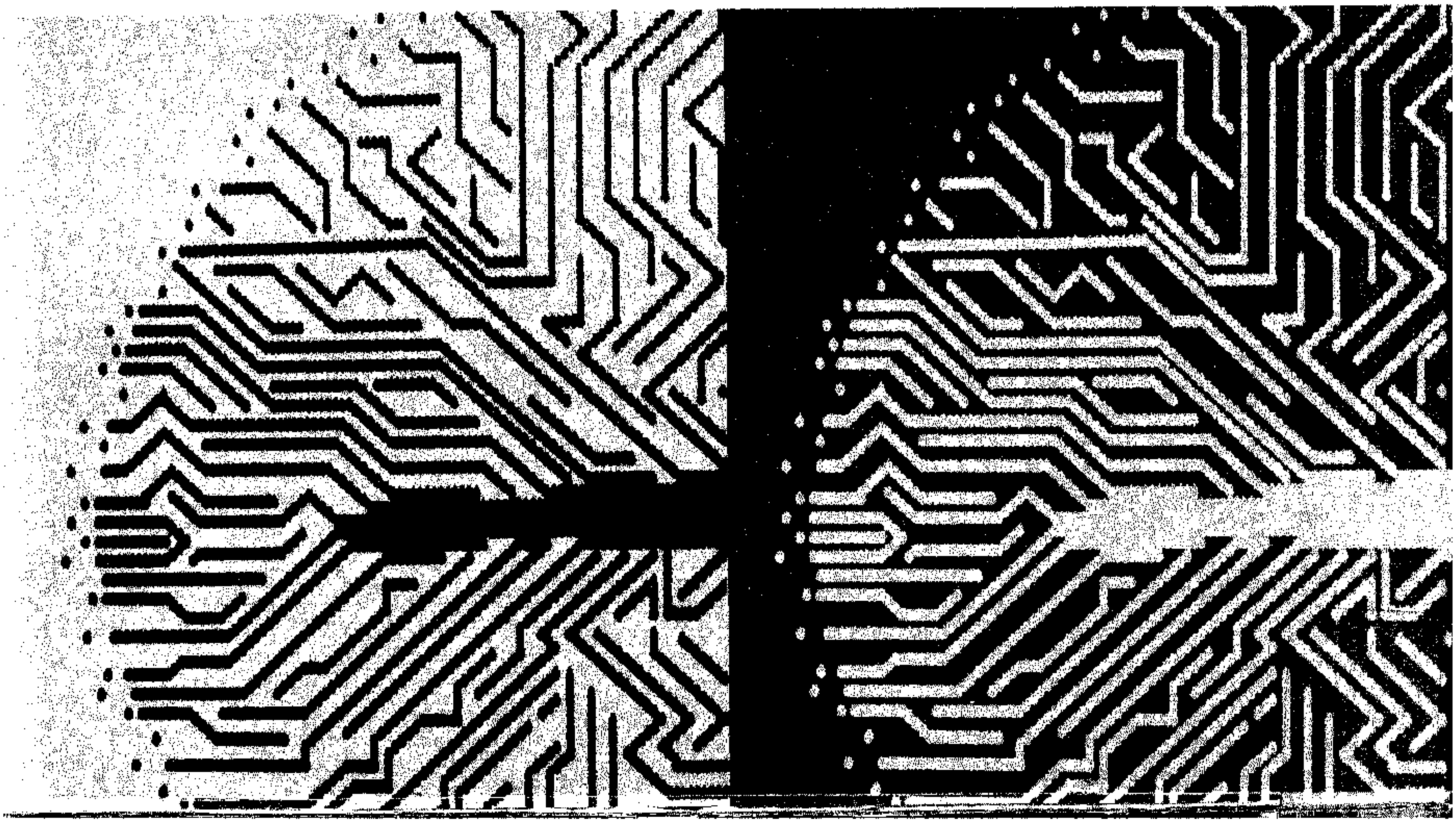
THE ARCHITECTURE OF VIDEOTEX SYSTEMS

JAN GECSEI



The Architecture of Videotex Systems

THE ARCHITECTURE OF VIDEOTEX SYSTEMS

Jan Gecsei

Université de Montréal

Prentice-Hall, Inc.

Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Gecsel, Ján (date)
The architecture of videotex systems.

Bibliography: p.
Includes index.
1. Videotex (Data transmission system) I. Title.
TK5105.G43 1983 384 83-3158
ISBN 0-13-044776-5

Editorial/production supervision
and interior design by *Kathryn Gollin Marshak*
Cover design by *Ben Santora*
Manufacturing buyer: *Gordon Osbourne*

Cover photograph courtesy of the Department of Communications,
Government of Canada

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-044776-5

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

To Alice, Korynna, and Dora,
who were patient and fed me

Contents

Preface

Videotex is one of the recent developments combining advances in computers, telecommunications technology, and consumer electronics. The ambition of videotex is no less than to upgrade today's mass communication media into computerized mass information utilities.

It might seem that videotex is suffering from an identity crisis; indeed, there are numerous closely related developments, such as electronic mail, on-line information services, and home computing, to name only a few. But in spite of the lack of a clean academic definition, a number of well-established systems (such as Prestel, Telidon, Bildschirmtext, and Télétel) are operational and under development in many countries, backed by technical expertise and sizeable investments from business interests. Underlying these systems, there is a considerable amount of accumulated knowledge, experience, and open research problems, most of which are highly technical and videotex-specific. However, so far there are few publications attempting to explain systematically these issues and to relate them to the traditional disciplines of computer networks, communications, and database technology. This may be caused, at least partly, by the enormous rate of change witnessed by videotex, and the ensuing danger of rapid obsolescence of any book on the subject. However, these very same reasons call for such an undertaking in order to clarify the basic concepts behind the maze of systems, approaches, standards, proposals, and counter-proposals flooding the desks of videotex managers, designers, engineers, and students.

The emphasis of this book is thus on the technical aspects of television-based videotex. The subject matter is divided into five parts. Part I provides an introduction to videotex and background material about similar systems and computer networks. Parts II, III, and IV are the core of the book, and they loosely follow the architectural layers of the ISO Open Systems Interconnection model (described in Chapter 3). Part II treats ISO layers 1 through 5, here collectively called the communications level. The subjects involved are the physical media used to transmit and deliver videotex information, the underlying protocols, and the communication structure of videotex networks. Part III is concerned with the presentation level (ISO layer 6), treating in detail the image-coding options in current use, their impact on terminal design, and the related problems of national and international standardization. Part IV—the application level—deals with databases for videotex and teletext, gateways, service computers, and service providers' equipment. The three chapters of Part V touch upon themes important for the future of videotex: alternative methods of interfacing with the user, telesoftware (seen as the key to distributed processing in videotex) and methods of performance evaluation.

Readers familiar with the basic notions involving computers, communications, and databases, or having some experience in videotex, should have no difficulty in following the text. (Chapter 15 is a possible exception.) Most concepts used in the text are defined or explained, although not necessarily at their first occurrence.

A few words are in order on what the book is *not* about. The limitations imposed by our book's scope and extent have excluded the treatment of a number of areas closely related and vitally important to videotex technology. Some of these areas are: social impact, legal and regulatory issues, financial and marketing aspects, field trial and user penetration statistics, the dynamics of service provider activity, and the esthetic and psychological aspects of page creation. For an in-depth treatment of these (and other) issues the reader is referred to a recent book by Tydeman et al. (1982) and to the carefully prepared and informative Videotex Report Series published by Butler Cox & Partners Ltd. in London.

ACKNOWLEDGEMENTS

The author is indebted to many persons for their assistance in the preparation of this text. Dr. Mike Sablatash, the owner of probably the best collection of literature on videotex in Canada, kindly opened up his files and provided invaluable help in documentation. Dr. Walter Ciciora, Professor Herman Maurer, Professor Frank Tompa, Andy Tenne-Sens, Bob Baser, Professor Stan Selkow, Professor Gregor von Bochmann and Dr. Mike Sablatash read the manuscript and suggested a large number of significant improvements in both style and substance. Many people from many organizations in many countries contributed their valuable time to interviews with the author. Special thanks go to Denise St-Michel for her expert work on the preparation of the numerous versions of the manuscript. The author gratefully acknowledges the support of Université de Montréal and of the Natural Sciences and Engineering Research Council of Canada during a sabbatical leave that permitted the writing of this book.

Figures 5.1 and 7.4, and Plates 1, 2, and 3 were provided by Dr. W. Ciciora of Zenith Corporation; Plates 4, 5, 8 by CCETT Rennes; Figure 7.2 by W. C. Treurniet and Figure 7.7 by S. Shlien, both of the Government of Canada Department of Communications (DOC); and Figure 9.10 by Bell-Northern Research. Plate 7 is reproduced courtesy of the DOC. Copyright permissions for materials included in Chapters 6, 10, 12 and 13 are acknowledged to IEEE, Press Porcépic, and DOC.

The specific sequences in all lists of people, countries, systems, etc., in this book should be considered as arbitrary in terms of relative significance. The word "he" used in a nonspecific sense should be interpreted throughout the text as "she or he."

Jan Gecsei

The Architecture of Videotex Systems

Part I
BACKGROUND

The purpose of the chapters in Part I is twofold. Chapters 1 and 2 give a "first pass" assessment of the nature, structure, terminology, and applicability of videotex systems. Chapters 2 and 3 provide background information on related topics, such as complementary videotex-like information systems, computer networks, and the Open Systems Interconnection reference model.

1
Introduction

1.1 THE REALM OF VIDEOTEX

Costs of mass-produced digital devices declined in the mid-1970s to the critical level at which forms of this technology came within the reach of the average consumer. Digital watches and pocket calculators (a "mass-computation medium") were among the first manifestations of this new situation.

Videotex, a new digital mass-communication medium based on a blend of television, communications, and computer technologies, is another development driven by the same cost trend. Just as calculators continue to evolve towards models with more complex features and towards home computers, so is videotex moving rapidly from its initial scope (as a simple means of information retrieval for use in the home), to incorporate advanced applications known so far only in systems designed for specialized users. Therefore, it is becoming increasingly difficult to ask—or rather to answer—the question: What is videotex?

Numerous authors have offered definitions ranging from crisp, factual technical description to almost philosophical statements (that in effect amount to a refusal to define) in which they argue that the technology is not yet mature and stable enough to support a durable definition.

To illustrate the first type of definition, we include three formal definitions of videotex:

1. Videotex consists of "systems for the widespread dissemination of textual and graphic information by wholly electronic means for display on low-cost terminals (often suitably equipped television receivers) under the selective control of the recipient, using control procedures easily understood by untrained users" (see Tyler, 1979, in the reference section at the back of the book).
2. "Videotex is a medium for transmitting text and simple graphics. The usual display is a color TV receiver. The information is digitally encoded for transmission. The information is organized into pages" (see Ciciora et al., 1979).
3. "Videotex is the generic name used for electronic systems that use a modified TV set to display computer-based information. Interactive systems using, typically, the TV set and the telephone line, are called telephone-based or interactive videotex. Broadcast services are called broadcast videotex (teletext)" (see Winsbury, 1979).

As an example of the more cautious approach to definition we quote Plummer (1979):

"Given this prenatal stage in the evolution of the field, we think it most appropriate to consider teletext and videotex as a phenomenon—not just technologies." He goes on to define seven basic dimensions of the "phenomenon": technology, system design, content, users, service providers, economics of system operation, and regulatory and policy environment.

The early date of this observation does little harm to its timeliness. On the contrary: today, when the potential for new applications and technology alternatives is more fully appreciated, and with the emergence of parallel home information systems, videotex becomes increasingly harder to characterize than it was in 1979.

Instead of trying to draw a sharp boundary between videotex and the rest of the world, it is more realistic to imagine videotex as a fuzzy set represented by a number of concentric circles, as in Figure 1.1. Besides serving as illustration and reference, definitions 1–3 above reflect well what is at the hard core of the set: low-cost and simple-to-use information services for the general public, employing the television as display. The degree of membership in the fuzzy set (or "videotex-ness") is highest in the center and decreases towards the periphery as videotex blends into similar and related systems and applications (present and planned for the future).

The fuzziness of the set has been accentuated by the advent of gateways that enable links to be made between videotex and virtually any information or computing service. It might, after all, be less frustrating and more useful to think of videotex in an application-independent manner, as, for example, a new communication medium, value-added television, value-added telephone, or simply a milestone on the way

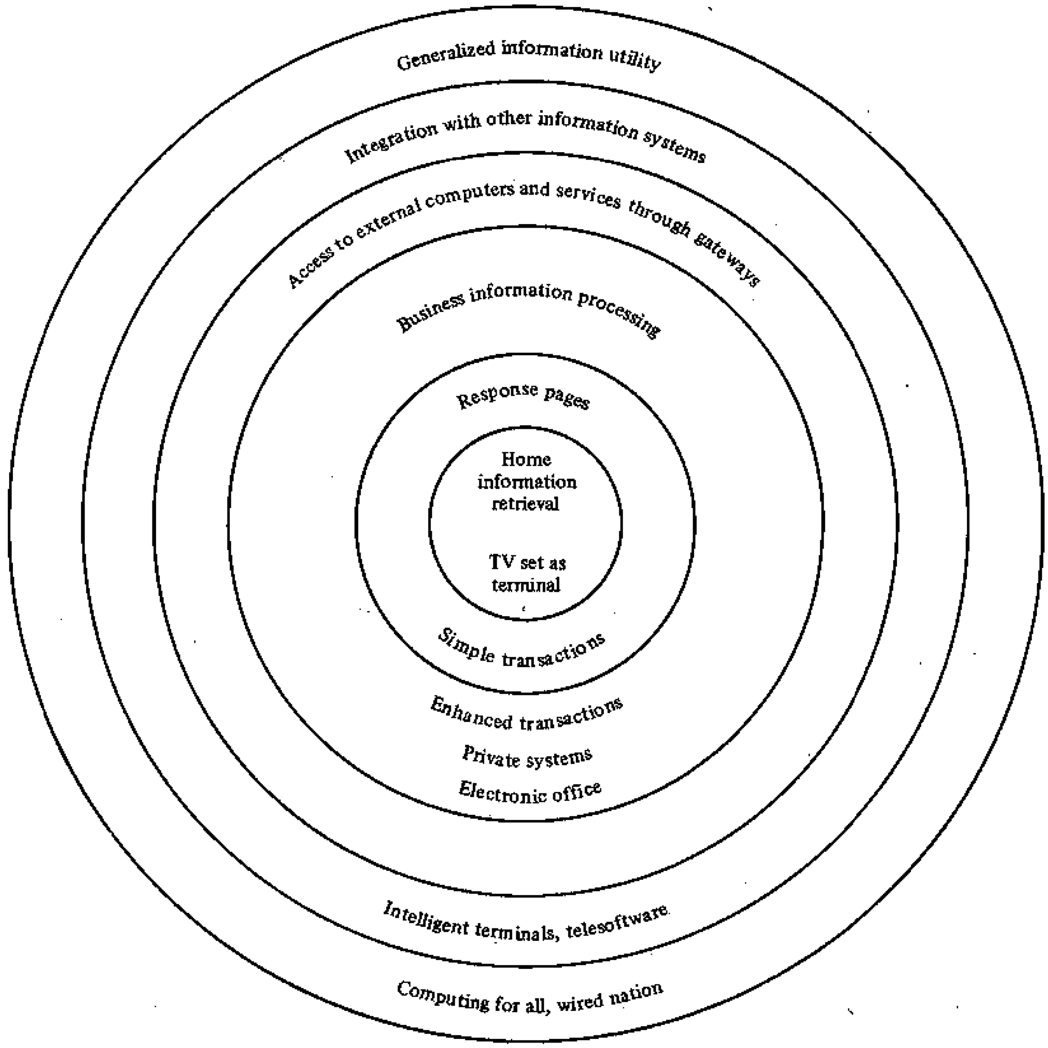


Figure 1.1 Videotex as a fuzzy set: "videotex-ness" decreases toward the periphery.

towards creating the "wired nation." One can even hear descriptions of videotex as "the cheap computer network," "friendly time-sharing with pictures," or "just another terminal."

However, the above observations apply mostly to two-way videotex, which tends to converge with other forms of data processing. Teletext, due to its unique one-way transmission scheme, is less prone to such loss of identity.

As a communication channel, videotex can be roughly characterized in terms of the size of the audience and the time to disseminate information (see Tanabe, 1981).

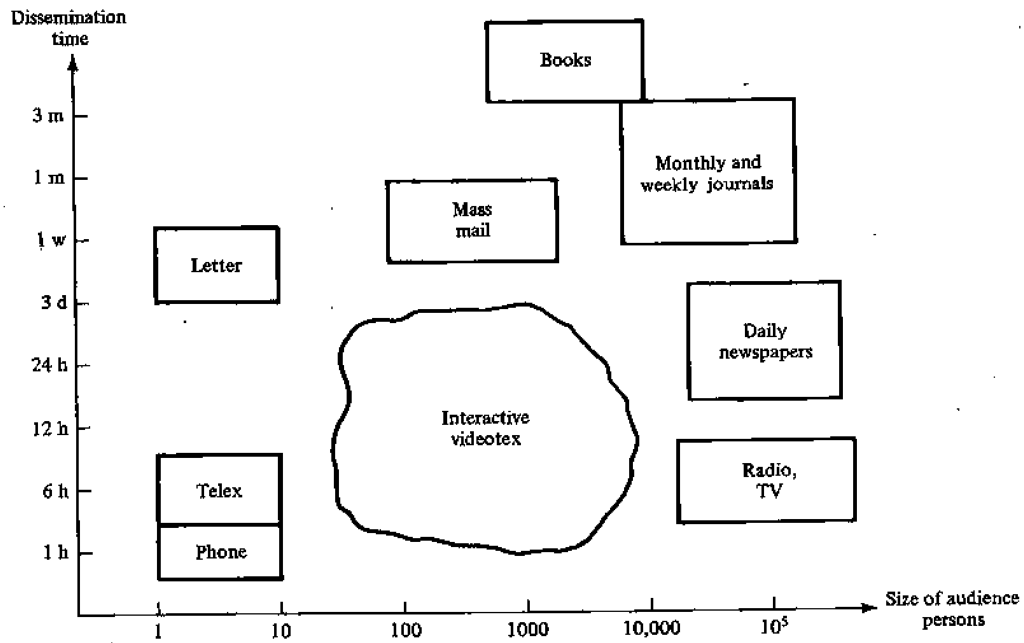


Figure 1.2 Comparison of communication media.

This is shown in Figure 1.2, in which several communication media are plotted. Interactive videotex fills a gap, as indicated.

From a technical point of view, videotex is a special case of computer networks. It is not a brand-new procedure based on some glamorous breakthrough; its originality is rather in its combination of existing technologies. Perhaps the most unique aspect of videotex is that the display characteristics available on television sets (namely color capabilities and limited resolution) are fully taken into account and anchored in a number of presentation standards.

A seldom-mentioned but important consequence of the fact that videotex is largely aimed at the general public and at computer-naive professionals is the prospect (or hope) of planting into the minds of these people practical notions of information technology, disguised as additional functions of the familiar television technology. This strategy, if it works, can lead to tremendous user penetration, proportional to the number of TV sets in use (about 160 million in the U.S. in 1982). Predictions for the percentage of U.S. TV households subscribing to some form of videotex service in 1990 vary between 5% and 90%. A similar strategy aims at gradually supplementing every telephone set in France with an electronic directory terminal. It is such psychological strategy factors, and not the various particularities of coding, display, and possible services, that distinquish videotex from similar information systems.

The prospective user of an information network such as The Source has to make a major decision in becoming a subscriber. In order to make this decision to subscribe, he has to be well-informed about the benefits of the available services, and will probably be sufficiently motivated to subscribe (and buy a terminal or home computer) only if the offered applications match his needs. With videotex, the decision is less dramatic; it is rather like adding a cable converter or a remote control unit to one's television, or buying a more expensive model with new built-in features. This is especially the case with teletext, which requires neither the additional manipulation of a telephone nor an ongoing charge.

1.2 COMPONENTS AND TERMINOLOGY

A typical two-way videotex system (if such a thing exists) is outlined in Figure 1.3. It has four (groups of) main physical components:

- user terminals
- computers
- service-provider terminals and systems
- communication networks to interconnect with other components.

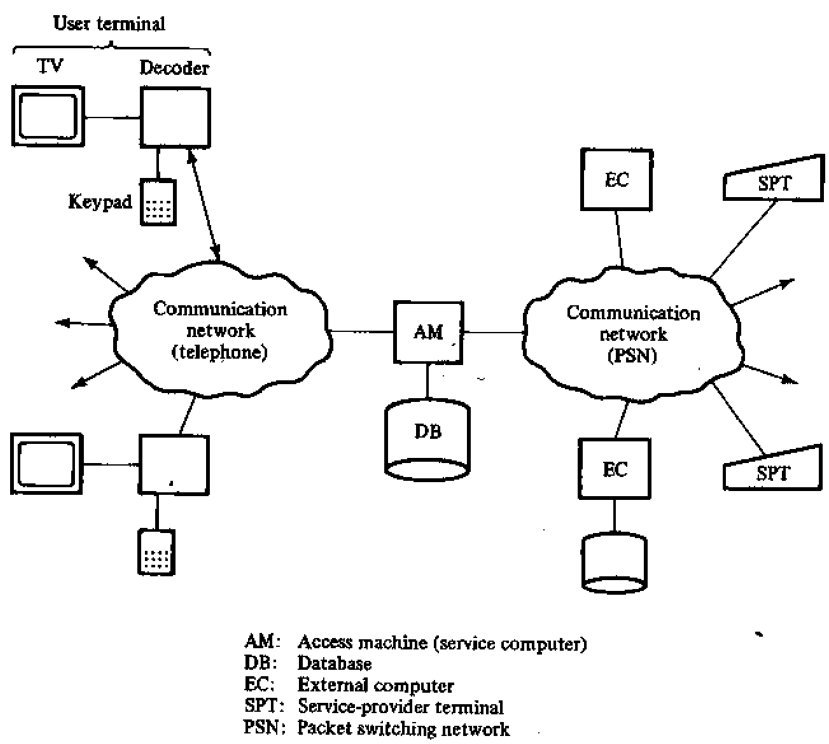


Figure 1.3 Typical two-way videotex system.

Roughly speaking, these components are administered and/or owned by the following organizational entities: *users* (service consumers), who are often the owners of terminal equipment; *videotex system operators*, who run some of the service computers; *service providers*, who supply and maintain the contents of databases and other applications; and *common carriers*, who operate the communication networks. The actual situation is often more complex than this. For example, some computers are owned and operated by independent ("third-party" or "external") organizations, and can also be used for purposes other than videotex. (This would be true in the case of a department store's inventory computer that can process teleshopping transactions initiated by videotex users.) Further, communication networks are often under the mixed jurisdiction of the system operators and the carriers.

From a functional point of view, a basic videotex system can be seen either as a data retrieval system or as a communication medium among users and service providers. Experience shows that, on the whole, the latter view is more appropriate because of the increasing importance of transactional applications and user-to-user messaging, and because the largest demand in database applications is for fast-changing, "hot" data pages, and not for seldom- or never-updated encyclopedic information.

User Terminals

As already mentioned, most user terminals are ordinary TV sets upgraded with additional components: a *decoder* (*controller*) and a *keypad* or keyboard. The controller serves to receive pages of digitally encoded data (text or graphics) and to generate a synthetic image to be displayed on the screen. The user interacts with the system (e.g., to select an appropriate page) via commands entered through the keypad. Most transactional applications require the use of alphanumeric keyboards.

Computers

The computers found in contemporary systems are used in a great variety of ways. Among them, *access machines* (also called service computers) are of central importance. An access machine can best be seen as an intelligent interface placed between users and the rest of the network. Its main functions are handling of dialogs with users and supervising interactions with other computers. Access machines are usually owned and operated by the videotex operator. They may contain local databases and other applications, as well as control functions, such as password verification and billing, that enable them to operate in stand-alone mode (typical for smaller systems).

Application computers are machines dedicated to particular applications (database, banking, etc.). Frequently these applications are designed, owned, and operated by enterprises independent of videotex. In such cases they are called *external computers*. There is much current interest in interfacing with external computers through gateways. Other types of computers in a videotex network may be dedicated to system control and monitoring, service provider support, gateways, or traffic concentration. More details on network components are found in Chapter 6.

Service-Provider Terminals

These terminals serve mainly for the editing of information pages to be attached to videotex databases. The equipment varies in functionality from simple text-editing intelligent terminals to image composition and filing facilities often implemented as stand-alone computer systems.

Service providers (also called information providers) can be divided into two large groups. In the first are those individuals or firms marketing their own information and services (e.g., a travel agency advertising various package tours). The second group consists of specialized information brokers acting as middlemen between clients and the database.

Communication Networks

Virtually all types of links and networks can be employed to build a videotex system. A fundamental distinction can be made according to whether the network delivering information to the users is a *one-way* or a *two-way* system. In one-way systems, also called *broadcast videotex*, or *teletext*, the data base is continuously and cyclically transmitted and available to all users, much like radio or television programs. Selected pages are captured ("grabbed") by the decoder, locally stored, and presented on the display. The user has the impression of interacting with the system (by issuing commands); however, the interaction is on the same level as the selection of a TV channel—that is, it does not exceed the limits of the user's local equipment. Hence the term *pseudo-interactive* is often used to describe this type of system. Television transmission (over the air or cable) is the typical communication medium.

In *two-way* systems, also termed *interactive videotex* or *viewdata*, the user's commands are actually forwarded to the service computer. There, the desired information is retrieved and sent back to the terminal.

Terminology

It is important to remember some idiosyncrasies of terminology. First, *videotex* generally (and in this book) has two connotations: it is used as a generic term for both one-way and two-way TV-based systems, and also as a specific term for two-way systems. This double usage is unfortunate, but the specific meaning is usually clear from the context. The term *viewdata* is commonly used for two-way videotex in Europe.

A second idiosyncrasy of terminology involves the words *teletext* and *teletex*. Teletex, although dangerously close in spelling to teletext, is definitely different from it, and involves an enhanced form of text communication (and is a succesor of telex). We should also mention *videography* as a general term covering all digital techniques for text and image transmission. A related term is the French *télématique* (from the words *télé*-communications and infor*matique*), often used even in English for videotex-like systems.

In addition to being referred to by the above generic terms, particular systems and implementations also have brand names. Here are a few examples, including mostly European names that are often used with definitional connotation (that is, to denote particular transmission or coding schemes):

Antiope: Image coding system used in teletext and viewdata in France; also used to designate the French teletext service

Didon: French system for broadcast of teletext and other data over the TV channel

Télétel: Interactive videotex system in France

Prestel: British interactive videotex system, sometimes also designating the image coding scheme used in the U.K.

Ceefax, Oracle: Teletext systems in the United Kingdom

Telidon: Image coding scheme used in both teletext and viewdata in Canada

Bildschirmtext: Interactive videotex system in West Germany

Captain: Interactive videotex in Japan.

1.3 TELETEXT AND VIEWDATA

Broadcast and interactive videotext offer differing and complementary system characteristics and application potential. The major areas of difference are the database size, the number of simultaneously active users, and the transactional capabilities.

As in other broadcast systems, the number of terminals that can simultaneously receive teletext is virtually unlimited. However, the amount of data that can be included in a broadcast cycle is severely limited by the average time a user can be expected to wait for a selected page. For example, in typical teletex transmission (in the vertical blanking interval of a TV signal) the page rate is about 4 pages/sec. Then, in order to keep the average waiting time at 15 sec, only $4 \times 30 = 120$ pages can be included in the cycle. This limit can be substantially increased by using faster media (full-field TV or fully digital transmission), or by giving up the requirement of (pseudo-) interactivity. In the latter case the user indicates a page (or page set) selection and then allows his terminal to continue automatically capturing subsequently (e.g., overnight) the information for later display.

The situation in interactive systems is quite the opposite. Since every active user must be connected to the access machine and every page request must be serviced by the database, there are limits imposed by the number of physical ports (phone line terminations) and the number of requests serviceable per unit time. Supposing, for example, one disk access per page request and 33 ms/access, we obtain a retrieval rate of approximately 30 pages/sec (if the database is stored on a single disk drive). If the average

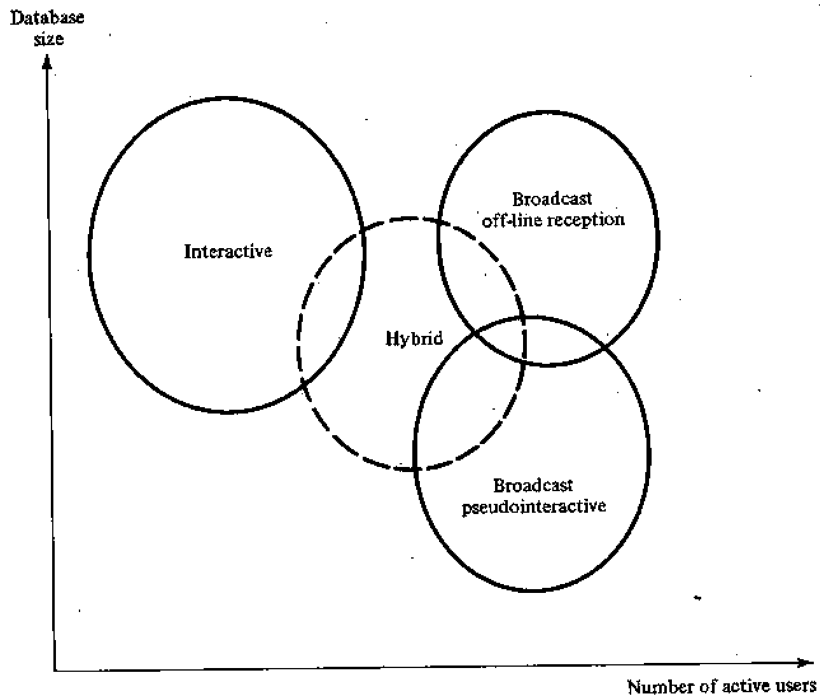


Figure 1.4 Application areas of interactive, pseudointeractive (standard teletext), off-line broadcast, and hybrid videotex.

thinking time between two requests from a user is 15 sec, this leaves space for 30 × 15 = 450 active users. Although the above calculation is oversimplified, it illustrates the point that faster, hierarchical, or duplicated mass memories are necessary in order to satisfy substantially more users.

Proposals have been made of hybrid systems that combine the advantages of broadcast and interactive approaches by cyclically transmitting a small set of highly demanded pages and accessing interactively the remainder of the database. However, to date, no quantitative results are available on the actual behavior of such systems (see Figure 1.4).

Only two-way systems can serve for applications requiring true interaction between users and the service computers. Teleshopping and telebanking are well-known examples.

There are a number of additional differences between teletext and viewdata. Teletext is appealing because it leads to simpler overall system designs and less expensive service computers than viewdata. Viewdata requires the additional availability of a telephone line or some other two-way medium. This may be annoying, given long retrieval sessions that would tie up the phone.

1.4 EVOLUTION

It would make little sense to try to retrace here the evolution of videotex; the catalog of significant events and systems would be far too long and somewhat displaced. Instead, we are concerned in this section with a few important developments such as the key contributions to the state of the art from different countries and the differences between the videotex atmosphere in Europe and North America.

The idea of videotex first surfaced in England; its inventor, Sam Fedida, demonstrated the concept of viewdata as early as 1974. Nationwide public service of Prestel began in 1979. The system now operates a database of more than 200,000 pages. Prestel is the largest viewdata system in operation today, with some 21,000 terminals; the ratio of business versus residential users is about 85:15. There are about 900 information providers (see Hooper, 1982). Teletext can be traced back to attempts in the early 1970s to provide captioning of television programs for the deaf. Ceefax and Oracle, two teletext systems operated by BBC (British Broadcasting Corporation) and IBA (Independent Broadcasting Authority), respectively, have been operational since 1978. In 1982 there were about 550,000 teletext terminals in the U.K.

Developments in France have paralleled those in Britian, with a certain delay. France has a national plan for the development of telematics, including teletext, viewdata, and other public data-processing applications. A number of field trials are under way in France; nationwide public Télétel service has been available since the fall of 1982.

A number of other European countries are conducting field trials with systems based mostly on Prestel technology. These countries include West Germany, Sweden, Switzerland, Austria, Denmark, the Netherlands, Finland, Norway, and Italy.

Canada became involved in videotex in the mid 1970s. The Canadian federal government's Department of Communications financed the development of a geometric coding system called Telidon, and several Telidon-based field trials.

The common feature of videotex evolution in Europe, Canada, and Japan is continued government support and coordination. The resulting policies, systems, and coding standards are being planned on a nationwide and, more recently, on a Europe-wide scale. The leading roles in implementing these policies in Europe and Japan are played by the national postal telephone and telegraph administrations (PTT) in each country. The Department of Communications and the Canadian telephone companies play a similar role in Canada. CEPT (Conférence Européenne des Postes et Télécommunications) plays the leading role in coordinating videotex activities in Europe.

The situation is quite different in the United States, where the initiative is in the hands of private enterprise. There is a minimal interference or support from the government and a lack of coordinated national policy or standards. This has resulted in a much wider range of system solutions and applications than has resulted elsewhere. The U.S. scene is actually a testbed for virtually all known coding techniques and transmission methods.

While the prevalent application of videotex in Europe was originally towards data retrieval, the trend in the U.S. was right from the beginning towards exploratory interactive applications, both home- and business-oriented. Many of these systems are based on cable.

Key Contributions

Each country in the forefront of videotex research has contributed to the state of the art by original ideas and achievements, most of which are discussed later in the book. Here is a partial list.

United Kingdom: The idea and first implementations of videotex; fixed format synchronous transmission with serial attributes; comparatively less emphasis on image-coding issues than on extending available services

France: Asynchronous teletext transmission; conception of teletext as a general data transmission medium; parallel coding of attributes

West Germany: Access to external computers through gateways

Japan: Point-by-point (photographic) transmission of complicated characters; efficient coding methods

United States: Cable-based systems; emphasis on variety of experience and natural selection under free competition; first coding standard proposal incorporating different coding options (mosaic, geometric, photographic)

Canada: Introduction of the geometric coding method

Austria: Systematic use of telesoftware and intelligent terminals for increased system functionality.

2
Applications and Complementary Systems

2.1 VIDEOTEX APPLICATIONS

Videotex in the United Kingdom was originally aimed at general-purpose data retrieval for the home market. After a few years of experience it is gradually being realized that the main application area might be neither data retrieval nor the residential market, but rather, interactive services and retrieval of highly volatile information for the business sector. An almost complete reversal such as this one can be interpreted as good or bad news. To the optimist it is a demonstration of the viability and flexibility of the concept: videotex is certainly good if it can accommodate such a wide span of applications. To the skeptic (sometimes defined as an experienced optimist) it is an indication that videotex is an ill-defined, unstable, experimental idea that will soon disappear amidst competing systems. To the opportunist it may be the signal to act—to attempt to influence the course of videotex history in his favor.

Diversity

The list of actual and possible applications of videotex is endless, for several reasons. First, as we have already seen, videotex is suffering from a crisis of identity; as it stands now, it is a fuzzy and changing set of related services and facilities, each with its own real and potential applications. Second, system operators and service providers (especially in the United States) seem to be directing their attention towards innovative applications, which are hard to predict and to fit into predetermined classification schemes. The reaction of customers is equally unpredictable; often the capacity of people to learn new and complex systems is underestimated. Third, with the emergence of gateways, virtually any computer system and application can be made accessible to videotex users.

To illustrate the arrival rate and diversity of new applications, we include a list of services that were newly announced in the November 9, 1981, issue of *VideoPrint*, a twice-monthly newsletter covering trends in home information, videotex and teletext systems.

1. "First Hand," a videotex trial operated by a Minneapolis bank and using Antiope/Télétel technology was announced. The prime orientation of the system was transaction processing, with data retrieval taking second place. Applications included:

- home banking
- book ordering
- news, weather, analysis
- teleshopping in a department store chain
- software packages, e.g., accounting
- games
- shopping guide, catalogs
- specialized information for farmers (market forecasts, commodity prices).

2. A new version of Electronic Yellow Pages (designed especially for videotex) was added to the Mercury Telidon field trial's database.

3. ARCNET, a local computer network to link up to 235 TRS-80 home microcomputers by high-speed coaxial cable, was announced. Besides standard computer resource-sharing applications (file transfers, word processing), the system was considered as a starting point for a two-way local network.

4. The British Telecom decided to complement the existing Prestel message service with a separate electronic mail system supplied by Dialcom, Inc. (a firm serving about 30,000 electronic mail users). The new system was to be linked to Prestel via the Prestel Gateway computer.

5. According to Prestel management, a number of European countries proposed to interconnect their national videotex systems in 1982/83, using gateways.

6. Rediffusion Computers, a U.K. firm, announced the sale of 46 videotex computers and associated terminals to the Soviet Union for use along the new Siberian Gas Pipeline. The system was to be used according to *VideoPrint* for a "complete maintenance and logistics report and control system."

A Gross Classification

In spite of this diversity of services, it might still be useful to attempt a gross classification of application and customer areas, illustrated with a few typical examples. More details of some applications are given later in the book with the descriptions of particular systems.

Services offered by videotex and similar systems can be classified according to three criteria:

- public versus private
- residential versus business
- retrieval versus transaction orientation.

The first of these concerns the accessibility of a service. Public services are available to virtually anyone, whereas access to private services is limited. Access limitation has two basic forms: either certain areas of otherwise public databases are accessible only to members of a *closed user group*, or the whole system (computer, communications, etc.) is in private hands (typically, a company or institution). Hence, the name *private*, or *in-house*, videotex systems is used in the latter case.

Applications aimed at the home market are exemplified by general-purpose information retrieval (entertainment, yellow pages, news), teleshopping, and games. Stock price quotations and other financial news are oriented towards business use.

Services available through videotex can be roughly divided into retrieval-oriented and transaction-oriented services. The former are centered around the heavy use of databases requiring relatively modest amounts of computing. Transaction-oriented services typically involve some kind of "side effect"—for example, ordering a theater ticket or making a fund transfer with a credit card. These services often involve large databases, too, but they are specialized, and their purpose is not to provide direct information to the user, but rather to support the transaction in question (e.g., a bank database). Another type of transactional service involves various forms of user-to-user communications (real time or store-and-forward).

Some applications clearly fall into one of these categories, whereas others are not very easy to classify. In general, one ought not to get pushed too far by the urge to classify; otherwise, one might end up with only one application per category!

Application Areas

Following is an illustrative list of major application areas, implemented or realistically proposed, and their main characteristics.

Information retrieval. Examples include catalogs, library references, stock quotations, news, and entertainment programs. The databases are usually page-oriented, maintained expressly for videotex in dedicated computers. These applications create much heavier traffic in the downstream than in the upstream direction. Data retrieval is the prime application of teletext.

Commercial transactions. Some uses are teleshopping, banking in the home, paying bills, and reserving tickets. Downstream and upstream traffic are more evenly distributed. Applications can run on videotex service computers, but run more often on the service provider's own installation connected to videotex through gateways.

Advertising and interest matching. Examples include jobs, real estate, and dating services. The simplest form of advertising resembles data retrieval, displaying pages of advertisements. The idea can be pushed further by giving the user the opportunity to respond to the advertiser through use of a response page. Another enhancement is the establishment of two databases, one for "wanted" items, another for "available" items. The system can automatically search through these files and find the best match for a given ad. Relational databases seem to be well-suited for such applications.

Messaging and electronic mail. Again, such applications can be connected to videotex by gateways, or they may be implemented in a service computer. User terminals must be equipped with keyboards. Some text-editing facility must be available either in the terminal or elsewhere in the network. Sufficient storage capacity for mailboxes is required in the computers.

Teleconferencing. This can take different forms: video, audio, telewriting, or typewritten real-time communication among members of a group of subscribers. The main differences between these forms is in the bandwidth and terminal requirements. All four forms require, however, advanced communication facilities to connect arbitrary user subsets.

Education and computer-aided instruction (CAI). The potential of videotext in this area is particularly important, since it enables one to reach large numbers of people who may live in remote locations and have no other way to access the educational material interactively. In such applications provisions must be made for frequent interactions and enhanced graphic-display capabilities.

Computer games. Videotex enables one to extend the range of popular TV games towards more complex types played against the service computer or towards group games as a variation of teleconferencing. However, there is a difference between games graphics and standard videotex graphics. The former utilizes special circuits intended to display rapidly moving game objects. Many games rely on telesoftware for rapid interactions with the user.

Access to general computing facilities and computer networks. This gives the sophisticated user access to all the power and special capabilities of large computers and databanks (there were about 950 public databanks in the U.S. in 1981). Limited computing and personal filing capability may be implemented in the videotex access machine or in the terminal iteself.

Telesoftware. Microprocessors commonly contained in viewdata and teletext decoders can be used to execute programs downloaded from the service computer. This allows tighter interaction with the user, may offload the application computer and the telecommunication circuits, and may provide for a broad class of services that would otherwise be impossible to realize.

Remote monitoring and meter reading. The digital path between a household and a central computer can be used with little extra cost for automatic burglar and fire alarms and for energy management (e.g., programmed heating and utility-meter reading).

Almost all applications would significantly benefit from having alphanumeric keyboards attached to user terminals; certain transactions, such as CAI or general computing, are unthinkable without such keyboards. This fact is reflected in contemporary terminal designs, which either include keyboards or at least provide for their optional connection.

Some of the above application areas and their implications for system design will be taken up in later chapters. In particular, alternatives to the tree-structured databases, database distribution and cooperation, the issue of standards for efficient access to external computers through gateways, and ways to improve the user interface to new applications and telesoftware will be discussed at length.

2.2 COMPLEMENTARY SYSTEMS

The market for mass-information services has stimulated systems that are in many ways similar to videotex. It is often a matter of opinion whether or not these systems fall under the heading of videotex.

For example, networks such as The Source are regularly mentioned in specialized publications (e.g., *International Videotex Teletext News*). In this section we will briefly explore the periphery of the fuzzy set in Figure 1.1. The systems to be mentioned, which may or may not be considered to be competitive to videotex (see Woolfe, 1980), are mostly interactive, and they differ from videotex mainly because they do not use TV-like displays. All applications mentioned are within the reach of videotex through gateways.

Home Computer and Home Terminal Networks

These networks, usually accessible by telephone, offer a broad range of services to owners of home (personal) computers (according to the Yankee Group, there were about 4 million in use in the U.S. by the end of 1982) and ordinary alphanumeric terminals. The services have a mixture of features provided by the network operator (e.g., personal filing, messaging) and those acquired from external sources (UPI wire service). The networks typically offer friendly, simple interfaces.

The Source. This is a nationwide information utility network available by local call from about 300 American cities through the Telenet and Tymnet data networks. In November, 1982, it had 23,000 subscribers (*International Videotex Teletext News*, Nov. 1982). The Source, available since 1979, is the first network of its kind. Its directory of services (*Source Digest*, 1981) lists more than 700 entries. They fall into the following main service categories:

- Communication services, for example:
  electronic mail
  Chat, a program for written dialogues in real time
  bulletin board
- Information services, for example:
  business news
  catalog shopping
  government and politics
  home and leisure
  education
  science and technology
  travel
- Creating and computing, for example:
  filing
  programs from libraries
  programming.

In 1982, the entry fee to the network was $100. The hourly rate was $2.75 during off-business hours and weekends.

MicroNet. This network is similar to The Source; it runs on a general time-sharing system CompuServe. It offers

- general programming
- MicroQuote, a stock price index
- file editing
- software exchange from a central library
- community bulletin board
- electronic mail

The system was opened in 1979. The entry fee in 1982 was $9.00, and the hourly rate was $5.00.

Electronic Mail and Teletex

Users of a computer-based mail system can send each other digitally encoded textual, and possibly graphic messages. Since much of today's mail is generated and consumed by computers, electronic mail can save potentially large amounts of work and paper (possibly with increased security). National electronic mail systems are planned or operational in several countries. Their purpose is to replace parts of manual delivery systems. Videotex systems can be connected to electronic mail computers via gateways.

Text transmission necessary for electronic mail and similar applications can be handled by *teletex*, which is described in a set of new international standards. Besides basic terminal-to-terminal communications (such as in the earlier telex), teletex provides options for facsimile transmission, graphic characters, and storage of messages in the network.

Office Automation

The backbone of the electronic office is a system of communicating terminals and processors. Of the many functions these systems can carry out, the most typical are word processing, memo and document filing/distribution, form editing, form filing, messaging, and internal directory service. It is advantageous if the traditional data-processing functions of the enterprise (order entry, warehouse management, personnel) are integrated, or at least linked, with the office automation system.

Although not ideally suited for that purpose, private videotex systems offer attractive solutions to office automation problems. Following Rouilly (1981), here are some pros and cons of this approach:

1. The terminals are low in cost. Mass-produced videotex terminals are less expensive than those designed especially for office automation. The main drawback is lower image resolution, which is partly balanced by better color capability.

2. Videotex terminals cannot be used for sophisticated editing and word processing. However, these functions are not needed at every work station and can be supplied by the installation of a few special-purpose processors. The needs of the majority of employees can be satisfied by the display quality offered by videotex.

3. Videotex is immediately available from a number of sources as turnkey systems. Some companies offer enhanced functions on their videotex installations, such as memo transmission and distribution, messaging, and telephone directory.

4. There is no need for additional communication infrastructure. The company's telephone network is immediately useable, as long as it has reserve capacity sufficient for the additional load. Unless special modems are used, a telephone connection cannot be used simultaneously for voice and data.

5. There is easy access to other (public or private) videotex systems and external computers. This also offers the possibility of working at home.

Time-Sharing and On-Line Data Retrieval

Time-sharing companies offer programming, computing and file services on large computer systems. On-line databases offer access to large and specialized databanks (e.g., for bibliographical reference, news items, and legal or medical information). Once again, these services are good candidates for connection to videotex users through gateways.

Computer-Aided Instruction

These systems resemble videotex in that the educational material is frequently organized in pages and sequencing is often based on multiple choices. Videotex, therefore, seems to be a suitable medium for CAI implementations. However, typical applications may require finer resolution displays than television can provide and closer interaction (involving program execution in terminals) than is usual in videotex. One possibility is to use a dedicated videotex system or an intelligent terminal (or concentrator) into which the entire CAI course is downloaded from a central databank.

Private and Videotex-Derivative Systems

Government-sponsored development of national videotex networks (Prestel, Télétel) have stimulated a great deal of interest in the world of business data processing. Many inherent features of viewdata have been found attractive and enterprises have discovered the potential of using videotex for business applications (as well as for office automation). Today business users form an important and growing segment of the total customer population. In fact, there were about 100 private videotex systems in the U.K. by early 1983.

As mentioned previously, there are two large groups of business users of videotex technology: those who subscribe to the services of public systems, and those who prefer to install their own in-house systems. The difference between the two groups is similar to the difference between customers of general time-sharing services and owners of data-processing systems.

In return for the benefit of lower initial investment and fast availability, users of existing systems (like Prestel) have to fit their applications within the rigid limits of the host system. On the other hand, private videotex systems can be made to incorporate otherwise unattainable functional and performance features. These include, for example: nonstandard database structures and retrieval commands suited for specific application (e.g., Optel, the Open University's viewdata system); higher than normal transmission rates (9.6 kb/s in Topic, the London Stock Exchange system); and terminals with nonstandard geometric options or which are capable of displaying 40 or 80 characters on a row. Many applications require specific additional functions: terminal-to-terminal messaging, personal filing, and corporate directories are standard components of videotex-based office automation installations. There is a tendency to integrate such private systems with the existing data-processing equipment of the enterprise.

Some large computer manufacturers such as IBM and Digital Equipment seem to recognize this by developing their own versions of videotex. For example, IBM's prototype system described by Grimm (1982) is based on an IBM Series/1 computer. The basic system is similar (for the user) to Prestel; however, the main emphasis is not on maintaining large videotex databases, but rather on the conversion of data from other sources (not uniquely IBM) into videotex format. This conversion can be done in batch or real-time mode.

Through a process of adaptation to the needs of applications, many private videotex systems have become quite dissimilar to their public cousins and ancestors. Some of the remaining attributes and factors that warrant use of the term "videotex" in describing these derivative systems are:

- the use of terminals fabricated for videotex
- presentation level compatibility with videotex
- the use of telephone networks (public or in-house)
- access to public videotex and external computers via gateways
- access to parts of a private database from a public system
- the fact that private systems are often designed by videotex vendors (e.g., the French company Steria), and that the software is derived in an evolutionary way from videotex software
- the fact that the *spirit* of the private application is akin to the original idea of videotex: inexpensive access by many nonspecialized people to relatively simple services (with the obvious qualification that what is considered complex now will be "simple" in a few years' time).

3
Videotex Architecture

3.1 COMPUTER NETWORKS

Since videotex systems are a form of computer network, we shall give a short introduction to this latter area. For a comprehensive treatment of the subject see Tanenbaum (1981) and Martin (1981). *Computer networks* and *distributed systems* are related terms designating systems in which information processing is done on multiple, remote, mutually interconnected, and cooperating computers and other equipment. These systems came about as a result of developments in hardware fabrication technologies (economically enabling many computers and intelligent peripherals to be included in a system), and in telecommunications (especially data networks which enable computers to be interconnected efficiently). Resource sharing and other advantages inherent in computer networks often provide better, more economical solutions to the ever-increasing computing needs of society that are more flexible than those possible with centralized or separate computers.

Although hundreds of computer networks are in everyday use and will undoubtedly grow in size and sophistication, there is still no universally accepted criterion to determine just where the boundary between centralized and distributed systems is. Without going into details, we shall mention two criteria useful in getting some appreciation for the field and in locating the position of videotex appropriately in the space of computer networks. The first criterion is the *tightness of coupling* between participating computers. Tightly (or closely) coupled systems typically communicate through a shared main memory to which both computers have access. The units of exchanged information are small (e.g., a memory word or record), interactions are frequent (say, 100 – 10,000 per second) and the time to exchange information units is comparable to the instruction execution time. The communication medium is the processor bus and the physical distance between nodes is small (e.g., they may lie within a room). According to some views, such tightly coupled multiprocessors do not even fall into the category of networks.

The features of a loosely coupled system are quite opposite: interactions are infrequent (say 0.1 – 10 per minute), transmission delays are long (1 – 10 seconds) and the units of exchanged information (usually called messages) are in the order of hundreds or thousands of bytes. There is no common memory between systems, and the transmission media are telecommunication links and networks. The physical span of such loosely coupled networks may be thousands of miles. Clearly there are a great variety of intermediate forms between the above two extremes; one well-known example is the local area network.

The second criterion helpful in classifying distributed systems is the nature of what is really distributed in the system and what remains centralized. In particular, the distribution may involve data, processing, control, or any combination of the three. Martin's view (1981) is more detailed, distinguishing six distributable elements: application processing, files, databases, input/output, network control, and intelligence. A database whose different files—or records of the same file—reside in different nodes is an example of *data distribution*. A transaction-oriented system in which a unit of work (transaction, job) is passed between several computers, each specialized in a partial activity (compilation, data retrieval, etc.) exemplifies *distributed processing*. Finally, we speak of *distributed control* in systems in which operating system functions, such as coordination of events, resource allocation, data movement, and so on, are also distributed. Much theoretical and practical research is centered on these issues.

In some systems a primary design goal is to make distribution transparent to the end user. Such transparency would give the user the impression of dealing with a single centralized computer.

Computer networking, as most things in life, has its advantages and dangers (the latter being sometimes underestimated). Among the most-often-cited advantages are increased availability and reliability, better cost effectiveness, and increased capabilities for communication between people. The disadvantages stem mostly from the fact that networks are inherently more complex and harder to design (and to use properly) than are centralized systems. Also, maintenance of security is more difficult than it would be if data, communications, and computation were all contained in one room.

With regard to videotex, these emerging and rapidly evolving networks appear at the present to involve mostly data distribution. Access to complementary, partial, and external databases is of great current interest, and mechanisms for demand-based automatic migration of data pages between computers are being designed. There is less distribution of processing and control in videotex, although some processing functions for a transaction (such as access verification, maintaining of "user contexts" or keyword dictionary lookup) may be executed in access machines (concentrators) rather than in main database computers. Also, the use of external computers involves transfer (not necessarily distribution) of control and processing. At any rate, presently known videotex configurations do not yet rely on the sophisticated and theoretically interesting features of distributed control that are in the spotlight of the current literature on distributed systems; the emphasis is rather on compatibility, efficiency, and user interface problems. Insofar as videotex becomes in the future a generalized computer utility, this situation may rapidly change, however (especially given the eventual impact of telesoftware and intelligent terminals, discussed in Chapter 14).

3.2 THE OSI REFERENCE MODEL

All forms of distributed computing imply complex cooperation between the participating node computers and between the computers and communication media. This in turn leads to the necessity of *standards*. Without appropriate standard ways of interworking, each instance of a distributed system would be a rigid, unchangeable, and isolated construct with little space left for upgrading, component replacement, expansion, and linkage with other similar systems. This fact has been realized by many major manufacturers, who reacted to the new need by introducing their own standards; examples are DEC's DECNET, IBM's SNA (System Network Architecture), and Honeywell's DSA (Distributed System Architecture). Although these standards are conceptually similar, they are sufficiently different to prevent easy interworking between, say, a DECNET- and an SNA-based system. The need for a common, product- and supplier-independent standard has been recognized by international standards bodies, such as ISO and CCITT.

In 1979, ISO proposed the so-called reference model for Open Systems Interconnection (OSI). The model is not a standard in the usual sense of the word; it is rather a "metastandard," describing the general structure and relationships in a proposed system of standards whose goal, when implemented and adhered to, is to enhance interworking within distributed systems. The word "open" is intended to indicate openness to growth and cooperation in future systems.

The reference model has generated a great deal of interest in the computer and telecommunications communities, and many excellent works exist on the subject (e.g., Zimmerman, 1980). The following is a short account of its principal philosophy and recommendations.

The essence of the philosophy behind the OSI model is the recognition that interworking between nodes of distributed systems may take place at different levels and that the interactions on each of these levels are quite independent of those on other levels. This philosophy in turn partly follows from the fact that human designers think of complex systems, such as computers, in hierarchical terms, trying to divide them into subsystems (levels, layers) and to deal with each as separately as possible. The motivation is to reduce to manageable proportions the complexity of subsystems.

The model defines seven *layers* on which interworking can take place, as shown in Figure 3.1. Each column shows the layers in a node (typically, in one location). We note that not all layers have to be present in all systems and that there is nothing absolute

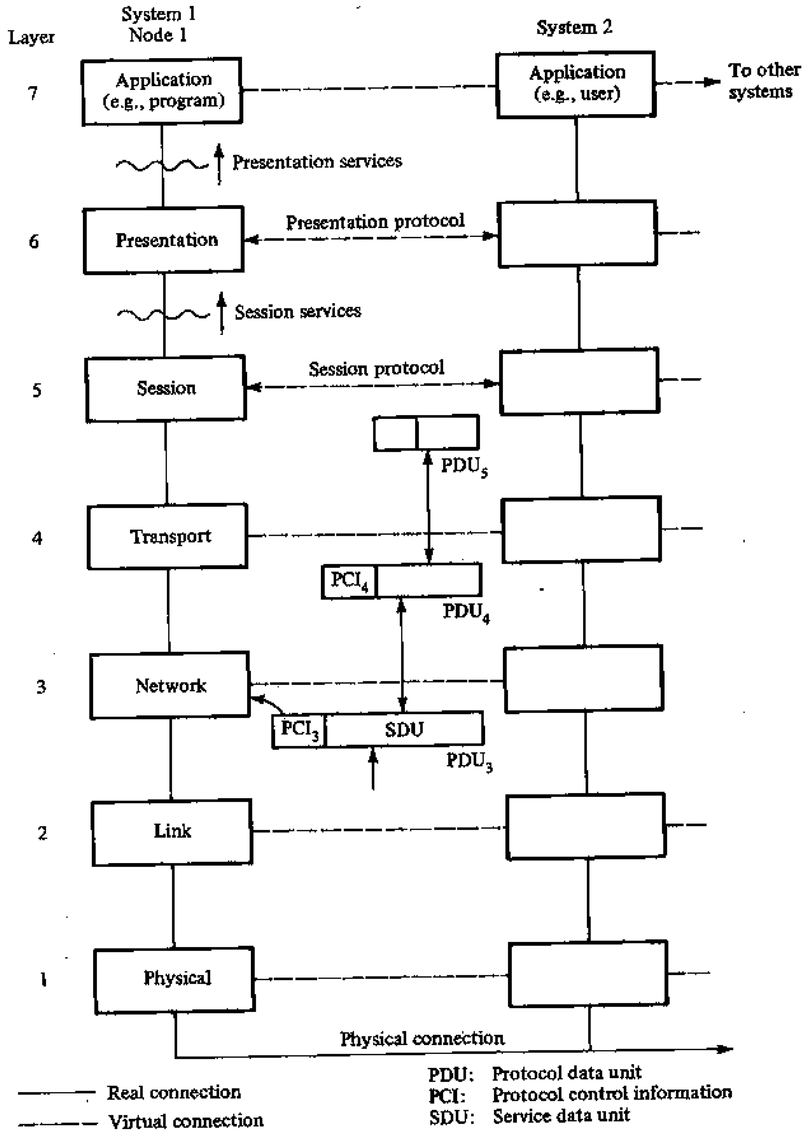


Figure 3.1 ISO Open Systems Interconnection reference model.

about the number of layers: seven is essentially a compromise between too few layers (each of which would be too complex) and too many layers (which would necessitate definitions of many interfaces and standards). Meaningful communication takes place between *peer* entities (processes), which are equally numbered layers in two nodes. This communication is carried out according to a set of rules, interaction sequences, and data formats, collectively called *protocols*. ISO is proposing services and protocols for each layer; some of these protocols are already defined, others are being studied by different study groups (see below).

The layers and approximate protocol functions in Figure 3.1 are the following:

1. The physical layer defines the physical and electrical characteristics of the transmission medium and provides for the transfer of a bit stream between systems.
2. The link layer provides for error-free transmission of information records over a communication link (line), as well as a certain amount of flow control to prevent one node from flooding another with information faster than it can be received.
3. The network layer controls transmission and routing of messages through communication networks.
4. The transport layer provides *end-to-end* communication between two peer processes. The term "end-to-end" indicates that the peer processes appear to talk to each other directly. Lower level protocols often span only parts of the path between two nodes, as drawn in Figure 3.2.
5. The session layer coordinates logon, option negotiation, application selection, and other information exchange (e.g., "your turn") necessary to establish and conduct correctly a dialog between two applications (sometimes even despite the presence of errors in layer 4).
6. The presentation layer determines the way in which information (characters and other data structures) are interpreted by the output (e.g., display) devices, including data compression and encryption.
7. The application layer determines the contents (meaning) of the dialog between the user and the selected application program.

The OSI model should be regarded as a broad guideline for system design and not as a straitjacket into which all systems must fit. The strictly hierarchical relationship is questionable in some cases and the identity of some levels may be ambiguous. For example, the rules defining the keypad commands in Prestel-like systems appear to belong to both levels 5 and 7. It would also be difficult to assign to a level an eventual telesoftware standard.

Communication between peer entities (shown by broken lines in Figure 3.1) does not take place directly in each layer; the only physical (real) transmission is on level 1, and all other functions up to the application are built up in successive layers, each relying on the *services* of the next lower level, and adding some functional elements (e.g., error detection, retransmission, and concatenation of received records into longer messages) and providing this new service to the next higher level. Therefore, horizontal connections in the diagram, each governed by an appropriate protocol, are sometimes called *virtual* communications.

Communication between adjacent layers of a given node is in terms of information blocks called Protocol Data Units (PDU). A typical PDU has two components: Protocol Control Information (PCI), used for the protocol services in a given layer, and Service Data Unit (SDU), which is the (uninterpreted) information sent to the next higher level, where it is treated as a PDU and the appropriate PCI is removed and processed. The process is reversed in the downwards direction, as also shown in Figure 3.1.

Standardizing the Protocols

Following the generally positive response to the OSI reference model, ISO is now in the process of specifying standard services and protocols for each layer. Most of this work is currently concentrated around layers 4 and 5. Existing international standards such as X.21, HDLC, and X.25 (for packet switching networks as described in Chapter 6) are appropriate for levels 1−3.

On the transport level, there is a newly emerging protocol (ISO, 1982), defining five classes of end-to-end services. Class 0 is the basic teletex service; classes 1−4 introduce further functions, such as error recovery and multiplexing, and their combinations.

A session level protocol is also being studied by ISO and CCITT. Functions to be provided include session establishment/termination, option negotiation, token control (e.g., controlling whose turn it is to transmit), and checkpoints. Further services also under evaluation include virtual terminal and virtual file access, using the above functions and probably placed in layers 6−7.

3.3 VIDEOTEX IN OSI PERSPECTIVE

Since videotex networks are typically very large, spanning large geographical areas and including all levels identified by the OSI model, and since interworking between systems on national and international scales is a desirable objective, standardization is of great importance. There is quite hectic activity in the international videotex standardization arenas and although most of the newer efforts claim to fit the OSI framework, it is, unfortunately, not likely that a unique set of protocols will be agreed upon. In this section we present a typical videotex system, indicate its correspondence with the OSI model, and list some existing and emerging videotex-specific standards.

Figure 3.2 shows a somewhat simplified viewdata system consisting of user terminals connected to an access machine (AM) through the public telephone network, with the AM linked to a remote database via a packet switching data network (PSN). Service-provider terminals are linked to the database through the PSN. The brackets under the diagram indicate the approximate span of some protocols that could be found in this system. Lower-level protocols control individual transmission lines connecting to and within the data network; the presentation- and application-level protocol's control extends (in this example) from the terminal to the database. However, this may not always be the case—for example, in the Bildschirmtext gateway (discussed in Chapter 11).

As mentioned above, lower protocol layers are fairly well established and are not videotex-specific. They consist (for the PSN) of the CCITT X.25 standard and provide for reliable transmission of messages between the database and terminal (user or service provider). Note that in this case two instances of X.25 (AM-PSN and PSN-DB) are present. For transmission over the telephone network these layers are defined by the

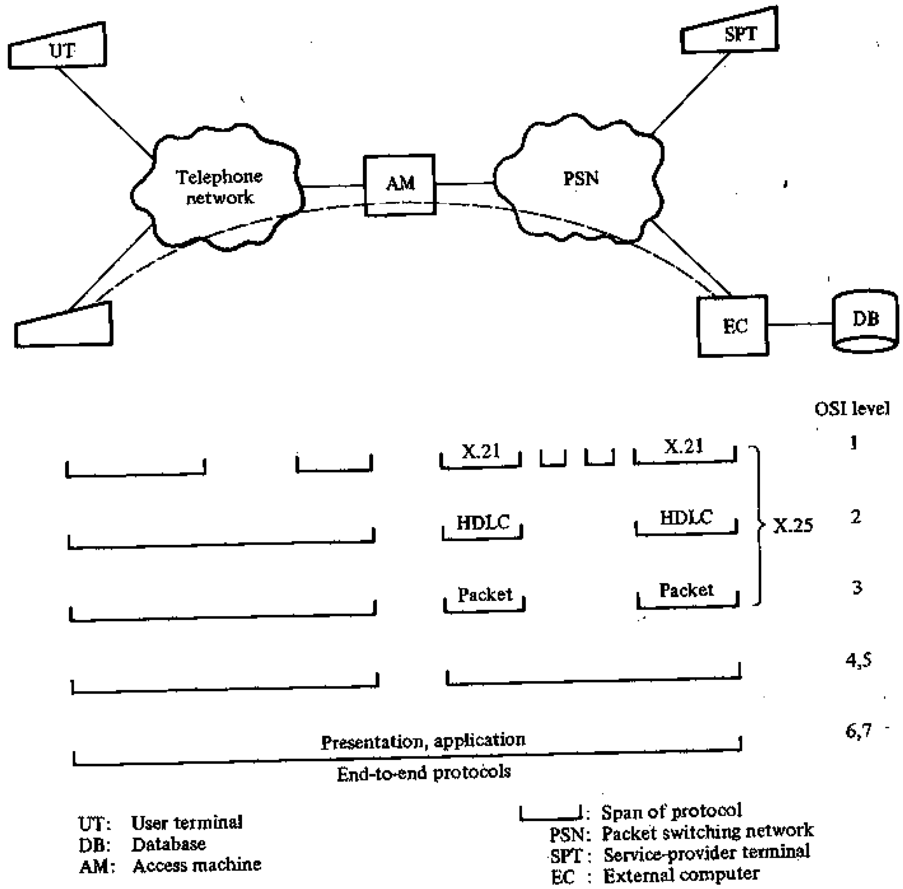


Figure 3.2 Some protocols and their spans in a typical videotex system. In this example a retrieval session involving a distant database is considered; the session is set up and controlled by AM. The transport and session protocols are not end-to-end here, whereas they may be in other systems.

parameters of modems, the asynchronous transmission, and the dialing conventions used. However, the corresponding levels in one-way systems are quite different, as exemplified by the France-Canada-CBS proposal (discussed in Chapter 5). At the session level, a protocol proposal published recently by AT&T (Bell, 1981) consists essentially of control codes for selecting presentation-level coding options. In other systems the session level consists of logon and logoff procedures, the rest belonging to the application layer. Level-6 standards are very important for videotex since they determine, among other things, the way in which pages to be displayed on color TV screens are to be encoded and stored. Pertinent standards are the CCITT S.100, the North American PLPS (essentially AT&T's Presentation-Level Protocol), and the CEPT European proposal. These and other protocols are dealt with (and referenced) in Chapter 8. Finally, the application level includes in most systems data retrieval from tree-structured databases;

the protocol proper is inherent in the syntax and semantics of the user commands and their relation to the database. These are the subject of the F.300 recommendation by CCITT. Other, mostly transactional, applications, such as teleshopping or banking, are rapidly gaining importance, and corresponding protocols are being proposed.

Architel Protocols

The direction of French Telecommunications initiated in 1982 the development of a set of OSI-compatible protocols, aiming at assuring compatibility between different telematic and other public information services (Télétel, teletex, facsimile, Electronic Directory, electronic mail, and others).

Architel (1982) covers OSI layers 4–6. The transport protocol is identical to Classes 0 and 2 of the ISO transport protocol (see above). Class 0 provides for a simple end-to-end connection without multiplexing or error recovery. Class 2 includes multiplexing and optional flow control.

The session protocol is aligned with the preliminary versions of the ISO protocol. Services (provided to the presentation layer) include connection establishment and termination, as well as control of data transfers. Two kinds of connections are supported, suitable for interactive (conversational) and document transfer sessions, respectively. The former type is suitable for interactive database access and data collection. Several kinds of tokens (e.g., data, synchronization, and termination) are used as means of controlling the protocol's actions; possession of a token means essentially the right to execute an action (e.g., send a data message).

The Architel presentation protocol is concerned with the ways in which information is displayed to the user and in which the user can respond to this information. The display screen (along with some supporting data structures) is considered as a "common space" shared by the application program and the user (peer application-level entities), and through which they interact. For this reason, the protocol is also called *virtual terminal* protocol.

The virtual terminal is a data structure whose main components are:

1. *Parameters of the terminal*: These include its resolution, color capabilities, whether it is capable of data collection, and many other facts. These parameters (the "terminal profile") are transmitted during session establishment to inform the peer presentation entity about the terminal's capabilities and to find out whether presentation-level cooperation can take place. This is generally called "option negotiation."
2. *Main data structure*: This consists of the *document* to be presented. Its components are:
   - (a) *Parameters of the document*: These include, for example, the size of the page, number of lines on the page, and number of characters on the line.
   - (b) *Page(s)*: These consist of *windows* (rectangular areas serving to display or to collect information), each having its *contents and attributes* (e.g., color) and belonging to a *class* (mosaic, geometric, photographic). A page may also have

a program associated with it, to control, for example, the sequence in which windows are to be displayed.

3. *Complementary data structures*:
   (a) *Cursors*: Show the position on the screen of the current page, current window, and current character; they can be controlled independently of each other.
   (b) *Color tables and dot patterns*: Define certain details of the interpretation of received displayable information.
   (c) *Dialog controller*: A finite-state machine controlling the sequence of data transfers, using the session level services.
   (d) *Working memory*: Holds variables for the above structures.

From this brief review it should be clear that "presentation protocol" may have many meanings and scopes. For example, the protocols and standards to be discussed in detail in Chapters 7 and 8 cover only a small (but important) subset of the virtual terminal protocol (essentially, the point covered under *Page(s)* above).

Part II THE COMMUNICATIONS LEVEL

Introduction

Videotex data can be transmitted from its origin to the user's terminal by virtually all available data communication media. Many of these are hidden inside communication networks and have no noticeable influence on the videotex service. For example, long-haul lines in telephone networks may be implemented as microwave, coaxial, or satellite links; however, this implementation has little influence on the end-to-end characteristics of phone circuits.

Chapter 4 deals with those transmission media and systems whose parameters have a rather direct impact on the operational characteristics of the videotex service supported: the media and systems used for the *delivery* of information to the user terminals. The principal physical media we consider are the public telephone network, television broadcast, coaxial cable-based systems, and fiber optics. Chapter 5 is entirely devoted to teletext, describing its technical foundations, implementations, and some standardization issues. Chapter 6 provides an overview of videotex network topologies.

Classification

Delivery systems can be classified according to four criteria:

- whether dedicated physical paths are provided to each user terminal, or physical channels may be shared by many logical paths

- whether they can transmit information in one or two directions
- whether individual users can be addressed—that is, whether information can be sent selectively to users or user groups as opposed to being broadcast to anyone who may be listening
- rate of data transmission.

In order to prevent confusion, it must be emphasized that the functional characteristics of a given delivery system generally depend not only on its physical nature but also on the higher-level protocol procedures used in the system. For example, a one-way cable system may or may not incorporate addressability.

There are three broad classes of delivery systems in use today: dedicated two-way, such as telephone; shared one-way, such as TV broadcast (off-air or cable); and shared two-way, such as two-way cable systems. Systems based on one-way delivery are often identified as teletext.

In shared two-way systems, the conflicts due to simultaneous upstream transmission by several users must be resolved. The methods available are basically those used in local area computer networks.

Finally, we must point out that a particular videotex system may rely on several delivery media—for example, one-way cable and telephone. Such systems are sometimes called *hybrid*.

Delivery Protocols

We are interested here mainly in the fashion in which videotex information is delivered to the user's terminal, and not in the way this information is interpreted by the terminal and used by an application. This is a useful, although possibly ambiguous distinction (e.g., in teletext, page numbers have to be analyzed before it is determined whether a page is useful at a given time; this processing may be thought of either as part of the interpretation taking place in the terminal, or as part of the delivery process). Without going into details, we can say that in terms of the OSI model, the separation between delivery and interpretation functions is somewhere around the session layer. In this part of the book we are thus concerned with the protocol functions associated with layers 1–5, treating them occasionally as a single *communications level* that executes the *delivery protocols*.

This low-resolution view of OSI is justifiable for several reasons. First, due to the particularities of present videotex, some layers (such as the transport layer) are sometimes missing and other layers are hard to identify. Second, many systems were not designed with OSI in mind, and the attempt to translate their notions and terminology into the terms of the seven OSI layers would result in more confusion than clarity. However, this situation is changing as a result of the mounting pressure for standardization. An attempt to define a complete teletext system in OSI terms is found in Sablatash and Fitzgerald (1982).

- Delivery protocols have the following functions (illustrated in Figure 4.1):

  Bit and byte synchronization: This assures the correct reception of a series of bits through the medium and the appropriate recognition of character boundaries. These operations affect the clock and bit-counter circuits in the receiving equipment.

  Error protection: Deleting or correcting errors that may occur during transmission.

  Data identification and continuity checking: This can involve, for example, processing of line, page, and magazine numbers in teletext.

  User addressing: Here belong processing of user addresses, closed user group identifiers, and channel numbers (in systems such as Didon). These functions are applicable in systems with shared addressable media.

  User command handling: Certain aspects of this are executed locally by the terminal in one-way systems and distantly in two-way systems.

4
Delivery Media

4.1 THE TELEPHONE NETWORK

The public telephone system is the best-established and most extensive communication network we know. There are some 300 million telephones in the world, 6 million in New York City alone, and practically every household in North America has access to a telephone. In principle at least, every phone in the world can be connected with every other. The cost of locally using a phone is well within the reach of the average person in the industrialized world.

The telephone network can be thought of approximately as a hierarchy of exchanges (switching offices), concentrators, and toll offices, interconnected by communication lines of different capacity. Generally the characteristics of these lines depend on their position in the hierarchy: at higher levels, links have higher bandwidth, are multiplexed, and span longer distances. Subscribers' phones are connected to local exchanges (lowest-level exchanges) through dedicated two-wire circuits called *local loops*.

The original purpose of the telephone system was, of course, communication of analog speech signals, and not transmission of digital data. Consequently, the bandwidth of a typical switched voice-grade circuit is about 3000 Hz (frequency range from 300 to 3400 Hz). However, this limited bandwidth applies only to end-to-end circuits in the network, that is, from phone to phone. Individual parts of the system can often carry much higher data rates. The end-to-end bandwidth is artificially limited in order to exploit

Function	Dedicated 2-way	Shared 1-way	Shared 2-way
Bit, byte synchronization	√	√	√
Error protection	√	√	√
Data identification continuity (page, magazine)	N.A. (only demanded pages sent)	√	√
User addressing	N.A. (done in access machine)	√ (In addressable systems only)	√
User command handling	√	Local function in terminal	√

Figure 4.1 Delivery protocol functions in different media.

better (i.e., to multiplex) the long-haul circuits. Thus, for instance, the local loop itself has a bandwidth of almost 1 MHz, and could therefore easily carry a digital voice circuit of 64 kb/s. But the extension of such circuits throughout the network would require expensive upgrading of the whole system. And since videotex, in order to be viable, must be cheap, those videotex systems that rely on the telephone network for delivery use it as it is. However, there is a trend toward the digitalization of the telephone network, as will be discussed later.

Modems

The most common way to use the switched telephone network for digital transmission is through devices called *modems*. The function of modems is to transform binary signals into a form suitable for transmission within the available bandwidth. Since the bandwidth does not extend down to 0 Hz, the circuit would not transmit well the dc levels corresponding to binary 0's and 1's. Instead, most modems employ a frequency modulation technique with four distinct frequencies used to carry 0's and 1's in both directions.

Low speed modems, such as those typically used for videotex, work in *asynchronous* mode. That is, each eight-bit character is bracketed between a start bit and a stop bit used to re-synchronize the bit-clock of the receiving modem. Thus, individual characters can be followed by pauses of any length.

Modems are available for a range of standardized transmission rates from 75 b/s to 9600 b/s. Speeds up to 1200 b/s (approximately 120 char/s) through switched circuits are attainable with relatively inexpensive modems; higher rates require specially designed modems, coding techniques, and, eventually, conditioned and/or dedicated (leased) lines. We point out that when the switched network is used, the physical circuits involved in the connection may be different from one call to another (except for the local loop).

Again for obvious economic reasons, low cost modems are a must for videotex. Fortunately, 1200 b/s is an acceptable speed for most picture-coding techniques, with the exception of photographic coding. (This is no surprise: in effect, image-coding methods for videotex were originally conceived with this limited bandwidth in mind!) At that rate, transmitting a full alphanumeric page takes less than 10 seconds, about 0.5 s per line, which is decidedly higher than the reading speed of an average person. About a minute is needed to build up a photographic image on the screen; this is decreased to 15 s at 4800 b/s, a quite acceptable delay.

Data Rates

In two-way videotex a reverse channel must exist to transmit user commands to the service computer. The bandwidth required for this is much less than it is in the forward direction; in most systems 75 b/s is used (150 b/s in certain trials in Canada). A CCITT recommendation (V.23) exists for the communication between computers and user terminals. This recommendation in effect specifies full duplex asynchronous communication at 1200/75 b/s in the forward and reverse directions, respectively. Most existing systems use modems complying with V.23. The provision for full duplex communication is important, however: it permits a command to be sent before a page transmission is completed in order, for example, to shorten the waiting time for the next selection, or to block the transmission if the page is of no interest to the user.

The 1200/75 b/s system is sufficient for data retrieval applications in which the user input is generated by typing and sending a few characters. But in other circumstances (such as local editing of messages in an intelligent terminal and sending of the entire message at once), rates of 1200 b/s may be needed in both directions. This is completely within the possibilities of the switched telephone network, and suitable modems are currently available.

Higher transmission rates are required for photographic and quasiphotographic image coding. For example, Captain uses 3600 b/s, and the British Telecom is experimenting with 6800 b/s for its Picture Prestel service (not yet publicly available).

Some form of flow control must be used in order to prevent overflowing of the receiver's buffers. In Telidon, control character DC3 is used to request the service computer to stop transmission, and DC1 is used to resume transmission.

There is much discussion of the merits and drawbacks of the telephone as a videotex delivery vehicle (see Link Research Memoranda, 1980). One of the original motivations behind Prestel was to improve the utilization of the network in off-peak hours (evening). Although this rationale is entirely valid for home applications (the initially targeted marketplace), it is almost entirely invalid for business applications, most of which coincide with peak telephone traffic. The main arguments about the suitablility of the telephone as a videotex delivery vehicle are summarized below.

Advantages:

1. It is a well-established system with very high penetration
2. It has ports (user entry points)
3. It has switching and addressing capability
4. It covers large geographical areas
5. It offers full duplex two-way communication
6. It has computerized functions similar to those required for videotex: automatic user identification, billing, programmable access restrictions, and facility to broadcast some messages.

Disadvantages:

1. It has a low data rate
2. It was not designed for the bursty traffic of videotex—i.e., long user sessions may be costly
3. It prevents normal use of the telephone (although proposals exist for use of a single telephone line for simultaneous voice and data transmission).

4.2 INTEGRATED SERVICES DIGITAL NETWORKS (ISDN)

Although many of today's public communication and mass media services are based on analog transmission technologies (telephone, radio, television), digital transmission is making fast and steady progress. The main reasons for this are that digital information is easier to protect against distortion and errors, and that computer technology can be readily applied to build and control sophisticated data networks. On a different level, the need for cooperation of distant computers has led to computer networks in which there is little place left for analog information.

Digital Carriers

All information services reaching the public, including television, can be coded and transmitted in digital form. Pulse coded modulation (PCM) transducers, called *codecs*, are currently available to digitally encode acoustic information, and a growing number of long-distance links carry telephone traffic in digital form. In the U.S., there are nearly 160 million km of T-carrier links (1.5 Mb/s) and about 250,000 digitized local loops installed. Figure 4.2 shows some commonly used digital data rates and carriers. Bell System plans a switched digital capability over the public switched network, and a large number of digital exchanges and toll offices are already installed and working in most Western countries. Cable companies are introducing into the customers' homes very large bandwidth connections (up to 500 MHz), which, though presently used mostly for

Multiples of 1.2 Kb/s:		
	1.2	Used over voice-grade lines with modems
	2.4	Used over voice-grade lines with modems
	4.8	Used over voice-grade lines with modems
	9.6	Used over voice-grade lines with modems
	19.2	
Multiples of 56 Kb/s (= PCM voice channel):		
	56	
	112	
	224	
T1 carrier	1.544 Mb/s	(≈24 X 56)
T2 carrier	6.312	(4 X T1)
T3 carrier	44.736	(7 X T2)
T4 carrier	281	(6 X T3)

Figure 4.2 Data rates commonly used in the U.S.

television, are excellent carriers of digital information. In addition, public data networks and computer networks are already in current use.

Service Integration

The idea of ISDN (see Dorros, 1981) is to unify this emerging infrastructure into a public, digital, end-to-end telecommunications network providing a wide range of applications through standardized interfaces. ISDNs are not yet common, nor is it clear what all the services carried will be like. What *is* clear is that even traditional services could be provided more economically and flexibly through a single unified digital medium than through separate and mixed media and networks (such as telephone, cable, and telex).

ISDN planners submit that the new network is relatively independent (say, up to the transport layer) of its future applications. The commodity offered to customers would be units of digital channel capacity. Each customer will "consume" as much capacity as needed. Channel capacity would be delivered through standardized interfaces. Some interface capacities currently planned by the Bell System are:

- 1.544 Mb/s for large business customers
- 144 kb/s for small business customers (sufficient for simultaneous voice and data)
- 80 kb/s for residential customers.

Cable systems (discussed in section 4.3) usually divide their bandwidth into 6-MHz slots; these can be used to carry digital traffic to a number of homes.

The obvious implication for videotex is that ISDN can serve as a convenient delivery system that is tailored to the real needs of the application in question, and not vice versa.

Although truly nationwide ISDN is still perhaps 10 years away, significant local implementations are being realized. One of them is the IDA project of the Manitoba Telephone Company, described in Chapter 6. Another major project is the ISDN service to be introduced in London in 1983. Users will have access to a 64 kb/s voice channel (useable also for data transmission) and an independent 8 kb/s data channel. Both of these will provide highly enhanced telephone and videotex services (see Childs, 1981). Another development is the recently announced Local Area Data Transport (Handler, 1982), which is in essence an X.25-based packet-switching network accessible through the local telephone loop.

4.3 CABLE SYSTEMS

The growing popularity and penetration of television in the 1950's brought about an increasing demand for more channels and better reception. Reception problems were observed in both rural and urban areas: in the former, due to the distance from the transmitters; in the latter, because of multipath reception and because there were many sources of noise. Community antenna television systems (CATV)—or simply cable TV—offered a solution to such problems. Today a considerable fraction of television owners in North America is serviced by cable. In Canada about 60% of all households have the cable installed, and the service is easily available (the cables passing nearby) to another 20%. In the U.S., cable penetration is about 28% and is rapidly growing. In Western Europe cable penetration varies widely between countries. According to Ball (1981) the leader is Belgium, with 75%. West Germany, France, and the U.K. have 35%, 1%, and 14% penetrations, respectively.

It was soon realized that video programming is not the only kind of information that can be carried (and sold) to customers through cable. Data can be superimposed onto the TV signal, as will be described later in this chapter. Additional possibilities are offered for full field data transmission on unused television channels. The next step is to consider the cable as a broadband, ISDN-type transmission medium with certain parts of the frequency spectrum allocated to purely digital (usually packet-based) data services that have nothing to do with the TV signal. Since packets can be addressed, individual sub-channels can be established for communicating with selected users or user groups. Finally, the system can be made interactive by incorporating reverse channels from the users to the cable head; the result is *two-way CATV*. Such systems are in actual operation, offering a variety of informational and other services and approaching what might be called "local" ISDNs.

Physical Structure

Current cable television systems use *coaxial cable* as a transmission medium. The topology is usually a three-level tree structure, as shown in Figure 4.3. The root of the tree is the head-end, or *hub*, which feeds the main trunk with the appropriate signals. This trunk is the backbone of the distribution system. Amplifiers must be inserted every 0.5−1.0 km in order to counter signal attenuation. Some of these amplifiers, called *trunk bridgers*, serve also as branching points for *distribution feeders*; each such feeder can serve 300−500 subscribers. Feeders consist physically of the same kind of coaxial cable

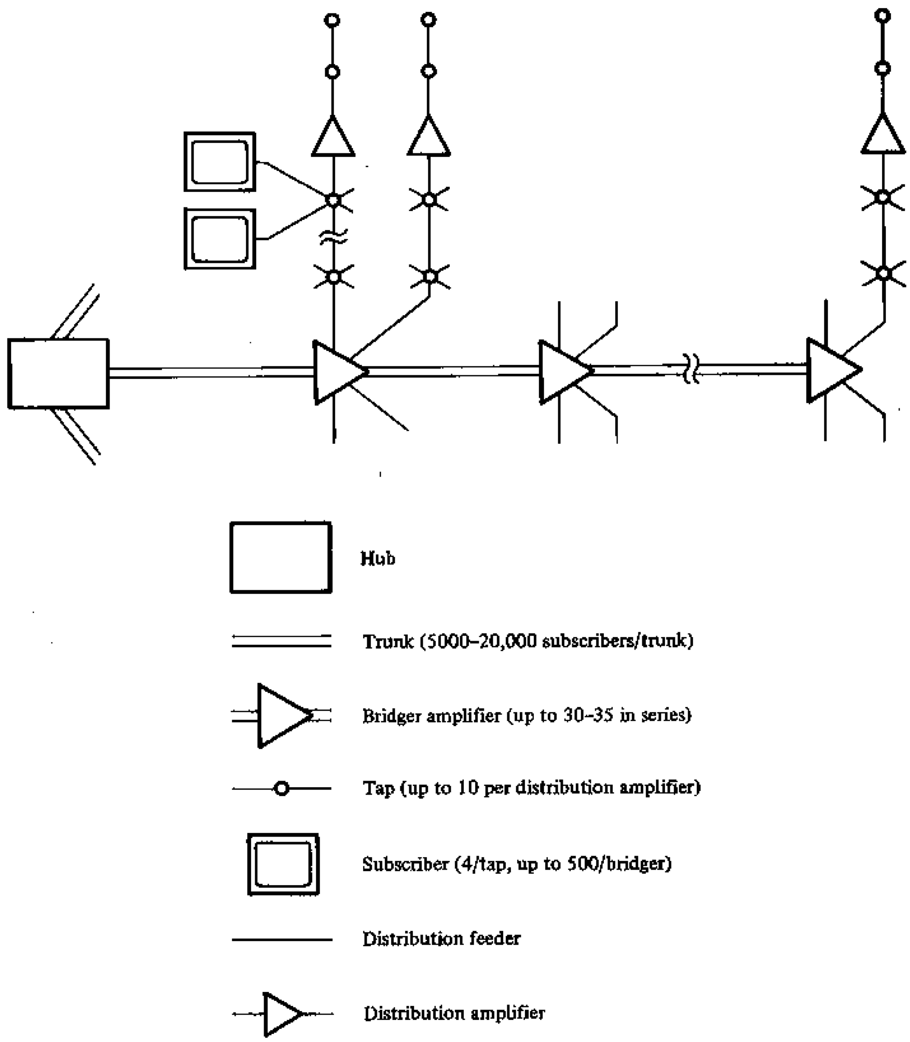


Figure 4.3 Structure of a typical CATV system.

and amplifiers as the main trunk. Individual users are connected to the feeder in branching devices called *directional couplers* or *taps* (4–8 users per coupler).

About 80% of cables installed in Canada are aerial cables; the remaining 20% are installed underground. The cost of installation of one km of cable is between $5000 and $10,000, depending on the factors such as the cable's bandwidth and whether the cable is above ground or buried.

The geographical span and the number of subscribers of such CATV systems is limited by the maximum number of amplifiers that can work in tandem in order to keep signal distortion to a reasonable level; a typical number is 30–35. These limitations can be partly relieved by using multiple hubs; this amounts to a star-like topology.

Two-Way Operation

Recent interest in interactive cable-based services (and in the potential revenues for operators and service providers) has motivated research into methods of upgrading existing CATV systems towards two-way operation. Presently there are about 400 cable operators in Canada. About 10% of the systems are "two-way compatible"—that is, they have physical provisions for the additional equipment required. But there are very few truly operational two-way systems. Examples of such systems are Cablesystems Engineering's system in London, Canada (see Allora-Abbondi, 1979), and Cox's Indax system (see Gates and Tjaden, 1982). Some of these systems will be described later.

Conversion to two-way operation fortunately does not always require replacement of the cable or addition of another one; the same medium can conduct signals both ways. This effect is achieved by splitting the cable's bandwidth and using one part for downstream, and the other for upstream communication. The main physical change that has to be made is to upgrade the amplifiers for bidirectional operation. The new devices consist essentially of two amplifiers with diplex filters, one used for each direction of transmission.

In addition to requiring physical modifications, upgrading to two-way traffic causes a number of further potential problems. The most important one may be the accumulation of noise from upstream traffic; in effect, a single faulty user terminal generating inappropriate signals or frequencies can jam the whole system. One solution is to cut off the noisy branch at the trunk bridger. This can be done by remotely controlling the bridgers by appropriate messages from the head end. Newer two-way designs incorporate a generalized remote control and monitoring facility, applicable to individual user terminals as well. Such a facility is used to implement functions such as statistics-gathering, pay television, burglar and fire alarms, and disconnection of the terminal if a bill is not paid on time.

Splitting the Bandwidth

Modern coaxial cables transmit signals in a frequency range of up to 500 MHz. The useful part of this range is from 5 MHz up. This is an enormous bandwidth, corresponding to more than 100,000 analog voice circuits. In one-way CATV, the bandwidth is allocated to as many as 60 television channels, spaced at 6 MHz intervals and for the most part unused.

In two-way systems the split between the upstream and downstream bands is typically at 50 MHz (but sometimes at 150 MHz). The upper part of the spectrum is used for downstream transmission (thus including the complete VHF television and FM bands). Upstream transmission is between 5 and 35 MHz. Although the above split is not fixed by any formal convention, it is used as a de facto standard by all North American systems. However, except for the allocation of TV channels and FM broadcast bands, there is no further consensus on the subject of subdivision and multiplexing methods for the above two main bands. This absence of standards is similar to the situation in ISDN. A set of conventions to facilitate future interworking of digitized parts of CATV systems was proposed by O'Brien (1981).

Transmission Techniques

Advanced services offered on one-way and two-way CATV rely on digital communication techniques. There follows a brief list of issues that must be addressed in such system design. Since these issues are not videotex-specific, we do not treat them in detail here (see Tanenbaum, 1981).

Addressability. It should be remembered that addressability is applicable even in one-way systems. Addressability permits control information or messages to be sent selectively to individual terminals. A common application is in pay TV, in which microprocessor-controlled terminals receive authorization and supplementary information to descramble selected TV signals. Another form of addressability is applied to cut off noisy branches in bridgers in two-way systems.

Multiplexing. Two basic techniques are used to split and share the bandwidth of a cable: FDM (frequency division multiplexing) and TDM (time division multiplexing). The former is used for analog TV signals, and allots 6-MHZ intervals per channel. TDM is applicable only to digital traffic and is at the base of the packet-switching technology commonly used in data networks. Most systems use elements of both FDM and TDM: for example, assigning a 6-MHz slot for digital traffic and subdividing the slot into a number of sub-channels, each carrying perhaps a 64-kb/s digital bit stream, which itself can be packetized.

Modulation. Analog and digital signals to be transmitted must be modulated onto carrier waves whose frequency is determined by the underlying FDM scheme. The functioning is somewhat similar to that of modems described earlier.

Access conflicts. Since two-way cable is a shared medium, provisions must be made to allocate to each user separate bandwidth (in the form of dedicated frequency or time intervals) in the upstream direction. Alternatively, if a common upstream channel is used, access conflicts must be detectable and resolvable. One commonly used method is the "carrier sense multiple access/collision detection" (CSMA/CD). Its underlying principle is that each transmitter tries to transmit its message in an empty time slot. If a conflict (simultaneous transmission) occurs, the transmitters will repeat their messages after a random delay. Another class of procedures is based on polling. Generally the same methods as those used in local area computer networks are applicable here.

Characteristics of Cable-Based Videotex Delivery Systems

The characteristics of such delivery systems are:

- high bandwidth
- a "box" (cable converter) that resembles a videotex keypad

- in the case of one-way CATV, useability for interactive services, e.g., in combination with telephone
- possible upgrading to two-way of most existing systems (90% of which are one-way)
- limited geographical extent
- no switching capability at present
- possibility of connecting local cable systems into large networks by using satellite links.

4.4 FIBER OPTICS

There is much discussion about the potential of fiber optics as a replacement for coaxial cable or microwave links. We briefly mention this technology because of its potential as a delivery medium for videotex.

Fiber optics are fabricated from highly transparent glass material capable of functioning as optical waveguides. The carrier of information is light that is generated by a light-emitting diode or a diode laser. Analog or digital signals are amplitude-modulated onto the light wave. A photodiode is usually employed at the receiving end to convert the incoming light intensity into an electrical signal. Fibers are packaged into cables, each containing a variable number (1–24) of fibers.

At present it is not clear whether—and when—fiber optics will become a viable alternative to cable (see Metz, 1981). The situation is analogous to that involving the replacement of mechanical disks in computers by new electronic memories, such as magnetic bubbles or charge-coupled devices. Disks represent a well-entrenched technology, and experience shows that in order to be replaced by newer, exploratory technologies, these technologies have to be significantly superior in terms of both cost and performance. Marginal disadvantages are not likely to win out against inertia resulting from investment in well-proven technologies. So far, fiber optics technology has been unable to demonstrate an overall superiority over cables. Therefore, it is no surprise that cable companies are hesitating to move ahead with large-scale replacement. A more detailed analysis would involve the following facts:

1. The bandwidth of the fiber optics medium itself (like that of the coaxial cable) is much higher than that of the coupling equipment (amplifiers, modulators, taps).

2. A single fiber today can carry five to six TV channels (as opposed to 54 in modern cables). However, the distance required between amplifiers is much larger with fibers (10-20 km) than with cable (0.5–1 km). Digital transmission rates of more than 1 Gb/s (over a single fiber) have been demonstrated in field test operations in Japan and Germany.

3. The major reason for the limited bandwidth of entire analog transmission systems is the nonlinearity of the light source. Thus, present technology is better suited for digital transmission (in which signal distortion is not critical).

4. The cost of production of fibers is decreasing, whereas cables require materials such as copper and aluminum, whose cost cannot but increase in the future.

5. Current cost-effective uses of fiber optics in the U.S. include medium-distance links having few or no repeaters and no taps (branching). A typical example is the link between a satellite receiver station and the cable head-end. At the present stage, application for cable trunks and distribution feeders would be uneconomical due to the high cost of branching contacters (taps). One foreseeable application is in "mini-ISDN" systems (see Mini-Hub, 1981) for the distribution of video and digital services in high-rise apartment buildings (in which each customer would have a direct connection with a local head-end in the building).

Fiber optic links are used or planned in several countries (U.S., Japan, France) as high-speed trunk lines in telephone networks. They seem to be gaining ground mainly in short- and medium-distance applications (e.g., within a large city) not requiring repeaters.

Fiber optics cables are being used in project ELIE. This is an ISDN-type system being tried in Manitoba, Canada (see Chang, 1982).

4.5 DATA TRANSMISSION OVER THE TV VIDEO SIGNAL (DOV)

The idea of piggybacking digital information on (analog) television and even FM radio signals is not new. A number of schemes have been devised to take advantage of unused portions of the frequency spectrum of the video signal. One proposal used a "flashing dot" displayed in the corner of the TV screen. These light signals were detected by a photodiode and used to drive a teletype.

DOV techniques in use today are based on different principles. Transmission takes place during the *unused time intervals* of the TV signal, and the received information is usually displayed in readable or graphical form on the screen. The unused times occur normally during the *vertical blanking interval* (VBI) between successive fields of picture information; an extension of the method, called *full-field* transmission, transmits only data (no picture information), while keeping the infrastructure of synchronizing (sync) pulses. The VBI technology originates in attempts to provide captioning and subtitling of TV programs. Since DOV is an important videotex-specific delivery method useable in both over-the-air and cable transmission, we will discuss it in considerable detail.

Principles of TV Transmission

There are three main standards used for color TV transmission: NTSC, PAL, and SECAM (Figure 4.4). For the following explanation, neither specification of the particular standard nor the presence of colors is essential. The TV picture is drawn on the inner face of the picture tube by an electron beam that sweeps across the picture area in a sequence of horizontal scan lines controlled by horizontal sync pulses. The intensity of the beam as it hits a given point determines the luminosity of that point. This intensity information is basically what is transmitted in analog form by a (black and white) TV signal. When the last scan line is completed, the beam returns to the top of the screen

System	Lines/frame	Frames/s	Video band width MHz	Countries	Bit rates used Mb/s
NTSC	525	30	4.2	USA Canada Japan	5.73 (BS-14) (and many others)
PAL	625	25	5.5*	Western Europe except France	6.9 (U.K.)
SECAM	625	25	6*	Eastern Europe and France	6.2 (Didon)

*Can be 5, 5.5, or 6 MHz

Figure 4.4 Comparison of TV standards and teletext bit rates.

and scanning starts anew. If this process is repeated quickly enough, the viewer has the impression of seeing the whole image at once; the effect of movement is achieved by slight changes in subsequent images, as in the projection of films.

In order to obtain better picture quality a full TV image (frame) is composed of a sequence of two interlaced *fields*. This allows a doubling of the rate at which images are presented, and thereby significantly reduces objectionable flicker. If we think of the lines of a frame as consecutively numbered, then the two fields will contain even-numbered and odd-numbered lines, respectively. However, the lines in each field are usually numbered separately; we adopt this convention here as well. Figure 4.5 illustrates the composition of a field.

The exact timing of the movement of the beam across the screen (that is, the times of horizontal and vertical retraces) is controlled by synchronizing pulses. While the beam is returning to start a new line or field, its intensity must be sent to the blanking (black, invisible) level. Hence the term VBI. Note that although horizontal retraces are also blanked, they are too short to carry very much data.

Data over Video

Strictly speaking, transmission of picture information is suppressed during the vertical retrace interval and also during the interval in which the first few scan lines appear in each field. These scan lines are normally not displayed on the visible area of the screen, but they can be seen as a black bar between two frames when the receiver is out of vertical synchronization. Data bits are modulated onto some of these invisible lines. The term VBI denotes collectively all lines that do not carry picture information.

Two remarks are appropriate at this point. First, the sequence of horizontal sync pulses is not interrupted through the vertical beam retrace. The first few lines of the VBI are used for the vertical sync pulse (which in reality consists of a sequence of pulses); only the subsequent lines are available for data. Figure 4.6 indicates the timing of a typical field. Second, the above-mentioned blank lines between fields are not to be confused with the so-called *overscan* lines, which do carry picture information, but are usually

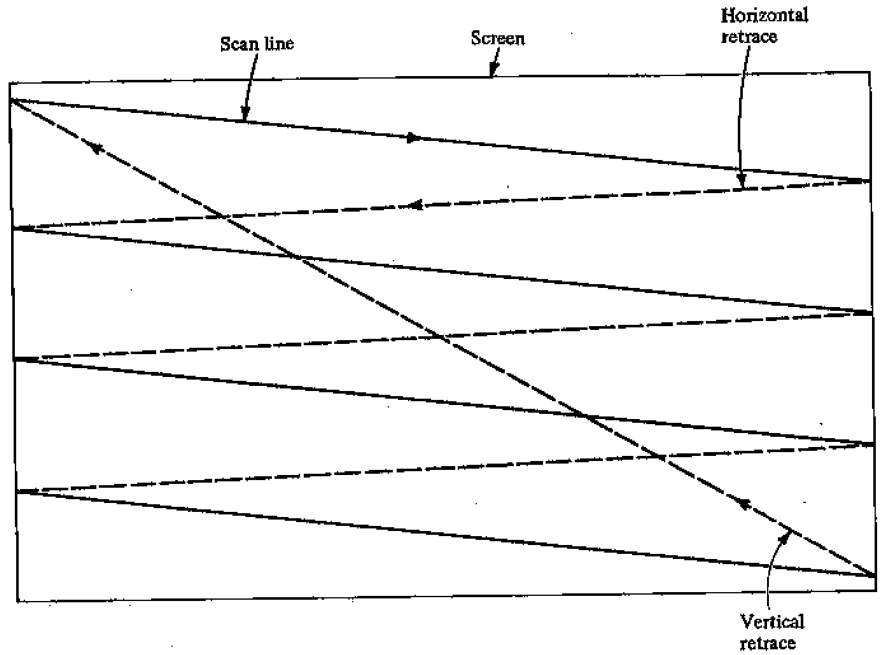


Figure 4.5 Composition of a TV field. A frame consists of two interlaced fields.

also invisible. The purpose of these overscan lines is to assure full coverage of the screen by the image in spite of variations and aging of the TV sets.

A somewhat simplified form of a TV signal between two horizontal sync pulses is illustrated in Figure 4.7 (note that a line carrying data is called a *data line*). In most systems a straightforward coding is used, with two signal levels corresponding to 0 and 1, respectively.

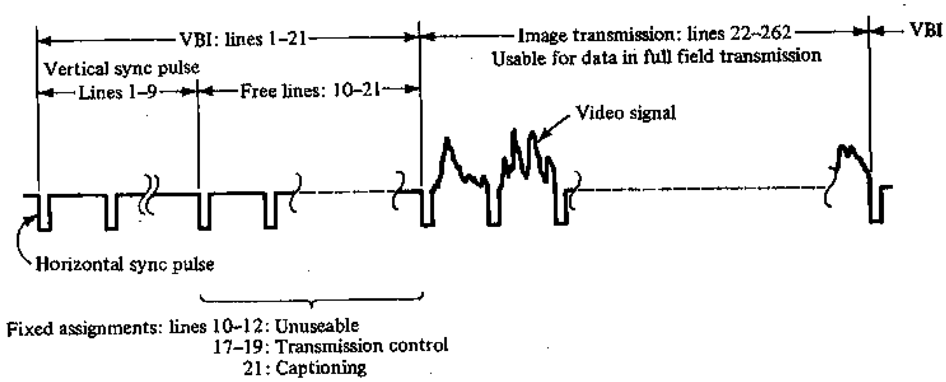


Figure 4.6 Simplified timing diagram of an NTSC field. VBI lines 13–16 and 20 are potentially available for data transmission (data lines).

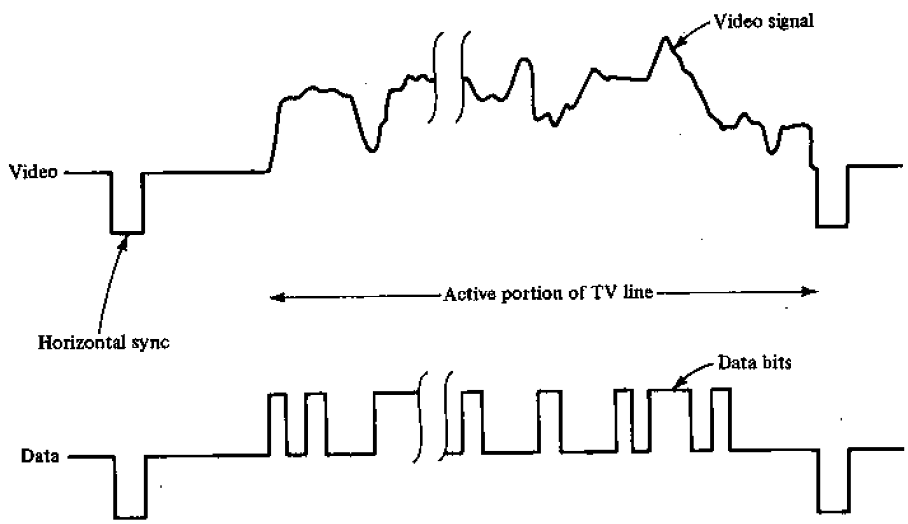


Figure 4.7 Timing of video lines and data lines.

In full-field transmission all available lines are data lines and no picture whatsoever is transmitted. This approach is particularly feasible in cable-based systems that have large channel capacity that is typically only partly used. These conditions make it possible to dedicate one or more entire channels to data transmission. In countries where TV is broadcast mostly over the air, the VBI technique is preferable since it permits simultaneous transmission of picture and data. Of course, if everything else is kept constant, the resulting effective data rate is proportional to the number of data lines in the signal.

Data-Line Assignment

The number and assignment of data lines in the VBI is subject to informal conventions and varies in different countries and different systems. There is no internationally adopted standard in this matter. This does not represent a problem for decoder design because the proper data lines are usually recognized by examining the waveforms and not by counting the scan lines.

Some lines of the VBI are occupied for purposes other than data transmission, such as controlling the quality of the TV signal (compensation for color variations), channel and program source identification, and transmission of test patterns (gray scale and color values). The remaining data lines can be used for program-related data (such as subtitling and captioning), teletext data (to be displayed on the screen), and independent applications, such as broadcast facsimile, not tied to the program or screen.

In NTSC, the VBI consists of the first 21 lines of each field (25 lines in PAL). According to current Canadian regulations, lines 15, 16, 20, and 21 are available for data; line 21 is reserved for closed captioning. The U.K. teletext originally used lines 17 and 18 (this was later extended to four lines), and, in a CBS trial in 1979 involving the U.K. and Antiope technologies, lines 13 and 16 were used.

In certain television receivers, electron beam blanking depends on the TV signal's being at black level. If data were inserted in these lines, dotted diagonal lines would faintly appear across the screen. More modern receivers electronically blank the electron beam during VBI time. As older receivers are retired from service, more VBI lines will become available for teletext.

Transmission Rates

The *instantaneous bit rate*, B, is the speed (in Mb/s) at which bits are transmitted during the periods when data are sent. It is determined by a number of system parameters:

1. The video bandwidth W (MHz) of the transmission medium (including the receiving equipment). If two signal levels are used for coding, the maximum obtainable rate is $B_{max} = 2W$. For North America, this yields $B_{max} = 2 \times 4$ MHz $= 8$ MHz. (Note that without the restriction to two signal levels, a channel can carry an arbitrarily high bit rate as long as the signal-to-noise ratio is high enough.)
2. Acceptable error rates. The observed bit error rates in a given system increase as the transmission rate approaches the above limit. Although various theoretical models exist for this dependence, they do not entirely cover practical situations, and error statistics must be determined empirically, usually in field trials. These tests can also cover the influence of the technology of the receiving equipment.
3. Error detection and correction methods used (see below).
4. Evaluation of the psychological effect of various error types (such as loss of character, displaced lines, mixup in page numbers) on the user.
5. The value of B should be a multiple of the TV line frequency.

Within the above constraints, the bit rate should obviously be maximized. Practically used values of B are between $0.8\,W$ and $1.4\,W$. The U.K. teletext rate is 6.9 Mb/s (too large for NTSC). In North America numerous tests were and are being conducted at various rates between 3 and 6 MHz. The bit rate recommended in the France-Canada-CBS joint standard proposal (discussed in the next chapter) is 5.72 Mb/s.

The *effective bit rate*, F, is the average number of bits transmitted over an extended period of time. F is given by

$$F = B \times (\text{fraction of the data lines in the TV signal}) \times (\text{portion of the TV line carrying data bits}).$$

For example in the U.K. teletext,

$$F = 6.9 \times (2/312.5) \times 0.81 = 35.7\,\text{kb/s}.$$

Further parameters related to the effective bit rate are the effective character rate ($F/8$) and the useful character rate, U. The latter is the average rate of useful characters, which are those characters remaining after overhead information in a data line (such as addressing and error control) is discounted. (Evidently, the designation "useful" is subject to interpretation: is a display attribute character overhead or useful?) From U, the line and page rates are easily obtained. In U.K. teletext (40 characters per display row, 24 rows per page, 40 useful characters out of 45 per data line, 2 data lines per field), the page rate works out to 4.13 pages/s. Thus the average waiting time in a 160-page teletext cycle is about 20 seconds.

Transmission rates for full field operation are calculated similarly. Taking 300, instead of 2, lines in each field will give some 600 pages/s. Rates attainable in NTSC are slightly reduced due to the smaller bandwidth of NTSC.

The following rule of thumb can be used both in Europe and U.S.: for each scan line devoted to teletext in a TV frame, the transmission rate is one page per second.

Error Handling in DOV

A number of DOV system characteristics enter into consideration in the design of suitable error strategies. First, the basic system is one-way. Therefore the forward data flow cannot be interrupted for retransmission of erroneous packets of information. Although much (but not all) information is cyclically retransmitted, the cycle time is too long to be relied upon systematically for error correction. As a result, forward error correction techniques are used.

Further considerations touch upon error probabilities and statistics, which are investigated in numerous field tests and other experiments. The principal parameters having influence on the occurrence of errors are the transmitter power, conditions of TV reception (especially in urban areas, in which waves reflected from buildings, hills, etc., cause the received signal to be the vector sum of a multiplicity of signals arriving over many paths), bit transmission rates, and the implementation of the receiving equipment. One important finding from a European field test (see Blineau, 1980) is that most errors are single bit errors and not bursts.

There is a fundamental difference between European television reception and North American television reception. In most European countries there are but two or three programs transmitted over hundreds of repeaters. Nearly everyone can receive at least one good signal for each program. In addition, television receivers are substantially more expensive. Thus, the proportionate cost of a quality antenna is less. Furthermore, most European stations are UHF. Since UHF wavelengths are shorter than VHF wavelengths, UHF antennas are less expensive, have higher gain, and are more effective in discriminating against ghosts (faint shifted images appearing on the screen due to multipath propagation). In the U.S., the situation is opposite in almost every respect. There are over 1000 stations in the U.S., and almost no repeaters; nearly everyone is certain to receive at least one poor signal. Most transmitters are VHF, requiring larger, more expensive, less directional, and lower-gain antennas. Because color receivers are so inexpensive, a quality antenna installation may exceed the cost of the receiver. Thus, Americans are accustomed to purchasing a color receiver with "rabbit ears" and expecting perfect reception.

Furthermore, U.S. rules on ignition and electrical motor brush noise are much less stringent than are European rules. One need only have a teenage daughter with a hair drier to know that errors caused by motor noise destroy more than one bit. Watching the television screen with this kind of interference yields broad bands of bursty noise.

A particularly troublesome case is that of close-in ghosts. These are caused by reflections from structures very close to the transmitting antenna. The result is not a separate ghost image, but rather a blurring or smearing of the image. This defocusing of the picture is undesirable but by no means catastrophic. The picture is usually perfectly watchable. However, the consequences for teletext data could be disastrous. This phenomenon of close-in ghosts is a serious but mostly unappreciated threat to the fortunes of teletext in the U.S. The threat is so severe that the EIA Teletext Committee formed a task force to study possible solutions to the problem.

Different types of errors and their consequences are more or less tolerable by the user. Each coding system has specific points of vulnerability that have to be carefully protected against error. More sophisticated coding schemes, such as Telidon, are generally more vulnerable than are simpler ones; as a consequence, all data sent in broadcast Telidon are covered by an error-correcting code, whereas only the address part of the U.K. teletext is protected by such a code.

Technical Criteria Used to Assess Reception Quality

In order to assess the performance of a given teletext system installation, some easily testable criteria must be established. One such criterion is the bit error probability and related measures of burstiness. However, such error statistics do not give a direct indication of the subjective quality of reception. A given error level may yield quite different results with different coding schemes and error strategies. Specialized equipment is also required for such measurements.

Croll (1977) describes a set of simple criteria suitable for judging teletext reception (and encompassing the entire system from the transmitter to the receiver's screen). They are (in order of decreasing stringency):

- no errors received in 10 seconds of transmission
- no visible errors in each of three consecutive new acquisitions of one page
- no errors remaining after two writes of one page (each character of a page is error-free in at least one of two consecutive receptions)
- about half of the characters correct for each new acquisition of one page.

It is suggested by Croll that complying with the second or third criterion should yield acceptable reception quality.

Eye height. Another method commonly used to judge the quality of received teletext signals is the so-called *eye height* test. As will be discussed in Chapter 9, the waveform obtained after demodulation of data line is more or less deteriorated, as shown in Figure 4.8(a). This deterioration is due to factors such as intersymbol interference and noise. Intersymbol interference is the smearing of one logic signal into the time space occupied by another. Principal causes of such interference are bandwidth limitations and multipath propagation. An important parameter that has impact on the occurrence of decoding errors is the difference, h, between the lowest received value for a 1 and the highest received value for a 0 (over some period of time).

The value of h (as a percentage of the distance between the 0 and 1 reference values) can be measured directly by displaying the received signal. That is, waveforms in individual bit periods are displayed superimposed on a screen, creating the typical

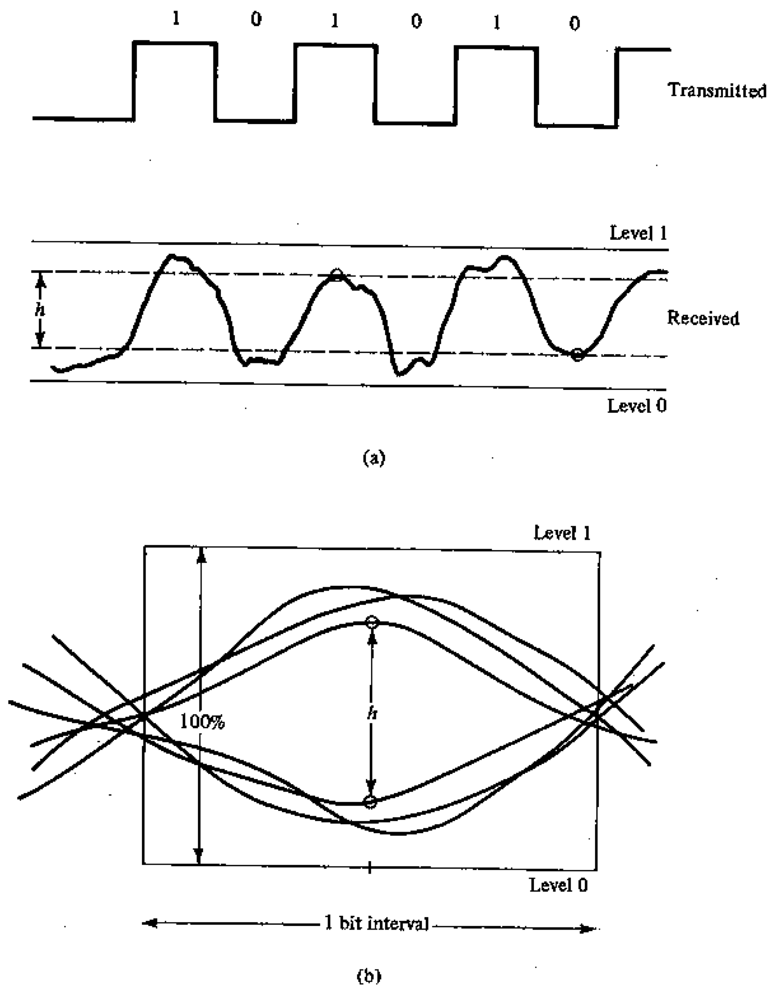


Figure 4.8 (a) Deterioration of received signal; (b) display of eye and determination of eye height h.

eye-shaped pattern shown in Figure 4.8(b). The value of h is read in the middle of bit intervals. Under normal receiving conditions the eye height is about 60–80%, but most receivers are capable of handling values as low as 25% without too many errors. Note that other attributes of the eye, such as eye width and shape, are also significant (for more details, see Lucky and Saltz, 1968).

Error Detection and Correction

Current DOV systems use combinations of various error correction and detection techniques. See Peterson and Weldon (1972) for a comprehensive treatment of error-correcting codes. Vertical parity checks (one parity bit per character) permit the *detection* of an odd number of errors in each individual character. A combination of vertical parity and horizontal (longitudinal) parity (or checksum) enables the *correction* of any single error in a data block. The commonly used (8,4) Hamming encoding provides for the automatic correction of any single error and for detection of all multiple-bit errors in groups of four bits. Hamming codes have high redundancy and they are therefore used mostly on sensitive information, such as addressing and packet sequencing.

Figure 4.9 shows a block of 7-bit characters, each accompanied by a vertical parity bit. Appended to the end of the block is a longitudinal parity character, C. Parity bit p_i is generated by the transmitter in such a way that the number of 1's in the transmitted group of eight bits is, say, odd (even parity systems are also used). For example, if the character to be transmitted is 0110001, p_i will be 0. At the receiving end, the parity is recalculated from the received character bits and checked against the received p_i; if the two are different, the character is considered to be erroneous (although it may be that only p_i is in error). If the two bits are equal, the character is considered to be correct (although it may contain an even number of errors that cannot be detected by simple parity checks). However, two (four, six, etc.) errors in a character are generally much less likely than is a single error; this is why parity checks work well most of the time.

Bits in the parity character are generated in a similar way as above, one for each line in the data block shown in Figure 4.9. If a single error occurs in the data block, it can not only be detected, but also corrected. This correction is done by inverting the bit at the intersection of the column (character) and line in which the error is detected. A potential problem with longitudinal parity, especially in larger data blocks (e.g., of 1024 characters) is the increased probability of multiple errors in a line. Solutions include inserting parity words at shorter intervals (e.g., after every 32 characters) or using multiple check bits, called collectively *checksums*, at the end of the block. As a direct extension of the parity method, n check bits for a line might represent the sum modulo 2^n of a line ($n = 1$ for even parity).

Hamming codes. Another form of multiple parity checks used for error correction is found in Hamming codes. In the variant commonly used to protect control infor-

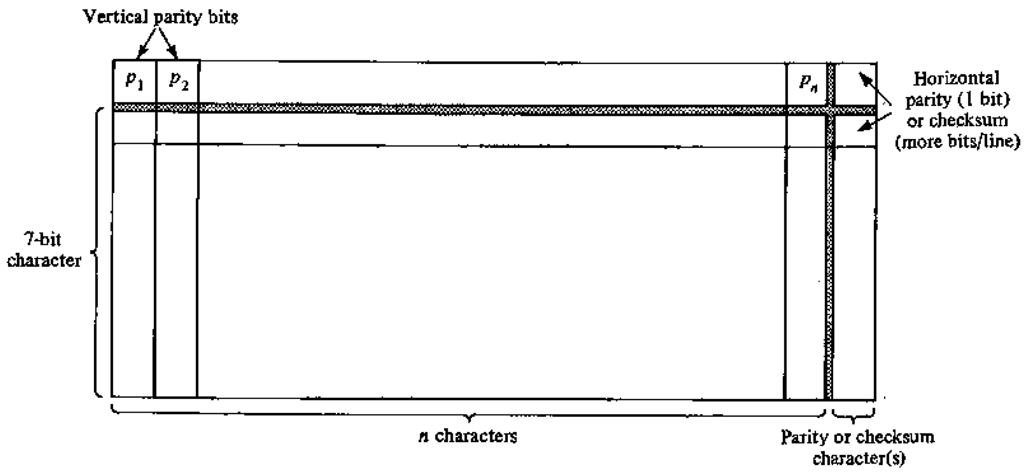


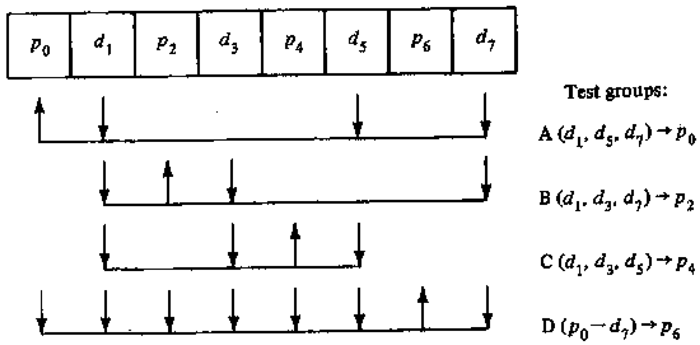
Figure 4.9 Horizontal and vertical parity and checksums.

mation in DOV systems, and called the (8,4) Hamming code, four check bits are used to protect four data bits in each byte. The parity checks apply to different groups of data bits in such a way that each possible outcome of the parity checks singles out a unique data bit which must be in error (assuming that only one error occurred in the transmission of the whole byte). Figure 4.10(a) shows a Hamming-encoded byte; d and p are data and parity bits respectively. Groups of data bits used to generate parity bits are also shown. The same groups, including the parity bit, are, at reception, subject to separate parity tests A, B, C, and D. Note that p_6 is the ordinary parity involving all bits; it is used to discriminate the case in which only one error is present. One can see, for example, that bit d_5 contributes to the formation of parities p_0, p_4, and p_6; therefore, if d_5 is received with error, parity checks A, C, and D will fail. Figure 4.10(b) summarizes all possible cases and the corrective actions to be taken (for example, inversion of the data bit in question).

General Data Format in DOV

Figure 4.11 illustrates the general format of a data line transmitted in DOV. In the terminology of the Open Systems Interconnection model, the data line is composed of two parts: Delivery Protocol Control Information (DPCI) and Delivery Service Data Unit (DSDU). Also shown in the figure are some related terms used in teletext jargon.

DSDU signifies information that is transferred to the higher (presentation and application) level protocols. DPCI groups all control information needed for the correct functioning of the delivery protocol. The bit and character synchronization sequences



(a)

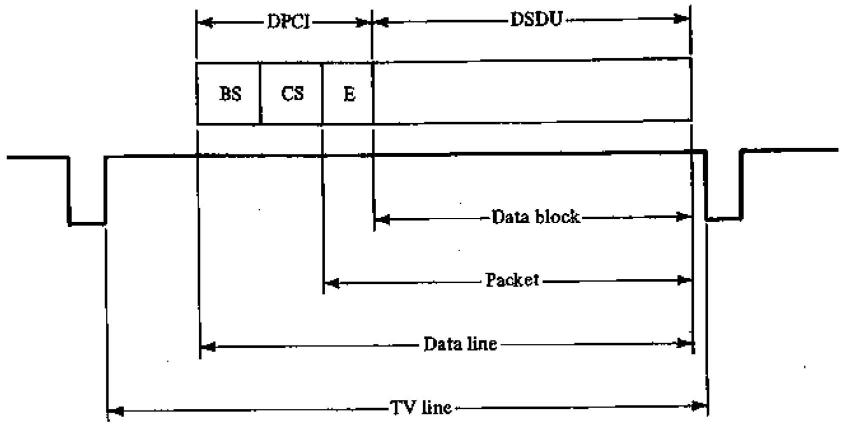
A	B	C	D	Result
0	0	0	0	No error
1	0	0	0	More than one error, reject
0	1	0	0	More than one error, reject
1	1	0	0	More than one error, reject
0	0	1	0	More than one error, reject
1	0	1	0	More than one error, reject
0	1	1	0	More than one error, reject
1	1	1	0	More than one error, reject
0	0	0	1	data bits OK
1	0	0	1	data bits OK
0	1	0	1	data bits OK
1	1	0	1	d_7 in error
0	0	1	1	data bits OK
1	0	1	1	d_5 in error
0	1	1	1	d_3 in error
1	1	1	1	d_1 in error

0: Parity check OK
1: Parity check fails

(b)

Figure 4.10 (a) Generation of parity bits p_0, p_2, p_4, p_6 from data bits d_1, d_3, d_5, d_7 in (8,4) Hamming code; (b) error correction in (8,4) Hamming code.

invariably occur at the beginning of the data line; they usually consist of fixed bit patterns serving to establish bit and character synchronism. *E* denotes collectively all error detection and correction bits that actually may be spread out over the whole line. The other elements of DPCI may change from one system to another; they will be illustrated by the examples in the next chapter.



BS: Bit synchronization
CS: Character synchronization
E: Error correction bits
DPCI: Delivery Protocol Control Information
DSDU: Delivery Service Data Unit

Figure 4.11 General format of DOV data line.

5 Teletext

Teletext (or broadcast, one-way videotex) is commonly defined as a "digital data broadcasting service that is primarily intended to display text or pictorial material reconstructed from coded data on the screens of suitably equipped TV sets" (Money, 1979). The underlying assumption is that coded data are received via DOV (data over video) techniques—which is true in almost all operational systems. Moreover, there is a sizeable infrastructure of technology (decoder chip sets) and standards supporting DOV. On the other hand, certain digital services to be delivered over ISDN imply purely digital data broadcast—that is, without the supporting synchronization structure of the TV signal. There is no reason why such services should not also be called teletext.

5.1 HISTORY: SYNCHRONOUS VERSUS ASYNCHRONOUS SYSTEMS

The first versions of VBI text transmission were developed around 1972–1973, in Britain, as an outgrowth of earlier efforts in captioning TV programs. Initially there were two competing VBI systems, called Oracle and Ceefax, operated by IBA and BBC, respectively. In 1974 the two systems were reconciled in the form of a teletext standard, which was further updated to its present form in 1976. Public teletext service began in the U.K. in 1974. In November 1982 there were more than 550,000 decoders in use in the U.K. (*International Videotex Teletex News*, Nov. 1982). Most color receivers now sold or leased in the U.K. incorporate teletext.

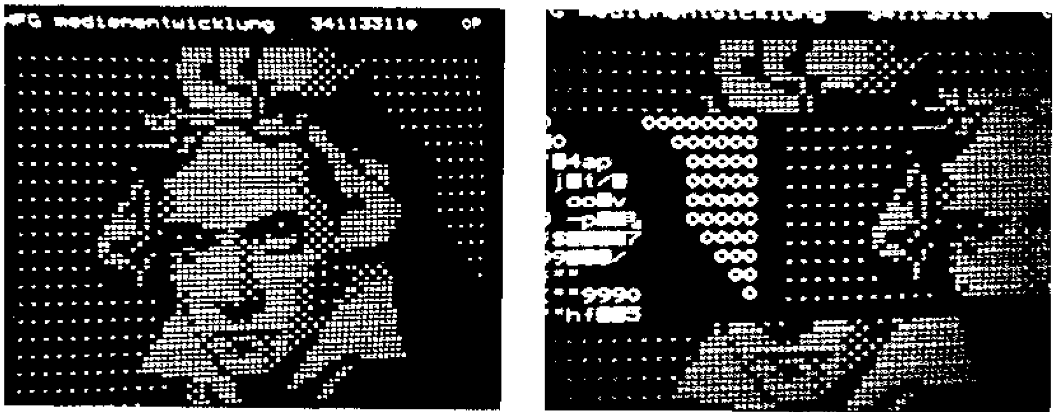
Synchronous Transmission

The hallmark of the British system is its *synchronous* method of transmission (also called *fixed* or *defined* format). As discussed in more detail in Chapters 7 and 9, the essence of this method is the doubly defined relationship between the position of a character on a TV-line (with respect to the line sync pulses), its place in the display memory, and its position in the row being displayed on the screen. Consequently:

1. One TV data line always carries the contents of one display row.
2. Control characters to set display attributes, such as color, are treated "serially"—that is, they are stored in the display memory and they occupy the place of a displayable character.
3. Decoders for synchronous transmission are relatively simple and cheap.
4. It is claimed, but not definitely demonstrated, that synchronous transmission has better error resilience than does asynchronous transmission. The argument is that the line sync pulses act as a barrier through which the effect of most errors cannot propagate. This is demonstrated by noting that a television picture is locked in position on the television screen even when it is so snowy as to be unwatchable. The U.K.'s defined format system attempts to take advantage of this fact by locking teletext to the rigid frame provided by the sync pulses. The typical error affects only one character position and cannot shift subsequent lines horizontally. On the other hand, certain types of multipath, in conjunction with fixed format transmission, may result in repetitive error patterns that synchronous broadcast cannot get rid of.

Since all characters in defined format systems occupy a position on the screen, there is no hazard that noise will convert a displayed character into a nondisplayed character, or vice versa. If this conversion takes place in the asynchronous approach, a character is either inserted or deleted from the displayed row, displacing the remainder of the row to the right or left. Upon next correct reception of that row, the remainder of the row pops into position, thereby calling attention to the problem. The defined format approach avoids the possibility of this type of error.

Figures 5.1(a) and 5.1(b) demonstrate a hazard that exists in viewdata (which is essentially an asynchronous transmission system). The first photograph is of a viewdata frame being correctly transmitted while transmission of the following frame was being interrupted by one noise burst. Asynchronous transmission uses Carriage Return/Line Feed (CR/LF) codes at the ends of lines containing less than forty characters. To avoid penalizing lines that have forty characters, the decoder counts forty characters and locally performs the CR/LF function. This works very well and increases transmission efficiency in a noise-free environment. However, a noise burst can introduce extraneous results that propagate until the next time CR/LF is employed. The defined-format



(a) (b)

Figure 5.1 (a), (b) Effect of errors in asynchronous transmission.

syncronous transmission scheme avoids this by using the television horizontal synchronization pulses as automatic positioning devices for the image on the screen. Errors are constrained to fixed boundaries and are thus not allowed to propagate.

As already mentioned, the U.K. teletext works with a fixed bit rate of 6.9 Mb/s. Here lies the main disadvantage of the system—namely, that it cannot be directly applied with the NTSC standard and in other situations requiring reduced data rates. (Although, as pointed out previously, the theoretical value for B_{max} in NTSC is 8 Mb/s, in practice even 6.9 Mb/s is too high, and lower rates, such as 5.72 Mb/s, must be used.) Any change of the data rate would result in a modification of display (and display memory) formats. The U.K. system can be used in North America in either of two ways:

1. By reducing the number of characters from 40 to, say, 32 per row
2. By "gearing"—that is, transmitting only a (fixed) portion of a row on each TV line (see below).

Asynchronous Transmission: The Didon Concept

Whereas in the U.K. teletext has been an outgrowth of TV captioning, the evolution in France has been characterized by a wider approach. The initial idea was to establish a data broadcast network analogous to the Transpac packet switching network. The design goals as described by Blineau (1980) were:

- to establish a packet network
- to provide a large number of independent channels
- to have the channels transparent (i.e., capable of transmitting all characters)
- to be able to change dynamically the capacity of the channels according to need
- to use an existing transmission medium.

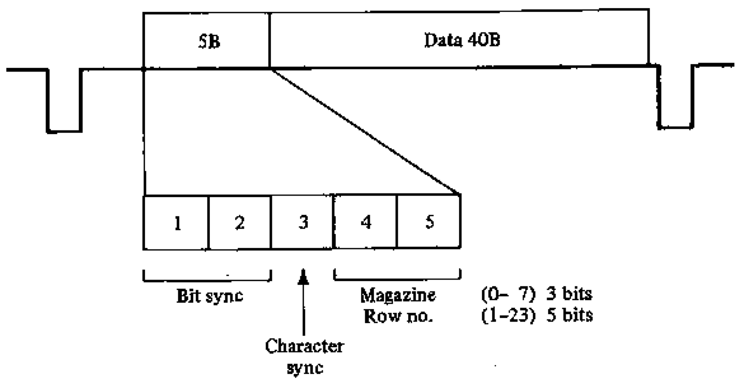
Due to its capability to transmit data and video simultaneously, and its potential for using the same methodology in cable and satellite transmission, television broadcast was the obvious choice satisfying this last goal. The network that resulted is named Didon, and teletext (there called Antiope) is only one—and the first—of its applications. Other applications will come gradually with the evolution of terminal technology (printers, facsimile machines, and secondary memories to store large amounts of data off-line). An interesting possibility is to use a Didon channel for digitized sound signals to accompany teletext pages.

In Didon-Antiope, TV data lines and the display format are uncoupled, hence the term *asynchronous*. Every TV line carries a *data packet*. Page and line delimiters and any other interpretation-oriented information are coded explicitly at arbitrary places in the sequence of packets. The advantages of asynchronous transmission stem from its easy adaptability to different transmission rates and TV standards. Its main disadvantages are increased overhead for format control characters and more complex decoder design. This is partly due to the "parallel" handling of display attributes which is common in asynchronous systems and which amounts to keeping displayable and control characters in separate memories. While they add some flexibility to display coding, it is important to realize that parallel attributes are not a logical necessity inherent in variable-format transmission!

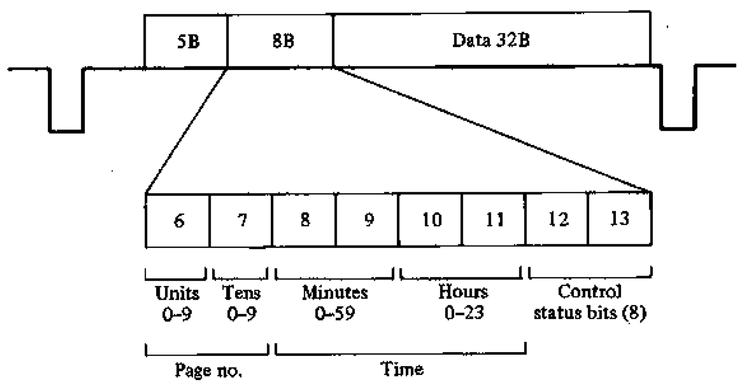
There is an unfortunate rivalry between the proponents of the two teletext transmission methods, their concomittant presentation systems (Prestel and Antiope), and their derivatives (the enhanced U.K. teletext proposal and the France-Canada-CBS standard submission). This split is also very visible and disturbing in terms of the evolution of presentation level standards, many of which must accommodate the two quite contradictory approaches.

5.2 U.K. TELETEXT

The display format presently used in the U.K. (see Money, 1979) contains 24 rows of 40 characters each. More precisely, the first row (the "header," or row 0) carries only 32 characters because the corresponding data line contains eight bytes of supplementary control information. The two line formats are illustrated in Figures 5.2(a) and 5.2(b). All bytes except for bit and byte synchronization and data are (8,4) Hamming-encoded, resulting in four information bits per byte. Actual page numbers selected by the user range between 0 and 799, and are a concatenation of the magazine number (0–7, but usually referred to as 1–8) and the page number (0–99). One might ask why the two are treated differently (i.e., why the magazine number appears in each row, whereas page numbers appear only in the header). The reason is that although the rows of a page are normally transmitted in sequence, in some situations it is desirable to *interleave* rows from several pages. More specifically, pages having the same number (modulo 100) but different magazine numbers can be interleaved (e.g., 25, 125, ..., 725). This can decrease the apparent access time to a page and increase the time interval between the reception of two successive rows, which is particularly important in full-field transmission (in which a sustained high data rate would tend to strain the hardware capabilities of



(a)



(b)

Figure 5.2 U.K. teletext data lines: (a) rows 1–23; (b) row 0.

decoders). On a different level, magazine numbers are used to distinguish between various sources of data (service providers, transmitters) and/or various types of information content.

The time code is used for various purposes, such as for display on the screen, for access of "time coded" pages (transmitted at precise times), or simply as an extension of page numbers (in which case it loses its significance as an expression of time).

The header contains 11 control bits (eight in bytes 12 and 13, and three in spare bits in the time code). The bits are used to control the following functions:

- subtitle
- newsflash (in which the page is displayed in "boxed" mode—that is, is inserted in a window into the current TV program)

- page erase (to clear the screen)
- suppression of header row display (in situations in which such display might be disturbing, such as in graphic pages)
- update (to remove a newsflash)
- interruption of page sequence (to signal to the decoder that pages are transmitted in other than sequential order)
- inhibition of display
- row interleave mode.

Enhanced U.K. Teletext

Because of its fixed format, the teletext system currently used in the U.K. is unsuitable for NTSC. To overcome this and other shortcomings (e.g., limitations stemming from serial attribute coding), engineers in the British Telecom have designed an enhanced teletext system. A detailed description of it is left for Chapter 8 because the system involves a number of presentation-level issues that we have not yet discussed.

The essential features of this system can be summarized as follows (see CCIR, 1981; Chambers, 1980):

1. It consists of five layers of coding, each including additional features. The first level corresponds to the basic teletext service in present use.

2. The layers (unrelated to OSI layers) were conceived with an eye to hardware implementation. A family of compatible decoders is envisaged; lower-level decoders would be able to interpret higher-level features by applying the best possible approximation (e.g., substitution in first-level decoders of nonaccented characters for accented ones, which are present only from the second layer up).

3. Provision is made for gearing. This would serve as a way to achieve 40 characters/row in NTSC while keeping the fixed transmission format.

Gearing

Gearing is a technique that permits fixed-format teletext transmission in NTSC, while keeping 40 characters per row. As already mentioned, the problem is the reduced bandwidth of NTSC that is practically insufficient for the 6.9 Mb/s rate used in the U.K. Therefore, it would seem that either the fixed format, or the 40 characters/row requirement must be abandoned in NTSC-based teletext. The solution lies in a slight redefinition of what is called "fixed format": in gearing, fixed format does not mean one display row, but rather a fixed part of a display row, per data line. The principle is illustrated in Figure 5.3. In this example, four rows are transmitted in five data lines in such a way that the first four lines, L1–L4, carry the first 32 characters of each row, and line L5 carries the remaining $4 \times 8 = 32$ characters; then the process is repeated for the following four rows. Alternatively, the 32 character positions of all 20 rows can be

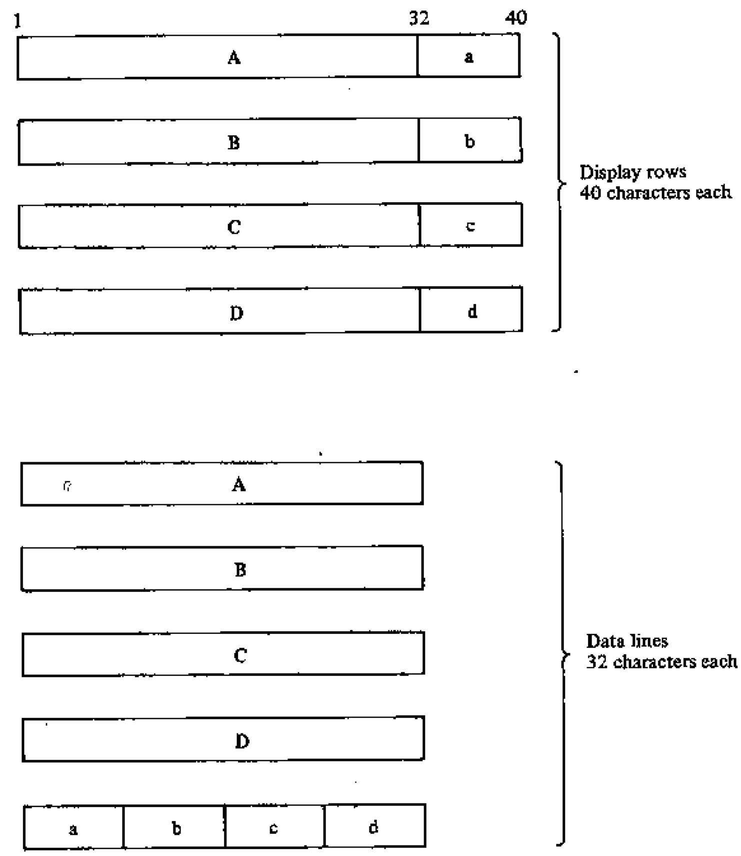


Figure 5.3 Principle of gearing.

transmitted first, and then the remainders. In any case, this amounts to a reduction of the data rate by a factor of $(a + 32)/(a + 40)$, where a is the number of overhead characters (addressing) in each line (see Crowther and Hobbs, 1979, for detailed calculations). Other gearing ratios, e.g., 30:10 (instead of 32:8), are evidently feasible; the choice depends on the desired data rate. The implication of all this for the teletext decoder is a slight modification of the display memory addressing mechanism.

At least two methods of gearing have been demonstrated and manufactured in small quantities by Zenith Radio Corporation. One method has been used in the Virtext and Virdata series of products (see Ciciora, 1981) and the other in the officially proposed adaptation of U.K. teletext for the North American market. In both cases modest external circuitry has permitted the Mullard integrated circuit chip set to be used.

5.3 DIDON-ANTIOPE

The display format in Didon-Antiope (see Noirel, 1979, and Storey, et al., 1980a) is independent of the data line format, as shown in Figure 5.4. (8,4) Hamming encoding is used for all control information except for the control character sequences identifying the

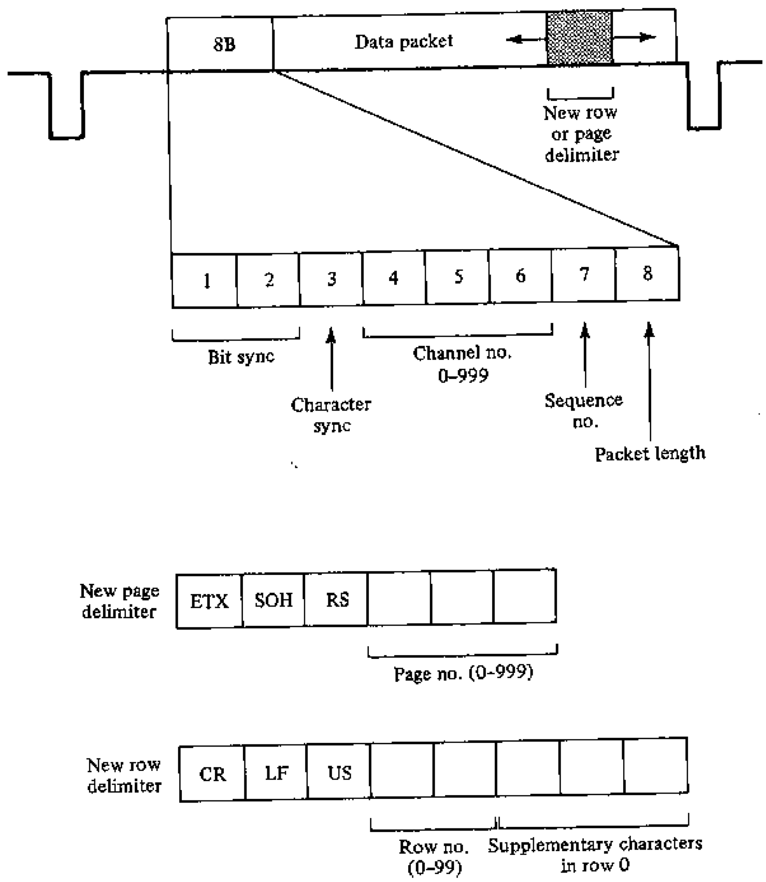


Figure 5.4 Didon data line format.

beginnings of pages and lines. These sequences can occur anywhere in the data stream. Recommended data packet sizes are 32 bytes in France (bit rate 6.2 Mb/s) and 20 bytes in North America (4.3 Mb/s). A large number (1000) of independent data channels can be specified; in teletext applications their function is similar to that of magazine numbers in the U.K. teletext. Since channels are generally multiplexed, a packet sequence number is used (independently within each channel) to check that no data packet has been lost. Didon's basic asynchronous scheme is used as a starting point for broadcast Telidon and for the joint teletext standard proposal of France, Canada, and CBS.

5.4 THE FRANCE-CANADA-CBS STANDARD PROPOSAL (FCSP)

In an effort to harmonize teletext developments on an international scale experts from France, Canada, and CBS (U.S.A.) produced, in 1981, a teletext transmission system to be outlined below. Detailed specifications of FCSP (and also of the U.K. and Japan teletext systems) are found in CCIR (1981). The proposal was submitted to different standardization and regulatory bodies, such as CCIR and FCC. In Canada the FCSP is known as Broadcast Specification BS-14 and is the de facto Canadian standard; its U.S. version is called the North American Broadcast Teletext Specification.

Technically, FCSP is an outgrowth of the teletext system used in Canadian field trials and of Didon. Its main features are

- conception of the transmission medium as a number of independent one-way data channels (from Didon)
- asynchronous, multi-standard VBI and full-field operation
- sophisticated page addressing mechanism equally usable for short magazine-oriented and large tree-oriented databases; page addressing and channel numbering are combinable in a variety of ways to achieve interleaving and optimized access to groups of pages
- conformity with the OSI model.

OSI Layers

The slightly simplified data formats recognized at the different architectural levels of FCSP (as interpreted in Sablatash and Fitzgerald, 1982) are illustrated in Figure 5.5. The following parameters and functions are defined in the various layers.

Physical layer. Parameters and functions include:

- data timing, method of modulation, amplitude, and waveforms
- bit rates (provisional): for NTSC 5.72 Mb/s (364 times horizontal line rate), a rate at which modem 300-MHz to 500-MHz cable systems have no difficulty, but at which older 12-channel systems have serious levels of distortion and interference; for SECAM 6.2 Mb/s (397 times the line rate)
- a *data line* consisting of 288 bits in NTSC and 320 bits in SECAM
- useability of any available lines in the TV signal
- useability of the first 16 bits in each data line for bit synchronization (BS) (alternating 1's and 0's).

Link layer. Parameters and functions include:

- byte framing; a sychronizing sequence CS, of 11100111, following after BS
- error correction, using vertical parity and longitudinal parity check bits contained in the suffix SUF; all control information in headers, addresses, etc. are Hamming encoded.

Network layer. Parameters and functions include:

- reception of data *packets* from the link layer; a channel (packet) number CH (0−4095), a continuity index CI (0−15), and a packet structure indicator PS, in

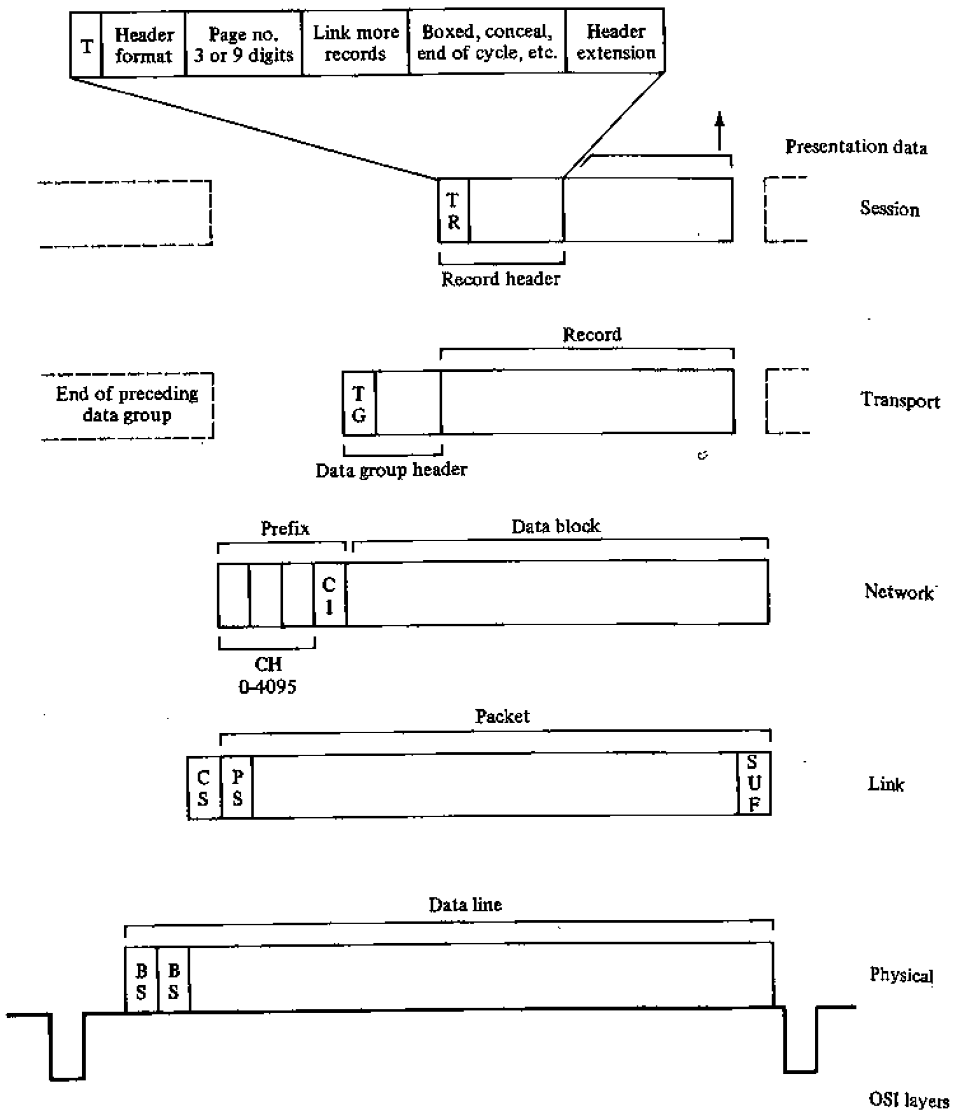


Figure 5.5 FCSP format hierarchy in OSI context. Symbols are explained in the text.

packet prefix; CI used to check the proper sequence of packets within channels, PS indicating the beginning of a *data group*

- demultiplexing of packets by channel number and sending of *data blocks* to the next layer.

Transport layer. Parameters and functions include:

- concatenation of data blocks (max. 255) from a selected channel into *data groups*; processing is controlled by the data group header, specifying the group type TG (e.g., teletext, facsimile, etc.), continuity within type, group size, etc.

Session layer. Parameters and functions include:

- reception of *records* (one per data group) of given type TR; a record is composed of a record header and of presentation data to be sent to the next higher layer (to be displayed); the data corresponds to database pages
- specification by the record header of the record (page) number (3- or 9-digit addresses are possible) and further parameters—e.g., whether the page is cyclically retransmitted or not (captioning), boxing, concealed presentation, end-of-cycle marker, and updated version of a page; 3-digit addresses represent a page or record number, 9-digit addresses represent a document and page (within document) number.

Presentation and application layers. These layers are theoretically not teletext-specific because it is highly desirable that they be identical with those used in other delivery systems. Presentation protocols, treated later in the book, generally should conform with CCITT recommendations S.100 and F.300. However, as we shall see shortly, certain aspects of the application layer do have repercussions for the lower layers.

Finally, we note that there is no true operational experience with FCSP; it will be especially interesting to see the effect of its relatively high overhead (need for control information, headers, etc., in all layers) on the overall transmission efficiency.

Channel Numbers, Page Numbers, and Multiplexing

Data transmitted in early teletext systems were organized as cyclic sequences called *magazines*, each consisting of a few tens or hundreds of pages. Magazines are distinguished by numbers, which are either implicit in the first digit of the page number and thus invisible to the user (as in the U.K. teletext), or selected by an explicit command (as in Didon-Antiope, where the magazine number is in effect the channel number). In both cases the magazine numbers can also be used to multiplex (interleave) data over the communication medium. The field trial version of broadcast Telidon uses a numbering technique similar to that of U.K. teletext.

FCSP must reconcile additional requirements with respect to database structures and multiplexing. First, it is desirable that one-way and two-way videotex services be compatible not only at the presentation level, but also, as much as possible, at the application level. Although this is not achievable for interactive applications, it is feasible in data retrieval. High page rates in full-field operation enable the transmission of larger, tree-structured databases and give the user (the impression of) interaction similar to that in a two-way environment (see also Section 10.4, on teletext databases). Second, more sophisticated image coding and data formatting methods—in particular, geometric coding—require more processing time in the terminal than is available between the reception of two packets. Therefore either the code for an entire page must be buffered in the terminal (1−2 kB), or interleaving must be used at the packet or data block level to reduce the effective rate of information transfer into the presentation portion of the terminal. (Of course, other functions of the terminal, such as detection of packet numbers, must work at real time speed.)

The channel numbering mechanism provided in FCSP is to be used both for the purpose of interleaving and for service or magazine identification. But the situation is further complicated by the fact that tree-structured databases are not easy to partition into "magazines" (channels) of related pages. How is a large tree to be broadcast at all, and how is it to be partitioned over the channels in such a way that related pages (e.g., a subtree) are kept in one channel, and interleaving is still provided at the packet level? Coherent solutions to these problems involve a further factor: the user interface. In particular, does the user see magazine numbers as logically separate from the page numbers? If so, appropriate commands must be provided and standardized, which is not yet done.

From this example we see that protocol layers cannot always be completely separated in practice: the user command interface (application layer) has direct implications for the functions of session and network layers.

5.5 STANDARDIZATION IN TELETEXT

So far, no single teletext system has been adopted as the international standard. The description of three major systems currently under discussion is found in CCITT, 1981. One reason why a separate teletext standard is hard to agree upon is the realization that the presentation (and certain parts of the application) layers of an eventual standard should be compatible with the corresponding standards for interactive videotex. Since it is impossible completely to separate out presentation coding issues from lower layers, development of a standard will probably have to wait until some solution is found to the current debate on presentation coding.

In the U.S., the France-Canada-CBS proposal (FCSP) and the enhanced U.K. teletext were examined by the FCC (Federal Communications Commission) as two major candidates for a U.S. standard. In October, 1981, the long-awaited FCC ruling on the issue was in effect a non-decision, implying that it is up to the market to decide which proposal is preferable. It may be a wise ruling, given that neither of the two contending systems has been fully implemented and tested in practice.

Numerous public service and field trial systems are operational in many countries. These implementations are based mostly on Antiope and the U.K. teletext (in Europe). In North America a great variety of systems and bit rates are used by different organizations. In addition to Didon-Antiope and the U.K. teletext, Canadian field trials use a system called Broadcast Telidon (see Storey et al., 1980b), a predecessor of FCSP. Broadcast Telidon is also used by the WETA TV station in Washington, D.C. Another system similar, but not identical, to the enhanced U.K. teletext is used in the Virtext and Virdata services in the U.S. (see Ciciora, 1981a).

5.6 SHORT SUMMARY OF TELETEXT EVOLUTION

Evolution in teletext seems to have proceeded along the following directions:

- from fixed (synchronous) format transmission towards asynchronous systems in which the units of transmitted information (packets) are independent of the application
- from magazine-oriented (small-cycle databases) towards multipurpose, multiservice systems; DOV is becoming a generalized one-way shared medium, useable simultaneously by many information sources and destinations; examples of such services are teletext, telesoftware, captioning and broadcast facsimile; Didon was the first to apply this approach
- towards independence from the underlying TV standard and the number of lines available for data transmission
- towards inclusion of all modes of presentation and compatibility between one-way and two-way operation
- towards creation of a hierarchy of presentation codings (as in the enhanced U.K. teletext proposal), the first level containing the "lowest form of life," higher levels including advanced features such as graphics and high-resolution still pictures, lower-level decoders (the cheapest) presenting "fallback" screens when receiving higher level codes.

6 Topology of Videotex Networks

We have seen in Chapter 3 that a typical videotex system is a distributed computer network involving three types of physical components: user and service-provider terminals, communication links and networks, and various computers supporting applications, service providers, system control, and gateway functions.

So far in Part II we have dealt with means and systems to deliver videotex information to user terminals. Now, to complete the treatment of the communications level, we shall discuss the layout of entire videotex networks, with emphasis on the topology of the communication subsystems involved. We do not discuss standards extensively here; the main concern of this chapter is the physical, and not the logical, structure of videotex systems.

In addition to delivery nets, videotex networks may include transmission components for communication between

- access machines and other computers (external or internal to the videotex system)
- service-provider terminals and service computers
- different service (eventually system control) computers
- user terminals.

Telecommunication facilities employed for the above functions include dedicated digital links, circuit-switching networks (public telephone and digital), packet-switching networks (PSN), and possibly satellite transmission. Because of their importance for videotex, we shall first briefly summarize the evolution and principles of PSN.

6.1 PACKET SWITCHING NETWORKS

Public Networks

The telecommunications industry reacted to increasing demands for data communications by offering leased digital circuits and by establishing *public data networks* (see Halsey et al., 1979). The term *public* implies that just as with the telephone network, there are no restrictions upon access to the offered transmission services; whoever is willing to pay can use the network.

Distinction should be made between computer networks (such as videotex) and data networks. The latter are merely data carriers, whereas the former are complete systems that include various applications and computer-based services.

A well-known example of a public data network is the telex system and its modernized successor, the teletex. Data networks designed for communication between computers require much higher data rates, shorter call setup times, and lower error rates than telex can provide.

Data networks fall into two large groups: *circuit switching* networks and *packet switching* networks. Circuit switching is a technique similar to that used in telephone networks. The underlying principle is that following an initial call, a physical path is established between the caller and the receiver, and lasts until the end of the data dialog. Included in the notion of physical path are the time and frequency intervals used to carry data over multiplexed trunks (the circuit behaves as if it were a real electrical connection with a characteristic bandwidth and delay). The important thing to note is that charges are paid for connection time, and not for the amount of data transmitted just as, in the case of a telephone call, one does not pay for the number of words spoken but for the call time. This is in contrast with the tariff structure of leased circuits, which are billed on a per-month basis, independently of actual connection time.

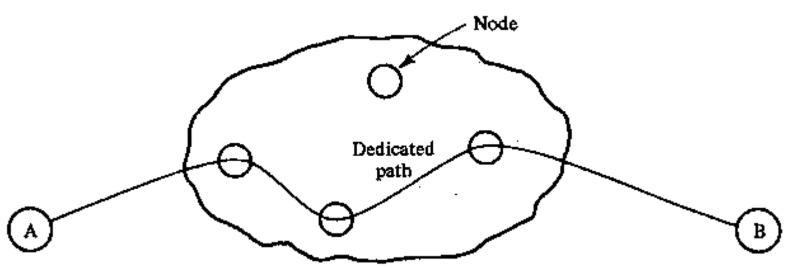
Since data transmission in most applications is remarkably bursty (as for example in retrieval from videotex databases, in which a few characters of user commands are followed first by a few seconds of page transmission, and then by perhaps a 30-second thinking period), even circuit switching is fundamentally uneconomical. A different kind of connection is needed—one for which the user would pay for the amount of data transmitted—in effect, "for the number of words spoken" but not for the periods of silence. The solution lies in the technique of packet switching.

Packet Switching

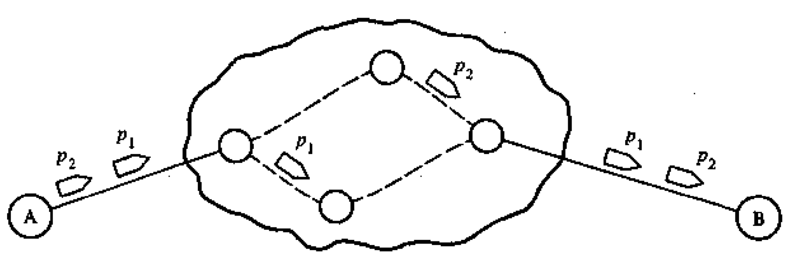
The basic principle is simple: the sender (computer, terminal) *A*, accumulates a certain quantity of data in a buffer, appends to it a header containing the receiver's (*B*'s) address, and transmits the resulting information *packet* to the PSN. The network, using the address as routing information, will forward the packet through its nodes until it reaches the receiving computer or terminal. Figures 6.1(a) and 6.1(b) illustrate the difference between the two classes of networks. In packet switching two packets, p_1 and p_2, sent by A may be forwarded inside the network through entirely different paths and with different delays. Internally, packets are treated as distinct messages passed from node to node; usually there is no dedicated circuit from A to B.

There are three different communication services typically offered by PSNs: *datagram*, *temporary virtual circuit*, and *permanent virtual circuit*. Datagrams can be likened to a mail service: individual packets (datagrams) are transmitted independently through the network, and they may arrive out of order (in a different sequence than sent) or may be lost. Since datagram service is intended to be a low-overhead and inexpensive service, error correction is usually not included: it is up to the receiver to detect errors and request retransmission if necessary.

Virtual circuits, on the other hand, simulate a perfect channel; all packets are (supposed to be) delivered error-free and in correct sequence. Permanent virtual circuits resemble full duplex leased lines; no call or addressing is necessary because the receiver's address is associated permanently with the circuit at setup time. (The charging structure is a fixed fee plus a per-packet charge.) Temporary virtual circuits are established by virtual calls (similar to telephone calls); after circuit setup, packets do not need



(a)



(b)

Figure 6.1 (a) Circuit-switching network; (b) packet-switching network. Packets p_1 and p_2 may go by different paths and may arrive out of order in datagram service.

to carry their destination address, even though they may travel from source to destination by different paths. A circuit can be cleared at the request of either participant.

In addition to sequencing and error correction, virtual circuits provide for *flow control*; the network can instruct the sender to stop or restart transmitting packets to prevent congestion and loss of packets that cannot be accepted fast enough.

Applications of PSN

None of the above two types of service is superior to the other in the sense that there are applications for which one service is more suitable than the other. For example in videotex, when a user signs on for a data retrieval session from a remote database, a temporary virtual circuit is the optimal means of communication. The best way to carry out frequent transactions between an access machine and an external computer might be through permanent virtual circuits (as in the Bildschirmtext network). When a service provider wants to update a database page, he might use a single datagram (with additional error protection) to avoid the overhead for circuit setup. The user-to-user messaging envisioned in some videotex systems is another suitable application of datagrams. In situations in which the reliability inherent in datagrams or PSN-provided virtual circuits is not sufficient (circuits can occasionally be broken due to network malfunctions), the sender and receiver can build their own "virtual circuit," incorporating features best suiting their needs. (This is essentially done in the new ISO transport protocol.)

Access to PSN

It is of prime importance to have standardized procedures for accessing network services. Most PSNs comply with CCITT Recommendation X.25 covering the rules for the communication between a PSN and a subscriber.

X.25 is structured into three layers, corresponding to the first three layers of the OSI model. The physical layer is defined by Recommendation X.21, which covers electrical characteristics and bit signaling sequences for synchronous communication between terminals and data networks. The link level is essentially defined by a procedure called HDLC (High-level Data Link Control), standardized by ISO. It provides for a full duplex transmission of error-free blocks of information and for flow control. The main function of the network level of X.25 is to establish and control virtual circuits and to multiplex them onto a single HDLC link. All information is formatted into packets; different packets such as Data Packets, Call Packets, etc. are distinguished by codes in packet headers.

Two clarifications are appropriate at this point. First, although X.25 specifies virtual circuits, it is *not* an end-to-end protocol (see Figure 6.2). When terminal A wishes to be connected with B, two virtual circuits are in effect established: between A and the network node N_1, and between node N_2 and B. However, this is largely unseen to the users A and B.

Second, formatting and analyzing packets requires programmable intelligence in the terminals. In order to enable older "dumb" terminals (e.g., teletypes) to be attached

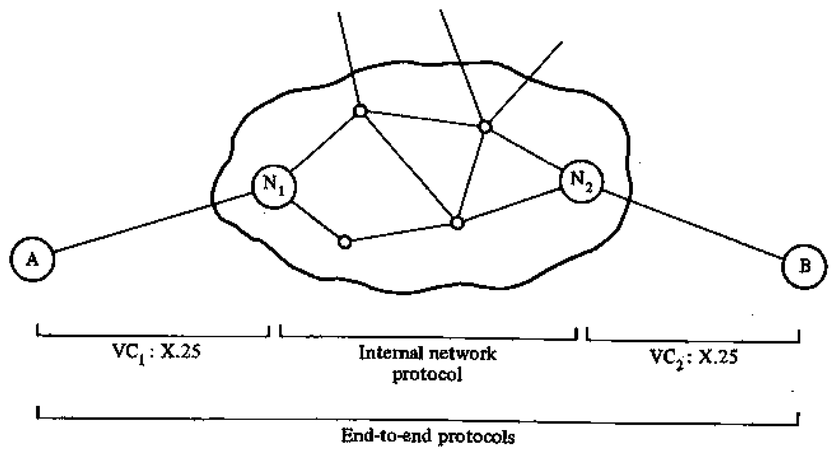


Figure 6.2 Virtual circuits (VC) in a packet-switching network and the span of end-to-end (transport) protocols.

to PSN, CCITT devised another set of recommendations—X.3, X.28, and X.29—which define interfaces to a PAD (packet assembly/disassembly) function implemented in some network nodes. For more details see Tanenbaum (1981). Many PSNs are in current use all over the world; some of the largest are Telenet and Tymnet in the U.S., Datapac in Canada, Datex-P in West Germany, Transpac in France, and Euronet, a network linking most Western European countries. Haber (1981) discusses the possibility of using Euronet for interworking between videotex and other data services.

It is desirable that data be exchangeable between PSNs. Such interworking is not simple, either technically or politically. There are minor differences even between X.25-compatible networks (such as in maximum packet size, address assignments, and flow control parameters), not to speak of difficulties encountered in PSNs offering datagrams and virtual circuits. Interfacing between networks is accomplished by *gateways* (different from videotex gateways) implemented either as dedicated computers or as software modules residing in some nodes of the networks involved.

6.2 ELEMENTS OF VIDEOTEX NETWORKS

Existing and planned videotex systems employ many combinations of available means of communication, and it is hard to fit them into a classification scheme. An illuminating analysis of networks is found in a number of reports published by Butler Cox & Partners (1982). The following two (not necessarily independent) criteria can be helpful in distinguishing between various classes of networks: (1) the degree of "system distribution," measured by the number of levels of concentration found between terminals and service computers, and which roughly correlates with the size, performance, and maturity of the network, and (2) the ways in which user-oriented data is distributed and moved around between computers. First we will show a number of typical although somewhat simpified network "archetypes" (see also Ball et al., 1980), followed by examples of real networks. Data distribution, gateways, and compatibility issues will be discussed further in Chapter 11.

Access Machines

The access machine (AM) is a key element present in practically all viewdata systems. It can be defined as the first point, on the way from standard user terminals into the network, at which processing power and storage are found. A wide range of names has been used for AM, including videotex exchange, server, enquiry center, service computer, videotex center, concentrator, and retrieval center. This variety reflects quite well the variety of roles that AMs have in various systems.

Here is a representative list of the functions (in the order of increasing complexity) of AMs:

Communication. Functions include:

- line control and user terminal handling, echoing, and error detection
- local buffering and concentration
- gateway functions, such as displaying lists of available services and external computers, providing automatic logon to these services, and providing dialog contol with external service computers.

Application. Functions include:

- user assistance—e.g., in the form of "Help" commands
- service provider support
- partial database (to save communication cost and time, a subset of frequently used pages can be loaded and stored in the AM; the subset can be static—loaded by operator commands, or dynamic—e.g., a set of the currently most popular pages)
- mailboxes for user-to-user messaging
- personal computing and telesoftware, applications that require frequent user interactions and that thus can be executed advantageously in AM.

System management and control. Functions include:

- user identification, access control, and statistics
- monitoring of network status.

In actual implementations the functionality and power of AMs vary according to the capabilities of the rest of the system. At one extreme the AM may be the only computer in the system (e.g., in smaller stand-alone trial and private systems) and would then carry all vital functions. At the other extreme (typically in large commerical networks) AMs may be limited to communication and gateway functions, the remaining tasks being taken over by other computers.

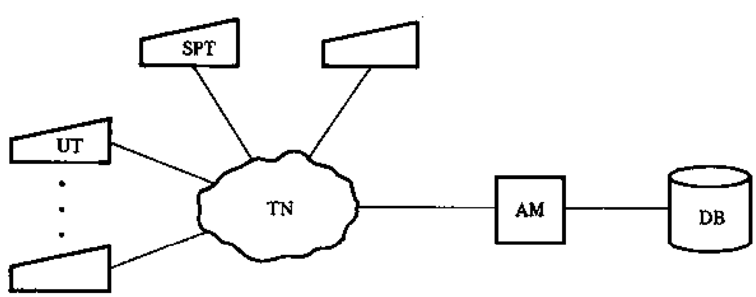
Specialized applications, such as database retrieval, banking, and teleshopping, are typically supported by dedicated computers. These fall into two classes, in the first of which services are designed and managed by the videotex operator (these can be logically considered as an extension of access machines). The second, and more important, class is constituted by external computers, which run already existing applications that are in principle independent of videotex. Incompatiblities between these applications and videotex formats and commands are resolved by gateway software residing partly in the external computer, partly in the access machine (see Chapter 11).

6.3 NETWORK TYPES

Centralized and Replicated Databases

The network in Figure 6.3 is centered around a single access machine whose main functions usually include, along with all other tasks of system management and terminal handling, supporting the database. It is obviously the simplest possible configuration and is relatively easy to implement. Such a system is typically used for experimental field-trial purposes or as a private system, and is generally used in situations in which the number of active users does not exceed a few hundred people within the reach of the local telephone network. Disadvantages include limited capacity, the eventual necessity of long-distance calls, and vulnerability to failures. Importantly, similar centralized configurations frequently recur as local subsystems of larger networks that serve a cluster of users. This is an example par excellence of the fact that a system is a subsystem (and vice versa).

A certain improvement can be realized by separating the terminal buffering and line-handling functions from the main computer, as shown in Figure 6.4. The front-end computers are physically close to the main computer; if they were remote, they would become, in effect, concentrators.



SPT: Service-provider terminal
UT: User terminal
TN: Telephone network
DB: Database
AM: Access machine

Remark: These symbols will be used in most of the figures throughout this chapter; additional symbols are introduced in later figures as they are used.

Figure 6.3 Simple, centralized system/subsystem.

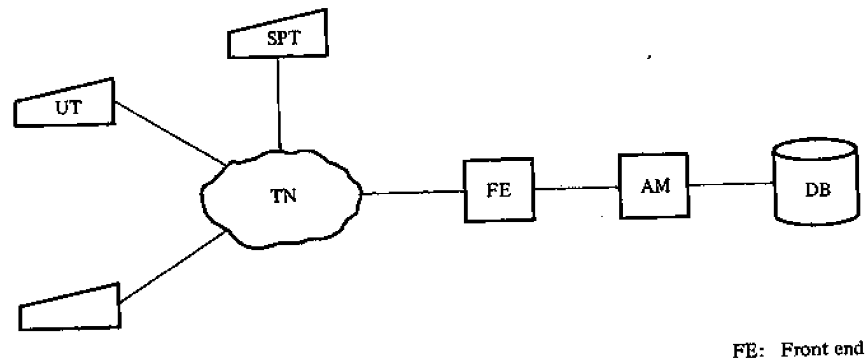


Figure 6.4 Centralized system with front end.

A brute-force solution to the limitations of centralized systems is sketched in figure 6.5. Here, each of a number of access machines with identical (replicated) databases serves a local group of customers. However, database updating is done in a dedicated computer that maintains a master database and to which all service provider terminals are connected. The replicated databases are kept in step through identical, periodic updates. This type of system has been demonstrated to be workable on a nationwide scale (e.g., the initial Prestel implementation). Its clear disadvantage is the wasteful usage of storage and computer hardware; in addition, updating cannot be practically

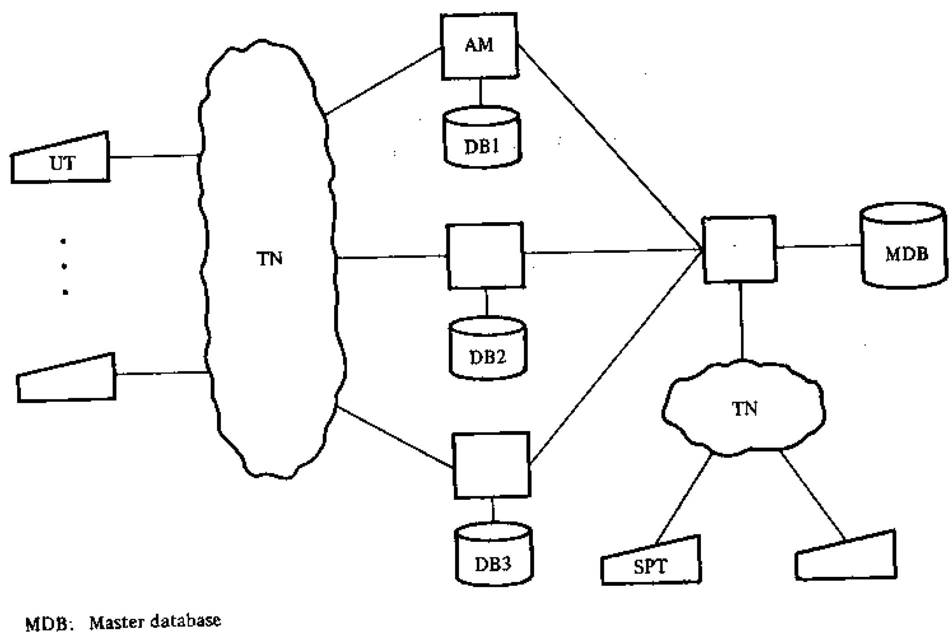
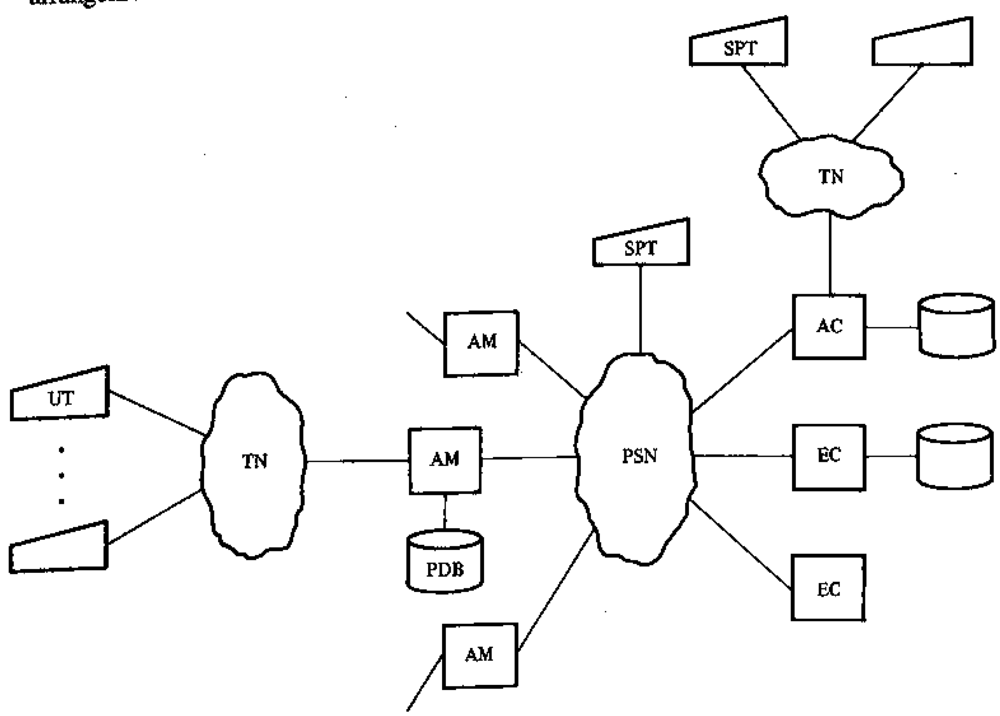


Figure 6.5 Replicated databases.

done in exact synchronism, which leads to temporary inconsistencies and potential problems with rapidly changing items of information.

Inclusion of Packet Switching Networks

A more generalized network that overcomes some of the above difficulties and provides extended (and extensible) services involves long-distance data communications through PSN (see Figure 6.6). Local groups of users are served by access machines. Large service computers specialized for videotex (and operated under videotex management), as well as external computers, can be accessed via PSN through gateways. This type of network is found, with some variations, in many large systems; examples are Bildschirmtext, the French Télétel, and the Panda network planned for Prestel. In the latter, dedicated links between distant databases and AMs will be used, in addition to PSN, to improve transmission speed and reliability. The contents of partial databases will dynamically change according to local demand. The overall objective of this arrangement is to minimize the combined storage and transmission costs.



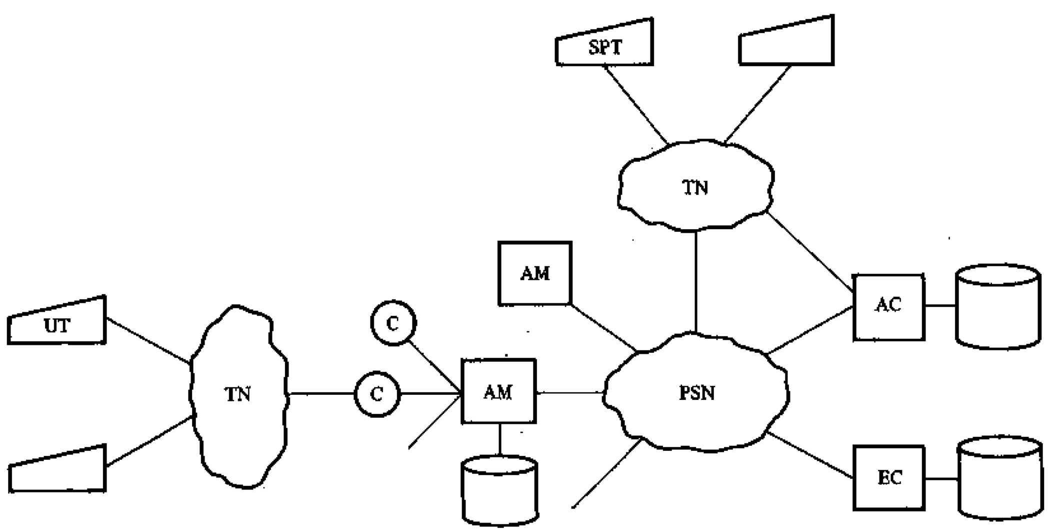
EC: External computer
PDB: Partial database
PSN: Packet switching network
AC: Application computer with videotex database

Figure 6.6 Distributed system with one level of concentration (in AMs) and with access to external computers.

In a sense, the network in Figure 6.6 is a general framework that can accomodate probable future evolutionary forms of videotex presently under much discussion. In one conjecture, future home information systems will be merely a collection of independently owned and managed services. In that case, AMs would contain only simple, transparent gateway functions (e.g., directory of available services), or might vanish altogether. At the other extreme, a significant part of the whole network would appear as an integrated and centrally managed distinct videotex system in which the identity of the computer from which a user is receiving service might be hidden to him. The network arrangement in Figure 6.6 can clearly accommodate a number of intermediate cases between these two extremes.

The AMs in the preceding network implement one level of concentration. A more general scheme (see Bosman, 1980), such as the one illustrated in Figure 6.7, includes an additional level of concentrators whose primary role is, again, to offload the AMs, and which later might become more sophisticated and eventually become another layer of AMs.

From the standpoint of database design, the tendency seems to point towards multilevel hierarchies of databases with one or more master databases and a network of AMs, each holding a partial database. Very little is known at the present about typical access patterns of individual users, and almost nothing about the collective accessing behavior of larger user populations. Only after such data (e.g., on the nature of the "working set" and its evolution in time) are obtained, can the design of distributed databases with so many degrees of freedom be optimized.



C: Concentrator

Figure 6.7 System with two levels of concentration.

Hybrid Networks

In order to simplify the issues, the only transmission networks included in our preceding network diagrams were the public telephone network and the PSN. Other means of one- and two-way communications can be and are actually used in many combinations. Figure 6.8 illustrates an example of such a combined or hybrid network. An especially attractive solution is to combine cable delivery systems (which are high-capacity local networks) with satellite transmission, which has a large geographic span. In this way large amounts of teletext data can be distributed nationwide.

Another possibility is to link one-way cable systems or DOV transmission with the telephone in order to combine the advantages of both. This can be done in two ways. The telephone can be used as a feedback path (e.g., for page requests); the demanded information is then included by the head end in the broadcast cycle. Alternatively, the most frequently used pages can be automatically included in the broadcast cycle while the others are retrieved from the database and sent over the telephone to individual users on demand. Examples of hybrid systems are Diode, a French system described by Berger and Noirel (1981), INDAX (see below), and Touch-Tone Teletext (see Robinson and Loveless, 1979).

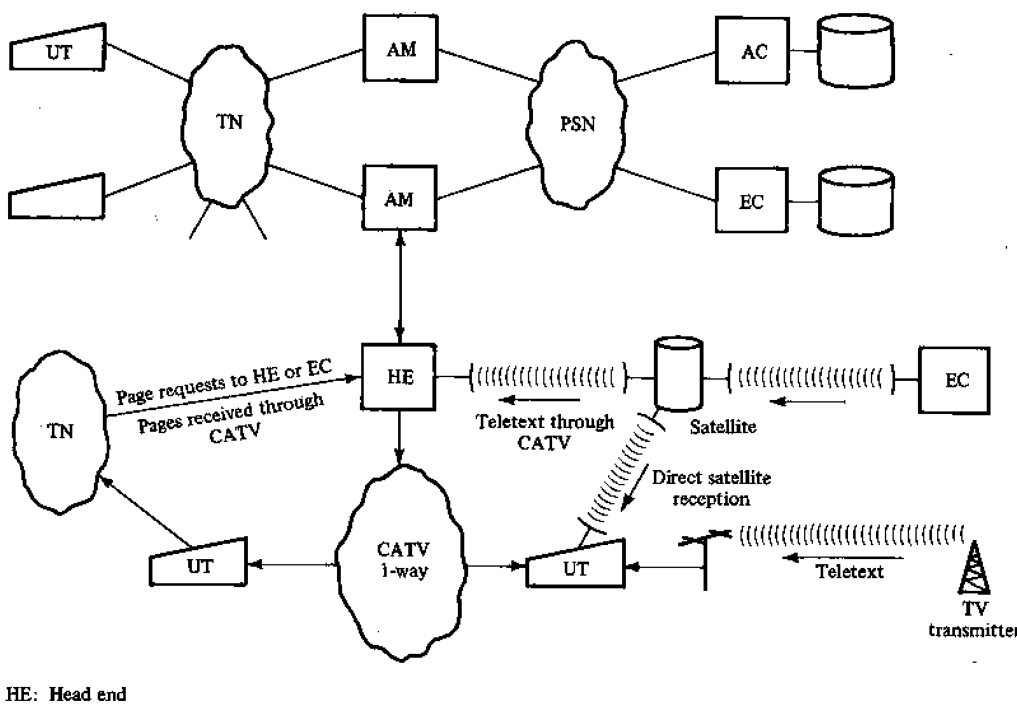


Figure 6.8 Example of combining transmission media.

6.4 NETWORK EXAMPLES

Following are a few representative implementations of videotex networks.

Prestel Advanced Network Design Architecture (PANDA)

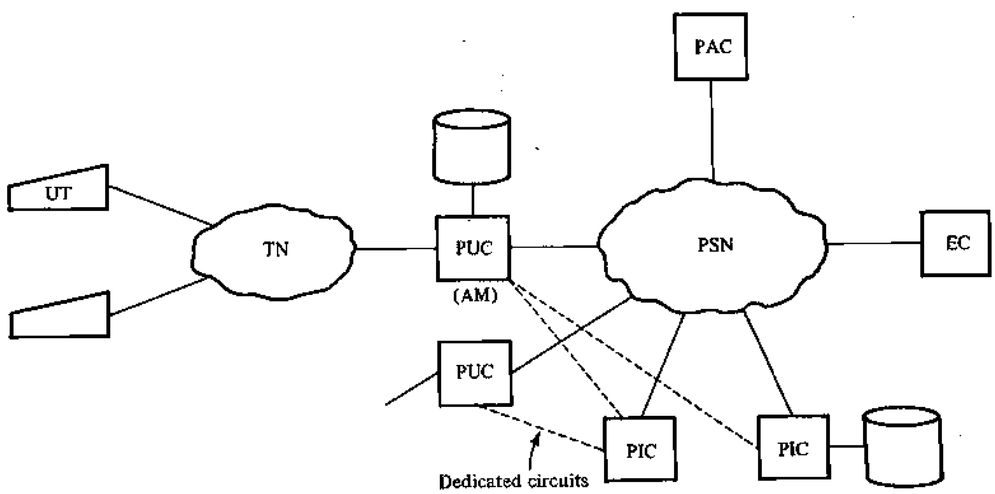
PANDA is the new network that is to replace Prestel's current system of replicated databases. The replacement process is planned for a period of about four years (1982–1986). It has three types of component nodes (in addition to external computers), each based on a local network of processors (see Childs, 1981). Their names and functions are

Prestel Administration Center (*PAC*): This will carry out billing, administration, statistics, and service-provider registration.

Prestel Information Center (*PIC*): In their totality (1 or 2) the PICs will contain a master database, together with the update (service-provider) functions.

Prestel User Center (*PUC*): This is an access point (access machine) for the users. Each PUC will hold a partial database of the most frequently requested pages. If a page is not found in the PUC, then it will be brought in from PIC, eventually replacing a seldom-used page. In effect, each PUC, together with the PIC, can be regarded as a two-level demand-staging memory hierarchy.

The resulting network is illustrated in Figure 6.9. Note that the most heavily used links between PICs and PUCs can be backed up by dedicated circuits.



PUC: Prestel user center
PAC: Prestel accounting center
PIC: Prestel information center

Figure 6.9 PANDA network.

Bildschirmtext Network

The network to be used in the BTX public service to begin in 1983 is being designed by IBM Corporation (see Grieble, 1982). It retains from the trial network the notions of BTX-center and access to external computers through gateways. It can be seen as a three-level hierarchy of databases (see Figure 6.10). Level 1 contains the central facility. It consists of two IBM 4341 processors holding the master database (up to 2 million pages), all user data, and a global mailbox. Level 2 is the database level of BTX-centers. Each center is a multicomputer composed of up to eight IBM System/1 processors connected by a high-speed bus. Two of these processors hold partial databases. The plans call for 10 BTX-centers by 1984. Level 3 is the lower level of BTX-centers, occupied by the remaining six System/1 processors, called *customer computers* (CC) (which are access machines), two of which may be remote instead of being located on the common bus. Each CC has its own partial database and local mailbox and can accommodate up to 96 user ports. Service provider terminals are connected through the telephone network or by dedicated lines, and external computers are connected through an X.25 network. The purpose of remote CCs is to reduce the telephone traffic.

Management of data. The databases in the three levels form a demand-staging hierarchy. Pages are moved automatically between the levels according to demand; if a page asked for by a user is not found in his CC, a request is made to the second or, eventually, the first-level database. When it is found, its copy is staged (sent) to the appropriate CC, replacing there a seldom-used page. About 50,000 and 100,000 pages are stored in levels 3 and 2, respectively. It is expected that 95% of all requests will be satisfied from level 3, and 98% from either level 3 or 2. Thus only 2% of all page requests will reach the central facility.

When a page is updated (in level 3), all of its copies are cancelled and the updated copy is sent to level 1, thus assuring that when demanded again, the proper version will be obtained. All this is done in a way transparent to the user. User data and messages (mailboxes) are handled in a similar way.

The Vélizy Field Trial

This extensive field trial, supported by the French Government and realized by the Ministry of Post and Telecommunications, aims to test the technical concepts behind Télétel (the French interactive videotex) and its acceptance by the providers and consumers of services. The trial has been running since 1981 in three cities near Paris: Vélizy, Versailles, and Val-de-Bièvre; hence the acronym 3V often used to designate it. In January 1982, there were nearly 3000 user terminals (mostly in homes), with 150 services available. Many of the services are transactional, running on external computers. One of these is a messaging service that includes mailboxes for all subscribers. Most terminals are based on TV sets and all of them are equipped with a small alphanumerical keyboard augmented by a number of keys for dedicated functions (see Section 10.3).

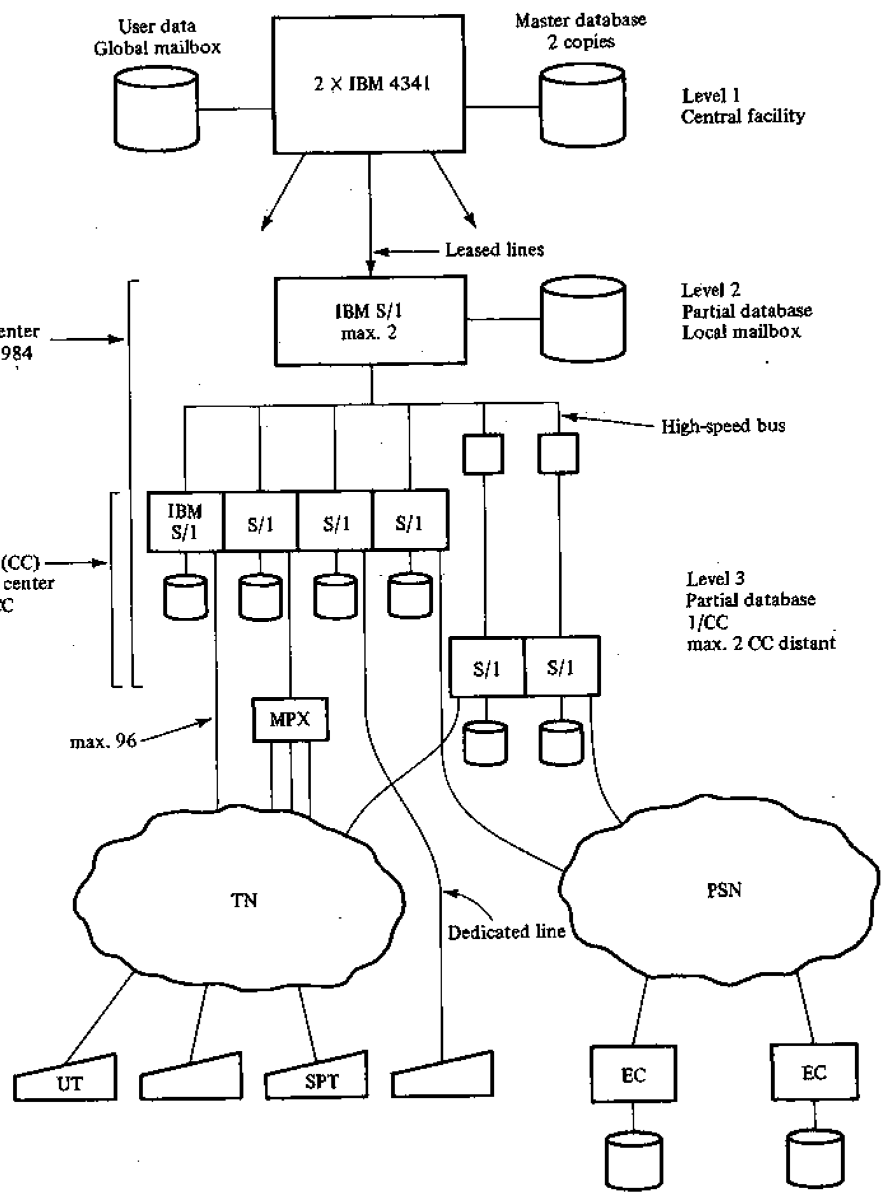


Figure 6.10 Bildschirmtext commercial network.

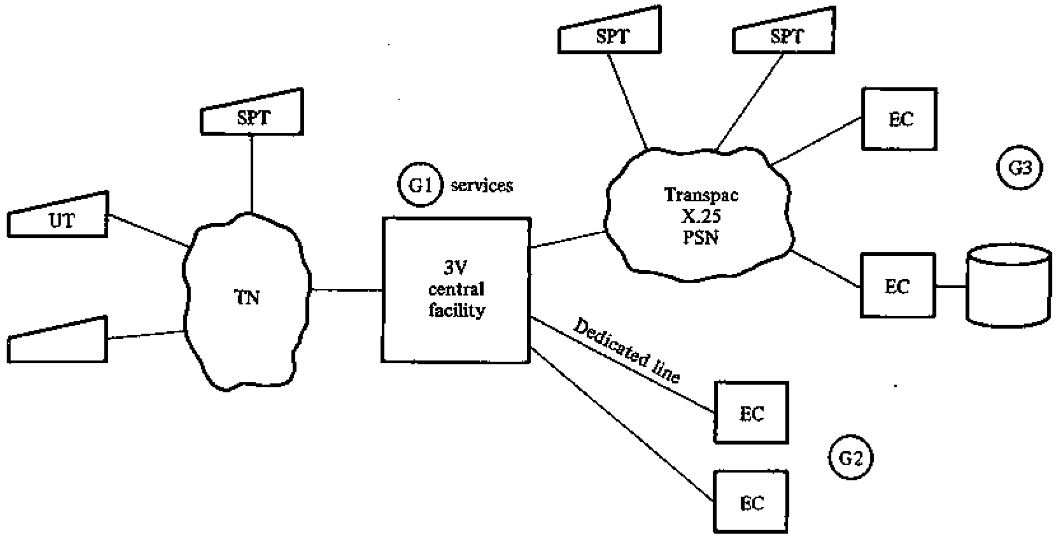


Figure 6.11 Télétel 3V network.

System structure. The overall network structure is illustrated in Figure 6.11. It consists of a central facility (see Figure 6.12), user terminals, service-provider terminals connected through the telephone network or Transpac, and external computers working either through Transpac or dedicated lines; the latter are generally used for smaller distances.

Applications (here called *services*) fall functionally into two large groups: *database* and *transactional*. The former conform with the Télétel page-oriented database format. All other services are transactional. From the network point of view services are classified into three groups: G 1, running on the central facility (these are database services), G 2, and G 3. The latter two applications are those running on external computers, linked through dedicated lines (in the case of G 2) or Transpac (in the case of G 3). (See Télétel, 1980).

The central facility is a multicomputer complex composed of seven processors communicating through dedicated 38 kb/s HDLC links. The communication function is assured by two Datanet 7103 processors that are also charged with the control of two access lines to the Transpac packet switching network (each capable of carrying up to twenty 2400 b/s virtual circuits) and 25 dedicated lines to external computers.

User terminals and G1 services are handled by three identical Honeywell-Bull Mini 6 computers (access machines), each carrying a copy of the master database. Each processor has 100 ports; thus the system can support up to 300 active users at any time. In addition, these access machines handle the dialogs with external services. Two Mini 6 computers serve for system management (statistics, billing, reconfiguration) and service-provider support (master database, updates).

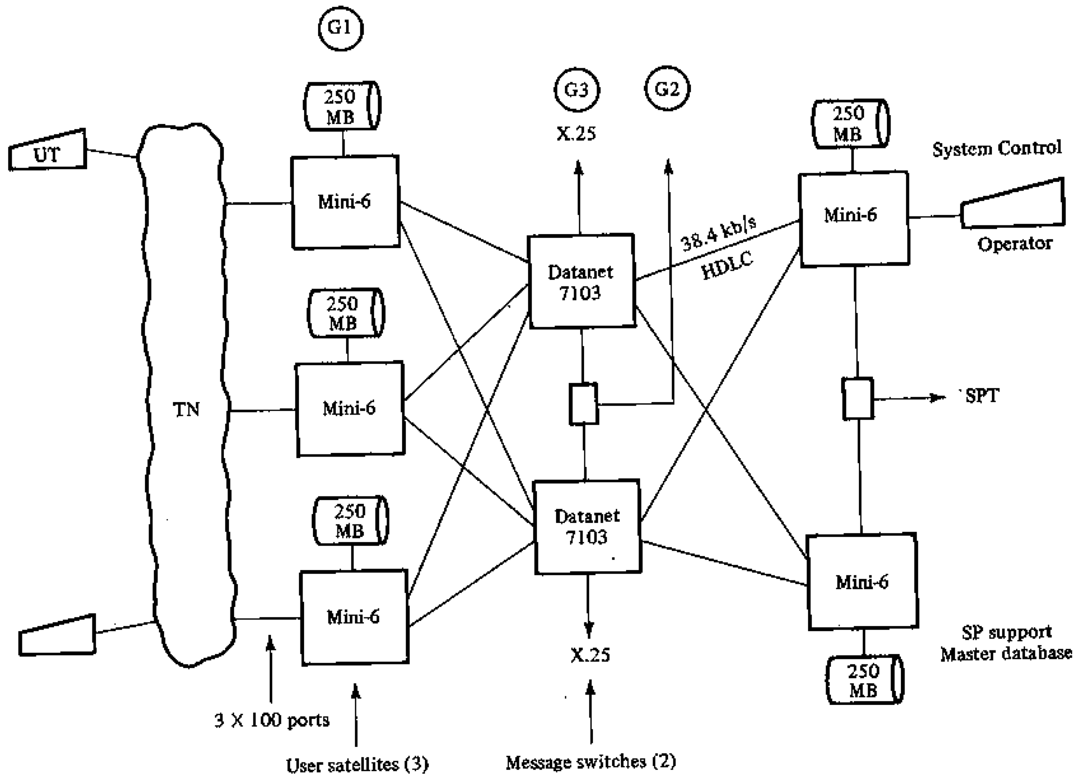


Figure 6.12 Télétel 3V Central Facility.

Connection to external services. In Télétel, as in Prestel and Bildschirmtext, great emphasis is placed on the access to services running on external computers. As discussed in the section on gateways (see Chapter 11), these services may or may not have been designed with serving videotex terminals in mind. In the former case (see Télétel, 1979), connection to external services takes place on three protocol levels, as shown in Figure 6.13.

Network level: The interface is X.25 (for G 3 services); standard software packages for this function exist for a variety of computers.

Télétel dialog level: The establishment and closing of a logical connection with a service is done by the access machine on behalf of the requesting terminal. The message exchange accomplishing this and some other control functions (reinitialization, test, modification of the permissible number of users, etc.) constitutes this protocol level.

Terminal dialog: When the logical connection is established and initialized, a dialog between the terminal and the remote service takes place while the access machine remains essentially transparent. For this dialog to be meaningful, it must be carried out in conformity to the Antiope presentation standard.

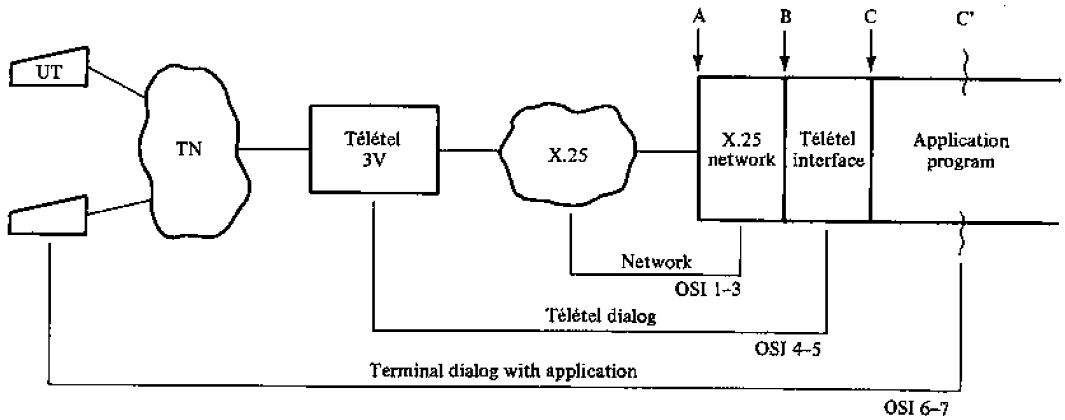


Figure 6.13 Protocols in Télétel external service connection.

There are several alternatives for the implementation of the above three protocol functions:

1. All three functions may be implemented in the external computer. If the application is not designed initially for this, then the necessary modifications may turn out to be quite complex and costly (see interface A in Figure 6.13).

2. The X.25 interface may be in a simple front-end processor (see interface B).

3. A complex front-end computer may take charge of the network and Télétel dialogs (see interface C). In effect, once the decision is made to include a front-end computer, it can offload the external computer by taking over further functions, such as conversion into videotex format, handling of error messages, and processing of user dialogs (see interface C′).

When interfacing to existing external applications is required without the possibility of changing them, all necessary interfacing is done in the access machine, which appears to the external computer as one of its terminals.

3V trial statistics. The following statistical data were made public, after eight months of functioning, in February, 1982:

- terminals:

domestic	190
schools, service providers, public places	755

- services (approx.):

local (G 1)	150
distant (G2, G 3)	40

- calls:

mean number of calls/terminal/week	2
mean session time (min.)	14
mean number of services consulted/session	3.5

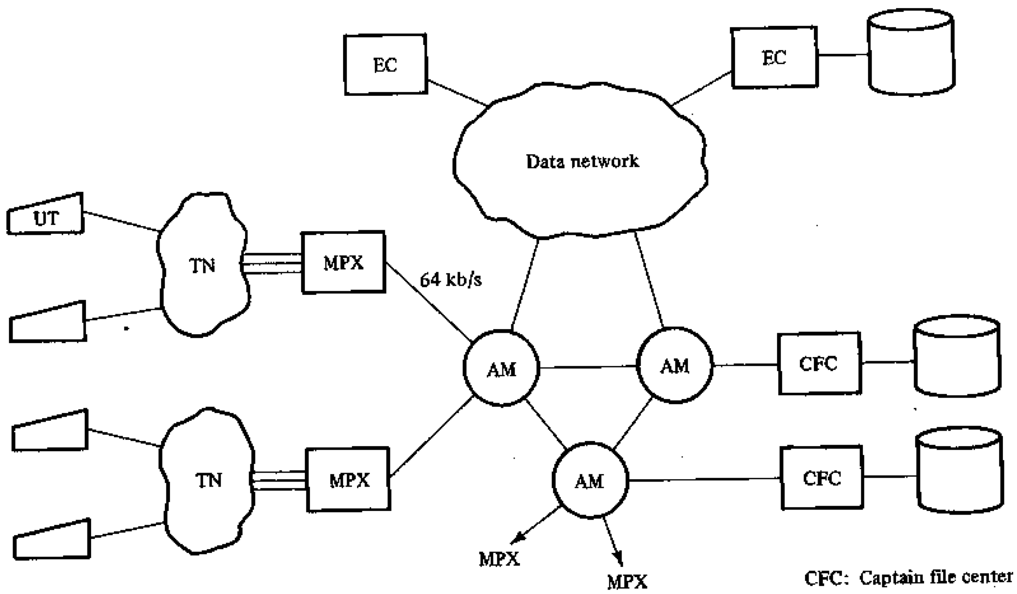


Figure 6.14 Captain network.

Captain Network Architecture

The network planned for the commercial version of the Japanese videotex service Captain has two levels of concentration (see Harashima and Kobayashi, 1981), as shown in Figure 6.14. On the first level are simple multiplexers that concentrate the telephone traffic from a group of users. The multiplexers connect to access machines through 64-kb/s lines. The Captain version of the access machine has no database; its functions, in addition to the usual line and terminal handling, include

- formatting of data from databases
- character generation (since information from databases comes in coded form); this function is to be moved later to user terminals
- billing and statistics
- access to external computers.

Similarly to all other major national networks, databases and other applications run on computers of two types: Captain file centers (under videotex management) and external computers.

INDAX Interactive Cable Network

INDAX is a set of local area CATV-based two-way networks for residential applications, developed by Cox Cable Communications in the U.S. (see Gates and Tjaden, 1982).

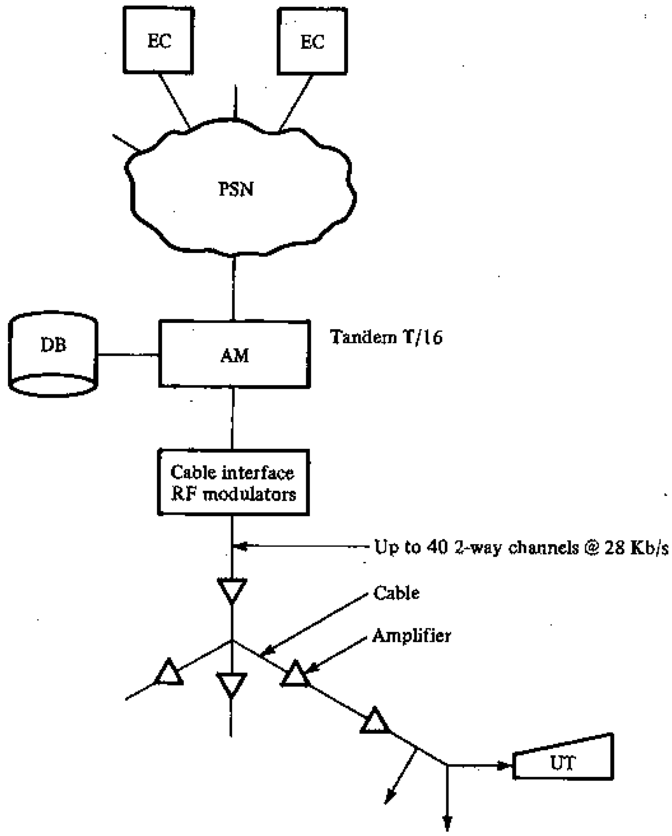


Figure 6.15 INDAX system diagram.

Individual local networks are linked through a PSN with remote external service-provider computers (e.g., The Source, banking, and teleshopping computers).

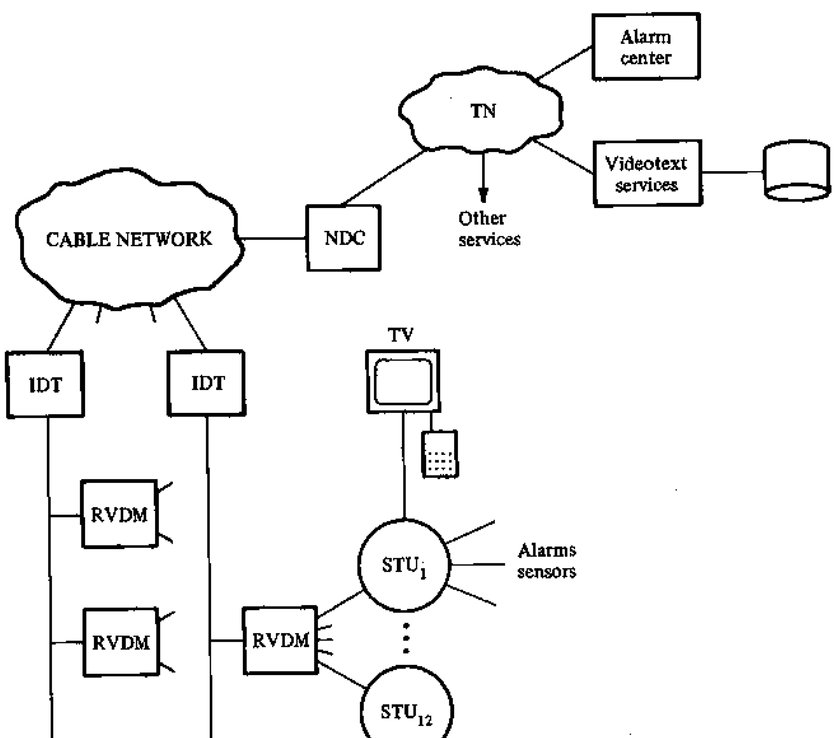
The access machines used in the local networks (cable head-ends illustrated in Figure 6.15) are Tandem T/16 processors; their number can be between two and 16, according to need. These machines control all local databases and transactional applications, as well as two-way communications with users.

Some interesting data on the utilization of the cable's 400 MHz band are available. Most of this bandwidth is used for standard TV channels. Downstream and upstream data are transmitted in the 108–132 MHz and 17–29 MHz bands respectively. Within these bands are placed a number of data channels, each occupying 300 kHz; the bit rate in each is 28 kb/s. This allows for up to 40 two-way data channels (each employing a pair of upstream and downstream channels), which are available throughout the network. Downstream traffic is by addressed packets (messages). A CSMA/CD scheme (see Chapter 4) is used to resolve access conflicts in the shared upstream channels. Addressable control messages are used for frequency reassignment or for cutting off noisy parts of the cable network.

This type of network is capable of combining in an integrated fashion one-way and two-way delivery as done in INDAX for data retrieval applications. Highly demanded pages are included in a teletext-like broadcast cycle, whereas personalized or rarely used pages are transmitted on demand. Data channels are dynamically assigned to either type of traffic.

The IDA Trial

IDA is the name of a field trial sponsored by the Manitoba Telephone System in Canada (see Coyne, 1981). It involves Omnitel, a CATV-based network. The two-way cable serves as an ISDN-type medium that delivers to the participating homes a wide variety of services: standard and pay TV, digital telephone, data, remote metering, and alarm services. Here again, most of the bandwidth is consumed by analog TV. Digital traffic for up to 2,000 (100 in the actual trial) homes is carried on 16 1.5 Mb/s channels. These



NDC: Network and distribution control
IDT: Intermediate distribution terminal
RVDM: Remote Video and Data Multiplexer
STU: Subscriber Terminal Unit

Figure 6.16 IDA trial network.

effectively form a packet switching subnetwork. The network topology is illustrated in Figure 6.16. It contains two levels of concentration, represented by IDT (Intermediate Distribution Terminals) and RVDM (Remote Video and Data Multiplexer) units. The main functions of IDT are traffic control and multiplexing. RVDMs contain a number of display memories and display generators for videotex, to be shared among 12 subscribers. This permits a substantial lowering of the terminal costs. Homes are equipped with microprocessor-based Subscriber Terminal Units, which control all sensors and other devices.

Part III
THE PRESENTATION LEVEL

The three chapters in Part III deal with presentation-level issues. Since these issues are highly videotex-specific (and somewhat controversial), they are also of central interest for our purposes. First, in Chapter 7, we discuss the underlying coding principles, such as the available options and the meaning of attributes. Chapter 8 is about real-life systems and standards, and Chapter 9 is devoted entirely to the structure and function of terminals.

Although in many places we go into considerable detail, our main goal remains throughout to explain principles, and not to copy technical manuals. Thus, coding examples are given with simplified, symbolic versions of the actual codes.

It should be emphasized that in the broader context of the OSI architecture, the issues covered in Part III correspond only to a subset of OSI presentation level (virtual terminal) protocols, which were also discussed in Chapter 3.

7
Principles of Presentation Coding

7.1 THE TELEVISION SET AS DISPLAY

One of the fundamental premises of videotex is the use of the home television as a display terminal. This of course does not exclude the use of other display technologies. Nevertheless, the methods of presenting videotex information have been shaped to a large extent by the capabilities and limitations of standard TV sets. It is unfortunate that these capabilities do not completely cover those of standard business video display units (VDU). Life would be easier for standardmakers and system designers if the two could easily be made compatible.

The Display Area

When a page of videotex information is being displayed, it is essential that the whole page (*information area*) be entirely contained in the visible area of the screen, which is called the *display area*. Because of variations and aging of TV sets, it is impossible to align precisely the page boundary with the display area; a safety margin, called the *border area*, must be introduced around the information area. This can be considered a kind of "underscan," in the sense that it is the opposite to overscan, which is aimed at assuring that the display area be completely covered with scan lines. These notions are illustrated in Figure 7.1. The net effect of overscan, underscan, and VBI (vertical blank-

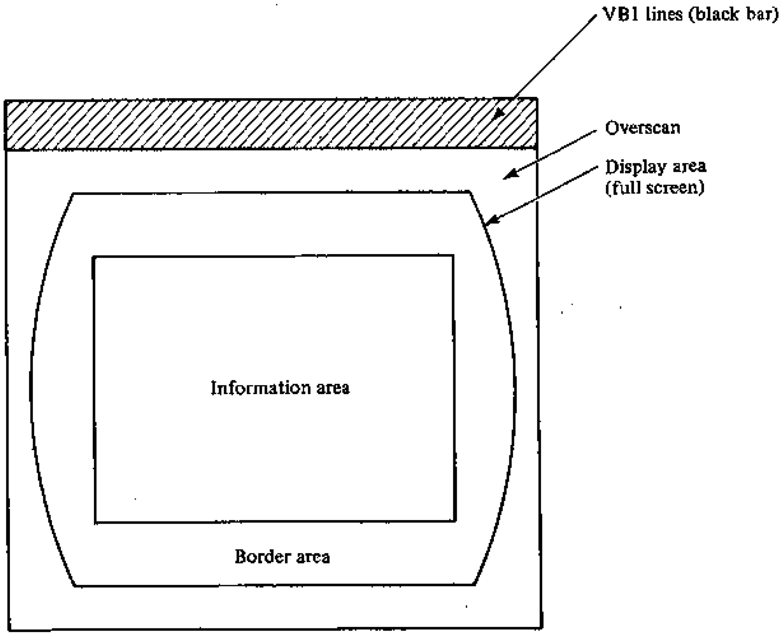


Figure 7.1 Areas of the television display.

ing interval) lines is the reduction of the number of TV lines available for page display. Practically no more than about 440 (out of 525) lines are left in NTSC, and 550 (out of 625) lines in PAL and SECAM.

Vertical and Horizontal Resolution

We define a *pixel* (picture element) as a dot in the image that can be distinguished from neighboring pixels by its color and intensity. The *resolution* (definition) of a display device (measured in pixels or pixels/cm) is the maximum number of pixels the device is capable of displaying in each of the horizontal and vertical directions. From the preceding it follows that the vertical resolution of TV-based videotex display is around 400–440 pixels in NTSC.

The issue of flicker in a television image is a complex one. Flicker is a subjective phenomenon that increases in objectionableness with increasing brightness, image size, space between scan lines, and reduction of refresh rate. Large bright areas will flicker more than small or dim areas. Thus, for example, flicker in the upper-case letter E may be objectionable if the letter is bright and the long upper and lower strokes of the letter do not have equal representation in both even and odd interlaced fields. (This leads some to estimate conservatively the vertical resolution as no more than 200–220 pixels.) If the center horizontal stroke is shorter, it will display less flicker and the elements of the vertical bar will show essentially little flicker. By careful application of these principles (and of character rounding), vertical resolution can be greater than half the number of scan lines.

The horizontal resolution is limited by the bandwidth of the video signal. In particular, the number of pixels displayable on a horizontal line is equal to the maximum number of times the electron beam can be switched on and off while sweeping the line. This number can be obtained by multiplying the upper limit of the bandwidth (around 4 MHz in North America) by the time the beam takes to cross the information area (about 80% of the scan line cycle—that is, about 50 μs). The scan line cycle is $1/(n_f \cdot n_\ell)$, where n_f is the number of TV frames per second and n_ℓ the number of lines per frame. The results for NTSC and PAL scan line cycles are very nearly the same: about 64 μs. All this works out to a horizontal resolution of $4 \times 10^6 \times 5 \times 10^{-5} = 200$ pixels. This value can be compared to the resolution of standard VDUs, which is usually 512 × 512, and of high-resolution graphic terminals (e.g., 4096 × 4096).

Unfortunately, even the above-derived low resolution of the TV set cannot be taken for granted. First, at the upper limit of the bandwidth, the signal modulating the electron beam can only be a sine wave (and not a square wave); therefore the pixels will not be drawn as sharp dots, but as somewhat fuzzy areas. Second, the video bandwidth as indicated by various TV standards relates to the transmission, and there is no guarantee that all TV sets will actually amplify all frequencies within this bandwidth. In this respect, the RF tuner is especially susceptible to misalignment and aging, which invariably causes deterioration of the frequency response. Therefore, when the TV set is used for videotex, it is desirable to bypass the tuner; this can be done in newer types of sets having either integrated videotex/teletext decoders or external sockets for the composite video or RGB signal. See Plates 1 and 2 for the display improvement obtained when the RGB input is used. (RGB stands for Red, Green, and Blue, the component colors of a color TV image. The three signals carrying the color intensities are referred to collectively as the "RGB signal.")

One might be led to think that the resolution of a TV set can be increased by purely electronic means (i.e., by increasing the number of scan lines and widening the bandwidth or by decreasing the scan rate). Unfortunately this is not the case, because the definition is also built into certain fabrication parameters of picture tubes, such as the granularity of the active display surface and the diameter of the electron beam. The luminescent material that converts the energy of the beam into light consists not of a continuous layer, but rather of individual dots of phosphor (or triplets of dots for the three basic colors in color television). The spacing of these dots is normally about 0.7 mm, and this sets an upper limit to the attainable resolution. To go beyond this limit requires use of finer-grain picture tubes, which are more difficult to fabricate. Such CRTs are currently used in high-resolution color graphic systems, but presently are much too expensive for standard TV sets. However, display technology is moving fast, and high-resolution digital TV (1125 lines) may become a commercial reality in a few years (see Fink, 1981).

This will, of course, require a separate transmission medium, such as specially allocated Cable TV frequencies or Direct Broadcast Satellites. High-resolution digital TV cannot be displayed on the conventional television receivers presently installed in homes.

Character Cells

Given their limited resolution, how can TV displays best be utilized to represent textual information? A common approach is to divide up the information area into a grid of rectangular *cells* (see the mosaic option below), each capable of carrying one character symbol with the necessary surrounding space. The basic parameters to be determined are the minimum character dimensions (in pixels) necessary to acommodate all character shapes found in Latin-based alphabets and the spacing requirements between characters and rows. In particular, provision must be made for the following character features:

- upper/lower case
- variations in vertical space occupied (e.g., *k* and *g* may extend upwards and downwards respectively, from a central position)
- accents and diacritics, such as ´, ¨, ^, etc.
- underlining.

The cell size must be sufficient for all these geometric features. A certain margin must be allowed for esthetic appearance, legibility, and receiver bandwidth deterioration.

Experiments and accumulated experience show that the minimum necessary character size is 5 pixels wide and 7 pixels high. This minimum size already compromises legibility and the precision of diacritics (compromise of the latter is no problem, however, in English).

In an extensive study on character design and legibility, Treurniet (1981) presents the following conclusions:

1. Legibility of the basic 5×7 character is greatly increased if descenders (such as in the characters *g* and *y*) are allowed to protrude one line below the body of characters, and if one line is left for diacritics. The basic character size then becomes 5×9 pixels;

2. Inter-row and inter-character space should be about 2 pixels; however, both can be reduced to 1 pixel in order to accommodate 20 rows by 40 characters in NTSC. Each character then occupies a cell of 6×10 pixels (including space).

Several character sets were designed for use in Telidon, based on this study. One of them, the Bronsard 3 set, is shown in detail in Figure 7.2. No character rounding is used.

In other videotex systems, a number of cell sizes in practical use range from 6×8 (in Prestel) to 8×10 (in Télétel). Horizontal and vertical spacing varies between 1 and 3 lines.

Character Rounding

A technique called *character rounding* is employed in some character generators in order to improve the appearance of displayed characters. It makes use of the fact that a row of pixels is composed of two scan lines from alternate fields. Normally the two lines are

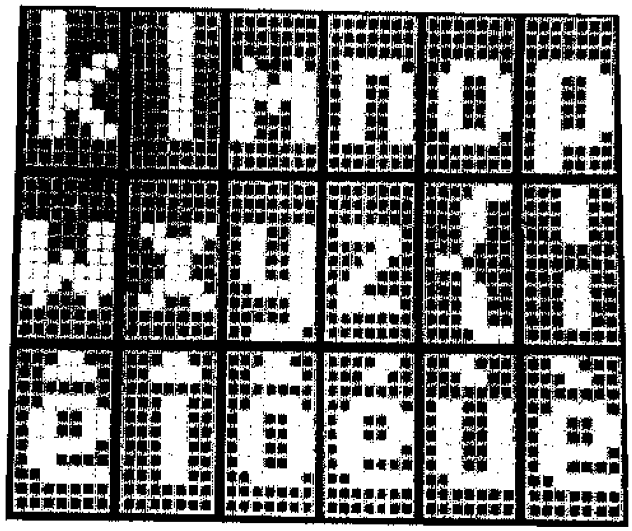


Figure 7.2 Details of Bronsard 3 character set.

identical. With rounding, neighboring pixel lines are compared in a special circuit and, if "diagonal" contours are detected, the intervening scan lines will display dots shifted by one half of a pixel. This principle is illustrated in Figure 7.3. Figures 7.4(a) and 7.4(b) are photographs of the two fields of a character-rounded display to highlight the differences. Figure 7.4(c) shows the combined rounded display.

Number of Characters Per Row

We have seen that this is primarily determined by the bandwidth of the video part of the TV receiver. Figure 7.5 shows approximately how many character cells can be contained in a row with given resolution and character width. The row length of major European and North American systems (U.K. teletext, Prestel, Antiope, Telidon) is usually 40 characters, and occasionally 32 characters in the U.S.

From the point of view of video bandwidth, the European systems might squeeze more than 40 characters into a row. One reason why this does not in fact occur is the fixed format design of the U.K. teletext, in which one data line carries data bits for only

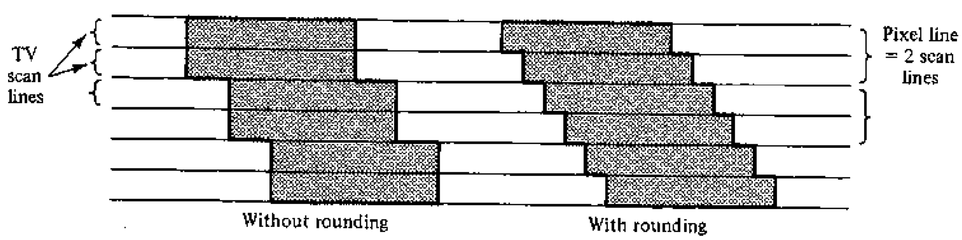
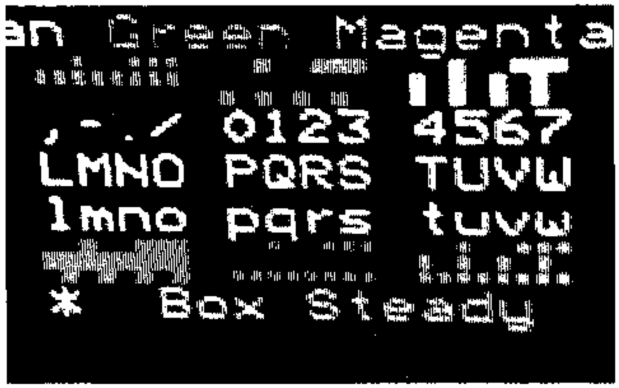
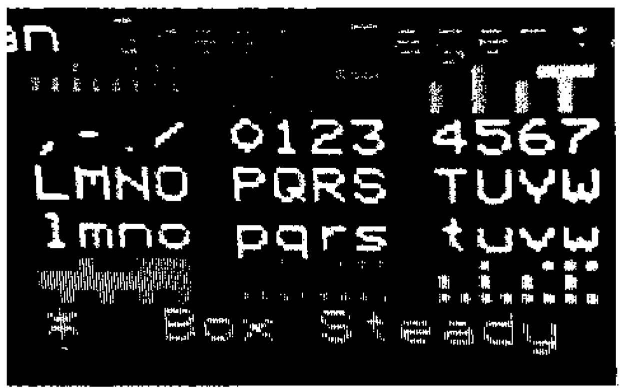


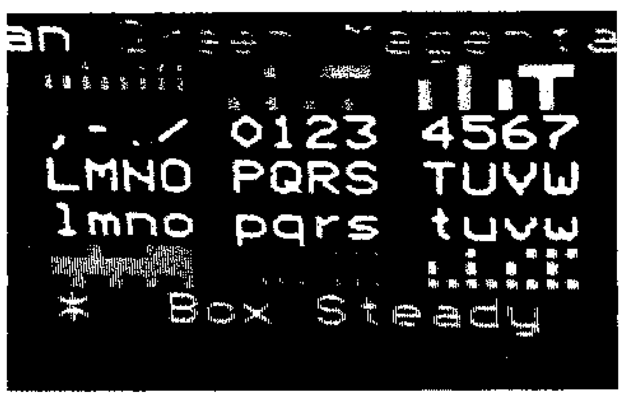
Figure 7.3 Principle of character rounding.



(a)



(b)



(c)

Figure 7.4 (a), (b) Two fields of a character-rounded display; (c) Combined rounded display.

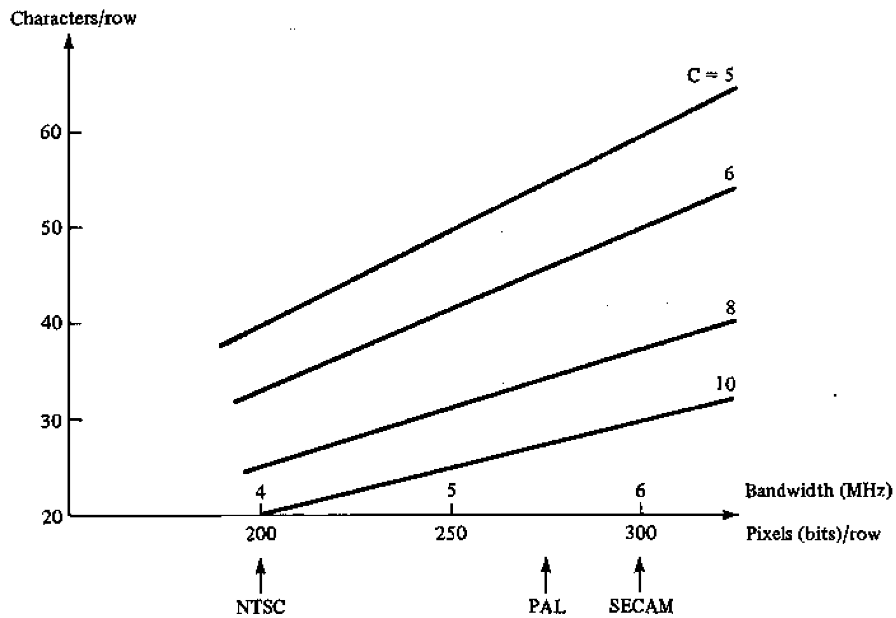


Figure 7.5 Number of displayable characters per row as a function of video bandwidth and character width C (in pixels). Assumptions: 50 μs/scan line, bit rate = bandwidth. Another limiting factor, the picture tube's granularity, is not taken into account.

one row of 40 characters. At eight bits per character and 52 μs available on each data line for transmission, a rate of 6.9 Mb/s results—a value just within the possibilities of the PAL's 5.5 MHz bandwidth. Except for presentation compatibility, this has, of course, no significance in asynchronous teletext like Didon, or in interactive systems.

Number of Rows Per Page

Taking 10 pixels (20 scan lines) as the minimum necessary character height (with space), it is seen that 625-line systems can display about 24–26 lines of characters, 525-line systems only about 20 rows. Since the information content of a page should be maximized, European systems commonly use 24 rows, although the North Americans favor 20 (or 16). This is a major source of incompatibility. It should be noted that with 6 × 8 cell size, one can achieve 24 rows even in NTSC; the resulting quality is acceptable, but the quality of diacritical marks in non-English alphabets would suffer.

Recent results reported by Ciciora (1981b) indicate that there are practical ways to achieve 24 rows even in the 525-line standard. The author's two main arguments, supported by practical demonstrations are

1. Due to decreasing hardware costs, increasing numbers of TV sets will have built-in videotex decoders. Such sets, when switched to videotex mode, will feature

compressed rasters (vertical, or both vertical and horizontal), thus eliminating the loss of scan lines due to overscan. This yields more than sufficient space for 24 rows at 10 lines/row.

2. With set-top decoders in which scan compression is impossible, 24 rows can be displayed by using 6 × 8 cells. These were obtained simply by removing two "judiciously" selected scan lines from a 6 × 10 font, and by applying character rounding. An alternative method, using 8.5 lines per row (8 and 9 in alternate fields), yields very satisfactory displays even of accented characters and with RF connection to the set. A teletext system in Chicago (Field Electronic Publishing) actually uses the 6 × 8-pixel, 24-row format.

The above results are important because they indicate that there are ways to avoid the accumulation of incompatibilities due to the difference between the 625 and 525 line-based systems.

7.2 IMAGE-CODING OPTIONS

We have seen that videotex employs ways of transmission fundamentally different from ordinary television. Transmission is digital; pages to be displayed are generally stored in computer memories, and transmission rates can be quite slow. To transmit the digitized version of a video signal—e.g., using PCM (pulse-code modulation)—one would need a 92 Mb/s channel. At that rate, one frame would require some 3 Mb of storage, about 400 times more than a typical Telidon page. Thus, in order to save memory and bandwidth, new and more economical ways of representing text and image information have had to be developed. These issues are centered around the presentation layer of the OSI reference model.

There are three main coding *options* in use in today's videotex systems:

- mosaic (or character-based)
- geometric
- photographic.

Many documents use the terms alpha-mosaic and alpha-geometric (and even alpha-DRCS). Here we will use the above, less redundant terminology.

Mosaic Coding

This method is taken from conventional VDU technology, with extensions to handle graphic shapes and colors. Three *modes* of mosaic coding can be distinguished: alphanumeric, graphic, and DRCS (Dynamically Redefinable Character Sets). The diplay area in all modes is thought of as a fixed matrix of cells, each cell being a matrix of pixels. A cell can display a *mosaic character*; the shapes and origins of these characters are different in the three options. All cell shapes displayable by a receiving terminal are regrouped in *alphabets* and stored in the terminal's memory as binary dot matrices of the same dimensions as the cell. Since the number of characters stored in the terminal at any given moment (a few hundred) is much less than the number of all possible cell designs (2^{80} with 8×10 cells), coding of mosaics is comparatively very economical.

Alphanumeric alphabets are based on ISO standard 646 and its extensions to special symbols and national alphabets. Its North American version is the familiar ASCII code.

Graphic alphabets are used to create simple pictures and diagrams. Cells are divided up into six elements (three lines by two columns), which gives 64 possible black-and-white cell designs (see Figure 7.6(a)). There are various modalities for these standard mosaic graphics, called *separated*, *smoothed*, and *line graphics*. See Figures 7.6(b), (c) and (d). Various graphic alphabets will also be found in Figure 8.5.

DRCS, a significant improvement of the mosaic option, is realized by permitting the downloading of the dot patterns of a new alphabet from the service computer, instead of permanently storing all displayable characters in the terminal's read-only memory. This permits an application to use any desired symbols (such as Greek letters or the silhouettes of spaceships often seen in animated TV games). Proponents of DRCS sometimes argue that the flexibility inherent in the method makes the geometric option unnecessary. The argument is substantiated by the fact that Captain, the Japanese videotext, relies heavily on a multiple-cell-size DRCS coding scheme.

Once the characters are loaded in the terminal, they can be considered as merely another alphabet. The penalty for this flexibility is of course the transmission overhead due to downloading. It would take about seven seconds to transmit 96 symbols with 8×10 cell size at 1200 b/s. This overhead is prorated over the total number of uses of the downloaded symbols. If they are used many times, the efficiency approaches that of the other mosaic modes; if used only once, the efficiency becomes similar to that of photographic transmission (see below), with the difference that in the basic DRCS technique only the character *shapes* are transmitted (1 bit/dot), whereas in photographic coding every pixel has its color defined independently. It also has to be realized that the representation of DRCS shapes in a database is terminal-dependent; a DRCS coded for 8×10 cell size cannot be directly used with a terminal having 10×12 size cells.

The above two inconveniences of DRCS (absence of color and dependence on cell size) are alleviated through specifications for a CEPT terminal (see Section 8.2) in which provisions are made for 15 possible cell sizes, from 16×24 to 4×5 dots. The number

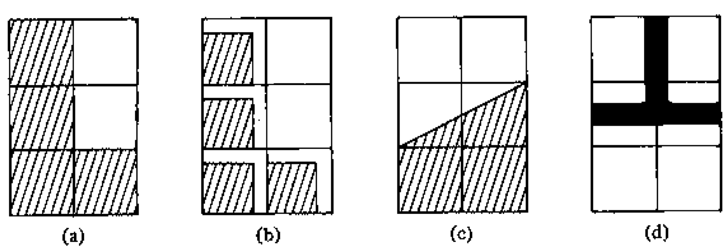


Figure 7.6 Examples of mosaic graphic characters: (a) standard; (b) separated; (c) smoothed; (d) line.

of different dot colors is 1, 2, 4, or 16. Multiple cell sizes and color modes can be used in the same frame.

Geometric Coding

In the geometric option, images to be displayed are described in terms of the "drawing instructions" of an image description language. The instructions correspond to certain geometric elements found in many drawings, namely, points, lines, polygons, and arcs. This approach has its roots in many graphical languages used in computer graphics systems and was adapted to the needs and possibilities of videotex.

Geometric coding can convey relatively complex images more efficiently than can mosaics; the price to pay is in terms of complexity and increased display memory requirements in the terminal, and the time required to draw the image. The confrontation of the two approaches is one of the hot issues being discussed in videotex circles. We will return to this point later.

Photographic Coding

This option is applied generally in cases in which the other methods would fail to represent the objects in sufficient detail. The photographic method is the closest of the three options to digital television. However, only still images are transmitted by this method. Since the transmission speed is lower than that required for direct display at the nominal scan rate, the pictures must first be stored in a display memory.

The main problem with the photographic option is conveying the enormous amount of information that is contained in a (still) color TV frame. Taking 300 pixels per line and 500 lines gives 150,000 pixels; 3 bytes per pixel to encode the luminance and two chrominance signals yields 0.45 MB. This is a great deal to store in the database and in the terminal, even ignoring the fact that at 1200 b/s such a picture would take almost an hour to transmit. Nevertheless, modern coding methods permit reduction of this quantity by a factor of as much as 32 (with only moderate deterioration of picture quality).

There are two large classes of pertinent coding schemes: *point-by-point* coding and *transform* coding. In the first, information on pixels is transmitted in scanning order, much as in (digital) television. Transform coding subjects the source image first to a Fourier, Hadamard, or other transformation, yielding a number of coefficients from which the image can be reconstructed on reception. This requires computational power at both ends, but yields better coding efficiency (in terms of number of bits transmitted) than do point-by-point methods. Note that besides coding efficiency, transform coding has a number of related appealing characteristics, such as insensitivity to noise in transmission and recovery from blurring. (See Rosenfeld, 1977; see also Shlien et al., 1981, for a review and extensive bibliography on these techniques.)

A particularly interesting feature of certain transforms is that they permit the sending of images with "gross information first": that is, the image appears rapidly in its crude outlines, to which subsequent information adds more detail. This "progressive" effect is achieved by suitably ordering the coefficients for transmission (e.g., low-frequency components first). This is clearly of time-saving importance in videotex, because crude approximations may be sufficient to permit users to reject the rest. Figures 7.7(a), 7.7(b), and 7.7(c) show an example obtained by the "singular value decomposition" method (Shlien et al., 1981). Note, however, that progressive transmission can be achieved by other means, too. For example, Knowlton's method (1980) for the coding of digitized gray scale images treats the image in a *hierarchical* fashion. First, it calls for transmission of the overall average intensity (gray) value for the entire image. Then the image is cut into two halves, and the average for one half is transmitted. The value

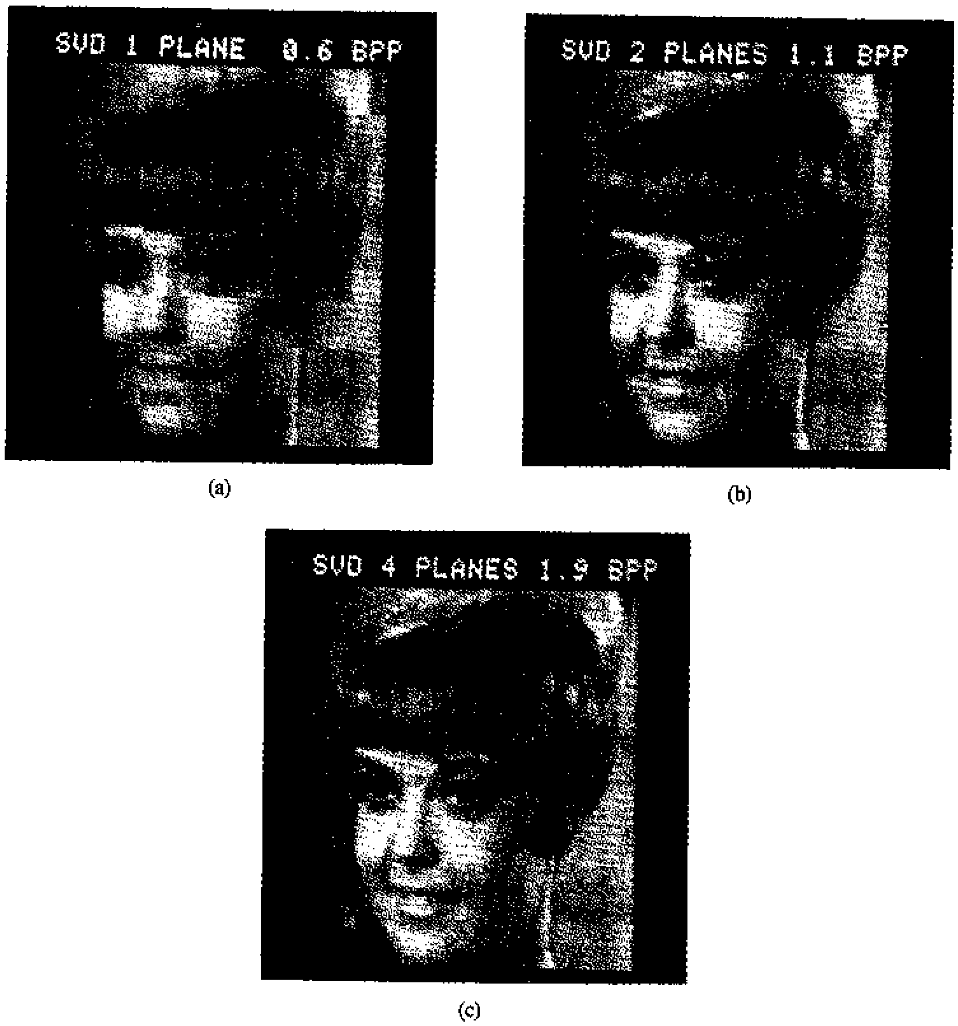


Figure 7.7 (a), (b), (c) Sequence of progressive reconstructions of a picture.

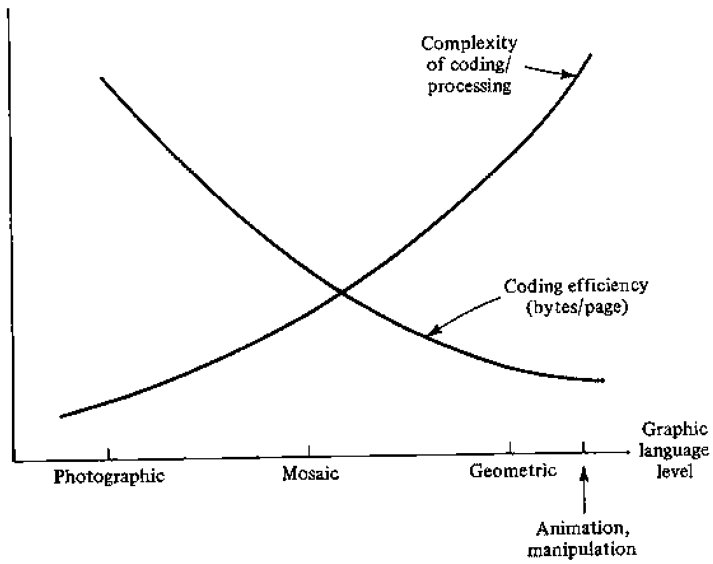


Figure 7.8 Efficiency and complexity of image coding options (qualitative).

for the other half can be calculated from this and the overall average. The process of halving is repeated until full detail is obtained.

Whatever method is used for encoding the image, the resulting bit stream may be *compressed* for storage and/or transmission by well-known techniques, such as run-length or Huffman coding.

With the exception of mosaics, none of the three options is sufficient as a unique means of coding in any practical system. First-generation systems (such as Prestel and Antiope) are based on mosaic coding; later versions include DRCS. On the other hand, newer standards (PLPS, CEPT) rely on all options and provide unified coding schemes and tables for this purpose. The following sections of this chapter describe such coding schemes in detail, stating their functional capabilities and impact on terminal design. Figure 7.8 relates qualitatively the mosaic, geometric, and photographic options in terms of language level, complexity of encoding/decoding, and required display storage capacity per typical page. Plates 3–8 are photographs of sample displays created by various coding options.

Note that geometric graphics are fundamentally different from mosaic or photographic graphics. In the latter two cases, there is a relationship between bytes of code and picture area that is direct and independent of picture content. The amount of code required for geometric graphics is related to image complexity. For example, a contour map of the state of Colorado would require many fewer bytes than would a map of the Great Lakes.

7.3 CODE TABLES AND CODE EXTENSION

Structure of Code Tables

Mosaic characters come in sets, called *alphabets*, which are defined in *code tables*. Certain alphabets have been standardized and used for text transmission for quite some time; others (e.g., mosaic graphic sets) are specifically videotex-oriented. In effect, not only alphabets but also the structure of code tables are standardized to a certain extent. Figure 7.9 shows the now generally accepted ISO 7-bit table with standard ASCII characters.

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1	Column / Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	TC7 (DLE)	SP	0	@	P	`	p
0	0	0	1	1	TC1 (SON)	DC1	!	1	A	Q	a	q
0	0	1	0	2	TC2 (STX)	DC2	"	2	B	R	b	r
0	0	1	1	3	TC3 (ETX)	DC3	#	3	C	S	c	s
0	1	0	0	4	TC4 (EOT)	DC4	$	4	D	T	d	t
0	1	0	1	5	TC5 (ENQ)	TC8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	TC6 (ACK)	TC9 (STN)	&	6	F	V	f	v
0	1	1	1	7	BEL	TC10 (ETB)	'	7	G	W	g	w
1	0	0	0	8	FE0 APB (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	FE APF (HT)	SS2 (EM)	)	9	I	Y	i	y
1	0	1	0	10	FE2 APD (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11	FE3 APU (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12	FE4 CS (FF)	IS (FS)	,	<	L	\	l	|
1	1	0	1	13	FE5 APR (CR)	SS3 IS (GS)	–	=	M	]	m	}
1	1	1	0	14	SO	IS APH (RS)	.	>	N	^	n	~
1	1	1	1	15	SI	IS (US)	/	?	O	_	o	DEL

Figure 7.9 ISO 646-based code table with C0 control set (col. 0–1) and G0 graphic set (ASCII).

The ASCII alphabet is the North American version of the international standard alphabet (see ISO 646, 1982). National versions of ISO 646 may differ in 13 character positions shown shaded in Figure 7.9. Seven bits are used because storage and transmission are universally performed in bytes, with one bit reserved for parity. The table defines the meaning of each of the 128 codewords (or codes) of the form $b_7, b_6, \ldots, b_1$. It is organized into eight columns and 16 rows. The three most significant bits (b_7 through b_5) specify the column, the other four bits (b_4 through b_1) the row number. A position in the table will be designated by a shorthand notation—e.g., 4/9 is the codeword at the intersection of the fourth column and ninth row (100 1001). Its signification in ASCII is "I." Bits are usually transmitted in the order $b_1, b_2, \ldots, b_7, p$; (p is the parity bit).

The code table has two parts. Columns 2 to 7 contain 96 *displayable* characters; the first two columns regroup 32 *control* codes. The two groups of codewords are named G-set (graphic) and C-set respectively. Control codes control the transmission and formatting of displayable characters. Transmission control codes (crosshatched in Figure 7.9) are used at lower levels of the OSI hierarchy and should not normally be used at the presentation level. Formatting codes will be explained later.

Alphabets other than ASCII can be defined by means of G-sets in alternative tables. Examples are the so-called supplementary alphanumeric set, and various sets of mosaic graphic characters. The former is a complement to the basic ASCII table, defining some special symbols and diacritical signs to be combined with other characters (see Figure 7.10).

It is important to note that code tables generally define the meaning, but not the precise dot pattern (font), of the characters. This (as well as the cell dimensions) is an implementation detail local to the terminal, and there is no need for standardization, except in the case of DRCS, for which the downloaded dot patterns must match the cell size used.

Code Extension Standard

All information to be displayed on videotex terminals must be coded for transmission (and storage) as a sequence of 7- or 8-bit characters. This is true regardless of the speed, communication medium, and option.

We have seen how the meaning of these characters is defined in code tables. The basic ASCII table is sufficient for simple text transmission applications, but certainly not for the full range of options, alphabets, and attributes inherent in videotex. How can all these additional data be incorporated into a reasonable coding scheme? A brute-force solution would, of course, be to use a new code structure with sufficient number of bits, say 14 (two 7-bit codewords). But in addition to being wasteful, such a scheme would be incompatible with other ASCII-based services such as telex, database retrieval, and word processing.

The ISO devised another solution, called *code extension* (described in ISO 2022, 1982); note that it is not a videotex-specific procedure. It is based on the above-described code table, and extends the capability of 7-bit codes by *temporarily changing their*

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1	Column / Row	0	1	2	3	4	5	6	7
0	0	0	0	0				°		—	Ω	ĸ
0	0	0	1	1			¡	±	`	¹	Æ	æ
0	0	1	0	2			¢	²	´	®	Đ	đ
0	0	1	1	3			£	³	^	©	ª	ð
0	1	0	0	4			$	×	~	™	Ħ	ħ
0	1	0	1	5			¥	µ	¯	♪		ı
0	1	1	0	6			#	¶	˘		Ĳ	ĳ
0	1	1	1	7			§	•	˙		Ŀ	ŀ
1	0	0	0	8			¤	÷	¨		Ł	ł
1	0	0	1	9			‘	’			Ø	ø
1	0	1	0	10			“	”	˚		Œ	œ
1	0	1	1	11			«	»	¸		º	ß
1	1	0	0	12			←	¼		⅛	Þ	þ
1	1	0	1	13			↑	½	˝	⅜	Ŧ	ŧ
1	1	1	0	14			→	¾	˛	⅝	Ŋ	ŋ
1	1	1	1	15			↓	¿	ˇ	⅞	ŉ	

Figure 7.10 Supplementary graphic set.

meaning. The principle is similar to the method by which computer systems address very large virtual memories (e.g., 16 million words) with much less address bits than would normally be required by the address space (say, 12 instead of 24 bits). The trick is to recognize that although the total number of required code meanings (virtual addresses) is very large, typically only limited subsets are used during relatively long periods of time. Therefore it is more economical (if not done too frequently) to change the currently valid subsets (such as memory segments or alphabets) by means of a few additional codes, and then use fewer bits for further specification within the subset.

This code extension scheme for a 7-bit environment is shown in Figure 7.11. The terminal is supposed to interpret all received codewords with reference to an *in-use* (current) 128-character table. The in-use C-set and the in-use G-set are subsets of this table. Certain characters (or sequences) from the in-use C-set are used to replace the in-use table or its individual elements. This replacement can be *temporary* (extending for one character only) or *locking* (extending until further replacement is invoked by a control sequence).

There are four G-sets, G0, G1, G2, and G3, which can be invoked as the in-use set by single control characters. G0 and G1 can be invoked in a locking manner by SI (Shift In, 0/15) and SO (Shift Out, 0/14), respectively. G2 and G3 are invoked temporarily by control characters SS2 and SS3. (The 1982 version of ISO 2022 also provides for locking invocation of G2 and G3.)

Besides these four "immediately" invocable G-sets, a whole repertory of further

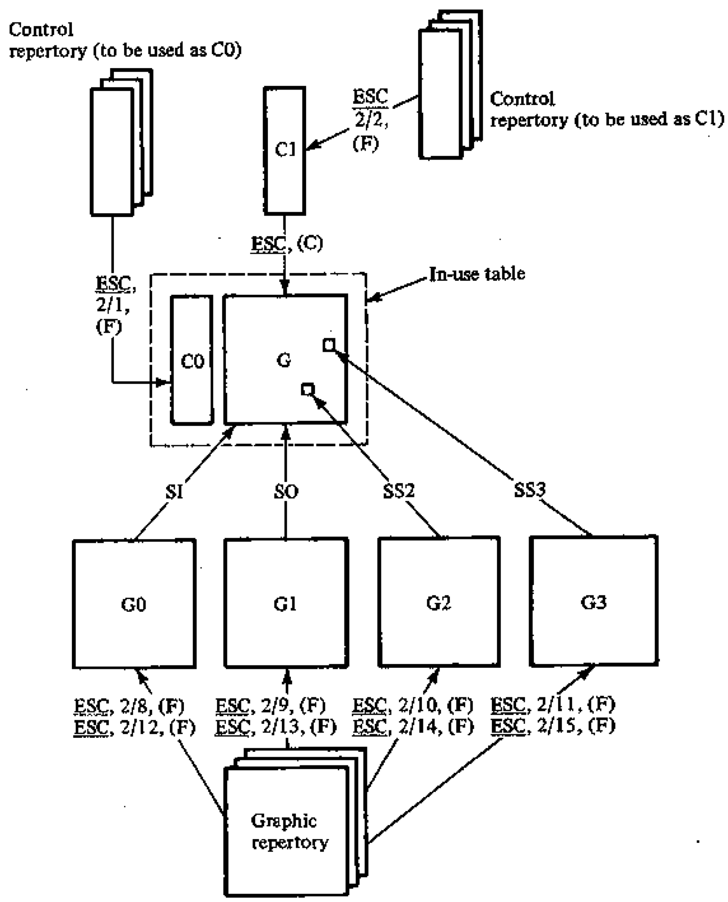


Figure 7.11 Principle of code extension.

G-sets may be defined; any of these can replace any one of the G0–G3 sets through three-character control sequences also shown n Figure 7.11. The graphic repertories and the default assignments of G0–G3 can vary between applications. In most videotex systems G0 and C0 are as shown in Figure 7.9. G1 designates mosaic or geometric graphic sets, and G2 the supplementary graphic set. Alphabets used in some particular systems can be found in Chapter 8. The above control sequences always begin with ESC (1/11), followed by a character defining which of the four G-sets is to be replaced; the last character, (F), defines a set from the library (graphic repertory). This character is subject to a gradual standardization by means of a registration procedure by ISO. The intention is to assign unique characters for every future G-set. As a result, a very large number of interpretations can be given to the codewords.

The extension of control sets is handled in a similar fashion. However, the in-use C-set (called C0) is practically never replaced in a locking manner (although provision for this is made through the ESC, 2/1, (F) sequence). All extensions are invoked temporarily from a C1-set (by ESC, followed by a control character, (C), from C1). Codewords of C1-sets are placed in columns 4 and 5 of the code table in order to avoid confusion with the C0-set. Position 5/11 in some C1-sets is reserved for CSI (Control Sequence Introducer) to be used for purposes analogous to ESC in C0 (as described in a related standard, ISO 6429, 1982).

The C1 set can be replaced by any set from an extended control repertory by means of the sequence ESC, 2/2, (F), where (F) designates the selected set. The character (F) is subject to gradual standardization by ISO.

The C0- and G0-sets in the ASCII code table are the default current sets. ISO 2022 does not prescribe the ways in which the additional graphic and control sets should be used. This is done in other, videotex-related standards and systems, as discussed in this chapter and Chapter 8.

The ISO 2022 standard also covers the extension of 8-bit codes. The method is similar to the one just described, except that the in-use table has 256 positions permitting two G-sets and both C-sets to be used simultaneously. The eighth bit is used to distinguish between the two sets. There are minor variations in the ways in which this code extension scheme is applied in various videotex coding systems.

Control Set Functions

C0 functions (except those reserved for transmission control (see Figure 7.9) are standardized for videotex in CCITT Recommendation S.100 (1980). Since C0 is also used by other related text transmission services with which videotex should be compatible to the maximum possible extent, the interpretation of these controls is essentially the same as elsewhere. They include the following (note that "active position" means the cursor):

- format effector controls, such as
  APB (active position backwards)
  APF (action position forwards)

APD (active position down: line feed)
APU (active position up)
APR (active position return to the beginning of row: carriage return)
APH (active position home)
CS (clear screen)
SP (space)
CAN (cancel row, interrupt)
US (cursor positioning, reset of some attributes)

- code extension controls, such as
  ESC (to change the meaning of subsequent character(s))
  SO, SI, SS2, SS3 (to invoke graphic sets G0–G3).

Control functions in C1-sets are videotex-specific and their detailed assignment differs in various systems. In the European approach (see CEPT, 1981) two C1-sets are used for the control of parallel and serial display functions and attributes. North Americans do not distinguish explicitly between the two kinds of attributes. Their standard PLPS (1982) employs a single C1-set fulfilling some (but not all) attribute controls, cursor controls (steady/flash), text editing, formatting, etc.

7.4 CODING OF DISPLAY ATTRIBUTES

Unlike the display of simple alphanumeric text on VDUs, videotex information can be displayed with a number of color, shape, and other variations. These variations, or *attributes*, include among others

- color of the characters
- color of background
- size of characters (double height, double width, or both)
- blinking or steady characters
- characters underlined or not
- display field concealed or revealed
- display field protected or not (writeable by user)
- display intensity.

Further attributes, such as orientation of objects and texture, are applicable in the geometric option. The above list contains only typical mosaic attributes and is by no means exhaustive. Note that the question of what an "attribute" is, is relative; an object (mosaic character) can be seen as nothing but a collection of attributes, including shape. Attributes affect not only the appearance of individual shapes (mosaic characters or geometric figures) but also the background against which the shapes are displayed. The display process is generally controlled by control characters from C-sets; some of these characters control attributes and their modalities. The span of control may extend to

- individual characters or shapes (as to color, size, whether underlined, etc.)
- an entire row, or field of the display
- the full screen or information area.

It is convenient to imagine the visible display area as composed of three layers. On the "deepest" level is a standard TV picture on the full screen. The next level is the information area background (see Figure 7.1), followed on the "top" by the information area foreground (display shapes). Different attributes can control the three layers independently. Thus in order to see a TV program, one must set all layers to transparent state. Similarly, "boxes" (e.g., with text) that overlay the TV image can be created.

Handling of Control Codes in the Terminal

The manner in which attributes are represented and manipulated is closely related to the internal architecture of the terminal controller. A component that must be present in any terminal design is the *display memory* (DM). Logically DM appears under the same format as the display area—that is, as a matrix of, for example, 40 × 24 cells. Its basic function is to accumulate and hold all data for the currently displayed page; it can also be thought of as a buffer, necessary to smooth the disparity between the data reception and display rates. The second important component is the display generator, which serves to create the RGB signals that drive the display tube. Refer to Chapter 9 for more details.

While graphic (displayable) characters as received by the terminal are immediately loaded into DM, characters from the C0 and C1 sets can be active ("executed") at two different times: on reception or on display generation. C0 codes (e.g., cursor controls, ESC sequences) and attribute controls are examples of the two cases, respectively. The former serve to control and modify the process of loading DM; the latter control characters must be stored in the terminal for use by the display generator.

The following discussion is focused on an issue that has raised heated controversies in the past: the treatment of attributes in the mosaic option. The essence of the problem is that at the moment of display generation, the complete set of attribute values must be known for each character being displayed. The problem is to determine where these attribute bits would come from.

Fortunately, attributes do not have to be coded and transmitted explicitly for every character. Advantage can be taken of the fact that attribute values typically do not change very often; therefore one transmits (as control characters) *only the new attribute values* when changes occur. This, along with the use of default values, can bring down to a reasonable level the overhead due to attributes.

For historical reasons, two approaches have been worked out for the handling and storing of attributes in DM, as illustrated in Figure 7.12. In the first approach (Figure

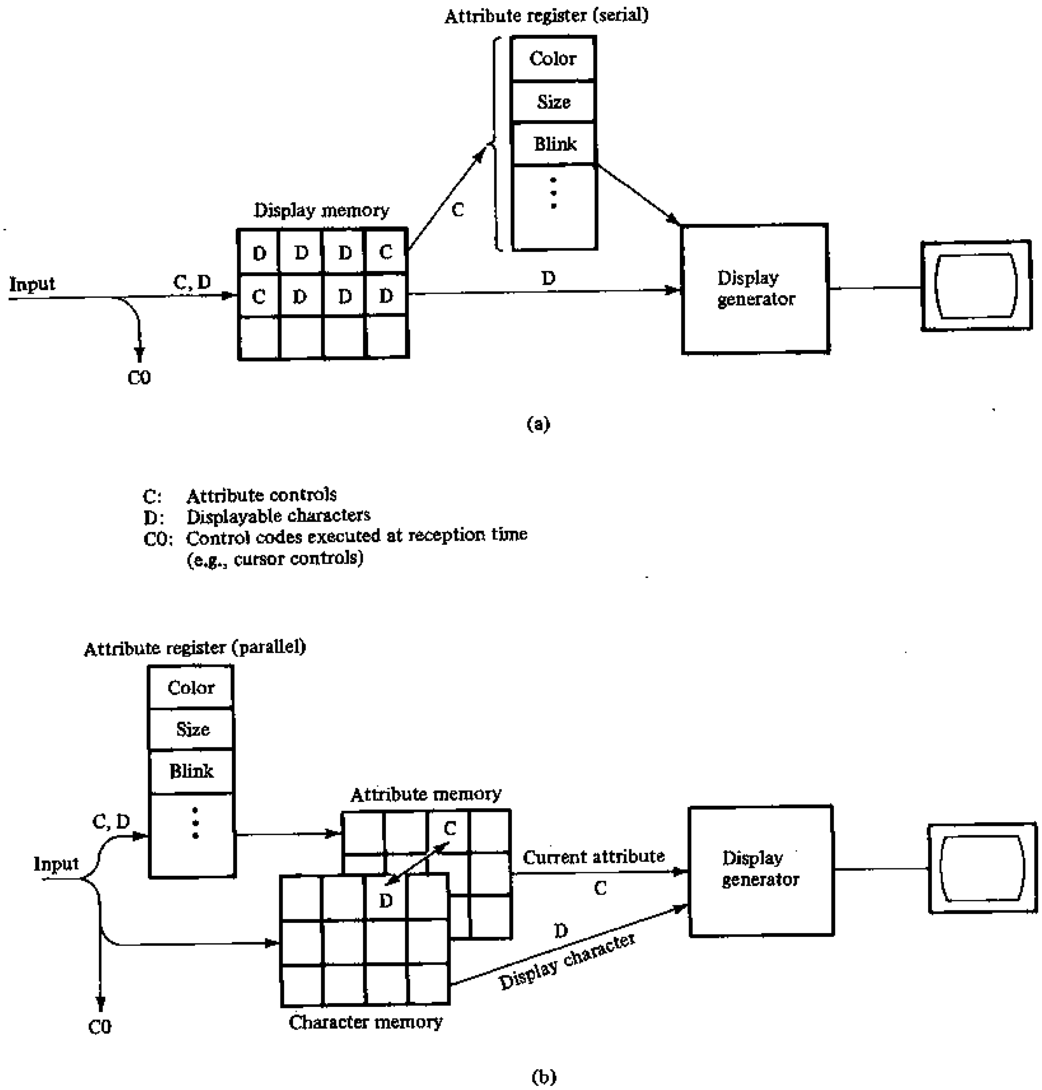


Figure 7.12 (a) Conceptual handling of serial attributes. Display memory holds *D* and *C* characters; (b) conceptual handling of parallel attributes. Attribute register moves with cursor.

7.12(a)) DM has only eight bits per cell, the minimum necessary to hold the coded form of a character; consequently attribute controls (*C*) must be mixed with displayable characters (*D*). On display, DM is read line by line and the control characters are (conceptually) fed to an "attribute register" which contains the current values of all attributes and supplies them to the generator. (The attribute register is usually incorporated into the generator.) The net effect of all this is that an attribute value will propagate along a line until a new value is defined or the end of the line is reached; at this point all attributes are reset to default values. This method of treating the attributes is said to be *serial*. It was originated in the U.K. teletext and was carried over to Prestel.

In the second solution (Figure 7.12(b)) DM has two components—the character memory CM and the attribute memory AM that explicitly stores the attributes for each cell (typically 8–16 bits/cell). Since attributes are transmitted only on change, the latest values received are again conceptually fed into an attribute register and written to AM in parallel with the contents of CM as the cursor proceeds across the lines. This is the source of the term *parallel* attribute handling. In this mode the attribute register is considered to be attached to the cursor and writes its contents whenever a displayable character is placed into CM. Parallel attributes were introduced by Antiope.

It is crucial to note that insofar as the DM is filled sequentially line by line (no explicit cursor controls are used to move it arbitrarily across DM), the serial and parallel modes give practically the same results (the difference is that a serial attribute requires an extra character space). This is because the serial order of scanning DM by the display generator imitates exactly the cursor movement on writing DM, and thus the sequence of readings from AM. This situation is radically altered with random movements of the cursor; the serial and parallel modes then behave quite differently. It took many years of discussion and controversy until the discovery was made that the two modes could be reconciled into a single unified system—the CEPT standard.

Serial Attributes in Fixed-Format Teletext

In a serial attribute system, such as that illustrated in Figure 7.12(a), some cells must be occupied by attribute control characters that are not themselves displayable. As a default, either a blank (current background color) or a repetition of the last non-control character may be displayed in these cells. The blank is suitable for text applications in which attributes, such as underlining, typically change between (but not in the middle of) words. In mosaic graphics, on the other hand, a blank occurring between two adjacent areas of different color might be disturbing; in such cases the previous mosaic character should be repeated. This is achieved by a "hold graphics" control. The main problem with this approach is that attributes are *spacing*,—that is, they advance the cursor. Therefore, certain display patterns are impossible to create. For example, two adjacent letters cannot have different color. Figure 7.13 is a simple example of the use of serial attributes.

It is often claimed that these restrictions are not serious and that they are outweighed by the advantages of the system, namely its simplicity and minimized memory requirements. The simplicity stems from the fact that on reception, both control and displayable characters receive the same treatment: they are sequentially loaded into the memory. Simplicity becomes even more pronounced when data are arriving through synchronous VBI transmission (in which one TV data line carries the equivalent of one display line), as sketched in Figure 7.14. Note that this scheme does not permit random

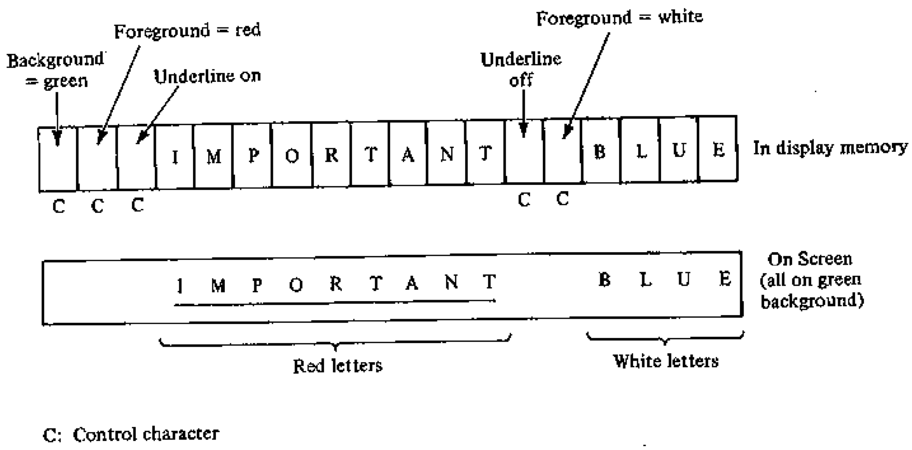


Figure 7.13 The effect of serial attributes.

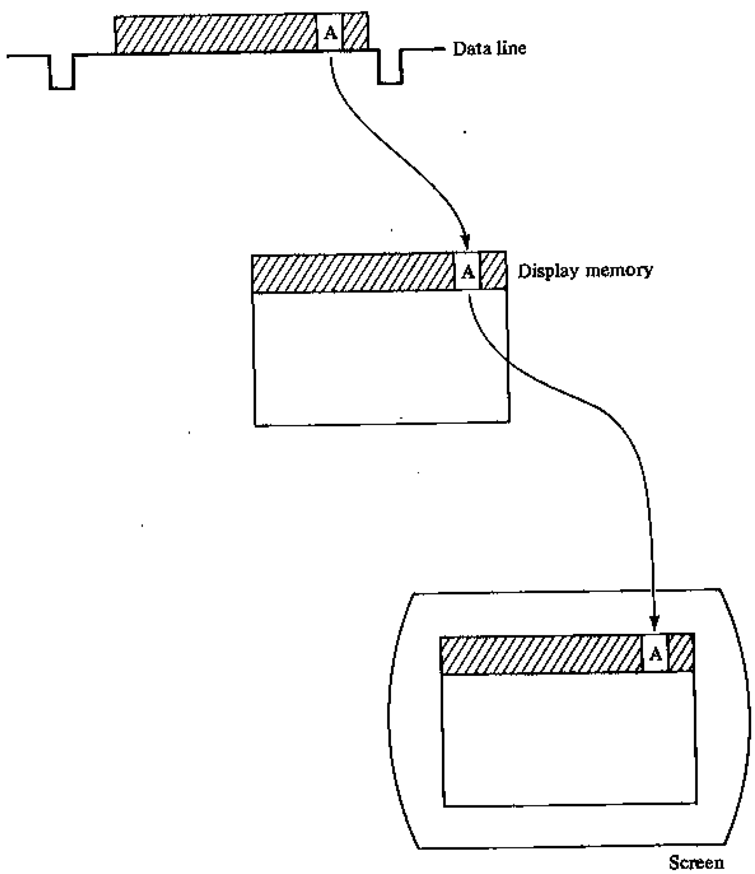


Figure 7.14 Fixed relation between character positions on a data line, display memory, and screen in synchronous teletext.

cursor movements. However, this is an inherent quality of synchronous transmission and not of serial attributes; Prestel, for example, can use cursor controls.

Although serial attributes permit neat terminal designs that do not require a microprocessor and that use minimal memory, these advantages tend to erode with decreasing hardware costs and with the tendency towards intelligent and multifunctional terminals. This is *not* to say that serial attributes are inferior to parallel ones, however.

Applicability of Attributes

Roughly speaking, parallel attributes are advantageous for drawing graphical shapes with mosaic graphics involving cursor movements in directions other than horizontal. For example, a diagonal line over blue background and consisting of alternating groups of three red and three yellow squares, such as in Figure 7.15(a), would be drawn by using the following sequence of symbolic codes (see Section 7.3 for meaning of the codes):

- CS
- set background color (full screen) = blue
- APH
- set foreground color = red
- WMC (Write Mosaic Character)
- APD
- APF
- WMC
- APD
- APF
- WMC
- set foreground color = yellow
- APD
- APF
- WMC, etc.

To do the same with serial attributes would require setting the foreground color before every WMC code.

Serial attributes enable one to pre-set the attribute structure of a page before the characters are actually written. This can be applied, for example, to set up a box on the screen into which text will be loaded later. Subsequently the colors of foreground and background may be changed, or the box may be made to flash *without* rewriting its content, which cannot be done with parallel attributes. This is illustrated in the following

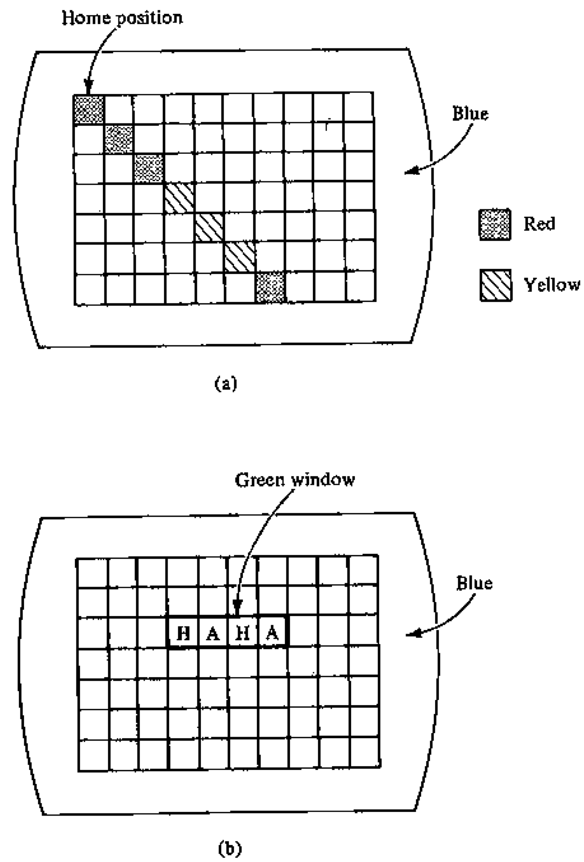


Figure 7.15 (a) Use of parallel attributes; (b) use of serial attributes.

simple example (see Figure 7.15(b)). On a blue screen, the word "HAHA" has to be written in a green window. Here is the resulting code:

- CS
- set background color (full screen) = blue
- APH
- APD
- APD
- set forground color = red (spacing)
- set background color = green (spacing) [red and green colors are now set up to the end of line, remaining invisible until displayable characters are written]
- APF
- write H

- write A
- write H
- write A [HAHA displayed in red letters in green box]
- pause
- APR
- set foreground color = black
- set flashing [HAHA displayed in black, flashing in green box].

Note that other, more general, ways of predefining the attribute structure can be imagined than propagating attributes line-wise (for example, by use of cursor movements). The important distinction to make is to specify whether AM can or cannot be loaded independently of loading DM.

Reconciling Serial and Parallel Attributes

For some time it was thought that the two approaches to attribute handling were entirely incompatible, and that the only way to reconcile them was to accept both as separate options in a standard; this was done in CCITT Recommendation S.100. However, deeper analysis and recent efforts to produce a unified European videotex standard have demonstrated that the two schemes are less different than one might have gathered, given the intensity of discussions among their proponents.

The key element of the new insight is the shift of emphasis from whether or not a separate AM (attribute memory) should exist in the terminal, to the way the AM is loaded with attribute values. It has been recognized that the additional cost of 1 or 2 kB of AM is becoming of little significance; the main problem is how to avoid explicit transmission of all attributes with each character. To recapitulate, this can be done by extending the span of each attribute control *until the next change* of that attribute is received. However, there are two modes for doing this.

1. The cursor (active position) is imagined to carry a register with all current attribute values. Whenever a displayable character (including space) arrives, it is deposited into the CM and the attribute values are copied from the cursor into the identical position of AM. Attributes are *not* copied during cursor displacements (APF, APH, etc.). This is precisely what is happening when parallel attribute controls are used.

2. Attribute controls are imagined to write a *marker* at the AM address pointed to by the cursor. The new attribute value is copied (row-wise) into all subsequent positions in AM until a previous marker (for that attribute) or the end of the row is encountered. Such action is invoked by serial attribute control characters. Displayable characters are written without affecting attributes. Note that, granted a separate AM, serial attributes need no longer necessarily be spacing (they are spacing in the CEPT standard for compatibility with Prestel).

A short reflection shows that there is no reason why both kinds of attribute controls could not be used in building up a display frame. All that matters is to load appropriately the attribute memory prior to the display process. Since a given location in AM can be addressed several times and by both serial and parallel attribute controls (contained in two separate C1-sets), a natural rule has been adopted, that says in essence that the *latest* attribute control takes precedence, whether received in serial or parallel mode. This principle is the basis of the so-called "time-dependent unified alphamosaic system" (TDUAM) used in the CEPT European presentation standard (see Section 8.2). There the terminal's display memory is visualized as composed of three 40-by-24 planes: character memory, attribute memory, and marker memory. Smaller memories for full screen and line attributes are also included.

In the marker memory (MM) every attribute (or other display function) may be marked by a flag. A flag in a given position indicates a change in the value of that attribute, a new value starting at the flag position and continuing to the right.

The rules for TDUAM can be summarized as follows:

1. The terminal at any time is in serial or parallel mode, depending on the most recently invoked C1-set.
2. All attribute values, regardless of the mode in which they are received, are stored in AM.
3. A marker is written in MM whenever a newly stored attribute value is different from the value in the preceding (left adjacent) position.
4. Writing of attributes and characters in either serial or parallel mode is as described above (page 119).
5. Markers cannot exist between two positions that have the same attribute value (e.g., as a consequence of writing a series of characters in parallel mode, all overwritten markers are deleted).
6. Full row attributes overwrite all attributes and delete all markers.

The working of these rules is illustrated by the example in Figure 7.16.

Stack Model

The above conceptual model of a display memory can be economically implemented by a *stack* architecture, as described by Childs (1981). The architecture is centered around the idea of storing, in a stack of the same size as CM, only the attribute *changes* that occur while scanning CM. Preliminary analysis shows that this capacity (1 kB) is largely sufficient for all changes occurring even in the most complicated frames. The eighth bit in each position of CM is reserved to indicate whether there is any attribute change at that character position. If there is one, access is made to the stack for details on the required changes. This information can include any number of bytes; a special indicator delimits the last byte describing a given change. (Note that a marker memory may still be needed to build up correctly the contents of the stack!) Figure 7.17(a) shows schematically this memory architecture, including memories for full screen and line attributes; the stack organization is illustrated in Figure 7.17(b). Since the order of characters in the

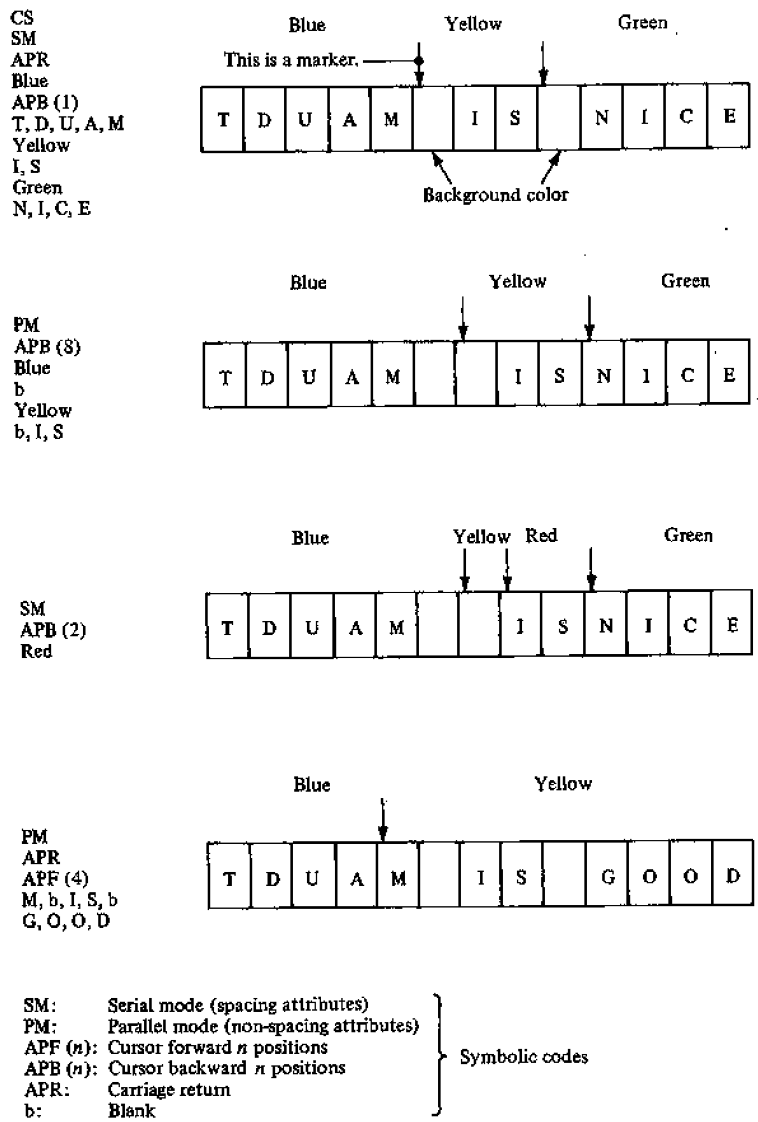


Figure 7.16 Example of display under TDUAM (only foreground colors manipulated).

generally different from the order in which they are stored in DM, additional processing power (a microprocessor) and working memory are necessary in the decoder. For example, one must keep track of the addresses of all serial attribute markers and the current position of the cursor, in order to be able to insert (rapidly) a new attribute change into the correct place in the stack. If this processing cannot keep up with the rate of incoming data, extensive buffering may become necessary.

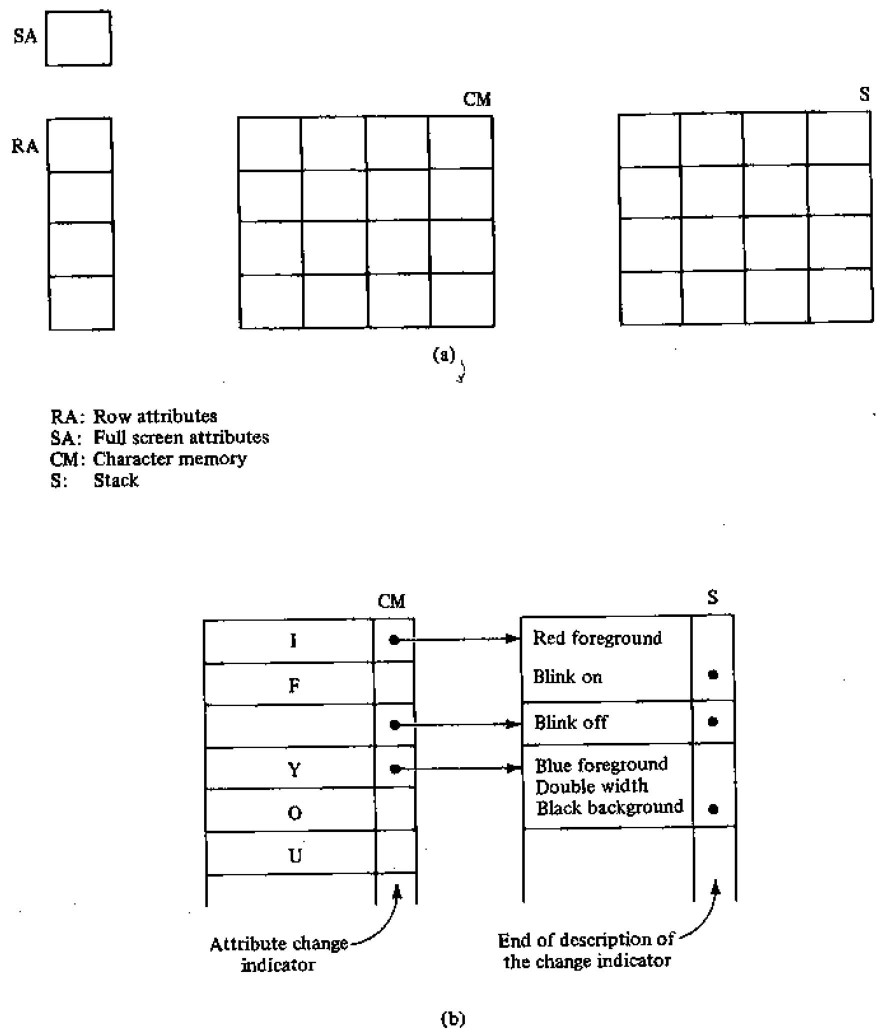


Figure 7.17 (a) Stack architecture of display memory; (b) stack use example.

Note that parallel attributes are inherent in geometric and photographic coding. Individual pixels have no shape and are defined only by their attributes in the display memory. In geometric coding attributes are set either by explicit controls (until the next change) or by drawing instruction opcode variations (valid for the instruction in question).

Coding of Colors and Gray Scales

The method used to represent colors in videotex derives from the way in which the colors are generated on the screen of a color TV set. A color picture is the superposition of three separate pictures, each in one of the basic colors: Red, Green, and Blue. These

Basic color:	Red	Green	Blue
Black	0	0	0
Blue	0	0	1
Green	0	1	0
Red	1	0	0
Cyan	0	1	1
Magenta	1	0	1
Yellow	1	1	0
White	1	1	1

0: Off
1: On

Figure 7.18 Combinations of the three basic colors: red, green, and blue.

colors have the property that any other (composite) color can be obtained by superposing the basic colors at appropriate intensities. The picture tube has three electron guns producing beams that can be modulated independently. The beams create the component pictures by hitting phosphor dots of appropriate color on the screen.

In all major videotex systems in use, colors are coded with three bits, each causing a basic color to be turned fully on (1) or off (0). (Sometimes two intensity levels are available, the brighter used to highlight selected parts of the image.) The resulting eight combinations are shown in Figure 7.18. Displays generated by videotex are generally considered to cover an underlying TV image. To uncover parts of this image, a ninth "color," or mode, called *transparent*, can be used for applications such as captioning and newsflashes. Transparent mode is set by an independent control code. While this repertory of colors is sufficient for mosaic and perhaps geometric displays, it becomes too crude if used with photographic displays of, say, portraits or landscapes.

A fuller range of colors can be obtained by permitting more intensity levels to represent each basic color. Four bits per color will result in a color resolution practically equivalent to TV pictures. Newer systems have provisions for such extended color sets, in the form of *color tables*. This makes possible any degree of color definition without the need to transmit all necessary bits when switching between colors. A number of predefined color bit configurations is loaded into the table by special commands; after that is done, only the row number of the desired color has to be transmitted.

Color impression depends roughly on the ratio of the basic RGB intensities; brightness is given by their absolute value. Thus it is easy to obtain different levels of gray as points on the continuum between black and white. This is achieved by keeping the three color intensities equal and changing their absolute values simultaneously. An independent control bit is used to set color or monochrome mode; in the latter case the RGB bits in Figure 7.18 are interpreted as gray-level information.

8
Presentation Coding in Practice

8.1 MAJOR PRESENTATION SYSTEMS: AN OVERVIEW

Mosaic Systems: Prestel and Antiope

The first major presentation systems to be used and standardized on a national scale were Prestel in the U.K. and Antiope in France. Pilot trials with both systems began in 1978; public service started in 1980 in England, and in 1982 in France.

Although numerous theoretical and experimental enhancements, including DRCS and the photographic option, were reported (see Lambert, 1980; Clarke, 1980), the hard core of both systems is made up of three mosaic alphabets—G0 (standard alphanumeric), G1 (mosaic graphic), and G2 (supplementary alphanumeric)—along with the C0 and one C1-set (display and attribute controls). The G0-and G2-sets show only minor variations in the two systems. These three graphic sets are shown in Figures 7.9, 8.5(a), and 7.10, respectively. Columns 4 and 5 in Figure 8.5(a) apply only to Prestel.

The main differences between the two systems are:

- serial (and spacing) in Prestel versus parallel attributes in Antiope, resulting in 8 versus 16 bits/character in display memories
- 24 rows in Prestel versus (25 of which only 24 are used) in Antiope

- minor differences in some character attributes (e.g., double-height characters extend upwards in Prestel, downwards in Antiope)
- $G1$-set in Prestel invoked through a $C1$ control and not by SO as specified in ISO 2022 (carried over to the CEPT standard for compatibility).

Experience shows that the basic mosaic system seems to work well in most applications.

Picture Prestel

It has been recognized by the British Telecom that some applications require better resolution than it is possible to obtain with mosaic graphics. As a result, an experimental service feature called Picture Prestel has been developed, permitting the insertion of photographically coded images on the screen. Due to constraints on transmission time and storage capacities (both in the database and terminal), the maximum size of photographic pictures has been chosen to be one ninth of the display area. Even so, and with advanced coding methods, at 1200 b/s it takes about one minute to transmit an image in full detail. In this context it is interesting to mention British plans to upgrade their telephone network to provide ISDN-like 64-kb/s channels to the user's premises. This will make practical the transmission of full still TV frames, as well as image communication between users.

It is a distinctive disadvantage of mosaic-based systems that extra display memory must be installed in the terminals (24 kB in Picture Prestel); in effect, there is a whole subsystem dedicated to the photographic option. On the other hand, in systems with bit plane memories, the display memory is already paid for. The cost increment for a photographic option with the resolution of the existing bit plane is the cost of the processing circuits and software. This can be negligible for the simpler, more straightforward approaches.

In Picture Prestel (which is still in the experimental stage), emphasis has been placed on the development of coding methods to reduce transmission requirements. Two such methods have been demonstrated. The first is based on point-by-point transmission encoded in DPCM (differential pulse code modulation). The second is a progressive transform coding (see Nicol and Fenn, 1979). Transform coding permits rapid transmission of a low resolution version of the image (as if with large pixel size); subsequent information refines the details until full resolution is achieved. In this way the user can quickly interrupt the data stream when he decides that he has seen enough information.

Captain

The Japanese videotex system Captain (see Harashima and Kobayashi, 1981) has been developed to work in an environment quite different from that of the Western countries, all of which use variants of the Latin alphabet. The Japanese written language uses three kinds of alphabets: katakana, hiragana, and kanji. There are about 120 characters in the two former, each character phonetically representing a syllable. Kanji characters are ideographic (conveying whole words and notions), have complicated shapes, and, impor-

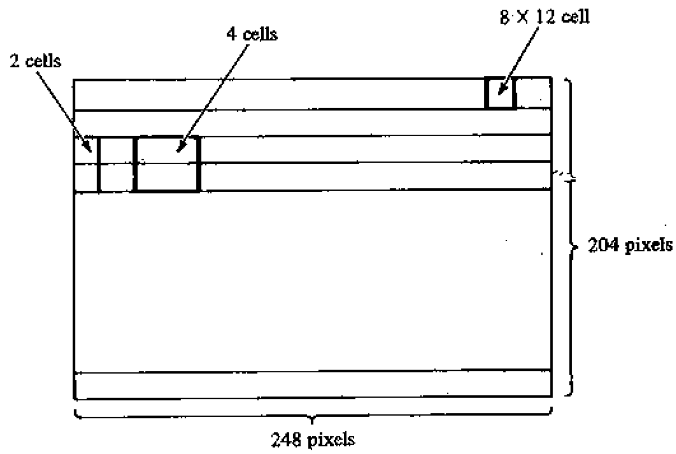


Figure 8.1 Screen organization in Captain

tantly, number around 3000 in daily use. In addition to these alphabets the presentation coding system must cater to Latin alphanumerics, as well as to mosaic-like graphics.

As a result, Captain uses a combination of the photographic and mosaic approaches. The display is a matrix of 248 ×204 pixels, backed in the terminal by a bit plane display memory of 1 bit/pixel, giving a capacity of 50.6 kb. Superimposed on the pixel structure is a matrix of 31 × 17 cells (called subblocks) each dimensioned at 8 × 12 pixels, as shown in Figure 8.1. The terminal contains an additional attribute memory, in which attributes such as color, blinking, and concealing are stored on a *per-cell* basis. Various characters, drawings, and graphic elements are defined in three sizes comprising 1, 2, or 4 cells. Figure 8.2 illustrates the different possibilities. In the trial system (which has been in operation since 1979) the shapes of all these elements are transmitted photographically and loaded into the display memory. This method is being modified for the commercial service (to begin in 1983) in the following ways. All characters will be

Size of area	Dot matrix transmitted without border area	Type of pattern	No. of different patterns
4 cells	15 X 18	Kanji characters Kana characters Latin characters Special symbols	2987 566 (Kana and Latin characters) 4
2 cells	7 X 11	Kana characters Latin characters	356
1 cell	7 X 9	Kana characters Latin characters	230
1 cell	8 X 12	Mosaic graphic	186

Figure 8.2 Patterns displayable in Captain

stored and transmitted in coded form; photographic transmission is retained only for drawings and diagrams. This will require the inclusion of a large (about 1 Mb) character memory in the terminals. Data transmission is by packets, some packets carrying dot patterns, others attribute values, page header information, or entire rows of dots or character codes. The following provisions are made in order to cope with excessive amounts of data normally required in this mode of operation.

1. Transmission rate over phone lines is 3200 b/s; it will be upgraded to 4800 b/s in 1983.
2. Data compression techniques, such as run-length coding (coding of long sequences of identical bits by transmitting the length of the sequence rather than the sequence itself—e.g., (12×0), (50×1) instead of 000000000000111...1) are heavily used.
3. Cells can be individually addressed; empty areas do not take up transmission time.
4. New images can be formed through modification of old ones.

The resulting transmission times are reported to be between 3 and 10 seconds per page. This will be reduced to about 1 second in the commercial version. A typical Captain display is shown in Plate 6.

Telidon: The Geometric Option

Telidon has been under development by the Canadian Department of Communications since 1975. It became known as the first of the "second generation" videotex coding systems, going significantly beyond the mosaic-based Prestel and Antiope.

In May, 1981, Bell Systems announced its Presentation Level Protocol (see PLP, 1981), which incorporates the geometric option of Telidon with only minor modifications. Subsequently Canada announced its adoption of PLP, (see PLPS, 1982). This made the original Telidon system obsolete (although it is, nevertheless, practically a subset of PLP). Therefore we limit this section to a general discussion of the geometric option and its characteristics. PLPS is described in more detail in Section 8.2.

Telidon is perhaps best known for its geometric coding option. However, it also coherently integrates all three display options (mosaic, geometric, and photographic) and solves some problems that stem from this integration, such as unified handling of attributes and combining of different options in one picture.

Computer Graphics

In order to place Telidon in a proper perspective, it might be useful to recall the principles of computer graphic systems. Figures 8.3(a) et 8.3(b) show two basic types of CRT-based graphic systems in use today: the *random deflection* type and the *raster-scan* type. Both are centered around a *display processor* capable of interpreting simple display commands, such as drawing of points, vectors (lines), and, sometimes, arcs, at arbitrary positions on the screen. Commands are regrouped into *display files*, compiled in a computer from a

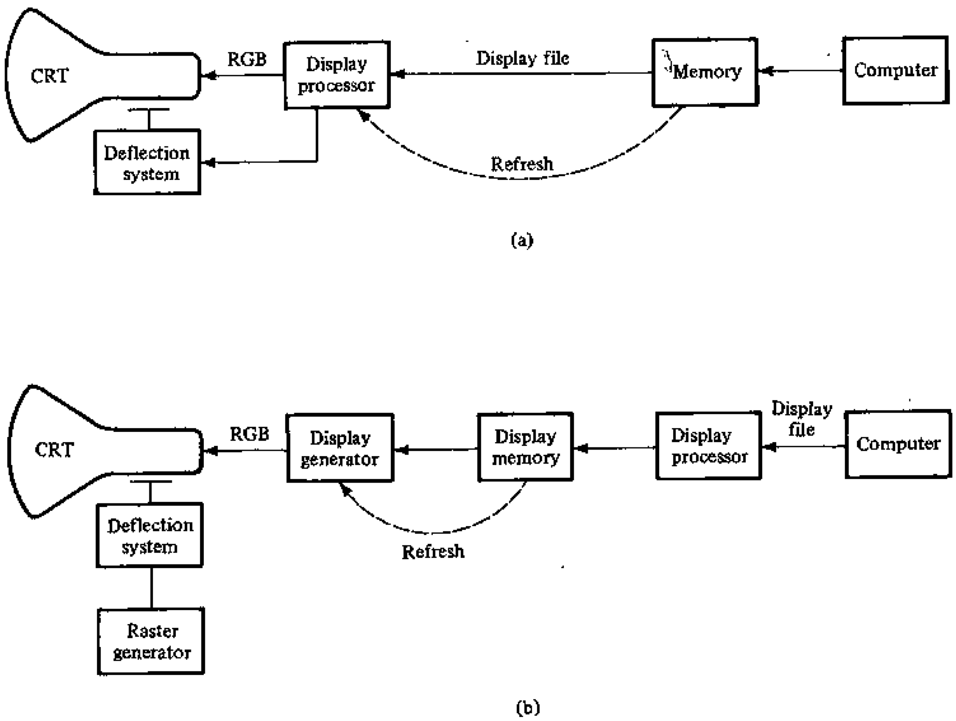


Figure 8.3 Principle of graphic systems: (a) random deflection; (b) raster scan.

(usually high-level, and structured) *graphic language*, such as, e.g., CORE (see CORE, 1979). In the random deflection system the display processor controls the CRT (deflection and intensity of the beam) directly. Commands are executed as they arrive by "painting" the points, lines, etc., directly onto the screen. The process has to be repeated with sufficient refresh frequency in order to obtain the impression of a steady image. Shapes displayed by this method are smooth; the image is not composed of individual pixels, but is drawn by the movement of the beam. The display file is read out repetitively from the refresh memory, which is usually the main memory of the computer.

Raster scan systems compose the image in the same way as television does. The image to be displayed is stored, pixel by pixel, in a display memory whose dimensions (in pixels) are identical to those of the display area. This type of memory is called a *bit plane* memory. The display file is interpreted only once; the image is refreshed at TV rates from the memory. Therefore the display processor may work much more slowly than would be required in a random deflection system.

Telidon is evidently a raster-scan-type system; its display processor is the terminal controller (decoder). Telidon's commands are called PDIs (picture description instructions); functionally they are roughly equivalent in capability to that of modern graphic terminals, such as the Tektronix 4027A (see Tektronix 1982). PDIs are not a high-level graphic language, despite claims to the contrary.

PDIs

One of the design goals of Telidon was to achieve independence of the coded form of picture descriptions from the display terminal's resolution. In this manner a whole range of terminal and display technologies might be used without having to recode the database (mosaic-based systems evidently do not have this type of code independence). This goal is reflected in the choice of the PDI set. The six available commands (instructions) are the basis of a language. Its general syntax is simple: A display file is a sequence of PDIs, each followed by a variable amount of coordinate or other supplementary information. The instruction semantics are the following:

1. POINT sets the drawing beam to a position given by the subsequent coordinates and optionally draws a point.
2. LINE draws a line based on its end points.
3. ARC draws a circular arc that traverses three given points.
4. RECTANGLE draws a rectangle with given height and width (filled with color or not).
5. POLYGON draws a polygonal outline based on a series of given vertices (filled or not).
6. INCREMENTAL (point, line, polygon) draws images point by point, as in photographic mode, or from small lines segments. (This PDI is a PLPS generalization of the Telidon BIT PDI.)

The coordinate system is also chosen in a manner that provides independence from the resolution of the display (and display memory). All absolute and relative coordinates are expressed as signed binary fractions in the interval $(0, +1)$ and $(-1, +1)$, respectively. The display is a rectangular area with the origin $(0, 0)$ at the lower-left-hand corner; the right margin is at $x = 0.999\ldots$, the upper margin at approximately $y = 0.75$ (for an aspect ratio of 4:3). The accuracy of coordinate values (default of 8 bits) typically exceeds the resolution of terminals (e.g., 200×256 pixels in current TV-based terminals). It is the responsibility of the decoder to interpret the received coordinate values with respect to the given resolution (this is usually done by simply truncating the coordinates to the required precision).

As a simple example consider the (virtually colored) drawing shown in Figure 8.4. It can be described by the following symbolic sequence of PDIs (ignoring the details of coding the coordinates and the precise form of attribute controls):

- background color = blue
- color = yellow
- polygon filled (ABCD)
- color = red
- polygon filled (EFG)

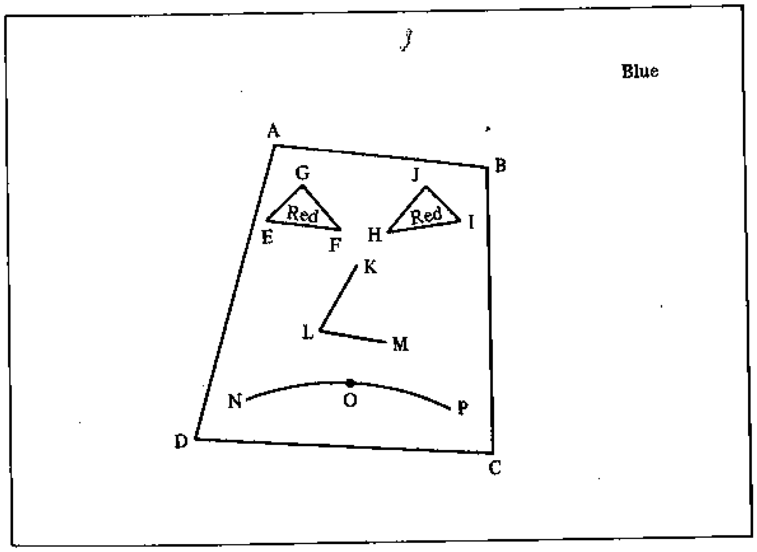


Figure 8.4 Illustration for PDI example.

- polygon filled (HIJ)
- color = black
- line (KL)
- line (LM)
- arc (NOP).

Alternatives to Geometric Coding

PDIs are by no means the only way of coding graphical information in a resolution-independent way. Methods for real time coding and transmission of handwritten and drawn information have been developed for so-called *telewriting* equipment, enabling two or more remote participants to have a common "visual space." In a recent article, Lorig et al. (1981) describe an implementation of a telewriting system. The principle used for coding of drawn and handwritten information is that the whole image is considered to be a complicated line drawing: namely, the trajectory of the drawer's pencil as he creates the image. The lines have different attributes (thickness, color, etc.). What is encoded (and transmitted, stored, and displayed) is the evolution of this trajectory in time. The input device is usually a drawing tablet. The starting point of each segment (after a pen-off) is transmitted in absolute coordinates; then, at sampling intervals of about 10–30 ms, the current speed and differential angle of the line are sent by the tablet to the controlling processor. Adding a few features, such as automatic filling of closed areas and erasing, makes this system quite powerful. It has been demonstrated that by using Huffman encoding, most drawings can be accommodated with a data rate of about 300 b/s. The terminal implemented by Lorig et al. can work in videotex (mosaic), telewriting, or combined mode, creating and displaying pictures such as those shown in Plate 8. Another well-known telewriting system called Vidibord has been developed in Holland.

Several propositions for geometric coding in videotex exist which are quite similar in style to Telidon; for example Vivian and Day (1981) describe a coding system for which geometric instructions are conceived of as an extension of telesoftware. The language is a possible candidate for Level 4 of the new British viewdata and teletext standard.

8.2 STANDARDS AND UNIFIED PRESENTATION SYSTEMS

S.100

Recommendation S.100, "International Information Exchange for Interactive Videotex" (see CCITT, 1980) was approved by CCITT in November, 1980. Its basic purposes are:

- to facilitate introduction of videotex services and interworking with other text communication systems, such as teletex
- to identify parameters needed to design videotex terminals.

Although the title of the document includes the words "interactive videotex," the body of the document is not specific to use with one-way or two-way media. In fact, the view is expressed at the beginning that compatibility between the two kinds of terminals is desirable (that is, compatibility between the parts dealing with the interpretation, not the reception, of presentation codes).

The recommendation itself is little more than an attempt to reconcile the three major presentation systems, Prestel, Telidon, and Antiope, by giving them the status of international standards (however, it must be added that certain features of Telidon have been superseded by PLPS in the meantime). This goal is carried out by defining a code extension scheme based on ISO 2022, by specifying the required control sequences, and, finally, by including the sum total of alphabets, attributes, and commands found in the three component systems. The two C1 control sets defined as "serial" (à la Prestel) and "parallel" (à la Antiope) are essentially those of the CEPT proposal (see below). As a consequence, Prestel, Telidon, and Antiope are compatible with S.100 in the sense that they are more or less proper subsets of it. On the other hand, implementing an "S.100 terminal" would practically amount to putting three terminals under one cover. There is no generalization or indication as to how future enhancements would fit naturally into the structure (except by obvious 2022-style code extensions). There is practically no provision for DRCS or photographic mode; the section on DRCS is just a motherhood-style cursory enumeration of a few general functions, of which "one or more" should be present in an implementation. The language and style of the document are full of unnecessary complications and ambiguities (the "one" coding scheme mentioned for the geometric option has already been the subject of hair-splitting discussions over whether

Telidon—now obsolete—is "the one," or "one of" the schemes). The reading of this document would be perhaps less irritating if its philosophy (of putting the three systems under one cover) were, at least, openly admitted, discussed, and justified.

CEPT Recommendation

This document, called "Current Status of Harmonisation for Videotex Display Aspects and Transmission Coding for 26 Countries of Europe" (first part), (see CEPT, 1981) summarizes the results of European efforts to establish a common denominator for videotex services.

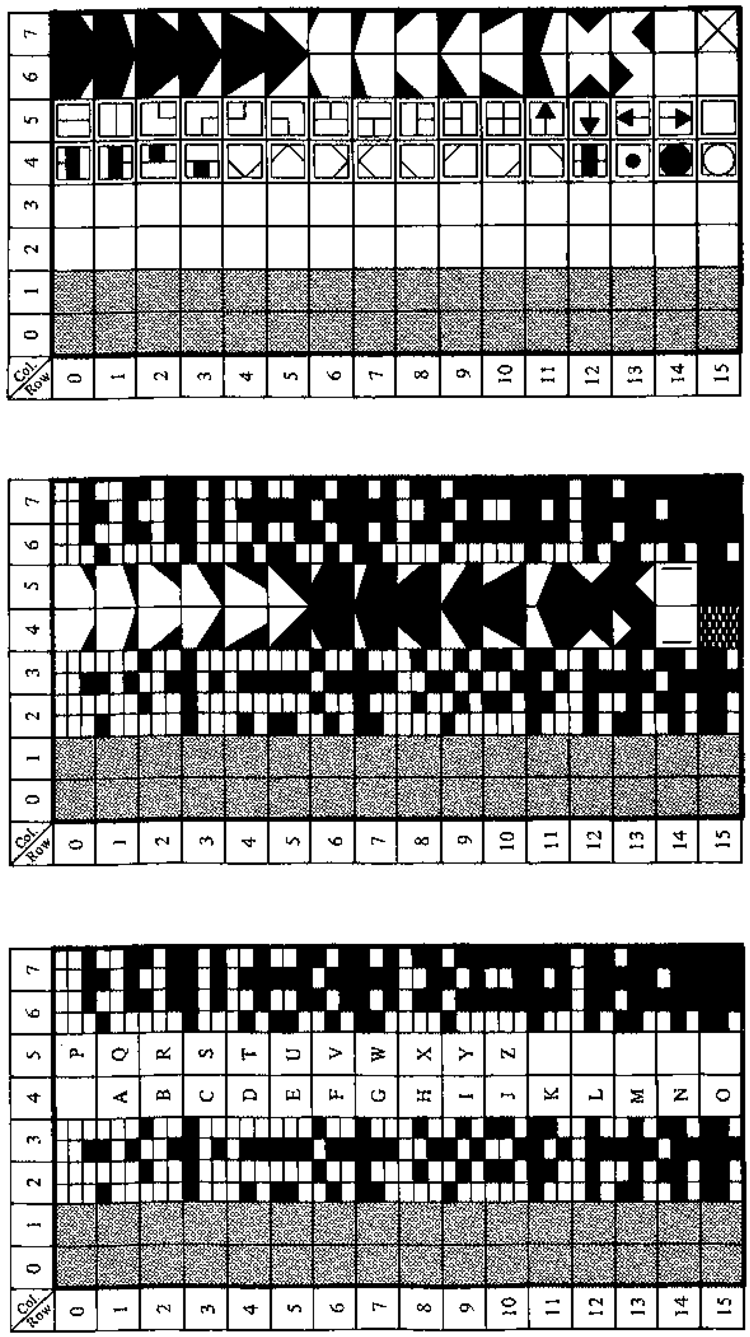
The document defines the principles for a unified mosaic-based presentation system, containing Télétel and Prestel, that are an elaboration in the following directions of the principles contained in S.100:

1. A comprehensive alphanumeric character repertory covering all European Latin-based alphabets is defined. The relevant standard is ISO 6937 (1981). There are about 320 characters in total, many of them formed by combining characters with accents (non-spacing) from the G2-set. The C0 and G0-sets are based on ISO 646 and are nearly identical to those shown in Figure 7.9. The supplementary graphic set is similar to that shown in Figure 7.10.

2. Three mosaic graphic sets are included. Two of them (see Figures 8.5(a) and (b)) are to be used with serial and parallel attributes, respectively. The 64 basic mosaic shapes are the same in both. The remaining 32 positions (columns 4 and 5) are used differently. In the first set (Prestel-compatible) they contain all upper-case characters, which can be thus mixed with mosaics without intervening control characters. This is common practice in Prestel. The free space in the second table is used for smoothed mosaics. The parallel mosaic set is invoked as G1-set, the serial, through certain control codes, from the corresponding C1-set. A third graphic alphabet, with smoothed and lined characters (see Figure 8.5(c)), is available as a G3-set.

3. There are two C1-sets, controlling serial and parallel attributes, respectively. They contain the following attributes and controls:

- foreground color
- background color (of characters)
- full-screen and full-line attributes
- flashing
- conceal/reveal
- character size
- window/box
- separated graphics
- underline
- reduced intensity



(a) (b) (c)

Figure 8.5 (a) CEPT (and Prestel) mosaic graphic set for serial attributes; (b) CEPT (and Antiope) graphic set for parallel attributes; (c) additional CEPT graphic set with smoothed and line graphics.

- inverse video
- hold graphics (for serial attributes only)

4. Serial and parallel controls (attributes) can be freely mixed under the TDUAM scheme (see Chapter 7). Serial attributes are spacing.

5. The page format is defined to be 24 rows and 40 characters.

6. The presentation aspects are intended for both videotex and teletext.

7. Three different cell sizes are defined for DRCS (6×5, 6×10, and 12×10).

8. An outline of fall-back procedures is presented, stating that if a terminal is incapable of displaying a given character precisely, the best available approximation should be used; if this is unsatisfactory, a unique default character should be displayed instead.

The recommendation will have two more parts, which are supposed to contain specifications for a reference terminal and a protocol for interworking between videotex networks of CEPT member nations. For further details on the CEPT standard (e.g., coding of C1-sets) the reader should consult the source documents.

CEPT Terminal

The CEPT proposal was met with enthusiasm among the 26 member nations, and several manufacturers have announced plans to produce the new terminals. The Deutsche Bundespost (1982) has issued functional specifications for a European videotex terminal that complies with the CEPT proposal and contains some additional features, such as:

- protected and marked areas on the screen (in order to designate from erasure or designate for selective transmission to an output device, such as as a printer)
- scrolling area definable between any two lines on the screen
- 32 colors (including transparent) useable on each page, from a total of 4096
- enhanced DRCS capability, with 15 different cell sizes and the possibility to define (and dynamically re-define) the color of individual dots.

The U.K. Multi-Level Presentation System

This standard proposal (Vivian and Day, 1981) consists of five levels. The lowest level corresponds to current public services in the U.K.; higher levels add new features to the terminals and services in such a way that lower-level (older or cheaper) terminals can receive higher-level services with fallback procedures. The five layers are

- current public service: standard mosaic sets with serial attributes
- enhanced mosaics: extended character set, enhanced mosaic set with parallel attributes (supposed to be identical to the basic European service defined in the CEPT recommendation)

- DRCS (12 × 10-cell size) and non-Latin character sets
- geometric option and telesoftware
- photographic option.

North American PLPS

This Presentation Level Protocol Syntax (see PLPS, 1982) was developed by a joint CSA/ANSI working group, and is now in the approval process through CSA and ANSI (the Canadian and U.S. standardizing organizations). PLPS is based on Bell System's PLP and it builds on the coding schemes established by ISO 2022 and CCITT Recommendation S.100.

The most important characteristic of PLPS is that it incorporates coherently all three coding options and all modes of the mosaic option (alphanumeric, graphic, and DRCS), and adds some new functions, such as Macro PDI. Unfortunately, PLPS is not compatible with recent European proposals.

Alphabets. Figure 8.6 shows the code extension for 7-bit codes and the graphic repertory used. The G0- and C0-sets are shown in the standard ASCII table in Figure 7.9.

The supplementary character set (G2 default) is essentially as shown in Figure 7.10, in which free spaces were filled with additional symbols. The set contains accents and some special characters. Accents are non-spacing; for example, the letter "à" would be coded as SS2,`,a. One mosaic set with 64 shapes is included (the same as in columns 2–3 and 6–7 in Figure 8.5(a).

The more specific features of PLPS are supported by three additional G-sets and one C1-set. The G-sets are the PDI, Macro PDI, and DRCS. The purpose of the DRCS set is to hold up to 96 downloaded characters. Once loaded, they can be used as just another mosaic set. The Macro PDI set is not really an alphabet; its 96 positions are named M0–M95 and are used to invoke previously defined strings of PDIs (macros).

The PDI set is a slightly modified version of Telidon's geometric option and includes all opcodes and controls. There is only one C1 control set defined in PLPS.

PDI set. Figures 8.7(a) and (b) illustrate the PDI set at two levels of detail. All 7-bit combinations are interpreted as instructions (columns 2 and 3) or as supplementary information, such as coordinate values.

Instructions and coordinate values for the six drawing instructions (POINT, LINE, ARC, RECT, POLY, INCR, explained in Section 8.1) have the general format indicated in Figure 8.8(a). P is the parity bit, OP the opcode. "Facilities bits" b_1 and b_2 provide for four modalities of each instruction: b_1 controls whether the picture primitive is to be drawn from the position at which the preceding instruction ended or from a freshly defined starting point. (This is specified by SET, as shown in Figure 8.7(b).) The role of b_2 is different for each instruction. In the first two PDIs it decides whether the subsequent coordinates are to be interpreted as absolute (with reference to the origin) or relative (with reference to the final position arrived at by the preceding instruction). For the

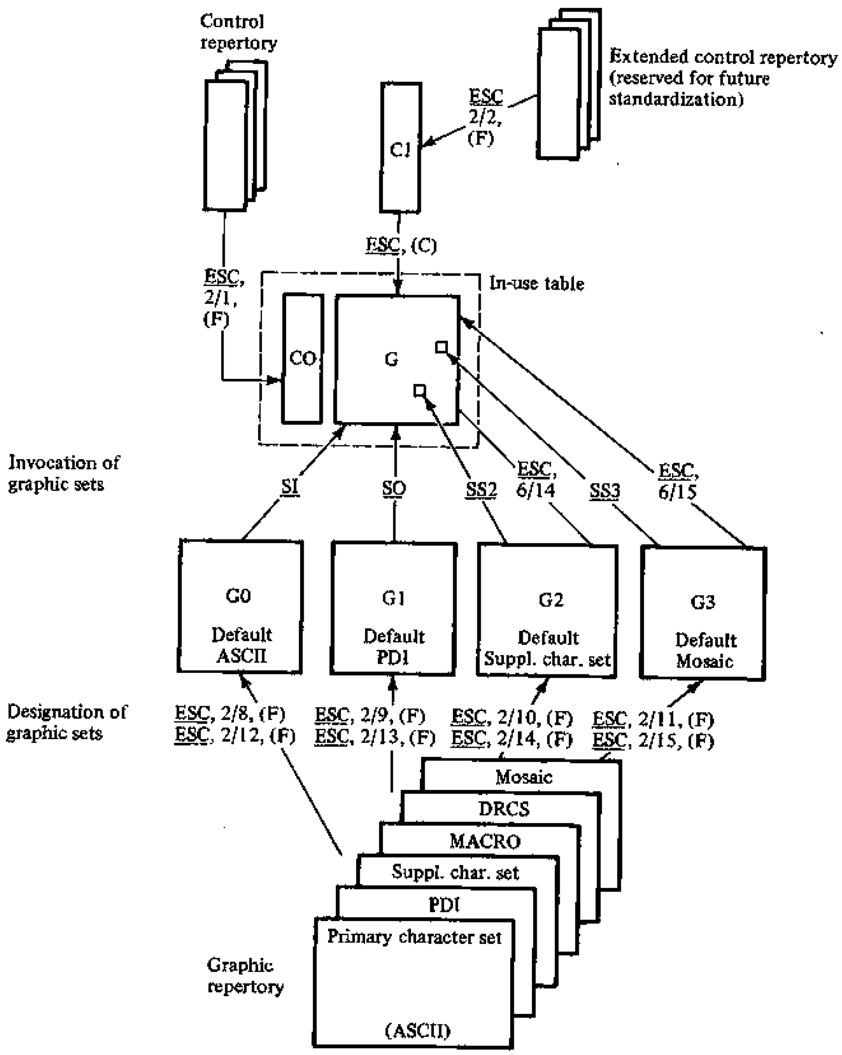


Figure 8.6 PLPS code extension scheme.

other PDIs it controls the filling of the defined area. Other modalities of the drawing PDIs are controlled by CONTROL instructions.

Coordinates for each point are usually coded in three bytes, as shown in Figure 8.8(b). The first byte contains the three most significant bits of both coordinates x and y. The resulting 8-bit precision (and sign) can be changed by a CONTROL instruction (DOMAIN).

Row \ Column	2	3	4	5	6	7
0	CONTROL	RECT	NUMERIC DATA			
1						
2						
3						
4	POINT	POLY				
5						
6						
7						
8	LINE	INCR				
9						
10						
11						
12	ARC	CONTROL				
13						
14						
15						

(a)

Row \ Column	2	3	4	5	6	7
0	RESET	RECT (OUTLINED)	NUMERIC DATA			
1	DOMAIN	RECT (FILLED)				
2	TEXT	SET & RECT (OUTLINED)				
3	TEXTURE	SET & RECT (FILLED)				
4	POINT SET (ABS)	POLY (OUTLINED)				
5	POINT SET (REL)	POLY (FILLED)				
6	POINT (ABS)	SET & POLY (OUTLINED)				
7	POINT (REL)	SET & POLY (FILLED)				
8	LINE (ABS)	FIELD				
9	LINE (REL)	INCR POINT				
10	SET & LINE (ABS)	INCR LINE				
11	SET & LINE (REL)	INCR POLY (FILLED)				
12	ARC (OUTLINED)	SET COLOR				
13	ARC (FILLED)	CONTROL STATUS (WAIT)				
14	SET & ARC (OUTLINED)	SELECT COLOR				
15	SET & ARC (FILLED)	BLINK				

(b)

Figure 8.7 PLPS PDI set: (a) instructions; (b) instruction modalities.

All instructions are followed by coordinate data for a number of points. The sequence is terminated upon reception of another opcode, or an SI, SO, SS2 or ESC control code.

CONTROL instructions. Control codes of the PDI set serve to specify certain attributes and display-related parameters. Although not stated explicitly, attributes in PLPS are essentially parallel. Without going into a detailed description, we give a list of functions assured by these codes.

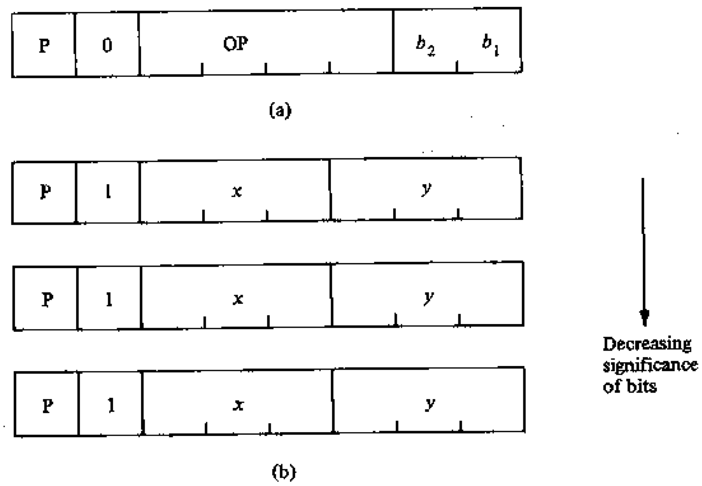


Figure 8.8 Coding of PDIs: (a) instructions; (b) coordinate values.

1. *DOMAIN* defines the size of *logical pixels* (and is essentially the minimum feature size that can be displayed by PDIs; its shape is rectangular, covering one or more display pixels). It also describes the length (in bytes) of subsequent coordinate values.

2. *TEXT* controls attributes of text and the behavior of the cursor. Characters can be displayed with controllable rotational position, size, and spacing. In purely mosaic systems there is only one cursor (active position), defined as the position at which the next character is to be placed. In PLPS the situation is more complicated because there are in effect two "active positions": the character cursor and the current drawing point (at which subsequent drawings should start). The relation between the two becomes important when text and drawings are mixed in one display. There are four possibilities set by the TEXT control: (a) the two move together, (b) the two are independent, (c) the cursor is moved when the drawing point is moved, but not vice versa, and (d) the drawing point is moved when the cursor is moved, but not vice versa.

3. *TEXTURE* controls the texture of lines (solid, dotted, dashed, etc.) and of areas (solid, cross hatched, etc.). Area texture patterns can also be downloaded from the host computer.

4. *SET COLOR* and *SELECT COLOR* serve to control the current color for drawings and mosaics, as well as the color repertory available. In addition to the eight basic colors and gray shades, a large number of additional colors and shades can be used through color look-up tables. Color tables, like textures, can be loaded from the database.

5. *BLINK* occurs when any entry in the color table is made to change its color value periodically to another color. When this control is in effect, all elements of display drawn with the color in question will blink. (Another form of blinking is specified in the C1-set.)

6. *WAIT* indicates a programmable delay of up to 6 seconds in the execution of subsequent presentation codes.

7. *RESET* selectively reinitializes control and attribute values to their default values.

Photographic option. Photographic coding is realized in PLPS by the first two of the three INCREMENTAL instructions:

1. *FIELD* establishes a rectangular area (in terms of logical pixels) to be filled by photographic transmission. The logical pixel size determines the resolution of the resulting image (and the required amount of data to be transmitted).

2. *INCREMENTAL POINT* is followed by values that specify successive logical pixels in the field. The drawing order is first by lines and then by columns. The color codes, precision, etc., are determined by the current settings of the DOMAIN and COLOR controls.

3. *INCREMENTAL LINE AND POLYGON* permit the drawing of figures as a series of short lines (steps) of the size of the currently defined logical pixel. Similarly to techniques used in incremental plotters, steps can be horizontal, vertical, or diagonal. The resulting shape is filled with the POLYGON instruction. These instructions are advantageous for precise drawings, such as signatures.

C1 control set. PLPS relies on a single C1-set, shown in Figure 8.9. The set provides additional attribute controls not contained in the PDI set, as well as controls over the process of downloading PDI macros, DRCS, and texture patterns. We list some of the more important codes.

1. *DEF MACRO* is followed by a character whose name is given to a subsequent string of PDI codes (not executed when downloading). The macro is invoked through the Macro G-set. Macro definitions can be nested.

2. *DEFP MACRO* is the same as above, except that the PDI string is, in this case, simultaneously stored and executed.

3. *DEFT MACRO* is a macro that, on invocation, will be transmitted back to the host computer (as if it were a remotely programmable function key).

4. *DEF DRCS* permits loading of DRCS characters selectively or sequentially. The (96) shapes are defined as bit patterns of the dimension given by the current character size.

5. *SMALL, MEDIUM, NORMAL TEXT* set the current character size to fit 20 rows by 80 characters, 16 by 32, and 20 by 40, respectively. Further, 10-by-40 and 10-by-20 displays can be created by DOUBLE HEIGHT and DOUBLE SIZE controls. Note that another way to control the character size is through the TEXT PDI control.

6. *DEF TEXTURE* permits up to four texture masks to be downloaded, in a manner similar to the defining of DRCS.

Row \ Column	4	5
0	DEF MACRO	PROTECT
1	DEFP MACRO	EDC_1
2	DEFT MACRO	EDC_2
3	DEF DRCS	EDC_3
4	DEF TEXTURE	EDC_4
5	END	WORD WRAP ON
6	REPEAT	WORD WRAP OFF
7	REPEAT TO EOL	SCROLL ON
8	REVERSE VIDEO	SCROLL OFF
9	NORMAL VIDEO	UNDER LINE START
10	SMALL TEXT	UNDER LINE STOP
11	MED TEXT	FLASH CURSOR
12	NORMAL TEXT	STEADY CURSOR
13	DOUBLE HEIGHT	CURSOR OFF
14	BLINK START	BLINK STOP
15	DOUBLE SIZE	UNPRO-TECT

Figure 8.9 PLPS C1 control set.

In summary, PLPS is a consistent image-coding scheme that incorporates, with new extensions, all presently used coding options. Its display capability is certainly adequate for a wide range of videotex applications, and it permits conception of a range of compatible terminal models of increasing cost and functionality. Fabrication of terminals compatible with PLPS has already begun in Canada and the U.S. A subset of PLPS is also defined for implementation in teletext terminals.

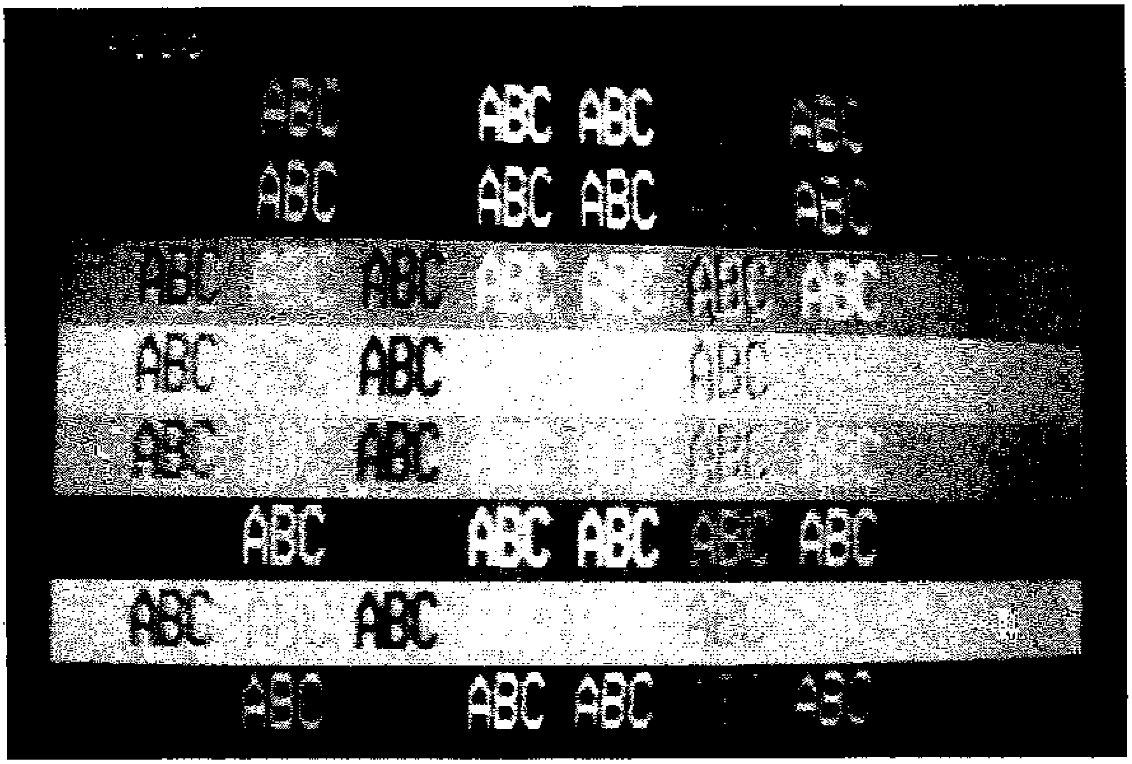
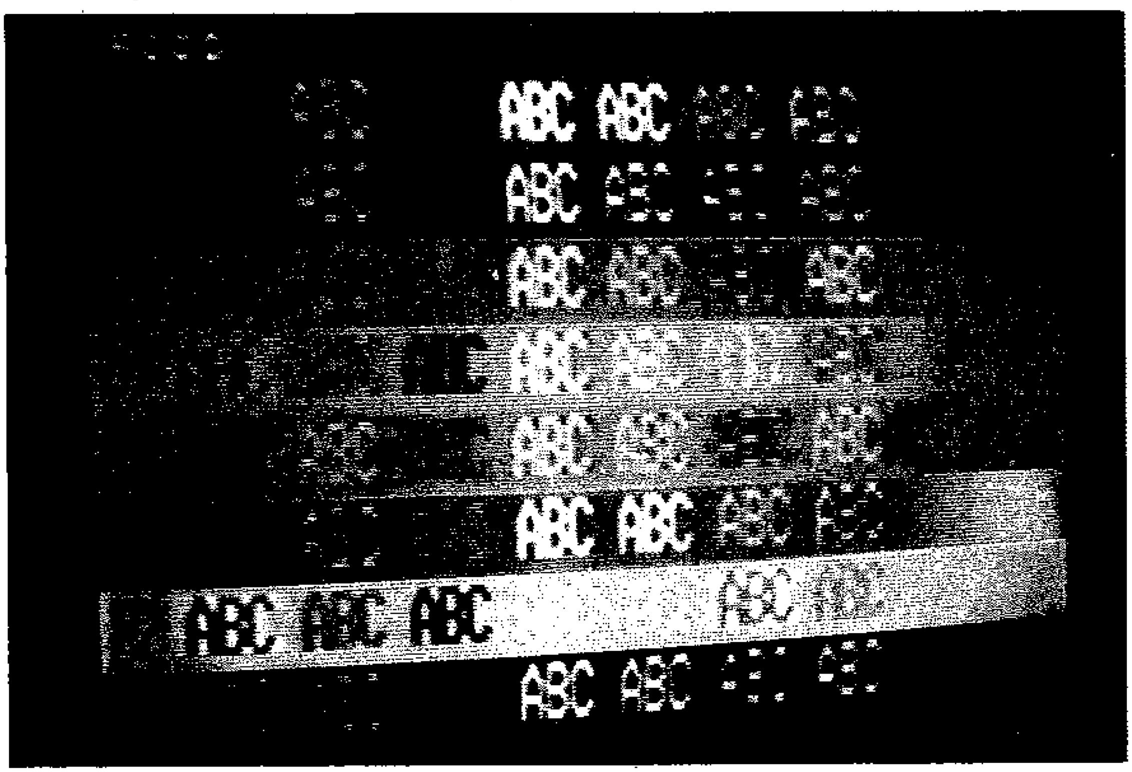


Plate 1 Sample image using RGB input.

Plate 2 Degradation due to RF connection through the antenna socket.



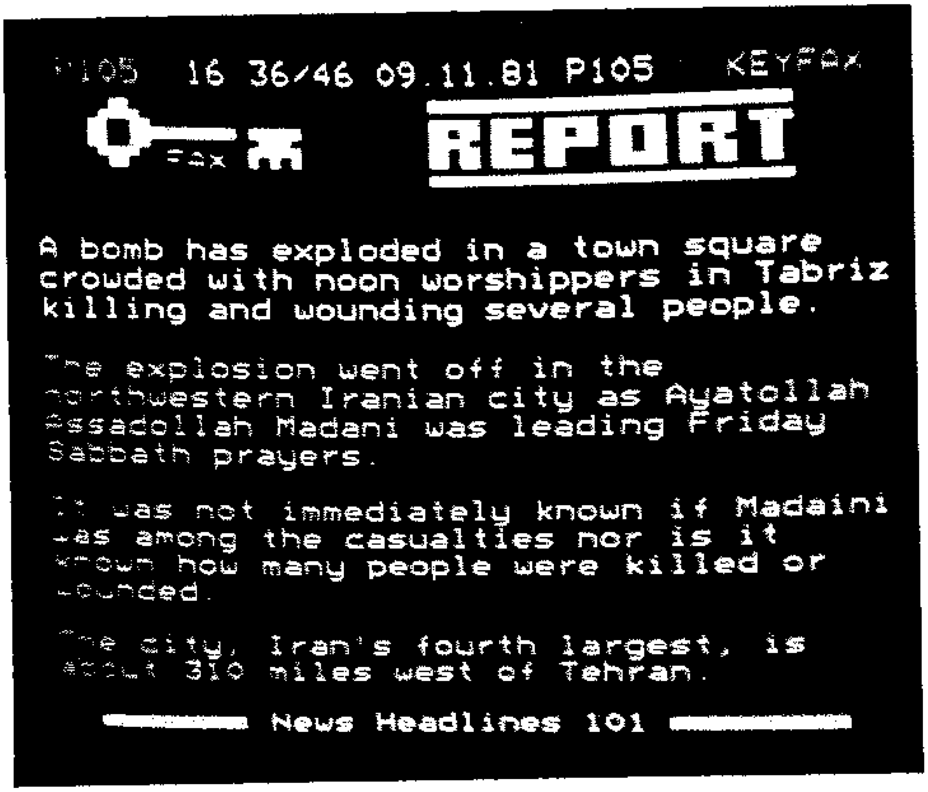
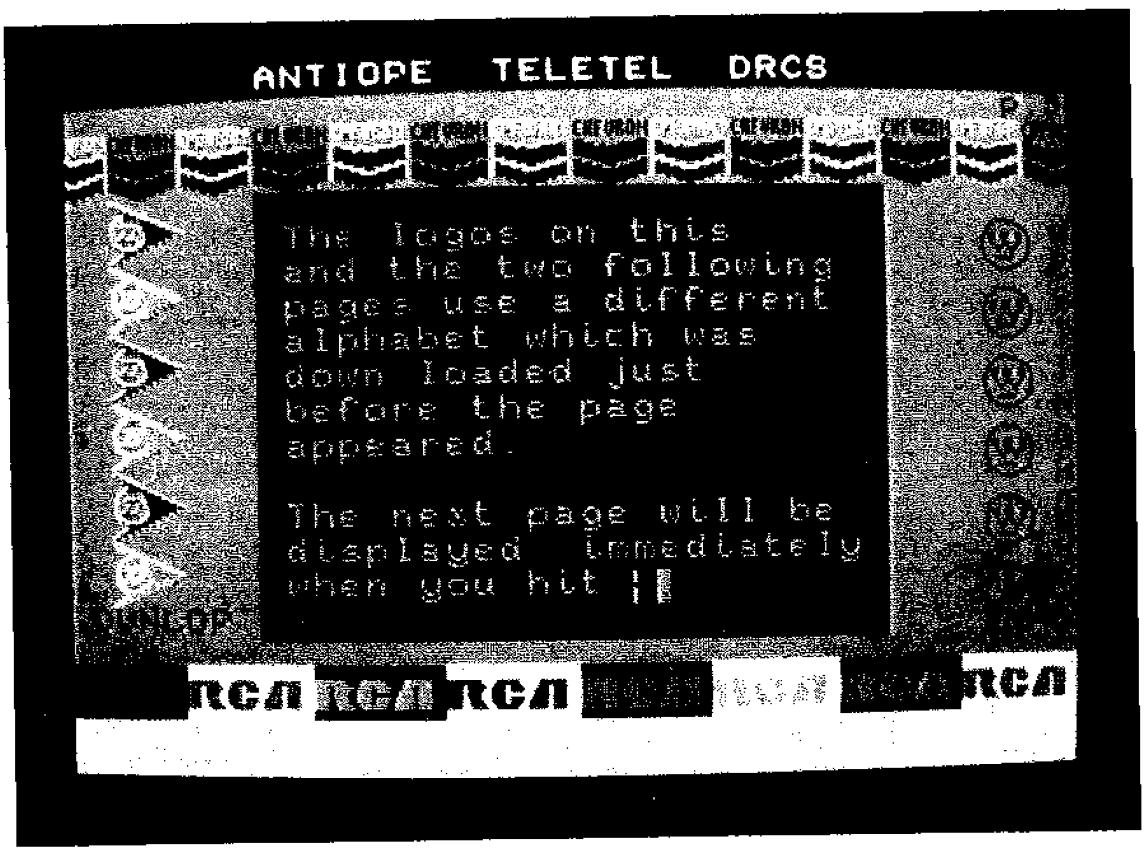


Plate 3
Typical mosaic page (teletext).

Plate 4
Sample DRCS page.



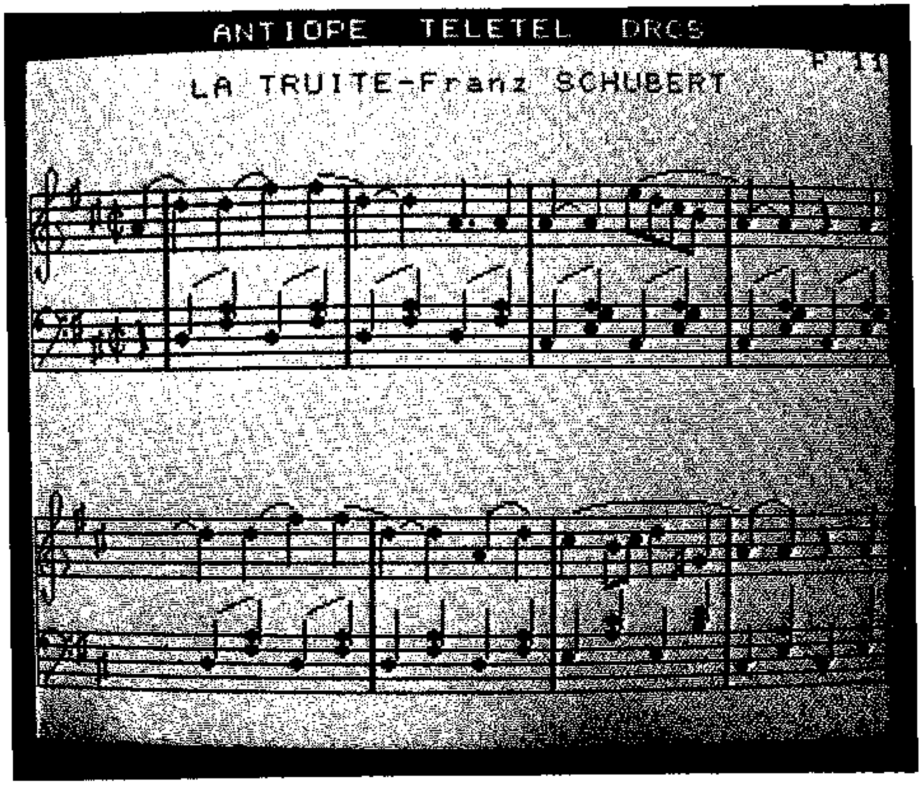


Plate 5
Sample DRCS page.

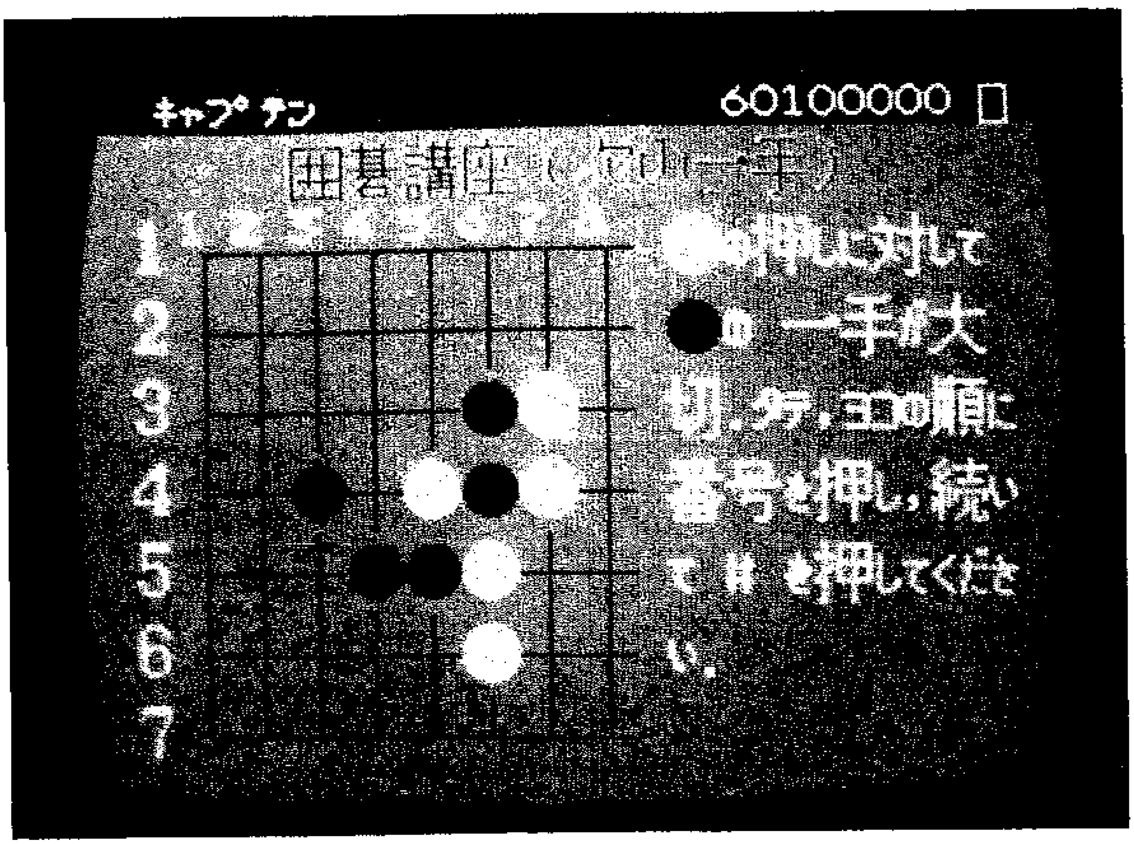
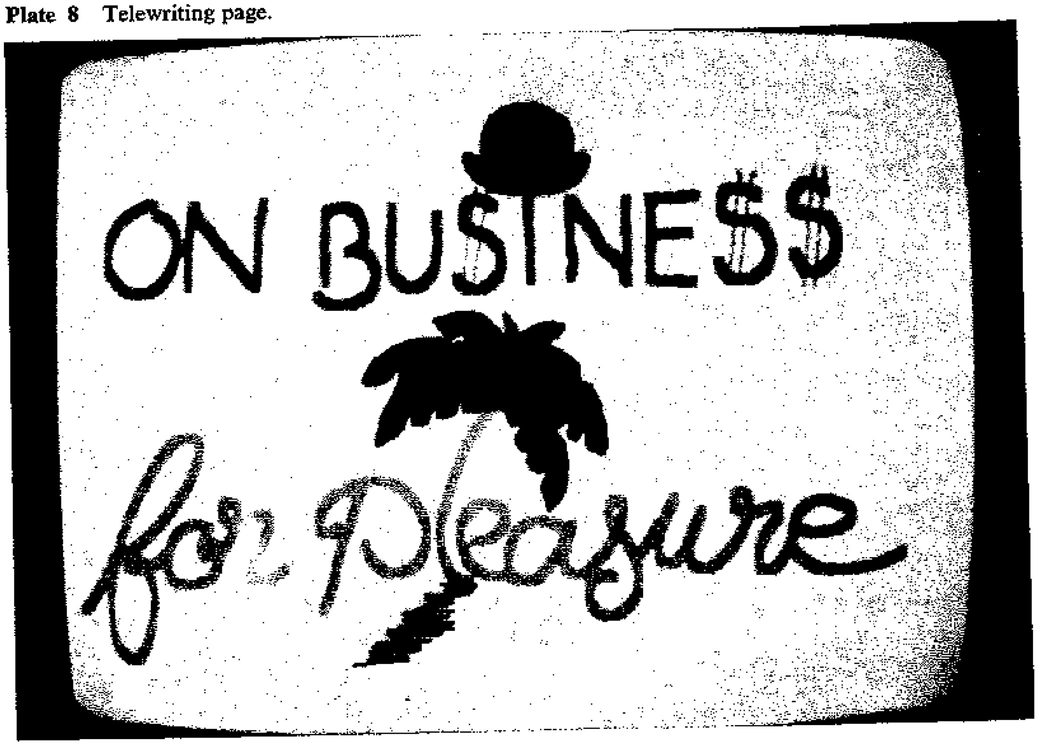


Plate 6
Captain page.

Plate 7 Sample Telidon page (geometric coding).

Plate 8 Telewriting page.



8.3 COMPATIBILITY AND STANDARDS

The Nature of Compatibility

"Compatibility" is one of the most abused terms in the videotex literature. Nevertheless, the issue of interworking between services and systems is of considerable technical, economical, and political importance. Unfortunately, evolution does not always act in a positive sense; history is filled with cases in which new, incompatible standards and systems were created and promoted by various groups in the hope of economical and political gains. Without further elaborating on this subject, we present a few remarks and definitions that should clarify the notion of compatibility, at least from the technical point of view.

1. Nothing can be compatible in itself, but often it is not at all clear from the context what should be compatible with what, and in which sense. These details of interpretation often make all the difference in the world and have far-reaching implications for technical implementations.

2. Two systems (material or abstract) are said to be *strongly compatible* if they can cooperate in a predefined sense, or if one can be replaced by the other without impairment of its functions (example of such systems: two Prestel terminals fabricated by different companies.

3. Given two systems considered to be evolving in time towards more advanced versions, *forward* and *backward* compatibility assure that later versions of one can cooperate with earlier versions of the other and possibly vice versa. What is "forward" and what "backward" often depends on the context. It is important, though that the relation may work in both ways or only in one way. A well-known example of forward and backward compatibility includes monochrome TV sets that can receive color TV signals and color sets that can receive monochrome signals.

4. The terms "upward compatibility" and "downward compatibility" are usually applied to coexisting versions of systems (e.g., computers or videotex terminals). Higher (and more expensive) versions usually include all the functional capabilities of lower versions. *Downward* compatibility assures that higher versions can execute programs (or presentation codes) developed for lower versions; *upward* compatibility assures that lower-version systems will interpret correctly some portions of higher version programs and codes, and will make reasonable default decisions for portions it cannot deal with (example: the U.K. multi-level presentation system.

5. Frequently "compatibility" is used in a quantitative rather than logical sense. There are various degrees and interpretations of the term that, if not precisely stated, may deprive the word of any meaning. For example, the statement that "the PLP mosaics are compatible with those mosaics used by the Prestel and Antiope coding schemes" (see PLP, 1981) requires the important qualification that only the mosaic characters, and not the way their attributes are coded by the two systems, are compatible.

6. A common way to establish compatibility between two interworking systems is to add hardware/software adaptation modules (converters) in front of one or both systems. The degree of compatibility can then be measured in terms of the converter complexity.

7. This leads us to a loophole that exists in many compatibility issues. There is a difference between de facto, or strong, compatibility (which simply means that two systems *are* compatible as they stand) and various degrees of weak compatibility (which usually means that the systems *can be made* compatible). It is of course the amount of work necessary that makes claims of compatibility a meaningful or ridiculous. Thus, for instance, one could claim to have invented a terminal (or presentation protocol) compatible both with Captain and PLPS by simply putting two separate decoders in one box (but perhaps sharing the power supply). However, there remains the question of how to decide which decoder to use at a given time. The general solution to this would be to agree on some identification code carried by the Captain and PLPS data streams—that is, to change the present coding structures.

8. With respect to videotex, compatibility issues arise at all levels of the OSI model. (These issues are treated in the appropriate chapters.) However, at higher levels (presentation, application) an important distinction exists between what might be called "on-line" and "off-line" compatibility. Consider as an example access to a Prestel database by PLPS terminals. To achieve this some amount of transcoding must be done at the presentation and application levels. To do this on-line—that is, "on the fly"—would be quite complicated; one would rather do it the other way—that is, convert the whole database into Telidon format, and then connect it to the users.

PLPS and CEPT Standards

PLPS and the CEPT proposal are the major candidates for formal standardization in North America and Europe, respectively. At least partial compatibility of the two standards (if there must be two standards at all) would be clearly desirable. Such compatibility would facilitate data exchange between the two continents. Unfortunately, the two standards are only compatible at the level of the 64 basic mosaic characters, and (subsets of) the basic and supplementary character sets.

Here are the major areas of incompatibility:

1. Almost all mosaic attribute controls are coded differently in the two standards.
2. There are no serial attributes in PLPS.
3. There are no geometric and photographic options in CEPT; if PDIs are included (as per S.100), new incompatibility may result because the control sequence to invoke them in PLPS is different from the one foreseen in S.100.
4. There are 24 rows per page in Europe versus 20 rows in North America (although 24 rows are not precluded in PLPS).

As Tanenbaum (1981) said: "The nice thing about standards is that you have so many to choose from; furthermore, if you don't like any of them you can just wait for next year's model." The next year's model may well be on its way: Schmid and Maurer (1981) propose a "European" PLPS with features similar to those of the Bell protocol, but with control structure close to that of CEPT.

9
Terminals

9.1 CLASSIFICATION AND FEATURES

There are two broad categories of users of videotex systems: service consumers (e.g., households or businesses) and service providers (e.g., information providers). In this chapter we shall refer to the terminals used by these two classes as *user terminals* and *service-provider terminals*. On the whole, the distinction between the two is fairly clear: user terminals are (relatively) cheap, simple, and numerous; service-provider terminals may be complicated and expensive. Most of the material in the following two sections deals with user terminals; Section 9.3 is devoted to service-provider terminals and systems.

The terminal is the interface, or point of contact, between a videotex system and its user, who usually owns it. As such, it should have two characteristics, both of which are essential for the success or failure of the whole concept of videotex. First, the cost of the terminal should be sufficiently low; simple structure, advanced technologies, and economies of scale are factors in achieving this. Second, the terminal has to offer attractive displays and easy-to-understand controls that are not a source of frustration for the average user.

Types of User Terminals

So far the most important class of user terminals is based on use of the color TV set as the display device. This has an important price advantage in that, in general, the user already owns one and perceives only the differential cost of additional hardware, called the videotex *decoder* or *controller* (and which includes a keypad for entering user commands).

Two important types of decoders can be distinguished, according to the way in which the decoder and the set are connected: *add-on* (also called set-top) and *built-in* decoders. Add-on decoders have the advantage that they can connect to unmodified TV sets through the antenna socket. As already discussed, there is also the disadvantage that the signal has to pass through two RF (radio frequency) stages (the modulator in the decoder and the tuner in the receiver) and through the IF amplifier. This inevitably results in higher decoder cost and deterioration of image quality. All TV sets in France (and most sets in other countries) are now fabricated with external sockets permitting input of the RGB signal directly from the decoder's generator; this eliminates the above disadvantage of add-on decoders.

Built-in decoders provide the best display quality—a quality even better than TV reception in ideal circumstances. Additional advantages are the possibility of sharing components and tighter interaction between decoder and display (enabling, for example, scan compression). Several firms—most of them European—offer top-of-the-line receivers with built-in decoders for teletext and/or viewdata.

However, as discussed by Ciciora (1980), the decision to incorporate teletext or videotex decoders into TV sets produced at high volume is complicated by several factors. First, there is the "chicken-and-egg" problem: Will the increased cost (10–20% increment) be justified by the additional services? Such services will, however, be provided only if there are enough customers, that is, receivers with decoders.

A second group of problems is related to the design changes necessary in order to incorporate a decoder. Receiver designs are carefully optimized for mass production, and the optimal design if a decoder is present may be significantly different than the optimal design without a decoder. This is especially true for teletext decoders that share many circuits with the set. Yet it is forecast that in the U.S., teletext, due to its simplicity, will achieve large-scale penetration before interactive services will.

Other types of terminals use TV monitors (essentially TV sets without an RF tuner); however, they may cost more than standard TV sets because of economies of scale, even though they contain less.

A variety of non-TV-based videotex terminals have been developed for different special applications. Among these are conventional VDUs adapted to receive subsets of videotex code (alphanumerics) either in text format, in which case a stream of characters is re-formatted into 80-character lines, or in page format, in which case two videotex pages are fitted side by side onto a screen. Newer designs favor microprocessor-based combined business and full-capacity videotex terminals (see Hussein, 1981). Another approach to terminal design (and marketing) is taken by the French PTT, which supplies free terminals for the telephone directory trial. Plans call for such terminals to accompany every telephone in France by 1990. These terminals have a small monochrome CRT display and an alphanumeric keyboard with dedicated keys for standard functions ("Help," "Send," etc.). These latter terminals can also receive Antiope code; colors are represented by various shades of grey. A recently announced teletext receiver employs a liquid crystal display packaged in pocket-calculator-like format.

Terminals of still another class feature high resolution display—e.g., 1024 × 1024 pixels. Such displays require special fine-grain picture tubes, not just better amplifiers and more scan lines. Therefore at the present they are quite expensive ($5000–$20,000), but progress in display technology is rapid, and the situation may soon change. In this respect, the possibility of high-resolution commercial television should be kept in mind.

Multistandard Terminals

One way for the equipment manufacturers to counter the uncertainty about standards is to design terminals capable of switching between different presentation codes. We have already mentioned combined ASCII-videotex terminals; however, these are not really multistandard, ASCII being a subset of videotex codes. In all such terminals the objective is to share as much hardware as possible among different presentation modes.

A terminal capable of receiving Prestel and Antiope code has been implemented and described by Bourassin et al. (1981). It is basically an Antiope terminal augmented by a microprocessor and buffer memory to hold one Prestel page. The microprocessor is dedicated to *transcoding* of the Prestel page into an Antiope code sequence. Thus the modem, character generator, and display unit are shared by both standards. Sony and Matsushita presented new, multistandard teletext/videotex terminals at the Videotex'82 show.

Intelligent Terminals

Videotex is without doubt evolving toward more complex and diversified services. One indicator of this trend is the popularity of gateways in the U.K., France, and Germany. Putting intelligence into terminals (and exploiting it) may open the way for a number of unusual and attractive applications.

Intelligent terminals come about in two ways: by the augmenting of a personal computer with videotex capability (e.g., the interface board for Apple computers marketed by Norpak), or the equipping of a terminal with computing capability. For videotex, the latter alternative seems to be more attractive, for the simple reason that all modern videotex terminals already contain an internal processor and memory. Therefore the cost of design modifications that open these resources for purposes other than decoding of images is lower than the cost of a personal computer.

An interesting example of this latter approach is described by Maurer (1982). The terminal in question, called MUPID (Multipurpose Universally Programmable Intelligent Decoder), is produced in Austria and is available to videotex subscribers on a rental basis. Another type of terminal is NABU's home computer, which is capable of receiving telesoftware to interpret PLPS code. We will return to the implications of intelligent terminals in Chapter 14.

Component Television

Although the TV set has always been considered to be the principal display vehicle for videotex, its present form is gradually becoming inadequate for that purpose. One already-mentioned symptom of this is the duplication of the RF circuits in external controllers; this is particularly annoying for teletext reception. Another related phenomenon is the need for cable converters, which make the built-in VHF/UHF tuner superfluous.

From a broader perspective, one can observe an expansion and confluence in three related areas:

1. Appearance of television-related appliances, such as video games, video discs, and video cassette recorders (VCR) which have the possibility of remote control by program identification codes, analog frame-grabbers, home surveillance systems by closed-circuit TV, and home computers.
2. Advanced features, frequently computer-controlled, built into the top-of-the-line TV receivers, such as zooming, playback, superimposing of several programs on the screen, remote control, and high-resolution and digital TV.
3. New home and business information services, such as narrowcasting, pay-TV, and various digital interactive services based on two-way cable.

If the television set is to maintain its central role as a home terminal, it must evidently be capable of interfacing and accomodating all these "TV-peripherals." One solution would be to build a range of superterminals, each incorporating some set of peripheral functions. Another approach would borrow from the philosophy of audio systems that is based on use of individually connectable and upgradable modules. Such *component television* (see Storey, 1978) would offer advantages similar to those of today's hi-fi systems: the possibility of adding, changing, and upgrading individual components according to the owner's preferences and budgetary limits.

With this latter approach, the question of standard interfaces between components becomes a key issue. An example of an already existing standard involves the voltage levels for RGB and composite video signals. In France, a "peritelevision connector" has been standardized, and has been obligatory in all new sets since 1980. It can connnect to a single device, such as a video recorder, or to a special control box that holds plugs for a number of other devices and that controls their coordination.

A possible component television system is outlined in Figure 9.1. It includes the following modules and groups:

Display module: color picture tube, video amplifier, controllable scan generators.

Video sources (RGB or composite video): video disk, VCR, camera, VHF/UHF (standard) tuner/demodulator, cable tuner, digital TV receiver, synthetic video generator.

Video control box: basicaly a selector/mixer of video signals; it might also contain features for video manipulation, such as zooming, insertion/overlaying of pictures and scan control (for multistandard receivers or reduction of overscan); it would also be the source of video signal for some devices, such as VCR or teletext decoder.

Display memory: catering to teletext, videotex, home computer, and other data services.

Interfaces for digital services: e.g., two-way cable or ISDN.

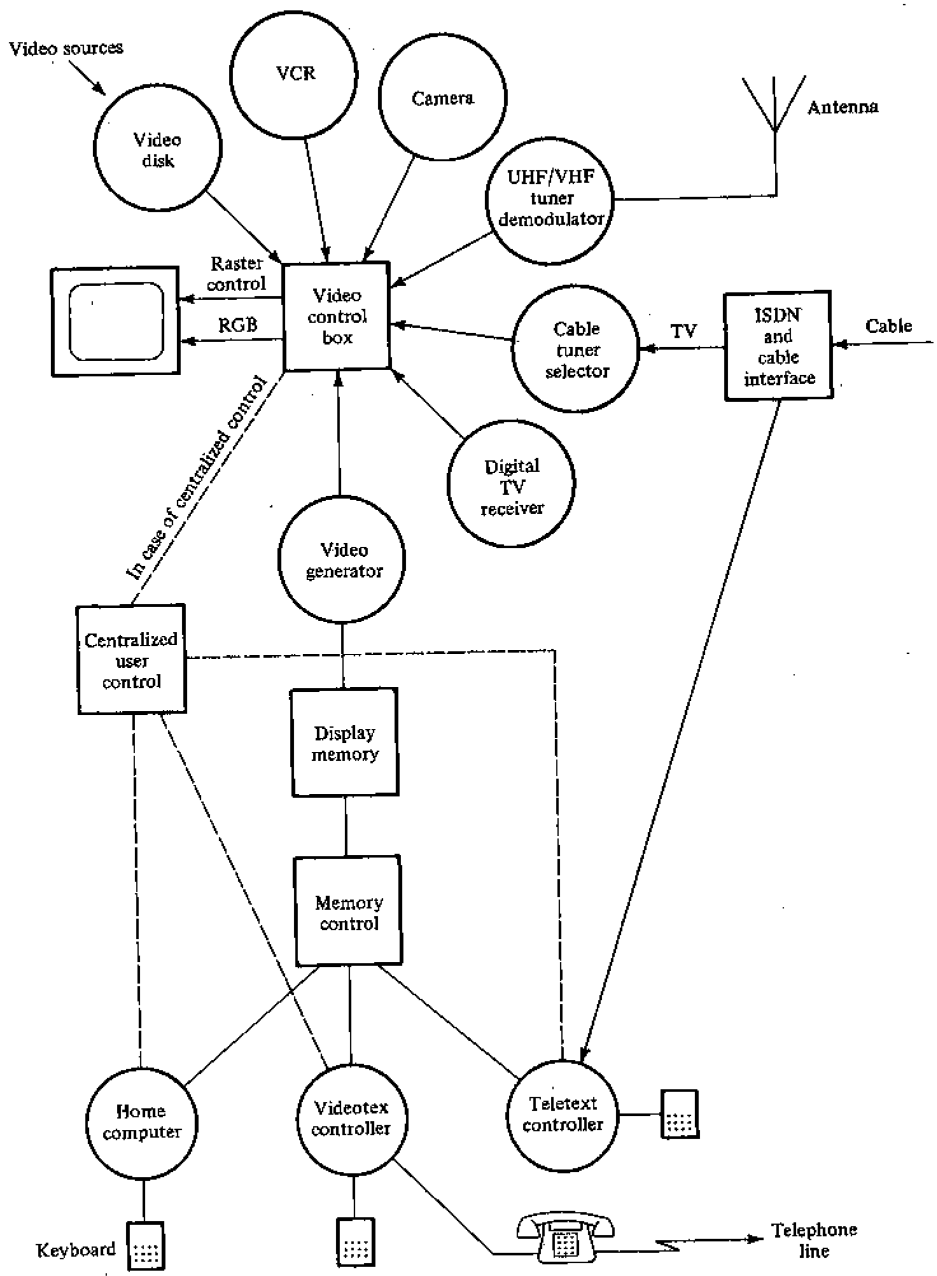


Figure 9.1 Possible arrangement of component TV system.

The control and coordination of this certainly-not-simple system might be centralized and advantageously located in the home computer. Alternatively, several loci of control might exist (e.g., for video, teletext/viewdata, and ISDN).

It is worth noting that alternate or specialized display devices (not shown in Figure 9.1), such as VDU, high-resolution terminals, printer, facsimile, and voice output, will also have to fit into modular systems of the kind outlined above. Some of these devices can be permanently connected to certain data services carried by DOV or ISDN (e.g., facsimile broadcast).

9.2 INTERNAL STRUCTURE OF TERMINALS

The basic components found in all videotex decoders are indicated in Figure 9.2. Their functions are the following:

Data reception: produces a sequence of bits without interpretation.

Data selection: converts the bit stream into bytes and selects those demanded by the current user command.

Control processor: assists in data selection and processes incoming data if necessary.

Display memory: holds all data of the page currently being displayed.

Display generator: creates video or RF-modulated signals to drive the display.

Display unit: consists of TV set or other device.

Input device: is utilized for the user's commands (keypad or keyboard).

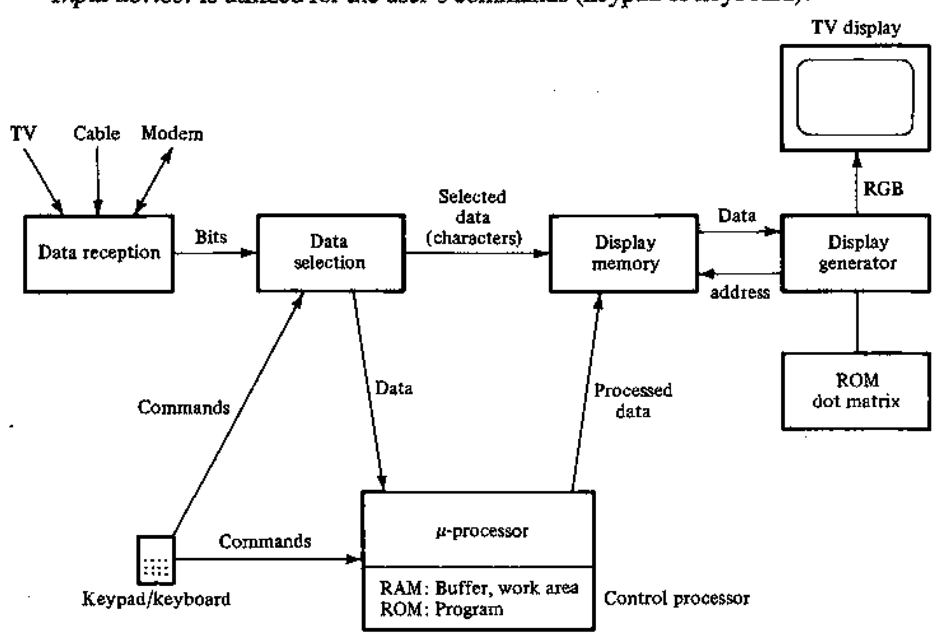


Figure 9.2 General schema of videotex decoder.

These components must be present in some form in every terminal (with the possible exception of the control processor). The term controller or decoder often designates the terminal's add-on part, except for the display device.

Next we deal in more detail with the component parts listed above. Differences exist, of course, between different implementations of controllers; these are mainly due to (1) the delivery medium (telephone, data over video or ISDN), (2) the coding options supported, and (3) technological factors (such as availability of special purpose LSI or VLSI chips).

Data Reception

With telephone, the reception function resides in the modem. Standard asynchronous modems employ frequency modulation for encoding bits. This requires the use of four different frequencies (for the transmission of 0's and 1's in each direction). The operation in the modem is therefore demodulation of the received signal and marking of the bit boundaries. A similar but reverse process takes place in the modem when user commands are transmitted to the service computer. Modems are usually equipped with acoustic couplers for use with standard telephone sets; some models have built-in auto-dial facilities.

When videotex is received over the television signal, add-on teletext decoders must replicate the RF, IF, and demodulation circuitry already present in the TV set; therefore it is particularly desirable to build the decoders into the TV set. After demodulation the digital information carried on a data line must be separated from the rest of the video signal. This is done in two steps. First, the video signal is fed to a threshold circuit ("slicer") producing 0 or 1 (logic) as a result, depending on whether the input voltage is below or above the threshold level. This level is adaptively set to be halfway between the received 0 and 1 levels. Thresholding is necessary because the received signals practically never have a square-shaped form. The second step serves to separate out the information bits from the thresholded version of the whole video signal. This is done by a shift register and a decoder that detects the bit synchronizing sequence 010101 . . . at the beginning of a data line, and effects the synchronization of the local clock. Figure 9.3 shows the above-described operations. For more details on teletext decoder circuits, see, e.g., Money (1979).

Data Selection

The bit stream received from the previous stage is first converted into byte-parallel form. The key to this is the recognition of the byte framing code that follows the bit synchronizing sequence. The simplest form of data selection is found in the first versions of U.K. teletext. Each data line there has a fixed format, consisting of the page number, row number, and the characters of that row. (This was later changed to include the page number in the header row only.) Therefore, data selection can be done by a few hardware circuits, as shown in Figure 9.4. The scheme is centered around an address comparator, which compares the address keyed in by the user with the currently received page number. If the two match up, the subsequent characters are sent to the display

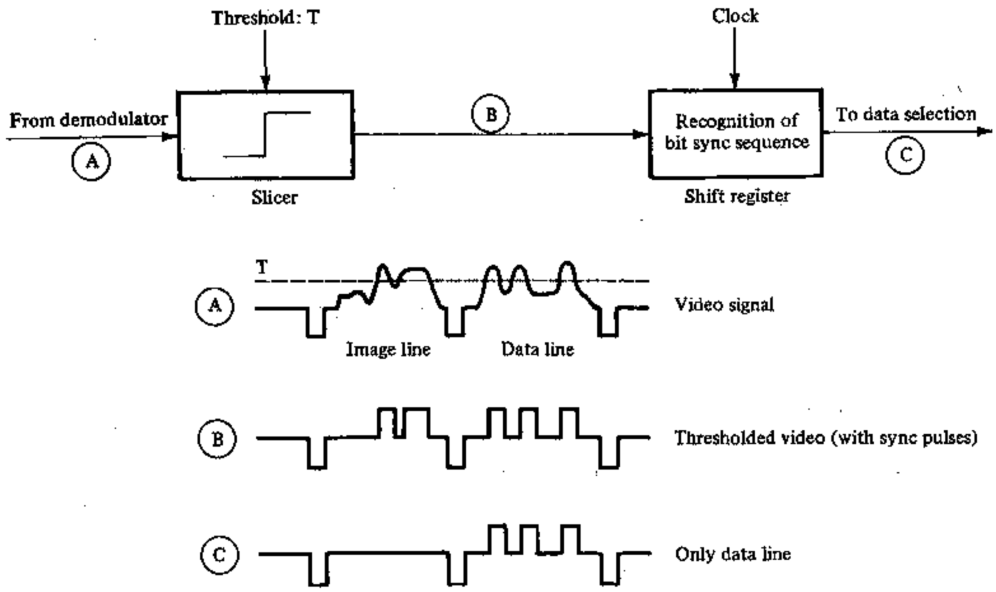


Figure 9.3 DOV data reception: slicing and bit synchronization.

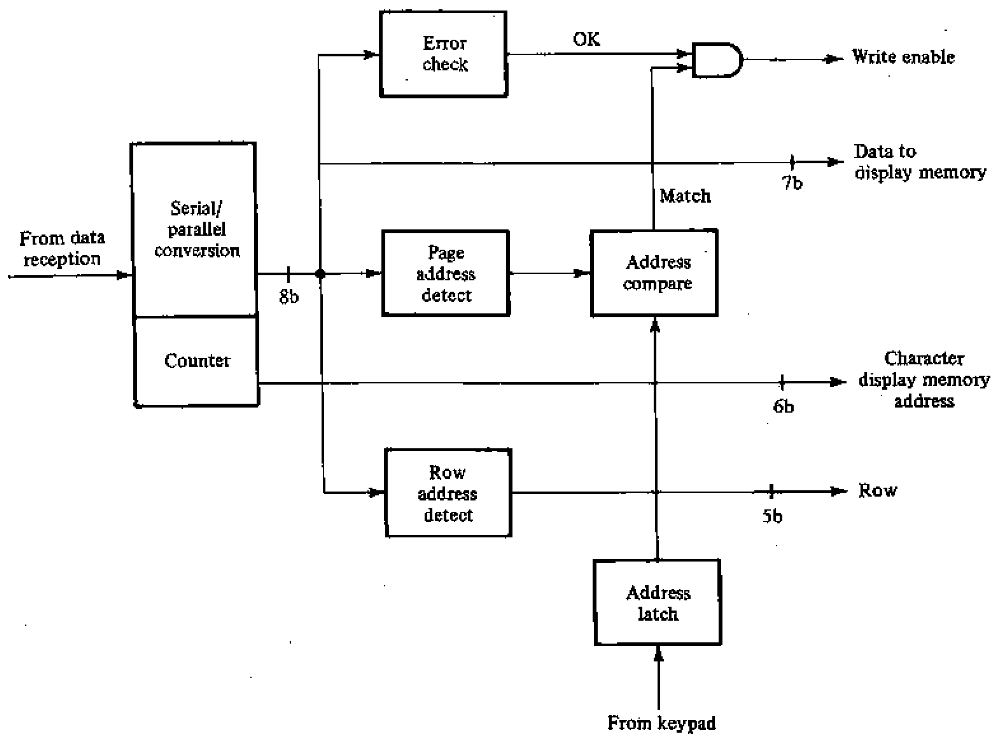


Figure 9.4 Data selection in fixed-format mosaic system.

memory; and because of the fixed-format transmission, addressing of memory words is simply derived from the current row address and a byte counter.

Errors are handled in a separate circuit. Single errors in data protected by error-correcting codes (usually, addressing information) are corrected automatically. On unrecoverable errors, writing into display memory is inhibited. Characters received with parity error are usually displayed as blanks.

Newer teletext transmission systems function asynchronously, and have much more complex header structures, including data channel identification, continuity checks, channel interleaving, and other facilities. Such complications cannot be handled uniquely by hardwired logic, so that a programmable processor becomes necessary. Once included, the microprocessor can also handle other terminal functions, such as interpretation of geometric codes and overall control of the terminal.

It must be emphasized that this control processor cannot be charged with *all* data selection functions. The reason is that the instantaneous rate at which data are received approaches 1 MB/s, which leaves only a few instructions/byte even with the fastest microprocessors; this is clearly insufficient. The data rate can be smoothed by buffers with VBI transmission; however, modern teletext decoders have to work with full field transmission as well. Therefore, it is inevitable that some preliminary data selection be done by dedicated hardware.

Control Processor

As already mentioned, function and complexity of the control processor will depend on the type of transmission and presentation system in question. Here is a partial list of things done (not necessarily in the same system) by the processor:

- input buffer management
- dialog with the user (handling of commands and messages)
- interpretation of control codes (e.g., proper placement of display attributes and DRCS patterns, switching of character sets)
- interpretation of geometric codes
- storing and recalling of macro sequences
- assisting in data selection and error checking.

Display Memory and Generator

Recall that there are two fundamentally different types of display memories, each suitable for different coding options: *character* memories (also called *page* memories), used with the mosaic option, and *bit-plane* memories, used for the geometric and photographic options. The purpose of the memory is the same in both cases: to hold data necessary for the display of a screenful of videotex information. Its content is completely read out 30 (or 25) times per second by the display generator while synthesizing the video signal.

One can, in principle, make Telidon terminals with character, rather than bit-plane, memories. PDIs are interpreted by the microprocessor to provide the best approximation of the image in terms of the available mosaic characters or in terms of "internally generated DRCS." The latter type of decoder has been demonstrated (see Tenne-Sens, 1982).

Page Memories

Character memories lead to the most economic realizations of terminals because only seven or eight bits have to be stored per mosaic cell (in the simplest case of serially coded attributes). A few additional bits are required for parallel attributes. Conceptually, thus, the character memory can consist of several planes (matrices of 24 rows by 40 columns), the first containing character codes, the second parallel attributes, and the third eventually holding markers of attribute changes (see Chapter 7).

The internal structure of the memory depends on the type of integrated circuit used for implementation. In one system, the Multitext (fabricated by Mullard, Ltd.), two standard 1 kb × 4 static RAM chips are used in parallel (each chip has 1024 words of four bits each; only three bits of the second chip are used).

The externally supplied address consists of 11 bits (five for row, six for column address). These external addresses are transformed by simple arithmetic manipulations into the 10 bits required to address the chips (see upper part of Figure 9.5).

Access time requirements for the page memory can be calculated if one realizes that for each TV scan line participating in the display of a given row of characters, all characters of that row have to be accessed once. This yields, for NTSC and 40 characters/row, a cycle time of roughly (60 μs)/40 = 1.5 μs. On writing, the data rate of 1 MB/s yields 1 μs/write. Thus with a 500-μs memory the write and read operations can be interlaced. The fact that both reading and writing are sequential can be used to reduce access requirements (by accessing several cells at once).

The economy that can be realized with mosaic-oriented page memories is partially offset by the complexity of the appropriate display generator. Note that only codes, and not the actual mosaic shapes, are written in the locations of the memory. The set of displayable alphabets (pixel patterns of G-sets) must be stored in a further memory called *dot matrix memory*. It contains one bit per displayable pixel, and is implemented in read-only memory, except for DRCS, which of course requires writeable memory. It is essentially a hardware embodiment of the code table (precisely, its G-set), as shown in Figure 9.6. Each character is stored as a matrix of given size (say 7 × 9 dots). The address has again two components: character address (seven bits coming from the display memory, identical to bits $b_1 \ldots b_7$ in the code table) and dot line address (coming from timing circuits). The output from the memory is the bit combination corresponding to the chosen dot line in the chosen character, and is used to modulate the beam while traversing the character position on the screen.

Physically the dot memory for a G-set has to contain 96 × 9 = 864 words of seven bits (with 7 × 9 cell size), realizeable again with two ROM circuits 1 kb × 4, with a similar address transformation logic, as in the display memory. The simplified internal

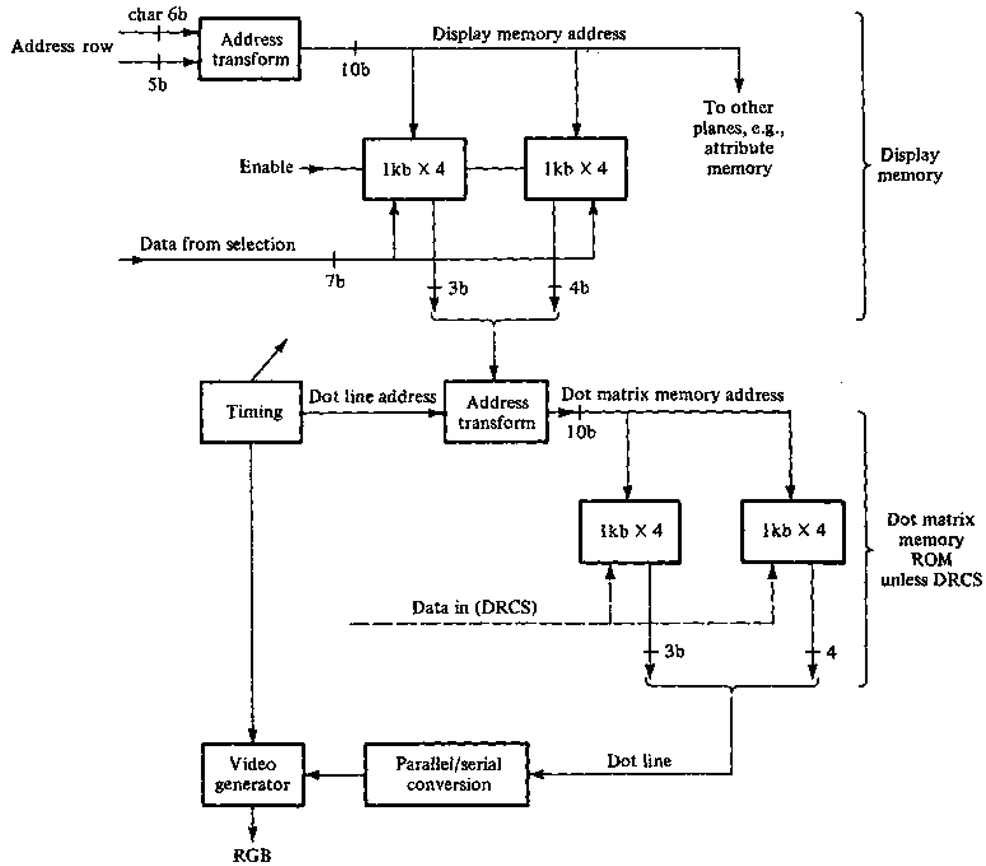


Figure 9.5 Structure of display and dot matrix memories.

structure of the display and dot matrix memories with the above-used parameters are sketched in Figure 9.5. The required cycle time for this memory is the same as for the display memory, about 1.5 μs. In practical implementations the dot matrix memory, display generator, and various circuits for implementing the attribute controls are integrated within one unit, such as the SAA5050 LSI chip of the Mullard system (see Mullard, 1978).

Bit Plane Memories

The cascade of display and dot matrix memories is essentially a two-level memory in which the upper level generates addressing information for the lower. The dot patterns can be likened to graphical "macros" invoked by "names" stored in the upper level. In contrast to this, geometric and photographic coding are not based on the repetitive use of a limited set of encapsulated patterns. In these options a one-level display memory in which each displayable pixel is explicitly represented (addressable), is more appropriate.

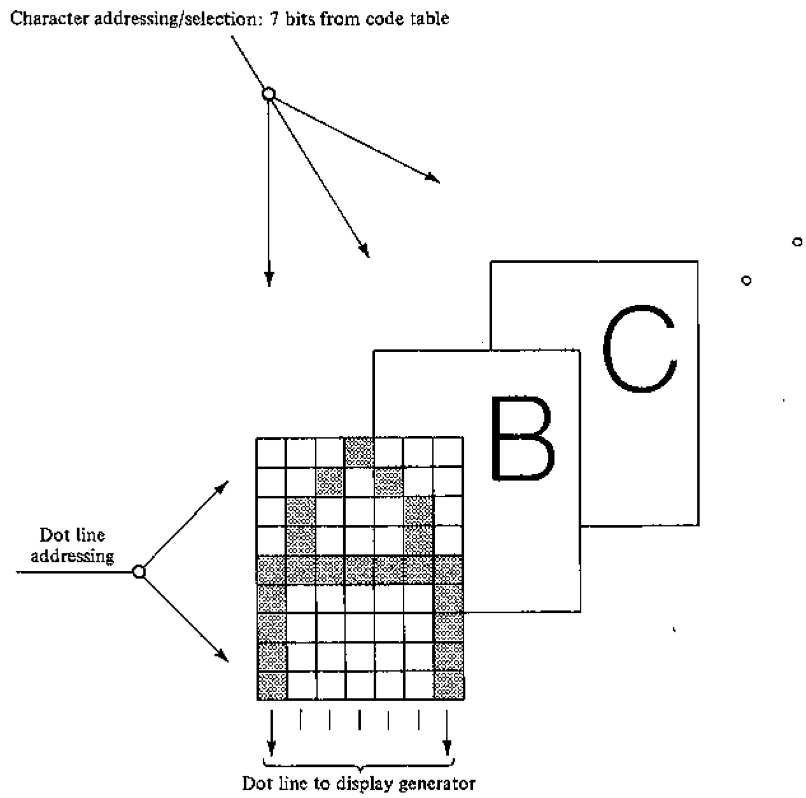


Figure 9.6 Conceptual view of dot matrix memory.

It should be realized that the "identity" of pixels is now merely the sum of their attributes; there is no shape to distinguish between them as there is in the case of mosaic characters. The attributes are inherently parallel. This is reflected in the structure of bit plane memories: conceptually, they consist of bit matrices (planes) with dimensions identical to the pixel dimensions of the display. For example, current medium-resolution implementations of Telidon terminals define 200×256 pixels. The number of bits representing a pixel available for the attributes will determine the range of colors, intensities, and other variations in the resulting image. Currently fabricated Telidon terminals have four bit planes. From the hardware point of view it must be taken into account that a whole row of, say, 256 pixels must be accessed by the display generator every 60 μs. Thus physical access is organized not on an individual pixel basis but rather in terms of suitable size blocks of horizontally adjacent pixels. On writing, access is essentially random, as determined by the sequence of geometric shapes generated by the microprocessor. However, the speed of writing into a bit plane memory is not bound to the display scan rate.

Geometric and photographic coding options are applicable with much smaller memories than would correspond to the resolution of the display device. For example, even a standard page memory with parallel attributes can be made into a 24×40 pixel

Terminal type	Display memory	μ-processor	Cost of memory (0.2¢/B)
Mosaic serial attributes	960 B	No	$2
Mosaic parallel attributes	≈2 kB	Yes	$4
Mosaic low-resolution geometric	≈2 kB	Yes	$4
Mosaic + geometric (200 X 256 X 4) + photographic	≈25 kB	Yes	$50
High resolution (1024 X 1024 X 8)	1 MB	Yes	$2000

Figure 9.7 Cost comparison of different types of display memories.

bit plane memory by connecting it directly to the video generator (i.e., by bypassing the dot matrix memory shown in Figure 9.5).

Figure 9.7 shows comparative display memory requirements for a range of terminal implementations. (See Bown, 1980, for an excellent discussion of terminal design alternatives). It must be emphasized that there is a great deal of controversy over cost. Some less popular memory parts are actually increasing in price. Some memory prices are artificially low due to dumping. Only the most popular and mass-produced configurations decline steadily in price.

The concept of the *multipage decoder* (locally storing multiple pages) adds an interesting twist to the question of memory cost. A given number of dollars spent in memory will buy more pages of storage in a system that requires fewer bytes per page.

The merits of enhanced mosaic systems (such as those of the CEPT proposal) and basically geometric systems (PLPS) are being hotly debated. The arguments advanced by the proponents of the first approach are mainly based on two points: (1) the memory requirement is an order of magnitude lower, and (2) enhanced mosaic graphics and DRCS provide sufficient flexibility and display quality. Proponents of geometric coding point out (1) that the disadvantage of larger memories and complex processing tends to fade away with diminishing hardware costs and with the advent of a "decoder on a VLSI chip," (2) that once the terminal has a sufficiently large bit plane memory, all three coding options can be naturally incorporated, (3) that geometrically coded information is independent of display resolution and even of display technology, (4) that scrolling is easier to implement, and (5) that general compatibility with techniques used in computer graphics can be used for enhancements, such as animation, three-dimensionality, and other manipulations of images. This last observation points in the same direction as do some recent developments in databases for videotex, in which after a period of specialized work the issues are now being placed in the larger context of existing database technologies.

Reference Terminal

The actual, quite hectic advance in the state of the art of videotex is accompanied by a proliferation of coding schemes, ad hoc proposals, ill-defined standards, extensions, and add-on features, gyrations whose purpose is to reconcile mutually inconsistent principles (not to mention subjectively and politically motivated and not-quite-scientific opinions). The inevitable net effect of all this is frustration. It might be reassuring to call things by their names, and to see the key principles that are involved in many designs filtered out and summarized in what might be called a "reference terminal" (which will never become a real terminal because there is already too much pressure for compatibility with existing code structures and terminologies).

The basic components (see Figure 9.8) are a processor with attached RAM and ROM, and a bit plane display memory whose content is continually displayed. The incoming serial code is always interpreed in one of six modes:

- direct mosaic
- direct geometric
- macro mosaic
- macro geometric
- mosaic macro definition (MMD)
- geometric macro definition (GMD).

The direct modes cause the processor to process incoming information and deposit it into the display memory, much as in geometric and photographic options of Telidon. The macro definition modes do not affect the contents of the display memory; in MMD, named pixel patterns (objects) are placed in the processor memory. Attributes can be defined for individual pixels or for the whole object, and some attributes may be missing at definition time. Position and size are also considered to be attributes. In GMD, strings of incoming code are named and stored, possibly with some parts missing.

Finally, macro modes would be used for the "expansion" of previously defined macros. As in conventional languages, this would be done at the moment when the macro name is encountered in the input string. Such a macro call can be accompanied by a set of attributes (parameters) that would complement or override those in the macro definition. Thus, at execution time one might conveniently change the color, position, etc., of predefined objects. Expansion of mosaic macros results in copying a pixel pattern into the display memory; geometric macros are executed as though substituted in the input string (with the appropriate parameters).

By going into further details at this point we would run the double risk of too much speculation and of reinventing the wheel. We conclude this excursion into the "possible" with a few remarks:

1. There is no essential difference between the photographic and mosaic options (in the sense used throughout this book), other than the fact that the first refers to the direct mode, and the second (especially DRCS) to the macro mode. Standard G-

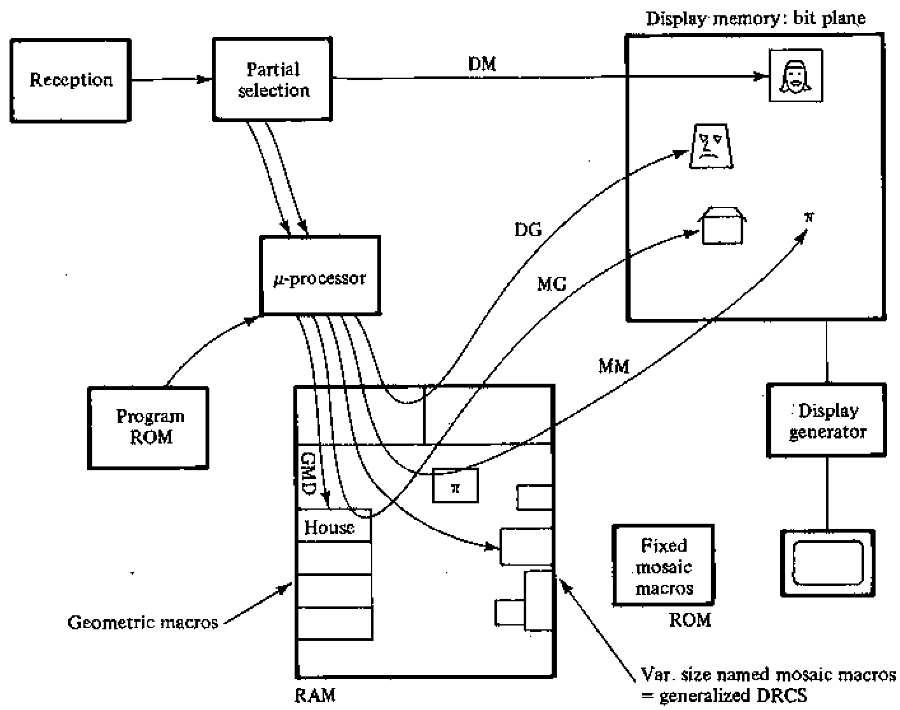


Figure 9.8 Reference terminal. The six modes of operation and the data paths involved are: direct mosaic DM, direct geometric DF, macro mosaic MM, macro geometric MG, mosaic macro definition MMD, and geometric macro definition GMD.

sets stored in ROM are in effect built-in system macros. It would be useful to recognize this fact and to treat the two uniformly.

2. Macros as defined in PLPS are in effect geometric macros except that no provision is made for parameters, which removes some flexibility.
3. The Reference terminal basically also covers the Captain coding scheme.
4. The Reference terminal is comparable in complexity to a full PLPS terminal.
5. More attention should be paid to unified handling of attributes, which might be based on constructs such as types in Pascal.
6. Similar ideas are involved in the graphic language used in the VIPS information provider system (see Section 9.3).

LSI Technology in Terminals

We have repeatedly stressed that the cost of user terminals is a critical factor, and that a key element in bringing down the cost is the availability of specialized LSI components fabricated in high volumes. A few manufacturers have already judged that this chicken-and-egg problem is sufficiently near to resolution to permit venture into the fabrication of chip sets suitable for both teletext and two-way videotex terminal controllers.

An example of a flexible set widely used in Europe and USA is the Mullard Multitext system (see Mullard, 1978; Beakhust and Gander, 1979). The components permit the building of different configurations, including remote control from a single keypad of the TV set, teletext, and viewdata. The system contains no microprocessor, and its teletext part is compatible with the fixed-format serial attribute of U.K. teletext. However, its remarkable flexibility permits it to be used in systems like the Virtext and Virdata teletext systems running on NTSC and employing gearing to conserve the 24 row by 40 character format.

Figure 9.9 shows the block diagram of the Multitext system. The more important LSI circuits and their functions are:

- TAC, teletext data acquisition and control
- VIP, video input processor
- TIC, timing chain
- TROM, teletext read-only memory (dot matrix memory)
- VAC, viewdata acquisition and control module
- LCU, line-coupling unit.

Another set of similar circuits made by General Instruments is described in McDonald (1980). It has a microprocessor-based controller and can store up to six teletext or Prestel pages. In 1982 Texas Instruments (France) announced the production of a VLSI chip to decode PLPS.

9.3 SERVICE-PROVIDER TERMINALS (SPT)

Features

Features of SPTs are somewhat opposite to those of user terminals, which are made to assist page editors and graphic artists in creating and effectively managing large numbers of pages. The main economic effect of SPTs is to reduce specialized labor requirements for page creation. To create a complex graphic page with primitive tools (e.g., through manual assembly of PDIs) may easily take a workday; the same job done on a modern SPT would take perhaps 30 minutes. In terms of wages (of, say, $100/day) this represents a saving of almost $100/page, thus amortizing a $20,000 SPT after the first 200 pages (done in less than 13 days)! Even if this calculation is oversimplified, it illustrates the point that although SPTs tend to be sophisticated and quite expensive, the cost is quickly made up by increased productivity.

There are three broad classes of SPTs: (1) stand-alone systems, which contain all facilities for creating and storing pages, (2) computer-dependent SPTs, which rely on a supporting computer for processing and filing functions, and (3) user terminals enhanced by additional hardware or software components.

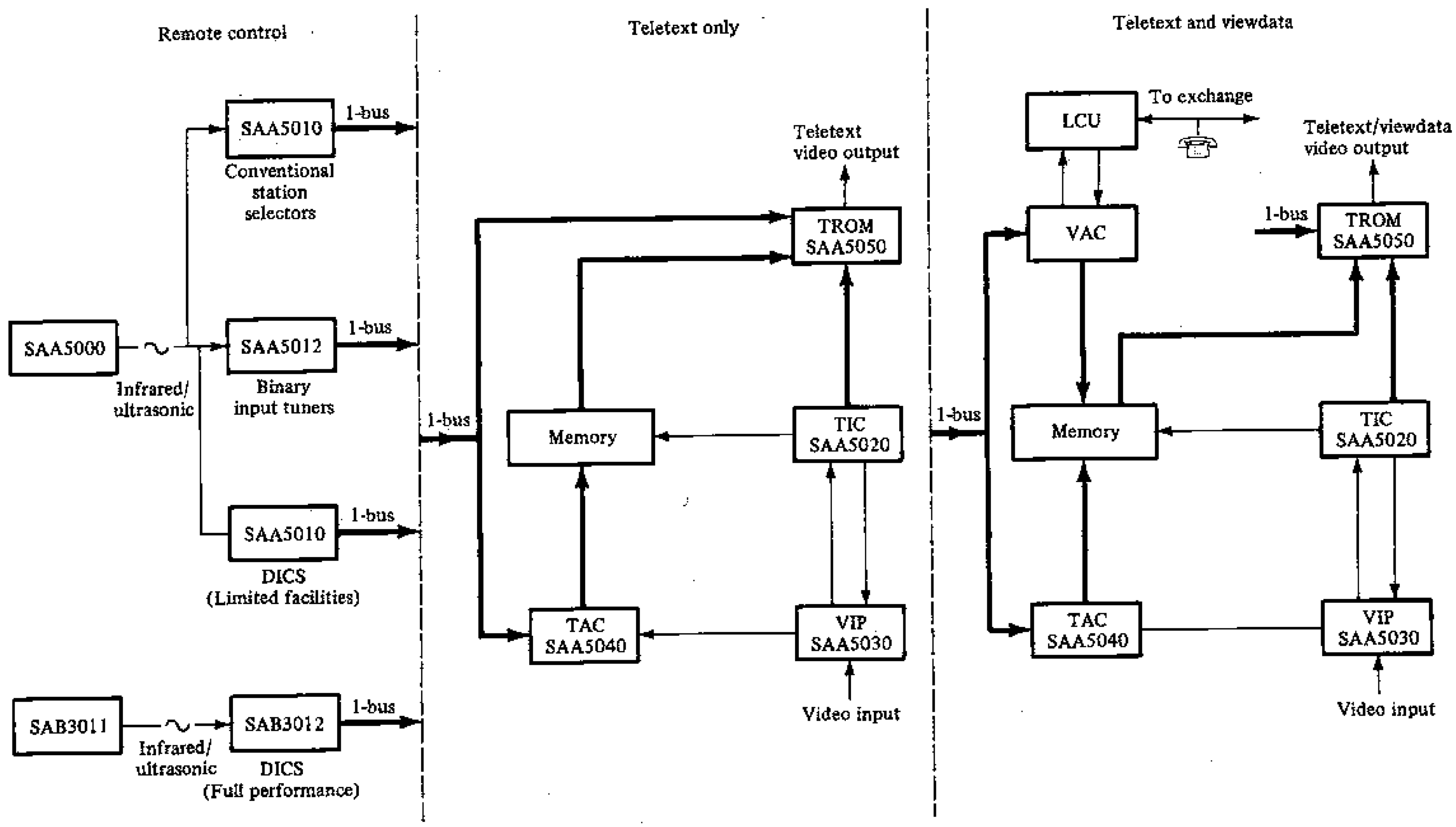


Figure 9.9 Block diagram of Mullard's Multitext system. Source: Mullard, Ltd.

The functional components generally present in SPTs include:

- text editing facility, equivalent to standard word processors
- graphic editor, enabling one to interactively compose, correct, and update graphical material; implementation and features depending on the preferred graphic coding option (mosaic or geometric)
- page editor, permitting the combination of graphics and text, and placing of logos and numbers in pages
- filing system, permitting local storage, cataloging of pages, and transmission of these to a database computer (this communication being an important standard issue)
- input facilities for text (e.g., from wire services, other databases or optical character readers) and graphics (tablets for handmade sketches, digitizing cameras for pictures and DRCS fonts).

There are many SPTs on the market today. We will describe here the VIPS developed by Bell Northern Research in Canada.

VIPS (Videotex Information-Provider System)

VIPS (see Figure 9.10) is a stand-alone system, designed for the Vista field trial in Canada, for which a database of about 100,000 pages had to be created (see Chitnis and Papp, 1982). Figure 9.11 shows the hardware configuration. The central element is a Cromemco System III microcomputer with built in floppy disk drives. The system is controlled from a keyboard/VDU terminal. The VDU displays command menus, messages, and user input. The page being edited can be continuously displayed for visual inspection on any PDI-compatible display terminal—e.g., on a standard TV set with RF (antenna socket) or RGB input. Thus the resolution, color effects, etc., can be immediately checked on the intended type of receiving terminal. Text of external origin can be received through an RS-232 interface. A digitizing tablet makes entering sketches and coordinates (points) easy (although this can also be done from the system console). A second function of the tablet is its programmable keys. Any sequence of system commands can be programmed under any "soft" key, and executed upon pressing it. The video camera serves to capture outlines of objects.

The VIPS software consists of the following modules:

- text editor (standard word processing functions)
- graphics editor
- information-provider language
- local database
- transmit
- auxiliary software (to receive text from an external source and to control the optional printer).

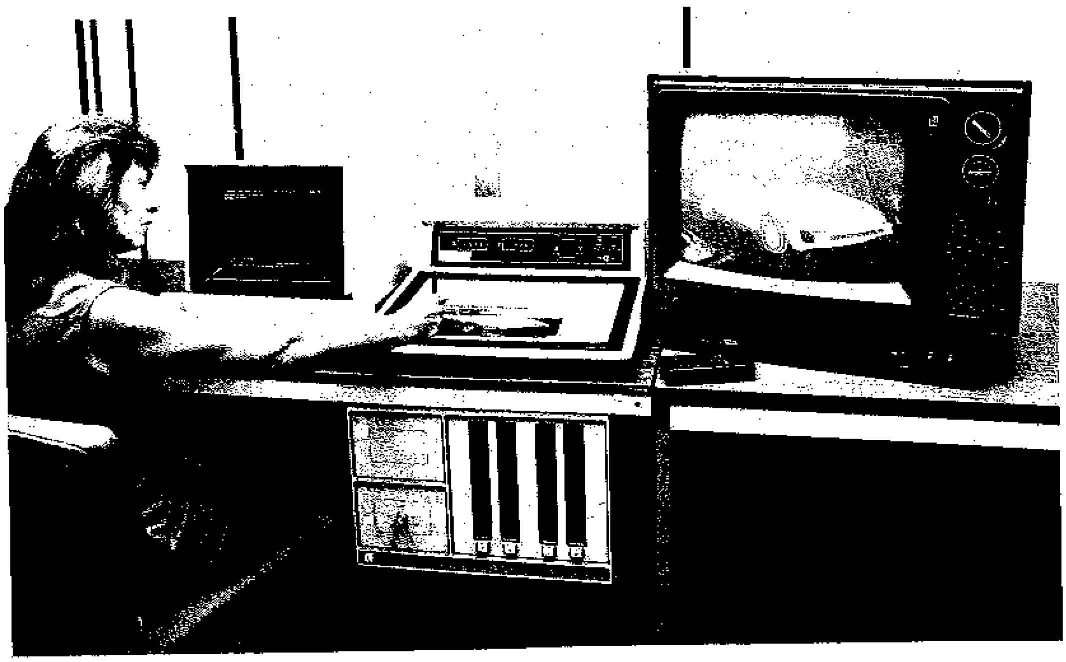


Figure 9.10 VIPS service provider system (Bell-Northern Research).

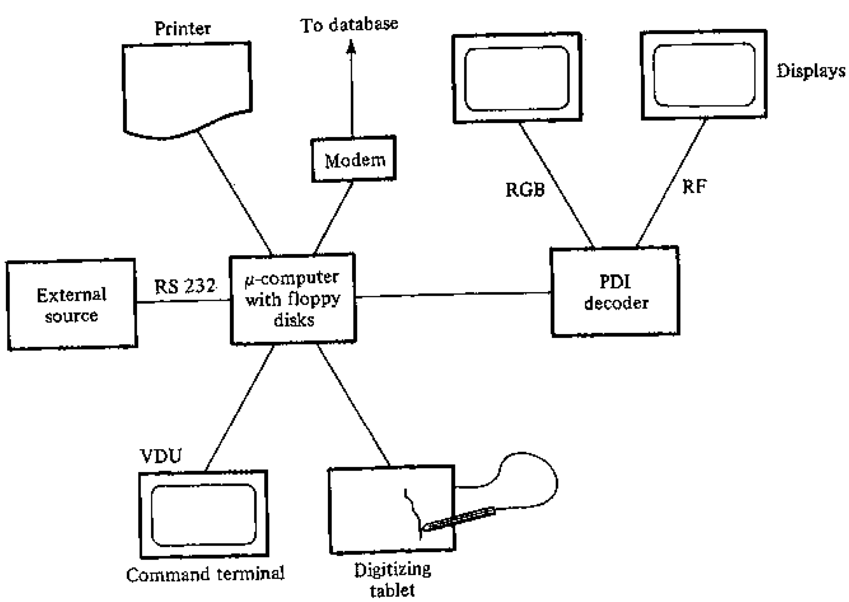


Figure 9.11 VIPS hardware configuration

The largest and most interesting of these modules is the graphics editor. In effect, it is a powerful language based on PDIs. Its main feature is its ability to create named *objects*; an object is any combination of PDI primitives and other previously created objects. Their structure is represented and manipulated in an internal language that is translated into PDIs (or other graphic language) at display time. Any depth of nesting of objects within objects is permitted; a change in an object's description is automatically reflected in all of its occurrences in other objects. Alternatively, such occurrences within embedding objects can even be changed individually. Thus, an artist can create a house (object) composed of the roof (object) and walls (object), and then a village (object) by combining multiple copies of the house. Then he can change in one operation the color of the roofs of the whole village, or he can modify any attributes of individual houses.

All declared objects are stored in a library. Images and objects are internally represented with a resolution of 1600-by-1200 pixels (6 times the resolution of TV-based Telidon). Here is a short list of operations available on objects:

- catalog (name an object and put it into the library)
- remove (an object from library)
- delete (an object from the screen)
- move, rotate, and duplicate (objects on the screen)
- sequence (change the display order of component objects)
- scale (from 0.01 to 100 times the original size)
- change object (e.g., change color of all houses in an image)

The information-provider language is a tool used to integrate text and graphic information into final pages. It also includes facilities for formatting unformatted text strings into lines of specified length and spacing. Resulting pages are numbered and automatically placed into a tree-structured local database. Menu selection pages are formatted automatically.

The transmit module initiates sending of any part of the local database to a remote service computer, to be integrated into its database.

VIPS is presently not able to handle mosaic graphics and photographic images, although these extensions can be added at moderate development cost.

Part IV THE APPLICATION LEVEL

Having discussed the ways and means of bringing information pages to the screens of TV sets, we arrive finally at the application layer of OSI. Our goal is not to treat individual applications and their characteristics; this is done elsewhere (see Chapter 2). The three chapters in this part of the book are about the support structures necessary to *implement* applications: databases used in videotex and teletext, gateways to access services running on external computers, and finally, service computers and systems, considered in terms of their architecture.

10
Databases for Videotex

10.1 A BRIEF REVIEW

To put the database facilities typically found in videotex in proper perspective, first we very briefly review the range of database architectures that are in general use today. The architectures can be divided into two large groups: *formatted* (record-oriented) and *textual* (unformatted) databases. The basic information units contained in and managed by the former are records and files. There are three well-known structural models of formatted databases: hierarchical, network. and relational. Typical applications are in business and enterprise-oriented systems; the arch-examples are the "personnel" and "part-and-supplier" files. Search mechanisms typically work at the field and record levels; queries into formatted files typically produce sets of records satisfying specified search criteria. Date (1981) is an excellent reference for further study.

Textual (e.g., bibliographic) databases deal with *documents* more or less in their original form, such as newspaper articles or their abstracts. There are two ways of accessing information in such databases: either through *indexes* (lists of keywords characterizing the contents on the documents), or through sequential searches of the documents themselves for given word patterns and their combinations. The user communicates with the database system through an interface language (query language) incorporating all necessary commands and parameters.

In another type of database, called *knowledge bases,* information is contained in highly coded abstract forms. Queries into knowledge bases are processed by complex programs, often based on artificial intelligence techniques whose goal is to find the most appropriate response. The meaning of queries themselves often has to be clarified by a dialog between the user and database system. The information sought for is not found prepackaged and ready for output, but has to be synthesized while the query is processed. Examples are certain medical databases capable of accumulating and using the clinical decision criteria of expert physicians in order to advise practitioners in their diagnostics.

The main difference between the three kinds of databases can be traced to the organization and meaning of information with which they deal. Formatted databases, which are at one end of the scale, have simply structured and highly repetitive sets of items (employees, parts, orders); knowledge bases, at the other end, may for example contain experts' criteria for recognizing a rare disease, criteria that may be highly individual, intuitive, and hard to formalize.

None of the above three approaches to database design is quite appropriate as a model for the mass access typical for videotex, for two main reasons. First, their query languages are too complicated for uninitiated users, and, second, searches may be too expensive in terms of computer resources, preventing the simultaneous use of a facility by hundreds or thousands of subscribers. However, there is no reason why such databases could not be accessed by videotex users on a more selective basis through gateways.

10.2 PAGE-ORIENTED DATABASES

The following discussion is based on Tompa et al (1981). Databases developed for the first versions of all major videotex systems are remarkably similar in that they use *pages* as logical units of information. A page was originally meant as both a logical and physical unit (displayable on one screen). However, a further distinction between the two meanings has proved to be useful: a page may consist of several *frames* (screenfuls of information). The internal representation of frames, used to maintain the information in computer memory (records, blocks, sectors, etc.) is a generally unrelated implementation detail dictated by the supporting data management system.

The second peculiarity of videotex databases is that they incorporate indexes and data in the same structure; in effect, index information is sometimes indistinguishable from data. This makes searching in the database a relatively straightforward task.

The underlying data structure in current videotex databases is a rooted tree of page nodes (see Figure 10.1), each node serving as the root for a number of subtrees. The content of any page can include text and/or graphics and can be used to convey application information, indexes, and routing prompts (instructions about what to do next). Pages are numerically identified by their position in the tree: for example, the root is page 0; its descendents are numbered from 1 to 9; the descendents of page 1 are pages 11, 12 . . . , 19, and the descendents of page 12 are pages 121, . . . , 129. A user can request a page directly at any time by entering its number via a keypad. Note that in this system of

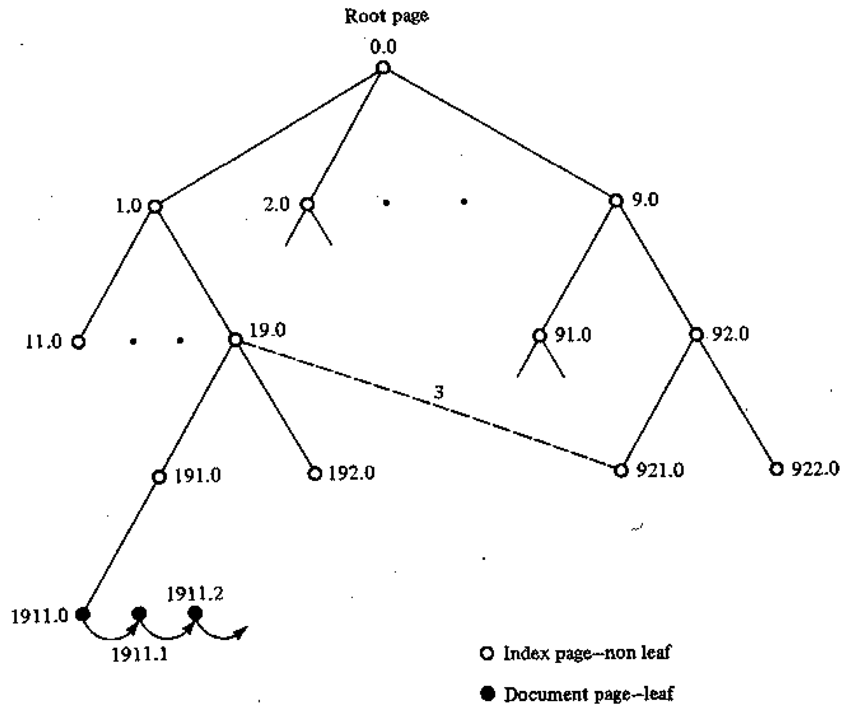


Figure 10.1 Page nodes in a tree-structured database. Nodes are numbered as in Telidon. a cross link from page 19.0 to 921.0 is shown with dashed lines; in this case page 193.0 does not exist: page 921.0 is taking its role, transparently to the user.

labeling, page numbers have a double role: they identify both the page *and* the (unique) way leading to it from the root. This may sometimes help the browsing user to maintain his orientation.

Providing several frames of information for one page has taken two distinct forms. In Prestel, any node (page) in the tree is a sequence of one to 26 frames, distinguished by the appending of an appropriate letter of the alphabet to the page identifier. Direct access to a page results in display of the primary frame (labeled A), the secondary frames being reachable by subsequent sequential access only. Similarly, in Captain each page has one to 10 frames (distinguished by an appended digit). Telidon allows multiple frames at the leaves of the tree only, thus distinguishing between so-called index and document pages (the content of either is under complete control of the information provider). A document page can consist of up to 1000 frames identified by one, two, or three digits that follow the decimal point in the page identifier (e.g., 122.0, 122.1, . . . , 122.999), and, unlike the situation with Captain, Prestel, and Télétel, any frame can be directly accessed by entering its numeric identifier. (For consistency, a Telidon index page has a decimal point followed by the digit 0 at the end of its identfier—e.g., 12.0.)

The tree structure also serves as a framework for allocating pages to information providers. For example, in Prestel and Captain each information provider is assigned a page number identifying the root of the subtree in which pages, frames, and their contents may be constructed under that information provider's editorial control. Both systems also reserve pages to contain general-purpose and index information controlled by the videotex system operator. Such localization of the information provider's pages simplifies the system operator's tasks of preventing interference among information providers and providing accounting (statistical and financial) to each information provider. The visibility of the tree structure through use of page identifiers also reinforces an information provider's identity to the user because it explicitly contains the corresponding root's page number.

Accessing the Information

Information contained in a tree-structured database typically can be accessed in two ways. (1) When the user knows precisely what information he wants, and on which page he can find it, all he has to do is enter the page number in question (e.g., to find the price quotation of his favorite stock). This access modality can (ironically) be supported by printed directories. Similar directories organized by subject or service provider can be found in higher-level pages of some databases. (2) When the user knows only vaguely the kind of information he wants (e.g., "some entertainment for tonight"), but has no idea where and under what heading (and whether at all) this information is found in the database, he will indulge in a search through the database, using a set of navigational commands provided for this purpose. They permit him to browse in the tree while trying to locate the desired information. The basic set of commands provided through the keypad are again similar in all existing systems (but sufficiently different to preclude direct compatibility between Prestel, Télétel, and Telidon). Section 10.3 discusses this in further detail. Navigational commands are either *absolute* (the target page is independent of the actual position in the tree) or *relative* (with respect to the current page). This is illustrated in Figure 10.2. Roughly, absolute and relative commands are designed to fit situations (1) and (2) above, respectively.

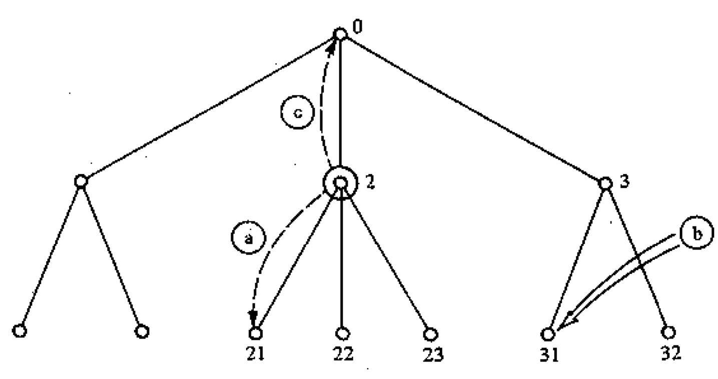


Figure 10.2 Effect of navigational commands: (a) relative command (menu selection no. 1); (b) absolute command (page no. 31); (c) immediate ancestor of page 2.

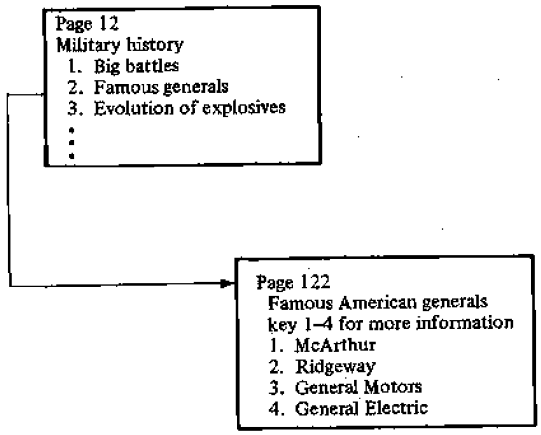


Figure 10.3 Non-leaf page with index and information content.

The above method of search is often said to be *menu-based* (or multiple-choice) because a typical (non-leaf) page displays the list of its descendants with short descriptions, one of which is selected by a relative command. The contents of a page (such as in Figure 10.3) can have both index and information content (e.g., answering the question: "Who are the most famous American generals?").

Problems with the Tree Structure

Tree-structured databases work well as long as the information they contain fits comfortably into a unique hierarchy—which happens only rarely. For example, it seems unreasonable to require information about Greek restaurants to be listed exclusively under the Greek subheading in the restaurant subtree or under the restaurant subheading in the Greek subtree, thus forcing all users to adopt the same hierarchical ordering for retrieving the information. Furthermore, if users are given only one tree structure to access data, they will often be unable to locate required information without first traversing many irrelevant sections of the tree. As a result, the effective response time of the system and the effective financial cost to the user (resulting from charges for connect time, page access, and possibly communications) will be large despite relatively efficient individual page retrievals. Information providers will also have only limited opportunity to entice users to request related information elsewhere in the tree.

Cross-links. The problem is somewhat alleviated by allowing information providers to superimpose arbitrary directed-graph structures on the underlying tree. Rather than interpreting relative page numbers in the context of the underlying tree structure, an information provider can establish a transparent cross-link to any selected page. In building such cross-links, the information provider simply associates the absolute page identifier of the target page with a numeric label to be entered by the user. The page itself will usually contain a multiple-choice display that indicates the valid numeric labels together with content descriptors for the corresponding pages (see dashed line in Figure 10.1).

The use of such cross-links has the consequence that multiple paths can exist to any given page, and therefore a unique correspondence between page number and path no longer exists. This has led in some systems to a complete separation of a page's name and position in the database, permitting the use of names with mnemonic or symbolic significance.

Another inherent limitation of tree-structured databases is the fact that all available information is pre-formatted into fixed pages. One cannot ask, for example, for a page that contains a list of all cinemas showing Italian films, although this information may well exist, dispersed within the database. Yet it seems that this kind of "dynamic page creation," required in such cases and common practice in database technology, is within the bounds imposed by the "keep-it-simple-and-economical" videotex imperative.

Getting lost. Experiments have shown (See Telidon, 1981a) that people easily become disoriented or lost in larger hierarchies, especially when they do not know exactly what information they are looking for. Alternatively, they are often unable to find a piece of information, even if it exists in the database, because the meanings of menu choices often overlap and cannot uniquely characterize all information in a given subtree. Another important factor is the weakness of human short-term memory. It is very easy to forget what was seen two or three pages ago, yet an earlier page may contain some crucial information about the context of the present page. The use of backtracking commands, user-defined markers to indicate important points on the search path, or personalized page labels such as in the Captain database, does not quite solve the problem, which seems to be linked to the fact that page contents change discontinuously: a new page does not contain anything from the previous one. It may well be that some form of multiscreen display would help, showing simultaneously detailed information on a subject and the global position of the searcher in the data space. Herot (1980) and Feiner et al. (1981) describe systems based on this approach.

In spite of these and other deficiencies, there exist a large number of well-established applications for which tree-structured, page-oriented databases seem suitable. Common characteristics of these applications are that they are relatively small and that they feature stable question-answer patterns well-known to the general public (see Ball and Gecsei, 1981). In these services the user always expects to find the same kinds of information or options under standard headings, although the details may be varied by the information provider. A user should not be forced to traverse long sequences of selections; such traversal tends to strain his tolerance of the menu-choice interface, with its tedious and potentially disorienting qualities. All this boils down to the following moral: The key to the successful application of menu-based systems in videotex lies in conceiving the database as a collection of well-established stable services (with few cross-links reaching outside a service), and not as a single, large, and changing construct with evenly spread cross-links. Enhancements to the basic page and tree structure, as well as complementary approaches to database and interface organization, will be further discussed in Chapter 13.

10.3 THE USER INTERFACE

Commands available to users for database retrieval, along with others for logon, selection of service, etc., together form a simple interface language. Similarly, information providers dispose of a set of facilities permitting them to interact with a videotex database.

As mentioned earlier, data retrieval commands for tree-structured databases found in different systems are similar but not identical. However, there is a possibility of standardizing at least a core set of these (and some other system control) commands. Such standardization and the ensuing application-level compatibility between similar systems is clearly desirable, as expressed in CCITT Recommendation F.300 (see CCITT, 1980).

Before considering F.300 in some detail, note that three aspects of commands can be considered for standardization:

- command semantics (function)
- coding as seen by the user (symbols written on the keypad/keyboard)
- coding as seen by the database computer.

In order for a terminal of one system (say, Prestel) to be strongly compatible with a service computer of another system (say, Telidon), compatibility in all three aspects is required.

F.300

Recommendation F.300 is concerned with the semantic content of user commands. A videotex system is seen at two levels: the *system* (first level), which is a collection of *services* (second level). Here is a list of command functions to be considered (and partially implemented in major systems):

Control functions:

C1: clear unwanted entry

C2: interrupt action in progress (e.g., page transmission)

C3: send command to service computer

System functions:

S1: select a videotex service

S2: return to the entry point of service (e.g., database root)

S3: logoff

S4: logoff with billing information

S5: provide billing information

Functions common to all services:

F1: return to system level

F2: system assistance

F3: service assistance

Information retrieval functions:

R1: direct access to a page (absolute page request)

R2: relative page request (menu choice)

R3: next frame of the same page

R4: preceding frame of the same page

R5: retransmit last frame

R6: retransmit last page with changes that may have occurred in the meantime

R7: retrace previously seen pages (in reverse time order).

Prestel, Telidon, Télétel, and F.300. There is practically no consensus concerning the coding aspects of the above framework of commands. Figure 10.4 illustrates this by comparing some key sequences in Telidon (field trial version), Prestel, Télétel, and the corresponding functions from F.300. Note that Telidon has a number of functions that are absent from Recommendation F.300.

10.4 DATABASES FOR TELETEXT

The fundamental difference between interactive and broadcast videotex is inevitably reflected in the ways databases should be organized to take advantage of the two modes.

F.300	Telidon	Prestel	Télétel
C1	DEL	**	Annulation or correction
C2	F ■		
C3	■		Envoi
S1			*K* Envoi
S2	↑↑ ■	*O#	Sommaire
S3	F6 ■		
S5	F7 ■		*K* Guide
F1	F5.*n* ■		Connexion-Fin
F2			Guide
F3	F1 ■		Guide
R1	*N*. ■ or *N.M.* ■	**N*#	*K* Envoi or K_1*K_2 Envoi or $n_1*n_2*\ldots n_i$ Envoi
R2	*n* ■	*n*	*n* Envoi
R3	→ ■	#	Suite
R4			Retour
R5	‖ ■	*OO	Repetition
R7	← ■	*#	*Retour

N: Absolute page no.
M: Document number
n: 0, 1, ... 9
K: Keyword

Dedicated keys in Télétel:

Annulation	Guide
Correction	Suite
Envoi	Retour
Sommaire	Repetition
Connexion-Fin	

Figure 10.4 Comparison of some Telidon, Prestel and Télétel commands and their F.300 equivalent.

The main constraint of VBI (vertical blanking interval) transmission is its low page rate (e.g., 4 pages/second). This, in order to assure reasonable waiting times, limits the size of the broadcast cycle to a few hundred pages. This constraint is somewhat relaxed in full-field transmission, but reasonable cycle sizes (up to 10,000 pages) are still an order of magnitude smaller than the size of databases used in interactive systems. Note that the above restrictions relate to broadcast databases used in pseudo-interactive (on-line) mode, in which the user is waiting at the receiver for the page he has requested. This is the case in all first-generation systems.

In effect, the basic address format in the U.K. teletext permits inclusion of only up to 800 pages on a single VBI channel. However, the use of time codes allows for virtually limitless teletext page numbers. The on-line transmission cycle (see Figure 10.5(a)) consists of sub-cycles called "magazines" (up to eight in the U.K. teletext, each identified by the most significant digit of the page number), which differ in program origin and contents. Interleaved transmission of pages in different magazines improves the access time to the beginning of each page, but not to the entire page.

Transmission Sequence

Now it is being realized that there is another way of exploiting data broadcast systems—namely, by admitting off-line reception and storing of some selected pages. The tremendous potential of this approach stems from the inherent simplicity of broadcast mode and very high transmission capacities (a 4-MHz TV band can transmit 1.8 million pages per hour). On-line and off-line reception can be mixed by controlling the repetition rate of transmitted pages. A few hundred pages of general interest can be repeated in a short cycle, whereas less important or less frequently updated pages are inserted in hourly or even daily intervals into specific slots in the cycle, or according to a predetermined schedule. This applies equally to VBI, full-field, ISDN, or any other type of transmission, and permits for the dissemination of very large amounts of data.

Pages that are broadcast at different times and with different frequencies fall into four groups:

1. On-line pages (the normal broadcast cycle)
2. Scheduled off-line pages (captured and stored in the receiver's memory in standby mode); pages recognized either by a time code (that is, transmitted at precisely known times) or by extended page numbers; same mechanism (additional page identification facility in the page header) useable for both modes, as in the U.K. teletext (see Chapter 5)
3. Pages transmitted on demand (on-line or off-line); see also Robinson and Loveless, 1979
4. "Real-time" pages, which must be displayed at the moment they are received, such as newsflash, captions, and subtitles; two methods of immediate insertion available: after the end of the current page, or over a separate TV line reserved for that purpose (e.g., line 21 for captioning). The latter method evidently faster because there is no need to wait until the end of the current page.

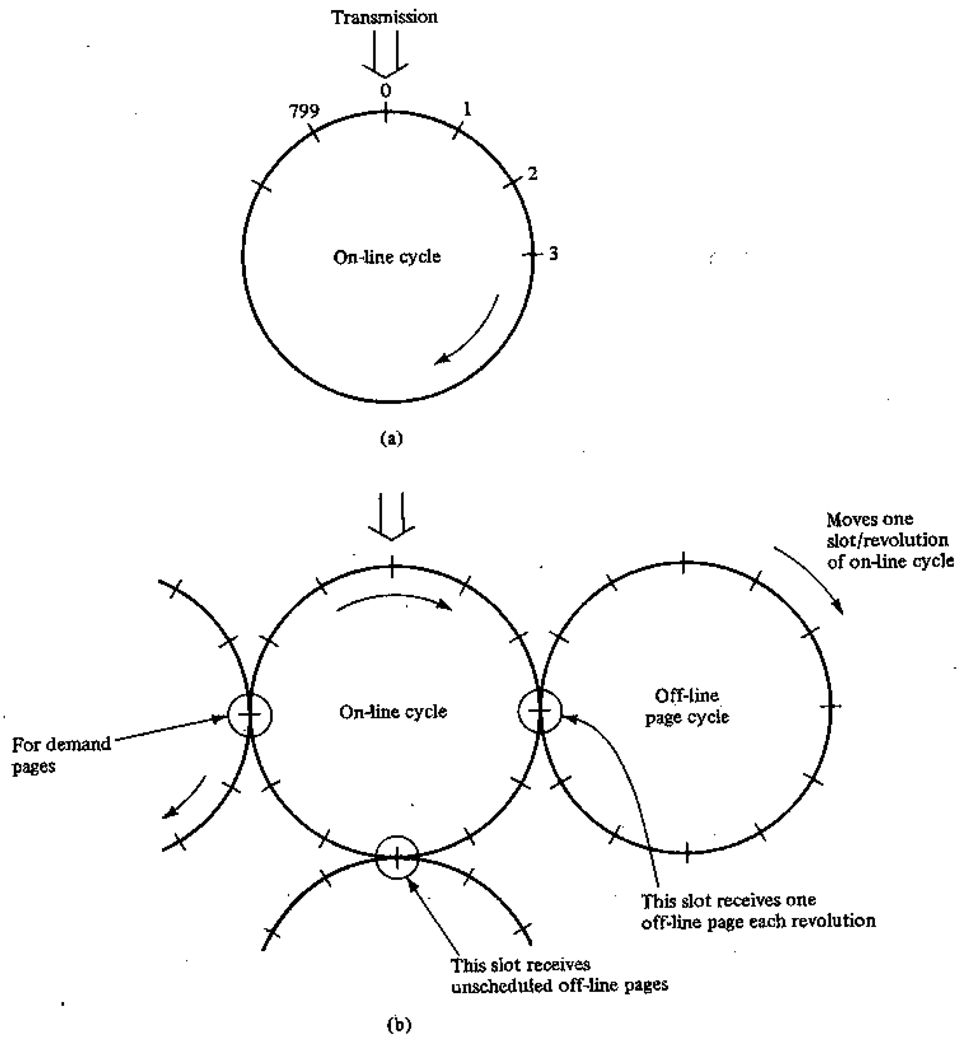


Figure 10.5 Schematic view of teletext databases: (a) simple cyclic online pages; (b) insertion of cyclic off-line, irregular offline and demand pages.

Figure 10.5(b) is a schematic view of some cycles and "sub-cycles" (possibly aperiodic) in teletext transmission. Other variations are also feasible.

Tree Structures

So far we have discussed the temporal aspects of teletext database transmission. We must now ask: What is the relation between the transmission sequence and the logical structure of the database as perceived by the user?

In principle, tree-organized databases can be transmitted over teletext simply by cyclically sending out all pages in the tree, each preceded by its number. The "user context" (information about the access history of a particular session) then has to be maintained in the terminal. This is necessary, for the correct execution of backtracking commands, for example. However, difficulties may arise if cross-links are also used in the database: if a relative command (menu choice) that leads to a page other than a descendent of the current page is issued, how can the terminal know the number of the target page? The problem can be resolved by including cross-link information in the header of each transmitted page—a solution implying a sizeable transmission and terminal hardware overhead.

Another issue is the size of the broadcast database: It may be too small for the tree structure to make sense; on the other hand, the tree structure may become ineffective for pure off-line reception from larger databases because of the lack of interactivity. (However, the combination of one-way and two-way transmission, as in the INDAX two-way cable system, is a viable possibility.)

Within the remaining limited range of sizes, an interesting optimization problem exists: given an (uneven) access pattern to the pages of a tree-structured database, what is the optimal transmission sequence in order to minimize the average access time? A solution is outlined in Chapter 15.

Captioning

Due to its nature, VBI transmission lends itself to a number of related applications serving to complement an underlying video program: newsflash, subtitling, and captioning. In these services, short text sequences are displayed on the screen simultaneously with TV images. Each service is identified with a page number.

In most cases, direct superposition of text over the image is not satisfactory because of the unpredictable contrast between the image and the fixed-color (say, white) characters. The standard solution is to display text in *boxed mode*—that is, in a (say, black) window of appropriate size, cut into the picture. Such a box is created by inserting the attribute controls Start Box (SB) and End Box (EB) at appropriate positions in the character stream. SB will cause suppression of the "transparent" full screen color until EB, which has the opposite effect, is encountered. The characters to be displayed are transmitted between SB and EB. In this way, boxes of any size can be created in any position on the screen. Note that unused positions on the screen do not require transmission of blanks; rather, cursor controls are used to skip such areas. All major teletext systems have provisions for captioning.

Newsflashes (when selected) are broadcast and displayed whenever a news item breaks. They remain displayed until the user clears the screen or a newer version is transmitted.

Subtitling and *captioning* are similar services designed to annotate TV programming such as films. *Open* and *closed* captioning should be distinguished. The former is "open" (visible) to everybody watching the program; the text is superimposed on the image in the studio and is transmitted as part of the video signal. Closed captioning is selective, transmitted over VBI, and receivable only with teletext decoders. Two important application areas of closed captioning are captioning for people with impaired hearing and multilingual subtitling.

From a technical stanspoint, it is important to distinguish between *live* and *stored* modes of operation. In live captioning the text is generated in real time—e.g., by a human translator or commentator—and displayed with a short delay of a few seconds. When typing is too slow (e.g., to capture real-time speech), special shorthand typing devices, whose output is automatically translated into a phonetic version of English, are used (see Green, 1979, for details).

A quite different setup is used for captioning pre-recorded material, such as films. Captions are prepared and edited in advance and stored sequentially on disk files. To solve the problem of synchronization—that is, of inserting the right caption at the right time (image)—a time code is added to each caption. The program source (e.g., video recorder) has a frame counter that indicates the elapsed time. When the code of the next caption matches the frame counter, VBI insertion is triggered.

Editing and other facilities for captioning are usually integrated into teletext computers (see Chapter 12). A number of countries in Europe, as well as Canada and the U.S., currently provide captioned television. In the U.S., activities are coordinated by the National Captioning Institute.

11
Gateways and Data Distribution

11.1 DATABASE DISTRIBUTION

In a nationwide public videotex system with many thousands of users, service cannot be assured by a single central service computer that holds the entire database. This was recognized when Prestel originally introduced five replicated databases in its network; the number has been increased in the meantime to about 20 (and subsequently decreased again). The problem that arises is how to organize the contents of individual databases, and provide for their cooperation and updating in such videotex networks. First we deal with the simpler case in which all participating databases are (strongly) compatible at the three upper OSI levels—that is, they present the same interfaces to the users and service providers. (This, of course, does not imply that they are supported by identical machines or operating systems).

The information content of a number of similar databases can exist in various relationships.

1. Replicated (identical) databases are probably the simplest to implement and operate. Yet there are potential problems: with simple updating from a unique master database via update messages to the component computers, absolute consistency between the copies cannot be guaranteed at all times. While this is of little or no concern in most applications, it can (and does) create controversy in areas such as stock price

quotations or betting, in which information is highly volatile. Several methods for consistent updating exist (e.g., locking updated pages until the update transaction is confirmed by all databases), but they generally require high overhead. This kind of system tends to waste memory space by duplicating information (e.g., weather, local advertisements, regional news) that may be of only local interest.

2. At the other end of the scale, databases can be disjoint, that is, have no duplication of information. This generally saves memory at the expense of increased communications. The principal problem here is to decide the optimal allocation of data to sites and the maintenance of associated directories (which give the locations of information). Another open question is whether the user should see one combined database (the physical distribution being transparent), or a set of partial databases from which he has to choose explicitly. If so, where should the directory of available databases reside?

A variant of the transparent scheme has been implemented at the Université de Montréal for the user-transparent interconnection of two Telidon databases. Certain pages of each database are actually "external pointers" to selected entry points in the other database. Here, invocation of such a pointer causes automatic call, login, and access to the other database. Then the user can freely navigate (using relative commands) in the subtree defined by the entry point. The link is automatically broken when a command would cause transgression of the limits of the subtree.

In the Télétel network the totality of (database and other) services is organized in two levels: the directory level and the service level. Every access machine holds a copy of a system-wide directory of services. When a non-local service is selected (this can be done from the directory tree or from certain pages in local services), the required connection is automatically established. Each service typically has a single entry point.

3. Contents of the databases, each containing a common core and specialized information of local interest can partially overlap; some of these pages may have the same logos and layouts, and may differ only in contents (e.g., weather).

Newer network designs (Prestel-PANDA, Bildschirmtext; see Chapter 6) incorporate database designs in which the page set residing in each component (e.g., access machine) depends on the momentary pattern of requests, much as in a cache memory.

11.2 ACCESS TO EXTERNAL COMPUTERS

Initial implementations of videotex provide access to large central databases created and maintained especially for videotex. But it has gradually become clear that this approach is too limiting and precludes many cost-effective applications. After all, once the communication channels to the users have been opened, why not use them as a window to the entire world of computerized information? Large amounts of potentially useful information already exist in various databanks and services that run on external computers (i.e., computers external to the videotex system in question). These services take many forms, run on a variety of equipment under different operating systems, and have differing security and integrity constraints. Automatic conversion of such databases to a videotex-compatible format is not a simple affair because of wide variations in display formats, internal data structures, search mechanisms, and interface languages found in external services.

The notion of *gateway*, which generally means the collection of functions and facilities used to access external (third party) computers, is receiving much current attention. Gateway functions may reside partially in the external computer and partially in the user's access machine. Alternatively, the gateway can be implemented as a dedicated processor which itself does not provide other videotex services; in this case it acts as an intelligent switch to remote computers.

Compatibility with External Computers

Gateways are familiar in the context of computer networks—e.g., between local area and long-haul data networks, or between different packet switching networks. What is unique in videoex gateways is the emphasis on higher-level (session, presentation, and application) concerns.

A videotex database and an external database (or service), accessed through a gateway, are generally incompatible. We can identify three areas in which differences might occur: the application, presentation, and session levels of the OSI model. The following discussion is partly based on Section 8.5 of *The Telidon Book* (ed. Godfrey, 1981).

At the *application level* are found differences in users' views of the data, as well as differences in the interface languages used to interact with the database. It is very difficult to mask these differences (that is, to achieve compatibility at this level), unless the external database in question has been designed with compatibility in mind. For example, it would be foolish to consider providing an interface program to make the ORBIT retrieval system behave like a tree-structured database. This kind of incompatibility is less of a problem than it might seem because it concerns human beings, who are capable, at least in principle, of adapting to (suitably explained) new interface procedures. However, it is highly desirable that provisions be made in the gateway for a minimal set of common commands and procedures, such as "Help," requesting directory information, or the interactive learning of a new language.

Consider now the *presentation* level. An external database may use character sets, codes, screen formats, row lengths, or graphic primitives other than those used by videotex databases and terminals. It is essential that some degree of compatibility be assured at this level, because it involves the terminal hardware (which cannot be easily changed).

In general, the difficulty involved in achieving compatibility at the presentation level depends on the structure of the data in the external database. Three cases can be distinguished.

1. If the data consists of unformatted character strings (such as conventional text), then the required format conversion would consist merely in generating a carriage return after each set of 40 characters. This can be done either in a gateway computer or in the terminal itself. Paragraphs can easily be detected and handled similarly.

2. External databases may contain formatted text, but not figures or diagrams. Examples are tables and responses to queries into record-oriented databases. The format of this information is generally known by the application; therefore it can be dynamically reformatted by conversion programs. This can be done either in the external computer (EC) or in the access machine (AM). In the latter case the process is simplified if, in addition to data, explicit information on its format (fields, length) is also provided to the AM. Such information may come from the EC and partly from an editing terminal in the AM (e.g., for colors). Orderly transfer of both kinds of information is accomplished by a "gateway protocol" (i.e., another presentation-level protocol).

3. Lastly, an external database may contain figures, diagrams or similar graphic material. Everything depends on the way in which this material is coded. For example, on-line transcoding into PLPS is reasonable only if the source code is not too different from PLPS (e.g., Telidon). Off-line conversion into PLPS from the CEPT or Picture Prestel format should be possible without too much difficulty, but the opposite is not necessarily true. A general method would consist in generating the source graphic image on a display screen, and re-encoding it photographically (with a TV camera). To date, little work has been done in this area.

Consider finally the *session* level, which usually deals with functions such as logoff and user identification procedures, option negotiation, accounting, transaction integrity, and the control of an ongoing session. Most of these functions can be made invisible to the user, as they are typically handled by programs in the gateway. This level should not cause insurmountable problems for achieving compatibility between videotex and external databases.

In conclusion, we point out that areas corresponding to the lower levels of the OSI model are practically invisible to the user. Still, mutually compatible protocols must be used by the videotex and external database computers.

Gateway Alternatives

There are several technical possibilities for providing access to external databases and other interactive services. In this section we consider the impact of these interface strategies on compatibility.

1. In the most general case, the EC software is not only incompatible with the videotex service, but it also contains no (add-on) component designed to interface with videotex. The gateway appears to the EC as just another of its terminals, and conversion programs (if any) must all reside in the gateway (See Figure 11.1). A simple but useful possibility in such cases is to provide for a transparent transport connection, and perhaps logon, between the videotex user terminal and the EC. Typically no compatibility in the application area can be expected. Thus, the user must know and use the command language of the EC, which will likely be more powerful and also more difficult to learn than that used in the simpler videotex systems. Obviously, the presentation aspects may be poor in this case.

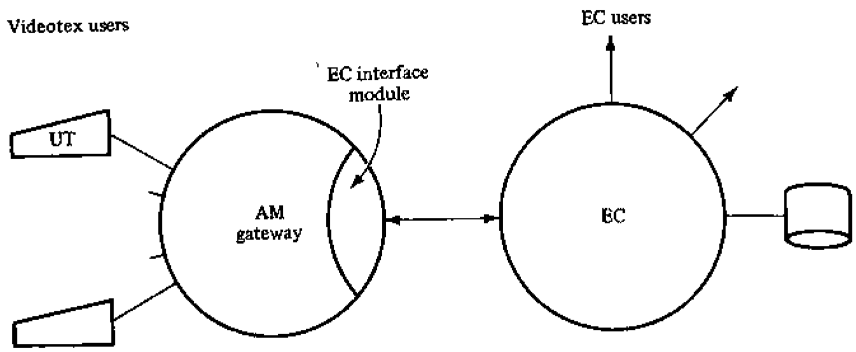


Figure 11.1 Gateway appearing as another user to EC.

A different solution was retained in the Vélizy trial system in France. There, while no changes are required in the EC, the AM contains a software module (called "front end") that is custom-programmed for the EC application in question. Typical functions include reformatting and dialog management (both with the user and EC). However, as also discussed in Chapter 6, the Vélizy system does not preclude incorporation of some interface functions in the EC, especially if done integrally with the application design.

2. Alternatively, even if the EC is independently designed, it may contain an add-on interface module for use with videotex customers (see Figure 11.2). This can make life much easier for both the users and gateway designers. The two interface modules together can be made to implement a number of functions at the three highest OSI levels. In Bildschirmtext these functions include limited application level compatibility (menu choices), and a specially designed protocol for connection establishment and data acquisition from the user in transactional applications.

3. Finally, the external database may be a third-party (externally controlled) or private—but videotex-compatible—database. If it is designed to the same standard, then compatibility can be easily achieved at all three levels. If the databases are merely simi-

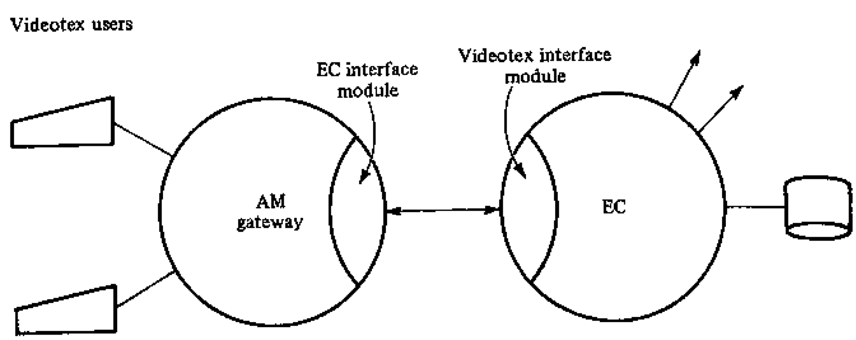


Figure 11.2 Gateway and EC cooperating through videotex interface module.

lar in structure (if, for example, one is Prestel-based and the other Telidon-based), then application- and presentation-level compatibility (especially in real time) are possible, but not simple to achieve.

11.3 EXAMPLES

Almost every modern videotex system features some form of gateway or similar function. We include short descriptions of a few representative examples.

Palme (1980) describes an interface program permitting existing computer applications (written for standard VDUs) to run with a viewdata terminal. Two problems are addressed:

1. It is necessary to replace the scrolling facility of a standard display unit. In order to avoid blanking the (full) screen before it is read by the user, the system displays a prompt whenever blanking becomes necessary. Blanking occurs only after the command has been given.

2. The viewdata terminal has a short line length (40 characters). Unformatted text lines are broken at the space that is closest to the middle of the line. This avoids systematic effects that occur in other methods, such as breaking always before (or always after) the middle. Tables are automatically split vertically between columns; the two halves are then displayed in an overlapped fashion (in different colors) with alternating lines from both parts. A simple example of this is shown in Figure 11.3. If text and tables are side by side, then only the text is broken (if possible).

The program can work even without explicit information on the data to be formatted; results can be improved by inserting a few control characters indicating, e.g., places

Before splitting

A	75	120	–	80	12
B	20	50	70	–	130
C	17	8	–	9	70
D	50	70	9	0	180

After splitting

A	75	120
–	*80*	*12*
B	20	50
70	–	*130*
C	17	8
–	*9*	*70*
D	50	70
9	*0*	*180*

Figure 11.3 Splitting columns of formatted text (italics indicate different color).

where color changes are advisable. Although far from perfect, this simple program is reported to work acceptably for most applications.

The next example (see Chauffault et al., 1982) comes from a different context. A specific application (a medical database system) had to be interfaced with three different terminals (video display unit, gray-scale videotex, and color videotex). These terminals were handled in an add-on module called "input/output machine" and residing in the application computer. This machine appeared to the application as a virtual terminal accepting commands of the form

EDIT (formatno, text, length, code)

where "code" stands for the terminal's response, "text" is the message to be displayed, and "length" its length. About 150 output formats were stored in the machine; each defines in terms of color, size, position, and other attributes the precise layout of the text to be displayed. High-quality displays could be obtained in this case due to the fact that the application explicitly communicated the desired format to the virtual terminal.

The Bildschirmtext/Prestel Gateway

Planners of Bildschirmtext (BTX) worked right from the beginning on the premise that their system would be a two-level network: all videotex functions were to reside in the first level (a network of access machines with databases), and the "rest of the world" in the second level. Importantly, a window was designed between the two levels: the gateway facility. The same approach has since been adopted by Prestel. In principle, any external computer in the rest of the world, satisfying certain interface requirements, can be accessed on behalf of a BTX user who can then sustain a dialog with the chosen external computer (see Gilbert and Taylor, 1981). There are three participants in this dialog: the user terminal, the access machine (BTX center), and the EC. The AM and EC communicate in terms of messages (blocks) according to a special gateway control protocol. The protocol as it stands now permits two kinds of services: data retrieval and data acquisition. The former permits page requests to be sent to EC and the responses to be routed to the user's terminal, much as in standard videotex (as seen by the user). Data acquisition serves in transactional applications to collect data through a form-filling procedure. The AM acts as an intermediary in the exchange between the user terminal and the EC. All control is assumed to lie with the AM and all dialogs are initiated from it. Figure 11.4 shows typical exchanges between the three participants, as defined by the protocol.

For data acquisition, a minimum of three blocks are required from the EC. Block *a* describes a frame to be initially displayed to the user; block *b* specifies its layout and the format of response items in terms of length, type, position, etc.; block *c* (and others, if necessary) contains the text of prompts for each question to be answered. Prompts are displayed one by one on line 23.

Gateway pages (through which a user can ask to be linked with an EC) are set up by authorized information providers. Each EC is identified on the BTX system by a logi-

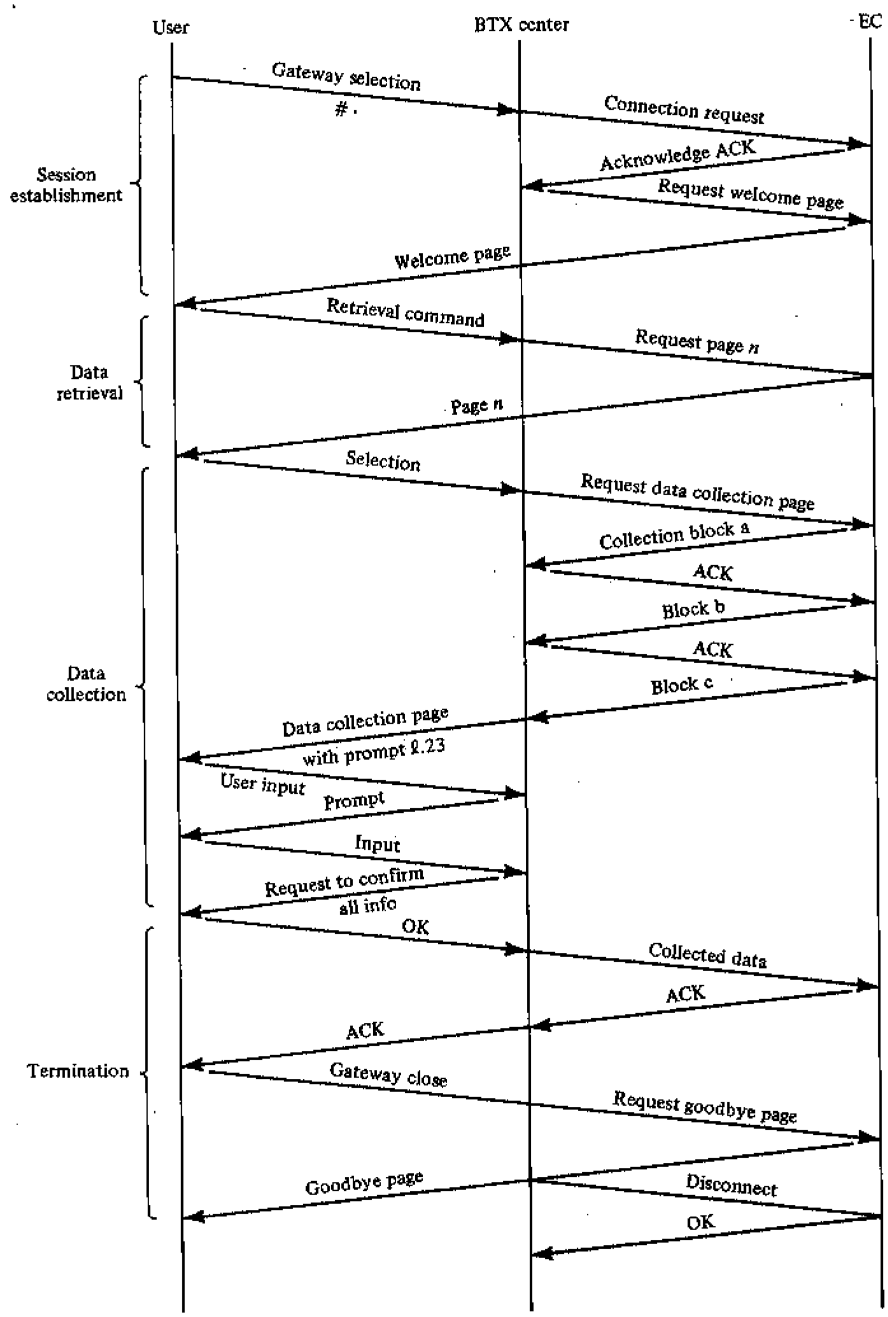


Figure 11.4 BTX gateway protocol exchanges.

cal number, with a number of logical channels assigned to it by the BTX center. This determines the maximum number of simultaneous connections to that EC.

Communication with the EC is accomplished through a packet-switching network accessible by X.25. When a connection is being established, one of the logical channels is mapped into a virtual circuit. In April, 1982, there were about 60 ECs accessible on the BTX trial system.

The software required in the EC has three main modules, as indicated in Figure 11.5 (see Holme, 1981): an X.25 interface, a gateway protocol handler (OSI level 4) to assure the correct transport of messages (blocks) to and from the access machine, and an application control (OSI level 5) to establish sessions, user verification, and interfacing with application programs. It should be pointed out, however, that certain elements of the above-described gateway protocol clearly involve the presentation and application layers.

British Telecom seems to favor this approach, and decided to establish its own gateway facility, initially based on the BTX software. One of the arguments for doing so was the possibility of linking up with a number of private videotex systems (as well as other ECs). In this way, access to the general-purpose database could be provided through Prestel, while private systems could offer specialized services.

At some point the question is inevitably asked: Why access an EC via videotex instead of dialing it directly? Chisholm (1981) gives the following answers:

1. With access via videotex there is no need for X.25 software; if a PAD (packet assembly/disassembly) facility is dialed instead, slower response and unfriendly interface will result.
2. The cost of a call is low compared to long-distance switched-circuit charges.

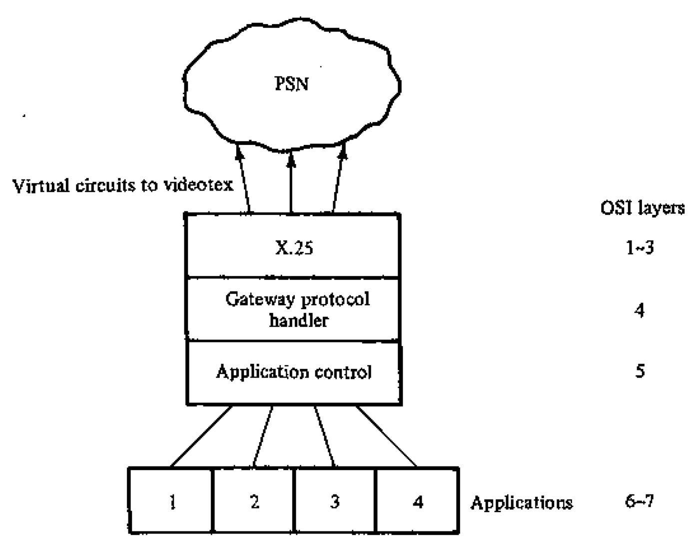


Figure 11.5 BTX gateway software structure.

3. Established Prestel billing service (no separate bills from ECs) is available.
4. Use is simple (there are no separate telephone numbers to maintain, and, more importantly, there are unified ways of interaction with transactional applications).

The above reference identifies five application areas in which gateways should prove to be particularly useful (because of shortcomings, such as limited response capability and limited database update volume, inherent in public videotex systems). The application areas are listed below:

- highly interactive applications, such as banking, CAI, and games
- rapidly changing information, such as bookings and reservations, stock market prices, and exchange rates
- high-volume data entry, such as mail order, airline seat booking, and teleshopping
- very large and specialized databases, such as timetables, encyclopedias, and bibliographic, educational, and telesoftware databases
- security-oriented applications (in which the owner prefers to have tighter access control than possible in public databases), such as banks, social security inquiries, and sensitive statistics.

BTX gateway session. The sequence of displays in Figure 11.6 illustrates in detail a typical Bildschirmtext session involving a banking computer. Frames 1–3 come from the AM; frame 3 is the gateway page initiating the connection to the banks' computer. Frames 4–22 come from the EC via the gateway protocol. The banking session is brought to an end when the user enters a wrong transaction number. In this system several transaction numbers are given to every user (by means other than videotex) as an additional protection measure. Each number can be used only once.

iNET

As opposed to the preceding example, iNET (developed by TransCanada Telephone System) is a stand-alone, microcomputer-based gateway system without local database capability (except for directory information). It offers a single point of access to a number of external databases and other information services (see Edwards, 1982). The central element in iNET is a set of directories of four types: national, regional, corporate, and personal. The first two are public directories listing all available information services on a nationwide or regional scale. The other two are tailored to the needs of particular subscribers or user groups. When a user makes a selection, the gateway automatically performs login to the selected service, and executes an initialization sequence (which can be pre-programmed individually for every user—e.g., to access a predetermined page). Then the gateway becomes transparent and the EC can communicate freely with the terminal. When the user signals a (standard) logoff to iNET, the gateway invokes the appropriate procedure. Messaging is available in the system as a separate service. iNET also maintains billing information in order to bill users for services used. A field trial providing access to some 500 services, began in 1982.

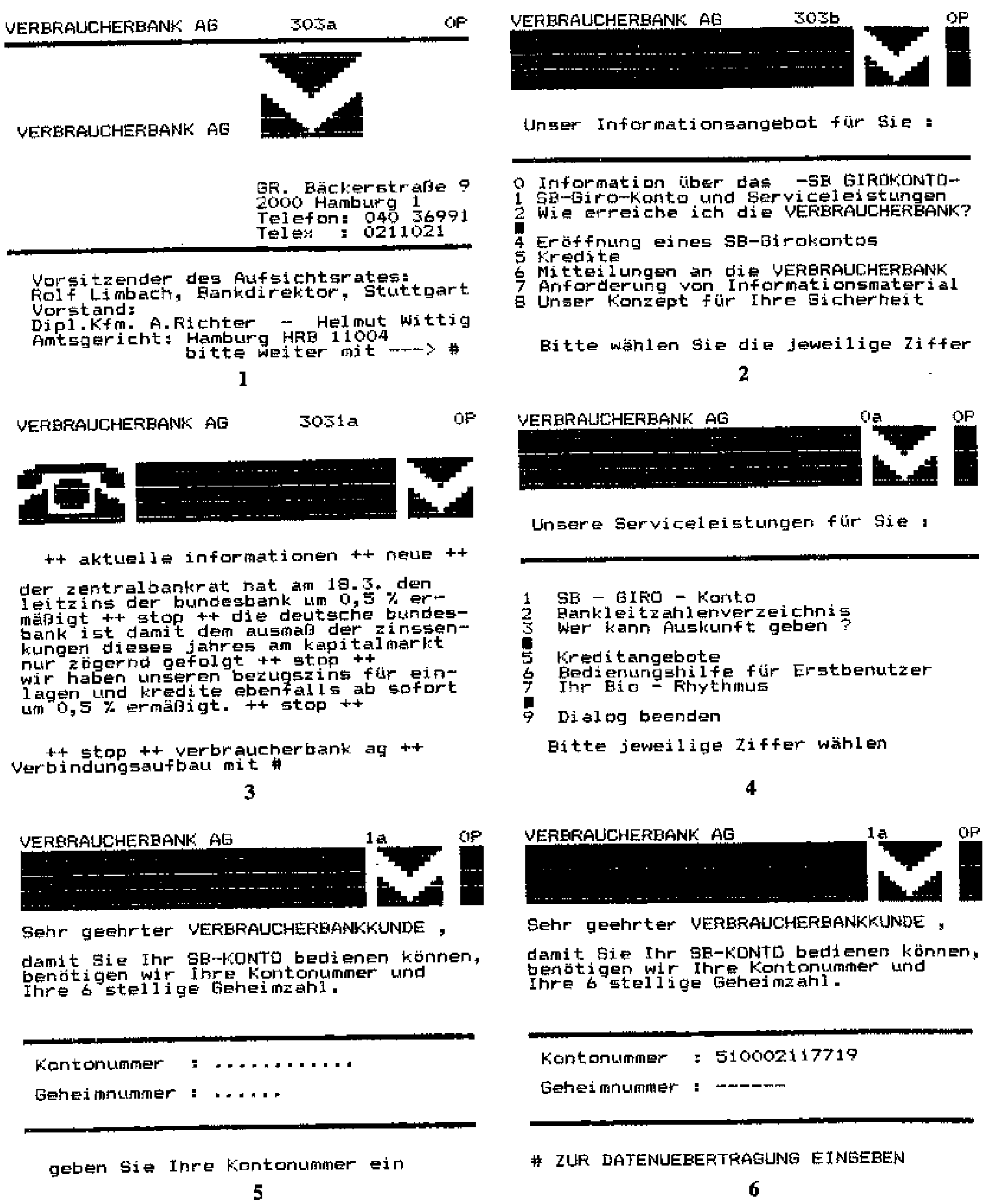


Figure 11.6 BTX banking session through gateway. Comments to the frames: 1: Entry page to the bank's subtree in BTX-database. Selection: #; 2: Available services. Selection: 1; 3: Gateway page including some service offerings. Selection: # (to initiate connection with EC); 4: First page from the bank's computer, available services. Selection: 1 (giro-accounts); 5: Data collection page; 6: Data collection page with user's response.

```
VERBRAUCHERBANK AG           100a        OP
           SB - GIRO - KONTO

   Kontonummer :   5/1000/211771/9

   Inhaber:Danke*Eric


 Wählen Sie eine der Möglichkeiten aus

          0 KONTOAUSZUG
          1 POSTBARANWEISUNG
          2 GELDANLAGE
          3 KONTOINFORMATION
          4 ÜBERWEISUNGEN
          5 DAUERAUFTRAG NEUANLAGE
          6 DAUERAUFTRAG ÄNDERUNG
          7 DAUERAUFTRAG LÖSCHUNG
          8 NACHRICHTEN VORSCHLÄGE
          9 PASSWORTÄNDERUNG und
            DIALOG BEENDEN (*9#)
```

7

```
VERBRAUCHERBANK AG          1000a        OP
  *****  K O N T O A U S Z U G   *****
  Kontonummer:   5/1000/211771/9
  Inhaber: Danke*Eric

 Wir halten für Ihren Kontoauszug alle

 Umsätze der letzten 3 Monate bereit.

 Um Ihnen das Blättern in alten Umsät-

 zen zu ersparen, können Sie nun den

 Beginn des Kontoauszuges bestimmen.

 ________________________________________
 Kontoauszug ab : Tag .. Monat ..
 ________________________________________

  Bitte den Tag 2stellig eingeben
```

8

```
VERBRAUCHERBANK AG          1000a        OP
  *****  K O N T O A U S Z U G   *****
  Kontonummer:   5/1000/211771/9
  Inhaber: Danke*Eric

 Wir halten für Ihren Kontoauszug alle

 Umsätze der letzten 3 Monate bereit.

 Um Ihnen das Blättern in alten Umsät-

 zen zu ersparen, können Sie nun den

 Beginn des Kontoauszuges bestimmen.

 ________________________________________
 Kontoauszug ab : Tag 18 Monat ..
 ________________________________________

  Bitte Monat 2stellig eingeben
```

9

```
VERBRAUCHERBANK AG          1000a        OP
  *****  K O N T O A U S Z U G   *****
  Kontonummer:   5/1000/211771/9
  Inhaber: Danke*Eric

 Wir halten für Ihren Kontoauszug alle

 Umsätze der letzten 3 Monate bereit.

 Um Ihnen das Blättern in alten Umsät-

 zen zu ersparen, können Sie nun den

 Beginn des Kontoauszuges bestimmen.

 ________________________________________
 Kontoauszug ab : Tag 18 Monat 03
 ________________________________________

 # ZUR DATENUEBERTRAGUNG EINGEBEN
```

10

```
VERBRAUCHERBANK AG          10001a        OP
  ******* KONTOAUSZUG ******* Seite 001
  Kontonummer :   5/1000/211771/9
  Inhaber :Danke*Eric

wir buchten am 22.03.1982
E.UND I.DANKE                      1.000,00+
 Ihr Guthaben     per 22.03.       1.462,13+
wir buchten am 25.03.1982
EUROCHEQUE-NR* 81029                  884,00-
 Ihr Guthaben     per 25.03.         578,13+
wir buchten am 26.03.1982
LA ERIC DANKE      10010010        1.800,00+
 Ihr Guthaben     per 26.03.       2.378,13+
wir buchten am 30.03.1982
EUROCHEQUE-NR* 81028                  257,55-
 Ihr Guthaben     per 30.03.       2.120,58+
wir buchten am 31.03.1982
ZINSGUTSCHRIFT MARZ                    25,38+
 Ihr Guthaben     per 31.03.       2.145,96+

weitere Umsätze bitte ->#
  Bitte weiter mit #
```

11

```
VERBRAUCHERBANK AG          10001a        OP
  ******* KONTOAUSZUG ******* Seite 002
  Kontonummer :   5/1000/211771/9
  Inhaber :Danke*Eric

Soll-Zinsen+Umsatz Heute              0,00+
Haben-Zinsen+Umsatz Heute             0,32+
 Ihr Guthaben     per 02.04.      2.146,28+







   Bitte weiter mit #
```

12

Figure 11.6 BTX banking session through gateway (continued). Comments to the frames: 7: Choice of transactions. Selection: 0 (to obtain account statement); 8–10: Data collection for date of statement; 11–12: Statement. Selection: #;

```
VERBRAUCHERBANK AG           100a        OP
          SB - GIRO - KONTO

  Kontonummer :   5/1000/211771/9

  Inhaber:Danke*Eric


 Wählen Sie eine der Möglichkeiten aus

          0 KONTOAUSZUG
          1 POSTBARANWEISUNG
          2 GELDANLAGE
          3 KONTOINFORMATION
          4 ÜBERWEISUNGEN
          5 DAUERAUFTRAG NEUANLAGE
          6 DAUERAUFTRAG ÄNDERUNG
          7 DAUERAUFTRAG LÖSCHUNG
          8 NACHRICHTEN VORSCHLÄGE
          9 PASSWORTÄNDERUNG und
            DIALOG BEENDEN (*9#)
```
13

```
VERBRAUCHERBANK AG           1004a       OP


 1 ÜBERWEISUNG
   (für Alphabettastaturen)
 3 ABRUF SCHNELLÜBERWEISUNGEN 1 - 8
   (für num. Tastaturen)
 4 ABRUF SCHNELLÜBERWEISUNGEN 9 -16
   (für num. Tastaturen)
 7 SCHNELLÜBERWEISUNG 1 - 8
   (Änderung - Neuanlage)
 8 SCHNELLÜBERWEISUNG 9 -16
   (Änderung - Neuanlage)

 bitte die entsprechende ziffer wählen
```
14

```
VERBRAUCHERBANK AG        10042a       OP
 - überweisungsauftrag an 20220300 -
      VERBRAUCHERBANK AG Hamburg
 Auftraggeber : Danke*Eric
 Kontonummer  :    5 1000 211771 9

 Empfänger   (Name und Anschrift)
......................................
......................................
 Bankleitzahl         Empfängerkontonr.
  ........           ...............
 Postleitzahl des Bankenortes
  ....
 Bezeichnung der Bank
  ...................
 Verwendungszweck für den Empfänger
......................................
......................................
   Betrag ....... DM .. Pfg.

 Geldtransaktionsnummer : ..........

 Bitte Empfaenger eingeben
```
15

```
VERBRAUCHERBANK AG        10042a       OP
 - überweisungsauftrag an 20220300 -
      VERBRAUCHERBANK AG Hamburg
 Auftraggeber : Danke*Eric
 Kontonummer  :    5 1000 211771 9

 Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
 Bankleitzahl         Empfängerkontonr.
  ........           ...............
 Postleitzahl des Bankenortes
  ....
 Bezeichnung der Bank
  ...................
 Verwendungszweck für den Empfänger
......................................
......................................
   Betrag ....... DM .. Pfg.

 Geldtransaktionsnummer : ..........

 Bitte Bankleitzahl eingeben
```
16

```
VERBRAUCHERBANK AG        10042a       OP
 - überweisungsauftrag an 20220300 -
      VERBRAUCHERBANK AG Hamburg
 Auftraggeber : Danke*Eric
 Kontonummer  :    5 1000 211771 9

 Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
 Bankleitzahl         Empfängerkontonr.
  10010010           ...............
 Postleitzahl des Bankenortes
  ....
 Bezeichnung der Bank
  ...................
 Verwendungszweck für den Empfänger
......................................
......................................
   Betrag ....... DM .. Pfg.

 Geldtransaktionsnummer : ..........

 Empfaengerkontonummer eingeben.
```
17

```
VERBRAUCHERBANK AG        10042a       OP
 - überweisungsauftrag an 20220300 -
      VERBRAUCHERBANK AG Hamburg
 Auftraggeber : Danke*Eric
 Kontonummer  :    5 1000 211771 9

 Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
 Bankleitzahl         Empfängerkontonr.
  10010010            171529-105
 Postleitzahl des Bankenortes
  ....
 Bezeichnung der Bank
  ...................
 Verwendungszweck für den Empfänger
Rechnung vom 17.03....................
......................................
   Betrag ....... DM .. Pfg.

 Geldtransaktionsnummer : ..........

 Bitte Verwendungszweck eingeben
```
18

Figure 11.6 BTX banking session through gateway (continued). Comments to the frames: 13: Choice of transactions (return to 7). Selection: 4 (to transfer funds); 14: Further choice. Selection: 1 (for alphanumeric keyboard); 15–18: Data collection for details on desired transfer.

```
VERBRAUCHERBANK AG          10042a       OP
  - Überweisungsauftrag an 20220300 -
        VERBRAUCHERBANK AG Hamburg
  Auftraggeber : Danke*Eric
  Kontonummer  :    5 1000 211771 9

  Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
  Bankleitzahl          Empfängerkontonr.
▌  10010010    ▌       171529-105        ▌
  Postleitzahl des Bankenortes
▌ ....                                   ▌
  Bezeichnung der Bank
▌ ...................                    ▌
  Verwendungszweck für den Empfänger
Rechnung vom 17.03.....................
........................................
   Betrag ....... DM .. Pfg.

  Geldtransaktionsnummer : ..........

  Bitte Verwendungszweck eingeben
```

19

```
VERBRAUCHERBANK AG          10042a       OP
  - Überweisungsauftrag an 20220300 -
        VERBRAUCHERBANK AG Hamburg
  Auftraggeber : Danke*Eric
  Kontonummer  :    5 1000 211771 9

  Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
  Bankleitzahl          Empfängerkontonr.
▌  10010010    ▌       171529-105        ▌
  Postleitzahl des Bankenortes
▌ ....                                   ▌
  Bezeichnung der Bank
▌ ...................                    ▌
  Verwendungszweck für den Empfänger
Rechnung vom 17.03.
........................................
   Betrag 125      DM 45 Pfg.

  Geldtransaktionsnummer : ..........

  Bitte Transaktionsnummer eingeben
```

20

```
VERBRAUCHERBANK AG          10042a       OP
  - Überweisungsauftrag an 20220300 -
        VERBRAUCHERBANK AG Hamburg
  Auftraggeber : Danke*Eric
  Kontonummer  :    5 1000 211771 9

  Empfänger   (Name und Anschrift)
Meier, Hans
Hamburg
  Bankleitzahl          Empfängerkontonr.
▌  10010010    ▌       171529-105        ▌
  Postleitzahl des Bankenortes
▌ ....                                   ▌
  Bezeichnung der Bank
▌ ...................                    ▌
  Verwendungszweck für den Empfänger
Rechnung vom 17.03.
........................................
   Betrag 125      DM 45 Pfg.

  Geldtransaktionsnummer : 2658912359

# ZUR DATENUEBERTRAGUNG EINGEBEN
```

21

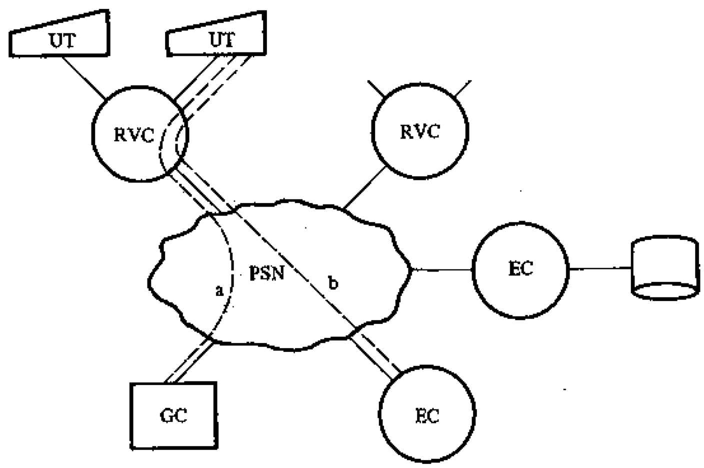
```
VERBRAUCHERBANK AG               14a        OP



Ihre Geldtransaktionsnummer war

        F A L S C H  !!

Aus Sicherheitsgründen müssen wir den

Zugang zum Konto sperren.




     zur BTX-Uebersicht bitte *0#
Verbindung zum externen Rechner beendet
```

22

Figure 11.6 BTX Banking session through gateway (continued). Comments to the frames: 19-21: Data collection for details on desired transfer; 22: Good-bye page from EC (wrong transaction number entered).

Cableshare Viewdata Gateway

The remarkable feature of the gateway developed by Cableshare, Inc., (London, Ontario), is that the gateway computer goes off-line once the connection between a user terminal and the selected service has been established. The network configuration is illustrated in Figure 11.7. Users are connected to the packet-switching network via remote videotex concentrators (RVC) whose main task is packet assembly/disassembly. When a user call is received, the RVC first connects the gateway computer (GC) for account verification and directory information, which is displayed to the user. The user's selection is then sent back to the GC (connection *a* in Figure 11.7), which in turn responds with the network address of the appropriate EC. At that point, RVC establishes a virtual circuit to the EC (connection *b*), and GC goes off-line. The advantage of this approach is that it might support a higher total traffic volume than would be possible if all traffic passed through the GC. However, the method of handling accounting information in this system is not clear.



GC: Gateway computer
RVC: Remote viewdata concentrator (access machine)
UT: User terminal

Figure 11.7 Cableshare's gateway.

12
Videotex Computers and Systems

This chapter is concerned with a few examples of the hardware/software structure of access machines, entire videotex systems, and teletext computers.

12.1 FEATURES OF ACCESS MACHINES

Access machine configurations found in the field vary from small minicomputers (such as in some Canadian field trials) to large mainframes (as in the Prestel network) and multiple processor systems (as in the BTX videotex center). The main design parameters include:

- level of functionality (data retrieval, messaging, service provider support, possibility of adding new applications, etc.)
- number of anticipated users (expressed, e.g., as the number of input ports in interactive systems) and response times
- communication media used
- required level of availability (different in trials and in commercial service)
- total system configuration (essentially, whether functions are distributed over several access machines or concentrators, or centralized into one)
- possibility for expansion
- available hardware and software technology.

We recall from Chapter 6 that some important functions found in access machines are:

- maintaining page-oriented databases including page retrieval, access authority control, and servicing update requests for service providers (interactively or in batch mode)
- supporting extensions and applications other than data retrieval, including response pages, mailboxes, keyword indexes, and transactional applications
- maintaining user contexts, that is, information about the current session of each active user, such as the current page number (or its internal representation), access trace, user-controlled markers, etc.; user context is necessary for the interpretation of most commands
- maintaining permanent user information, such as cumulative page charges, passwords, and access rights
- handling of ports and control of user terminals, including echoing, parity control, and assembly/disassembly of data blocks
- controlling and organizing the communication with other service computers and external computers
- billing and statistics
- controlling and monitoring of overall system including reconfiguration control, error statistics, and traffic flow control.

12.2 THE TELIDON COMPUTER OF THE DOC

Implemented at the Canadian Department of Communications (DOC), this centralized, minicomputer-based system serves primarily for experimental and field trial purposes (see Telidon, 1981b). A number of Canadian software houses, including Genesys, Systemhouse, Intellitech, and Prior Data Sciences, participated in the design. Genesys is marketing the system in several configurations. The hardware configuration is sketched in Figure 12.1. The main computer (PDP-11/60) performs all database, login, and statistics functions; the optional front end (LSI-11) provides terminal handling and data buffering in both directions. The two processors are linked by a direct memory access unit for fast data transfer. The operating systems are RSX-11M in the main computer and its memory-resident subset RSX-11S in the LSI-11. The Telidon database management functions are implemented with standard multi-keyed indexed-sequential files, supported by the RMS-11K record management package. A separate database is used for user-related information.

The software is organized as a number of cooperating, concurrent tasks, executing in both processors. The tasks, their interrelationships, and the databases are outlined in Figure 12.2. The main tasks and their functions are:

- DBA controls database updates
- DBAR controls database retrievals
- FREND issues output commands for the transmission of pages to terminals and processes command input

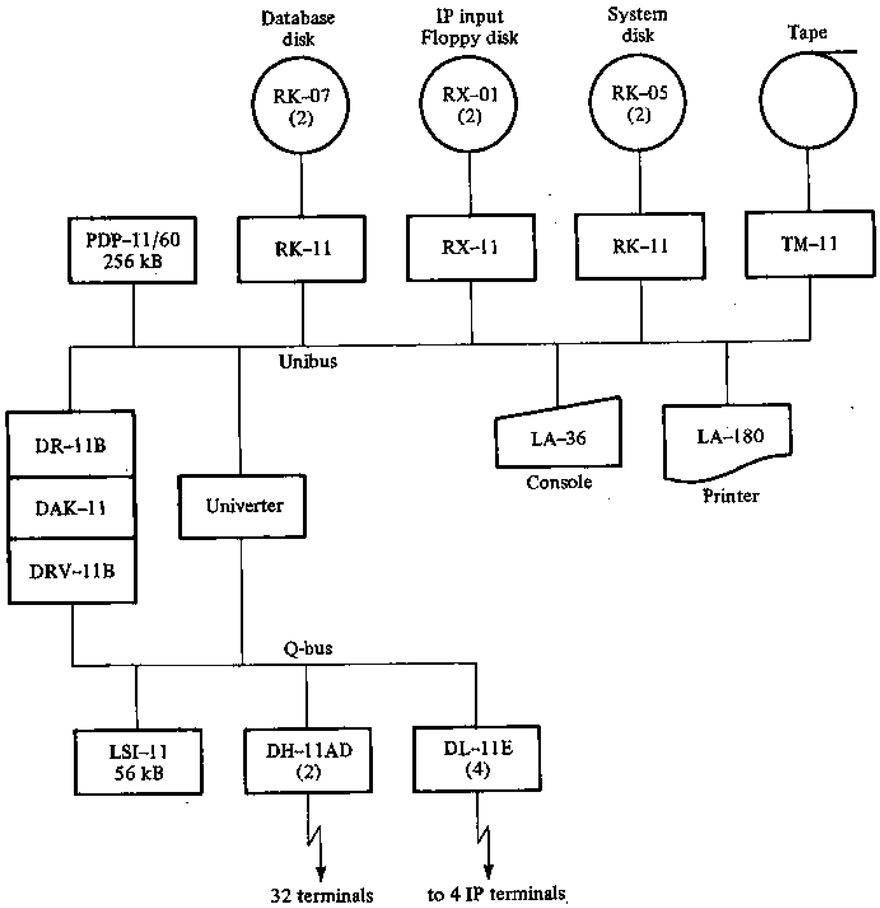


Figure 12.1 Hardware configuration of the DOC - Telidon system with one front end (LSI-11).

- OUTPUT uses the outgoing page headers to update user contexts
- INPUT parses and interprets incoming page requests, using context information
- HOSTSW distributes all messages to the appropriate tasks
- STATS carries out statistics processing
- UCP, RCP control on-line and off-line database update activity from information providers.

The software system is designed in such a way that it can execute entirely on the database computer. Tests on a PDP-11/70 have shown that 96 terminals can be supported when the operating system has been optimized; 64 ports were supported for a period of 6 months in the Bell Vista trial. If more terminals are needed, the FREND task has to be moved to one or more front ends.

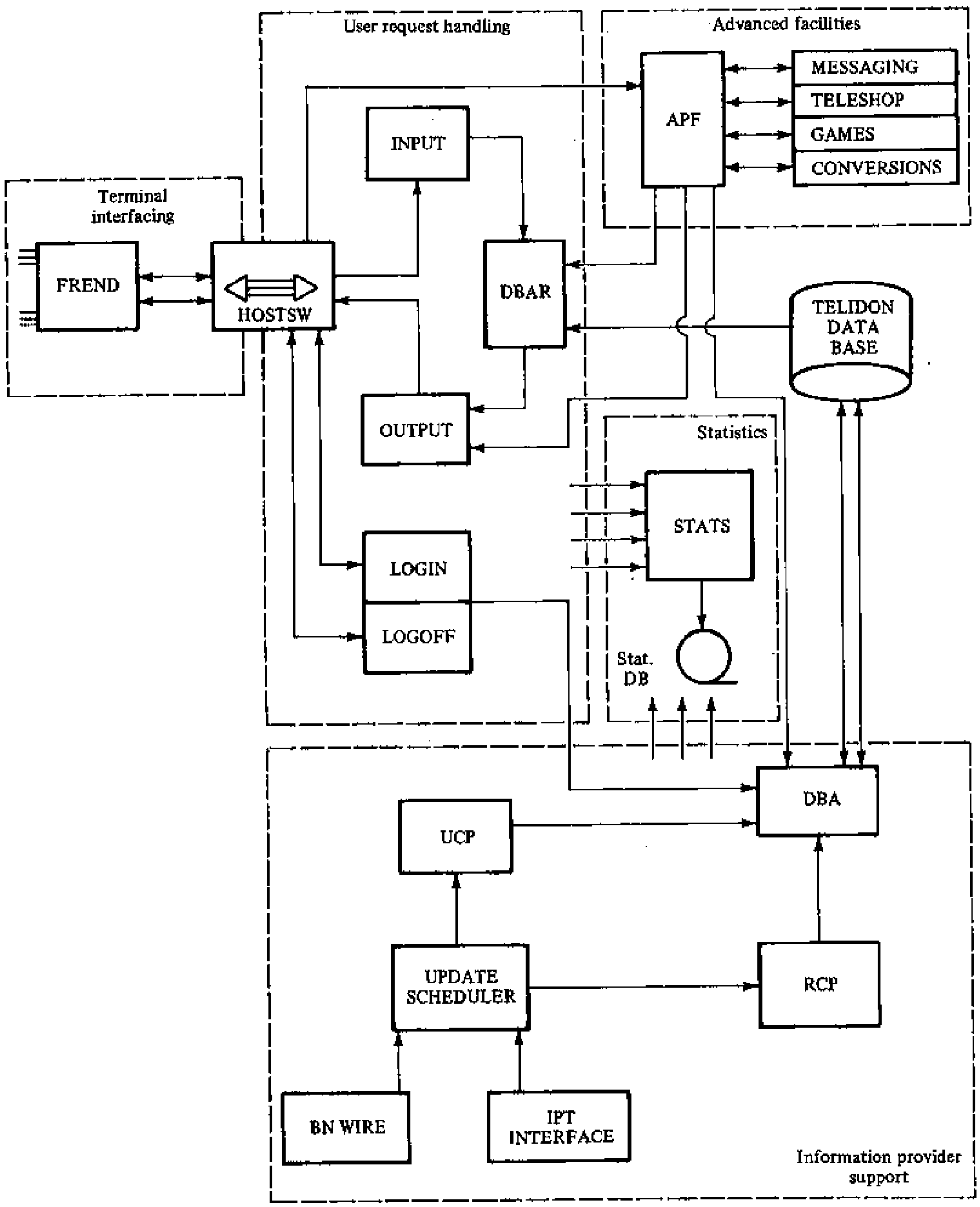
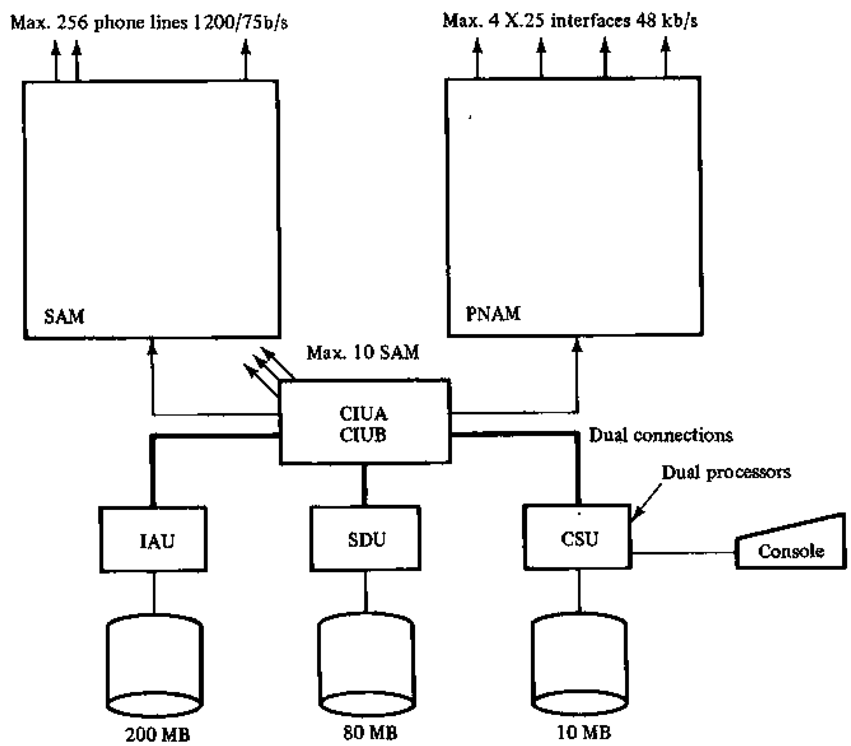


Figure 12.2 DOC Telidon database software organization.

12.3 THE BILDSCHIRMTEXT VIDEOTEX CENTER

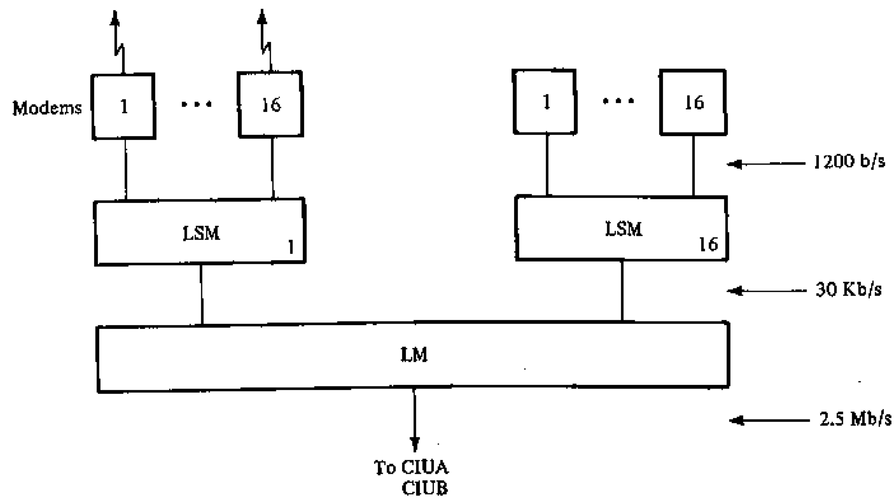
This prototype multiprocessor system is an example of a full-fledged service computer designed for the BTX trial (see Mantel, 1981). Although it will not be used in the public system, we will describe it in detail (which is rarely available for this type of machine). The contract for the public network has been awarded to IBM (see Chapter 6 for the configuration).

The main hardware components of the system are shown in Figure 12.3. Communication with user terminals is assured by up to 10 Subscriber Access Modules (SAM). Each SAM is a two-level concentrator consisting of a Line Module (LM), up to 16 Line Submodules (LSM), and up to 16 modems (1200/75 b/s voice lines) or 8 modems (2400/2400 b/s data lines) per LSM (see Figure 12.4). Internally, LMs and



SAM: Subscriber access module
PNAM: Packet network access module
CIUA, CIUB: Central interconnection units
IAU: Information access unit
SDU: Subscriber data and message unit
CSU: Central supervision unit

Figure 12.3 BTX videotex center configuration



LSM: Line submodule
LM: Line module

Figure 12.4 Subscriber access module (SAM).

LSMs are microprocessor-based systems with local memory (256 kB/LM). The main functions of SAM are:

- control of transmission to user terminals
- logon, session control (dialog with user)
- recording of page charges.

The Packet Network Access Module (PNAM, in Figure 12.3) is in charge of controlling up to four X.25-based physical connections (each multiplexing a number of virtual circuits) to the Datex-P packet-switching network. These circuits are used for cooperating with other videotex centers and with external computers through gateways. The internal structure of PNAM is similar to that of SAM except in terms of numbers: there is only one PNAM and up to four LSMs, each controlling a network interface. The main function of PNAM is allocating and controlling virtual circuits and transforming X.25 packets into messages used for internal communication.

The Central Intercommunication Unit (CIUA and CIUB) is a duplicated message switching facility serving for communication between all functional components. The Information Access Unit (IAU), Subscriber Data Unit (SDU), and Central Supervision Unit (CSU) have a common structure, outlined in Figure 12.5. The only difference is in the disk capacities. These units are dual computers with identical buses, processors, and memories, and each is capable of functioning alone (the other of the pair being in standby or maintenance mode). Note the dual connections from the two CIUs. The IAU supports the page-oriented database and performs update and retrieval functions for

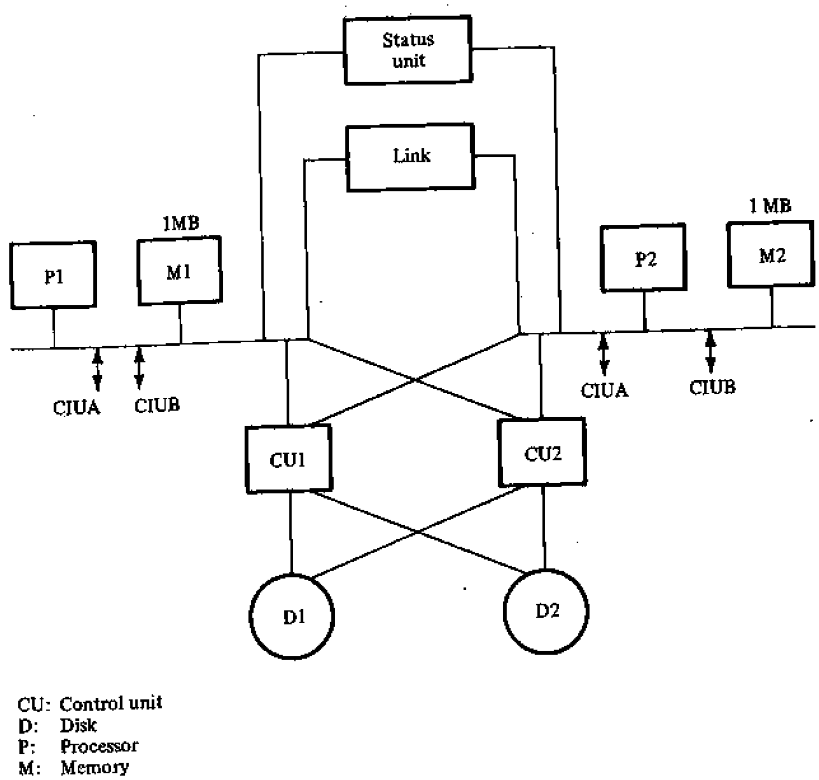


Figure 12.5 Structure of IAU, SDU, and CSU.

terminals and other BTX centers. It is also in charge of accounting for the information providers. The SDU manages all user-oriented data, keywords, and billing, and the messaging system. The CSU is charged with the control and supervision of the whole complex, dialog with the operator, system reconfiguration, diagnostic programs, and statistics.

The capacity of mass memories is about 100,000 pages (information and messages), extensible to 2 million. The response time is said to be between one and four seconds (four seconds if a page is retrieved from another BTX center).

12.4 STERIA-VIDEOPAC

Videopac, a comprehensive viewdata software package, is marketed by the French company Steria. We include a description of it because it is representative of many similar general-purpose packages available from a number of European and North American sources. The product is the outgrowth of the software originally commissioned for the Vélizy trial (see Chapter 6). Since then, a number of Videopac systems have been sold and ordered, and one system is functioning as a service bureau under the name Videotel. Videopac packages are modular, turnkey systems running on Honeywell-Bull Mini-6 computers in mono- or multiprocessing configurations. The systems are intended to act as nodes of different kinds in videotex networks; the six basic software configurations and their capabilities are (see Carteron, 1981; Steria, 1982):

- Videopac 200: interface between videotex terminals and any external application; can emulate a number of standard terminals and be extended by custom programs necessary for dialog and format translation
- Videopac 300: stand-alone access machine with its own database and interactive applications
- Videopac 400: configuration that runs its own applications and that can handle requests to other remote nodes, including external computers
- Videopac 450: configuration that runs its own applications and that can handle requests from remote nodes and external computers
- Videopac 500: configuration that can handle the above two functions (of 400 and 450)
- Videopac 1000: configuration that cannot handle user terminals; specializes in database creation and distribution to other nodes.

Some networking possibilities from the above systems are sketched in Figure 12.6.

Every Videopac configuration is centered around a number of standard software modules: the GECOS supervisor, the DSA (Distributed System Architecture) communications subsystem, and the DTF (Distributed Transaction Facility), collectively called the *core* of the system. This core assures the following functions:

- user terminal and line handling
- routing of commands to different applications
- communications network interface (different in each of the above six configurations)
- interfacing with local and remote transactional applications (to make remote terminals appear as local ones)
- configuration control (automatic resource switching in case of failure, restart, statistics and accounting, communication with operator console)

The core is supplemented by five optional modules:

- database management system (see below), intended to support Télétel database applications (present in Videopac 400, 450, 500)
- database creation and update
- page-editing facility
- development system for transactional applications (e.g., design of dialogs with user, and data collection)
- interface module with external computers and services not originally intended for videotex and that may require dialog and format translation (used in Videopac 200).

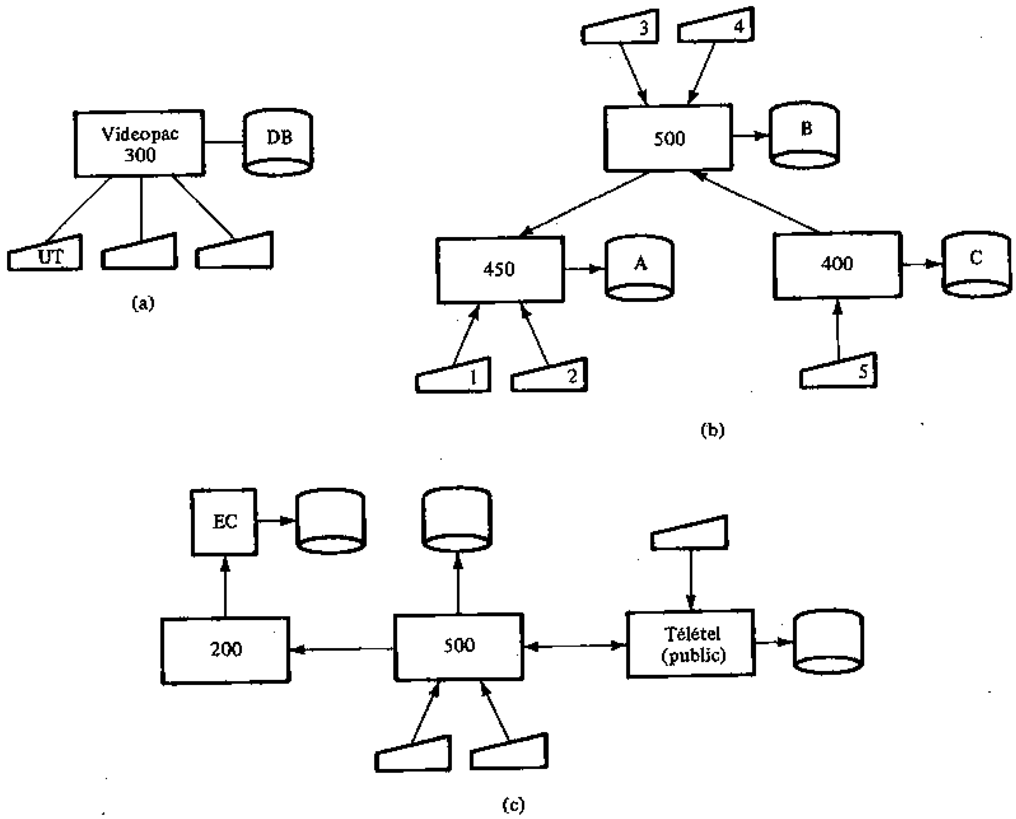


Figure 12.6 Sample configurations of Steria-Videopac software: (a) small private system; (b) larger private system; the arrows indicate accessibility to databases. For example, terminals 1 and 2 can access only database A, 5 can access A, B and C; (c) system allowing access to an external computer EC and to Télé; Télétel users can also access the other two systems.

Database Management

The database format of Télétel (and Videopac) has two levels: the directory level, which essentially contains a directory of available services, and the service level, (see also Section 10.3 on F.300). Databases on both levels are tree- and page-oriented, with multiple sequentially accessed frames per page. Services are independent applications usually maintained by different service providers; however, cross-links can exist between services.

Within this structure a number of different page types can be found (see Figure 12.7):

- R, root page, containing welcome information and parameters specific to the service in question
- I, index page, carrying multiple choices
- D, data page, carrying leaf nodes of the service trees

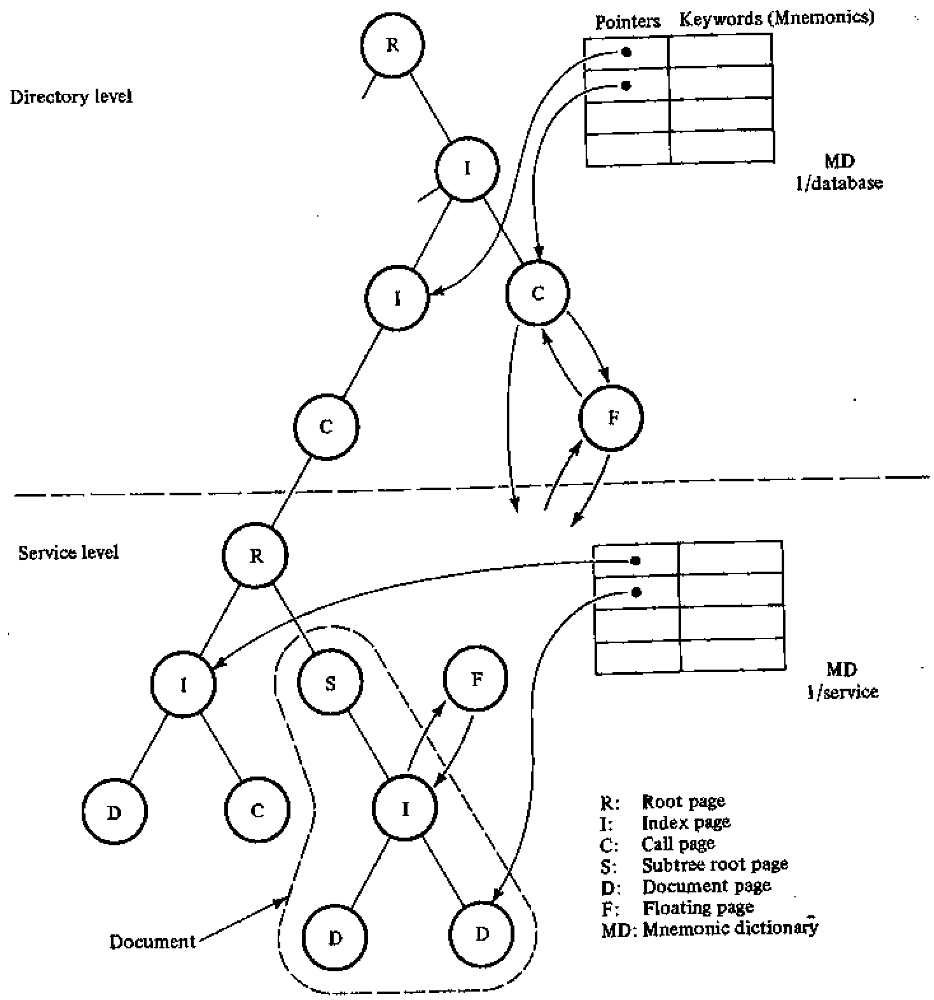


Figure 12.7 Videopac database structure.

- S, root of subtree, carrying certain parameters concerning its subtree (e.g., the set of floating pages that can be invoked from the subtree), or identifying the document represented by the subtree (see below)
- C, call page, invoking, when accessed, the execution of a program serving either to interact with the user (as response pages in Prestel), or to invoke branching to a local or distant service; certain index pages also having an attached program in order to interpret the user's response, possibly as a function of his previous choices
- F, floating pages, pages that can be called, by appropriate commands, from any place in the tree without modifying the search context, and that can be *global* (accessible from the directory and all services) or *local* to a given subtree (there are 10 global and 16 local i.e., redefinable in a subtree, floating page identifiers).

Accessing Pages

Access modes are available to the terminal user by multiple choice, by mnemonics, and by document retrieval.

1. Multiple choice access mode may be numeric or alphanumeric. Alphanumeric choices are interpreted by programs associated with some pages.

2. A mnemonic is an alphanumeric identifier that points to a *page*. (In effect, it is a keyword in the sense of the discussion in Chapter 13). Mnemonics are regrouped in dictionaries (MD). There is one such dictionary for the dictionary level of the database, and one for each service. The command structure permits issuing of single or double mnemonics. The latter have the form

$$M_1 * M_2$$

where M_1 is a menomic to be searched for in the directory dictionary; if it points to a service, M_2 is searched for in the corresponding service's dictionary. Thus, for example, typing

Medical * Ambulance

might yield a page containing telephone numbers and other information on available ambulances, in a service called Medical. Such double keywords can be issued at any time during a search; if necessary, the user will be automatically switched to a new service even if it is located in a remote computer.

3. In document retrieval, individual pages or subtrees are considered as *documents*, each identified by one or more keywords (e.g., extracted from the document text). Keywords are regrouped in a number of keyword indexes optionally supported by a thesaurus. As in other document retrieval systems, queries can contain logical operators on keywords. An example of such a retrieval is shown in Figure 13.1. Facilities are provided to set up dialogs and questionnaires for the inexperienced user, and for dynamic composition of summary frames ("virtual pages") at retrieval time.

12.5 THE FRENCH ELECTRONIC DIRECTORY

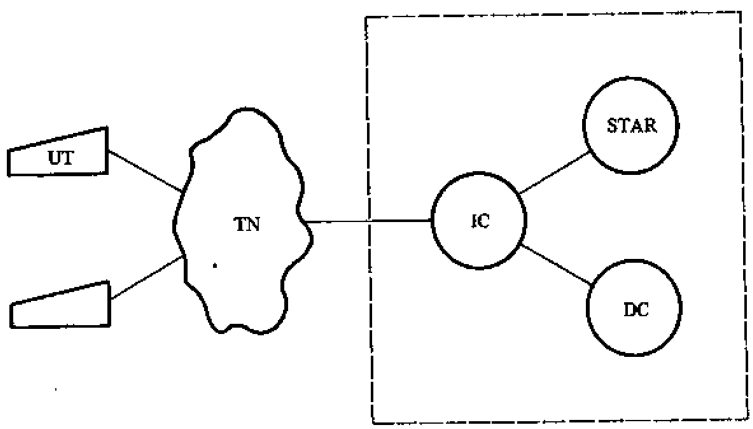
The Electronic Directory (see Maury, 1980) is one of the two large videotex trials undertaken in France. In contrast to the Vélizy trial (and the Videopac packages) it is a special-purpose design, optimized for a given application. The underlying idea is to replace printed telephone directories by small stand-alone videotex terminals that are supplied free of charge initially and that can be connected to a database. The initial phase is being carried out in the region of Ille-et-Vilaine, where about 160,000 terminals were installed by the end of 1982. Original plans call for over 20 million terminals by the end of the decade, replacing virtually all printed directories in France.

A series of experiments on a smaller scale have been conducted to optimize the user interface and the file structures. Two modes of search are provided, roughly corresponding to white pages (search by name) and yellow pages (search by profession).

However, the analogy stops here: the services largely exceed those available from the printed version. The resulting system demonstrates well the possibilities of intelligently combining various interface features, such as form filling, keyword search, multiple choices and clarification dialogs (see Chapter 13).

One of the most interesting features (see Le Moign, 1981) was included by the designers after realizing the extent of the difference between manual and computerized directory searching. For example, in manual mode the user can rapidly scan large portions of the directory (pages); this is impossible in the computerized mode because of transmission bandwith and screen-size limitations. Thus, the user must assume a more active attitude: he must type his data instead of reading them. This brings about a host of problems involving typing errors and spelling variants. In order to provide the robustness necessary to cope with such errors in a public system, all information entered by users (names of persons, firms, professions, and locations) are subject to different kinds of "normalizations"; the main databases contain all items in such normalized form. In the case of family names this consists of "phonemization": the names are transformed into a phonetic representation identical with that of other names written differently but pronounced the same way—e.g., "KREISLER" and "CHRYSLER." Similar transformations are performed on geographical names—e.g., "S", "SAINT", "ST", "STE" would all transform to the common phonetic "SENT". After that a unique numeric code is assigned to each name from a catalog of known geographical names. Professions are treated through synonym dictionaries, e.g., (AUTO, GARAGE) → CAR → KAR. The above (as well as other) processing is carried out in a clarification dialog with the user; the main search is triggered only afterwards.

The three main components of the experimental directory software are shown in Figure 12.8. User terminals are controlled by the Interrogation Center (IC), whose task is



IC: Interrogation center
DC: Documentation center
STAR: Videotex server
TN: Telephone network

Figure 12.8 French Electronic Directory system components.

to analyze the user's commands and data, and to execute the clarification dialog until all data are in required normalized form. STAR is a general-purpose package for the development of videotex applications. In this system its role is to support a number of auxiliary services, such as catalogs of towns, villages, counties, districts of larger cities (in the form of maps with districts numbered for multiple choice), lists of professions recognized by the system, and publicity pages optionally attached to the database records of professionals. The two main databases of the system for search by (*family name, location*) or (*profession, location*) are maintained in the Documentation Center. The above three components communicate through messages; this facilitates their implementation in separate, distant computers. Figure 12.9 shows the structure of the first of these databases and illustrates a typical retrieval operation.

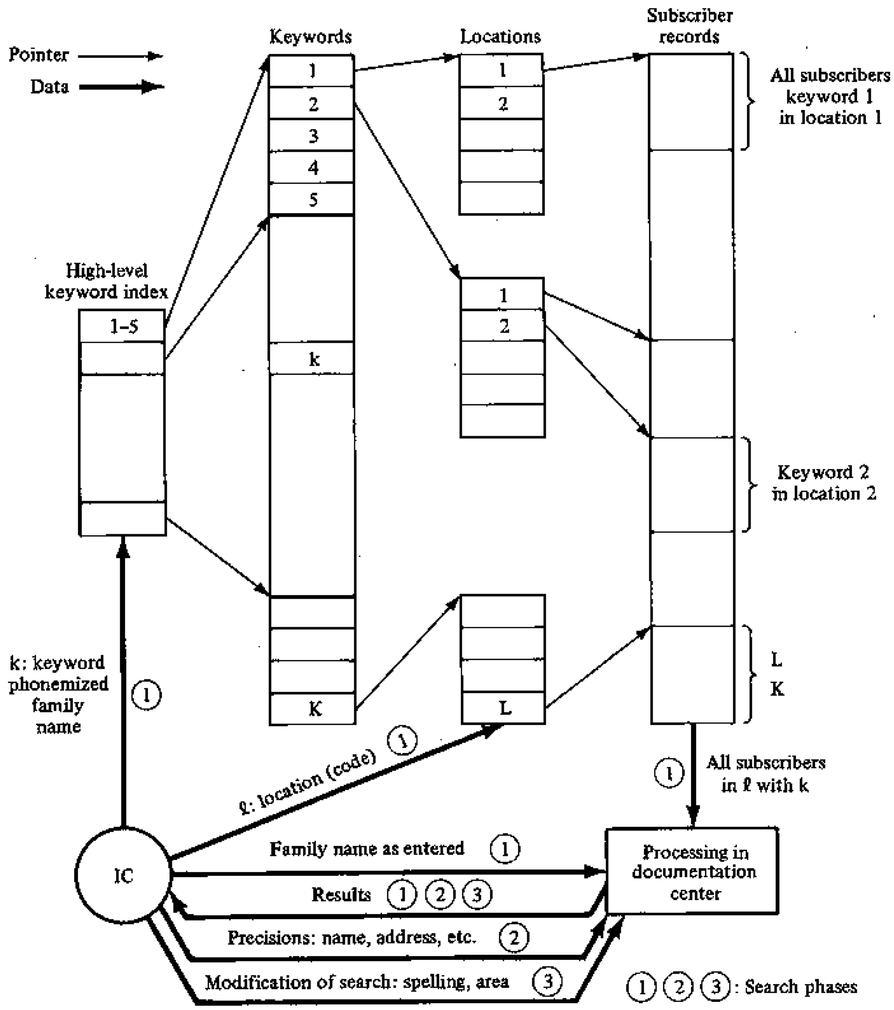


Figure 12.9 Processing of search by (*family name, location*) in French Electronic Directory.

The database is composed of files on three levels, called the keyword, location, and subscriber levels. Keywords (essentially the phonetic versions of all names in the directory) are arranged alphabetically in the first level; there is a high-level index in the main memory. The location level contains lists of locations in which each keyword occurs at least once, along with pointers to the third level. Here, subscriber records (*first name, family name, location, address, profession, telephone number*) are held in blocks by location and keyword. Thus, all subscribers who have similar-sounding names and who live in a given location can be retrieved in three disk accesses.

A typical inquiry has up to three phases (see Figure 12.9). In the first the user is asked to enter the family name and location of the person sought; this may involve a clarification dialog and access to the STAR services. Then a search is triggered in DC, as described, and the retrieved block is examined. If only a few orthographically matching names are found, their records are displayed to the user. Phase 2 is executed if there are too many "hits"; then the user is asked for further specifications to narrow down the choice; he may supply the first name, address, and/or profession. If he is not satisfied, he can modify his search criteria in Phase 3. In particular, he can relax the orthograph he has given originally and receive as a result the list of similar-sounding names, or he can relax or restrain his geographical criterion by asking for a search in a larger or smaller district than that specified at first. Searching in the professional (yellow pages) directory is carried out along similar lines, but in a separate database.

Some interesting technical data were made available about the directory (experimental version): the phonemization process is table driven, the table size is 1600 B, and 600 CPU instructions/character are required on the average. The processing of blocks retrieved by DC requires 4000 instructions/request. The hardware to serve 50 simultaneous requests includes two computers (SEMS-SOLAR 16–75, 64 kB private memory on each, 256 kB shared memory), I/O processor, one disk of 10 MB and two disks of 300 MB each.

12.6 TELETEXT COMPUTERS

We shall close this chapter with a discussion of computers controlling teletext transmission, sometimes called *head ends*. Requirements placed on such teletext computers are quite different from those for viewdata access machines. They are briefly summarized below.

1. The main requirement is that the system be capable of continuously furnishing data at the rate dictated by the transmission medium. Recall that the instantaneous data rate on the VBI may be close to 1 MB/s, and the effective rate approaches this limit in full field operation, implying that at least a significant part of the current broadcast cycle must reside in the main memory.
2. Pages not in the current on-line cycle, as well as other related information (e.g., transmission schedules), must be available in mass memory, usually hard or floppy disk.

3. There must be a page-editing facility (at least for alphanumeric information).
4. There must be operator-controlled and pre-programmable modification of the broadcast cycle (with inclusion of time-coded, real-time, or interactively demanded pages; see Chapter 10).
5. There must be communication facilities with remote service providers and with remote sources of data (such as news wire services).

The block diagram of a typical system is illustrated in Figure 12.10. The teletext encoder generates bit sequences to be superimposed on TV lines, and including synchronization, byte framing, and time codes. The data inserter modulates the bit sequence onto selected TV lines.

Actual systems vary in their individual capabilities. Smaller configurations may have only floppy disks and main memory capacities for 50–100 pages. Larger systems typically have several editing terminals, large hard disks, and main memories. An intermediate level of memory may be included to hold large broadcast cycles. Electronic disk memories (CCD, magnetic bubble, or semiconductor) are ideal for that purpose, since they are cyclically organized and feature millisecond-range access times and sufficient data rate. With such equipment it is possible to generate data streams for multiple TV signals.

Data Exchange between Teletext Systems

An often-needed capability is that of transporting the data content of one video signal onto another, with the possible change of TV lines used. One application is for smaller,

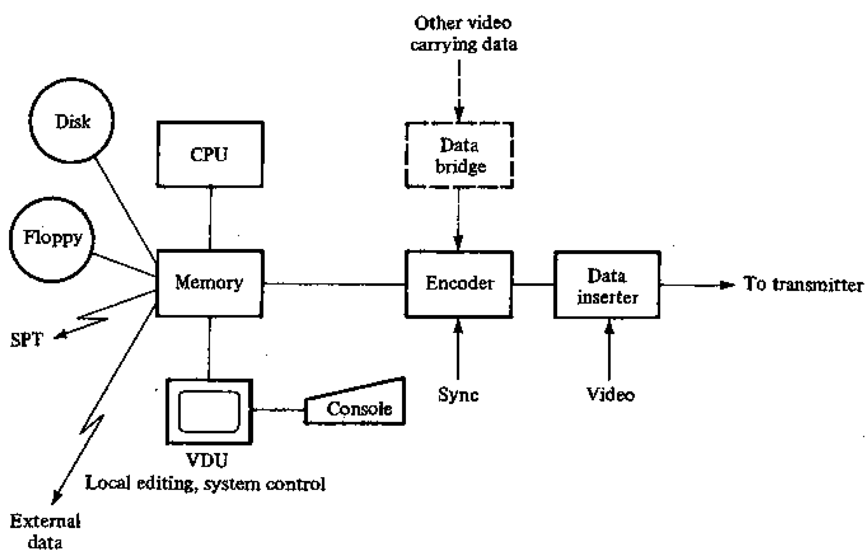


Figure 12.10 Block diagram of typical teletext system.

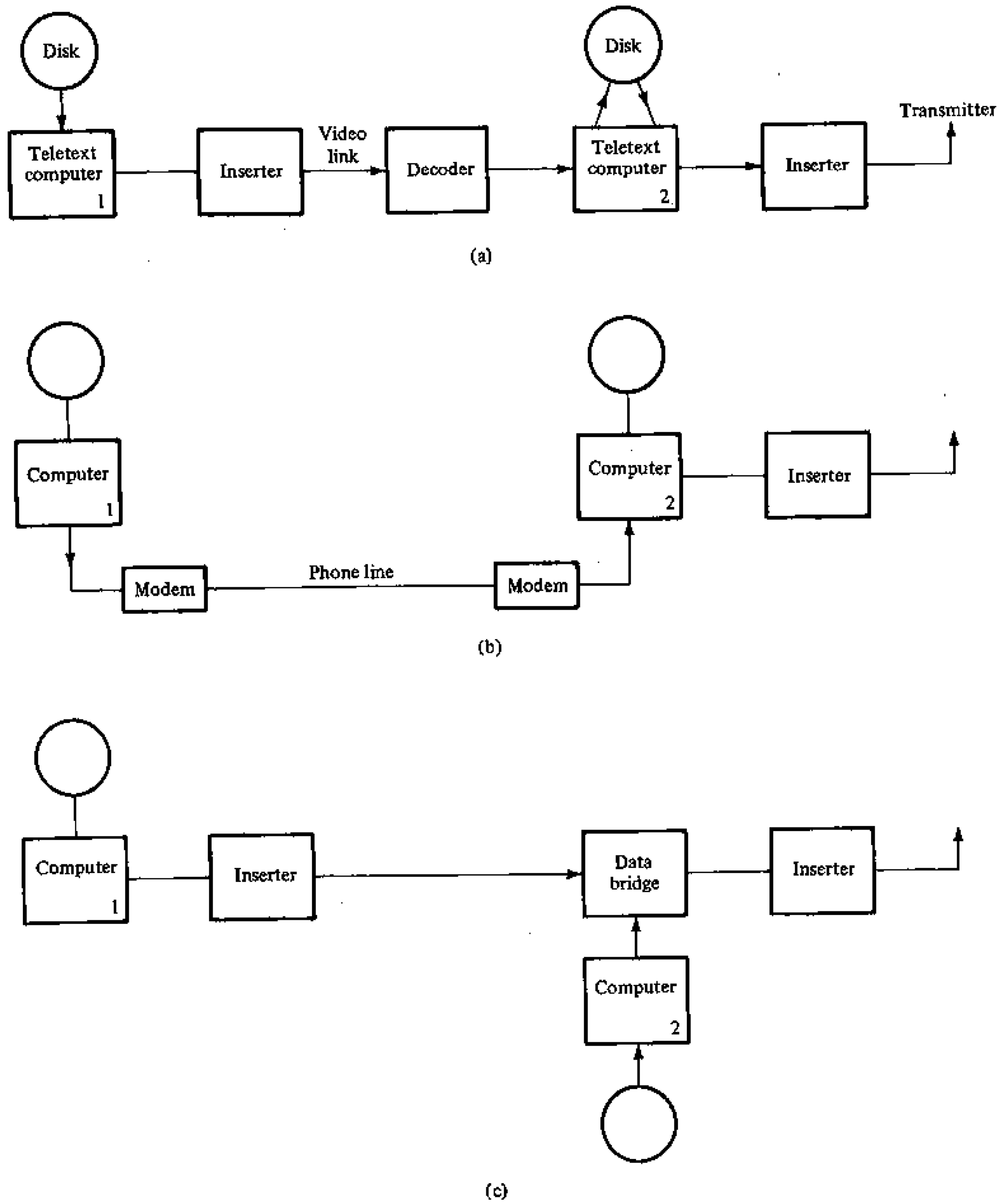


Figure 12.11 Possibilities for linking two teletext systems.

regional TV or cable systems to relay teletext data carried by larger or specialized networks, as described by Mothersole (1979). This can be done in several ways. The most straightforward method (see Figure 12.11(a)) is to receive and decode data by a conventional decoder, store the pages on disk and re-insert them on the locally generated program. The second method is similar, but uses a standard data link between the two teletext computers (see Figure 12.11(b)). The disadvantage of both these approaches is that the receiving party must have a complete teletext system, which may be too costly for a small station. In such cases a third solution is practical: use of the *data bridge* facility. This is basically a re-synchronizing circuit (the two TV signals are generally unsynchronized) that extracts the data lines from one video signal and deposits them in a small buffer that feeds the second inserter. The timing relations are derived from the synchronization pulses of both TV signals (see Figure 12.11(c)).

Part V
COMPLEMENTARY ISSUES

While the first four parts of this book deal more or less with the *status quo* of videotex, the nature of this part is different: it is concerned with some of the functional and performance enhancements that are likely to occur as videotex gradually evolves into generalized information systems. The next subject is telesoftware (Chapter 14), which is recognized for its potential to radically extend the range of applications and to introduce distributed processing into videotex networks. Finally, we include a few videotex-specific topics on efficiency and performance evaluation.

13
The User Interface

One of the very few common features of all videotex systems is the pervasive role of untrained casual clientele. Therefore, the success of the whole concept of videotex hinges largely on whether the user will find the system sufficiently understandable and pleasant to deal with. This chapter (partly based on a report by Ball and Gecsei, 1981), is devoted to the study of actual and potentially available user interfaces. We explore some ways in which videotex design can profit from the many parallel developments in the fields of friendly interfacing with databases, interactive systems, and the newly emerging discipline of "user psychology."

Roughly speaking, the interface is that part of the system hardware and software that is "visible" to the user and through which all interactions take place. (This is a somewhat simplified view because users are often also aware of certain internal features and behavioral characteristics of the system, such as data organization and response time. Incorrect conceptual models of these structures may cause false expectations and confusion on the part of the user and, therefore, it is sometimes desirable to provide him with supplementary information on "how the system works.")

Man-machine dialogs are conducted in terms of *interface languages*. They have two natural components: the requests originating in man, and the machine's responses. We deliberately focus our attention on the request part, because formulating requests requires knowledge of rules and action on the part of users and, therefore, it is the main factor in determining the suitability of an interface language. The interested user will find more information, e.g., in Shneiderman (1978), in Reisner (1981), and in Martin (1973).

Since interface functions serve for a blend of data retrieval, transactional applications, and system control, we do not separate these three issues in the subsequent discussion. Instead, in this chapter we will briefly review the following four broad categories of interface languages, and comment on their applicability to videotex.

- question-answer methods (including menu selection and form-filling)
- keyword-based methods (pre-coordinate and post-coordinate)
- query languages (artificial and natural)
- interfaces with browsing facilities.

The syntax of question-answer and pre-coordinate keyword approaches is extremely simple and requires only minimal training. The user has only to recognize what he wants, and does not have to memorize rules to express it. In this sense, his role is essentially passive. Therefore these interfaces prevail in all major videotex systems. However, the amount of information passed to the system per request is often too limited and one cannot formulate requests with sufficient precision. The remaining languages are more powerful, but they have more complicated syntactic rules requiring user training and/or considerable computer processing power. Their introduction into videotex is slower but nevertheless advancing, mainly due to the accessibility of specialized services through gateways.

13.1 QUESTION-ANSWER INTERFACES

The basic pattern here is that the user is prompted by the computer to answer a series of precisely formulated questions. The permissible answers or their formats are indicated (e.g., "Please type Yes or No," or "Enter Name and Flight Number"), and errors can be immediately corrected. Many transactional applications are based on this type of interface (see the Bildschirmtext banking example in Chapter 11).

Menu selection is the most widely used interface in videotex. Its concommitant database organization is the tree (or directed graph) of pages. Recall the disadvantages of this approach:

- rigid organization of the tree, which forces all users through search paths anticipated by the database editor
- page contents fixed in advance, so that information cannot be edited at search time
- browsing in larger databases is tedious, and can easily produce disorientation
- costly construction and maintenance, especially with large numbers of cross links; when the database is frequently reorganized (due to changes and growth) some page numbers must be altered, and all changes must be reflected in the cross-link structure.

Enhancements of Menu Selection Systems

A number of ways to counter the shortcomings of the page tree have been developed. As a result, enhanced page-oriented databases are still the workhorse of practically all contemporary videotex implementations.

One class of improvements concerns facilities for automatic creation of menus and documents, and their loading into the database. In this context we should mention the (non-videotex) generalized menu-based system ZOG (see Robertson et al., 1979). It can be used for a wide range of tasks, including interfacing with existing database systems. Successful applications include a browsing system for a library, project management, guidance and interface for a medical diagnosis system, and, recently, a management system for a nuclear-powered aircraft carrier. ZOG differs from standard videotex in that the contents of a display page can be stored independently of the page format; the two are combined at database load time. It is possible to pre-define a class of menu structures in ZOG so that given, for example, a description of a restaurant, the ZOG software could synthesize an appropriate display page and then automatically load that page into the network of menus. The connections that allow the user to search the restaurant database would be established automatically. This is in contrast to Telidon and Prestel, in which pages are units whose cross-links must be explicitly specified. Further features are a text-editing facility that (in principle) enables every user to create new pages, and the possibility of executing arbitrary programs when accessing a page.

We also mention in this context the existence of specialized software packages that automatically create Telidon- and Prestel-compatible databases from machine-readable text sources. An example is Preview, a software package marketed by Langton Information Systems, Ltd., in England.

Applying conventional database technology. Another possibility of improving page-oriented databases involves capitalizing on achievements of traditional database technology. It may seem strange that although database technology has advanced significantly over the years, videotex systems are still based on facilities that closely resemble early file systems. Surely the research and development devoted to conventional record-oriented structures can be applied to videotex facilities (see Tompa et al., 1981).

Conventional database technology can be applied to videotex systems as soon as it is realized that "videotex page" can be made a valid domain in a record-oriented system. Consider, for example, a relational database of data on houses that are available for sale. A relation could be defined with the attributes: number of bedrooms, price, dining-room type, address, layout, picture, description (where the first two attributes' domains are integers, the next two are character strings, and the last three are videotex pages—which would not normally be part of a query expression). Thus, conventional relational processing could retrieve the set of addresses, layouts, and pictures for which number of bedrooms = 4, dining room type = "formal," and $150,000 ≤ price ≤ $180,000.

A distributed architecture could be used to combine the videotex page-management and the relational systems. A user would request a house inquiry page from the videotex system, such a page having been constructed as a response page by the information provider. This page would be filled in by the user in some convenient notation and format, and then passed to the conventional database system for processing. The response would then be passed back to the videotex access machine, which would either request the target pages from the videotex page depository, request more information from the user, or display a dynamically constructed index page for further selection by the user.

Response pages. Although at first videotex systems were primarily designed for information retrieval, it quickly became evident that viable systems must not be limited to retrieval. The purpose of many retrievals is to obtain information in order to begin a transaction. For example, a search for Greek restaurants is not likely to be undertaken merely for reasons of academic interest, but rather for making a reservation. Similarly, looking at stock market quotations is intended to determine whether to buy, sell, or to stay pat.

In Prestel, an information provider can create a so-called *response page*, as distinguished from an information page. Such a response page contains a form to be filled in by the user. The form's layout and its blend of prompting information, system-generated response data (such as automatic fill-in of user's name and address), and user-generated response are under the control of the service providers. When completed, the form is appended to a queue of user's responses, which can subsequently be examined by the appropriate service provider.

The response page facility is an embryonic form of other interactive applications. For example, banking or teleshopping transactions can be conducted if the users are equipped with alphanumeric keyboards and if they can be linked up with the bank's or store's computer. This kind of application is carried out in a systematic manner through gateways. Electronic mail service can be created by providing sufficient "mailbox" space in the videotex (or an external) computer. All the above facilities are being implemented by Prestel, Bildschirmtext, and Télétel.

In this context we should point to the emerging central role of form filling and *form management*. Tsichritzis (1982) argues that general ways of form processing can be seen as the common denominator for messaging, communication, document retrieval, and other office automation functions. This might also be true for videotex: response pages, gateway protocols, and transactional applications are all based on forms.

Page-control programs and action pages. A significant improvement can be brought about by executing a program when selecting a page (instead of just passively displaying the information). Response pages are only one of the possibilities; another is to attach "page-control programs" to each (or some) displayable page. These programs can serve to interpret the user's selections and responses and to direct further branching, as in computer-assisted instruction. Action pages (a term used in Telidon systems) invoke a program that can sustain a dialog with the user. For another embodiment of this idea, see, e.g., the Steria database management system treated in Chapter 12.

13.2 KEYWORD-BASED INTERACTION

Many database and document retrieval systems use keywords as a basic means of retrieval specification. In principle, a keyword-based system appears to the user as an associative store capable of retrieving documents or records containing (or associated with) a given keyword. The principal advantage of this approach is that it permits the user to access the desired information "directly," without concern about the way in which the data are structured and stored. Interesting VLSI designs exist for such "dictionary machines", described, e.g., by Ottmann et al. (1982).

In the following discussion, the term *keyword* means a descriptor that indicates the subject of a document in a database, such as, e.g., a videotex page. Keywords can be assigned to documents manually or automatically in advance, or documents can be searched for the occurrence of given keywords at retrieval time. In formatted (record-oriented) databases a keyword may be the value of a field. For search purposes keywords are arranged in lists called *indexes* or *dictionaries*. The arrangement is in a known searchable order (alphabetical, numerical, chronological).

Pre- and Post-Coordinate Processing

Two important classes of retrieval techniques based on keywords can be identified: pre-coordinate and post-coordinate. In the former, keywords or combinations of keywords (e.g., "contemporary music") designating subject documents are fixed in advance and all pertinent documents are assembled under these fixed (pre-coordinated) headings. In post-coordinate systems the index is composed (mostly) of single keywords; coordination of these keywords is performed at retrieval time by forming Boolean sums, products, and complements on the sets of documents found under individual index entries. For example, documents describing contemporary music would be obtained as the intersection of the sets of all documents described by "contemporary" and all documents described by "music". The command leading to such a search would be (essentially) "contemporary AND music (or "music AND contemporary"). Other operations on sets can be used as well, e.g., "contemporary AND music AND (NOT instrumental OR Schönberg)."

Note that document databases are not the only systems using post-coordinate processing. Many record-oriented database management systems provide Boolean operations to facilitate retrieval, which is an essential criterion of post-coordinate indexing. In particular, relational databases may be thought of as systems in which keywords are arranged in relations. A relational search is, in essence, the retrieval of tuples (documents) that contain logical combinations of keywords in specified columns, as defined by the query (see the real-estate example in the preceding section).

We briefly summarize the differences between pre- and post-coordinate processing:

1. In pre-coordinate processing search patterns must be anticipated by the indexer (e.g., the combination of "contemporary" and "music" is pre-specified in antici-

pation of such requests). Post-coordinate systems permit specification later, at search time; this is a way of shifting the burden of combining the keywords from the indexer to the computer.

2. With pre-coordinate fixed indexes (unless their size is enormous) one cannot achieve minute precision (selectivity).
3. Post-coordinate requests can be arbitrarily tuned and refined by adding new conditions until the user is satisfied with the results. Different search strategies can be realized and results of previous searches can be reused under modified conditions.
4. Post-coordinate searching implies that the system's response is created and edited at search time, and not beforehand.
5. The difference between the two approaches is not always as clear as it might seem. The French Electronic Directory (see Chapter 12) is basically pre-coordinate with features resembling post-coordinate processing (such as refinement of requests).
6. Post-coordinate systems use query languages with explicit and, sometimes, complicated syntax.

13.3 KEYWORDS IN VIDEOTEX

There are four major ways to access videotex databases in use today: menus, keywords (pre-coordinate), alphabetic index, and printed directory. These four techniques have been the subject of many studies; in most of them (e.g., in Stewart, 1979) keyword search emerged as the most efficient method.

Due to its simplicity, pre-coordinate keyword access is more widespread in videotex: the only action required from the user is to type a keyword (with or without being prompted)—and then wait to see what happens. The method can provide shortcuts to important points in the database. However, in itself it is probably not a sufficient substitute for menu selection but rather a useful complement. Numerous implementations of keyword access exist. The Steria-Videopac system (see Chapter 12) has a two-level database consisting of a collection of independent services (lower level) and a directory of services (upper level). The directory and each service have their own keyword dictionaries; the system accepts commands in the form of pairs of keywords, making it possible to jump at once to any point in any service. The first keyword must be the name of a service.

Another example is the so-called Montreal Keyword System (see Bochmann et al., 1981). In contrast to the above approach, the assignment of dictionaries to subtrees is entirely under the service provider's control: any subtree can have its own dictionary, and alternative dictionaries can be searched in case of failure.

A third example is operational in the Danish videotex system (see Orsnaes, 1981). It features the possibility of sequentially combining several keywords (e.g., "cars—Ford—1978") in one request. This is not unlike applying AND operators in a post-coordinate system; however, the effect is the same as applying the three keywords in a series of three separate requests (i.e., one cannot change the order and say "Ford—

1978—cars'' and expect the same answer). The underlying principle is the same as in Prestel, in which rapidly typing 7, 3, 8 would lead to the same page as would indicating the three menu choices: 7, then 3, and then 8.

Design Issues

A number of issues must be resolved by designers of keyword access systems.

1. The first issue is the size and implementation of dictionaries. Here the primary concern is not to compromise the retrieval efficiency (low CPU time and number of disk accesses per request) that is a fundamental design requirement in videotex. Small dictionaries can be held in main memory (e.g., with the user context) and can be searched sequentially.

2. In general, the *scope* of a dictionary is the set of pages from which the dictionary can be accessed. At one extreme there may be a single, usually large, dictionary that is searched on all keyword requests; its scope would span the whole database. Alternatively, the scope may be limited to a single page; then it becomes similar to a menu. In intermediate cases, different regions in the database (applications, subtrees, or even individual pages) may have separate dictionaries. This has the advantage that the same keyword can be interpreted differently (i.e., can point to different pages) according to the search context. For example, ''French'' may point to French restaurants or to the French Revolution in the dictionaries of a restaurant guide and of an encyclopedia of revolutions, respectively. In addition, individual users or user groups may have private dictionaries.

3. A search in a dictionary can either succeed (a ''matching'' entry is found) or fail. However, matching can have many different meanings such as:

- exact match, in which a keyword in the dictionary is found that is identical to the search argument
- prefix match, in which the search argument is identical to the prefix of a keyword in the dictionary (example: *Sport—Sports*man)
- approximate match, in which a keyword differing from the argument in less than $k \geq 0$ characters is considered as a match
- phonetic match, in which, if no exact match is found, search is continued for keywords with similar pronunciation to the search argument (this type of search is implemented in the French Electronic Directory).

4. Issues involving the number of pointers from a given keyword in a dictionary and to a given page have consequences for implementation (since space is taken up by multiple pointers) and system integrity (since means must exist to update automatically the dictionaries as the database grows and changes). A related problem is determining how many keywords should be associated with a given page.

5. If a search yields more than one matching keyword, or more than one page is pointed to by a keyword, how should such multiple matches be presented to the user? Usually, if this number is small (e.g., six or less), the pages are sequentially displayed; if not, the system can request the user to further specify his request (e.g., by a longer prefix).

6. The issue of no-match (or "miss") situations also must be resolved. In such cases, the question is: What should the system's reaction be?

7. Interaction between keyword and other search methods (e.g., menu) is desirable to permit to switch between methods at any time. The questions involved are whether there should be separate "search contexts" for each method? and whether the users should be aware of the switch.

Similarities Between Menus and Keywords

Traditionally, menu selection, keyword, and alphabetic index search have been considered to be distinct approaches to data retrieval. Closer scrutiny shows a number of similarities among them; it is useful to stress these similarities because the resulting unifying view may point to more generalized database structures and search methods (see also Tompa, 1982).

In essence, the user of a page-oriented database is always confronted with the same situation: he must select the next page to be retrieved. There always exists a fixed list of possibilities, called the *current dictionary* (CD), from which he can choose. The selection has the following modalities:

1. The CD may or may not be automatically displayed (however, it is practically always available on request).
2. The scope of the CD may vary.
3. The length and other parameters of the CD, such as number of pointers, etc., may vary.
4. The choice can be indicated directly or indirectly; in the latter case, entries in CD are labelled, usually by means of numbers or letters.

Now we see that a standard menu is a short displayed CD whose scope is limited to itself, by the use of indirect choice. A keyword index is a CD that has a typically wide scope and direct selection (through typing of the keyword), and that is displayed or not, according to the particular situation. An alphabetic index is an intermediate case in which CD entries are labeled for easier entry, using a keypad. The scheme is often designed to provide for gradual refinement (e.g., selection 1: A–D, 2: E–H, etc.) through a series of labelled selections.

We complete the discussion with two remarks. First, the necessity of labeling CD entries stems from the lack of an adequate pointing device in current videotex systems. It would certainly be easier to point directly to a keyword or any object on the screen, than to have to read the labels and then look at the keypad and enter the label. (However, videotex systems with touch-sensitive screens have been implemented.)

The second remark concerns situations in which displaying the CD may be omitted. This can generally be done when the set of possible choices is known from the context (e.g., in an electronic yellow-pages application, everyone should know that one is likely to find "television," but not "transfiguration.")

Post-Coordinate Search in Télétel

One of the search modes in the Steria-Videopac database management system is "document search." It offers an interesting form of post-coordinate search by allowing for keyword dictionaries that have overlapping scopes. Imagine, for example, a statewide classified advertisement application. Each node in the database is a page containing the description of an article. Three keyword dictionaries are provided, each identified by a name: City, Transaction, and Article. The names are used as "attributes" in formulating Boolean queries. To find all cars or motorcycles for sale in Ottawa, one would formulate a query (the precise language of which is not important here):

CITY = Ottawa AND TRANSACTION = Sale

AND (ARTICLE = Car OR Motorcycle)

The query would be decomposed into four dictionary searches, and the corresponding set operations on the pointers would yield pages C and H, as indicated in Figure 13.1.

Note that there is hardly anything new in the above processing. Yet in the context of videotex such processing is seen as an innovation.

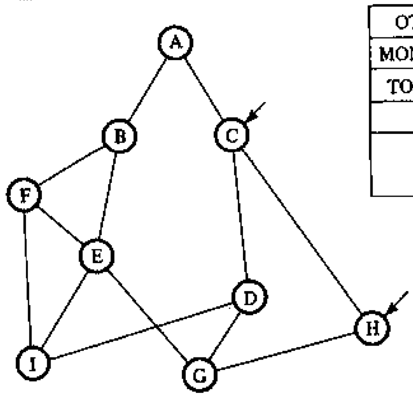
13.4 QUERY LANGUAGES

Interfaces whose complexity exceeds the possibilities of question-answer or simpler keyword methods are frequently based on *query languages*, with explicit syntax and command structure. Query languages can be classified according to four criteria:

1. The languages may be either artificial or natural. Most of them are artificial, for the simple reason that they are close to the programming languages used to implement the underlying system (sometimes the interface language is an integral part of a data retrieval system). Therefore, it takes less effort to design the language. Frequently such languages suffer from a lack of user-friendliness and they are hard to learn and to remember, especially for the inexperienced user. Natural languages, on the other hand, require complex processing in order to translate natural sentences into simpler and unambiguous commands acceptable to the rest of the system.

2. Artificial intelligence techniques may be involved in the process of interpretation and execution of interface language statements. This applies mostly (but not uniquely) to natural languages. Artificial intelligence can be used for parsing, for under-

VIDEOTEX DATABASE



CITY

OTTAWA	CAH
MONTREAL	BEF
TORONTO	DG

TRANSACTION

SALE	BCEH
BUY	DF
RENT	AG
GIVE	

ARTICLE

CAR	CE
HOUSE	ABD
MOTO	GH
HORSE	F

Keyword dictionaries (3)

A, B, C, . . . : Pages

<u>REQUEST:</u> ARTICLE = (Car <u>OR</u> Moto)
<u>AND</u>
TRANSACTION = Sale
<u>AND</u>
CITY = Ottawa

<u>EVALUATION:</u> ((CE) ∨ (GH)) ∧ (BCEH) ∧ (CAH) = (CH)

<u>RESPONSE:</u> pages C, H.

Figure 13.1 Post-coordinate search with multiple keyword dictionaries.

standing (or rather simulation of understanding) the sense of a query, and for searching and representing data in sophisticated knowledge bases and expert systems. An excellent overview of this area is found in Smith (1980).

3. The number of levels in system-user interfaces may vary. An increasing number of systems feature more than one level of interface language. One way of creating multiple levels is to provide nested subsets of available commands; only the simplest and most frequently used facilities are available on the first level, less easy and more powerful ones are available on the next, etc. This is analogous to various "easy" subsets of PL/1, Pascal, and other programming languages. Another method consists of inserting a *user agent* between the user and the database, thereby creating two interfaces: one between the database and the agent, and the other between the agent and the user. The former (called database interface) can be optimized for database interaction and typically has a concise syntax and command structure. It is primarily an interface for the user agent and other application programs; but it may also be available to the expert user. The other interface (user interface) is designed for the casual user, with friendliness, robustness and ease of use in mind; the user agent becomes a translator between the two language levels. In addition, the notion of user agent lends itself naturally to system distribution: it can be implemented in intelligent terminals or in access machines (in which case they are

shared by multiple users). The IC module in the French Electronic Dictionary or the keyword search implemented in the MUPID intelligent terminal (see Chapter 14) are examples of such user agents. Note also the resemblance between certain functions of the Prestel/Bildschirmtext gateway and those of a user agent.

4. Languages may be either procedural or specification. In a procedural language the programmer has to describe how to obtain a given objective; in specification languages he has to describe (specify) the objective, but not the method of obtaining it. The difference between the two is often blurred. As an example, we can say that data retrieval by menus is, relatively speaking, more procedural than is keyword access because in the former the user has to give a sequence specifying how to get to the desired data. It is not clear which of these two types of languages is more suitable for videotex; in fact, it is possible that this parameter is irrelevant in the long run.

Artificial Languages

Computer control languages—notorious for their rigid and unforgiving syntax, conciseness, and cryptic, hard-to-remember symbols—are classic examples of artificial languages. Such languages are designed for efficient use by experts. Of course, limited subsets of control languages can easily be learned by the casual user of general-purpose systems.

The situation is somewhat different with database query languages. Here commercial considerations often dictate that the systems be easily useable by nonexpert users, and considerable amounts of work have been invested in various types of user-friendly, easy-to-learn and error-tolerant interfaces (see Ball and Hayes, 1982). Obviously this is extremely important for videotex applications.

Examples. ORBIT is an example of a large post-coordinate keyword system for bibliographic citations. A typical query in this system would be

SET1 = Restaurant AND (Greek OR Italian).

If too many documents were found, then the user might add a further condition:

SET2 = SET1 AND Outremont

so as to reduce the number of restaurants to only those in the Outremont district of Montreal. In more advanced systems the user might be able to search the content of the advertisement for particular expressions so as to retrieve more precise information.

QUIC/LAW (of QL Systems, Ltd., Ottawa) is a full-text database featuring a Boolean default of AND. A request may consist solely of several keywords separated by commas, like

Greek, Restaurant, Outremont

which is interpreted to mean "Find all texts containing the words 'Greek,' 'Restaurant,' 'Outremont'." Text output is automatically sorted and ranked according to the aggregate frequency of occurrence of the keywords, so that text containing all three terms would be listed first. Since its inception the system has also added explicit Boolean algebra-based query commands for citation databases. Telidon-like user interfaces can be seen similarly as default-based languages.

A number of interesting interfaces to relational databases have beeen created. A microcomputer-based system called WHATSIT stores information in 3-tuples of subject, tag, and object—e.g., name, "phone," 7-digit number. Separators are either ' or s, giving a syntax such as:

"Bob's phone's?" reply: "Bob's phone's 737-8765."

WHATSIT can match on any domain of the 3-tuple for retrieval, and the user can add new tuples at any time by typing in the new information. Even though WHATSIT's power is relatively limited, the approach it uses is certainly worth pursuing further, as it seems to be suitable for simple videotex applications—e.g., personal file service.

Another well-known relational database management system is IBM's System R. Several interface languages have been conceived for interactive applications (the system can also be accessed by application programs running in batch), notably SEQUEL-2 and SQUARE. SEQUEL-2 is a powerful language equipped with commands for retrieval, data creation and update, manipulation of relations, and control of user access rights. Queries with attribute-value pairs and Boolean operators are embedded in SELECT-type commands for retrieval. Additional facilities for the specification of queries, such as built-in functions SUM, MAX, and AVG are available. These are applicable in cases in which the query specifies calculations involving a number of tuples—e.g., to find the average salary of all employees in a department, or to find the cheapest Greek restaurant in Outremont. Parameters of a command can be entered in any order, each preceded by a label uniquely identifying the parameter.

SQUARE is a language that is nearly identical to SEQUEL-2 in its capabilities; however it has a different syntax, called "positional." Here the composition of a command is a combination of menu selections and form-filling.

Query By Example is yet another example of an artificial language. It is a very popular relational query language based on form filling (for details, see Date, 1981).

Natural-Language Interfaces

The possibility of communicating in natural language with an information service has been received with enthusiasm as the ultimate man-machine interface. However, using natural language means ignoring computer's "psychology," thereby creating a mismatch of the same kind between two communicating entities, as occurs when humans are forced to learn cryptic command languages.

Currently the suitability of natural language interfaces is being intensely debated in the literature; the outcome will probably be that finer distinctions must be made concerning the class of users, situations, and applications for which such interfaces are really suitable. To quote one such conclusion (see Shneiderman, 1978): "Realistic applications for natural language would be situations where people have familiarity with the application area, data structure and queries, but are infrequent users. Typical situations that fit this description include library card catalogs, airline schedules or banking transactions". Here is a list of limitations and problems of natural-language interfaces that are treated in the same article:

1. Knowledge of natural-language syntax does not assure knowledge of the semantics of interaction or data content. Therefore unanswerable questions are often asked and, generally, the user may have unrealistic expectations of the computer's power and the content of the database.
2. The ambiguities typical of natural-language syntax may necessitate tedious clarification dialogs.
3. The overhead of natural-language interfaces is much greater than that of artificial languages.

Most natural language interface systems fall into two large groups: two-level interfaces (in which the database in question is a more or less standard system with an associated query language, and the natural language input from the user is translated by a user agent); and knowledge bases and expert systems (in which answers to queries are synthesized by highly nontrivial deductive processing). Both of these groups frequently rely on artificial intelligence—e.g., in the areas of parsing, dealing with ambiguities, data representation, and search strategies.

Examples. The typical pattern observed at the user interface is a clarification dialog that serves to elucidate the precise meaning and intention of a query. A simple example of such a system is SMART (see Salton, 1979). This system emphasizes feedback from the user as a means of refining and improving the accuracy of bibliographic searches. Although SMART is basically a command language system, it allows the user to enter value judgments about previous searches, and these judgments are automatically included in the parameters for subsequent searches. An example of feedback is assigning weights to keywords that are judged relevant to the search. Another is the automatic incorporation of additional keywords derived from citations but not used in the user's original query.

ROBOT (see Harris, 1977) is a commercially available natural language translator (user agent). It accepts the user's inquiry in English, produces a number of possible parses and then proceeds to consult the database to choose the most likely interpretation. If there is no clear winner, the user is asked to clarify the query. ROBOT handles relatively simple inquiries in a restricted commercial-style database. For example, it can successfully process queries such as the following: "Singles in Chicago?" "Who are the secretaries?" "Which of them live in Detroit?" The latter question would yield a list of secretaries living in Detroit rather than a reply such as: "No singles in Chicago live in Detroit." ROBOT requires 200 kB of memory and takes less than 1 second to process a typical query (not including data retrieval time).

13.5 BROWSING IN DATA SPACES

A database user is comparable to a shopper who may be in three basic types of "shopping situations":

1. He needs a pair of tennis shoes, and he knows he can either find them at Woolworth's on the ground floor, right side, or he can order them by mail. He knows precisely what he wants and where to find it.
2. He needs a pair of tennis shoes but does not know precisely where to get them, so he goes to Macy's, looks at the floor plan, and finds the sporting goods department. Thus, he knows what he wants but has no precise idea of where to find it.
3. He wants to buy a birthday gift. Not knowing exactly what it should be nor where to get it, he goes on a shopping spree to a fashionable street and and gets inspiration by looking around.

The first two situations are like those of a database searcher who is after a precise piece of information (e.g., the telephone number of Mr. Smith). In a case like the first one he knows the database and proceeds directly to the formulation of a precise query (this is the typical pattern in data processing); in a case like the second he first might have to learn something about the system's files or indexes and might find after all that the desired information is not available in the database.

The third situation corresponds to what is usually called *browsing* or *navigation* in a database (or data space). The user must inspect a number of data items (and learn something about them or their existence) before making up his mind as to precisely what data would best fit his needs. It is characteristic of this kind of situation that it is impossible to formulate exactly a query beforehand; the "query" is a target that moves as the searcher gradually acquires knowledge on what data is available, and in which form. Similarly to window shopping, advertisements, catalogs, information booths, and floor plans, which are designed to inform shoppers before they buy, databases designed to support browsing users must provide appropriate navigational aid, or "metainformation." We note that:

- the distinction between browsing and the final act of retrieval is not clear-cut: browsing *is* a form of retrieval
- browsing is not an aid for the naive or occasional user; it denotes an access mode, not a user attribute.

Current database interfaces typically do not support browsing very well; in videotex this is a problem of particular importance because most page-oriented databases contain heterogeneous information with no simple overall structure. In current videotex databases the main navigational tool lies in organizing the data in a hierarchy of indexes; however, users commonly get disoriented, for reasons discussed elsewhere. A similar growing need for efficient browsing is experienced in the area of record-oriented databases.

A database browser, in order to navigate in the data space successfully, should be able to perceive the space at different levels of detail as through a series of "windows," or telescopes of increasing power. At any time he should be free to switch between levels, either "downwards" to see fewer items in more detail, or "upwards" to see a larger region of the space including, importantly, the indication of his current position or previous trajectory. On each level the user should be able to travel horizontally to explore neighboring items (as one does with a microfiche reader). Displacements in higher levels would correspond to fast travel to remote areas of the data space (see Lochovsky and Tsichritzis, 1981).

Such a view is appealing because it calls upon a natural orientation reflex present to a certain degree in every human being. In the absence of such explicit navigational mechanisms the browser will try to build mental models of the database in which he can navigate. However, his model will likely be unrealistic and result in frustration, unless he is provided with efficient and realistic means of model-building.

Two problem areas appear immediately when one contemplates the design of a database that has such features. First, there are technical obstacles to the simulation of zooming and displacement: considerable display and processing power are required. Second, and on a more fundamental level, there is the problem of how to define a data space that can be consistently enlarged or reduced. If the database were, say, the surface of the Earth, then zooming would amount simply to stretching along two dimensions. But what are the "dimensions" along which one can travel and which can be consistently stretched and contracted in a database that contains abstract and mutually interlinked items? What should be the interpretation of "closeness" of two items in the data space? For example, a relation in a relational database can be viewed as a multidimensional entity, each attribute being a dimension. At least some of the attribute values must be transformed into geometric coordinates, others into different, easily visualizable form (e.g., color), and still others kept in written form (e.g., detailed job specifications). Who should decide on all such questions: the database designer, the administrator, or the user?

When trying to implement the above ideas practically, a number of compromises must be made. For example, one has to abandon the idea of continuous zooming, which is both technically and conceptually next to impossible. One should stay with a small, fixed number of levels of detail, each showing possibly different coordinates (attributes) of the objects displayed.

This principle is implemented in embryonic form in some menu-based videotex databases in which, by proceeding towards deeper levels, one gets more "detailed" information. However, this is only a kind of "mental" zooming that is not analogous to movement with a physical space—an analogy with which we are so familiar—nor does it provide for effective navigation.

Spatial Management of Data

A significant development towards the realization of the above-outlined ideas is a system called Spatial Management of Data (SMD), reported by Herot (1980). It works in conjunction with a conventional relational database management system, INGRES. Objects (usually tuples in a relation) processed by SMD are defined in several "data surfaces," each displayed on a separate screen. Three screens are provided in the experimental system, the middle and right screen showing, respectively, an enlarged portion of the left and middle screens. "Enlargement" here is to be understood as nonlinear, in the sense that an enlarged image generally contains some coordinates (attribute values) not represented in the original, and vice versa. It is a mixture of geometric and "abstract" enlargement, and the latter simply meaning that more detailed information is included in some form. The decision as to which attributes are to be interpreted as geometric dimensions and coordinates, and at what level of enlargement, is up to the database manager. Currently enlarged regions are shown highlighted on the more global screens.

This browsing apparatus is controlled by a joystick. Moving the stick causes a magnified window to be shifted in the x-y plane (as in a microfiche reader), with corresponding movement of a highlighted area on the other screen(s). Turning the stick causes passing of control to a less or more detailed plane.

An example given by Herot and paraphrased here shows how SMD techniques can be used to inspect a relation containing ships as entities. The attributes are, among others: the ship's name, type, nationality, radio code, size, degree of readiness, captain's name, and technical details. The representation of the global level (in which all ships are represented) was chosen as follows. The vertical and horizontal coordinates are the nationality (U.S. or Russian) and type (there are five different type values), respectively. Each ship is represented as a rectangle whose size indicates the ship's size and whose color indicates its degree of readiness (see Figure 13.2(a)). On the second level, groups of ships can be examined with some more detail fitted in (ship's name, code, and captain). On the third level, individual ships are seen in detail with a silhouette that stands for type, and all other technical data are written in or coded by color (see Figures 13.2(b) and (c)).

SMD provides for the database administrator a language in which he can define the layout of screens and the shapes (called icons) to represent tuples in the relation. An icon may be defined at several levels of detail (this is the basic mechanism used to achieve the zooming effect). Attributes in a tuple can be programmed to control the visual attributes of icons, such as color, size, orientation, position on the screen, and choice of shape. Shapes must be pre-defined and may come from digital storage (as bit arrays) or videodisk (photographs). Symbolic information to be inserted into tuples comes, of course, from the relational database.

The underlying relational database is accessed by the query language QUEL. Although the objective of SMD is to locate and access data spatially, in some cases it is more convenient to locate a piece of data through a standard query expressed in QUEL (e.g., to find the size of a given ship). The two modes of access (browsing and by query) are therefore linked and can be used simultaneously. The completion of a QUEL query results in highlighting (blinking) of the selected tuples on the screen(s) and eventually in automatic repositioning the user's active search position.

When contrasted with conventional database management systems and query languages, we would argue (loosely following Herot) that SMD offers several advantages:

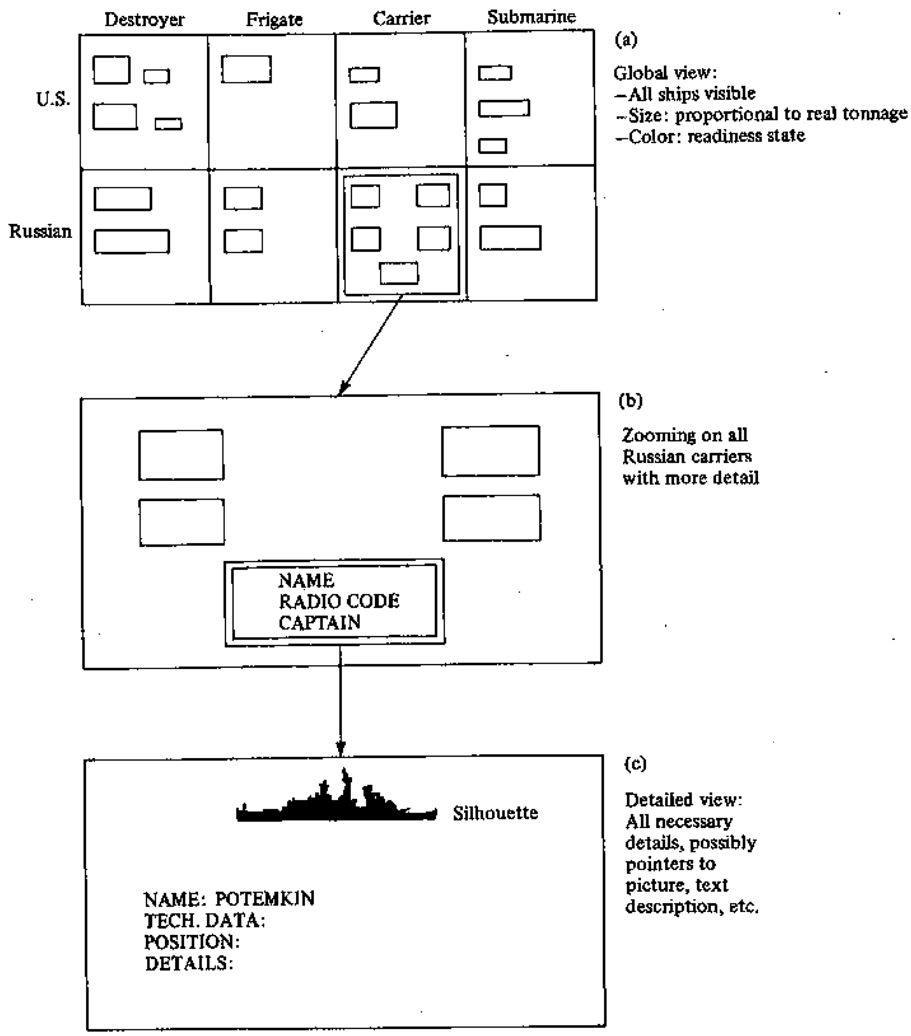


Figure 13.2 Three views of ship database in SMD.

1. Motion through the database is simple and natural: the joystick and multiple displays provide an intuitive, non-verbal and non-symbolic language that is easy and fast to learn.
2. The presentation of the data encourages browsing: in effect the database is its own directory, and makes the contents and structure apparent in a unified fashion. The contents and possibilities of a database (e.g., the kind of queries that can be answered) are quickly seen by the user.
3. While browsing, the user is almost always presented with more information than he immediately needs. He can actually see the neighborhood of his present focus of

attention, a collection of items similar in some sense to the one actually examined. Graphical representations can help in perceiving trends in sets of data. If the ship database contained information on position and velocity of ships, one could, e.g., detect that a fleet of ships is heading towards the Falklands.

4. SMD can accomodate in a uniform way a range of data types such as alphanumerics, photographs, and drawings.

From this brief description it should be clear that many characteristics of SMD are desirable in videotex: simple interface, intuitive presentation of data, and accomodation of graphical presentation types. Telidon-like geometric drawing instructions would certainly encourage and facilitate the construction of graphical representations of data. For these reasons, the principles underlying SMD should be seriously considered for inclusion into videotex user agents or gateways. A similar approach using a single high resolution display has been reported by Feiner et al. (1981).

Interface Languages for Videotex

It is not easy to evaluate the applicability to and potential impact upon videotex of the great variety of existing interface languages. There seem to be two groups of criteria for such an evaluation. The first is related to psychological factors such as simplicity, error tolerance, and friendliness. The second concerns efficiency and resource utilization issues.

As far as psychological adequacy is concerned, note that people are able to learn and manipulate quite complex systems and machines when they are sufficiently motivated (economically or intellectually). Therefore, the problem of whether a given interface mechanism is applicable or "simple enough" for videotex is closely linked to the service provider's ability to find useful and interesting applications. It is therefore essential that the interface not be studied in isolation from applications. Further, with gateways it is no longer an imperative that all available services have interfaces simple enough for mass use; insofar as videotex is regarded as a link to external services, all existing interfaces may become "videotex interfaces" for those who are interested.

The key to successfully dealing with the performance problems (those, for example, inherent in most natural language interfaces) is our ability to modularize, distribute, and, if necessary, duplicate system functions. As already discussed, this is relatively easy to achieve with two-level interfaces or with keyword systems in which indexes and user agents might be implemented separately from the main databases.

13.6 TOWARDS A USER PSYCHOLOGY

Nobody nowadays denies the importance of human factors in man-machine communications; and yet, we have no reliable methodology to evaluate, let alone predict, how appropriate these interfaces are for various (groups of) users. The situation is similar in other neighboring disciplines, such as the psychology of programming, where a large body of ad hoc experimental results exist, but without any underlying theory capable of explaining or extrapolating the results.

Since videotex is a blend of consumerism and computer science, it is doubly important that its consumers be pleased. Most users do not need videotex as badly as some promoters would like to make them believe, and they will use the system only if sufficiently motivated by attractive services and interfaces.

The purpose of this section is not to summarize the numerous (videotex) user behavior studies published to date, but rather to outline what might become in the future a coherent discipline of applied psychology of computer users. We do this in the conviction that videotex designers should be at least summarily acquainted with this new emerging discipline, which eventually should evolve into "videotex psychology."

Following an inspired article by Moran (1981), user issues are not really neglected, but rather dominated by non-psychological approaches such as the "technologist's" and "designer's" attitudes. According to the first, it is not psychology, but new achievements in hardware and software that are the determinants of better interfaces—e.g., better displays and higher data rates. The designer's story is essentially that better interfaces can be obtained merely by more careful designs whose creation is best left to the designer himself. These and similar approaches seem simply to be telling the user what is good for him, and not reflecting the fact that any optimization of man-machine (and any other) communications must take into account both communicators, and not only the machine (despite the fact that the man is by far the more complex and less known of the two participants!).

Moran states that "the main contribution of an applied (user) psychology is to *reliably* assure satisfactory user-computer interaction, just as we can reliably assure other performance aspects of computer systems, such as response times or crash recovery." To achieve this, one must discover the principles governing user's behavior; then one may be able to predict or control it in different situations during interaction.

So far we know relatively little about the structure of user behavior simply because it stems directly from the immensely complex build-up of the cognitive mind (leaving the emotional and motivational part aside). Roughly, a rational user's behavior is determined by four factors: the user's goal (e.g., to get the numerical results of a crucial calculation); the syntax and semantics of the interface language, which determine the span of his possible actions; the user's experience (novices often use entirely different strategies than do professionals); and the limits of the user's information processing capability (e.g., to memorize and formulate complicated error-free commands). It appears that this latter is directly linked with the capabilities of the uppermost element in the hierarchy of (two or three) data storage/manipulation "units" that are said to compose the rational mind. The last two factors seem to be linked: experience can increase processing capability, for example, by building strategies that are best suited to the user's ways of thinking.

Short of having sufficient insight into the deeper structure of user behavior, we consider the user frequently as a black box and concentrate on the measurement of some more or less easily observable aspects of his input/output behavior. Some of these aspects are error rates, reaction time, rate of learning, and the overall performance of a broad set of tasks.

Methods and approaches of user psychology can be roughly classified by a number of criteria:

1. The primary data collected in an experiment may be of qualitative or quantitative nature.
2. The approach may be either structural or black-box. Structural methods attempt to explain why observed phenomena happen, in terms of deeper knowledge (or hypothesis) about the process causing a given behavior. The black-box view is concerned mainly with description of the observations. This is at once its strength and weakness: because of the absence of an ambition to build an underlying theory, the view is applicable to almost all situations. By the same token its power to explain the observations is limited. However, sometimes even the black-box approach can give rise to mathematical models, providing a certain capability to extrapolate (but not to explain).
3. The environment may be either controlled or uncontrolled. The former yields less, but more reliable, data than does the latter.
4. Observation may be either objective or subjective. The latter is centered around introspection, a tool whose value is being extensively debated.

A few examples will illustrate the essence of some of the above approaches. Most experiments concerning the overall evaluation (often comparative) of database query languages is of the quantitative, black-box, controlled, objective kind (see, e.g., Reisner, 1981). The same holds true about videotex-related experiments to evaluate different character sets and the effectiveness of keywords.

An example of the black-box approach yielding a mathematical model is a formula for the time necessary for a user to position the cursor on a display screen, as a function of the distance D to be moved and the size S of the target (see Embley and Nagy, 1981):

$$\text{Time} = K_0 + K \log_2 (D / S + 0.5) \text{ seconds}$$

where K_0 and K are constants depending on the pointing device (joystick, mouse, step-keys).

Structural experiments are rare in user-oriented behavioral research. An example from the same reference is the decomposition of text-editing tasks into the constituents: goals, operators, methods for achieving the goals, selection rules for choosing among competing methods, and examination of the execution times for these constituents under various conditions.

Finally, we touch upon the conclusions drawn from experiments. A well-known phenomenon is the overemphasizing of a (usually differential) result—such as often found in comparative studies of query languages. It may turn out that the issue in question is a non-problem, or that it is simply the result of whether the subject's dominant way of thinking is left or right hemispherical (symbolic-deductive or visual-intuitive).

14
Distributed Processing in Videotex

In the opinion of many, videotex has not quite fulfilled the euphoric expectations aroused at the time of its initial development. The actual number of Prestel users, for instance, is far below the numbers originally projected (see Arnold, 1981). Also, all too often one hears statements such as "videotex is a technology in search of applications." Such statements indicate a certain disappointment of both service providers and consumers. Part of the problem seems to be the familiar chicken-and-egg phenomenon: people will not buy videotex terminals before a sufficient variety of useful and amusing applications are available, and service providers will not venture into providing such applications before a sufficient number of consumers are in sight.

Another factor is that technical characteristics (such as the numerical keypad, access by menu selection and limited interactivity) imposed by early systems and inherited by many later models effectively hamper and complicate the implementation of more imaginative services. Just consider how awkward the selection from alphabetical system indexes with numerical keypad is, and compare it to the simplicity of keyboard operation. The same applies to keywords (which can be entered through keypads in a manner similar to using the letters that accompany numbers on most telephone dials, but only at the price of unnecessary ambiguities and other complications).

There are many ways to improve the functionality of videotex, and thus prepare the ground for breaking the vicious circle. According to the proponents of geometric coding philosophy the key lies in improved graphic capabilities. Although geometric graphics are certainly superior to CEPT-like mosaics in an absolute sense, it is far from certain that they alone are sufficient to make videotex "fly." According to another argument, most useful information in videotex is in textual form; graphics are a useful complement, but the possibilities offered by mosaics are sufficient for the purpose. This is the view of most European administrations, which consider accessibility to external and transactional applications more important than better graphics.

Still another direction of improvement lies in gifting the terminals with more intelligence. More precisely, one can open up the resources already present in most modern terminals for uses other than mere interpretation of user commands and incoming character strings: namely, for the local execution of programs in support of existing and new applications. In general, such programs can reside permanently in the terminal's memory, can be written by the user, or can be received from a distant computer as a *teleprogram* (or telesoftware).

Systems designed around intelligent terminals and the systematic use of telesoftware offer not only the enhancement of existing applications, but have the potential to open a new dimension of services with far-reaching implications that will be realized only gradually. In effect, videotex networks (sometimes considered to be feebleminded cousins of computer networks) may become at once full-fledged universal distributed processing systems offering a virtually unlimited range of applications.

14.1 INTELLIGENT TERMINALS AND TELESOFTWARE

Intelligent videotex terminals can be made in three ways.

1. Personal computers can be fitted with hard or soft modules to interface with videotex.
2. Personal or small business computers can be designed integrally with videotex capability. Examples include Nabu's cable-based computer fabricated in Canada (see Nabu, 1981), and the BBC Microcomputer (see Moir, 1982). The latter was intended for the more serious viewers of a series of programs on computers on the BBC Television. This computer was to serve as support for demonstrations, including downloading of telesoftware through Prestel.
3. Enhanced decoders with telesoftware capability and local programmability can be designed. A significant example is MUPID (discussed in the next section), which was developed in Austria.

The difference between these three alternatives is not only technical but, perhaps more significantly, economic and psychological. Before discussing in more detail the functional possibilities offered by intelligent terminals, let us briefly compare from economic and psychological viewpoints the three alternatives. While personal computers made according to 1. and 2. above certainly offer much functionality (possibly too much for the average user), their cost ($1000–15,000) largely exceeds the cost of videotex decoders (which is in the hundreds of dollars). This puts an a priori limit on the number of potential customers. Further, the buyer of a personal computer is buying a *computer*, and not just an enhancement to his TV set, as in case 3.; the psychological difference should not be underestimated. But perhaps the best marketing strategy is to offer an inexpensive videotex terminal/personal computer combination.

Enhanced terminal decoders cost less because

- they are based on a volume-fabricated product
- they offer less function than do typical personal computers (e.g., they do not require floppy disks and multiple compilers).

The idea behind telesoftware, as originally conceived, is quite straightforward: it is to use videotex channels, of whatever nature, for the transmission of computer programs. The downloaded programs are stored in an enhanced terminal and then executed under user control. The first telesoftware experiments were carried out around 1977 in the U.K. over the Oracle teletext system (see Hedger et al., 1980). The necessary decoders were built by Mullard, Ltd., and consisted of a teletext receiver augmented internally by a microprocessor, 4 kB of memory to hold the program, and 2 kB of PROM for a local control program. This terminal was not user-programmable.

Today, numerous videotex-like systems feature downloading of programs to the terminal. However, most projects reported to date are quite ad hoc applications, without aspiration to exploit systematically all the inherent potential of telesoftware. One example is Playcable, operational in the U.S. Subscribers are equipped with special-purpose terminals capable of receiving and executing software for screen-based games. The terminal is not programmable by the user and no details about the system operation, language used, etc., are disclosed. Programs cannot be stored on local secondary storage. When the terminal is turned on, the directory of available games is displayed. Upon selection the desired program is transmitted and run in the terminal's microprocessor. Cooperation with other systems is practically impossible.

Another well-known system is run by the Council of Educational Technology (CET) in the U.K. CET has acquired and developed a number of educational programs for various microcomputers, among them the BBC Microcomputer, Pet, Apple, and Research Machines 380Z. The programs are stored and distributed through Prestel, although they do not necessarily use Prestel presentation code (see Thompson, 1981). Figure 14.1 shows a sequence of pages leading to the downloading of a program. It should also be mentioned that networks such as The Source feature "software banks" through which users can make their programs available for other participants.

Telesoftware Protocols and Standards

In general-purpose computer networks, software transfers between participating computers are supported by *file transfer protocols*. The aim of such protocols is to ensure error-free transmission, and they are often designed integrally with the operating systems. The situation with telesoftware is somewhat different in that no such protocol was originally designed into most videotex systems. An additional problem at present is the lack of a universally accepted language for telesoftware. This will remain a major obstacle to the spread of telesoftware applications until intelligent terminals turn into even more intelligent multilingual general-purpose computers, or until a standard language is agreed upon. So far, there is no consensus concerning even the level of such a language. Proposals have been floated for machine language, intermediate level (P-code) language, and high-level languages (Basic, Pascal, ADA).

In Prestel and the U.K. teletext, program pages are handled by the central facilities

Figure 14.1 Partial sequence of pages leading to downloading of a CET telesoftware file from Prestel: 1. Welcome page; 2. Subject index; 3. List of programs on languages (partial). Note that page numbers do not necessarily correspond to the sequence of selections.

```
CET                              21147Ø141a   Øp
 HOUSE                              DESCRIPTION
 Simple drill and practice
 in 5 languages

 Dewey No.  4Ø7  Language Study
 Age level  any
 Description
  Simple vocabulary practice in 5
  languages - French, German, Spanish,
  Finnish and English.
  The computer draws a picture of a
  house piece by piece and at the
  same time displays the word
  corresponding to that piece. The
  user then types the words himself
  as the picture is redrawn. If stuck
  press the space key for the next
  character.

 Key 1 to continue

 9  Other LANGUAGES programs
```

4

```
CET                              2114Ø014a    Øp
 HOUSE                                   HEADER

 7 frames: 6 at 1Øp  1 at 2Øp
 Machine    38ØZ
 Minimum memory size   2ØK
 Program language   BASIC
 Dialect   Extended Basic Version 5
 Copyright notice
  Copyright 1981 Advisory Unit for
  Computer Based Education.
  This package is sold for use in a
  single institution only. Permission
  is granted to make copies for
  backup and archive purposes and
  for no other purposes.
 To obtain program start
 TELESOFTWARE ROUTINE

 2  Guide to using CET TELESOFTWARE
 9  Other LANGUAGES programs
 99 Full program index
AHOUSE.SAS!L7!ZØ51
```

5

Figure 14.1 (continued) 4. Description of HOUSE; 5. Technical details on HOUSE.

in exactly the same way as are display pages. However, they are treated completely differently at the receiving end: telesoftware pages have to be recognized as such and provisions must be made to assure their error-free transmission. More precisely, the following technical issues must be taken care of:

1. Error-free transmission of program files must be ensured. If the error protection built into the supporting videotex system is insufficient, additional protocol facilities (at the transport level)—e.g., automatic request for retransmission and/or more powerful error detection—must be added.

2. In early experiments the received programs had to be typed manually from the screen into the terminal's program memory. This is clearly unacceptable and therefore telesoftware pages must be automatically recognized and loaded.

3. Most videotex systems employ 7-bit character sets, and some codes may be subject to changes done automatically by the database software. For example, trailing blanks in a display line can be suppressed and replaced by carriage return and line-feed characters. Some codes from the control set can cause undesirable effects if they occur in the file being transmitted. Since telesoftware (especially in machine language form) is basically binary 8-bit code, it must be mapped from the (0−255) range onto the set of allowable 7-bit characters. One method is to use two 7-bit characters per byte (as in the Oracle experiments); another is based on inserting escape sequences into the stream of 8-bit codes, causing transcoding of the subsequent bytes into the permissible range. This method is used in the CET system (see Brown, 1981).

4. No language standard exists at the present for telesoftware, nor will one exist in the near future. (It is reassuring to be able to assert at least something certain about standards.) Telesoftware programs are produced for a variety of languages and dialects. Consequently, before downloading, some kind of option negotiation must take place, whereby the terminal makes sure that the program it is about to receive fits its language,

memory, and other capabilities. This option negotiation is performed manually in today's systems; however, it is being considered for automation by CET. This may eventually lead to a "metastandard," in the sense that instead of specifying a standard language, one would have a standard way of choosing from a set of languages. New languages might be added to the set by a registration procedure similar to the registration (by ISO) of new graphic alphabets and escape sequences.

5. An easy method of searching the directory of telesoftware programs should be available. Keyword access is a good candidate for this; once the desired program is located, the sequence of necessary actions (downloading, etc.) can be executed automatically by the resident control program.

14.2 THE AUSTRIAN APPROACH TO TELESOFTWARE

A significant step towards the systematic utilization of intelligent terminals and telesoftware has been taken in the Austrian public videotex network, described by Maurer (1982), and by Maurer and Sebestyen (1982). The core of the design is MUPID, a Multipurpose Universally Programmable Intelligent Decoder that has been available since October, 1982, for a monthly rental rate of about $6.00. The list of applications so far available (see below) is impressive and illustrates well the possibilities of such an approach. What makes this even more remarkable is the fact that the central database system is a practically unmodified Prestel software package. The potential of intelligent terminals and telesoftware could be utilized even better in a system whose design integrates these concepts from the beginning.

Structurally, MUPID is a single-board extensible microcomputer. The processor is a Z80A operating at a 4-MHz clock rate. In the standard configuration, it contains 16 kB of EPROM for resident data and software, and 64 kB of writeable memory, used partly as a bit-plane display memory (320 × 240 pixels, 4 bits/pixel) and partly for the storage of locally written programs and/or telesoftware. Two serial interfaces are provided (e.g., for printer and cassette memory), as well as 16 analog inputs (e.g., for game controls and graphic editing). The unit contains a full ASCII keyboard with a numeric block including a few dedicated control keys, a sound generator, and a speaker. The terminal can handle all presentation options (mosaic, geometric, and photographic); at the present it can decode the Prestel standard and PLPS-like geometric PDIs (available also for local programming in Basic). New standards, such as full PLPS or CEPT, can be accommodated simply by replacing some EPROM memory modules. Alternatively, this could be accomplished by telesoftware as well—a solution to the problem of multiple standards. MUPID is connected to the TV set through a standard plug. MUPID can accommodate transmission speeds of 1200/75 b/s (standard videotex), 300/300 b/s (applications for teletypes with acoustic couplers), and 2400/2400 b/s (to be introduced in Austria in 1984), which is necessary for two-way transfer of files. Additional features built into the terminal's firmware are cursor positioning, scrolling, and the creating of "windows" over parts of the screen (e.g., for teleconferencing).

A considerable number of MUPID applications have been or are being realized; Maurer classifies them into five categories:

1. The central database can be used as a program and file library. This obviates the need for secondary storage in the terminal and opens the way for sharing and exchanging programs among users. A Basic interpret is available as a teleprogram for local programming. Educational software also belongs in this class.

2. Certain applications are intended to improve the quality of videotex service for the users. Typical cases are teleprograms enabling search by keywords (although the main database can be accessed only by numerical selections) and local animation or other manipulation of downloaded graphic images. For example, a map of a city can be displayed; an accompanying teleprogram indicates by flashing the shortest route between two points, or the location of historical sites. Most of these programs are "hidden" from the user—that is, they are loaded and started automatically upon the selection of a given page. In these applications the terminal acts as a user agent. Going further, one can imagine sophisticated users writing their own user agent programs—e.g., for regularly scanning the news pages for the occurrence of the word "Lebanon," and storing the corresponding material for later display.

3. Telesoftware is available for service providers. Text editing and graphic editing (and other utility-oriented) teleprograms are available that turn MUPID into a service-provider terminal.

4. Another application involves improvement of communications among users. Teleprograms exist for written user-to-user conversation (on split screen), for teleconferencing, and for messaging. Another group of programs enables games to be played against another user or against the terminal or a distant computer. For some games the terminal can be teleprogrammed to act as an umpire or score-keeper. Still other possibilities are "tele-lottery" and "tele-gambling."

5. Telesoftware can be aimed at off-loading external computers and gateways. Dialog management with external computers, and certain types of transactions can be simplified by the use of intelligent terminals. For example, the data-gathering function in the Prestel/Bildschirmtext gateway might be carried out in the terminal.

One might be led to see telesoftware as merely a means of processing distribution (which is in itself a desirable goal): after all, teleprograms could be executed in suitably dimensioned access machines as well. However, this is definitely not always possible. First, many applications require interaction speeds and frequencies that are unrealizable with remote shared computers (e.g., animation in games, educational courses, graphic editing). Second, certain functions are impossible to execute elsewhere than in the terminal itself: take, for instance, the above-mentioned capability of MUPID to be reprogrammed for different presentation standards. If done in an access machine (instead of in the terminal), this would amount to real-time transcoding from one standard into another, a task beyond the capability of today's technology.

In conclusion, we can comment favorably on the potential of Austria's approach to videotex. That approach may well prove to be a much-needed impulse for the final take-off of videotex.

14.3 TOWARDS DISTRIBUTED PROCESSING IN VIDEOTEX NETWORKS

The ideas of the preceding section can be extended, in a natural way, to lead towards the development of a network of nodes (intelligent terminals, access machines, and external computers), each of which is programmable by others. Such a system would have a number of unique characteristics that would place it among the most advanced distributed processing computer networks. These characteristics include:

- very large numbers of terminals
- the unifying influence of videotex (as opposed to external) databases, serving as a common repository of information and as a communication medium
- nodes capable of receiving teleprograms from other nodes, and of supporting their execution
- applications in terms of the cooperative execution of programs in several nodes, preceded (and possibly interleaved) by the downloading of appropriate teleprograms
- interaction between coöperating programs that takes place over large distances.

The most obvious place in which such cooperative execution can take place (as in today's systems) is the terminal-access machine pair. However, it would be shortsighted to exclude other possibilities from the design, such as teleprogramming of one access machine by another, sending teleprograms to and from external computers, and even "reverse telesoftware" (the execution in an access machine of a program originating in a user terminal, which would add realism to the meaning of "user agent" as an entity or algorithm sent into another node with a specific task to accomplish.)

What is needed in order to realize all these possibilities is a coherent network architecture, to be developed along the lines of the OSI reference model. It should provide a convenient framework for the development of services, using telesoftware as a tool. The design should draw from the lessons and experience gained in the design of similar distributed database or transactional systems.

In particular, a number of technical issues must be resolved:

1. The architecture should provide for a reliable communication infrastructure required for the transfer of files (teleprograms, displayable pages, and other data). Existing standard transport protocols may be applicable for this purpose.
2. System-wide naming and addressing conventions must be established for transmission of teleprograms and communication between concurrently executing programs.

3. A suitable programming language for system-wide use must be agreed upon. Sychronization primitives must permit at least simple cooperation between concurrent programs.
4. Ownership control over teleprograms and other files must be established.

Within the above-outlined framework, many classes of applications are possible. Referring to Figure 14.2, we see that they can be characterized by the data paths involved.

- traditional videotex data retrieval (path a in the figure)
- traditional telesoftware: downloading of programs and local execution after disconnection from the source (b, then c)
- local processing of displayable material by telesoftware or locally written programs (d); example: automatic animation
- image manipulation under user control (that is, by a teleprogram interacting with the user) (b, then c and d); examples: games, educational programs
- interaction between a program in the terminal decoder and both the user and service computer (b, c, d); examples: various enhancements of database search procedures
- interaction between user terminals (c, b, e or c, f), such as in games, conferencing, messaging

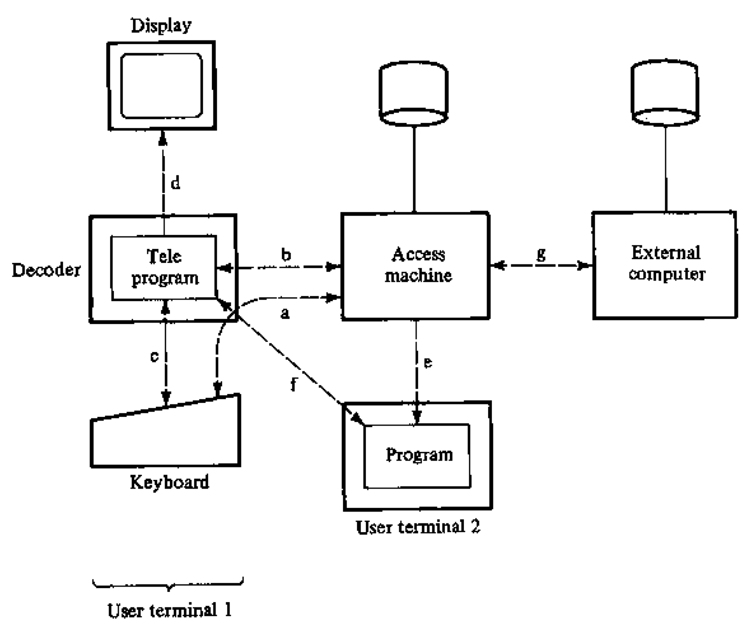


Figure 14.2 Interactions in distributed processing videotex system.

- sending files (data or programs) to distant locations for storage and eventual execution (b, e, g).

This section should be regarded as one possible scenario for the future development of videotex. For 4004 more scenarios refer to Plummer et al. (1979) and to Tydeman et al. (1982).

15 Efficiency and Performance in Videotex

There is no need to emphasize the importance of good performance evaluation tools for those who implement and operate videotex systems. The literature on computer performance is vast and flourishing, but there is very little published work that is specific to videotex. This is especially true for approaches involving experimental data—e.g., on page access frequencies, and session lengths. In this chapter we illustrate a few possible applications of standard techniques to videotex: queuing theory for building overall models of videotex networks, a Markov chain model to determine page access frequencies from parameters of user behavior, and a probabilistic approach to optimizing teletext transmission sequences.

15.1 QUEUING MODELS

When designing an entire videotex system, it is important to determine its overall performance in terms of response time, throughput, and resource utilization. From the many methods available (such as simulation, which is often too costly, and various rules of thumb, which are often oversimplifications), analytic queuing models seem to offer a good balance between cost, precision, and extrapolating power. Queuing models have become very popular for the evaluation of computer systems and networks in general; the tendency seems to be moving toward complicated models that structurally fit well the underlying system even when they are only approximately solvable.

In this section we give a few simple examples of how queuing theory can be applied to videotex systems. For the supporting theory and more elaborate (and realistic) models the reader is referred to specialized literature, such as Kleinrock (1976), Allen (1978), and Kobayashi (1978).

Classical viewdata networks differ in the following, mainly quantitative, respects from other interactive computer networks:

1. There are a large number of terminals (100–1000 per access machine, and tens of thousands nationwide).
2. Service time typically consists of short periods of CPU activity and then periods of waiting for access of secondary memory (usually 1–2 accesses). Therefore there is less need for time-sliced processor operation than in general time-sharing systems.
3. User think times (see below) are quite long (15–30 seconds between requests) because the amount of information to be absorbed before a new request can be generated is quite large (e.g., a page). (The above two points are valid for page-oriented retrieval only.)

Open and Closed Models

Before proceeding to calculations, we discuss qualitatively two classes of queuing models that can be used to evaluate videotex systems. Examples of the two types of models are shown in Figure 15.1; the notation is summarized later. The underlying videotex system is shown in Figure 15.1(a).

In both models there is a fixed population of K terminals generating collectively a stream of page requests arriving to the access machine at a rate of λ(requests/second). It can be obtained (for example) as Kr, where r is the rate at which each user generates his requests. Taking $r = 1/30$ seconds and $K = 300$ will result in $\lambda = 10$.

The access machine is modeled in this example by a single server with an associated waiting line (queue). The (average) service time (in seconds) of the server is $\bar{s} = 1/\mu$; it includes all times involved: CPU execution, channel time, disk access time and, eventually, communications time. More precise models would include servers and queues for each component.

The principal information to be obtained from the models is the server utilization $\rho = \lambda/\mu$ and the average response time $\bar{w}$ (the time a request spends in the dashed box of Figure 15.1). Clearly, stable operation can be achieved only when $\lambda < \mu$.

In the *open* system (Figure 15.1(b)), λ is considered to be given, independent of the response time and of the number of requests ("customers") already in the system. Therefore, situations can occur (e.g., due to adding new terminals, or in unusually heavy traffic) in which λ approaches or exceeds μ; this leads to (theoretically) unbounded queue size and response time. For this reason such models are said to be *infinite*: there is a (theoretically) unlimited supply of requests. All K terminals continue to issue requests regardless of whether and when the responses are received.

In *closed* (cyclic, finite) models there is a fixed number K of customers (visualized as tokens) circulating in the system. In the example of Figure 15.1(c), commonly called

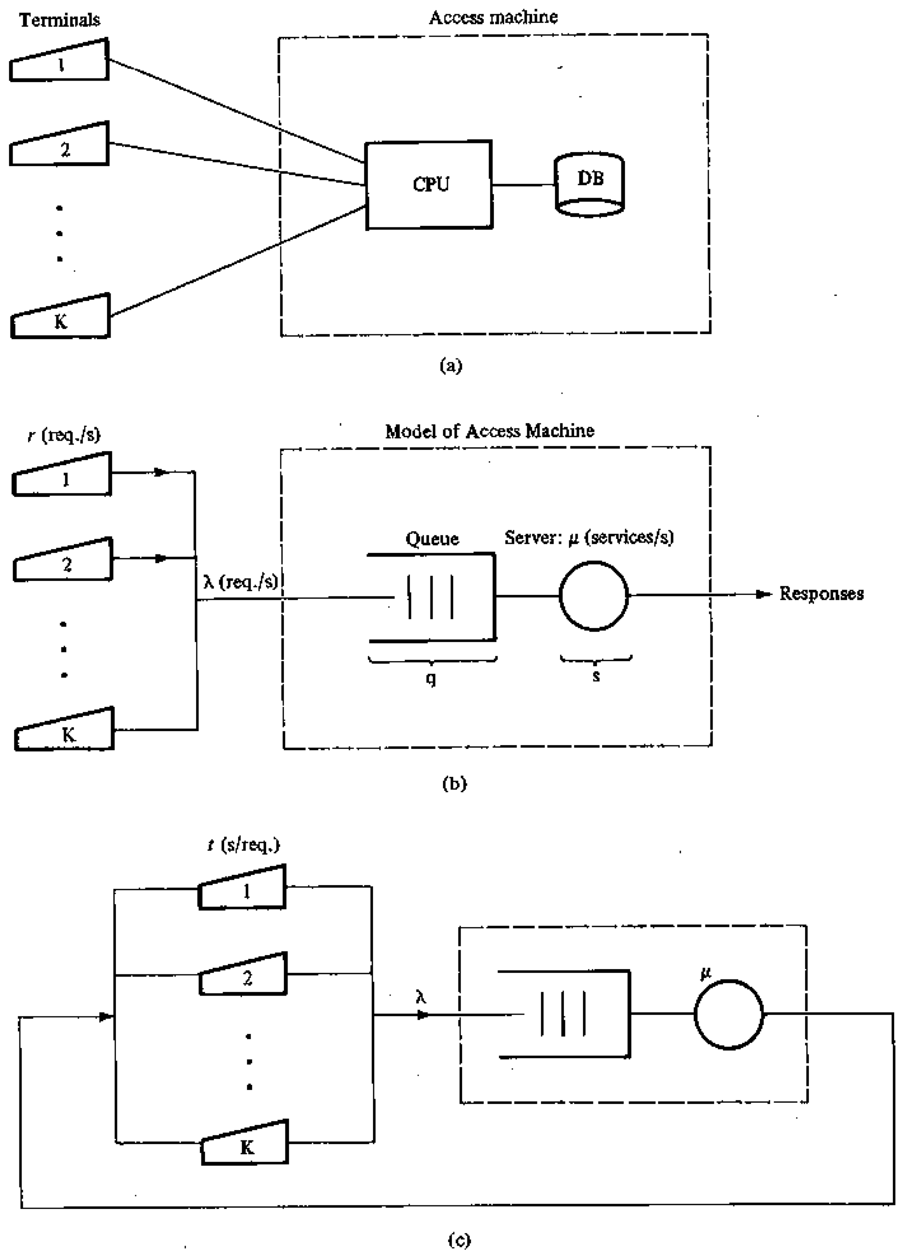


Figure 15.1 (a) Simple interactive videotex network; (b) open model; (c) closed model.

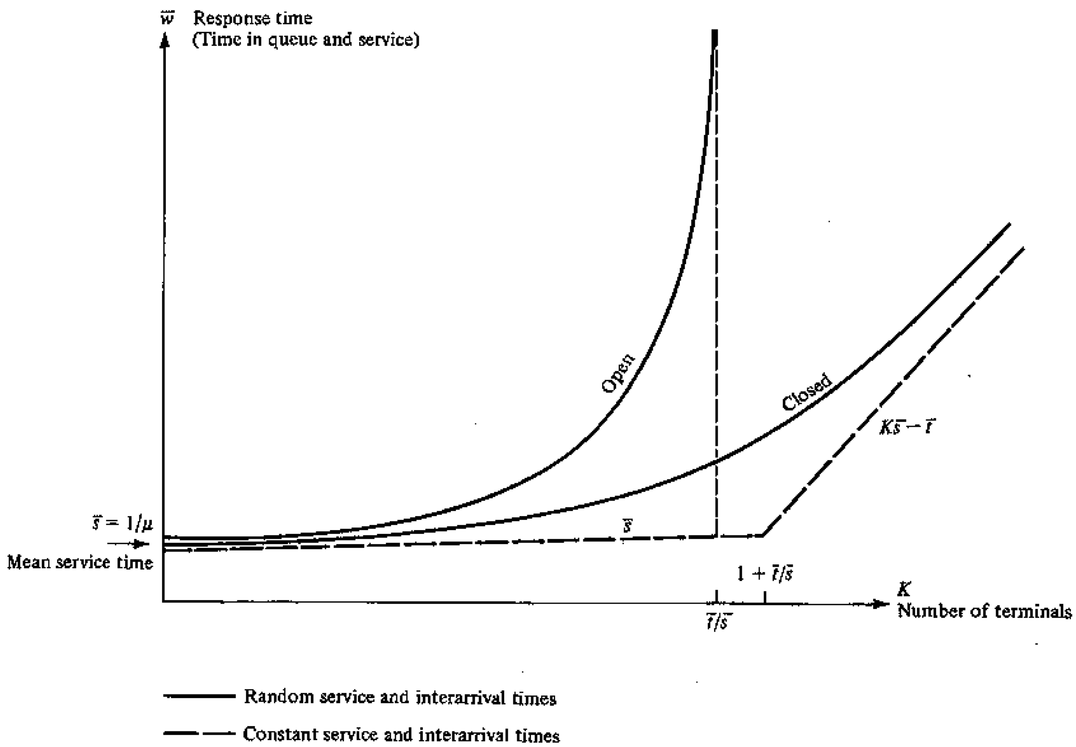


Figure 15.2 Comparison of open and closed models.

the "single-server machine repair model," terminals must wait for a response before generating new requests. Each token can be in one of three states: waiting in the queue, being serviced, or in the terminal. Thus, terminals are an integral part of the model (any source of requests would do in the open model!); new requests cannot be generated before a response arrives (they are generated $\bar{t}$ seconds, on the average, after the response); there is no queuing in the terminals, which can be collectively considered simply as a delay station; and λ (the throughput) is not constant, but has to be calculated along with $\bar{w}$.

In a loose analogy, open models can be likened to a nonlinear electric circuit fed from a source of current (requests). If the impedance (service time) of the circuit becomes too large, the voltage drop (system population) can grow without limit. Closed models behave analogously to a circuit fed from a source of voltage: the current (λ) may go to infinity if the impedance $(\bar{t}+\bar{s})$ approaches zero.

The behavior of the two models is summarized in Figure 15.2. While adding terminals leads to infinite $\bar{w}$ in the open system, it results only in a linear increase of $\bar{w}$ in the closed system. The exact shape of the curves depends on the probabilistic properties of interarrival, thinking, and service times, and on the queuing discipline. The dashed lines show $\bar{w}$ in the case when the above three times are constant (with zero variance). The same lines indicate the asymptotic behavior of the two systems for very small and very large values of K (compared to $\bar{t}/\bar{s}$). $K = 1 + \bar{t}/\bar{s}$ is the maximum number of terminals our closed system could accommodate under constant t and s, while keeping the response time equal to s. Therefore, $1 + \bar{t}/\bar{s}$ is called the "saturation" point of the system. Adding more terminals will result (again, for constant t and s) in increased $\bar{w}$ (by s for each new terminal).

We emphasize that the models in Figure 15.1 are mere examples of the two large classes of queuing models, and are intended to illustrate their basic behavior. Many sophisticated variations of both types have been described in the literature, each suited for a different purpose—e.g., the "central server model of multiprogramming", or the "straightforward model" (see Allen, 1978). In fact, methods exist for the resolution of arbitrary networks of M/M/c type queues (multiple servers with exponential service time and Poisson arrivals; see below).

Lacking extensive experimental data, we cannot answer with certainty the question: Are videotex networks closed or open? The characteristic of open systems that the request rate be independent of the response time is approximately valid in situations in which:

- users can re-issue their requests before the answer is received
- the user population is rapidly changing, and new customers issue only single requests
- the users' think time is much larger than the response time.

On the other hand, closed systems are the appropriate model when:

- new requests cannot be issued before the response has arrived
- think time can only start when a new page is received and displayed
- longer sessions are typical.

All of the above situations apply to videotex to a certain degree. Thus, all one can say about the above question is that real systems are a mixture of the two extreme cases (open and closed); appropriate queuing models can be created, e.g., by introducing a flow of new customers into (and out of) a closed system.

Sample Calculations

Now we give a few examples of typical queuing calculations. We limit ourselves to the M/M/c case because they are the simplest from a computational point of view. First, we give a summary of the usual notation and a few formulas sufficient for some useful calculations:

- q, time spent by a request waiting in the queue
- s, service time

- $w = s + q$, time in queue and service (response time)
- τ, interarrival time (between two requests)
- t, think time (delay in terminals in closed systems)
- N_q, number of requests in queue
- N_s, number of requests being served
- $N = N_q + N_s$, number of requests in queue or being served.

The above eight symbols represent random variables; their average values are $\bar{q}, \bar{s}, \bar{w}, \bar{\tau}$, $, \bar{N}_q, \bar{N}_s, \bar{N}$.

- $\tau = 1/\bar{\tau}$, arrival rate to the queue
- r, request rate from a terminal
- $\mu = 1/\bar{s}$, service rate per server
- c, number of servers in multiserver queues
- $\rho = \lambda/(\mu\, c)$, server utilization
- $u = \lambda/\mu$, traffic density
- $p(\mathrm{E})$, probability of event E.

Formulas for M/M/c queuing systems with first come-first served queuing discipline:

$$p(\tau \leqslant x) = 1 - e^{-\lambda x} \qquad \text{(Poisson arrival process)}$$

$$p(s \leqslant x) = 1 - e^{-\mu x} \qquad \text{(exponential service times)}$$

$$C(c,u) = \frac{\dfrac{u^c}{c!}}{\dfrac{u^c}{c!} + (1-\rho)\displaystyle\sum_{n=0}^{c-1} \frac{u^n}{n!}} \qquad \text{(probability that all } c \text{ servers are busy).}$$

$$C(1,u) = \rho$$

$$\bar{q} = \frac{C(c,u)\,\bar{s}}{c(1-\rho)}$$

$$\bar{w} = \bar{q} + \bar{s} \qquad (= \bar{s}/(1-\rho) \text{ for } c = 1)$$

$\pi_w(90) = \bar{w} \ln 10$ (90th percentile time in the system, i.e., in 90% of all cases,

$w \leqslant \pi_w(90)$)

$$\bar{N} = \bar{w}\lambda .$$

The last equation is a consequence of Little's law, which states that the relationship between average values of the number of customers n inside a service facility, time l a customer spends in the facility and customer flow f (customers/second) through the facility, is

$$n = l\, f.$$

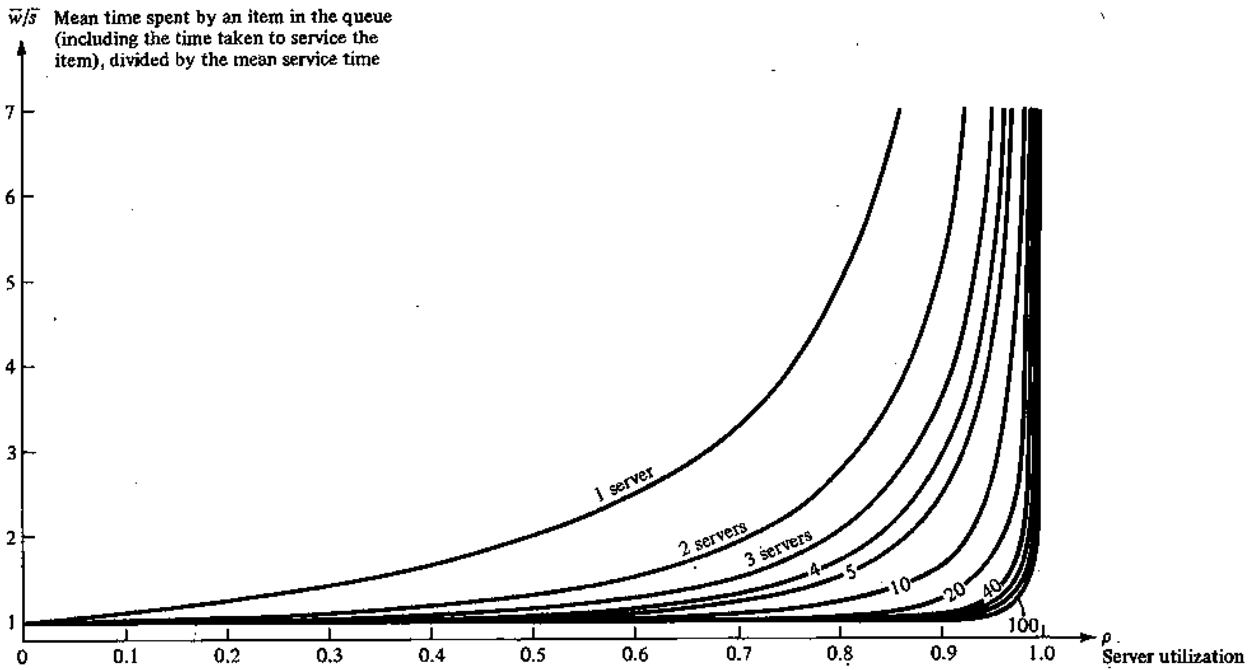


Figure 15.3 Queuing times for multiserver queues, exponential interarrival and service times.

Little's formula is valid for any distribution of n, l, and f.

Figure 15.3 is a graphical representation of the equations for $\bar{w}$, advantageous especially for the evaluation of multiserver queues involving the formula for $C(c,u)$. The curve for $c = 1$ (single server queue) is essentially the same as the one for open system response time in Figure 15.2.

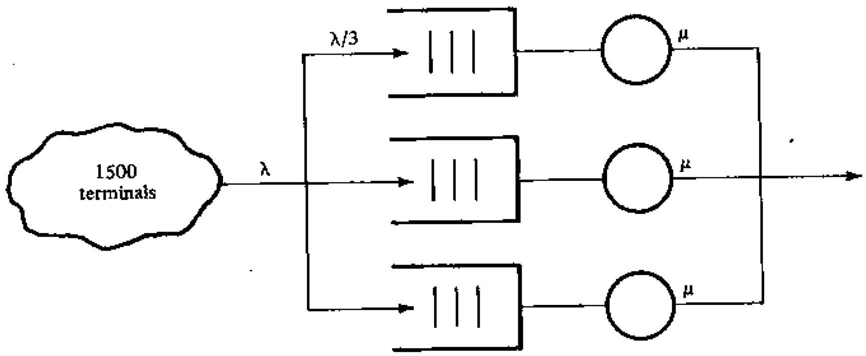
Example 1 (single server)

For an open type videotex system (see Figure 15.1(b)), with $K = 500$ terminals, $r = 1/30$ (request/second/terminal) and average service time $\bar{s} = 50$ ms, calculate the response time $\bar{w}$, ρ, $\bar{N}$, and π_w (90).

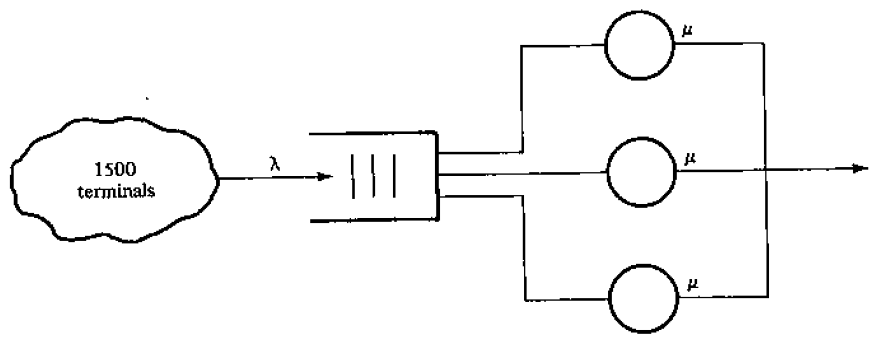
Solution: First we determine λ as $\lambda = Kr = 16.67$ requests/second. From this $\rho = \lambda \bar{s} = 0.833$. It follows that $\bar{w} = \bar{s}/(1\text{-}\rho) = 300$ ms. $\pi_w(90) = 300 \times 2.3 = 690$ ms. This means that the response time will be less than 690 ms in 90% of all cases. Note that a 10% increase of K (or r) would result in approximately doubling $\bar{w}$ to 600 msecond!

Example 2 (multiserver queue versus multiple single-server queues)

The operator of the previously analyzed network is so successful that he wishes to extend his system to 1500 terminals. To sustain the resulting request rate of 50 requests/second he decides to triple the database; he wants to compare the response times obtainable with separate queues to the three databases and a common multiserver queue with $c = 3$ (see Figure 15.4).



(a)



(b)

Figure 15.4 (a) Model with three separate single server queues. Requests are split in three streams coming from groups of 500 terminals; (b) Multiserver queue. A request is taken by whichever server becomes first free.

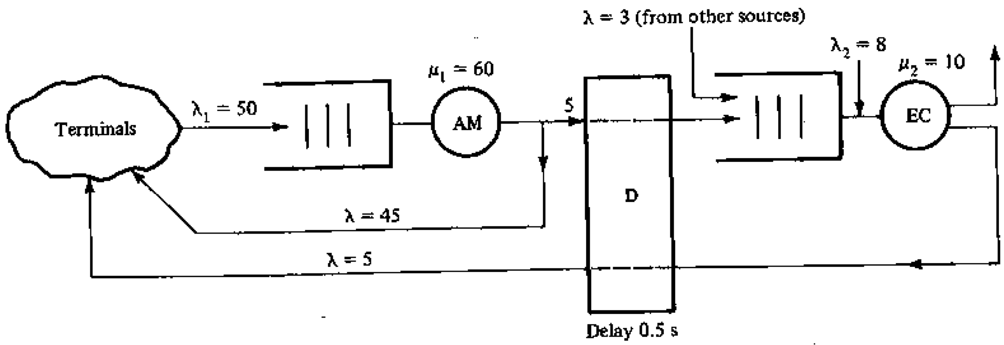
Solution: The response time of the system with three queues is, as calculated previously, $\bar{w} = 300$ ms. For the multiserver queue, we find $\rho = \lambda/\bar{w}c = 0.833$. From Figure 15.3 we have $\bar{w}/\bar{s} = 2.5$; therefore $\bar{w} = 2.5 \times 50$ ms $= 250$ ms.

Example 3 (multistage open system)

Calculate the response time of a videotex network in which 90% of all requests are satisfied from the access machine and 10% are transmitted (after local service) through a data network D to an external computer. The average delay thorough D is 0.5 second in each direction. The system and its parameters are shown in Figure 15.5.

Solution: $\rho_1 = 50/60 = 0.833$; $\bar{w}_1 = 100$ ms. $\lambda_2 = 0.1 \times 50 + 3 = 8$; $\rho_2 = 8/10 = 0.8$; $\bar{w}_2 = 500$ ms. Thus the response time in 90% of cases is 100 ms, and $0.5 + 0.5 + 0.5 = 1.5$ seconds in the remaining cases. This yields for the response time $0.9 \times 0.1 + 0.1 \times 1.5 = 0.24$ second.

Note that in this example we did not account for the communication delay between the terminals and the access machine. The calculation is exact providing only that the arrival



AM: Access machine
EC: External computer
D: Data network

Figure 15.5 Multistage open system.

pattern in the second queue is Poisson; this many not be quite true—e.g., because of the influence of branching at the output of the first server. However it is the most reasonable approximation we can make unless further assumptions are made about the branching pattern, the distribution of delays in D, and the arrivals to EC from other sources.

Example 4 (closed system)

Find $\bar{w}$, ρ, and λ for the system from Example 1 considered as a closed single-server system (see Figure 15.1(c)). The think time $\bar{t} = 1/r$.

Solution: Although exact methods exist for this case, we prefer an approximate solution based on the fact that for large K the arrival pattern generated by K terminals at the input of the queue approaches a Poisson pattern. Therefore the server can be considered as an M/M/1 system and the simple equations listed above can be applied.

The system can be seen as composed of two "stations," both traversed by a flow of (unknown) intensity λ. The station represented by the queue and server will have a population of $\bar{N}_1 = \bar{w}\lambda$. The other station represented by the K terminals will have an average population of $\bar{N}_2 = \lambda$ (from Little's formula). We must find λ such that $\bar{N}_1 + \bar{N}_2 = K$. Thus,

$$\lambda = \frac{K}{\bar{w} + \bar{t}}$$

which leads to

$$\lambda^2 \bar{s}\,\bar{t} - \lambda(\bar{s} + \bar{t} + \bar{s}K) - K = 0\,.$$

The feasible solution of this equation is

$$\lambda = 16.509 \text{ requests/second}$$

yielding $\bar{N}_1 = 4.73$; $\bar{N}_2 = 495.27$; and $\rho = 0.825$.

Note that the system operates before the "saturation point" of Figure 15.2, corresponding in this case to $1 + \bar{t}/\bar{s} = 601$ terminals.

A final note on the use of simplifying assumptions (such as exponential distributions) and approximative solutions seems appropriate. First, such calculations are a useful first-cut; if the results seem interesting, more elaborate models or other methods (such as simulation) can follow. Second, it is all too often the case that the real parameters of a system are uncertain, intractable, or both (e.g., probability distributions and traffic intensities). Therefore the quality of the results cannot be improved merely by applying more complex and exact methods without improving the quality of input data. Let us quote the Second Principle of Nonsense, attributed to R. A. Rosanoff: "Rigorous argument from inapplicable assumptions produces the world's most durable nonsense." (For completeness, the First and Third Principles are: "For every durable item of nonsense, there exists an irrelevant frame of reference in which the item is sensible," and "The roots of most nonsense are found in the fact that people are more specialized than problems.")

15.2 ACCESS MODEL FOR TREE-STRUCTURED DATABASES

In the examples of the preceding section we considered essentially single-server models of videotex systems. In spite of their simplicity, it can be argued that they reflect reality well because the prevailing part of the service time is the access time to page memory (usually disk). In particular, even if the model included two servers (CPU and I/O), the CPU service time (a few mseconds) would typically be an order of magnitude smaller than the I/O time (around 50 mseconds for 1−2 accesses to a single disk pack). Therefore, the main bottleneck limiting the throughput occurs at the I/O stage, while there is negligible queuing for the CPU. Fortunately, several methods exist to partly alleviate this problem. For example, data can be spread out or replicated over several disk drives, or some frequently used pages may be duplicated even in the same disk, thus reducing the average seek time. Another technique is to introduce a memory hierarchy and to keep highly accessed pages in faster memory, such as various electronic equivalents to disks or extended main memory. Note that combined transmission of data over teletext and viewdata (e.g., in INDAX, Chapter 6) is based on a similar idea: the database can be significantly offloaded if popular pages are transmitted in broadcast mode.

All such techniques are based on the "locality of reference" property of practically all data: most requests against a database (or file or program) are directed to a relatively small subset of the data, called the *working set*. This working set may be static or may change with time; automatic methods exist to detect and maintain proper working sets in faster levels of memory hierarchies, thereby minimizing the average access time. This behavior of data has found many expressions—e.g., in the form of rules of thumb, such as the "80−20 rule": 80% of all requests go to 20% of data. Without locality, that is, if requests were scattered randomly, 20% of data would receive 20 and not 80 percent of requests. Another common access rule is Zipf's law (see Zipf, 1949). It predicts that the frequency of access to the kth most popular page should be proportional to $1/k$. Equivalently, the cumulative probabilities of access to the k most frequently accessed pages should be a linear function of log (k).

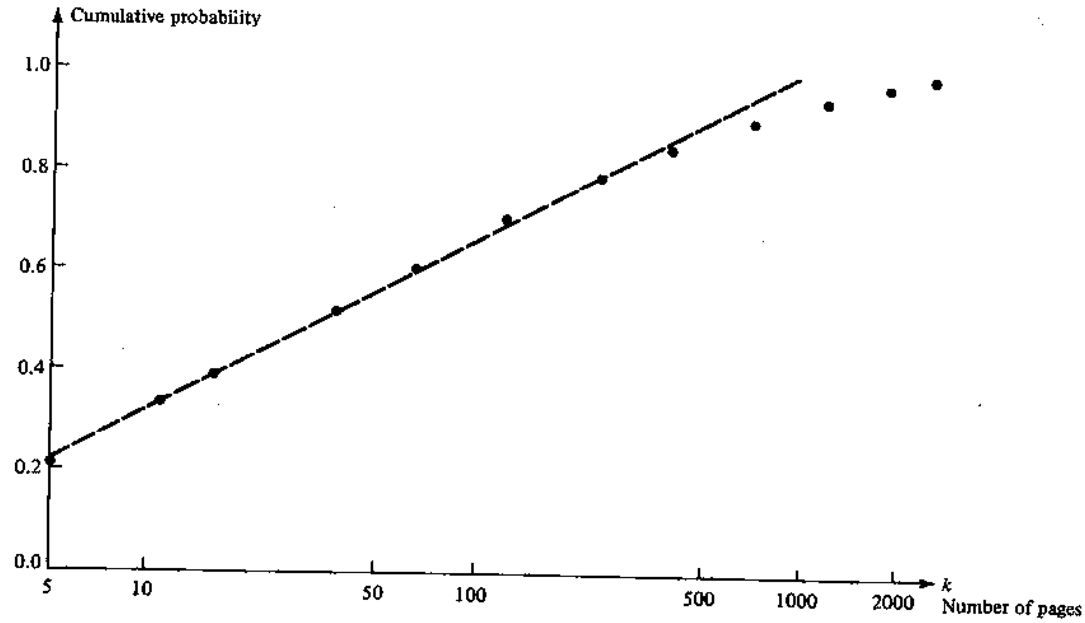


Figure 15.6 Fitting Zipf's formula to experimental data.

To check the validity of this latter assumption for videotex, experimental data from a small videotex trial in British Columbia are plotted in Figure 15.6 (taken from a report by Horspool and Bochmann, 1982). The graph is indeed linear until a value of about 400 pages.

To summarize: optimization of mass storage is of eminent importance to videotex; however, little can be done unless reliable data are available on page-accessing frequencies. The remainder of this section deals with an access model that permits calculation of such frequencies from the behavior of individual users.

Markov Chain Model of Page Access

Standard approaches to determining access probabilities typically rely on measurement as the primary source of data for database optimization. Although probably more exact, measurement data provide relatively little explanation of the reasons and mechanisms underlying the observed phenomena, and therefore one has to be very careful in extrapolating such data for different databases, users, and applications.

In contrast, the method presented below takes into account a description of elementary actions in the user's behavior and attempts to deduce his access pattern from that (see Madeleine et al., 1982). Therefore, once the user behavior is well understood, similar methods will lead to performance figures that not only describe, but also theoretically explain, these figures.

Assume a standard hierarchical videotex database in the form of a balanced tree such as in Figure 15.7. To describe the behavior of a user navigating in such a database,

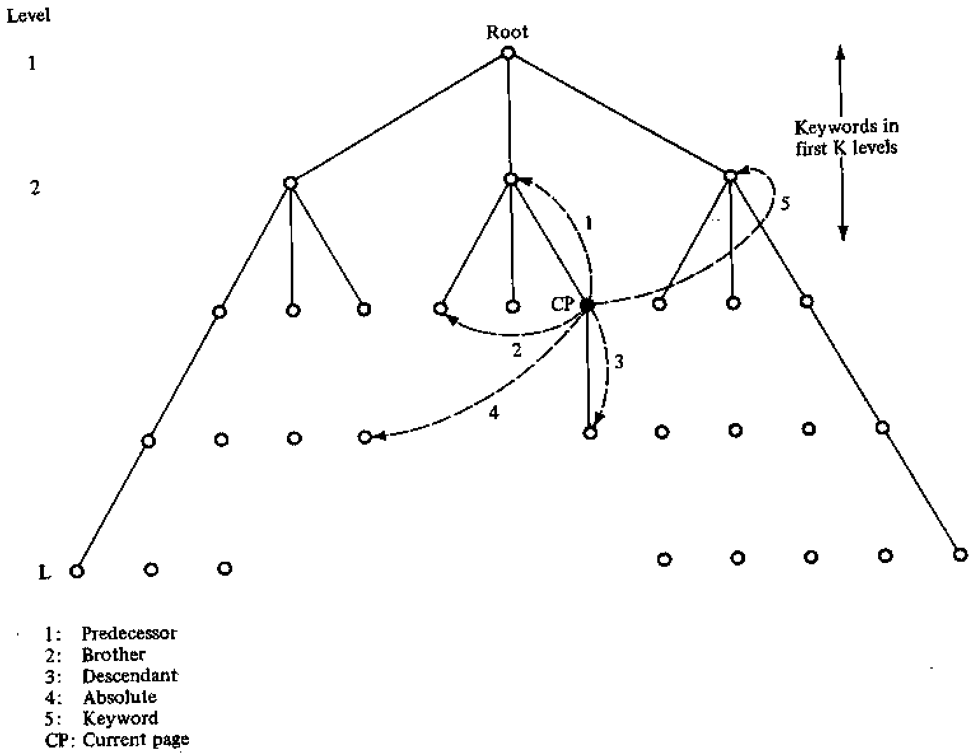


Figure 15.7 Access modes in hierarchical database.

we postulate the following five *access modes*, from which he can select any time he accesses a new page (CP is the currently accessed page):

1. CP's predecessor
2. CP's brothers (choosing one with equal probability)
3. CP's descendants (choosing one with equal probability)
4. absolute page selection (choosing any page in the database with equal probability)
5. keyword selection (same as 4, but only to pages in the top K levels in the tree, which are supposed to be addressable by keywords).

Further, let a, b, d, t, w, where $a + b + d + t + w = 1$, be the stationary probabilities of selection at every step one from the above modes, respectively. (The probabilities are assumed to be the same for all users.) Two more rules are necessary: when CP is the root (level 1), modes 1 and 2 will result in re-accessing the root; when CP is a leaf page, mode 3 will result in going to the root.

Such a search pattern can be modeled by a Markov chain (see Allen, 1978) with one state per page, where the transition probabilities follow directly from the above probabilities. But to find the steady-state probabilities (page access frequencies) one would have to solve a system of, say, 100,000 equations of 100,000 variables for a database of reasonable size.

To be practical, the number of equations can be drastically reduced by lumping together all states (pages) belonging to a given level of the tree; this results in systems of tractable size that still yield interesting results. (Without going into theoretical details we assert that the conditions for lumping together the states are satisfied in this case). The new states are labelled according to the tree levels to which they correspond.

In order to derive the transition probabilities of the new system (see Figure 15.8) we introduce a few new symbols:

- L, number of levels in the tree
- K, number of levels addressable by keywords
- M, branching factor (number of descendants of non-leaf pages)

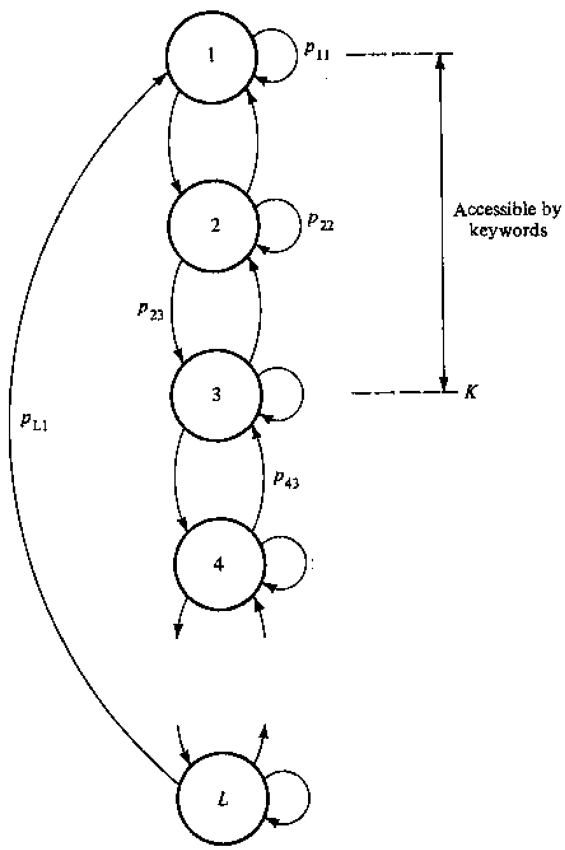


Figure 15.8 Reduced Markov chain with one state per tree level. Transition probabilities due to modes 4 and 5 are not shown.

- $N = (M^L - 1)/(M - 1)$, total number of pages
- $R = (M^K - 1)/(M - 1)$, number of pages in the upper K levels
- $C_i = M^{i-1}$, number of pages in level i.

Now, the transition probabilities p_{ij} (from state i to state j) are obtained as follows:

$$\begin{aligned}
p_{11} &= a + b + U \\
p_{ii} &= b + U && 1 < i \leq L \\
p_{i\,i+1} &= d + U && 1 \leq i < L \\
p_{L1} &= d + U \\
p_{i+1\,i} &= a + U && 1 \leq i < L \\
p_{ij} &= U && \text{for all pairs } ij \text{ not included above.}
\end{aligned}$$

U is the transition probability between non-adjacent levels:

$$\begin{aligned}
p_{ij} = U &= tC_j/N + wC_j/R && j \leq K \\
&= tC_j/N && j > K.
\end{aligned}$$

Steady-state probabilities $\pi_1, \ldots, \pi_L$ are obtained by solving the equations

$$\begin{aligned}
\pi_k &= \sum_{i=1}^{L} \pi_i p_{ik}, \qquad k = 1, \ldots, L \\
1 &= \sum_{i=1}^{L} \pi_i
\end{aligned}$$

A sample of typical results for $M = 10$, $L = 10$, $K = 3$ is plotted in Figure 15.9. Note that the case with $d = 1$ exactly corresponds to Zipf's law, while $t = 1$ leads to equiprobable access to every page. The other cases illustrate the sensitivity of the results as the parameters of user behavior change. The peaking effect at the third level is clearly due to accessing by keywords.

The above basic calculation can be refined in several ways. For example, one can obtain the access probability p_s to any subtree of the database as

$$p_s = \sum_{i=1}^{L} \pi_i c_i$$

where c_i is the proportion of level i covered by the subtree. For the subtree in Figure 15.10(a), $c_1 = c_2 = 0$; $c_3 = c_4 = c_5 = 0.25$.

So far we have considered equiprobable access to all pages within each access mode. This restriction can be partly removed at the expense of an increased number of states in the Markov chain. For example, should experimental data indicate that users prefer the right subtree of the database ($d_r > d_\ell$ in access mode 3), the states can be lumped together as in Figure 15.10(b). Transition probabilities within the levels of each subtree would be calculated as before: inter-subtree access is possible only in absolute or

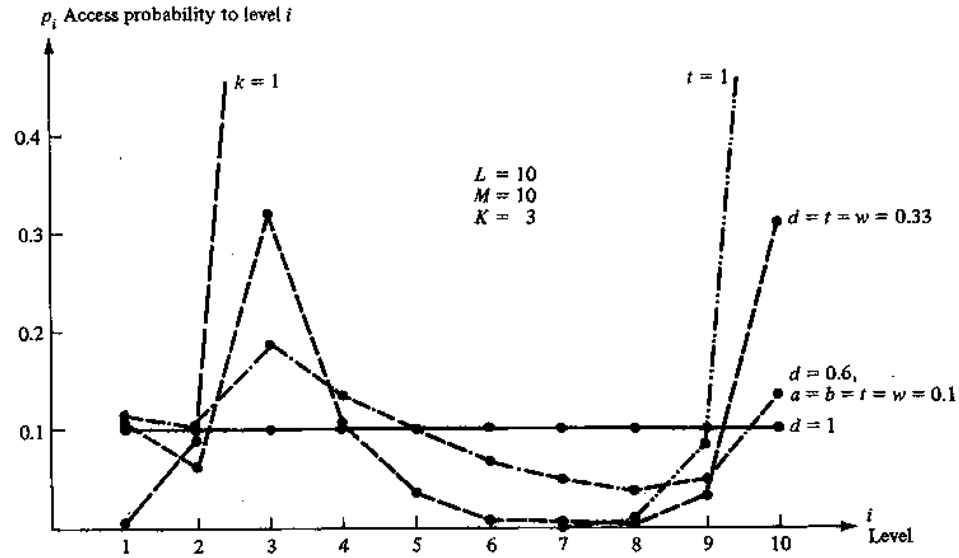


Figure 15.9 Access probabilities in a 10-level tree modeled by a Markov chain.

keyword mode (with one exception), and the probabilities would be obtained similarly as in calculating U, above. Thus, for example, in Figure 15.10(b), supposing $K = 2$,

$$p_{67} = p_{76} = tC_7/N = t(4/15)$$

$$p_{52} = tC_2/N + wC_2/R = t(1/15) + w(1/3).$$

The exception mentioned above occurs when accessing in mode 2 on the level at which the split occurs (level 2 in this case); here, trivially $p_{23} = p_{32} = b + U$.

The above situation (different access probabilities to descendants) may of course occur repeatedly in the tree. If all pairs of probabilities were different, we would end up with one state per page. Modeling other cases, such as unbalanced trees (see Figure 15.10(c)) or different values of a, d, etc., between different pairs of levels, is equally feasible with the above method.

15.3 OPTIMIZATION OF TELETEXT TRANSMISSION SEQUENCES

In teletext, as with viewdata, pages are not accessed with equal frequency by the user community. In most systems, however, pages are customarily transmitted in fixed cycles, with occasional repetition of evidently important pages (e.g., directories). One of the reasons for this lack of optimization is that access frequencies cannot be centrally monitored, as they are in interactive systems. But as teletext will offer more sophisticated services in the future, it will become increasingly important to minimize the access times as much as possible.

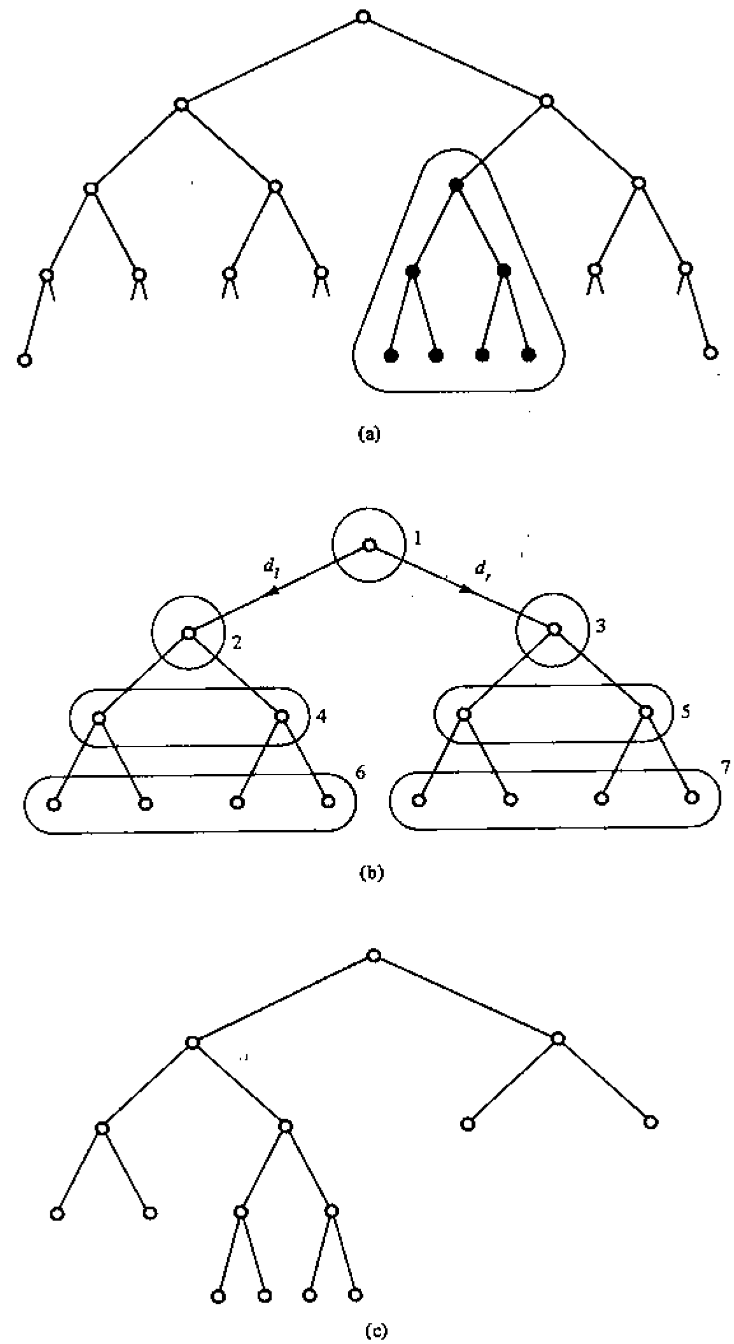


Figure 15.10 (a) Calculating subtree access probability; (b) Multiple states per level, $d_r \neq d_\ell$; (c) Unbalanced tree.

This section addresses itself to the problem of minimizing the average waiting time for a teletext page, under the assumption that we know the (fixed) probabilities of request of each page by a user. These probabilities can be obtained, e.g., by the method in the preceding section. The conclusions will be equally valid for a group of users requesting pages with the same probabilities.

Suppose the teletext magazine to be broadcast has n distinct pages. They are transmitted in a fixed cycle at regular intervals, one page at a time. The user issues a request at an arbitrary time (relative to the cycle) and then waits for w page intervals until he receives the selected page. The process is then repeated after a random delay (think time), which is long compared to the cycle time. (The purpose of this assumption is simply that every new request be issued at a random time relative to the cycle.) Let a_i, $i = 1, \ldots, n$, $\sum_{i=1}^{n} a_i = 1$, be the fixed probabilities used to generate page requests. The problem is then to determine the repetition rates of pages in the broadcast sequence that will minimize the average waiting time $\bar{w}$. Imagine, for instance, that $n = 9$ pages are broadcast. If all of them are requested with equal probability, then clearly the best one can do is to transmit them in a fixed cycle of length 9, each page sent once:

1 2 3 4 5 6 7 8 9 1 2 3 4, etc.

This results in $\bar{w} = n/2$. Now suppose that page 1 is requested four times as frequently as all other pages ($a_1 = 4a_2 = \ldots = 4a_9$). Clearly this page should be transmitted more frequently than the others; intuitively one might think that it should be repeated four times in the cycle (of length 11):

1 2 3 1 4 5 1 6 7 1 8 9 1 2 3 1 4 5, etc.

However, we will see that this is not the case: to minimize $\bar{w}$, it should be repeated only twice:

1 2 3 4 5 1 6 7 8 9 1 2 3 4 5 1 6 7, etc.

With arbitrary a_i the problem becomes a combinatorial one (to construct an optimal sequence of pages that might have a very long cycle or that might even by acyclic). To avoid this complication we will no longer suppose that pages are transmitted in a fixed repetitive cycle, but that they are generated randomly with stationary probabilities p_i, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$. It follows from elementary probability that the average distance between two occurrences of page i in such a sequence is $1/p_i$. Therefore our problem now becomes to find the set of p_i that will minimize $\bar{w} = \sum_{i=1}^{n} a_i/p_i$ for given a_i.

First we solve for $n = 2$ and then generalize for $n > 2$. We have

$$a_1 + a_2 = 1$$

$$p_1 + p_2 = 1.$$

Letting

$$a_1 = ra_2 \qquad \text{and} \qquad p_1 = xp_2$$

gives

$$a_1 = \frac{r}{1+r}, \qquad a_2 = \frac{1}{1+r}$$

and

$$p_1 = \frac{x}{1+x}, \qquad p_2 = \frac{1}{1+x}.$$

The expression to minimize now becomes (for given r)

$$\bar{w} = \frac{r/(1+r)}{x/(1+x)} + \frac{1/(1+r)}{1/(1+x)} = \frac{1+x}{1+r} + \frac{r}{x+rx} + \frac{r}{1+r}.$$

Letting

$$\frac{\partial \bar{w}}{\partial x} = \frac{1}{1+r} - \frac{r}{1+r} \cdot \frac{1}{x^2} = 0$$

yields finally

$$x = \sqrt{r}.$$

To generalize this result, suppose now that all a_i in

$$\bar{w} = \sum_{i=1}^{n} a_i / p_i, \qquad n > 2$$

already correspond to $\bar{w}_{min}$, except p_1 and p_2, which have to be found. Thus we have to minimize

$$u = \frac{a_1}{p_1} + \frac{a_2}{p_2},$$

where

$$a_1 + a_2 = K \leqslant 1$$

and

$$p_1 + p_2 = L \leqslant 1.$$

Letting again

$$a_1 = ra_2 \qquad \text{and} \qquad p_1 = xp_2$$

we obtain

$$p_1 = \frac{Lx}{1+x}, \qquad p_2 = \frac{L}{1+x}, \qquad a_1 = \frac{Kr}{1+r}, \qquad a_2 = \frac{K}{1+r}.$$

After doing the arithmetic as above we have

$$\frac{\partial u}{\partial x} = \frac{K}{L}\left[\frac{1}{1+r} - \frac{r}{1+r}\cdot\frac{1}{x^2}\right]$$

and therefore again $x = \sqrt{r}$ as the value minimizing $\overline{w}$. By applying the same reasoning to all pairs of p_i, p_j; we conclude that $\overline{w}$ is minimal when

$$p_1 : p_2 \ldots : p_n = \sqrt{a_1} : \sqrt{a_2} \ldots : \sqrt{a_n}$$

where a_i are the given request probabilities and p_i the optimal transmission probabilities.

Here is a final note on the practical utilization of the above result: should the person who implements the teletext system insist on transmitting a fixed cycle of reasonable length, then the page repetition rate can be made to conform in general only approximately with the theoretical values. Given the dubious precision of any data on a_i that he can conceivably obtain (by polls, etc.) and the negligible suboptimality of the solution, this would be a reasonable approach. On the other hand, abandoning the cycle and generating the sequence using a random number generator seems to be a simple and natural solution, too. However, some provision should be made in this case to limit the maximum interval between two repetitions of a page, which may occasionally become very large.

References

The following abbreviations are used for conferences:

Videotex 'XX—Proceedings of the Videotex 'XX Conference, Online Conferences Limited, London, 19XX.

Viewdata 'XX—Proceedings of the Viewdata 'XX Conference, Online Conferences Limited, London, 19XX.

ALLEN, A. O. *Probability, Statistics and Queuing Theory*, New York: Academic Press, 1978.

ALLORA-ABBONDI, G. "Transmission System Evaluation for Two-Way Cable," *IEEE Trans. of Cable Television*, Vol. CATV-4, July, 1979, pp. 111–118.

ARCHITEL. "Protocole de transport Architel." Note technique NT/PAA/DIR/746, NT/CNR/DIR/6, CNET (France), 1982.

ARCHITEL. "Protocole de session Architel." Note technique NT/PAA/DIR/747, NT/CNR/DIR/7, CNET (France), 1982.

ARCHITEL. "Protocole de présentation Architel." Note technique NT/PAA/DIR/748, NT/CNR/DIR/8, CNET (France), 1982.

ARNOLD, E. "Prestel—a Lemon in the Living Room?" Viewdata '81, p. 537.

BALL, A. "Copyright Squabbles Top Europe's Cable Problems." *Cable Age*, Nov. 30, 1981, pp. 22–34.

BALL, A. J. S., AND J. GECSEI. "User Interface for Future Videotex Systems." Publication 412, Dept. IRO, Université de Montréal, May, 1981.

BALL, A. J. S., G. V. BOCHMANN, AND J. GECSEI. "Videotex Networks." *Computer*, Dec., 1980, pp. 8–14.

BALL, E., AND P. HAYES. "A Test-Bed for User Interface Designs." Proc. Conf. on Human Factors in Computer Systems, Gaithersburg, Md., March 15–17, 1982, pp. 85–88.

BEAKHUST, D. J., AND M. C. GANDER. "Teletext and Viewdata—a Comprehensive Component Solution." *Proc. IEE*, Vol. 126, No. 12, Dec., 1979, pp. 1374–1396.

BELL SYSTEM. "Session Level Protocol" (provisional), AT&T, Parsippany, N.J., Nov., 1981.

BERGER, M., AND Y. NOIREL. "Diode: A New Tool for Data Retrieval." *Videotex '81*, pp. 51–56.

BLINEAU, J., ET AL. "Didon: Diffusion de données par paquets—etat de la technique." CCETT, Rennes, France, Doc. No. 3.80, Dec., 1980.

BOCHMANN, G. V., J. GECSEI, and E. LIN. "Keyword Access in Telidon; an Experiment." *Videotex '82*. pp. 321–332.

BOSMAN, N. J. "A Systems Approach to Interactive Videotex." Int. Conf. on New Systems and Services in Telecommunications, Liége, Belgium, Nov., 1980, pp. D.1.1–D.1.5.

BOURASSIN, L., J. SOTO, AND M. STEIN. "Architecture d'un terminal autonome compatible Télétel/Prestel." International Symposium on Graphics and Text Communication, Paris, Nov., 1981, pp. 18–22.

BOWN, H. G., ET AL. "Comparative Terminal Realizations with Alpha-Geometric Coding." IEEE Chicago Spring Conference on Consumer Electronics, June, 1980.

BROWN, M. "Viewdata Telesoftware Distribution." *Viewdata '81*, Oct., 1981.

BS-14. "Broadcast Specification, Television Broadcast Videotex." Dept. of Communications, Government of Canada, June, 1981.

BUTLER COX & PARTNERS. "Videotex Report Series." Butler Cox & Partners, London, 1982.

CARTERTON, J. "De Télétel à Videopac: une stratégie face à l'émergence du Videotex." International Symposium on Graphics and Text Communication, Paris, Nov., 1981, pp. 178–182.

CCIR Report AD/11(MOD F). "Characteristics of Teletext Systems." CCIR, Oct., 1981, pp. 62–76.

CCITT Recommendation F.300. "Videotex Service." CCITT, Geneva, 1980.

CCITT Recommendation S.100. "International Information Exchange for Interactive Videotex." CCITT, Geneva, 1980.

CEPT. "Current Status of Harmonisation for Videotex Display Aspects and Transmission Coding for 26 Countries in Europe." CCITT, Geneva, Oct., 1981.

CHAMBERS, J. P. "Enhanced U.K. Teletext Moves Toward Still Pictures." IEEE, Trans. on Consumer Electronics, Vol. CE-26, Aug., 1980.

CHANG, H. "Fiber Optic Broadband Integrated Distribution—Elie and Beyond." *IEEE Trans. on Communications*, Special Issue, April, 1982.

CHAUFFAULT, A., ET AL. "Accàs Videotex à une banque de données médicales." Publ. No. 165, IRISA, Rennes, France, May, 1982.

CHILDS, G. H. L. "Videotex Standardisation in Europe." Videotex '81, pp. 401–415.

CHISHOLM, I. G. "Prestel Gateway—the Commercial Aspects." Prestel Gateway to Third Party Databases Conference, London, March, 1981.

CHITNIS, A., AND B. PAPP. "The Videotex Information Provider System." *Canadian Electronics Engineering*, June, 1982, pp. 16–21.

CICIORA, W. S. "Virtext & Virdata, a Present U.S. Teletext Application." *Videotex '81a*, pp. 77–84.

CICIORA, W. S. "Twenty-four rows of Videotex in 525 Scan Lines." *IEEE Trans. on Consumer Electronics*, Vol. CE-27, Nov., 1981b, pp. 575–587.

CICIORA, W. S. "The Role of the Television Receiver Manufacturer in the United States." *Viewdata '80*, pp. 533–546.

CICIORA, W. S., ET AL. "An Introduction to Teletext and Viewdata with Comments on Compatibility." *IEEE Trans. on Consumer Electronics*, Vol. CE-25, July, 1979, pp. 235–245.

CLARKE, K. E. "The Application of Picture Coding Techniques to Viewdata." *IEEE Trans. on Consumer Electronics*, Vol. CE-26, Aug., 1980, pp. 568–577.

CORE. "Status Report of the Graphic Standards Planning Committee." SIGGRAPH-ACM, Vol. 13, No. 3, Aug., 1979.

COYNE, J. J. "An Integrated Program Distribution System for the 80's," *Proceedings, National Electronic Conference*, 1981, Chicago.

CROLL, M. G. "CEEFAX: Measurement Techniques." Report No. BBC RD 1977/6, BBC Research Department, 1977.

CROWTHER, G. O., AND D. S. HOBBS. "Teletext and Viewdata Systems and Their Possible Extension to the U.S.A." *Proc. IEE*, Vol. 126, Dec. 1979, pp. 1417–1424.

DATE, C. J. *An Introduction to Database Systems*. 3rd Edition, Reading, Mass.: Addison-Wesley, 1981.

DEUTSCHE BUNDESPOST. "Specification of a Basic Videotex Terminal Operating to the European Videotex Service," FTZ-T, Feb., 1982.

DORROS, I. "Challenge and Opportunity of the 1980s: the ISDN." *Telephony*, Jan. 26, 1981, pp. 43–45.

EDWARDS, G. C. "A National Videotex Intelligent Network and Its Role in the Information Industry." *Videotex '82*, pp. 149–158.

EMBLEY, D. W., AND G. NAGY. "Behavioral Aspects of Text Editors." *ACM Computing Surveys*, Vol. 13, No. 1, March, 1981, pp. 33–70.

FEINER, S., S. NAGY, AND A. VAN DAM. "An Integrated System for Creating and Presenting Complex Computer-Based Documents." *Computer Graphics*, Vol. 15, No. 3, Aug., 1981, pp. 181–189.

FINK, D. G. "Bridges to Television's Future." *IEEE Trans. on Consumer Electronics*, Vol. CE-27, Nov., 1981, pp. xii–xiii.

FITZGERALD, R. "Page Numbers and Channels in the BS-14 Broadcast Videotex Protocol." DOC Report, Ottawa, August, 1981.

GATES, G. W., AND G. S. TJADEN. "The INDAX Two-Way CATV Network for Videotex Services." *Videotex '82*, pp. 465–475.

GILBERT, D., AND I. TAYLOR. "The Bildschirmtext Gateway Facility—How does It Work?" Prestel Gateway to Third Party Databases Conference, London, March, 1981.

GODFREY, D. (ed.) *The Telidon Book*. Toronto: Press Porcépic, 1981.

GREEN, N. "Subtitling Using Teletext Service-Technical and Editorial Aspects." *Proc. IEE*, Vol. 126, Dec., 1979, pp. 1408–1411.

GRIEBLE, L. "Computer Networking in Bildschirmtext." *Videotex '82*, pp. 441–451.

GRIMM, P., JR. "The IBM Videotex Evaluation System." *Videotex '82*, pp. 493–500.

HABER, R. "Euronet and Videotex: Competitive or Complementary?" *Computer Communications*, Vol. 4, No. 3, June, 1981, pp. 99–105.

HALSEY, J. R., ET AL. "Public Data Networks: Their Evolution, Interfaces, and Status." *IBM Systems Journal*, Vol. 18, No. 2, 1979, pp. 223–243.

HANDLER, G. J. "Networking—Bit by Bit." *Videotex '82*, pp. 453–463.

HARASHIMA, S., AND M. KOBAYASHI. "Enhanced Functions and Hybrid Transmission in Captain." *Videotex '81*, pp. 79–93.

HARRIS, L. R. "User Oriented Data Base Query with the Robot Natural Language Query System." *International Journal of Man-Machine Studies*, No. 9, 1977, pp. 697–713.

HEDGER, J., M. RAGGETT, AND A. WARBURTON. "Telesoftware—Value Added Teletext." *IEEE Trans. on Consumer Electronics*, Vol. CE-26, Aug., 1980, pp. 555–567.

HEROT, C. F. "Spatial Management of Data." *ACM Trans. on Database Systems*, Vol. 5, No. 4, Dec., 1980, pp. 493–514.

HOLMÉ, J. M. "Gateway Connections in External Computers." *Videotex '81*, pp. 495–505.

HOOPER, R. "Advances in the United Kingdom." *Videotex '82*, pp. 527–536.

HORSPOOL, R. N., AND G. V. BOCHMANN. "Memory Structures for Videotex." Technical Report, Départment I.R.O., Université de Montréal, August, 1982.

HUSSEIN, S., ET AL. "Terminal Evolution for Business Videotex Services." *Proc. NTC*, 1981, pp. D1.5.5–D1.5.4.

ISO. "Draft Connection-Oriented Basic Transport Protocol Specification Version 1.2." ISO TC 97/SC16/WG6, Jan., 1982.

ISO 646. "Information Processing—7-Bit Coded Character Set for Information Exchange." Draft International Standard ISO/DIS 646, 1982.

ISO 2022. "Information Processing—ISO 7-bit and 8-bit Coded Character Sets—Code Extension Techniques." Draft International Standard ISO/DIS 2022.2, 1982.

ISO 6429. "Information Processing—ISO 7-bit and 8-bit Coded Character Sets—Additional Control Functions for Character—Imaging Devices." Draft International Standard ISO/DIS 6429.2, 1982.

ISO 6937. "Coded Character Sets for Text Communication." Draft Proposal ISO/DP 6937.2, 1981.

KLEINROCK, L. *Queuing Systems*, Vol. II, New York: Wiley, 1976.

KNOWLTON, K. "Progressive Transmission of Gray-Scale and Binary Pictures by Simple, Efficient and Lossless Encoding Schemes." *Proc. IEEE*, Vol. 68, No. 7, July, 1980, pp. 885–896.

KOBAYASHI, H. *Modeling and Analysis: an Introduction to System Performance Evaluation Methodology*, Reading, Mass.: Addison-Wesley, 1978.

LAMBERT, O., ET AL. "Antiope and DRCS." *IEEE Trans. on Consumer Electronics*, Vol. CE-26, Aug., 1980. pp. 600–604.

LE MOIGN, D. "La maquette annuaire électronique." CCETT, Rennes, France, 1981.

LINK Research Memoranda. "The Telephone Network as Viewdata 'Front-End' Processor." Videotex Research Memoranda, Vol. 2, Nov., 1980.

LOCHOVSKY, F. H., AND D. C. TSICHRITZIS. "Interactive Query Languages for External Databases." Technical Report, CRSG, University of Toronto, March, 1981.

LORIG, C., J.-C. RAHUEL, AND C. ROUX. "Combined Telewriting—Videotex Terminal." *L'écho des recherches* (English issue), 1981, pp. 27–34.

LUCKY, R. W., AND J. SALTZ. *Principles of Data Communication*, New York: McGraw-Hill, 1968.

MADELEINE, J., G. GARDARIN, AND J. GECSEI. "Comportement des bases de données videotex." Technical Report, Départment I.R.O., Université de Montréal, August, 1982.

MANTEL, H. "The Seltext Center for the German Bildschirmtext Network." *Videotex '81*, pp. 179–188.

MARTIN, J. *Design and Strategy for Distributed Data Processing*, Englewood Cliffs, N.J.: Prentice-Hall, 1981.

MARTIN, J. *Design of Man-Computer Dialogues*. Englewood Cliffs, N.J.: Prentice-Hall, 1973.

MAURER, H. A. "Will MUPID Revolutionize Austria's Videotex?" *Videotex '82*, pp.187–197.
MAURER, H. A. "Videotex Makes Dramatic Breakthrough." *Viewdata '82*, Online Conferences, London.
MAURER, H. A., AND I. SEBESTYEN. "Unorthodox Videotex Applications: Teleplaying, Telegambling, Telesoftware and Telecomputing." *Videotex '82*, pp. 199–210.
MAURY, J. P. "Plans and Projection for the Electronic Directory Service." *Viewdata '80*, pp. 39–50.
MCDONALD, A. "Teleview, a 3-Chip System for Teletext and Viewdata." General Instrument Bulletin 0412 A, 1980.
METZ, D. "Cost-effectiveness of Fiber Optics Getting Closer Look." *Cable Age*, Oct. 5, 1981.
MINI HUB. "Fiber Optic Subscriber Distribution System." Times Fiber Communications, Inc., 358 Hall Ave., Wallingford, Conn., 1981.
MOIR, C. "BBC Micro." *Practical Computing*, Jan., 1982, pp. 57–60.
MONEY, S. *"Teletext and Viewdata."* London City: Newness Technical Books, Butterworth & Co., Ltd., 1979.
MORAN, T. P. "An Applied Psychology of the User." *ACM Computing Surveys*, Vol. 13, No. 1, March, 1981, pp. 1–11.
MOTHERSOLE, P. L. "Teletext and Viewdata: New Information Systems Using the Domestic Television Receiver." *Proc. IEEE*, Vol. 126, Dec., 1979, pp. 1350–1354.
MULLARD. *"Multitext Technical Information."* London: Mullard Ltd., 1978.
NABU. "An Integrated Cable Solution for the 80s and 90s." Nabu Manufacturing Corp., Ottawa, 1981.
NICOL, R. C., AND B. A. FENN. "Coding for Viewdata 'Picture in Picture' Facility." Proc. Picture Coding Symp., Ipswich, England, 1979, p.13.
NOIREL, Y., AND G. DUBLET. "DIDON: A Data Broadcasting System." IFIP Conference, 1979 (Unesco).
O'BRIEN, T. E. "A Unified Approach to Data Transmission over CATV Networks." Cable '81, 30th Annual Convention and Exposition of NCTA, Los Angeles, May, 1981, pp. 119–123.
ORSNAES, J. "User Reactions to Keyword Access Videotex." *Videotex '82*, pp. 315–320.
OTTMANN, T. A., A. L. ROSENBERG, AND L. J. STOCKMEYER. "A Dictionary Machine (for VLSI)." *IEEE Trans. on Computers*, Vol. C-31, No. 9, Sept., 1982, pp. 892–897.
PALME, J. "A Viewdata Interface for Arbitrary Computer Programs." *Software Practice and Experience*, Vol. 10, 1980, pp. 987–991.
PETERSON, W. W., AND E. J. WELDON, JR. *Error-Correcting Codes*. Cambridge, Mass.: The M.I.T. Press, 1972.
PLP. "Presentation Level Protocol." American Telephone and Telegraph Co., Parsippany, N.J., May, 1981.
PLPS. "Videotex/Teletext Presentation Level Protocol Syntax (North American PLPS)." Preliminary Standard T.500-1982, Canadian Standards Association ISSN0317-7874, Aug., 1982.
PLUMMER, R. P., ET AL. "4004 Futures for Teletext and Videotex in the U.S. "*IEEE Trans. on Consumer Electronics*, Vol. CE-25, No. 3, July, 1979, pp. 318–326.
REISNER, P. "Human Factors Studies of Database Query Languages: A Survey and Assessment." *ACM Computing Surveys*, Vol. 13, No. 1, March, 1981, pp. 12–32.
ROBERTSON, G., D. MCCRACKEN, AND A. NEWELL. "The ZOG Approach to Man-Machine Communication." Technical Report CMU-CS-79-148, Carnegie-Mellon University, Oct., 1979.
ROBINSON, G., AND W. LOVELESS. "Touch-Tone Teletext—a Combined Teletext-Viewdata System." *IEEE Trans. on Consumer Electronics*, Vol. CE-25, No. 3, July, 1979, pp. 298–303.

ROSANOFF, R. A. "A Survey of Modern Nonsense as Applied to Matrix Computations." AIAA/ASME 10-th Structures, Structural Dynamics and Materials Conference, New Orleans, April, 1969.

ROSENFELD, A. "Picture Processing Survey: 1976." *Computer Graphics and Image Processing*, Vol. 6, 1977, pp. 157–183.

ROUILLY, Y. "The Role of Viewdata in Office Automation." *Vidéotex '81*, pp. 33–38.

SABLATASH, M., AND R. FITZGERALD. "An ISO OSI Layered Architecture for the Canadian Broadcast Telidon System." Proc. Globecom '82 Conference, (IEEE), Miami, Fla., Nov., 1982.

SALTON, G. "*The SMART Retrieval System: Experiments in Automatic Document Processing*, Englewood Cliffs, N.J.: Prentice-Hall, 1979.

SCHMID, P. E., AND C. A. MAURER. "Implementation of a Presentation Level Protocol in National Videotex Services." *Videotex '81*, pp. 387–399.

SHLIEN, S., R. FITZGERALD, M. GUILLET, AND B. CARON. "Suggested Schemes for Encoding Grey Level Bit Pictures for Telidon." Proc. CMCCS '81, 1981.

SHNEIDERMAN, B. "Improving the Human Factors Aspect of Database Interactions." *ACM Trans. on Database Systems*, Vol. 3, No. 4, Dec., 1978, pp. 417–439.

SMITH, L. C. "Artificial Intelligence Applications in Information Systems." *Annual Review of Information Science and Technology*, Vol. 15, 1980, Chapter 3.

SOURCE DIGEST. "A Directory of Services on The Source." Vol. 1, No. 1, Nov., 1981.

STERIA. "Videopac. Technical Manual," Steria, Vélizy, France, 1982.

STERIA. "Présentation fonctionnelle des modules logiciels de la gamme Videopac." Steria, 26 Av. de l'Europe, 78140 Vélizy, France, 1981.

STEWART, T. "PRESTEL—How Usable Is It?" Proc. of Human Aspects of Telecommunication Congress, Munich, Oct., 1979, Springer Verlag (1980).

STOREY, J. C., ET AL. "An Overview of Broadcast Teletext Systems of NTSC Television Standards." Proc. SMPTE Conference, Toronto, Feb., 1980a.

STOREY, J. C., ET AL. "A Description of the Broadcast Telidon System." *IEEE Trans. on Consumer Electronics*, Vol. CE-26, Aug., 1980b.

STOREY, J. R. "Terminals for Future Home Services Received via Wideband Networks." CCTA 21st Annual Convention, May, 1978, Montreal, pp. IV-85-IV-94.

TANABE, M. "Progress and Prospect of the Captain System." *Videotex '81*, pp. 65–78.

TANENBAUM, A. S. *Computer Networks*. Englewood Cliffs, N.J.: Prentice-Hall, 1981.

TEKTRONIX. "4027A Color Graphics Terminal." Technical Manual. Part No. 070-4173-00, Feb., 1982.

TÉLÉTEL. "La lettre de Télétel." Projet Télétel, 78140 Vélizy, France, Sept., 1979; Jan., 1980; Feb., 1982.

TÉLÉTEL. "Brochure technique Télétel." Projet Télétel, 78140 Vélizy, France, 1980.

TELIDON Behavioral Research, Vol. 2. "The Design of Videotex Tree Indexes." Canada Dept. of Communications, May, 1981a.

TELIDON Host Database System Design Specifications Manual. Canada Dept. of Communications, 1981b.

TENNE-SENS, A. "Telidon Graphics and Applications." *Displays Technology and Applications*, Vol. 3, No. 3, July, 1982.

THOMPSON, V. "Educational Trial of Prestel." *Viewdata '81*, p. 303.

TOMPA, F. W. "Retrieving Data Through Telidon." Proc. CIPS Conference, Saskatoon, Canada, 1982.

TOMPA, F. W., J. GECSEI, AND G. V. BOCHMANN. "Data Structuring Facilities for Interactive Videotex Systems." *Computer*, Vol. 14, No. 8., Aug., 1981, pp. 72–81.

TREURNIET, W. C. "Display of Text on Television." CRC Technical Note No. 705-E., Canada Dept. of Communications, Ottawa, May, 1981.

TSICHRITZIS, D. C. "Form Management." *Communications of the ACM*, Vol. 25, No. 7, July, 1982.

TYDEMAN, J., H. LIPINSKI, R. ADLER, M. NYHAN, AND L. ZWIMPFER. *Teletext and Videotex in the United States*, New York: McGraw-Hill, 1982.

TYLER, M. "Electronic Publishing: A Sketch of the European Experience." Workshop on Teletext and Videotex in the U.S., Institute for the Future, 1979.

VIDEO PRINT (newsletter) A. P. Brigish, (ed.). International Resource Development, Inc.

VIVIAN, R. H., AND S. DAY. "Alphageometrics Enhance Teletext Graphics." *Videotex '81*, pp. 5–17.

WINSBURY, R. "The Electronic Bookstall." International Institute of Communications, 1979.

WOOLFE, R. *Videotex, the New Television-Telephone Information Services*. London: Heyden Books, 1980.

ZIMMERMAN, H. "OSI Reference Model: The OSI Model of Architecture for Open Systems Interconnection." *IEEE Trans. Communications*, Vol. COM-28, April, 1980, pp. 425–432.

ZIPF, G. K. *Human Behaviour and the Principle of Least Effort.* Reading, Mass.: Addison-Wesley, 1949.

Index

A

B

C

D

E

F

G

H

I

K

THE ARCHITECTURE OF VIDEOTEX SYSTEMS

JAN GECSEI

In this new book **Jan Gecsei** focuses on the technical aspects of television-based videotex systems. Videotex, a recent development combining advances in computers, telecommunications, and consumer electronics, offers easy access to a variety of information services both to the public at large and to business customers.

Based on the ISO Open Interconnection model (OSI) *and divided into five parts, the book:*

- provides an introduction to videotex and background material about similar systems and computer networks;
- treats the communications media and networks used in videotex (corresponding to OSI layers 1-5);
- discusses the presentation level (layer 6), treating in detail the image-coding options in current use, their impact on terminal design, and the related problems of national and international standardization;
- covers the application level—deals with databases for videotex and teletext, gateways, service computers, and service providers' equipment;
- touches upon themes important for the future of videotex: alternative methods of interfacing with the user, telesoftware (seen as the key to distributed processing in videotex), and methods of performance evaluation.

PRENTICE-HALL, INC., Englewood Cliffs, N.J. 07632

ISBN 0-13-044776-5