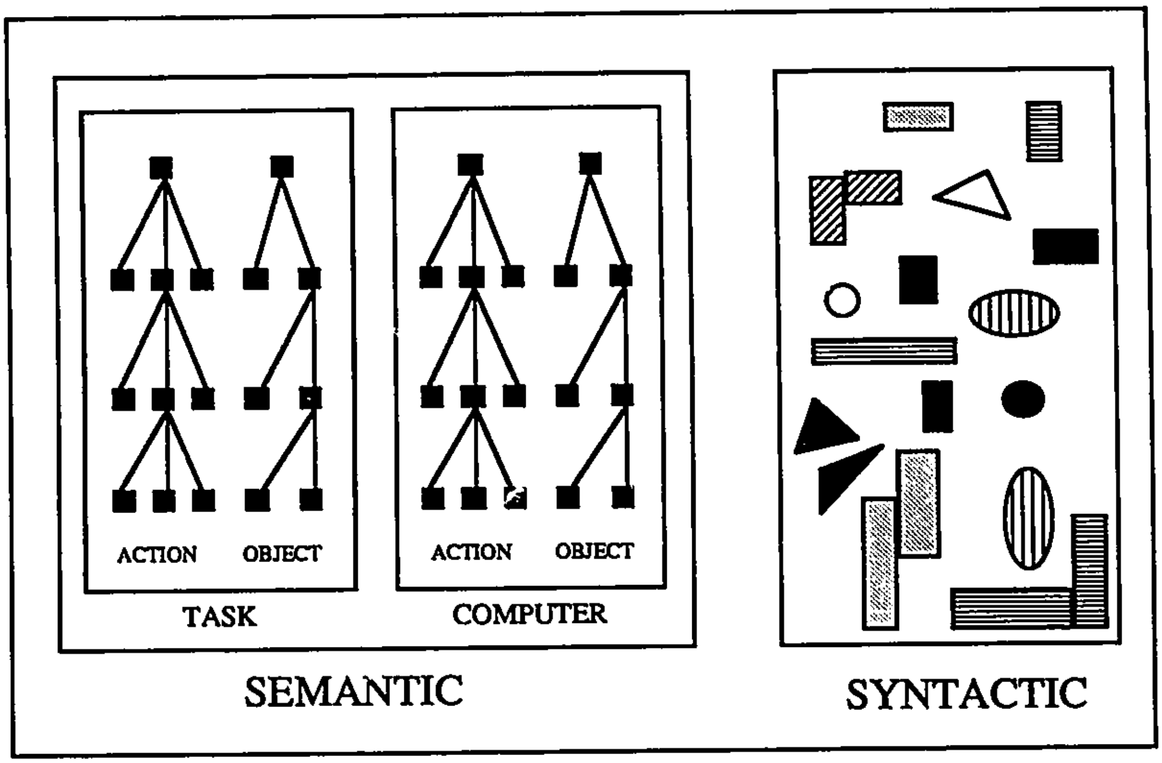


Figure 2.4: Syntactic/Semantic Model Diagram.

it will not. The syntactic knowledge is acquired by repeated usage and fades from the user's memory if not used regularly.

The semantic knowledge refers to the task concepts and the computer concepts. This knowledge is organized as hierarchies of actions and objects. The syntactic/semantic model decouples the computer and task concepts in order to accommodate users which have various levels of task expertise and computer expertise.

Computer objects refer to the various level of objects being manipulated by a computer. Examples of computer objects, from high to low level, are files, records, words and bits. *Computer actions* manipulate these computer objects. For example, saving a file may involve, at a high level, storing the file on a disk, verifying access control rights and assigning it a name; at a lower level, verifying the validity of the file type and allocating sufficient storage space; and at an even lower level, handling software and hardware errors.

Task objects refer to the various levels of objects which a user manipulates in the accomplishment of a task. For example, the author of a book deals with chapters, sections, paragraphs, sentences, and words. *Tasks actions* refer to the manipulation of these objects. For example, writing a paper involves the preparation of an abstract, then an outline, a detailed plan, the writing of the paper itself, and finally, the formatting for publication.

The user interface designer applying this model first gathers information about the semantic of the user's task objects and actions. Then, the proper computer objects and actions can be identified. Only at the end should the syntactic details be taken care of.

An object/action model such as the one above defines a user interface in terms of a set of object classes and a set of actions operating on these object classes. Olsen [OLSEN, 87] notes that the object/action model essentially "is a set of abstract data types which are defined in terms of the operations that can be performed on them." Olsen also points out that, "the object/action model does not capture information about the visual behavior of objects in response to each of the actions."

Sibert et al. [SIBERT, 86] note that user interface models fall into two broad categories: linguistic model (e.g. Foley, Wallace and Chan's model) and spatial model. A linguistic model views the interface as a dialogue between user and computer and focuses, in isolation, on the semantic, syntactic and lexical levels of the dialogue. Spatial models regroup the interactive graphic or direct manipulation models.

Sibert et al. propose to incorporate both the linguistic and the spatial components by adopting "an inherently spatial object-oriented paradigm" and including "the linguistic model by defining the boundaries of the lexical, syntactic, and semantic levels of the interface language within objects. Each boundary between two levels is embodied in a specialized object class with sufficient knowledge about adjacent linguistic levels to accomplish its function."

2.3.4 User Interface Design Tools

In order to assist the user interface designer, various tools have been devised. Buxton and Lamb [BUXTON, 83] divide them into two main categories: toolboxes and generic systems (Fig. 2.5 from [KOIVUNEN, 88]).

Different levels of Toolboxes	**Generic Systems**	
	Application Frameworks	User Interface Management Systems

Figure 2.5: User Interface Design Tools Taxonomy.

Toolboxes essentially are sets of building blocks (e.g. window handlers) assembled by the user interface designer. *Generic systems* are of two types: application frameworks and UIMSs (User Interface Management Systems). An *application framework* consists of a skeleton of a reusable user interface which the designer modifies to realize his design. An *UIMS* includes the above but also supports design specification, implementation and operation of the user interface. An UIMS is "a tool that provides support for the definition of the user/application dialogue, imposes external control on the application, provides support for the presentation of the application's output and includes an interactive component providing support for the interaction between an application and an end user" [BETTS, 87]. Ideally, an UIMS should be able to automatically produce a user interface from its model and design specifications.

2.4 User Interface Architecture

The main software constituents of a user interface [LANTZ, 87a] are the workstation agent, dialogue manager, and workstation manager (Fig. 2.6).

The *workstation agent* is responsible for providing the basic interface between the hardware and the rest of the system. It contains the various device- and media-dependent drivers or servers.

The *dialogue manager* is responsible for command specification and invocation, user response handling, and intelligent information presentation. Based on information such as user profiles, context, workstation and dialogue specifications, the dialogue manager determines the interaction styles and the communications media to employ in the human-computer dialogues. For example, in a given hypothetical system, two windows (dialogues) using different interaction styles and input devices are opened: one uses a mouse-driven menu-oriented style of interaction; the other uses commands entered via the keyboard.

The *workstation manager* is responsible for controlling multiple dialogue managers and their applications. It implements policies such as window tiling or overlapping.

The separability of the application and data presentation is a major architectural issue [HILL, 86]. There are two types of mutually exclusive separability: logical and physical. Logical separation implies no data sharing and very limited communication between the user interface and the applications. The issue of independent development of the user

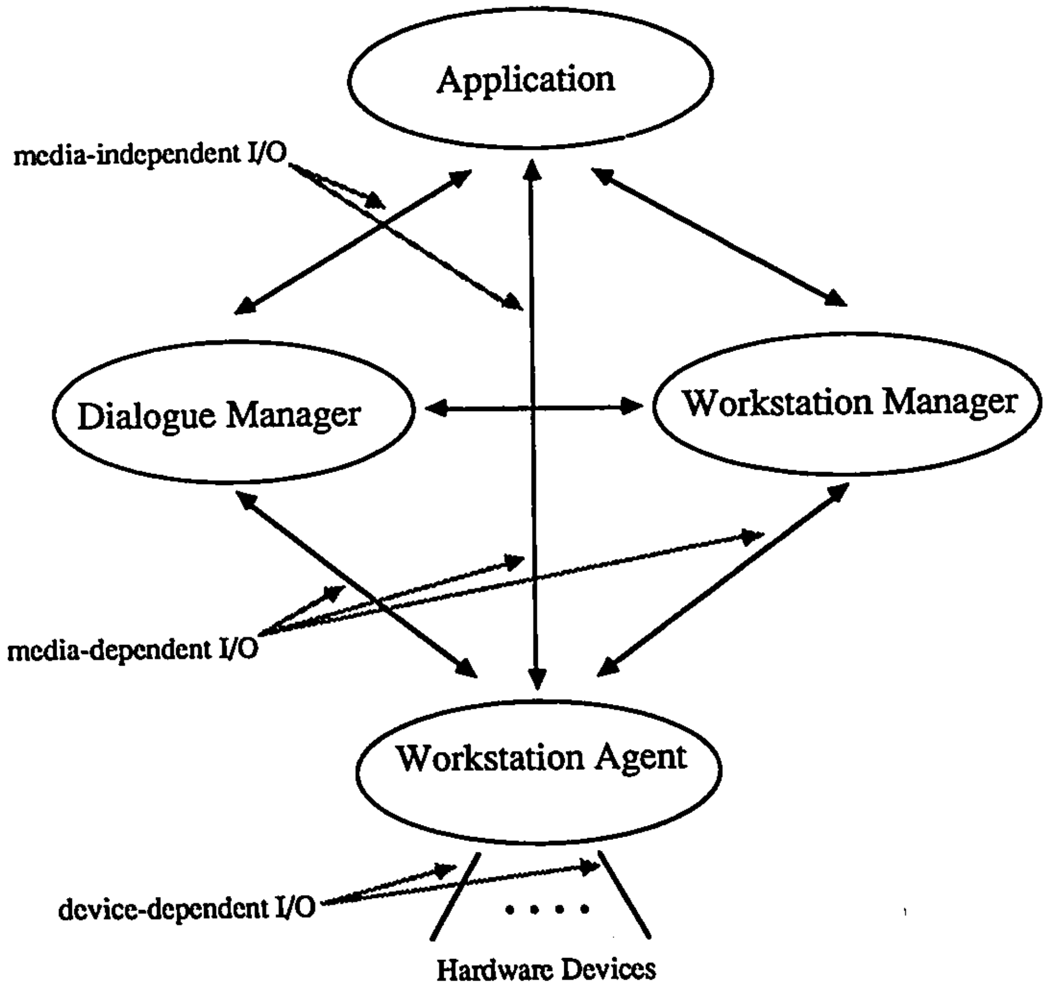


Figure 2.6: Basic User Interface Architecture.

interface and the applications is related to logical separation. Separation is physical when the user interface and the applications reside in separate modules or files. Physical separation mainly concerns programming styles and practices.

2.5 Summary

User interfaces are interactive systems which support two-way communications between a human user and a computer. There are various styles in which to carry the interaction, each reflecting the particularities of the human user and of the supported applications.

User interfaces are designed by first analyzing the intended users and their current tasks. The set of tasks to be supported by the computer is then selected. The user interface itself is then designed from guidelines and principles (based on human factors), and from the models of the users and of their tasks.

Bibliography

[BETTS, 87] B. Betts, D. Burlingame, G. Fischer, J. Foley, M. Green, D. Kasik, S.T. Kerr, D. Olsen, J. Thomas, "Goals and Objectives for User Interface Software", *Computer Graphics*, pp. 73-78, April 1987.

[BUXTON, 83] W. Buxton and M.R. Lamb, "Towards a Comprehensive User Interface Management System", *SIGGRAPH'83 Conference Proceedings*, pp. 35-42, July 1983.

[FISCHER, 87] G. Fischer, "An Object-oriented Construction and Tool Kit for Human-Computer Communication", *Computer Graphics*, pp. 105-109, April 1987.

[FOLEY, 84] J.D. Foley, V.L. Wallace and P. Chan, "The Human Factors of Computer Graphics Interaction Techniques", *IEEE Computer Graphics and Applications*, pp. 13-48, November 1984.

[HILL, 86] R.D. Hill, "Supporting Concurrency, Communication, and Synchronization in Human-Computer Interaction - The Sassafras UIMS", *ACM Transactions on Graphics*, pp. 179-210, July 1986.

[JOHNSON, 85] P. Johnson, "Towards a Task Model of Messaging: an Example of the Application of TAKD to User Interface Design", *Proceedings of the Conference of the British Computer Interaction Specialist Group*, pp. 46-62, September 1985.

[KOIVUNEN, 88] M.-R. Koivunen and M. Mantyla, "HutWindows: An Improved Architecture for a User Interface Management System", *IEEE Computer Graphics and Applications*, pp. 43-52, January 1988.

[LANTZ, 87a] K.A. Lantz, "Multi-Process Structuring of User Interface Software", *Computer Graphics*, pp. 124-130, April 1987.

[OLSEN, 87] D.R. Olsen, "Larger Issues in User Interface Management", *Computer Graphics*, pp. 134-137, April 1987.

[SCAPIN, 86] D.L. Scapin, "Guide ergonomique de conception des interfaces homme-machine", Rapports Techniques, *INRIA*, 92 p., Octobre 1986.

[SHNEIDERMAN, 87] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Adison-Wesley, 448p., 1987.

[SIBERT, 86] J.L. Sibert, W.D. Hurley and T.W. Bleser, "An Object-Oriented User Interface Management System", *SIGGRAPH'86 Conference Proceedings*, Dallas, pp. 259-268, Aug. 1986.

Chapter 3

Picture Archiving and Communication Systems Review

3.1 Introduction

Picture Archiving and Communication Systems (PACS) provide storage for digital pictures or images and means for retrieving and transmitting these over electronic media [HUANG, 87]. The three main components of a PACS are: an image storage (archival) subsystem (e.g. optical disk jukebox), a communication medium (e.g. fiber optic network), and end-point workstations (terminals) for entering, retrieving and manipulating images.

Growing research in the field of PACS applied to radiology highlights the need for adequate digital radiograph viewing workstations. The main goal is to provide the necessary information (patient file, radiographs) and facilities for radiologists to generate appropriate medical reports.

This chapter first reviews PACS workstation requirements and preferences studies, and then some PACS workstation designs.

3.2 PACS Workstation Requirements and Preferences Studies

This section presents the main findings of some studies conducted to investigate the requirements (e.g. system response time) and preferences (e.g. when placing films on viewboxes, hand views may be placed with fingers up or down; chest lateral views seen with pectorals facing left or right [WITTINGHAM, 88]).

Care must be taken though when analyzing viewing workstation requirements as most studies [ROGERS, 85] [HORII, 86] [PERRY, 84] were made using X-ray Computer Tomography (CT) images[1]. These results may not be readily generalized to other image modalities (e.g. Digital Subtraction Angiography (DSA), Magnetic Resonance Imaging (MRI), Ultrasound imaging (US), X-ray Fluoroscopic Imaging (FI), projectional X-ray films) because of differences in terms of resolution (e.g. A CT image has a resolution of 512x512 pixels whereas a "plain" projectional X-ray film of the chest may produce a digitized matrix of 7000x8000 pixels [PERRY, 84]), or in terms of relationship between the images themselves (e.g. CT images display slices of a body area spaced a few millimeters apart whereas Digital Subtraction Angiography (DSA) images form a chronological sequence as the injected radio-opaque dye dissolves in the blood vessels).

The PACS workstation environmental factors covered in the literature have been compiled in a recent paper by Horii et al. [HORII, 88]. These factors included ambient noise, lighting, temperature; furniture; aesthetics, etc.

The authors also presented a thorough examination of workstation requirements in terms of the quality of the data (images principally). Raised issues are data ordering (sorting), grouping (based on anatomic or time relationships for example), formatting (physical arrangement of data presented), scaling, quantity (amount of data presented),

[1] CT images originate from a digital process and are then printed on radiographic film.

complexity, and relevancy (noisiness of data).

Workstation design must also take into consideration qualities of the observer (user). The intrinsic characteristics (unlikely to be affected by workstation design) of an observer are the physical characteristics of physiognomy (physical human factors), personality (traits and predispositions), and memory (capacity of individual to retain data over time). Other factors susceptible to be affected by ergonomic considerations are perceptual skills (information scanning and extraction), learning, motivation, and training.

A list of specific ergonomic design issues are reported by the authors from previous papers. Of these, the following two points are specially relevant to radiology:

1. If multiple display monitors are used, keep the distances between these small enough so that excessive head movement is not required when scanning all displays.

2. Touch screens which have active areas over the image display should be avoided. Radiologists and those who view images with them are notorious for pointing out features on the images.

Rogers et al. [ROGERS, 85] conducted a repeatedly cited study of radiologists' reading habits. Three experiments were performed, to each of which three radiologists participated. The study was exclusively based on CT images. The radiograph reading process is decomposed into three main stages:

1. Review patient history,
   sort and mount images on viewboxes. .

2. Scan images,
   detect patient pathologies,
   select most relevant films.

3. Focus attention and compare images,
   interpret and evaluate case.

The authors concluded the study by listing a set of desired console features:

1. Allow access to the entire patient history (text and images).
2. Allow selection of subsets of patient history for review.
3. Provide automatic progression through CT slices series by scrolling mechanisms (forward, backward, variable speed, pause control).
4. Enable temporary marking of images while scrolling (for calling back marked images).
5. Provide a "multi image mode" where scrolling maintains constant the images' spatial relationship to each other.

The same group of authors later proposed a workstation evaluation protocol [JOHNSTON, 87]. The evaluation protocol required the targeted clinical tasks to be identified. The authors proposed a classification of tasks and their associated requirements on the workstation design.

Verification. This task involves the verification of the information adequacy: correctness of film exposure, appropriateness of views, correctness of angle of views. The complete set of images must be available but images may be displayed at less then full resolution.

Interpretation. The goal of this process is for the radiologist to generate a report of his findings. This process is divided into the three stages presented above [ROGERS, 85]. This process requires the ability to display images (in part or in full) at full acquisition resolution. Means for rapid indexing of the image set and for rapid access to any image or subset of images must also be made available. These means must not be so complex as to be disruptive to the task underway. Also, image processing must be automatic or easily instituted.

Consultation. This process includes consultation between a referring physician and a radiologist, or two colleague radiologists consulting on a difficult or interesting case. For such a process, a single screen may be sufficient for viewing the images (unless consultation leads to interpretation). Also, no longer required would be features such as rapid access, image processing capability, and large archiving capacity.

A study by Horii et al. [HORII, 86] attempted to determine the proper number and physical arrangement (geometric layout) of CT slice images. The study used films rather than video monitors. Four different cases were used (from the following categories: abdominal, brain, bone and chest) and seven radiologists from various specialties participated in the experiments.

The most preferred physical arrangement of the images was an horizontal display. Although simultaneous display ot all the images at full resolution was thought to be unnecessary, full resolution must be available for single images. Simultaneous display of many slice images at lower resolution was judged useful for giving a "quick look" at a case.

The authors noted that the preferred geometric layout of images varied among radiology specialties and personnel reading habits (e.g. one radiologist, used to reading CT images directly from the acquisition console, preferred to view images one at a time). Also, the results were strongly influenced by standard practice at the study site where CT slice images are normally presented horizontally.

A paper by Perry et al. [PERRY, 84] considered digital image display console design issues based on CT images. The authors suggested to display 16 slices at a time, to mask bright areas of the screen to minimize the impact of flicker, and to provide real time response to image manipulation functions.

They also gave an exhaustive "wish list" covering image display capabilities and qualitative issues, administrative issues, and human engineering issues. From the list, the authors prioritized the following issues for further study:

1. Resolution requirements for large and small films.

2. Automatic contrast adjustment.

3. Number of simultaneously displayed images.

4. Lightbox versus videoscreen reading habits and training.

5. Selection of preset functions.

6. Waiting tolerances of image film transmission time.

O'Malley and Guinta [O'MALLEY, 88] wrote a paper in which they derived an operational model and the performance characteristics of the film alternator.

The film alternator provides several features: a display region which provides immediate viewing capabilities, storage for a quantity of image information, and a control mechanism which provides easy access to the storage. The viewing or display region consists of two overhanging panels, or rows, of lightboxes, generally of adequate size to hold four films of 14 by 17 inches on each panel. There are basically two types of alternators: a "conveyor belt" type on which panels of information are accessed sequentially, and a "stacking" type, on which panels are accessed randomly by number.

The electronic counterpart of the film alternator includes the following subsystems: user control, display subsystem, local storage, and network interface. From an operational scenario the authors have derived a set of requirements for the electronic alternator. The display should have eight 2048x2048 monitors, a local storage capacity of 300 Mbytes of uncompressed data and a transfer rate of 3 Mbytes/second, and a network capacity of 1 Mbytes/second. The authors relaxed the display requirements by suggesting the incorporation of techniques such as minification/magnification, or more advanced image processing techniques such as 3-D reconstruction.

3.3 PACS Workstation Designs

This section reviews some existing PACS workstations. Each workstation is described in terms of its hardware (visual workspace and input devices) and of the main characteristics of the user interface. Most of these workstations present text information and images on separate monitors. All workstations use monitors of at least 1000 by 1000 pixels of resolution for displaying images.

3.3.1 CommView

The AT&T's *CommView* [KASDAY, 86] system is equipped with a keyboard, a mouse and three monitors aligned horizontally (Fig. 3.1). The left most monitor is reserved for text and, therefore, has the keyboard; the other two monitors are dedicated to image display. The user interface environment is based on folders. A folder is a rectangular object with a tag on top and a stack of pages inside (Fig. 3.2). A folder may contain text, graphics or images. Each page of a folder may contain more than one image (1, 2, 4, 6, 8, etc.).

Initially, a patient file is presented as a stack of folders on one monitor. Images or reports are then accessed by bringing the relevant folders to the top of the stack or to any other monitor. A folder may extend over more than one monitor. A single mouse pointer covers all three monitors and is used for selecting folders and flipping through pages of folders.

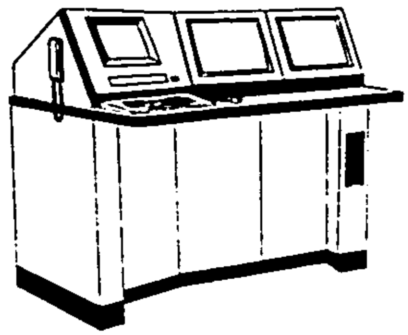


Figure 3.1: CommView Workstation.

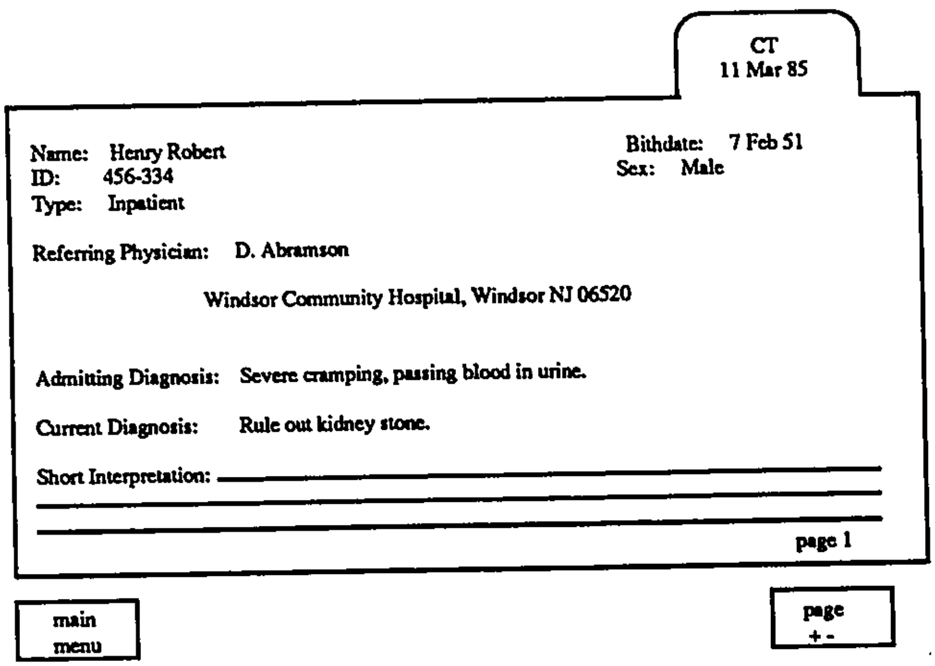


Figure 3.2: CommView Folder.

3.3.2 Arizona Viewing Console

The University of Arizona's *Arizona Viewing Console* [VERCILLO, 87] [McNEILL, 87] [DALLAS, 87] is equipped with a keyboard, a trackball, a set of knobs and three monitors (a color command monitor and two monochrome high resolution image display monitors).

A patient file is selected by typing the patient's name. Then, from a directory displayed on the command monitor, images are selected and placed anywhere on either display monitor. Image manipulation operations are available from menus on the command monitor and affect only one image at a time, the one designated as the active image. Image processing functions are controlled by the knobs. All three monitors are fitted with pressure sensitive screens which enable image selection and activation to be done by a simple finger touch. The interface may be personalized by user defaults for image display location, orientation, and initial filtering.

3.3.3 MarView

The Philips' *MarView* workstation [SCHUTTENBELD, 87] is equipped with a mouse and four monitors. The visual workspace is divided into two parts: the "desk-part" and the "lightbox-part". The desk-part, a single monitor, displays administrative and system information (Fig. 3.3). The remaining three monitors, which constitute the lightbox-part, are used as electronic lightboxes for image display. The three electronic lightboxes are represented on the desk-part by three horizontally adjacent windows. Images are displayed on a given electronic lightbox (monitor) by placing the appropriate image icon in the corresponding lightbox window.

3.3.4 Clinical Radiology Imaging System

The UCLA's *Clinical Radiology Imaging System* (CRIS) is composed of six monitors (for image display), a text screen, and a trackball [TAIRA, 87]. The image monitors are

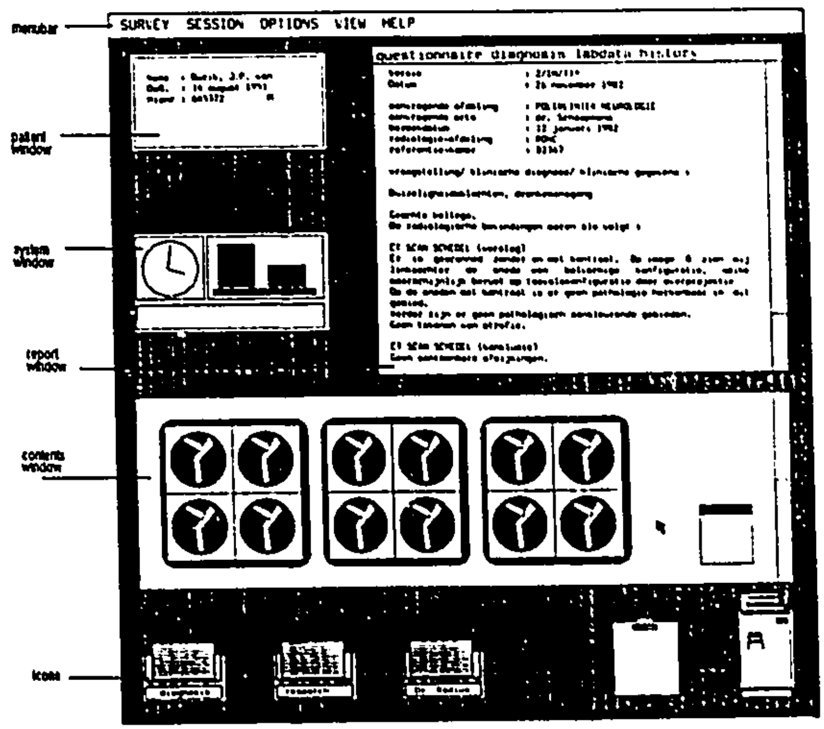


Figure 3.3: MarView desk-part of the user interface.

organized in two rows of three monitors each. The text screen is located below the set of image monitors (centered horizontally) and presents various menus and functions, all of which may be selected with the trackball.

Initially, a list of all patients is displayed on the text screen. The list is replaced by some general hospital information (e.g. date of birth, sex, etc.) once a patient is selected. A pictorial index of the patient's images is then displayed on the monitors. The pictorial index is a chronological display of minified versions of the images. Based on the pictorial index, a set of images of interest (to be displayed) are selected from a menu on the text screen. If there are more than six images in the image display set, the trackball is used for bringing other (or previous) images in groups of three at a time (essentially achieving vertical scrolling of the entire image display set).

3.3.5 FilmPlane

The University of North Carolina's *FilmPlane* is a single screen console for viewing images slices for Computer Tomography (CT) [BEARD, 88]. All slices of a patient are minified and presented on a two-dimensional plane. A square window covering four slices, called the "viewport", selects the images which can be seen at higher resolution (the group of four images then replaces the plane of minified images). A mouse is used for moving the viewport on the plane of minified images, thus selecting other groups of slices for viewing.

3.4 Summary

PACS workstations should be able to accommodate user preferences for viewing images. The entire patients history should be made available to the radiologist. The radiologist must also be able to select a subset of items (images, reports,etc.) of interest from the patient file to report his diagnosis.

A common characteristic of the workstation designs presented in this chapter is the constraint imposed by the physical image viewing surface on the spatial arrangement of the images: the number and layout of lightbox windows correspond to the number of physical monitors. This departs from the traditional arrangement of viewboxes in radiologists' offices: the total image workspace is thus significantly reduced. Radiologists normally use a visual working surface of 8 viewboxes, two rows of 4 viewboxes each. If more films need to be viewed than can be displayed at once on the viewbox surface then a radiologist has two options: to manually exchange films or use an alternator.

Bibliography

[BEARD, 88] D. Beard, J.L. Creasy, J. Symon, R. Cromartie, "Experiment comparing film and the FilmPlane radiology workstation", *SPIE Medical Imaging II*, Vol. 914, pp. 933-937, 1988.

[DALLAS, 87] W.J. Dallas, T.W. Ovitt, M.P. Capp, H. Roehrig, G.W. Seeley, R. Martinez, K.M. McNeill, R. Vercillo, R. Lamoreaux, C. Archwamety, M. Nematbakhsh, M. Kakegawa, K. Komatsu, E. Nishihara, M. Suzuki, T. Ozeki, H. Umino, "A Prototype Totally Digital Radiology Department: Conception and Initiation", *SPIE Medical Imaging*, Vol. 767, pp. 700-707, Feb. 1987.

[HORII, 86] S.C. Horii, G. Isles, R.T. Bergeron, "Display Preferences for Viewing CT Scans", *SPIE Medicine XIV/PACS IV*, Vol. 626, pp. 426-429, 1986.

[HORII, 88] S.C. Horii, H.N. Horii, P. Kowalski, "An Eclectic Look at Viewing Station Design", *SPIE Medical Imaging II*, Vol. 914, pp. 920-928, 1988.

[HUANG, 87] H.K. Huang, *Elements of Digital Radiology*, Prentice-Hall, 370p., 1987.

[JOHNSTON, 87] R.E. Johnston, D.C. Rogers, J.L. Creasy, D.V. Beard, B.M. Hemminger, J.R. Perry, S.M. Pizer, "How To Evaluate A Medical Imaging Display Workstation", *SPIE Medical Imaging*, Vol. 767, pp. 616-621, Feb. 1987.

[KASDAY, 86] L.R. Kasday, "Human Factors Considerations in PACS Design", *SPIE Medicine XIV/PACS IV*, Vol. 626, pp. 581-592, 1986.

[McNEILL, 87] K.M. McNeill and H.D. Fisher, "A Model for Radiologic Workstation User Interface Design", *SPIE Medical Imaging*, Vol. 767, pp. 713-716, Feb. 1987.

[O'MALLEY, 88] K.G. O'Malley, J.A. Giunta, "The Alternator: Determination of its Fundamental Features, as a Basis for Design of a PACS Diagnostic Workstation", *SPIE Medical Imaging II*, Vol. 914, pp. 988-994, 1988.

[PERRY, 84] J.R. Perry, R.E. Johnston, E.V. Staab, B.G. Thompson, B.C. Yankaskas, B.C. Brenton, "Digital Image Display Console Design Issues", *Proceedings of the IEEE Computer Society International Symposium on Medical Images and Icons*, pp. 18-22, 1984.

[ROGERS, 85] D.C. Rogers, R.E. Johnston, B.Brenton, E.V. Staab, B. Thompson and J.R. Perry, "Predicting PACS Console Requirements From Radiologists' Reading Habits", *SPIE Picture Archiving and Communication Systems (PACS III) for Medical Applications*, Vol. 536, pp. 88-95, 1985.

[SCHUTTENBELD, 87] H.H.W. Schuttenbeld and B.M. ter Haar Romeny, "Design of a User-interface for a PACS Viewing Station", *SPIE Medical Imaging*, Vol. 767, pp. 844-848, Feb. 1987.

[TAIRA, 87] R.K. Taira, "Operational Characteristics of Pediatric Radiology: Image Display Stations", *SPIE Medical Imaging*, Vol. 767, pp. 571-576, Feb. 1987.

[VERCILLO, 87] R. Vercillo, H.D. Fisher, R.D. Lamoreaux, K.M. McNeill, "Digital Image Review Console", *SPIE Medical Imaging*, Vol. 767, pp. 708-712, Feb. 1987.

[WITTINGHAM, 88] Wittingham, *Private Communications*, Ottawa Civic Hospital, Ottawa, 1988.

Chapter 4

Object-Oriented Programming Languages

4.1 Introduction

An important goal in design is to identify a program structure which simplifies both program development, maintenance and modifications [LISKOV, 87]. Data abstractions help achieve this goal by allowing us to abstract from the implementation of data structures to their behavior. They present a higher level interface which simplifies program structure.

Object-oriented programming is primarily a data abstraction technique. The primitive element of object-oriented programming is the object [STEFIK, 86], a named collection of data and procedures manipulating the data. Uniform use of objects in object-oriented programming contrasts with the use of separate data and procedures in traditional languages. An object-oriented language is defined as a language featuring objects and supporting object classes and class inheritance [WEGNER, 87].

We begin by defining the term "data abstraction". Then we introduce the concepts of classes and instances. Next we discuss the messaging principle and the inheritance mechanism which support object-oriented programming. We end this chapter with an overview of Smalltalk, an object-oriented programming language.

4.2 Data Abstraction

The purpose of abstraction in programming is to separate behavior from implementation [LISKOV, 87]. For example, the procedure is one such abstraction which is employed to accomplish a task, regardless of its implementation. The concept of *data abstraction* or *abstract data type* is an extension of the above idea, but applied to data. Data abstraction is achieved by *encapsulating* a data object's implementation or *representation* with a set of operations which manipulate it. Hence, the data's representation is hidden and can only be manipulated through the set of operations. In object-oriented programming languages, data abstraction is realized by class objects and their instance objects.

4.3 Classes and Instances

An object is a named collection of data and procedures manipulating the data. Let C be a *class object*. Then C is expressed as the set of *attributes*

$$\{v_1, v_2, \ldots, v_k; {}^cv_1, {}^cv_2, \ldots, {}^cv_p; m_1, m_2, \ldots, m_l; {}^cm_1, {}^cm_2, \ldots, {}^cm_q\},$$

where v_i is an *instance variable* or *slot*, ${}^cv_j$ a *class variable*, m_i an *instance method*, and ${}^cm_j$ a *class method*. Let $\mathcal{I}$ be an *instance object* obtained from C. Then $\mathcal{I}$ is the set $\{o_1, o_2, \ldots, o_k; m_1, m_2, \ldots, m_l;\}$, where o_i is an object *instantiating* the instance variable v_i. The set $\{o_1, \ldots, o_k\}$, is often referred to as the *state* of $\mathcal{I}$.

Thus we distinguish two major categories of objects: classes and instances [STEFIK, 86]. A class object is a description or template from which one or more similar[1] instance objects are created (by the *instantiation* process).

A class defines (specifies the names and initial or default values of) two kinds of variables: class variables and instance variables. Class variables hold information shared

[1] Sharing the same attributes but not their states or vales [STEIN, 87].

by all instances of the class. Instance variables contain the information specific to a particular instance. For example, the class **Point** = (**x,y** ; **PlaneOrigin** ; ;) specifies two instance variables, **x** and **y**, and a class variable associated with all points, **PlaneOrigin**. Each instance of **Point** has its own **x** and **y** instance variables, but all of the instances use the same **PlaneOrigin** class variable; any change made to the value of **PlaneOrigin** would be seen by all of the instances [STEFIK, 86].

A method is a procedure which performs a specific task for its object. A method is expressed as a triplet

$$\langle n, A, B \rangle,$$

where

n is the method's name or identifier,

A is a list of formal *arguments*,

B is the method's implementation or body.

A class specifies two kinds of *methods*: class methods and instance methods. Class methods can only access the class variables of their class and are used mainly to create new instances of the class. For example, the **Point** class may define the class method ⟨ `newX:Y:` , {x,y} , body ⟩ which creates a new **Point** instance with instance variables **x** := **x** and **y** := **y**. Instance methods have access to the instance variables of a particular instance and to the class variables of that instance's class. For example, the **Point** class may define the instance method ⟨ `distanceFromOrigin` , {} , body ⟩ which returns the distance between a point and the origin of the plane. In order to compute the distance, the method must access the instance variables **x** and **y**, and the class variable **PlaneOrigin**.

4.4 Messages

Actions in object-oriented programming are achieved by passing *messages* between objects [STEFIK, 86]. Message passing adheres to the data abstraction principle. A message is composed of three parts: the *receiver* object, the *message selector*, and zero or more *arguments* (which are also objects). A receiving object responds to an incoming message by first choosing the method which implements the message selector, executing the method, and then returning to the caller [COX, 86]. Message sending is a form of indirect procedure call [STEFIK, 86]: instead of naming a procedure to perform an operation on an object, one sends the object a message. The responsibility for interpreting a message (choosing an appropriate method) lies with the receiver.

Polymorphism is the property where different classes of objects respond to exactly the same message [STEFIK, 86]: the same message can elicit a different response depending on the receiver [PASCOE, 86].

The set of messages an object can respond to is called its *protocol* [ROBSON, 81]. Different objects related to the same application are usually made to respond to the same protocol. For example, in a windowing application, all windows and sub-windows would respond to the following protocol: move, overlap, display, hide, close.

4.5 Inheritance

Let $\mathcal{O}''$ be an object described as

$$\mathcal{O}'' = \{v_1, \ldots, v_k; m_1, \ldots, m_l\},$$

where

v_i is a variable (instance or class variable), and

$$m_j \text{ is a method (instance or class method).}$$

If $\mathcal{O}'$ *inherits* from $\mathcal{O}''$, then

$$\mathcal{O}' = \{v_1^*, \ldots, v_p^*; m_1^*, \ldots, m_i^*, m_{i+1}^*, \ldots, m_j^*, m_{j+1}^*, \ldots, m_q^*\}$$

where

$$\{v_1, \ldots, v_k\} \subseteq^2 \{v_1^*, \ldots, v_p^*\} \tag{4.1}$$

$$m_k = \langle n_k, A, B \rangle \tag{4.2}$$

$$\{m_1^*, \ldots, m_i^*\} \subseteq \{m_1, \ldots, m_l\} \tag{4.3}$$

$$n_f^* \notin \{n_1, \ldots, n_l\}, \quad \text{for} \quad i+1 \leq f \leq j \tag{4.4}$$

$$\{n_{j+1}^*, \ldots, n_q^*\} \subseteq \{n_1, \ldots, n_l\} \tag{4.5}$$

Inheritance is a mechanism by which an object acquires the attributes of other objects. Classes make direct use of the inheritance mechanism, whereas instances participate indirectly through their classes [STEFIK, 86]. A *subclass* inherits all variables (4.1) and methods (4.2, 4.3) from its *parent class* or *superclass*, but may also contain additional new variables and methods (4.4). If an added method has the same name or identifier as an inherited one does, then the added method *overrides* (replaces) the inherited one (4.5). Most object-oriented programming languages, however, provide a mechanism for a subclass to access the overridden methods of its superclass.

Multiple inheritance is realized by an acyclic directed graph (lattice) of classes; that is, when a subclass has one or more superclasses (a class may not, even indirectly, have itself as a superclass). Since a class inherits the union of variables and methods from all its superclasses, some precedence rule must be applied to resolve possible conflicts (i.e. multiple definitions of variables or methods). For example, Square in Fig. 4.1 inherits in

[2] $\subseteq$ denotes set inclusion.

the following precedence order: (**Parallelogram, Quadrilateral, EquilateralPolygon, Polygon, GeometricFigure**).

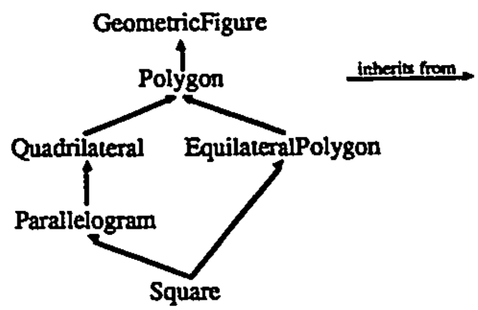


Figure 4.1: Multiple Inheritance.

Single inheritance or *hierarchical inheritance* is achieved when the lattice degenerates to a rooted tree of classes. Then each subclass may have one, and only one, superclass (except the root class of the tree, usually named the **Object** class, which has no superclass). The Smalltalk language described in the next section is based on hierarchical inheritance of classes.

Inheritance supports polymorphism: instances of a class understand at least the same messages understood by the instances of the superclass. Inheritance is also important for redesign and code reuse because it enables the easy creation (in terms of economy of expression) of new objects which are almost identical to other objects, except for a few incremental changes [FISCHER, 87]. Indeed, *subclassing* creates *specialized* abstractions from general abstractions (the superclasses).

4.6 Smalltalk

Smalltalk is both a language and an environment. In this section we concentrate on Smalltalk/V, a dialect of Smalltalk. First the language is reviewed in light of the previous sections. Then the characteristics of the environment are described. Finally the model-view-controller (MVC) paradigm is presented.

4.6.1 The Language

Instances and Classes

The central structure of the Smalltalk/V[3] language is based on data abstraction and encapsulation [SEIDEWITZ, 87]. All entities are objects of equal status: classes are objects, just as instances of classes are objects.

A class defines instance and class variables, and instance and class methods. Variables may only contain other objects. Instance variables can be *named* or *indexed*. Named instance variables are accessed by using their name. For example, the **x** and **y** variables of the **Point** class. Indexed instance variables are accessed with integer indices. For example, the 1^{st} element, the 2^{nd} element, ..., the n^{th} element of an n-element instance of the **Array** class. Instance variables are accessible only from instance methods. Class variables are accessible from both instance and class methods.

Methods determine an object's behavior. For example, the smallest method of the class **Array** returns the smallest element of an array and is defined as follows:

```
smallest
    "Answer the smallest element of the receiver."
| smallestElement |
smallestElement := self at: 1.
self do: [ :anElement|
    anElement < smallestElement
        ifTrue: [ smallestElement := anElement ]].
^smallestElement
```

[3] In this subsection, "Smalltalk/V" refers to the language, by opposition to the environment.

The quoted text is a general comment for the method. The *temporary variable* **smallestElement** is declared by enclosing it in vertical bars. Temporary variables of a particular method exist only for as long as the method is active. The pseudo-variable self refers to the object for which the method is active. The caret (^) signals the beginning of the *return expression*, the result of which is returned upon termination of the method. In this example, the **smallestElement** is returned.

Messages

Message sending serves as a uniform metaphor for objects to communicate with other objects: the message sending mechanism is the same for all messages [RENTSCH, 82]. A message is composed of the receiver object, the message selector, and of zero or more formal arguments. A message with no arguments is called an *unary* message. For example,

#(1, 0, -5, 10) `smallest`

sends the message selector `smallest` to the 4-element array **#(1, 0, -5, 10)**. A message with one or more arguments is called a *keyword* message. For example,

'The Little Prince' `copyFrom:` **5** `to:` **17**

has two arguments, **5** and **17**. The message selector `copyFrom:to:` is divided by the arguments (the colon (:) signals the presence of an argument).

Upon reception of a message the receiver object selects a method which implements the message selector. In Smalltalk/V, the method triggered by a message has a name which corresponds to the message selector: there is a direct mapping of message selectors onto method names.

An object responds to a message with an *answer* or *reply*. The reply confirms the completion of the action and returns the information requested. In the first example, **#(1, 0, -5, 10)** `smallest` returns -5. The second example returns **'Little Prince'**.

Smalltalk/V does not enforce *strong typing*: it does not determine the type compatibility of all expressions representing values from the static program representation at compile time [WEGNER, 87]. Hence, variables may contain objects of different classes at different moments of the execution. The absence of strong typing enforcement supports polymorphism. Indeed, in the **smallest** method, no type has been declared for the variables **smallestElement** and **anElement**. Hence, the method will behave properly for any array of homogeneous (from the same class) objects, as long as the message selector < is understood by each element object. For example, **#(1, 0, -5, 10)** `smallest` returns the integer **-5**, and **#('b', 'a', 'c')** `smallest` returns the string **'a'**. The same message selector < has been sent from the same method `smallest`, but was received by different objects (integers and strings) and interpreted differently (for integers, it compares magnitudes; for strings, it compares according to alphabetical order).

Inheritance

In Smalltalk/V, class inheritance is realized by a hierarchy (a tree), with class **Object** at the top (Fig. 4.2). Higher classes of the hierarchy represent *generalized* abstractions, whereas lower classes represent more *specialized* abstractions. For example, class **Collection** is the top of a hierarchy of subclasses: **IndexedCollection**, **FixedSizeCollection**, and then **Array**. There is a net progression towards specialization from the top of the hierarchy (class **Collection**) to the lower end (class **Array**). Methods, overridden by inheritance, may still be accessed from the immediate subclass by sending a message to the pseudo-variable **super**. For example, the **initialize** instance method of some class first initializes its superclass' instance variables, then its own:

```
initialize
    "Initializes the instance variables to 0."
super initialize.
instanceVariable1 := 0.
```

```
instanceVariable2 := 0.
```

The expression **super initialize** activates the **initialize** method of the superclass, which in turn initializes its own instance variables (i.e. the inherited instance variables of the subclass). The **initialize** method presented above essentially is an *incremental specialization* of the **initialize** method inherited from the superclass.

Class **Object** is the superclass of all classes and defines the protocol common to all objects [SMALLTALK, 86]. Class **Object** defines the default behavior for displaying, comparing, copying, accessing indexed instance variables and for error handling. Class **Object** includes capabilities to maintain dependency relationships between objects and to broadcast messages from an object to its dependents.

Classes are descriptions or templates from which instances are created. *Metaclasses* allow the descriptions of a class by another class, providing for greater abstraction [COINTE, 87]. The metaclass hierarchy (Fig. 4.3) (1) is parallel to the class hierarchy (2).

Class **MetaClass** is the class of all metaclasses (3). It contains the common protocol for creating classes. Every metaclass has one and only one instance which is the class of the same name (4). The metaclass contains the class methods while the class contains the instance methods.

Class **Class** is the superclass of all class classes (i.e. metaclasses) (5). It provides the common protocol for defining and accessing class variables. It also provides the implementation (methods) for the subclass messages.

Classes **Object** and **Class** are *abstract* classes: they are used to define common properties for subclasses. No instances (6) are created from abstract classes.

```
Object
    Behavior
        Class
        MetaClass
    BitBlt
        CharacterScanner
        Pen
            Animation
            Commander
    Boolean
        False
        True
    ClassBrowser
    ClassHierarchyBrowser
    ClassReader
    Collection
        Bag
        IndexedCollection
            FixedSizeCollecion
                Array
                Bitmap
                ByteArray
                    CompiledMethod
                    FileControlBlock
                Interval
                String
                    Symbol
            OrderedCollection
                Process
                SortedCollection
        Set
            Dictionary
                IdentityDictionary
                    MethodDictionary
                SystemDictionary
            SymbolSet
    Compiler
    Context
    CursorManager
        NoMouseCursor
    DemoClass
    Directory
    DiskBrowser
    Dispatcher
        GraphDispatcher
        PointDispatcher
        ScreenDispatcher
        ScrollDispatcher
            FormEditor
            ListSelector
            TextEditor
            PromptEditor
        TopDispatcher
    DispatchManager
    DisplayObject
        DisplayMedium
            Form
                DisplayScreen
                SelectorForm
        DisplayString
        InifiniteForm
    File
    Inspector
        Debugger
        DictionaryInspector
    Magnitude
        Association
        Character
        Date
        Number
            Float
            Fraction
            Integer
                LargeNegaiveInteger
                LargePositiveInteger
                SmallInteger
        Time
    Menu
    Message
    Pane
        SubPane
            GraphPane
            ListPane
            TextPane
        TopPane
    Pattern
        WildPattern
    Point
    Prompter
    Rectangle
    Stream
        ReadStream
        WriteStream
            ReadWriteStream
                FileStream
                TerminalSream
    StringModel
    TextSelector
    UndefinedObject
SwappedOutObject
```

Figure 4.2: Smalltalk/V Class Hierarchy.

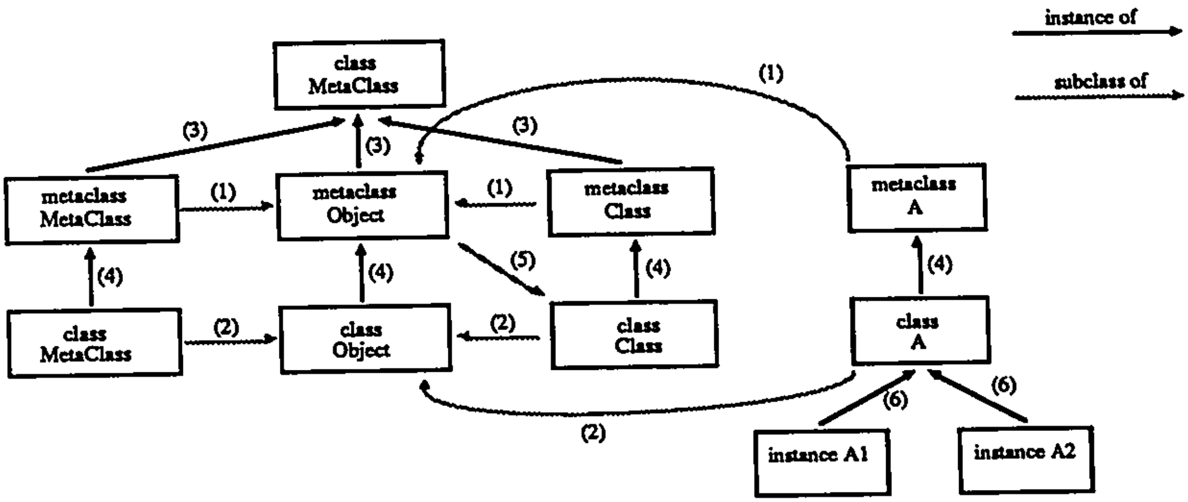


Figure 4.3: Instantiation and Classification Hierarchy.

4.6.2 The Environment

Smalltalk/V[4] is a *language-centered environment.* Language-centered environments are those in which the operating system and tool set are especially built to support a particular language [DART, 87]. Their purpose is to encourage an exploratory style of programming to aid rapid production of software. Applications are embedded in the environment and thus share the same language, both at development time and run time. The semantic information (source code and structure) of the environment is available, and applications can make use of all of the environment's facilities (e.g. window managers) by reusing parts of the environment's code. The environment itself can be extended to satisfy specific needs. Applications are usually built incrementally: first, code is "borrowed" from the environment and then, gradually refined (by specialization) as needed. Code can be developed, executed, tested, debugged and changed quickly as the need for context-switching between tools and applications is eliminated (applications being embedded in

[4]In this subsection, "Smalltalk/V" refers to the environment, by opposition to the language.

the environment).

The semantic information (methods' code and class hierarchy) of the Smalltalk/V environment is available through browsers. Browsing involves navigating through the hierarchy of classes (Fig. 4.2), with the Class Hierarchy Browser (Fig. 4.4), for accessing specific classes and then methods of classes.

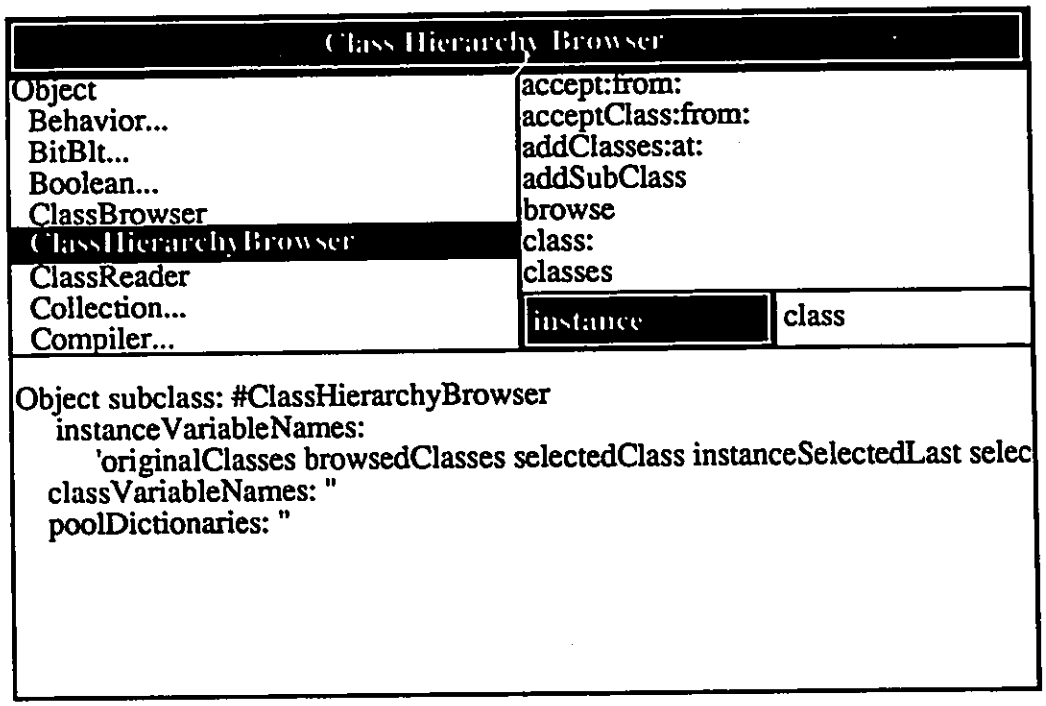


Figure 4.4: Class Hierarchy Browser.

In Smalltalk/V, fast prototyping is supported by such features as suspended time editing, incremental compilation, source code reusability and object inheritance. Suspended time editing is the ability to resume a code editing session in the state it was before being temporarily put aside to give way to some other process (browsing some other code for example). With incremental compilation, methods are compiled individually and dynamically linked or bounded to the bulk of the environment-application code. Incremental compilation reduces the impact of small code changes.

Smalltalk/V also provides the ability to examine the sequence of messages which lead to an error. The interrupted process may then be restarted from any given message of the sequence, after proper corrective actions were taken.

4.6.3 The Model-View-Controller Paradigm

The architecture of the Smalltalk/V display interface is based on the model-view-controller (MVC) paradigm (Fig. 4.5) [LONDON, 85]. A *view object* or *pane* is usually an instance of the **Pane** class. A pane is a subarea of a window. It takes care of framing, labeling, scrolling, bordering, data presentation, and transformations from local view coordinates to display coordinates. The appearance of an item of a list (normal video when non-selected and reversed video when selected) for example is defined at the level of the pane. Each pane object has a unique *controller* associated with it (i.e. the pane has a pointer to the controller, and vice-versa). A controller is usually an instance of the **Dispatcher** class. Its main function is to provide methods for processing input from the keyboard and mouse. The controller also communicates with the pane to keep its contents up to date and provides the protocol for opening, closing, activating, and deactivating a window. A *model* is the underlying program or application which communicates with the user. It may be any Smalltalk object. The model is responsible for the creation of all panes and controllers it requires. It is also responsible for the communication and synchronization among these panes and their related abstract objects. A special "dependency" relationship ties a pane to its model. When a pane's content is modified by a user for example, the underlying model is notified by a message. The model may then broadcast an update message to all its dependents, without having a priori knowledge of the number and nature of the views under its control.

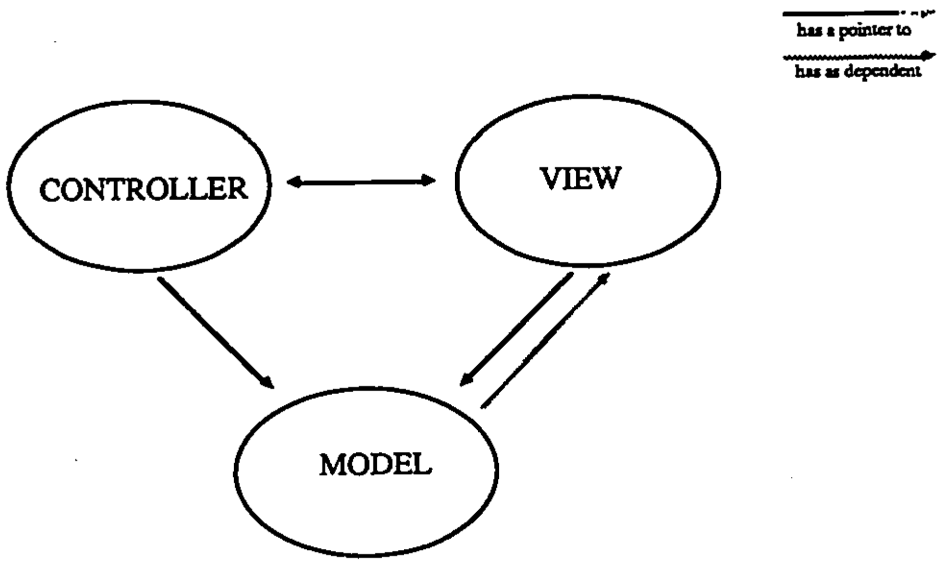


Figure 4.5: Model-View-Controller Structure.

4.7 Summary

Object-oriented programming is primarily a data abstraction technique. The primitive element of object-oriented programming is the object, a named collection of data and procedures manipulating the data. Object-oriented programming languages feature classes, instances, hierarchies of classes, and inheritance.

Smalltalk/V is both an object-oriented programming language and an environment. The source code of the environment (including the windowing code) is available and reusable by copy or by inheritance.

The environment also features suspended time editing and incremental compilation. All of the above features of the environment and of the language itself make Smalltalk/V an appropriate tool for the fast prototyping of windowing applications.

Bibliography

[COX, 86] B.J. Cox, *Object-Oriented Programming: An Evolutionary Approach,* Addison-Wesley, 274p., 1986.

[COINTE, 87] P. Cointe, "Metaclasses are First Class: the ObjVlisp Model", *OOPSLA' 87 Proceedings*, pp. 156-167, October 1987.

[DART, 87] S.A. Dart, R.J. Ellison, P.H. Feiler and A.N. Habermann, "Software Development Environments", *IEEE Computer*, Volume 20, Number 11, pp. 18-28, November 1987.

[FISCHER, 87] G. Fischer, "An Object-oriented Construction and Tool Kit for Human-Computer Communication", *Computer Graphics,* pp. 105-109, April 1987.

[LISKOV, 87] B. Liskov, "Data Abstraction and Hierarchy" *OOPSLA' 87 Addendum to the Proceedings*, pp. 17-34, October 1987.

[LONDON, 85] R.L. London and R.A. Duisberg, "Animating Programs Using Smalltalk", *IEEE Computer,* pp. 61-71, August 1985.

[PASCOE, 86] G.A. Pascoe, "Elements of Object-Oriented Programming", *BYTE Magazine*, pp. 15-20, August 1986.

[RENTSCH, 82] T. Rentsch, "Object Oriented Programming", *SIGPLAN Notices*, Volume 17, Number 9, pp. 51-57, September 1982.

[ROBSON, 81] D. Robson, "Object-Oriented Software Systems" *BYTE*, Volume 6, Number 8, pp. 74-86, August 1981.

[SEIDEWITZ, 87] E. Seidewitz, "Object-Oriented Programming in Smalltalk and ADA" *OOPSLA' 87 Proceedings*, pp. 202-213, October 1987.

[SMALLTALK, 86] *Smalltalk/V Tutorial and Programming Handbook,* Digitalk, 514p., 1986.

[STEFIK, 86] M. Stefik and D.G. Bobrow, "Object-Oriented Programming: Themes and Variations", *The AI Magazine*, Volume 6, Number 4, pp. 40-62, Winter 1986.

[STEIN, 87] L.A. Stein, "Delegation Is Inheritance" *OOPSLA' 87 Proceedings*, pp. 168-182, October 1987.

[WEGNER, 87] P. Wegner, "Dimensions of Object-Based Language Design" *OOPSLA' 87 Proceedings*, pp. 168-182, October 1987.

Chapter 5

Image Organization and Navigation Design Process

5.1 Introduction

In this chapter we first characterize the intended users of our workstation. Then we briefly describe the reading of radiographic films in particular, and analyze the way radiographic films are organized and accessed by radiologists. Finally, we present our version of image organization and navigation on a radiological workstation.

5.2 User Model

The intended users of the workstation are radiologists. Radiologists are physicians who specialize in radiographic film reading. Some specialize even further in anatomical areas such as chest or features such as bones. Radiologists also perform examinations on patients, especially in cases where the selection of the areas of interest (for which X-ray shots will be taken) involves in-depth knowledge of disease or pathologic signs (angiograms for example, where X-rays are taken while an injected radio-opaque substance is circulating in the patient's blood vessels). The film reading and reporting processes require considerable concentration from the radiologist. Variations in ambient light and noise levels are minimized in the areas where these processes take place. The working environment must also accommodate the consultations taking place between a radiologist or other radiologists, clinicians or technical staff.

Radiologists do not necessarily make use of computers in the course of their work. During patient examinations, computerized equipment may be used (e.g. for mammography or CT-scan), but is normally operated by specially trained technical personnel. Hence, as a user community characteristic, computer experience is low and may be non-existent. Initially, radiologists should be considered as novice computer users.

Radiologists are already subjected to a heavy workload. Very little time would be available for mastering the computing aspects of the workstation; therefore, voluminous user manuals and lengthy tutorial sessions are proscribed.

5.3 Radiograph Reading Process

Radiologists read radiographic images of patients and report their findings to the referring physicians. The entire patient radiograph file is presented to the radiologist together with a requisition form from the physician which states the reason for ordering the radiographic examination. All the radiographs of a patient are normally stored in one large envelope or "master envelope", within which may be found smaller envelopes or jackets. In general each jacket corresponds to one examination. The requisition statement is included in the examination jacket and could be as succinct as "? Fract LT Prox Tib" (Is left proximal tibia fractured?). To answer the requisition statement it is often necessary to inspect previous examinations. Initially, the radiologist places all the films of the latest examination on the viewboxes. If comparison with previous examinations is required, a search for relevant films begins. First of all, the radiologist may consult the list of the examinations contained in the master envelope. Secondly, he may consult the requisition forms of the older examinations. Thirdly, he may read the transcribed reports from these older examinations. Fourthly, he may partially pull-out films from their examination jackets, placing on the viewboxes those required for comparison with the present examination. At this point the set of relevant radiographs has been arranged in some manner on the viewboxes. The radiologist then reads the films and reports his findings in a general report which also responds to the requisition statement.

One can distinguish two distinct phases in the "analog" reading process. The first phase deals with the film selection and placement on the viewboxes (image organization), and the second with the actual film reading (involving image navigation).

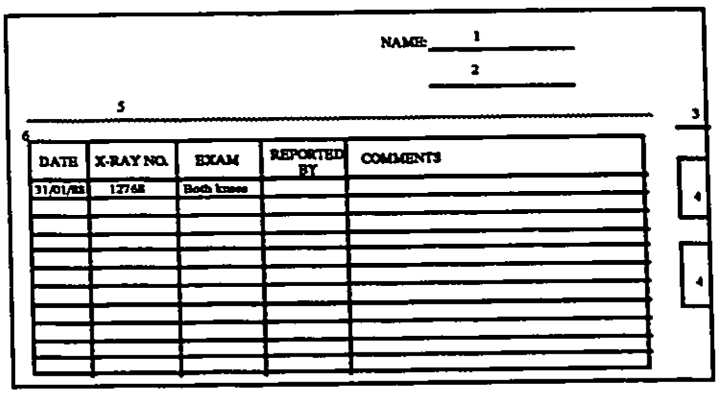
In the next section we analyze in more details the present "analog" way of handling radiographic image organization and navigation. This follows with a proposed "digital" replacement.

5.4 Image Organization and Navigation

5.4.1 Image Organization

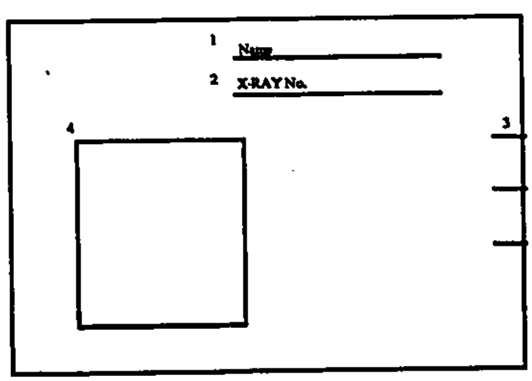
Radiographs are normally contained in a large master envelope (Fig. 5.1), within which may be found smaller envelopes or jackets (Fig. 5.2). On the master envelope is found the patient's full name along with an unique identification number, and an index listing the examinations contained therein. An examination is a set of radiographs (average 5) taken within a relatively short time span and in response to a specific requisition. The set of radiographs is collected in a jacket which is inserted into the master envelope. Each jacket also bears the patient's name and unique identification number in addition to an examination number. The requisition form is pasted onto this jacket. Examinations are listed and stored chronologically; each new examination is listed at the bottom of the index and inserted at the front of the master envelope. Within a jacket no specific order prevails for the individual images.

A number of problems characterize the "analog" image organization strategy. First of all, although the most recent examination jacket is usually found at the front of the mas-



1. Patient first and last name
2. Patient 7-digit unique identification number
3. Least significant digits (3) of the unique I.D. number
4. Color coded 4 most significant digits of the I.D. number
5. Examination reports pocket
6. List of examinations contained in the envelope

Figure 5.1: Master Envelope.



1. Patient first and last name
2. Patient 5-digit examination number
3. Patient unique identification number
4. Requisition form

Figure 5.2: Examination Envelope.

ter envelope, chronological ordering of the others is not guaranteed. Moreover, it is not uncommon to find misplaced radiographs of a particular examination in some other examination's jacket or even on occasion in some other patient's master envelope. Misplaced radiographs cause time consuming search and require extraneous envelope maintenance. Secondly, the organization is static and does not support any reorganization according to other criteria or user preferences. Indeed, during test sessions, radiologists suggested other methods of sorting examinations (other than chronological):

- by body area (e.g. HAND),
- by organ system (e.g. ABDOMEN),
- by clinical problem (e.g. PNEUMONIA).

Also, within an examination containing many views, it may be appropriate to sort the images chronologically (e.g. for time series) or in some specific order according to the examination at hand (e.g. for AP (anterior-posterior) and LATERAL CHEST pair of views, AP may always be listed first, then LATERAL). Thirdly, film manipulation is cumbersome and lengthy [ROGERS, 85]. During the reading process, many films, possibly from different examinations, are pulled out of their envelopes, only some of which will be placed on the viewboxes. A study done with CT films [ROGERS, 85] showed that about 35% of a radiologist's time is spent reviewing the patient's history, sorting and placing films on viewboxes. After report dictation the radiologist has to sort these films and place these back into their proper envelopes. Radiologists underscore the disturbance caused by physical film handling [BELANGER, 88].

5.4.2 Image Navigation

A radiographic film is read by placing it between a light source and the observer. Viewboxes are semi-transparent back lit surfaces (14 inches wide by 17 inches high). They give a fairly uniform source of light for the reading of films placed on their surface. Viewboxes are normally found in three different configurations: a single viewbox; a pair of adjacent viewboxes; or a set of 8, two rows of 4 viewboxes, one above the other. The latter configuration is the most prevalent form in radiology. Each viewbox can be illuminated independently from a set of switches. Normally, radiologists light only the viewboxes which receive a film; the others are kept turned off to avoid creating distracting light sources. The viewbox case enable radiologists to quickly refer to any image by a simple eye or head movement.

By arranging the films of interest on the viewbox case, radiologists achieve a well structured bi-dimensional indexing of these images [WILSON, 80]. Some radiologists [WITTINGHAM, 88] use the viewboxes to "reconstruct" the subject's body from the available views. However, when comparing a pair of images, radiologists place images of interest either in adjacent viewboxes or one image directly above the other since large eye movements, between distant viewboxes, are distracting to the radiologists. Also, since radiologists sit facing the center of the viewbox case, they tend to place the images of interest immediately in front of themselves (at the center of the viewbox case). In general, radiologists arrange the films of interest on the viewbox surface according to their preferences. For example, assuming four pairs of CHEST X-rays (an AP and a LATERAL view for each pair), two possible arrangements are:

- vertically, all AP's on one row and all LATERAL's on the other,
- or horizontally, an AP and a LATERAL side by side (left to right), two pairs per row.

In the next section we elaborate our electronic solution to the image organization and navigation.

5.5 Proposed Solution

Our image organization strategy attempts to solve some of the purely mechanical problems related to radiographs reading whereas the image navigation strategy mimics the conceptual aspect of viewboxes.

5.5.1 Image Organization

The proposed digital image organization strategy is based on an "anthropomorphic" organization (Fig. 5.3). The center of the Control Monitor is occupied by the sketch of a human body around which labels of images referring to the same body area are grouped. This graphical index of images provides an overview of the image contents of the patient folder. The list of image labels related to a body area is revealed by selecting the body area label (Fig. 5.4). The image labels convey the following information:

- Name of body area or procedure (e.g. "CHEST" or "CERVICAL SPINE - MYELOGRAM").
- Name of view (e.g. "LATERAL" or "AP").
- Time when the film was taken (e.g. "10:30").
- Examination number (e.g. "64220").

Labels can be sorted according to the user's preferences (e.g. chronologically or alphabetically by view). An image is displayed on the Image Monitor by selecting its corresponding label. The necessity for selecting a body area in order to access the related views, rather than permanently displaying these, is justified by the need to minimize screen clutter (information density) [SHNEIDERMAN, 87]. The thickness of the stack of buttons (round-cornered rectangles) beneath each body area label provides an indication of the number of images related to that area. As examinations usually comprise

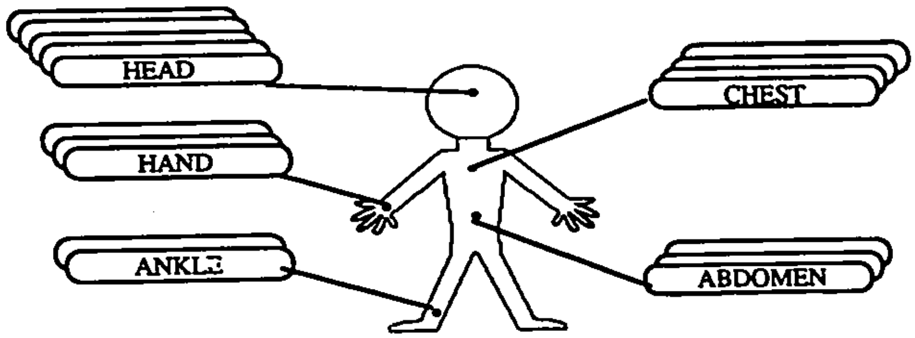


Figure 5.3: Anthropomorphic Image Organization Strategy.

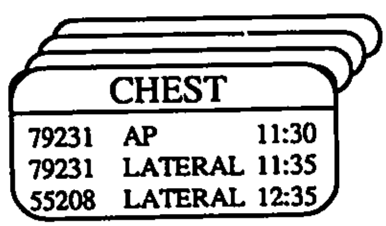


Figure 5.4: Body Area List of Views.

a standard number of views, this enables radiologists to rapidly locate areas for which the number of views differs from the norm. Such a deviation could be an indication that difficulties were encountered during the examination due to the patient's severe condition for instance.

The anthropomorphic organization of the images helps the radiologist create a mental picture of the entire radiographic file. Together with the requisition statement, a proper mental representation improves radiographic database search by constraining it to the relevance of the case under examination. This organization strategy supports different ways of sorting and classifying the images. Also, film manipulation and the need for sorting films after reporting have been eliminated which, in turn, obviates the problem of misplaced radiographs.

5.5.2 Image Navigation

Because radiologists rely on some form of spatial arrangement of the films in the analog world, a similar mechanism must be provided in the digital world. Hence, the image navigation strategy is based on a "soft" representation of the physical viewboxes (Fig. 5.5). This strategy essentially is a Spatial Data Management System (SDMS) [WILSON, 80]. Information in a SDMS is represented by icons arranged in some data space. Icons can be used as navigational aid through the database [CHANG, 85]. In general, an icon has two representations: a visual (physical) representation and an abstract (logical) representation. An icon is visually represented by some sketch which reflects the attributes of the related database entity. The abstract representation of an icon conveys the information which identifies and locates the related database entity.

The soft viewbox surface constitutes a bi-dimensional space on which the user arranges radiographs represented by icons. The image icon's visual representation is composed of alphanumeric characters and lists the name of the body area of the related radiograph,

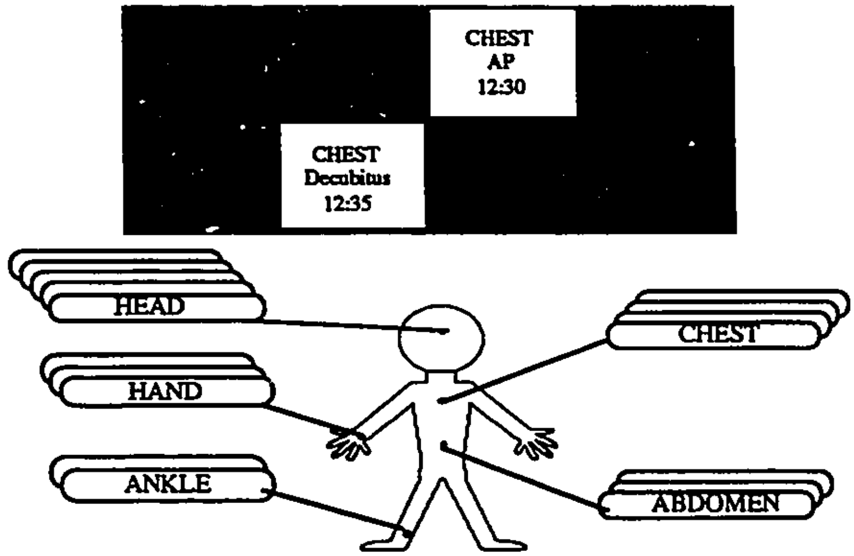


Figure 5.5: Soft Viewbox Based Image Navigation Strategy.

the name of the view and time of day it was taken. The visual representation could also be replaced by a reduced resolution view of the original image. The image icon's logical representation consists of a data structure (chapter 6) which contains the above data and storage information for radiograph retrieval.

In the same manner that radiographic films can be placed onto physical viewboxes, image labels can be dragged with the mouse onto the soft viewboxes for later viewing. Image icons placement on the viewboxes can also be rearranged by dragging icons from viewbox to viewbox. Users directly manipulate radiographs through their iconical representations which realize the semantic of their real object counterparts: physical films and viewboxes. An image is displayed on the Image Monitor by selecting its corresponding icon. In order to avoid disturbing the actual film reading process, images can also be selected for display from a replica of the viewboxes which are embedded on the Image Monitor (Fig. 5.6). Since a single pointing device covers the entire visual workspace, the cursor has to be moved over to the Image Monitor in order for the user to access various image enhancement functions (Fig. 5.6). Once the cursor is on the Image Monitor, the radiologist is able to concentrate on reading the radiographs and the associated report. To select an image which was placed previously on a viewbox, the embedded viewboxes are poped-up by clicking anywhere on the Image Monitor. Selecting an image then makes the pop-up viewboxes disappear to make way for the new image. The use of embedded viewboxes reduces the amount of interaction on the part of the radiologist.

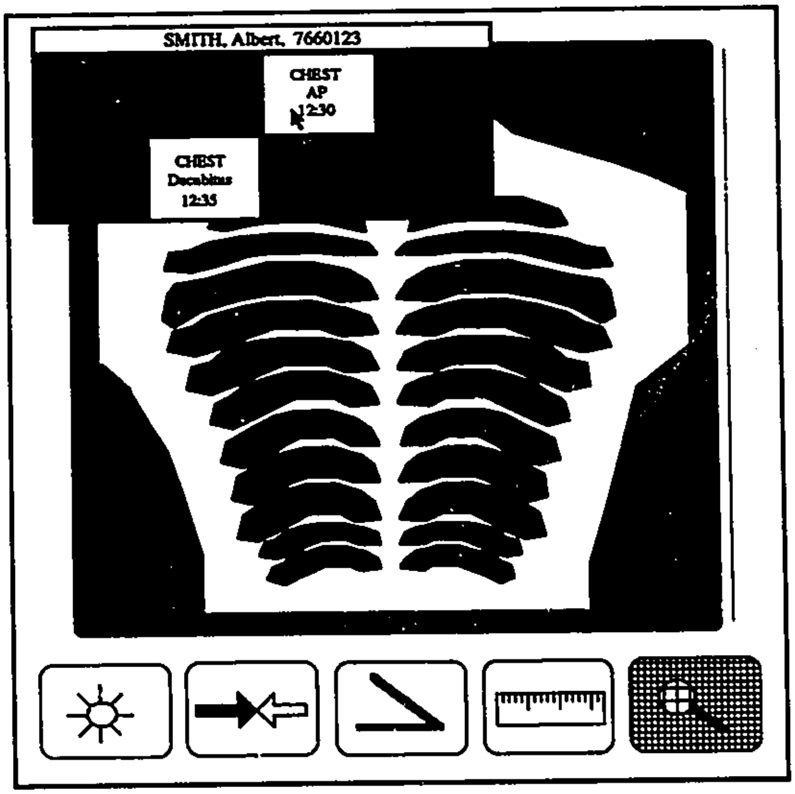


Figure 5.6: Image Monitor with Embedded Viewboxes.

5.6 Summary

In this chapter we presented our image organization and navigation strategies for a radiological image workstation.

The image organization strategy is based on an "anthropomorphic" presentation of the images (Fig. 5.3). The center of the monitor is occupied by the sketch of a human body around which are placed labels of images referring to the corresponding body areas.

The image navigation strategy is based on a "soft" representation of the physical viewboxes (Fig. 5.5). The soft viewbox surface is a bi-dimensional space on which to arrange radiographic images represented by icons.

In the following chapter we describe our implementation of the above strategies.

Bibliography

[BELANGER, 88] G. Bélanger, *Private Communications,* Ottawa Civic Hospital, Ottawa, 1988.

[CHANG, 85] S.K. Chang, "Image Information Systems", *Proceedings of the IEEE,* Vol. 73, No. 4, pp. 754-764, April 1985.

[ROGERS, 85] D.C. Rogers, R.E. Johnston, B.Brenton, E.V. Staab, B. Thompson and J.R. Perry, "Predicting PACS Console Requirements From Radiologists' Reading Habits", *SPIE Picture Archiving and Communication Systems (PACS III) for Medical Applications*, Vol. 536, pp. 88-95, 1985.

[SHNEIDERMAN, 87] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Adison-Wesley, 448p., 1987.

[WILSON, 80] G.A. Wilson and C.F. Herot, "Semantics VS Graphics - To Show or Not To Show", *Proceedings of the Sixth IEEE International Conference on Very Large Data Bases*, pp. 183-196, October 1980.

[WITTINGHAM, 88] Wittingham, *Private Communications,* Ottawa Civic Hospital, Ottawa, 1988.

Chapter 6

Implementation

6.1 Introduction

In this chapter we describe our implementation of the user interface for a radiological image workstation. First we present the hardware configuration of the workstation, and then the architecture of the user interface's software and its various constituents.

6.2 Hardware Configuration

The hardware configuration of the workstation is shown in Fig. 6.1. The core of the workstation is an AT-compatible computer equipped with a large hard disk (for image storage). The visual workspace is composed of two monitors: a low-resolution Control Monitor (640 x 350 x 1), and a high-resolution Image Monitor (1024 x 1280 x 8). A single mouse covers these two monitors. This reinforces the fact that the two monitors form a unified visual workspace. The Control Monitor is driven by an EGA controller; the Image Monitor by an Imagraph imaging board. The Imagraph board has a two plane image memory: an 8-bit main-memory plane, and a 2-bit overlay plane. Both planes are 1024 pixels wide by 2048 pixels high although at most 1024 by 1280 pixels can be displayed at a time. Radiographs are displayed in the main-memory plane. The image processing icons, the embedded viewboxes, and the cursor (Fig. 5.6) are displayed in the overlay plane.

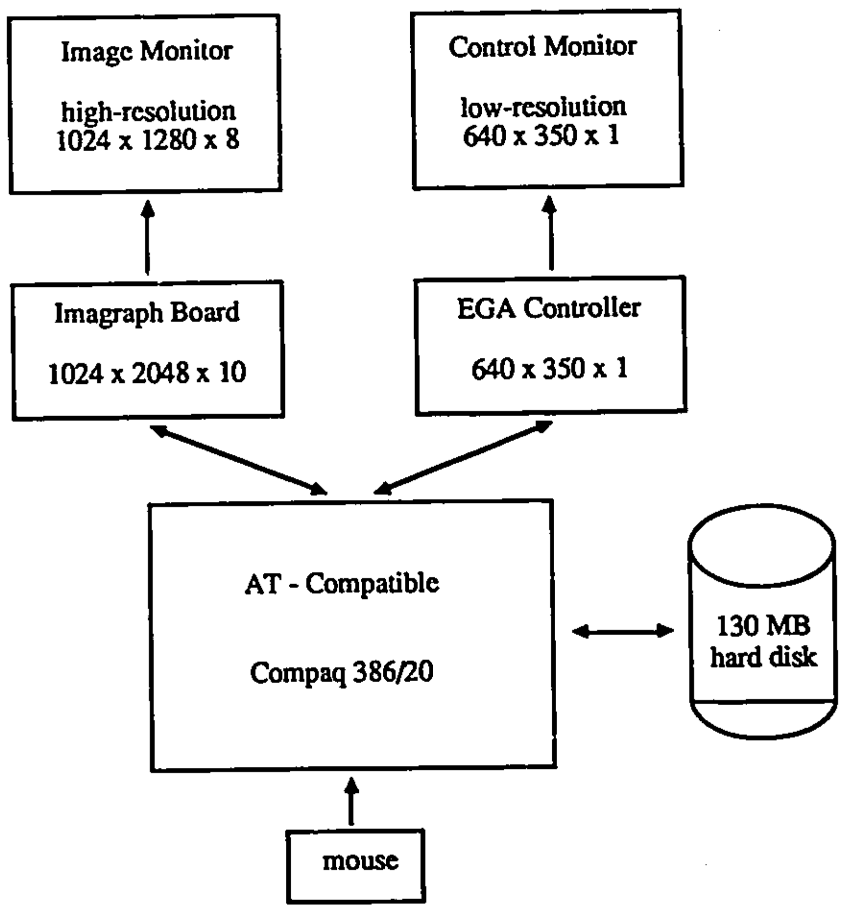


Figure 6.1: Workstation Hardware Configuration.

6.3 Software Architecture

The software architecture of the user interface, as modelled by Lantz (Section 2.4), is shown on Fig. 6.2. The software was mostly written in Smalltalk/V (Section 4.6). Smalltalk/V was selected as the software environment and development language for the following reasons:

1. The workstation is based on the IBM-PC computer family.

2. MS-DOS was originally selected as the workstation's operating system.

3. A flexible windowing package was required for fast prototyping.

Hence, the software architecture of the user interface is strongly influenced by the architecture of Smalltalk/V itself. Indeed, the usual programming technique in Smalltalk/V consists in refining the existing source code supplied with the language.

The user interface per se consists of three components: the workstation agent, the dialogue manager, and the workstation manager.

6.3.1 Workstation Agents

The two main workstation agents are the TerminalStream and the BitBlt classes.

TerminalStream

TerminalStream uses a finite state machine to decode the user input from the keyboard and mouse and provides a simple interface to fetch user input.

BitBlt

Bit block transfers (BitBlt) are the fundamental graphic operations required for windowing software.

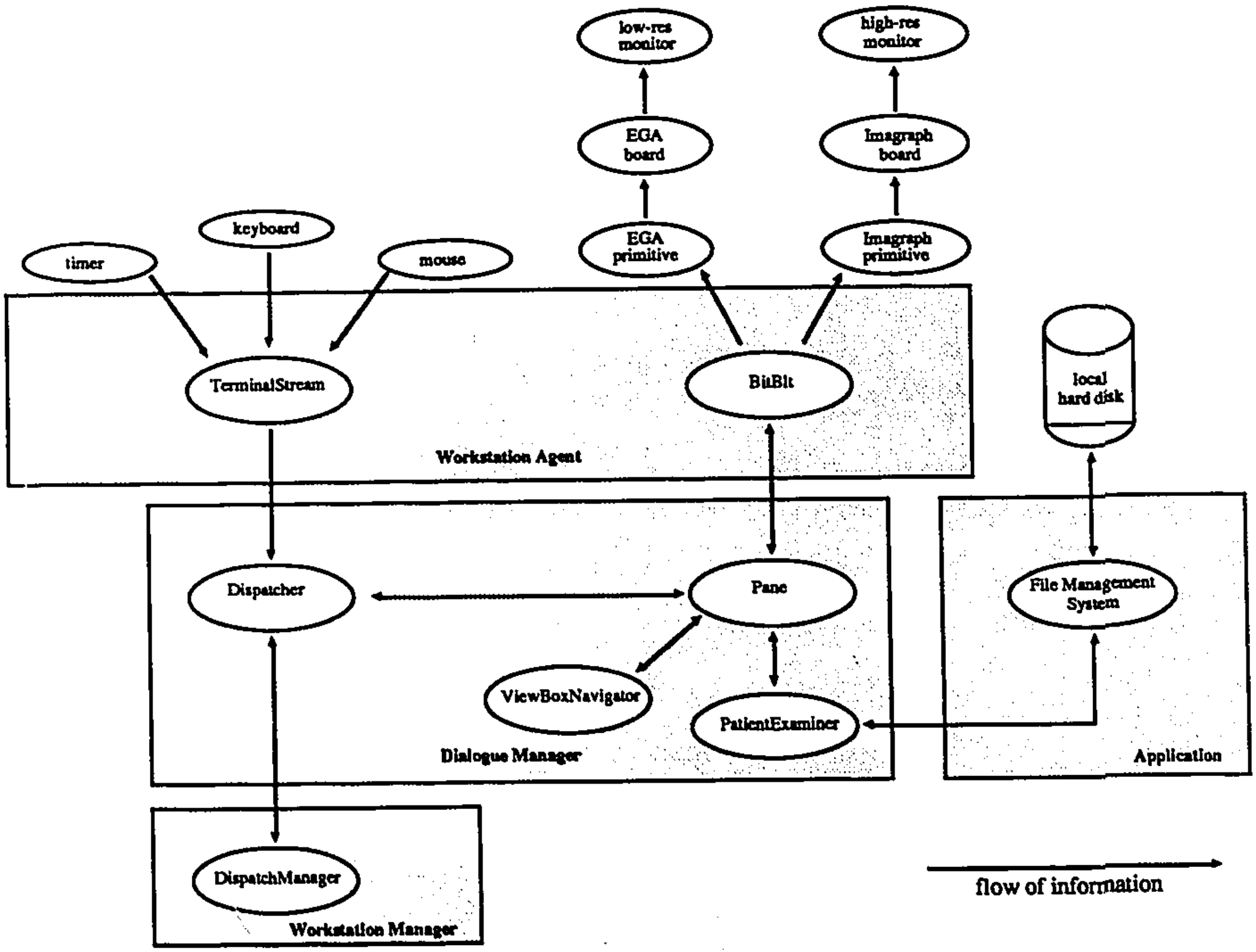


Figure 6.2: Software Architecture.

Both the Control and the Image Monitors are ***bit-mapped*** screens. A screen is bit-mapped when the information represented by its entire grid of pixels (**picture elements**) is fetched from the content of some dedicated memory (referred to as "display screen memory").

In Smalltalk/V, bitmaps (bi-dimensional arrays of bits) are held by **Form** objects. Forms describe two types of hardware: for bitmaps contained in the main memory (therefore not displayed) and for bitmaps stored in the display screen memory (hence visible on the display). All graphics operations involve forms.

The **BitBlt** class groups together all the basic graphic operations. Its main function is to transfer bits from one area to another. This involves three forms: the source form, destination form, and mask form. The bits in the source form are first ANDed with the bits in the mask form, and then merged into the destination form with a combination rule (Fig. 6.3). The combination rule specifies a logical operation (e.g. AND, OR, XOR, etc.) which combines a source bit with its corresponding destination bit. The areas involved in this bit transfer are specified by a source rectangle on the source form, a destination rectangle on the destination form, and a clipping rectangle also on the destination form (Fig. 6.4). After aligning the source rectangle origin with the destination rectangle origin, the resulting effective area corresponds to the intersection of all three rectangles and the extents of the source and destination forms.

Since the original BitBlt only handled forms for the Control Monitor (forms of bitmaps stored in the main memory or in the display screen memory), an extension was written (partly as modifications to the BitBlt class itself, and partly as a 'C' language primitive) to accommodate forms describing the bitmaps for the overlay plane on the Imagraph imaging board (Image Monitor).

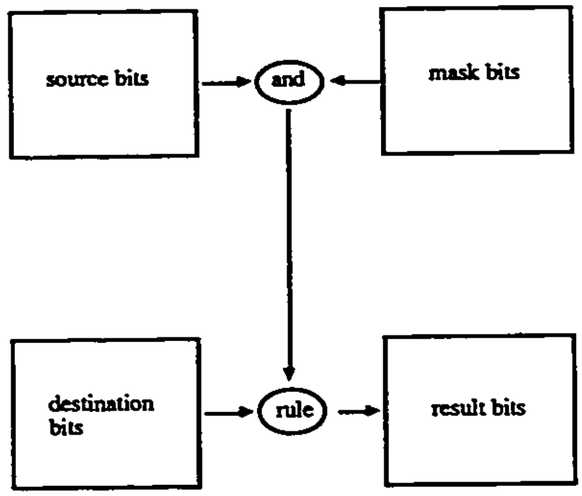


Figure 6.3: BitBlt Operations.

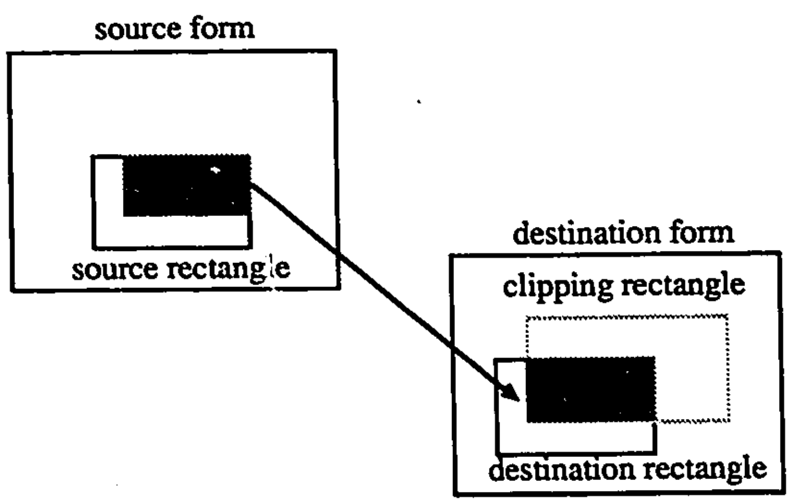


Figure 6.4: BitBlt Clipping.

6.3.2 Workstation Manager

The workstation manager is personified by the Scheduler, the only instance of the DispatchManager class. The Scheduler schedules windows by providing methods for adding and removing windows, displaying all the windows, or activating a specific window.

6.3.3 Dialogue Managers

The user communicates with the system through a collection of active *objects* represented on the display monitors (e.g. an image label on the Control Monitor). Each monitor of the workstation is controlled by a separate dialogue manager. The PatientExaminer and the ViewBoxNavigator are the main classes (models) of the two dialogue managers. Central to these two models are the ImageFolder class of abstract objects.

ImageFolder

An ImageFolder object (Fig. 6.5) is created for each radiograph of a patient. Each object contains the name of the body area and the view of the radiograph, the time of day when it was taken, the location of storage of the image in the system (file server, local hard disk or ram disk), and the examination number. ImageFolder objects may have more than one representation (Fig. 6.6): on viewboxes of the Control Monitor, on viewboxes of the Image Monitor, on a body area list of the Control Monitor. A user may then create as many different views as necessary for the same ImageFolder object.

name of body area
name of view
time of day film was taken
location of image in system
examination number

Figure 6.5: ImageFolder Object.

PatientExaminer

The PatientExaminer is the model which takes care of the functions handled on the Control Monitor (Fig. 6.6). It is responsible for the organization of the image labels and supports their placement on the viewboxes.

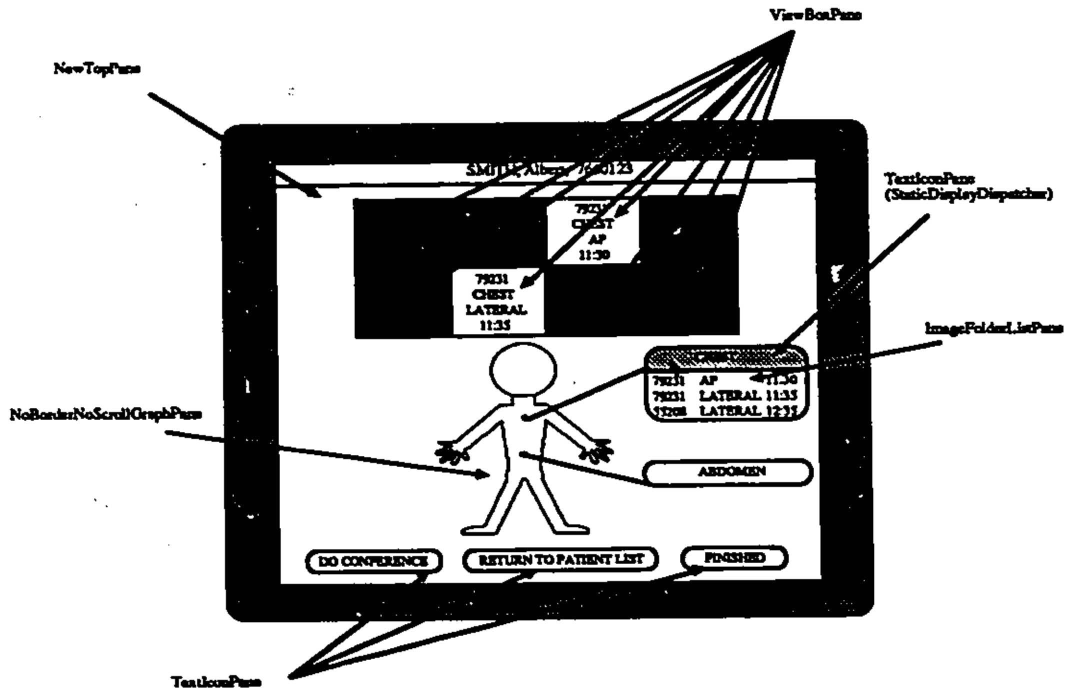


Figure 6.6: PatientExaminer Model.

ViewBoxNavigator

The ViewBoxNavigator is the model which handles the embedded viewboxes of the Image Monitor (Fig. 5.6).

Models Inter-Communication and Operation

Upon instantiation of the PatientExaminer model, all ImageFolder instances of the patient are then linked as dependents to the model. Only the ImageFolders of images placed on viewboxes by the radiologist are linked as dependents to the ViewBoxNavigator model. Hence, through the use of dependency relationships, both models operate on the same ImageFolder objects, without the need for duplication. The consistency of the ImageFolder objects is thus maintained (e.g. if an image is displayed from the ViewBoxNavigator, the PatientExaminer will be able to reflect this fact).

When an image icon is dragged from an ImageFolderListPane to a viewbox, the ImageFolderListPane sends a message to inform the PatientExaminer. The PatientExaminer queries the ViewBoxPane's dispatchers to identify the destination viewbox, and then adds the related ImageFolder to its list of dependents (the ImageFolder then adds the PatientExaminer to its own list of dependents). If an image icon is dragged from a viewbox to an other viewbox, the model empties the source viewbox and gives the image icon to the destination viewbox. When the user selects an image icon, the view object (ViewBoxPane or ImageFolderListPane) sends a message to inform the model. The model then retrieves the radiographic image from the database using the information provided by the ImageFolder object and displays it on the Image Monitor. The model also sends an update message to the appropriate ImageFolder object which then broadcasts the update message to all its dependent models. These models then update the views under their control to reflect the fact that the appropriate image is displayed on the Image Monitor.

6.4 Summary

In this chapter we presented our implementation of the image organization and navigation strategies for a radiological image workstation.

The radiological workstation is equipped with two monitors: a low-resolution Control Monitor, and a high-resolution Image Monitor. The software (user interface) is written for the most part in Smalltalk/V, a flexible environment and language for fast prototyping.

In the following chapter we describe the field tests which were conducted at the Ottawa Civic Hospital and present the results thereof.

Chapter 7

Field Tests

7.1 Introduction

Field testing is the last step closing the loop of the iterative design process (Section 2.3). Feedback from the users enables the designer to assess the correctness of the user models, tasks models and design choices. A workstation was installed in the radiograph reading room of the Department of Radiological Sciences at the Ottawa Civic Hospital. Radiologists and emergentologists were invited to participate in a series of supervised tests of the image organization and navigation aspects of the user interface. The tests were conducted over a two-month period (June - July, 1988).

7.2 Objectives

The objectives of the series of tests were as follows:

1. Determine the appropriateness of the anthropomorphic image organization strategy for organizing the images in patients files (Subsection 5.5.1).

2. Determine the relevance of the soft-viewbox based image navigation strategy (Subsection 5.5.2).

3. Refine and acquire additional knowledge regarding how radiologists and emergentologists read radiographs (Section 5.3).

4. Assess the differences in user interface requirements (functionality) and preferences between radiologists and emergentologists.

5. Verify the efficacy and satisfaction with some accessory issues (e.g. mouse sensitivity, visual feedback, etc...) regarding the human factors of the user interface in general.

7.3 Method

The tests were divided into two parts: a pre-experiment and an experiment. The pre-experiment introduced some basic computer terminology (e.g. mouse click, scrolling, etc...) and gave the participants (radiologists and emergentologists) practice using the corresponding skills until they showed sufficient proficiency. A specially written software package was employed for the pre-experiment rather than the actual user interface. The experiment covered all aspects of the user interface while concentrating on the image organization issue.

The pre-experiment and the experiment were performed together. In order to ensure consistency of the tests over time, a script outlining the functions and presentation order was written and rehearsed for each part of the tests. Any errors or any awkwardness the subject experienced while doing the various tests were noted. Formal questionnaires were also completed by participants.

7.4 Pre-Experiment

7.4.1 Objective

The object was to teach emergentologists and radiologists the computer terminology and skills required to participate in the experiment. The terms and skills were: (1) moving the mouse; (2) mouse "clicking"; (3) mouse "dragging"; and (4) scrolling.

7.4.2 Method

Subjects

Radiologists and emergentologists were invited to volunteer to participate in an experiment which would last approximately one hour (duration of the pre-experiment and experiment combined). Eleven (11) radiologists and four (4) emergentologists served in the experiment. The radiologists ranged in experience from residents to highly experienced radiologists with particular subspecialties.

The purpose of the pre-experiment was to establish a common terminology and to ensure that the set of necessary skills were understood and mastered by all subjects. Five subjects had prior experience with computers, and only three had used a mouse. Because there was such a diversity in computer use and knowledge in the subject sample, all subjects (15) were required to run through the pre-training.

Procedure

A software package was prepared[1], using HyperCard on the Apple's Macintosh microcomputer, for teaching the four terms and skills: moving (Fig. 7.1), clicking (Fig. 7.2), dragging (Fig. 7.3) and scrolling (Fig. 7.4). An experimenter controlled the learning session following a predetermined script (Appendix A). Each skill was repeated by the subject until sufficient confidence and proficiency was shown. The experimenter was responsible for assessing the level of confidence and proficiency reached by the subject, and determined when the competence level was sufficient to move to the next skill.

Once the subject had mastered all four skills, he was invited to sit at the workstation and participate in the experiment.

[1]The software package was designed and implemented by the Psychologists Group.

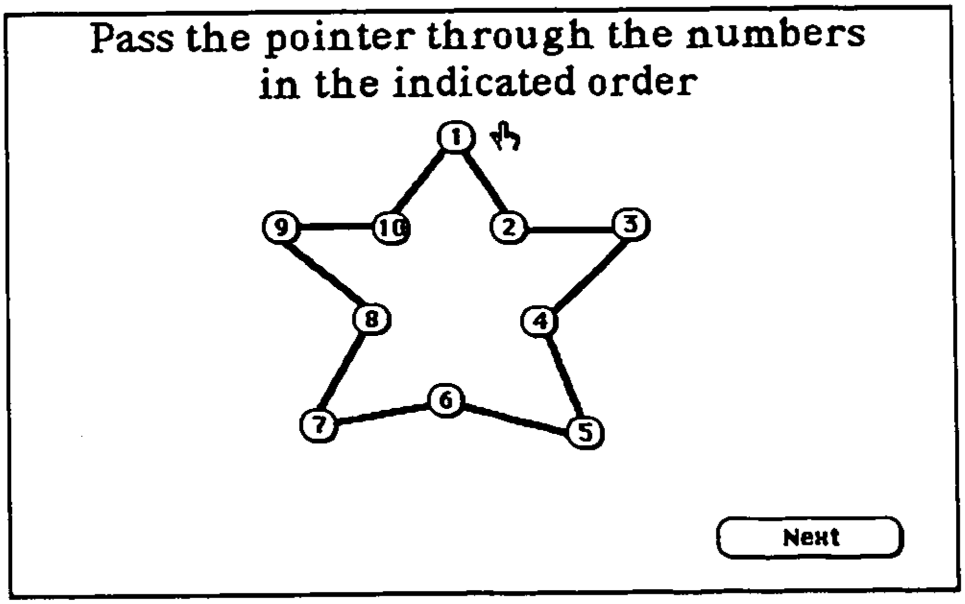


Figure 7.1: "Mouse Moving" Practice Screen.

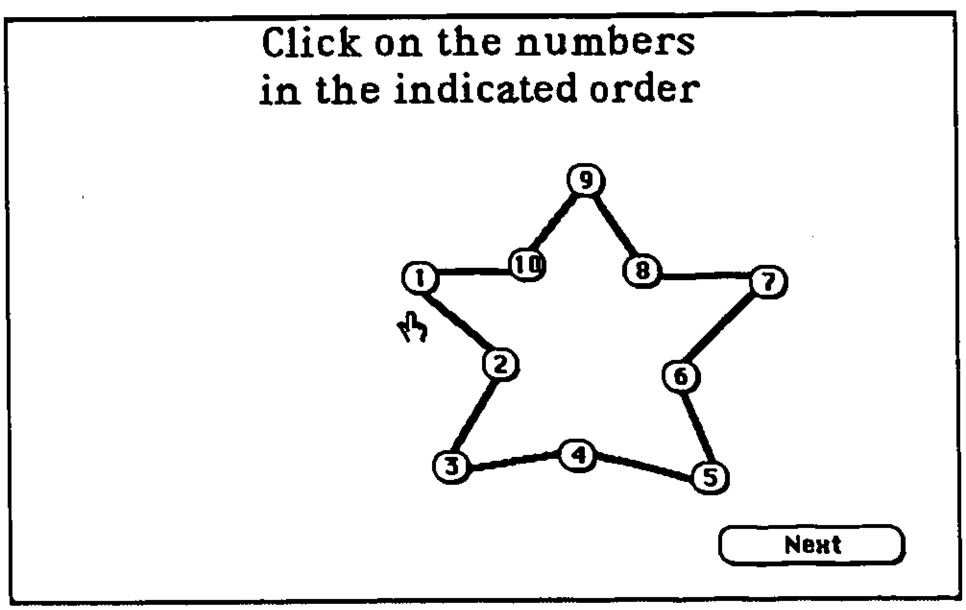


Figure 7.2: "Mouse Clicking" Practice Screen.

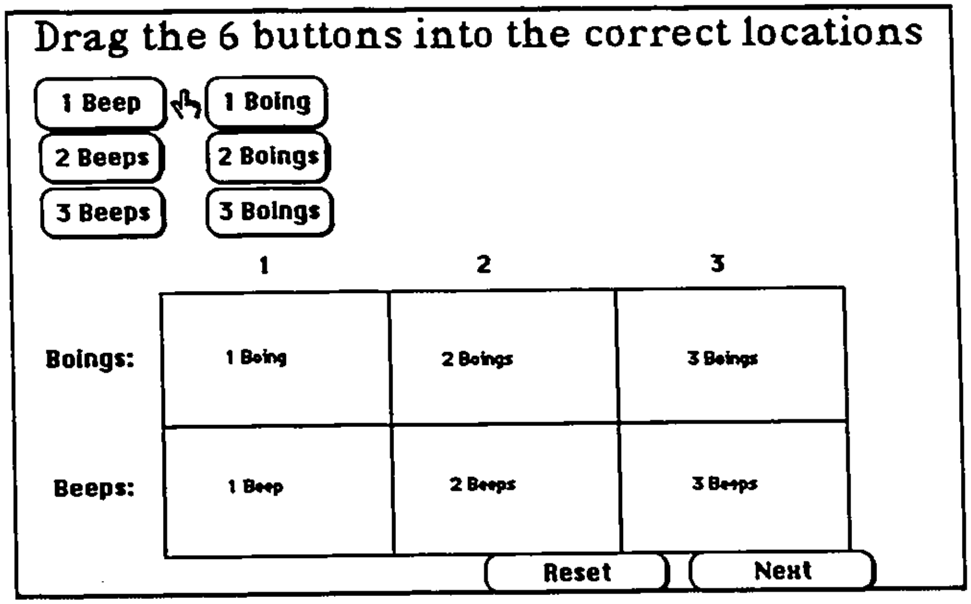


Figure 7.3: "Mouse Dragging" Practice Screen.

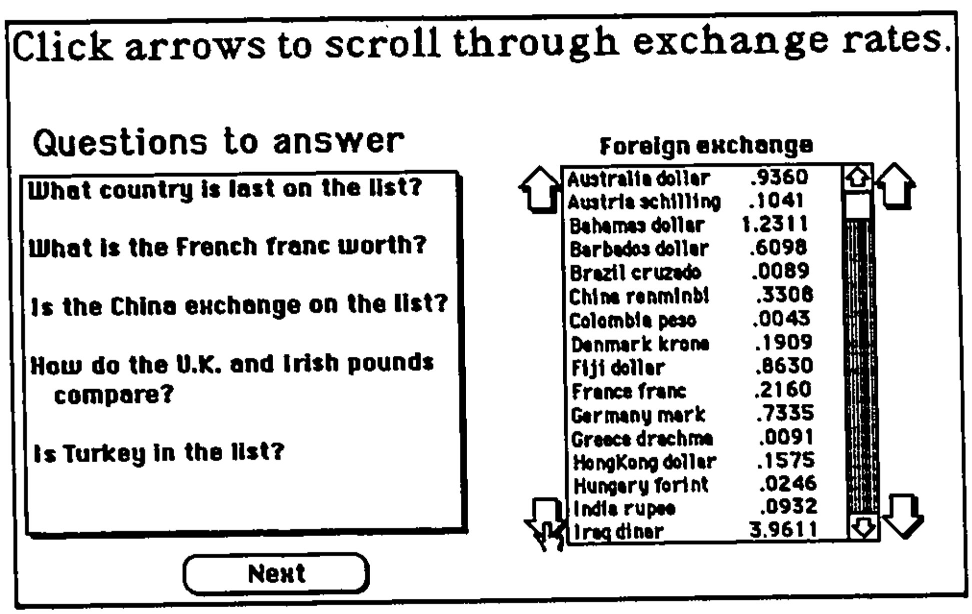


Figure 7.4: "Scrolling" Practice Screen.

7.5 Experiment

7.5.1 Objective

In this session, the subject was tutored on all facets of the user interface. The appropriateness of the anthropomorphic image organization strategy was tested against a linear image organization strategy. The usefulness of the soft-viewbox based image navigation strategy was investigated. In addition, some technical aspects of the interface were evaluated (e.g. mouse sensitivity, visual feedback, etc...).

7.5.2 Method

Material

Twenty radiographic images were used which were sorted into 5 different body areas for the anthropomorphic organization: chest, cervical spine, cervical spine (myelogram), lumbar spine, and lumbar spine (myelogram) (Appendix B). When preset in a linear organization, the scrolling function was used because the linear list could display at most 8 items at once (Fig. 7.6).

Procedure

Initially, the subject sat at the workstation (Section 6.2) and observed as the experimenter demonstrated a session covering various facets of the user interface following a written script (Appendix C). The intent of the script was not to reproduce a realistic environment where the subject would actually report about radiographic films. Rather the script introduced the subject to all the functional aspects of the interface, and demonstrated their use. Usage of the script also guaranteed consistency of topics and order of presentation of topics for each subject. Following the demonstration, the subject was required to participate in a repetition of the demonstration session, using each of the functions, but always remaining under the direction of the experimenter. The average duration of the pre-experiment and experiment combined was 45 minutes.

While the subject used the system, any questions which arose were noted and errors and hesitations were logged. At the end of the session the subject was asked to answer a number of questions from a formal questionnaire (Appendix D). The questions queried acceptability and usefulness of the specific functions and probed the subject's impressions of the organizational and technical aspects of the interface.

The appropriateness of the anthropomorphic image organization strategy (Fig. 7.5) was assessed by comparing it against a "linear" organization strategy (Fig. 7.6). The anthropomorphic strategy was designed to present a list of radiographs as a number of distinctive sublists of radiographs sorted by body parts or locations and thus to give some visual clues about the content of the patient file. The linear organization simply presented all of the films in a single scrolling list.

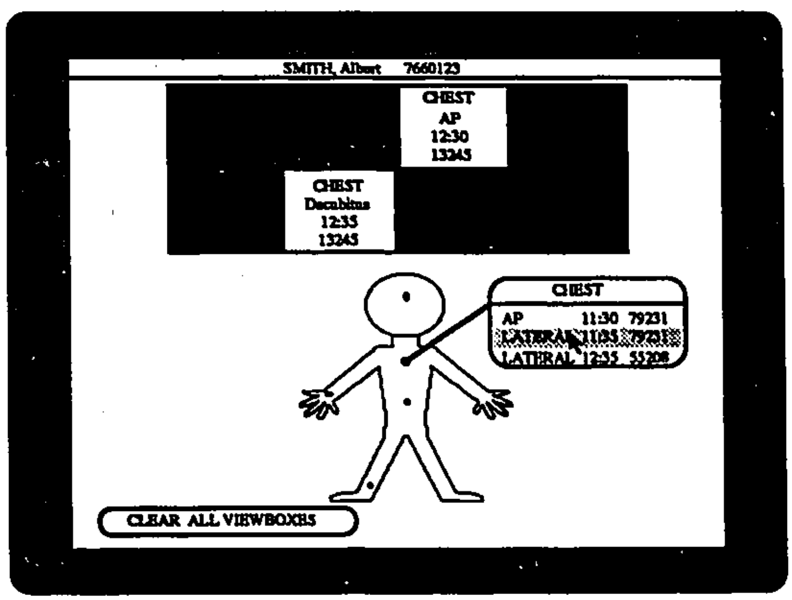


Figure 7.5: Anthropomorphic Image Organization.

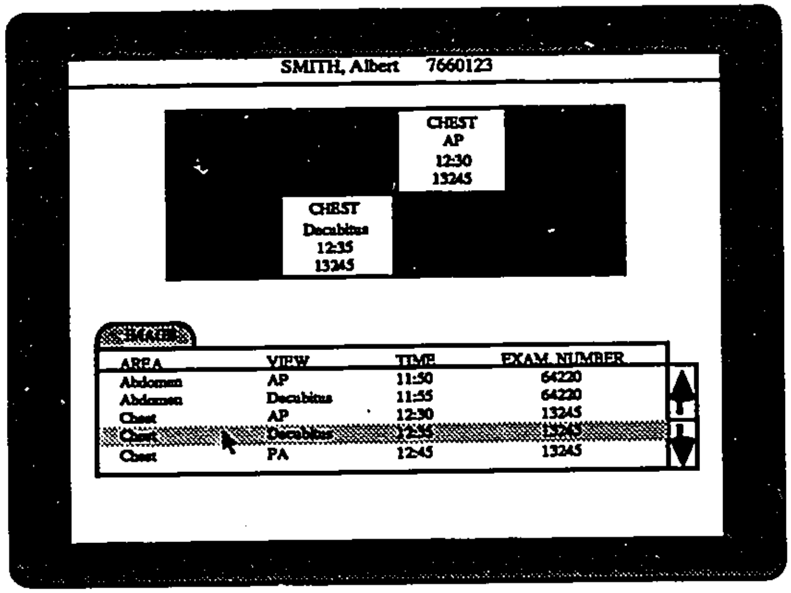


Figure 7.6: Linear Image Organization.

Each subject used both organization strategies. Half of the subjects experimented using the linear strategy first, and then the anthropomorphic strategy; and vice-versa for the other half of the sample. Separating the sample into two counterbalanced groups controlled for the possibility of confounding the preferences of each of the strategies with the "order effects" and with learning.

7.5.3 Results

The following results are based on responses from the questionnaire used at the end of the session.

Linear Image Organization Strategy

Questions 1 through 6 of the questionnaire (Table 7.1 and Fig. 7.7) asked subjects to rate aspects of the linear image organization strategy. Ratings were on a scale of 1 to 5. For the first five questions the scale was such that 1 meant "not at all a problem" and 5, "very much a problem"; for question 6, 1 meant "very easy" and 5, "very difficult."

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
1. Forgetting items	1.5	1.3	1.5
2. Scrolling wrong way	1.6	1.5	1.6
3. Trouble finding image	1.4	1.5	1.4
4. Getting images content overview	2.8	1.8	2.5
5. Single-item scrolling	3.0	1.8	2.7
6. Scrolling (overall)	2.9	1.5	2.5

Table 7.1: Linear Image Organization Strategy (distribution averages).

For both classes of subjects (radiologists and emergentologists), the following issues presented no difficulties: forgetting that the list had more items than could be seen at once (μ_1^t[2] $= 1.47$), scrolling in the wrong direction ($\mu_2^t = 1.6$), and finding a particular image ($\mu_3^t = 1.4$). For radiologists, getting an overview of the image content of the patient file (μ_4^r[3] $= 2.8$), being able to scroll only one item at a time (one item per click) ($\mu_5^r = 3.0$), and the overall scrolling technique ($\mu_6^r = 2.9$) were felt to be somewhat problematic. Emergentologists rated the same issues as less problematic than radiologists did (μ_4^e[4] $= 1.8$; $\mu_5^e = 1.8$; $\mu_6^e = 1.5$).

Question 10 addressed the order of presentation of the images on the list. The order of films used in the experiment (alphabetical by body area, then alphabetical by view, then numerical by examination number, and finally chronological) was believed to be appropriate by most emergentologists (3 of 4). Radiologists thought it was appropriate to first sort the images alphabetically by body area (one radiologist suggested to sort anatomically, from head to toe). Then, chronological ordering, followed by views ordering, was preferred by 3 radiologists. It was recommended that the sorting of the views be

[2] μ_i^t is the mean of the ratings from both radiologists and emergentologists for question i.
[3] μ_i^r is the mean of the radiologists' ratings for question i.
[4] μ_i^e is the mean of the emergentologists' ratings for question i.

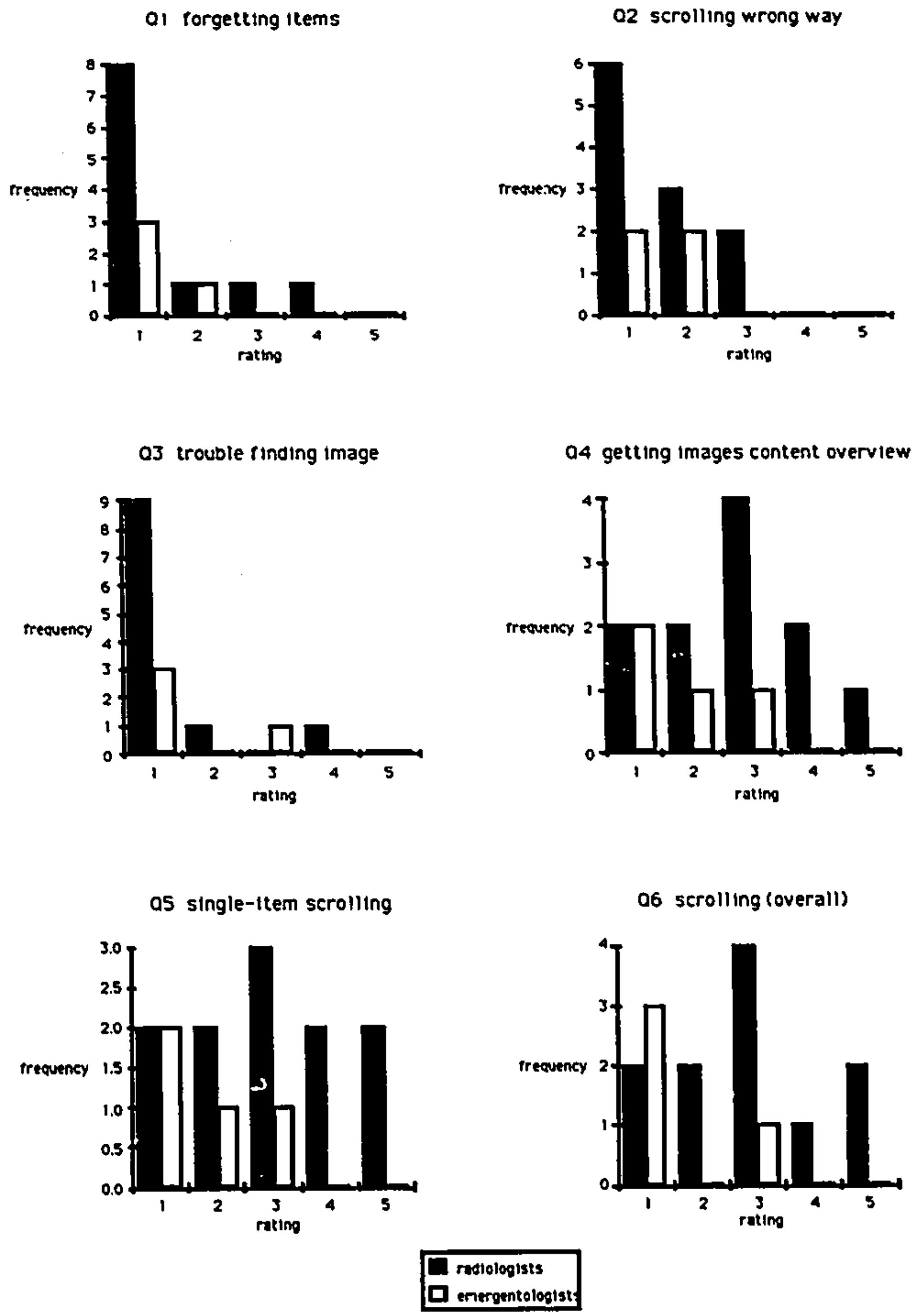


Figure 7.7: Linear Image Organization Strategy Issues.

Subject Sample: 11 radiologists, 4 emergentologists.
Q1-Q5 Rating Scale: from "not at all a problem" (1) to "very much a problem" (5).
Q6 Rating Scale: from "very easy" (1) to "very difficult" (5).

done according to standard medical practice by 3 radiologists. For example, there are radiological procedures which are labelled by a time stamp. IVP (Intra-Venous Pylogram) is a procedure which has four time-separated film series which traces the concentration and flow of a contrast medium through the intestinal track. The views are labelled as "3 minutes", "8 minutes", "15 minutes", and finally a view called "void" which is taken after the patient has eliminated the medium from his/her system. Assuming this procedure was performed a number of times on a given patient, alphabetical ordering of all the views would have two undesired effects: to separate the views of each examination, and to present the "15 minutes" views before all other ones. The first undesired effect may be solved by chronologically ordering the views. The second, by sorting according to medical practice. Hence, the "15 minutes" view would be properly presented between the "8 minutes" and the "void" views of its examination.

Anthropomorphic Image Organization Strategy

Questions 11 through 15 (Table 7.2 and Fig. 7.8) judged some aspects of the anthropomorphic image organization strategy using a rating scale of 1 to 5. For questions 11 to 14, 1 meant "not at all a problem" and 5, "very much a problem"; for question 15, 1 meant "very easy" and 5, "very difficult."

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
11. Selecting wrong category	1.1	1.3	1.1
12. Double clicking	2.1	2.0	2.1
13. Trouble finding image	1.2	1.5	1.3
14. Getting images content overview	1.7	1.8	1.7
15. Using body parts divided list	1.0	1.0	1.0

Table 7.2: Anthropomorphic Image Organization Strategy (distribution averages).

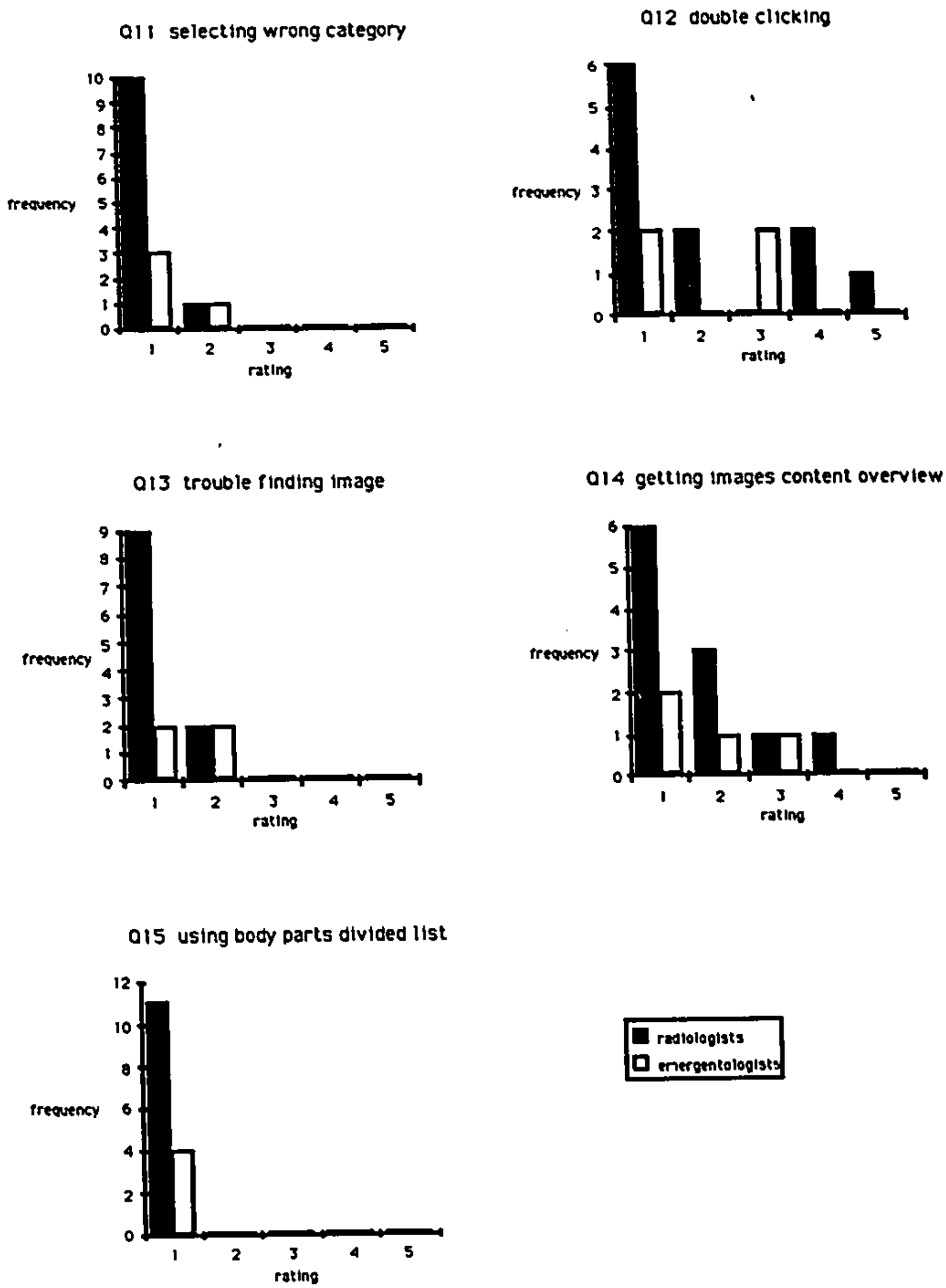


Figure 7.8: Anthropomorphic Image Organization Strategy Issues.

Subject Sample: 11 radiologists, 4 emergentologists.
Q11-Q14 Rating Scale: from "not at all a problem" (1) to "very much a problem" (5).
Q15 Rating Scale: from "very easy" (1) to "very difficult" (5).

For both classes of subjects, the following issues presented no difficulties: selecting a category or body area to find a specific image ($\mu^t_{11} = 1.1$), finding a particular image ($\mu^t_{13} = 1.3$), getting an overview of the image content of the patient file ($\mu^t_{14} = 1.7$), and using the list divided by body parts ($\mu^t_{15} = 1.0$). Having to click once to unfold the list of views for a body area and then to click once more to select a particular view ($\mu^t_{12} = 2.1$) was rated as a moderate problem by both classes of subjects.

Question 20 referred to the sorting order of the images for a given body area. The sorting order used in the experiment (alphabetical by view, then numerical by examination number, and finally chronological) was thought to be appropriate by all emergentologists. More than half the radiologists (7 of 11) insisted that images belonging to the same examination should not be separated by images from other examinations. Examinations themselves should be sorted chronologically. Views of an examination may be sorted alphabetically or according to standard medical practice.

Questions 30 and 31 (Fig. 7.9) pertained to the human sketch in the center of the Control Monitor. Subjects were asked to answer "yes" or "no" to both questions. The subjects were equally divided as to the usefulness of the sketch for quick film location (Q30). Two thirds of the subjects believed the sketch was helpful in giving them an overview of the film content of the patient file (Q31).

Linear Versus Anthropomorphic

Questions 22 through 25 (Table 7.3 and Fig. 7.10) compared the linear versus the anthropomorphic image organization strategies. Ratings were on a scale of 1 to 5, where 1 meant "much prefer linear" and 5, "much prefer anthropomorphic."

No systematic preferences emerged from the emergentologists' answers to all four comparison features: overall mouse usage ($\mu^e_{22} = 2.8$), locating specific images ($\mu^e_{23} = 2.5$), obtaining an overview of the image content of the patient file ($\mu^e_{24} = 2.8$), and setting

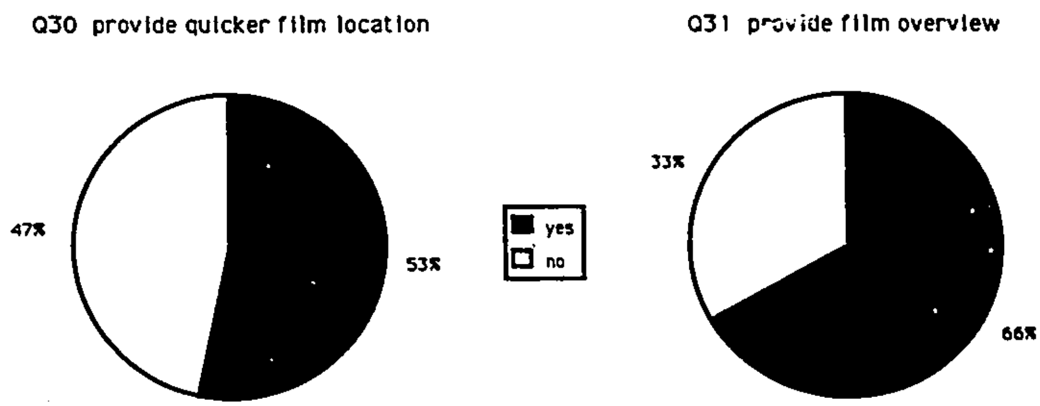


Figure 7.9: Helpfulness of Human Sketch in Anthropomorphic Strategy.

Subject Sample: 11 radiologists, 4 emergentologists.

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
22. Mouse use	3.7	2.8	3.5
23. Locating images	3.8	2.5	3.3
24. Getting file overview	3.8	2.8	3.5
25. Setting up viewboxes	3.0	2.5	2.7

Table 7.3: Linear Versus Anthropomorphic Strategies (distribution averages).

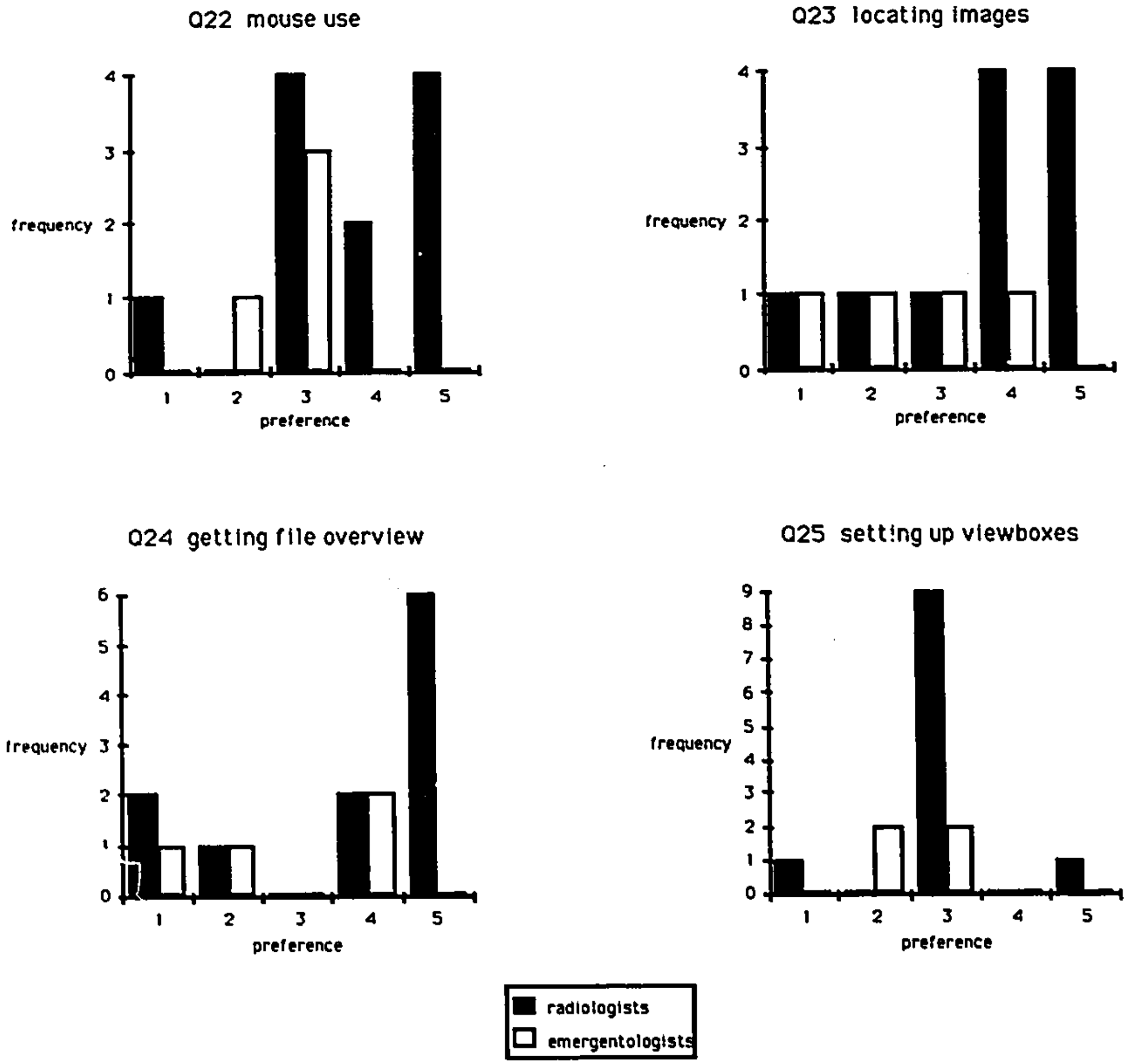


Figure 7.10: Linear Versus Anthropomorphic Strategies.

Subject Sample: 11 radiologists, 4 emergentologists.
Reference Scale: from "much prefer linear" (1) to "much prefer anthropomorphic" (5).

up the viewboxes (μ^e_{25} = 2.5). Radiologists' answers showed a preference towards the anthropomorphic strategy (μ^r_{22} = 3.7 ; μ^r_{23} = 3.8 ; μ^r_{24} = 3.8), except for setting up the viewboxes, in which case both strategies were considered equivalent (μ^r_{25} = 3.0).

Viewboxes

Questions 35 through 37 (Table 7.4 and Fig. 7.11) rated the usage of the viewboxes. Ratings were on a scale of 1 to 5, where 1 meant "very easy" and 5, "very difficult."

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
35. Dragging from list to viewbox	1.8	1.0	1.6
36. Dragging from viewbox to viewbox	1.5	1.0	1.3
37. Displaying viewboxes on Image Monitor	1.4	1.0	1.3

Table 7.4: Initial Reaction to the Viewboxes (distribution averages).

The subjects judged the following skills as very easy to easy: dragging image information from a list to a viewbox (μ^t_{35} = 1.6), dragging image information from a viewbox to another viewbox (μ^t_{36} = 1.3), and "popping-up" the set of viewboxes on the Image Monitor (μ^t_{37} = 1.3).

An image label in a viewbox listed the name of the body area, the name of the view, the date and the examination number (Fig. 7.5). Two thirds of the subjects (10 of 15) thought the information was satisfactory (Q38). Almost all subjects (13 of 15) believed the information was properly presented (organized) (Q39).

Mouse, Monitor and Cursor

Questions 40 through 50 (Table 7.5, Fig. 7.12 and 7.13) were designed to capture the impressions of the subjects on various issues such as using the mouse, the monitors, and

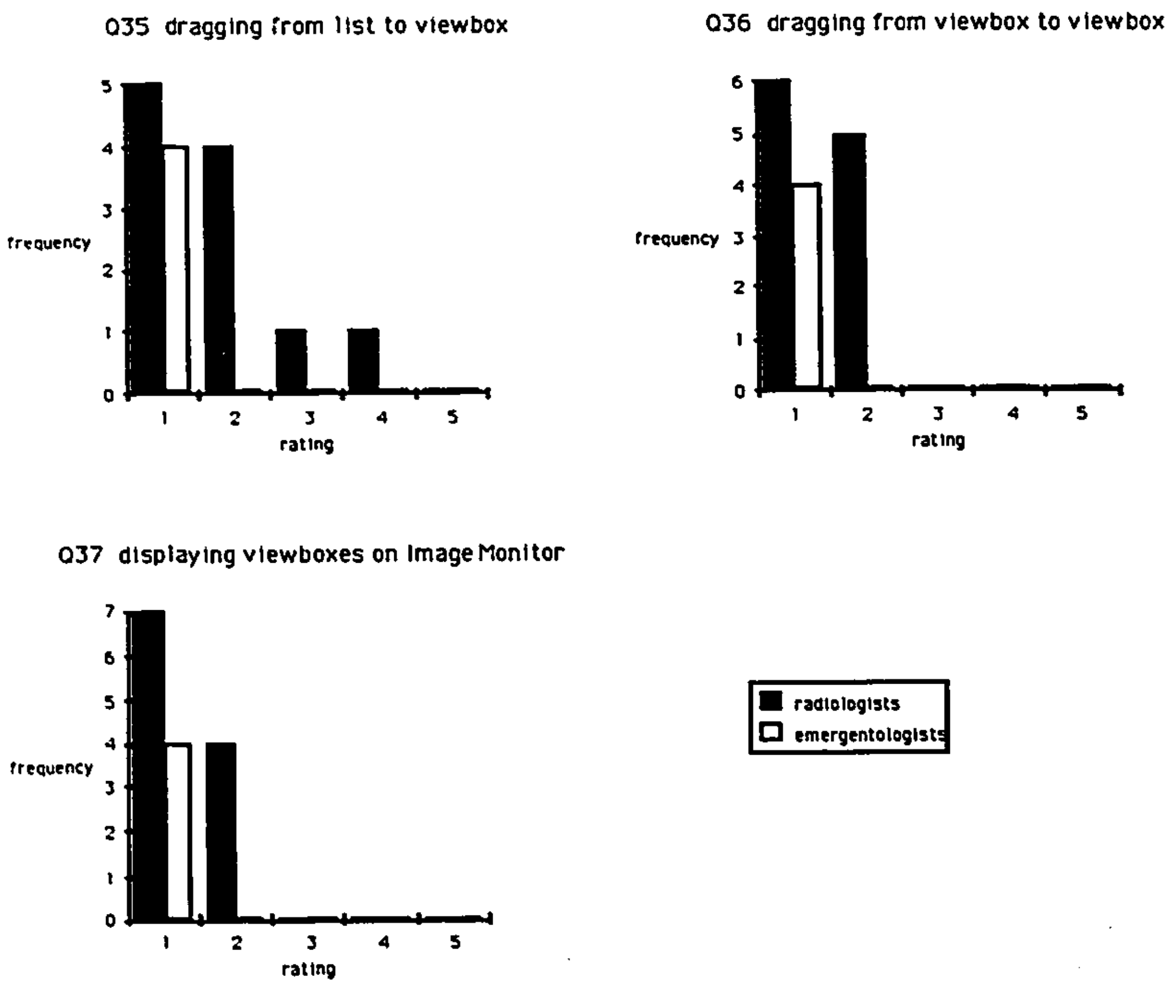


Figure 7.11: Initial Reaction to the Viewboxes.

Subject Sample: 11 radiologists, 4 emergentologists.
Reference Scale: from "very easy" (1) to "very difficult" (5).

the cursor. Ratings were on a scale of 1 to 5, where 1 meant "very easy" and 5, "very difficult."

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
40. Moving mouse from screen to screen	1.8	2.0	1.9
41. Locating cursor after screen change	2.9	2.8	2.9
42. Clicking to select an image	1.7	1.3	1.6
43. Dragging an image to viewbox	1.7	1.0	1.5
44. Displaying viewboxes on Image Monitor	1.1	1.0	1.1
45. Displaying an image	1.3	1.0	1.2
46. Knowing image selected	1.2	1.3	1.2
47. Locating cursor: control screen	1.4	1.3	1.3
48. Locating cursor: image screen	2.5	2.3	2.4
49. Moving mouse: control screen	1.2	1.3	1.2
50. Moving mouse: image screen	1.6	1.3	1.5

Table 7.5: Mouse, Monitor and Cursor Issues (distribution averages).

Moving the mouse from one monitor to the other ($\mu^t_{40} = 1.9$) was rated as fairly easy, but visually locating the cursor after changing monitor ($\mu^t_{41} = 2.9$) was found to be more difficult. Clicking on an image label to display the image ($\mu^t_{42} = 1.6$), and dragging an image label to a viewbox ($\mu^t_{43} = 1.5$) were thought to be easy. Popping-up the set of viewboxes on the Image Monitor ($\mu^t_{44} = 1.1$), displaying an image (from either screen) ($\mu^t_{45} = 1.2$), and knowing which image is selected by a click ($\mu^t_{46} = 1.2$) were also considered very easy. Visually locating the cursor on the Image Monitor ($\mu^t_{48} = 2.4$) was thought to be definitely more difficult than locating it on the Control Monitor ($\mu^t_{47} = 1.3$). Moving the mouse on either Monitor ($\mu^t_{49} = 1.2$ for the Control Monitor, and $\mu^t_{50} = 1.5$ for the Image Monitor) was found to be easy.

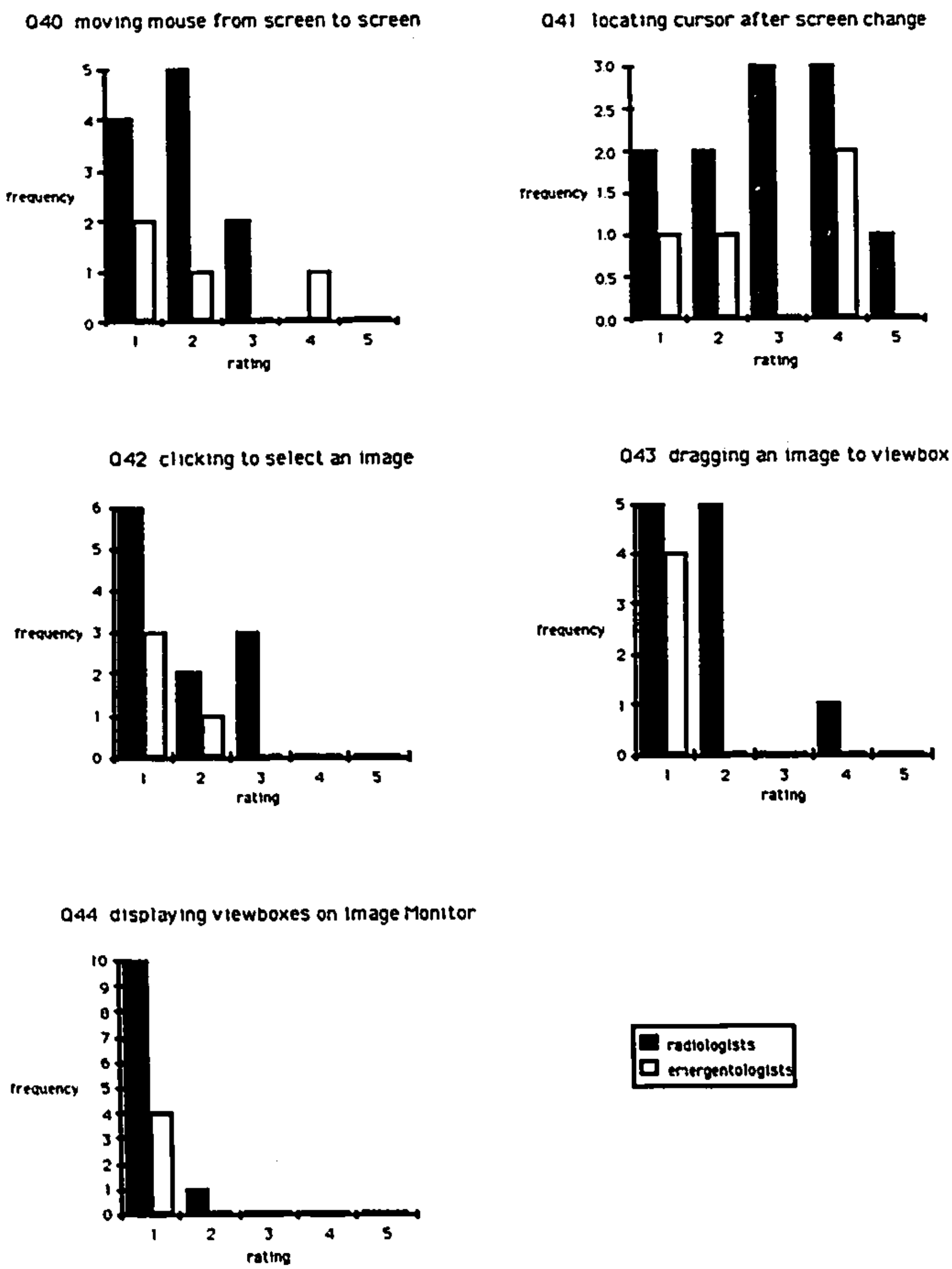


Figure 7.12: Mouse, Monitor and Cursor Issues.

Subject Sample: 11 radiologists, 4 emergentologists.
Reference Scale: from "very easy" (1) to "very difficult" (5).

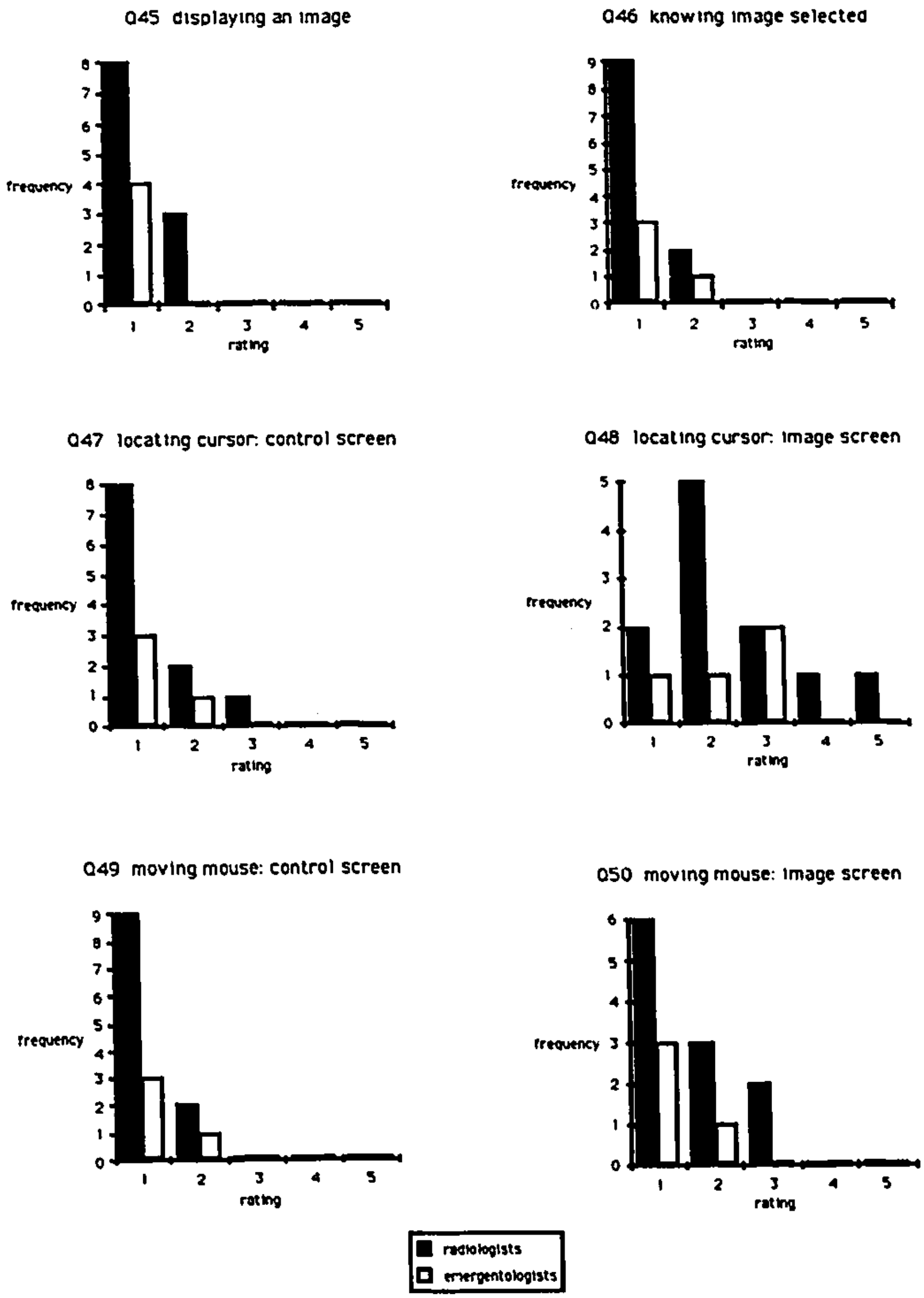


Figure 7.13: Mouse, Monitor and Cursor Issues.

Subject Sample: 11 radiologists, 4 emergentologists.
Rating Scale: from "very easy" (1) to "very difficult" (5).

Control Monitor

Questions 51 through 54 (Table 7.6 and Fig. 7.14) evaluated the Control Monitor. Ratings were on a scale of 1 to 5, where 1 meant "not at all a problem" and 5, "very much a problem."

Question	Mean Ratings		
	Radiologists n=11	Emergentologists n=4	Both n=15
51. Reading control screen	1.2	1.0	1.1
52. Character size	1.1	1.0	1.1
53. Screen clutter	1.2	1.3	1.2
54. Information adequacy	1.2	1.0	1.1

Table 7.6: Mouse, Monitor and Cursor Issues (distribution averages).

All following factors were judged not to be a problem: the ability to read the information presented on the screen ($\mu^t_{51} = 1.1$), the size of the characters ($\mu^t_{52} = 1.1$), the clutteredness or "business" of the information on the screen ($\mu^t_{53} = 1.2$), and the overall adequacy of the information provided ($\mu^t_{54} = 1.1$).

7.6 Summary

In this chapter we described the field tests of the image organization and navigation aspects of the user interface, and presented the results thereof.

The main objectives of these tests were to determine the appropriateness of the anthropomorphic image organization strategy for organizing the images of patient files, and to determine the relevance of the soft-viewbox based image navigation strategy.

In terms of functional mechanics, both image organization strategies were rated easy to use. Exceptions were single-item scrolling, which was used rather than continuous scrolling for the linear strategy; and two-step selection, which was required to actually

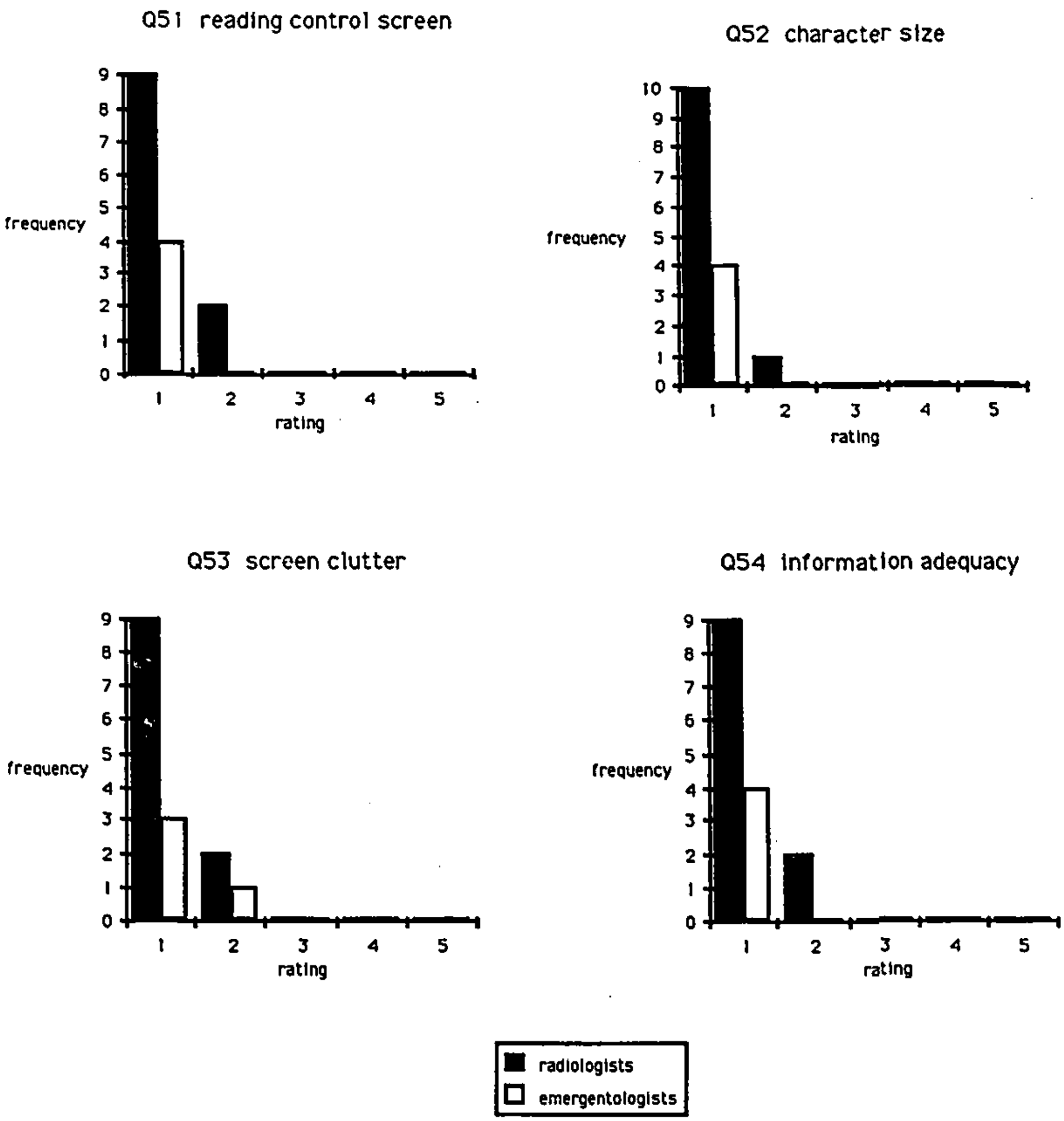


Figure 7.14: Mouse, Monitor and Cursor Issues.

Subject Sample: 11 radiologists, 4 emergentologists.
Rating Scale: from "not at all a problem" (1) to "very much a problem" (5).

select an image for viewing. The two-step selection was found to be a limiting factor with the anthropomorphic strategy. No preference for one strategy versus the other emerged from the emergentologists (although the subject sample size was only 4), but radiologists expressed a preference for the anthropomorphic strategy. This may be explained by the more complex nature of the radiologists' tasks (Section 5.2) which may require a more sophisticated strategy for presenting the image content of a patient file. For instance, radiologists reported that they often handle more than 10 films for a given body area, whereas emergentologists reported that they almost never do (Fig. 7.15).

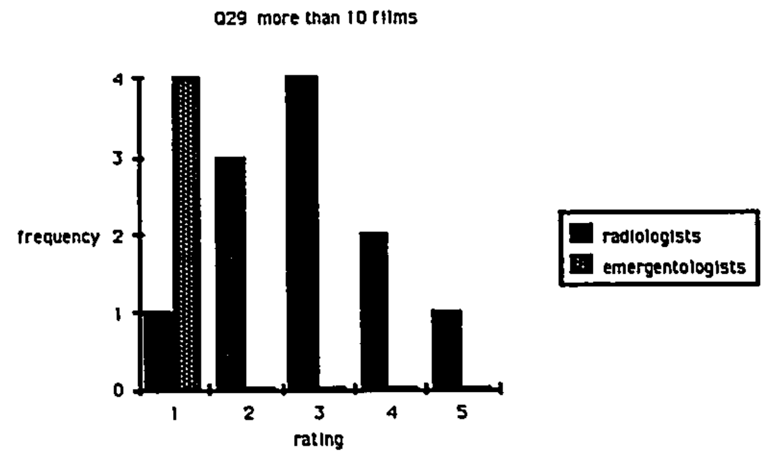


Figure 7.15: Rated Number of Images used per Body Area.

Subject Sample: 11 radiologists, 4 emergentologists.
Rating Scale: from "almost never" (1) to "always" (5).

In terms of mechanics, the soft viewboxes were thought to present no problems and were considered easy to use. There is suggestive evidence that soft viewboxes would be a relevant means for navigating through sets of images but their particular applied benefits remain to be experimentally assessed.

Chapter 8

Conclusions and Suggestions for Further Research

In this thesis, the image organization and image navigation issues of the user interface of a PACS workstation were investigated.

From our review of PACS workstation requirements, preference studies and existing designs, and from our own user and task analysis', we concluded that the image content of a patient file should be presented in an organized structure. Also, choice of a specific structure should be flexible depending on who is using the system (i.e. what category of users). We also noted that the limited physical image viewing surface (the actual number of monitors available for viewing images) could be better utilized by the radiologist, provided a more efficient image navigation strategy (i.e. the means for selecting an image for viewing) was available.

Our proposed digital image organization strategy is based on an "anthropomorphic" organization (Fig. 5.3). The center of the Control Monitor is occupied by the sketch of a human body around which labels of images referring to the same body area are grouped. This graphical index of images provides an overview of the image content of the patient folder. The list of image labels related to a body area is revealed by selecting the body area label. Our image navigation strategy is based on a soft representation of the physical viewboxes (Fig. 5.5). It mimics the spatial arrangement in the analog world (onto physical viewboxes). The soft viewbox surface constitutes a bi-dimensional space on which the user arranges radiographs represented by icons. Although our PACS workstation has only one monitor for displaying images, the resulting visual workspace is augmented by this strategy.

In order to evaluate our image organization and navigation strategies, we implemented these in Smalltalk/V, an object-oriented programming language, and conducted a series of tests at the Ottawa Civic Hospital. In addition to providing answers regarding the relevance of our image organization and navigation strategies, these tests deepened our understanding of the radiologists and the emergentologists, their differences, and their working practices.

Radiologists did prefer the anthropomorphic organization to a simple list-type organization used for comparison. The soft viewboxes were found to be of "theoretical" interest for image navigation but their certain usefulness remains to be experimentally assessed. Nevertheless, the radiologists' comments and suggestions let us believe in the usefulness of our viewbox-based image navigation strategy. Based on these comments and suggestions, we derived the following list of topics which should be investigated further.

1. A second experiment should be performed under a more realistic environment (i.e. generation of diagnosis' for real patient cases). Our experiment was rather artificial since subjects were closely guided by the experimenter.

2. Automatic loading of the viewboxes with all views of an examination. The order of the views could be determined by the nature of the examination and by user preferences.

3. Automatic sequencing of the display of the views placed on the viewboxes (e.g. Provision of a pair of "NEXT/PREVIOUS" buttons that would display images in sequence from left to right and top to bottom).

4. Provision of multiple planes of soft viewboxes (create a 3-D space) for cross-examination comparison of corresponding views.

5. Provision of a larger 2-D viewbox space (than the traditional 4x2 grid) for examinations with large number of views.

6. Provision of a facility enabling the display of more than one film at a time. At present, one film corresponds to one image. This may not always be optimum for display of CT-scan (eight to twelve views per film) or of some tomogram examinations (more than thirty small films).

Appendix A

Mouse Script

MOUSE SCRIPT

Introduction

This isn't the computer system that you will be using for X-rays. It is a Macintosh which we are going to use to the teach mousing skills you will need when you do use that system. (show the mouse)

Are you left or right handed? (adjust the mouse to suit them)

HAVE YOU EVER USED A MOUSE BEFORE?

Yes: Okay then we can run through this quickly
No: That's fine what we'll do is show you some skills using the mouse.

There are some differences between what occurs in this demonstration and what happens on the system that will be in the hospital . For example as you move the mouse you see a hand on the screen move, with the x-ray system you will see an arrow. Overall however,the basic mousing skills used here are needed for your system too.

We will have you start by getting used to how the mouse moves.Don't worry about the button at all.

(check mousing position-button should be at top)

Take hold of the mouse and move it around (left/ right, up/down). Try to make the pointer on the screen move to all corners.

See what happens when you lift the mouse off the table: the hand doesn't move. This allows you to reposition the mouse so that it's in a more convenient place for you. Try that a few times. Just move the mouse, then lift and replace it at a better location for you.

Moving Skill

Now look at the star pattern. We want you to get some practice moving the mouse to specific locations.

Move the mouse so that the finger points to each number in sequence.
Are you comfortable using the mouse?
Would you like to try it again?(yes: restart no: go on)

Note: if you feel they need more practice-suggest they try it again

When ready to go on: move the mouse so that the finger is on the word **NEXT**. Press the mouse button once (just as you would a typewriter key), this is called clicking and it will take you to the next practice skill.

Clicking Skill

Now we want you to practice "clicking". Let's do the star pattern again.

This time move the mouse so that the pointer (tip of the finger) is over the number. Make a quick click-and-release movement with the mouse button. Notice that the finger tip of the pointer must be inside the circle.

Try clicking the next circle with the fist in the circle but the finger tip out and you will see how this works.

(as before follow the numbers in sequence)

Move pointer to the **NEXT** button and "click" it.

<u>Dragging Skill</u>

Another skill is called dragging. It involves pressing the mouse button, but this time you do not release it until you have moved the pointer to a new position.

Place the pointer over the button named "1 beep". While holding the mouse button depressed move the pointer to the button's corresponding box. (ie. "1 beep " button to "1 beep " box). As you do so , an outline of the button follows the pointer while the shadow of the button remains in its original place. When you release the mouse button ,the "1 beep" button moves to its new location and performs its function.

Go ahead and move all of the buttons to their new boxes.

Click the **RESET** button to try again or the **NEXT** button to move on.

<u>Scroll Skill</u>

Another skill that you will sometimes be using is called scrolling.

Scroll boxes are frequently used for lists which are too long to fit into the provided viewing area. When you see a scroll bar there is probably information in your lists that you cannot see.

Scroll bars have arrows that are used to move through the list. We'll be using the big arrows which allow you to move more smoothly through the list. The two at the top do the same thing (point to these). These arrows allow you to see items further up on the list (ie. words that begin with letters closer to the beginning of the alphabet)

The two arrows on the bottom allow you to see items further down in the list (ie. words that begin with letters closer to the end of the alphabet).

Try moving back and forth through the list by placing the pointer on the big arrows.

notice: You will know you are at the top of the list because pressing the up arrow will no longer move it. You will know you are at the bottom of the list because pressing the down arrow will no longer move it.

Now by moving through the scroll find the answers to the questions that are written on the screen.

Click the **NEXT** button to move on.

<u>Overview</u>

You have now learned the following skills:

1.moving
2.clicking
3.dragging
4.scrolling

If you want further practice in a skill just "click on it". If not ,you're finished.

Do you have any questions?

Click the **NEXT** button.

Appendix B

Radiographic Images for Experiment

patient last name	MIKE
patient first name	Mike Kilo
date of birth	19??.??.??
identification number	1234500

	BODY AREA	VIEW	EXAM. #	DATE
1	cervical spine	oblique	63780	1988.06.15
2	cervical spine	oblique	63780	1988.06.15
3	cervical spine	lateral	63780	1988.06.15
4	cervical spine	odontoid	67780	1988.06.15
5	cervical spine	AP	67780	1988.06.15
6	cervical spine (myelo)	oblique	64220	1988.06.15
7	cervical spine (myelo)	lateral	64220	1988.06.15
8	cervical spine (myelo)	AP	64220	1988.06.15
9	cervical spine (myelo)	AP	64220	1988.06.15
10	chest	PA	63780	1988.06.15
11	chest	lateral	63780	1988.06.15
12	lumbar spine	cone down	64220	1988.06.15
13	lumbar spine	lateral	64220	1988.06.15
14	lumbar spine	AP	64220	1988.06.15
15	lumbar spine (myelo)	32	64220	1988.06.15
16	lumbar spine (myelo)	lateral	64220	1988.06.15
17	lumbar spine (myelo)	35	64220	1988.06.15
18	lumbar spine (myelo)	43	64220	1988.06.15
19	lumbar spine (myelo)	F	64220	1988.06.15
20	lumbar spine (myelo)	49	64220	1988.06.15

Table B.1: Radiographic Films Used in Experiment.

Appendix C

Script for Experiment

User Testing Scripts

Pre-experiment:

Users acquire profiency at using a mouse on a MAC. This involves basic operations such as:

- moving the mouse
- selecting objects with the mouse button (clicking)
- draging objects
- scrolling lists using up-down arow icons

Experiment #1

Objectives:

- User acquires profiency at using the system
- Compare learnability of both organization strategies

Methodology:

- A typical session is first demonstrated from beginning to end before user can try it.
- All users are trainned on both organization strategies.
- Half the users are first trainned on the linear strategy, the other half on anthropomorphic strategy first.
- Start with a large set of images so that scrolling can be demonstrated.
- Some steps in the script may appear to be repetitive but read the objectives again!

Experiment #1A

Submit subject to linear organization first and then anthropomorphic organization.

Script (Demonstration of a typical session - linear organization):

First I will go through a typical session with the system. I will explain to you what the system can do and how to use the various features. Please bear in mind that this is not the complete system, the purpose of these current trials is to investigate the image navigation tasks, that is how you will be able to find, select and examine images from a patient.

start on "select a patient" screen

The first thing to notice is that the cursor is an arrow rather than the hand you saw on the other computer.

The purpose of this screen is to show you a list of the patients for which there are images in the system. It is also from this screen that you select the patient you are interested in.

Let's say we are interested in patient X's case. So I will move the arrow towards the name of the patient X on the patient list. You notice that as the arrow moves over a name, that name gets displayed over a gray background. This is so that you know that if you were to click now, the name that is displayed over a gray background is the one that would get selected.

So, I will move the arrow over patient X's name and click on it. You notice that the selected name is now displayed in reverse, that is with white characters over a black background. This is so that you know which name of the list has indeed been selected.

Now, in order to see the images related to patient X, I will click on the "examine patient" button at the bottom of the screen.

linear organization screen comes up

The purpose of this screen is to expose to you the image content of the file of the patient previously selected. First thing to notice is that the last name, first name and unique I.D. number of the patient are displayed on a line at the top of the screen so that you don't forget whom you are examining. In the lower portion of the screen you can see a list of all the images for that patient. Each image is identified by the body area it comes from (eg. CHEST), also by the view (eg. AP), by the time it was taken (eg. 13:40) and by the examination number (eg. 12345). Notice that the images are first sorted alphabetically by body area, then by view, then by time and last by examination number.

The arrows on each side of the list are like the ones you played with on the other computer. They allow you to access parts of the list hidden to you.

Let us say we are interested in viewing x-ray A. To do so we simply click on the name of x-ray A (no scrolling required). You see again that as the arrow moves over the name of an image, it gets displayed over a gray background. Also, as I click on the name, it gets displayed in reverse (white characters over black background). Look at the Image Screen, this is where the image I selected gets displayed.

Let's say now that we are interested in viewing x-ray B. Notice that x-ray B does not appear on the portion of list that we can see. Because this is a long list and not all items are displayed, I will have to use the scrolling facility. Because x-ray B is further down the list I will have to use the bottom arrow to reach it. I then select it just as I did for x-ray A.

explain usage of viewboxes

In the upper portion of the screen you notice eight boxes. They represent the light boxes or viewboxes you usually work with. You can use them almost like you use the other ones you are familiar with to set up the films you want to see on the viewbox. The viewbox gives easy access to the images your interested in. You can place x-rays of special interest to you there. The first thing I will do is take a few images from the list and put them up on the viewboxes so that I have picked out the images I will want to examine. The way you "put up an image" is by dragging, just like you have done before on the other computer. So, I will put up x-ray C (no scrolling required): I move the arrow over the name of x-ray C, press and hold down the mouse button, drag the outline over a window, and release the button once the four-arrow cursor is in the box. Now I'll put up x-ray D (scroll up required) and x-ray E (scroll down required).

Now let's say x-ray D should have been placed between C and E. I just have to drag x-ray D in the other viewbox using the same dragging procedure I used earlier. We can also view an image from the viewboxes. To view x-ray D for example, I just have to click on it.

I will now move the arrow across to the Image Screen. Normally there would be image enhancement functions available on this screen but they have been removed for the time being. We will use the arrow just to point at the areas of interest on the image. You see that all that has to be done is to move the arrow towards the Image Screen and the arrow automatically moves to it. You see that the arrow is blue on that screen. To move back to the Control Screen I simply move the arrow back toward that screen. Let's move back to the Images Screen.

Now suppose we are interested in looking at either x-ray C or E. All I have to do is click the button wherever the arrow is located on the Image Screen, and as you can see, a replica of the set of viewboxes pops-up on the screen. These viewboxes behave just like the other ones. I can drag x-ray E to an other viewboxe, and select x-ray C for example. The selection of an image causes the viewboxes to disappear so that you can see the image being displayed underneath.

This concludes my demonstration. Do you have any questions? What we'll do now is repeat the whole thing but this time you'll be manipulating the mouse.

system is restarted

Script (User in driver's seat - linear organization):

start on " select a patient "screen

First show me the list of images for patient X.

linear organization screen comes up

Can we view x-ray A (no scrolling required)?

Now can we view x-ray B (scrolling required)?

Can you put x-ray C on this viewbox (no scrolling required)?

Now please put x-ray D on that viewbox (scroll up required),

And x-ray E on that other viewbox (scroll down required).

Could you move x-ray D on the viewbox above?

Could we view x-ray D?

Now you should try to move the arrow across the two screens a couple of times to get a feel for it. First move slowly to the Image Screen and wait. Now back to the Control Screen. Please do it again back and forth. Finally move the arrow to the Image Screen.

Now try to move the arrow on the x-ray image. Now pop-up the set of viewboxes by clicking with the button. Can you move x-ray E one viewbox to the left? Can you show me x-ray C? Can you show me x-ray E now?

Do you have any questions about what you have practice?

questions session

system is restarted

Script (Demonstration of a typical session - anthropomorphic organization):

Again I will go through a typical session with the system. You will notice that images are presented differently than on the previous system.

start on" select a patient" screen

Again let us say that we are interested in patient X's case so I select it and click on the "examine patient" button.

anthropomorphic organization screen comes up

The purpose of this screen is again to expose to you the image content of the file of the patient previously selected. The line at the top of the screen is used again to identify the patient. Just under you can see again the set of viewboxes.

In the lower portion of the screen you can see a sketch of the human body. Images are grouped by body area and each group points to the corresponding area on the sketch. The "thickness" of the button of an area gives indication on the number of images related to that area. In order to see the list of images related to a body area I simply click on the name of the body area. You see that the list pops-up just under the name selected. The images of the list are first sorted

alphabetically by name of view (eg. AP), then by the time is it was taken (eg. 13:40), and finally by examination number (eg. 12345). Now I can view x-ray A by clicking on it.
Notice that the list disappears after an image has been selected from it. I can put-up an image on a viewbox by first selecting an area and then dragging it on a viewbox (done twice). Again I can also move an image from a viewbox. The arrow can also be moved on the Image Screen where I can pop-up the viewboxes, move an image from a viewbox to an other, and select an other image.
So you see that the only thing different from the previous system is the way the images are organized. Any questions?
What we will do now is repeat the whole thing but this time you will be manipulating the mouse.

system is restarted

Script (User in driver's seat - anthropomorphic organization):

start on " select a patient "screen

First show me the images for patient X

anthropomorphic organization screen comes up

Can we view x-ray A?
Now can we view x-ray B (from a different body area than x-ray A was from)?
Can you put x-ray C on this viewbox?
Now please put x-ray D on that viewbox (from the same body area than x-ray C was from),
And x-ray E on that other viewbox (from a different body area than x-ray C was from).
Could you move x-ray D on the viewbox above?
Could we view x-ray D?
Now you should try to move the arrow across the two screens a couple of times to get a feel for it. First move slowly to the Image Screen and wait. Now back to the Control Screen.
Please do it again back and forth. Finally move the arrow to the Image Screen.
Now try to move the arrow on the x-ray image. Now pop-up the set of viewboxes by clicking with the button. Can you move x-ray E one viewbox to the left? Can you show me x-ray C?
Can you show me x-ray E now?
Do you have any questions about what you have practice?

questions session

system is restarted

Experiment #1B

Submit subject to anthropomorphic organization first and then linear organization.

Script (Demonstration of a typical session - anthropomorphic organization):

First I will go through a typical session with the system. I will explain to you what the system can do and how to use the various features. Please bear in mind that this is not the complete system, the purpose of these current trials is to investigate the image navigation tasks, that is how you will be able to find, select and examine images from a patient.

start on" select a patient" screen

The first thing to notice is that the cursor is an arrow rather than the hand you saw on the other computer.

The purpose of this screen is to show you a list of the patients for which there are images in the system. It is also from this screen that you select the patient you are interested in.

Let's say we are interested in patient X's case. So I will move the arrow towards the name of the patient X on the patient list. You notice that as the arrow moves over a name, that name gets displayed over a gray background. This is so that you know that if you were to click now, the name that is displayed over a gray background is the one that would get selected.

So, I will move the arrow over patient X's name and click on it. You notice that the selected name is now displayed in reverse, that is with white characters over a black background. This is so that you know which name of the list has indeed been selected.

Now, in order to see the images related to patient X, I will click on the "examine patient" button at the bottom of the screen.

anthropomorphic organization screen comes up

The purpose of this screen is to expose to you the image content of the file of the patient previously selected. First thing to notice is that the last name, first name and unique I.D. number of the patient are displayed on a line at the top of the screen so that you don't forget whom you are examining.

In the lower portion of the screen you can see a sketch of the human body. Images are grouped by body area and each group points to the corresponding area on the sketch. The "thickness" of the button of an area gives indication on the number of images related to that area. In order to see the list of images related to a body area I simply click on the name of the body area. You see that the list pops-up just under the name selected. The images of the list are first sorted alphabetically by name of view (eg. AP), then by the time is it was taken (eg. 13:40), and finally by examination number (eg. 12345).

In the upper portion of the screen you notice eight boxes. They represent the light boxes or viewboxes you usually work with. I will show you later how to use them.

Let us say we are interested in viewing x-ray A. To do so we simply click on the name of the body area of x-ray A. You see again that as the arrow moves over the name of a body area, it gets displayed over a gray background. Also, as I click on the name, it gets displayed in reverse (white characters over black background). Then I click on the name of the image. Look at the Image Screen, this is where the image I selected gets displayed.

Let's say now that we are interested in viewing x-ray B. As I did for x-ray A I first select the name of the body area. Then I select the name of the image.

explain usage of viewboxes

We will now make use of the viewboxes. You can use them almost like you use the other ones you are familiar with. The first thing I will do is take a few images from the list and put them up on the viewboxes. The way you "put up an image" is by dragging, just like you have done before on the other computer. So, I will put up x-ray C: I select the body area then I move the arrow over the name of x-ray C, press and hold down the mouse button, drag the outline over a window, and release the button. Now I'll put up x-ray D (from a different body area) and x-ray E (again, from a different body area).

Now let's say x-ray D should have been placed between C and E. You just have to drag x-ray D in the other viewbox. We can also view an image from the viewboxes. To view x-ray D for example, I just have to click on it.

I will now move the arrow across to the Image Screen. Normally there would be image enhancement functions available but they have been removed for the time being. We will use the arrow just to point at the areas of interest on the image. You see that all that have to be done is to move the arrow towards the Image Screen and the arrow automatically moves to it. You see that the arrow is blue on that screen. To move back to the Control Screen I simply move the arrow back toward that screen. Let's move back to the Images Screen.

Now suppose we are interested in looking at either x-ray C or E. All I have to do is click the button wherever the arrow is located on the Image Screen, and as you can see, a replica of the set of viewboxes pops-up on the screen. These viewboxes behave just like the other ones. I can drag x-ray E to an other viewboxe, and select x-ray C for example. The selection of an image causes the viewboxes to diappear so that you can see the image being displayed underneath.

This concludes my demonstration. Do you have any questions? What we'll do now is repeat the whole thing but this time you'll be manipulating the mouse.

system is restarted

Script (User in driver's seat - anthropomorphic organization):

start on " select a patient "screen

First show me the images for patient X

anthropomorphic organization screen comes up

Can we view x-ray A?

Now can we view x-ray B (from a different body area than x-ray A was from)?

Can you put x-ray C on this viewbox?

Now please put x-ray D on that viewbox (from the same body area than x-ray C was from), And x-ray E on that other viewbox (from a different body area than x-ray C was from).

Could you move x-ray D on the viewbox above?

Could we view x-ray D?

Now you should try to move the arrow across the two screens a couple of times to get a feel for it. First move slowly to the Image Screen and wait. Now back to the Control Screen. Please do it again back and forth. Finally move the arrow to the Image Screen.

Now try to move the arrow on the x-ray image. Now pop-up the set of viewboxes by clicking with the button. Can you move x-ray E one viewbox to the left? Can you show me x-ray C? Can you show me x-ray E now?

Do you have any questions about what you have practice?

questions session

system is restarted

Script (Demonstration of a typical session - linear organization):

Again I will go through a typical session with the system. You will notice that images are presented differently than on the previous system.

start on "select a patient" screen

Again let us say that we are interested in patient X's case so I select it and click on the "examine patient" button.

linear organization screen comes up

The purpose of this screen is again to expose to you the image content of the file of the patient previously selected. The line at the top of the screen is used again to identify the patient. Just under you can see again the set of viewboxes.

In the lower portion of the screen you can see a list of all the images for that patient. Each image is identified by the body area it comes from (eg. CHEST), also by the view (eg. AP), by the time it was taken (eg. 13:40) and by the examination number (eg. 12345). Notice that the images are first sorted alphabetically by body area, then by view, then by time and last by examination number.

The arrows on each side of the list are like the ones you played with on the other computer. They allow you to access parts of the list hidden to you.

Now I can view x-ray A by clicking on it.

I can put-up an image on a viewbox by dragging it on a viewbox (done twice and requiring scrolling each time). Again I can also move an image from a viewbox. The arrow can also be moved on the Image Screen where I can pop-up the viewboxes, move an image from a viewbox to an other, and select an other image.

So you see that the only thing different from the previous system is the way the images are organized. Any questions?

What we will do now is repeat the whole thing but this time you will be manipulating the mouse.

system is restarted

Script (User in driver's seat - linear organization):

start on " select a patient "screen

First show me the images for patient X.

linear organization screen comes up

Can we view x-ray A (no scrolling required)?

Now can we view x-ray B (scrolling required)?

Can you put x-ray C on this viewbox (no scrolling required)?

Now please put x-ray D on that viewbox (scroll up required),

And x-ray E on that other viewbox (scroll down required).

Could you move x-ray D on the viewbox above?

Could we view x-ray D?

Now you should try to move the arrow across the two screens a couple of times to get a feel for it. First move slowly to the Image Screen and wait. Now back to the Control Screen. Please do it again back and forth. Finally move the arrow to the Image Screen.

Now try to move the arrow on the x-ray image. Now pop-up the set of viewboxes by clicking with the button. Can you move x-ray E one viewbox to the left? Can you show me x-ray C? Can you show me x-ray E now?

Do you have any questions about what you have practice?

questions session

system is restarted

Appendix D

Questionnaire for Experiment

Subject: ____________________ **Date:** __________

Specialization: ________________ **Time:** __________

Started with: Linear/Anthropomorphic

Duration of Experiment: ____________

****

Experiment 1: Long list/Linear

I A.Scrolling: As you have seen this list was too long to be shown altogether.

1. Indicate the extent to which this presented the following problems:

		Not at all a Problem				Very much a Problem
Q1	a. Forgetting that the list had more items	1	2	3	4	5
Q2	b. Moving the list the wrong way	1	2	3	4	5
Q3	c. Trouble finding the desired image	1	2	3	4	5
Q4	d. Not being able to get an overview of the list of all the images at once	1	2	3	4	5
Q5	e. Only being able to move the list one item at a time	1	2	3	4	5

Q6 3. Use of the scrolling list was:

Very easy 1 2 3 4 5 Very difficult

II B. Organization. With each film there is additional information given such as the time film taken, view etc. We are interested in your impressions regarding this further information.

Q7 1. Is there any missing information about the images which should be displayed in the list?
a. [] Yes Identify ______________________________.
b. [] No.

Q8 2. Is any of the information about the images now shown superfluous?
a. [] Yes Identify ______________________________.
b. [] No.

Q9 3. Is the order in which the information is presented appropriate?
a. [] Yes
b. [] No Changes ______________________________.

III Looking at the organization of images in the list of files:

Q10 1.The list you were shown organized the images alphabetically by body location. Within each body part the films are sorted alphabetically by view, and each view is sorted by the time when it was taken. Is there a different organization which you would prefer?
a. [] Yes Identify ______________________________.
b. [] No.

Experiment 1 Long list/Anthropomorphic

IV A. Anthropomorphic: This list was divided according to body parts.

1. Indicate the extent to which this presented the following problems:

			Not at all a Problem				Very much a Problem
Q11	a.	Selecting the wrong category (body area)	1	2	3	4	5
Q12	b.	Having to choose twice to select an image. (Body part, then image)	1	2	3	4	5
Q13	c.	Finding the desired image	1	2	3	4	5
Q14	d.	Obtaing an overview of images contained in file	1	2	3	4	5

Q15 3 Using the list divided by body parts was:

Very easy 1 2 3 4 5 Very difficult.

V B. Organization: With each film there is aditional information given such as the time the film is taken, the view, etc. We are interested in your impressions regarding this further information.

Q16 1. Is there any missing information about the images which should be displayed in the list?
a. [] Yes Identify ________________________________.
b. [] No.

Q17 2. Is any of the information about the images now shown superfluous?
a. [] Yes Identify ________________________________.
b. [] No.

Q18 3. Is the order in which the information is presented appropriate?
a. [] Yes b. [] No
Changes ____________________________.

VI Looking at the organization of the images in the file:

Q19 1. The organization of the images was classified by body part. Is this type of organization appropriate?
a. [] Yes
b. [] No. Why? ____.

Q20 2. Under each classification (i.e., part of the body) the films are listed alphabetically by view, then each view is sorted by the time that the film was taken. Is this organization appropriate?
a. [] Yes
b. [] No. Suggestions? ____.

Q21 3. Would it be useful to have the facility to select an area and have all the views appear in the viewbox area?
______yes ______no

Comment:

VII Experiment 1 Long List/ Linear versus Anthropomorphic

You have seen the files organized in two ways. We want to get your general impressions of which you prefer.

	Much prefer Linear				Much prefer Anthropomorphic
Q22 1. Ease of mouse use	1	2	3	4	5
Q23 2. Ease in locating images	1	2	3	4	5
Q24 3. Ease of obtaining an overview of the file content.	1	2	3	4	5
Q25 4. Ease to set up viewbox case	1	2	3	4	5
Q26 5. Other? ____________________	1	2	3	4	5

Comments:

Q27 2. Would you ever want the capability of displaying the information about the images contained in a patient's case in a different manner that the two organizations provided?

[] No

[] Yes Identify:____

Q28 3. Would you want to be able to set up your own organization even though this would make the system more complicated?

[] No

[] Yes Why?____

VIII Experiment 1. Other features

Q29 1. How often would you receive many films (more than 10) for 1 area?

Almost never 1 2 3 4 5 **Always**

2. Do you find the human figure helpful:

Q30 a. for quicker film location, [] Yes [] No

Q31 b. in giving an overview of the films available [] Yes [] No

Q32 3. When you move a selection to the viewbox would you rather have a small form of the image displayed in the box rather than the film information presently being displayed?

[] Yes [] No

Q33 4. Is there any reason to display both?

[] No

[] Yes Identify:____

Q34 5. Do you have any suggestions concerning the utilization of the viewboxes?

IX A. Viewboxes:
We would like your initial reaction to the viewbox.

Q35 1. Moving image information from lists to viewbox was:

Very easy 1 2 3 4 5 Very difficult

Q36 2. Moving images from one box to another on the viewbox was:

Very easy 1 2 3 4 5 Very difficult

Q37 3. Displaying the viewbox on the image screen was:

Very easy 1 2 3 4 5 Very difficult .

4. Was the information about an image on the viewbox satisfactory:

Q38 In content?

a. [] Yes
b. [] No Changes ______________________

Q39 In organization?

a. [] Yes
b. [] No Changes.______________________________.

Comments:

X Mouse/Screen/Cursor:
We would like your impressions of how easy you found the following items:

	Very easy				Very difficult
Q40 1. Moving the mouse from one screen to another.	1	2	3	4	5
Q41 2. Visually locating the cursor after changing screens	1	2	3	4	5
Q42 3. Clicking an image to select it.	1	2	3	4	5
Q43 4. Dragging an image to the viewbox.	1	2	3	4	5
Q44 5. Displaying viewbox on image screen.	1	2	3	4	5
Q45 6. Displaying an image	1	2	3	4	5
Q46 7. Knowing what image is selected by a click.	1	2	3	4	5
Q47 8. Visually locating cursor on the control screen.	1	2	3	4	5
Q48 9. Visually locating cursor on the image screen.	1	2	3	4	5
Q49 10. Moving mouse on control screen.	1	2	3	4	5
Q50 11.Moving mouse on image screen	1	2	3	4	5

XI We would like your initial reactions to some of these aspects of the system:

1. Concerning the control screen, indicate the extent to which the following presented a problem:

		Not at all a Problem				Very much a Problem
Q51	a. Reading the control screen.	1	2	3	4	5
Q52	b. Character size.	1	2	3	4	5
Q53	c. Screen being too cluttered/busy.	1	2	3	4	5
Q54	d. Information provided is not adequate.	1	2	3	4	5

XII We would like your impressions of the time required for the system to respond to your actions:

	Very Fast				Very slow
Q55 1. Time from the selection of the patient's name to diplay the patient's file	1	2	3	4	5
Q56 2. Time from film selection to image display	1	2	3	4	5
Q57 3. Time from request to display of pop-up menu on the image screen	1	2	3	4	5
Q58 4. Time to change location of cursor from control screen to image screen	1	2	3	4	5
Q59 5. In general	1	2	3	4	5

XIII We are interested in your impressions of the timing of mouse functions:

1. As you move your mouse, the cursor on the screen follows your movements. Do you find the tracking speed to be:

Q60 <u>On the control screen:</u>

Too fast 1 2 3 4 5 **Too slow.**

Q61 <u>On the image screen:</u>

Too fast 1 2 3 4 5 **Too slow.**

Q62 2. As you drag a film to the viewbox, the outline follows your movement. Do you find the tracking speed to be:

<u>On the control screen:</u>

Too fast 1 2 3 4 5 **Too slow.**

Appendix E

Workstation Characteristics and Performance

Computer Platform	Compaq Deskpro/20
Processor	80386
Clock Speed	20 MHz
Control Monitor Height	350 pixels (18 cm)
Control Monitor Width	640 pixels (24 cm)
Control Monitor Character Height	14 pixels (0,72 cm)
Control Monitor Character Width	8 pixels (0,30 cm)
Image Monitor Height	1280 pixels (32,8 cm)
Image Monitor Width	1024 pixels (29,0 cm)
Image Monitor Character Height	14 pixels (0,40 cm)
Image Monitor Character Width	8 pixels (0,20 cm)
Time to Display 14'x17' Film on Image Monitor	3 sec.
Time for Cursor Transfer (Control to Image Monitor)	2 sec.
Time to Display Viewboxes on Image Monitor	5 sec.

Table E.1: Workstation Characteristics and Performance.