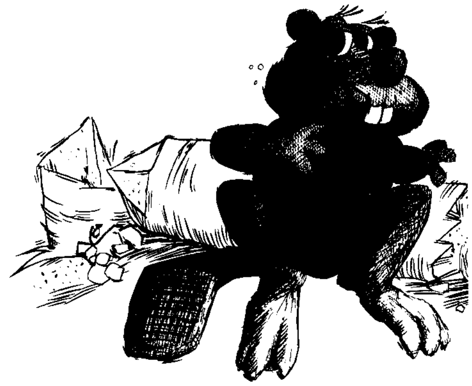
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



Data Structuring Facilities for Interactive Videotex Systems

Frank Wm. Tompa
Jan Gecsei
Gregor V. Bochmann

CS-80-50

November, 1980

Data Structuring Facilities for Interactive Videotex Systems

Frank Wm. Tompa †
Jan Gecsei ‡
Gregor V. Bochmann ‡

ABSTRACT

Interactive videotex systems will soon emerge as a principal information, entertainment, and communications medium. Until now there has been much written about the image presentation and data transfer facilities of the several current systems, but little attention has been devoted to the provisions for accessing pages of videotex data.

In this paper, existing data structuring facilities for interrelating videotex pages are described, and alternative structures are proposed. Throughout, emphasis is placed on the aspects of data organization that are visible from the perspective of videotex users as well as on those that affect the ability of information providers to present their data.

Key phrases: videotex, database, data structure, page organization, viewdata, Captain, Prestel, Teletel, Telidon.

CR *categories:* 4.33, 3.72, 4.32, 8.2.

November 13, 1980

† Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada.
‡ Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, P.Q. H3C 3J7, Canada.

Data Structuring Facilities for Interactive Videotex Systems

Frank Wm. Tompa
Jan Gecsei
Gregor V. Bochmann

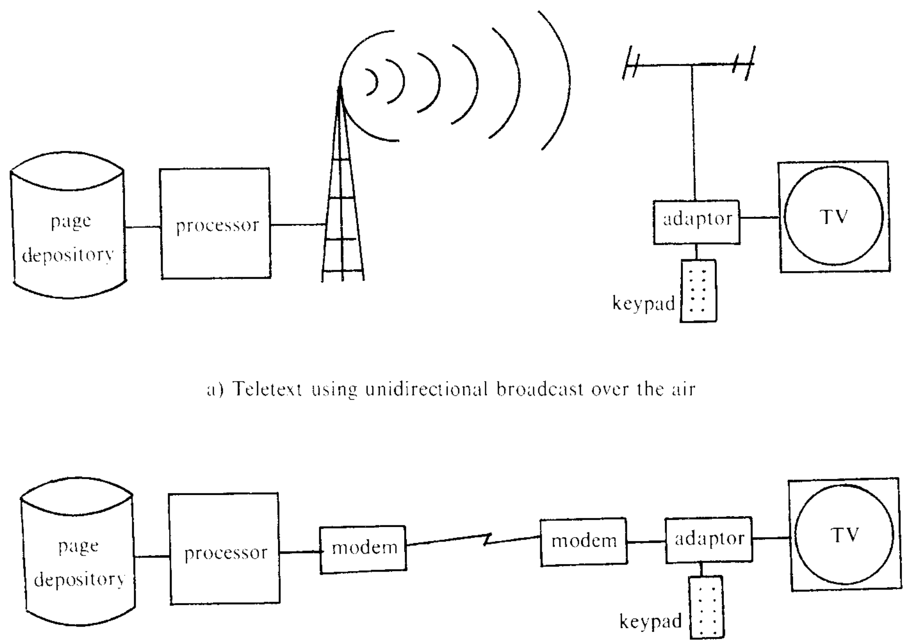
1. INTRODUCTION

The ability to access large data banks of information through the use of a television set in the home or office has long been proposed as a capability of the future. In fact, it has become a reality of the present, at least in a limited way. Around 1970 the British Broadcasting Corporation developed Ceefax, originally to provide captioning for the deaf, and by 1974 the British Post Office developed Prestel, originally called Viewdata and resulting from research towards a picture-phone.[11] Since that time, much research throughout the world has been devoted to the development of competitive and complementary systems (e.g., Captain,[16] Teletel,[3] and Telidon[12]).

Two modes of operation have developed to date: strictly broadcast, or one-way, *teletext* systems and interactive, or two-way, *videotex* systems (Figure 1). The first of these follows the pattern established by Ceefax, that is, transmitting pages of information in a cyclic pattern and using a local adaptor on each television receiver to capture the page selected by a user when it next appears in the cycle. As an immediate consequence of this mode of operation, there is a trade-off between the number of pages of information available on one channel (and therefore the length of the cycle) and the delay in servicing one user's request to see a page (which is proportional to the number of unrequested pages that occur in the cycle between the time a user posts a request and the appropriate page is next transmitted†). Teletext service is commonly, although not exclusively, provided by over-the-air broadcasting of the signals.

Interactive videotex systems, on the other hand, disseminate pages of information solely in response to user demand. A user's request for a page is transmitted from a videotex terminal to a host processor, which in response sends the appropriate information back to the originating user. Unlike teletext, there is (virtually) no trade-off between the number of pages available and the response time; however there is a dependence of the response time on the number of users requesting service simultaneously. Videotex systems typically connect users to the central system via telephone lines, packet subnetworks, or cable television.

† In fact, whole pages need not be transmitted contiguously, but instead subpages may be interleaved; the trade-off still applies.



a) Teletext using unidirectional broadcast over the air

b) Videotex using bidirectional point-to-point (e.g., telephone) connections

Figure 1: Schematics for teletext and interactive videotex reception

The provisions of two-way communication in videotex systems and the independence of response time on database size has encouraged the development of further interactive services, including teleshopping, telebanking, telemonitoring, and interactive access to other centralized facilities from the home or office. In fact, the integrated access to multiple interactive services has been promoted as the greatest potential for videotex systems and is now beginning to be realized by Omnitel[1,5] and Mitrenet.[8]

Overviews and further information about teletext and videotex systems can be found in several books[6,14] as well as in articles about individual systems.

Data organization facilities are a major part of videotex system design. In this paper, we first summarize the facilities for inter-relating pages of information within the major current videotex systems, i.e., Captain, Prestel (and its direct derivatives including Germany's Bildschirmtext), Teletel, and Telidon. † We then describe the data structure requirements imposed by additional videotex facilities, including response pages, gateway pages to other databases, and accounting. In Section 3 we describe the experimental Teletel/Star system developed by the French Centre Commun d'Etudes de Télévision et Télécommunications (CCETT).

† Henceforth the use of the term "Captain" will refer to the system operated by the Japanese Captain System Research and Development Center, "Prestel" will refer to the system operated by the British Post Office, "Telidon" to the system operated by the Canadian Department of Communications, and "Teletel" to the system operated by the French Direction Générale des Télécommunications.

Finally we propose some alternatives to existing structures and evaluate their implications on the providers and consumers of videotex-based information.

Before discussing videotex systems further, we introduce here some vocabulary that will enable us to describe the systems using a uniform terminology. Data is primarily divided into discrete *pages* identifiable as logical units. In some cases, for example if the contents of a page is more than can be displayed at one time on a videotex screen, the page may be sub-divided into *frames* each of which forms one display image. There need not be any relation between videotex pages or frames and the units of storage ("sectors", "pages" or "blocks") used to maintain the data. The computer system on which the videotex pages are stored is managed by a *videotex system operator* (for example, a post office, telephone company, or television cable company). The pages themselves are supplied and editorially controlled by an *information provider* (commonly referred to as an IP). In fact, an information provider may also serve as the videotex system operator, and in that role is often referred to as a "value-added system operator." The consumer of the information, that is, the individual accessing the videotex system for the purpose of information retrieval or to invoke a transaction, will simply be termed a *user*.

Because the research and development of videotex systems are progressing at a very fast rate on a worldwide scale, often in a proprietary manner, we have been forced to limit our discussion of existing facilities to publicized systems only. As a result, the details included in this paper are accurate as of mid-1980, and they occasionally ignore developments contained solely within research laboratories.

2. EXISTING DATA STRUCTURING FACILITIES

2.1. The basic tree structure

The underlying data structure for all current interactive videotex systems is a (positional, labelled, rooted) tree of page nodes (Figure 2), each node serving as the root for a number of subtrees, depending on the particular system. The content of any page can include text and/or graphics and can be used to convey application information and/or index or routing prompts. Pages are numerically identified by their position in the tree; for example the root is page 0, its descendents have identifiers from 1 to 9, the descendents of page 1 have identifiers 11, 12,..., 19, the descendents of page 12 have identifiers 121, 122,...,129, etc. A user can request a page directly at any time by entering its numeric identifier via a keypad or keyboard attached to the videotex terminal.

The provision of 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 twenty-six frames, distinguished by appending an appropriate letter of the alphabet to the page identifier. Direct access to a page results in the display of the *primary* frame (labelled *a*), the *secondary* frames being reachable by subsequent sequential access only. Similarly, in Captain each page has one to ten frames (distinguished by an appended digit). Alternatively, Telidon allows multiple frames at the leaves of the tree only, thus distinguishing in the system implementation between so-called index and document pages (the content of either is still under complete control of the information provider). A document page can consist of up to 1000

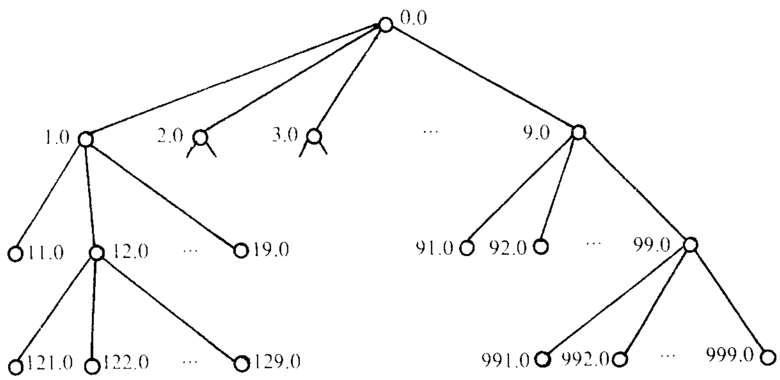


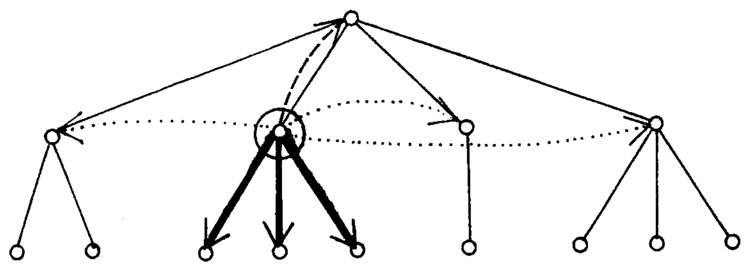
Figure 2: A tree of page nodes, using Telidon identifiers

frames identified by one, two, or three digits following the decimal point in the page identifier (e.g., 122.0, 122.1,..., 122.999), and, unlike Captain, Prestel, and Teletel, 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 identifier, e.g., 12.0.)

The tree structure 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 subtrees to contain general-purpose and index information controlled by the videotex system operator. Such localization of the information providers' pages simplifies the system operator's tasks of preventing interference among information providers and providing accounting (statistical and financial) to each information provider (see Section 2.3.3). The visibility of the tree structure through page identifiers also reinforces an information provider's identity to the user by explicitly containing the corresponding root page's identifier.

The tree structure may also serve as the framework in which data is presented to the user in cases where a logical hierarchy is inherent in the information. Telidon alone provides facilities for traversing the tree structure under direct user control (without any provisions by the information provider). From any index page, a user can request its immediate ancestor, an immediate descendent, or any sibling pages (i.e., those sharing the same parent), without using the target page's numeric identifier (Figure 3). Unlike the use of numeric identifiers as *absolute* page labels,† the interpretation of a user's directive is dependent on the page currently being accessed; this is therefore a form of relative page addressing, henceforth called access by *relative page label*. Such an ability to traverse the tree is extremely useful for browsing through hierarchically structured data.

† Throughout this paper, the term "identifier" will be used to refer to a unique page designator, whereas "label" will be used to refer to a name available to users for accessing a page.



From the circled node, relative access is provided to the immediate ancestor (--➤), immediate descendants (➡), and sibling pages (····➤) without reference to page identifiers.

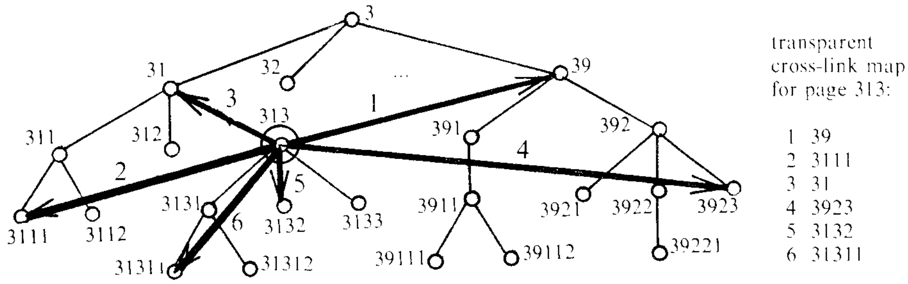
Figure 3: Tree traversals in Telidon

2.2. User-oriented access paths

Information rarely fits comfortably in a unique hierarchy. For example, it seems unreasonable to require information about Swedish restaurants to be listed exclusively under the Swedish subheading in the restaurant subtree or under the restaurant subheading in the Swedish subtree, thus forcing all users to adopt the same hierarchical ordering for retrieving the information. Furthermore, if users are given only one tree structure through which to access data, they will often be unable to locate required information without first traversing many irrelevant sections of the tree.[9] 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 accesses, and possibly communications) will be large despite individual page retrievals being relatively efficient. Secondly, information providers will have only limited opportunity to entice users to request related information elsewhere in the tree.

It is extremely wasteful to allow multiple listings solely by physically duplicating the data. Captain, Prestel, Teletel, and Telidon therefore have facilities for allowing information providers to superimpose arbitrary directed graph structures on the underlying tree. Rather than interpreting relative page labels in the context of the underlying tree structure, an information provider can build an arbitrary access structure by establishing a transparent cross-link map to selected pages. In building the cross-link map for a page, the information provider associates with a (typically one-digit) numeric label the absolute page identifier of the page to be retrieved when a user enters that label. The page itself will usually contain a multiple-choice display that indicates the valid numeric labels together with content descriptors for the corresponding pages (Figure 4).

Although the access structures can be arbitrarily constructed (as long as the number of edges leaving a node does not exceed some bound), it has been found useful to exercise some control over the access paths so that users maintain a sense of consistency. Thus, as part of their editorial policy, information providers typically select one or several presentation styles for relating their pages of data. (Some commonly used structures have been identified and defended elsewhere.[15]) In addition, certain digits may be reserved for common functions such as "return to table of contents" or "proceed to next".



display for page 313:

Page 313 **Great Lakes**

The Great Lakes are located between the United States and Canada, covering an area of approximately 94,690 square miles.

Key 1-6 for more information:

1	Overview
2	Lake Superior (31,810 sq. mi.)
3	Lake Huron (23,000 sq.mi.)
4	Lake Michigan (22,400 sq. mi.)
5	Lake Erie (9,940 sq. mi.)
6	Lake Ontario (7,540 sq. mi.)

Figure 4: Cross-link map for user-oriented access structures

In addition to allowing a user to enter a relative numeric directive to traverse access paths, Teletel also has a primitive keyword facility. The Teletel architecture is such that there are two levels of access: the first to select a *service* (possibly corresponding to all the pages of one information provider) and the second to traverse the pages within a service. The keyword structure matches this architecture exactly: one set of keywords is used to give mnemonic names as absolute labels to individual services and each service may use keywords to give mnemonic names as absolute labels to individual pages. For example, "travel" may be used to select a particular service within which "Toronto", "hotels", or "parliament" can be used at any time to access the corresponding page as determined by the information provider.

In summary users may select pages at random (using absolute page identifiers), may traverse the directed graphs designed by the information providers (using relative page labels), or may traverse the underlying tree structure supplied by the system operator (again using relative page labels). Finally all major systems give the user the opportunity to backtrack through retrieved pages in the reverse order of access (up to at most three pages in Prestel and ten pages in Telidon), and Telidon provides a simple command to allow a user to retrace the path in the forward direction again.

2.3. Support for other videotex facilities

To be useful, videotex systems must provide services other than mere page retrieval. Because it is important that such services are not overlooked in system designs and evaluations, some are briefly described here.

2.3.1. User response pages

Although at first videotex systems seem to be primarily information retrieval facilities, their functionality should not be limited to retrieval. The purpose of many retrievals is to obtain information in order to begin a transaction. For example a request for Swedish restaurants is not likely to be for mere academic interest, but rather so that a reservation can be made (and the food eventually consumed). Similarly, a look at stock market quotations is to determine whether to buy, to sell, or to stay pat.

Business practice dictates that a customer is more likely to respond to information if such a response can be made immediately. Thus, to be practical, videotex systems must be truly interactive and therefore distinguishable from teletext services, by (at the least) providing facilities for users to respond to pages produced by information providers.

In Prestel, an information provider can create a so-called *response page*, as distinguished from an *information page*. Such a 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 data is under the control of the information providers. When completed, the form is appended to a queue of users' responses, which can subsequently be examined by the appropriate information provider only.

For example, if a restaurant manger is an information provider (or uses the services of a videotex-based booking agency), he or she will likely create a videotex information page on which conventional data may be displayed (e.g. logo, name, address, telephone number, and menu extracts). As well as giving routing information for related information pages, the manager can give a prompt such as "key 9 to make a reservation." The page's cross-link map will have been set to designate a suitable response page, also created by the manager, as the corresponding target. That page will be formatted to ask the user to key in the date, time, and number in the party (all of which can be entered numerically) and to fill in the user's name and address automatically. When completed, the reservation form will be forwarded to the information provider, who can then process the information as if it were relayed by telephone. Similar transactions can be invoked for other applications.

The service provided by a response page facility is comparable to a limited electronic mail service. In fact, because response pages can be created to allow free-form input, arbitrary messages can be sent if full character set keyboards are used. However, since only recognized information providers are eligible recipients, only a limited number of "mailboxes" are provided, and mail traffic is not as heavy as in a more general mail system.

2.3.2. Gateway pages

Even as a pure information retrieval facility, videotex systems will not realize their potential if they can only be used in isolation. There are many databases that have been built up independently of videotex – for business, libraries, and entertainment. Many of these databases will continue to be maintained independently, and therefore access must be provided to such external, or *third-party*, databases from within videotex systems.

The data in some third-party databases may already be in a format that is compatible with the videotex 40-character line standard. For other databases, the data can be reformatted into videotex pages, either when requested or *en masse* in anticipation of request (possibly by splitting 80-character lines in half, but more typically through the use of specialized reformatting routines). Finally, for some third-party databases, the videotex terminal could be made to function as a conventional alphanumeric terminal having 80-character lines.† In any of these situations, a natural mechanism for a user to request access to a third-party database is to request the appropriate *gateway* page which will have been created by an information provider.

The role of a gateway page is to integrate access to non-videotex services with videotex's page-oriented system. When a user requests a gateway page for a third-party database, the log-in protocol for that database system is initiated on the user's behalf, possibly requiring additional input from the user. Thereafter, the videotex system becomes transparent to the user, merely serving as an intermediary node in the access network by relaying the interactions between the user and the database. When the user disconnects from the third-party database system, control returns to the videotex system, and the preceding user context is re-established in preparation for further videotex commands. Such a facility is incorporated into Germany's Bildschirmtext system.

The data structures required to support the simplest form of gateway are not complex: linking information, including the (network) address of the gateway's target and interface protocols, must be stored with the page. If the contents of the third-party database are to be converted to a page-oriented format, however, the interface protocols may require extremely sophisticated techniques to control the reshaping of tables, text, and graphics.

A closely-related facility is represented by Telidon's *action pages*, which allow access to executable programs. When the program is loaded into and executed in the user's terminal (instead of being executed by the remote processor), such a facility is known as *telesoftware*. Once again integrated access

† This last alternative currently requires that the terminal be provided with suitable hardware to provide adequate resolution for the small characters that result from double-length lines. Such an option is available, for example, with the new Telidon Integrated Videotex Terminal manufactured by Electrohome in Kitchener, Ontario, Canada.

can be achieved through the use of a gateway page. When an action page is requested, videotex's page orientation can become transparent in order that the user may interact directly with the running program. Again, upon termination, the control and context can be returned to the gateway for further page-oriented activity.

2.3.3. Accounting provisions

Accounting is a vital function in any system. In videotex systems, where services are provided from many independent sources (including the system operator, the communications providers, and the information providers), accounting is extremely critical. The information required for accounting will be briefly considered here from only two viewpoints: the data collected per information provider and the data collected per user. In neither case is the data page-oriented, but rather it may be handled in a conventional record-oriented manner (and, in fact, maintained in a separate database from that containing videotex pages).

There should be an accounting entry for each information provider. Such an entry must contain the information provider's name and address (for billing) as well as a description of the set of page identifiers that are under the information provider's control (this may be simply the page identifier of the root of the appropriate subtree). In addition there should be fields for the amount of storage consumed by the information provider's pages, the resources used for page updating (e.g., for creating pages, altering the contents, altering the routing), and the total of the page charges resulting from users' accesses (for crediting the information provider). In addition, more detailed accounting information (such as the number of accesses to each page, the number of traversals of each cross-link, and a collection of statistics on *complete access paths* to pages), would provide useful feedback on the effectiveness of page layouts and organization.

Similarly, a user's accounting information must be maintained. Again the user's name and address must be included, as well as accumulated charges for system usage (processor service, input/output operations, etc.), page access charges levied on behalf of information providers, and communications charges. If gateways to independent services are available through the videotex system, then the third-party charges should also be accumulated in the user's accounting entry in order that the user is billed from only one source.

3. TELETEL/STAR

The French Centre Commun d'Etudes de Télévision et Télécommunications (CCETT) has developed a videotex system which is expected to be publicly available (at least on a trial basis) in late 1981. The Star experimental system (Source Teletel Accès Réseau) is operational in a prototype form in Rennes, based on a multiprocessor-multiminicomputer host architecture and the French Transpac public communication network.[10] It is the Star system that forms the base of the "electronic telephone directory" trials taking place in Saint-Malo since 1980. Throughout this section, the term "Teletel/Star" will refer to the videotex system based on the Star system.

Similarly to the systems described in Section 2, Teletel/Star is based on pages consisting of a (non-empty) sequence of frames and organized into a tree upon which an information provider superimposes access paths. As in the conventional Teletel system, there are two distinct levels of access, the first to choose a service and the second to traverse a structure within one service. Unlike the other systems, there is no limit to the number of frames per page nor to the number of subtrees per node (the information provider can specify the number of digits that are necessary to identify a descendent; for example, the immediate descendents of page 25 may be labelled 250, 251,..., 259; or 2500, 2501,..., 2599; or 25000, 25001,..., 25999; etc.). The most striking aspects of the Teletel/Star design, however, lie in the sensitivity of the system to a user's access path when interpreting user directives and in the innovative provisions for numeric and keyword directives.

3.1. The basic node control programming language

Associated with each page in a service's tree is a program that is written by the information provider at the time of page creation and executed whenever a user accesses the page.[7] The function of the program is to control the display of information on the videotex terminal and to interpret the commands input by the user. To govern program execution, there is associated with each user a state vector that includes the page identifier of the current page whose program is being executed, the identifier of some other "base" page, a condition code array consisting of sixteen flags, and two stacks containing identifiers of (other) pages.

Flow of control within a page's program is governed in a manner similar to a decision table. The information provider specifies a sequence of conditional expressions involving the sixteen flags. The expressions are sequentially tested, and as soon as one is found to be true, a corresponding sequence of instructions is executed (either once only or until the expression becomes false).

In all instructions having a page label as an operand (e.g., for branching or calling other pages' programs), a label *expression* can be used in place of a numeric page identifier. To implement a standard tree structure, the page retrieved when a user enters an integer is the one having a label equal to the current page's label concatenated with the numeric input. In the language for Teletel/Star, this would be indicated as "$,CCT,@" where $ is a numeric register containing the current page label, CCT indicates concatenation, and @ is a register containing the user's input; for example, on page 25 when the user inputs a 3, "$,CCT,@" assumes the value 253. To produce non-tree structures, operators for addition and subtraction are also provided; on page 25 when the user inputs a 3, "$,ADD,@" assumes the value 28 and "$,SUB,@" assumes the value 22. By using (absolute) numeric page identifiers as well as register names in combination with these three operators, arbitrary user-oriented access paths can be constructed by the information provider.

3.2. Servicing keyword directives

In keeping with the Teletel keyword facility as described in Section 2.2, Teletel/Star allows users to traverse the pages within a service by using keywords as well as numeric directives. Unlike the conventional Teletel facility, however, the keyword is interpreted as a relative page label, i.e., the designated page depends on the page being accessed by the user at the time the directive is entered. Thus, in addition to user-oriented access structures based on directed graphs with numeric labels on the edges (as in Figure 4), an information provider can define a directed graph with keyword labels on the edges.

As an example, consider a hypothetical metals service provided via such a system (Figure 5). If a user accesses a page containing information about steel production in Canada, the keyword "copper" may lead to a page on Canadian *copper* production, "consumption" may lead to a page on Canadian steel *consumption*, and "France" may lead to a page on *French* steel production. If "iron" is entered after having entered "France", the user would be lead to *French iron* production. Thus the system is cognizant of the user's context within the database, and it interprets keyword directives accordingly.

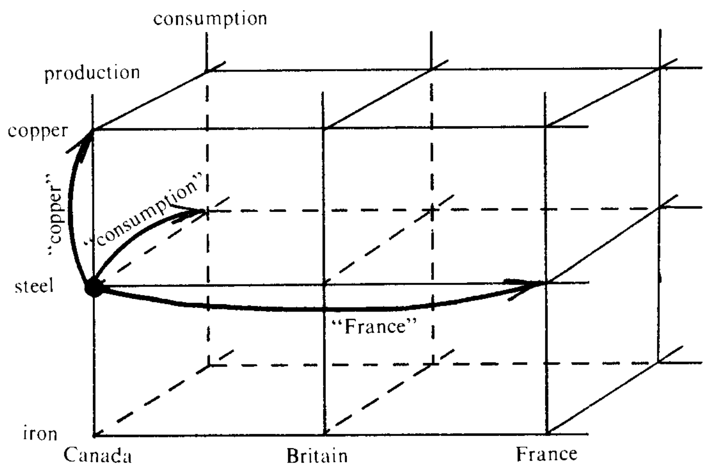


Figure 5: Grid structure using keywords

Rather than storing largely overlapping keyword directories at each node, the keyword facility is based on a service-wide set of indices defined by the information provider (Figure 6). As part of a page's program, the information provider can designate arbitrarily many keyword/index pairs, each of which causes that keyword within that index to refer to that page, henceforth known as the *target page*. Thus any page can have several keyword entries in arbitrarily many indices for which it is the target. In Figure 6, page 34 is the target page for "Britain" and "UK" in index IRON-PROD and for "iron" in index BRITISH-PROD.

Associated with each page is a sequence of up to three index names that together designate the dictionary to be used to interpret keyword directives issued when the user is accessing the page. These dictionaries can be configured to provide arbitrary user-oriented access structures. When a user enters a keyword, the first designated index is searched; if there is no match, the second and third indices are searched successively, stopping when/if a match is found. If a matching keyword is contained in one of the indices, the corresponding target page

IRON-PROD

Canada	31
Britain	34
UK	34
France	38
...	

STEEL-PROD

Canada	51
Britain	54
UK	54
France	58
...	

CANADA-PROD

iron	31
steel	51
copper	81
...	

BRITISH-PROD

iron	34
steel	54
copper	84
...	

contents of sample indices

page #	dictionary indices
31	IRON-PROD, CANADA-PROD, CANADA-IRON
34	IRON-PROD, BRITISH-PROD, BRITISH-IRON
51	STEEL-PROD, CANADA-PROD, CANADA-STEEL
81	COPPER-PROD, CANADA-PROD, CANADA-COPPER
...	...

Figure 6: Index structure for keyword processing

is accessed (i.e., its display is generated and its program executed); that is, the user proceeds along a directed labelled edge. Again referring to Figure 6, if a user enters the keyword "copper" while on page 31, the indices searched would be IRON-PROD, CANADA-PROD, and CANADA-IRON; because "copper" is not in index IRON-PROD, it would match the entry in CANADA-PROD, and thus page 81 would be the target page selected. If the keyword does not appear in any of the indices constituting the dictionary for the current page, an error flag is set and the current page's program may take corrective action if so-designed by the information provider.

3.3. Access path sensitivity

As described so far, the directed graph of pages is defined by the information provider and remains static until the information provider explicitly amends it. The Teletel/Star system, however, also provides facilities for causing the active set of labelled edges to be dependent not only on the current page, but also on the user's access path that was traced to reach that page.† For example, if a tour organizer wishes to include the same information about Toronto in the descriptions of several packaged tours, a videotex page for Toronto data can be constructed so that the keywords "schedule", "cost", and "hotel" are interpreted within the context of the tour being examined by the user. As a result, pages need

† It should be noted that the directed graphs in Teletel/Star are still static, predesigned access structures; they merely give the illusion of adapting to the user's access activity.

not be duplicated merely to provide different page-exit linkages.

The simplest form of access path dependent addressing is the ability to backtrack as provided by all videotex systems (see Section 2.2). As an extension of this idea, Teletel/Star provides an explicit stack of pages (controlled by push and pop) and a subroutine-like capability (controlled by call and return), both of which allow an information provider to cause the user's state to be saved and subsequently restored.

A third form of access path sensitivity results from the notion of a base page. As part of any page's program, the information provider can cause a register in the user's state vector (the *base*, denoted by #) to be assigned the identifier of an arbitrary page (e.g., selected by the label 25, $, or "$,CCT,@"). This base register may be used subsequently in any label expression (e.g., "#,ADD,@"), thus allowing the information provider to interpret a user's input with respect to some base that depends on the pages previously accessed.

To extend this capability to the processing of keyword as well as numeric directives, a label expression may involve the use of a dictionary. Normally a keyword is searched in the context of the current page's dictionary, but the use of the operator IDX in a label expression indicates that the dictionary of some (other) specified page is to be consulted. Thus, whereas normal keyword processing is defined by the program statement "$,IDX,@" (search for the input keyword in the current page's dictionary), the information provider can instead specify "#,IDX,@" (search in the dictionary associated with the base page) or use any other label expression involving IDX. For example, the tour organizer mentioned above could cause the base to be set at each tour's entry page and certain keywords to be interpreted with respect to that base.

4. ALTERNATIVE DATA STRUCTURING FACILITIES

The data structuring facilities of today's videotex systems are very similar. Only Teletel/Star has deviated from the others by introducing a sophisticated keyword facility and the concept of an executable program related to each page.

We will now re-examine some of the basic notions common to all systems and propose some alternatives. In Section 4.1 we discuss the role of the underlying tree structure, in Section 4.2 we propose the expunction of visible page identifiers, and in Section 4.3 we examine the applicability of conventional database system technology to videotex systems.

4.1. Eliminating the underlying tree structure

The basic tree structure was described in Section 2.1. As discussed subsequently, it is rare that the tree forms a suitable structure for presenting information (i.e., a suitable external schema); rather other structures are superimposed on the tree to provide better user-oriented access paths. In addition, the tree is not a convenient structure for data representation on the physical storage media (internal schema); rather the page identifiers are used independently of the tree structure for retrieval by a storage-oriented access method based on hashing, indexed sequential store, or some other keyed retrieval method. As a result, the primary use of the tree structure is as a storage management device for the convenience of the videotex system operator.

Instead of depending on the tree structure, which has on occasion been confusing to computer-naive users,[13] the underlying structure of the database may be better suited when viewed as a one-dimensional array of pages. The logical structure is then merely an arbitrarily large set of numbered pages having identifiers 1, 2,..., 9, 10, 11,..., 99, 100, 101,.... Such a structure is, in fact, the one most commonly used for other address spaces, such as for labelling the cells of a primary memory. In this section we will show that the facilities now provided by a tree would not be lost nor significantly impaired.

The tree structure provides a framework for storage management in terms of subtrees. Similarly the array structure can be managed in terms of logically contiguous blocks. Using an array, pages could be allocated to information providers in blocks of a suitable size, for example 100 or 1000 pages per block. Cross-links can still be established independently of the underlying structure (since there is no requirement that cross-links be contained within a subtree, there need not be a restriction that they remain within one block allocation), and thus there is no problem with the page identifier spaces for several information providers being interleaved, as long as the information providers are restricted to editing in their own subspaces. Such a structure is commonly used by operating systems: programs are allocated one or more regions in memory in which they can store arbitrary data, often including references to data within the system's and other users' regions. The extensive operating systems experience of managing primary memory and the allocation of blocks of tracks on secondary storage devices can be fruitfully applied to videotex page management. In fact, in keeping with this experience, it may be better if each information provider has a separate "virtual array of pages," some of which are designated to be "entry pages" to which cross-links may be made; this would likely simplify the task of maintaining consistency among the pages supplied by different information providers.

With the absence of the tree structure, the inclination to distinguish secondary frames from primary page frames also disappears. In fact, the distinction need not be made until an information provider constructs access paths. Whenever a new page/frame is needed, an unused slot is suballocated from one of the information provider's blocks and linked to the existing pages by cross-links or by insertion into a frame sequence. Thus, unlike in Prestel, Captain, and Telidon, frame sequences can be made arbitrarily long and the difference between primary and secondary frames is merely in their placement as designed by the information provider and in their access paths as interpreted by the users.

The remaining use of the tree structure is to serve as a focal point for an information provider; that is, the root of a subtree is identified as being an entry point and the set of page identifiers in the subtree is recognizable as belonging to the information provider. Designating an arbitrary page as the information provider's entry point causes no real problems, since it has already been found that commercial videotex systems require printed directories to catalog the information providers' entry labels. In Prestel, the information provider's name is placed next to the page identifier at the top of each frame; thus using the page identifier itself to indicate the information provider is unnecessary.

It may be argued that it is desirable for a user to return to the appropriate information provider's entry page by merely entering the first few digits of any current page's identifier, as is the case for Captain and Prestel. Because there is no obvious root for an array of pages, the alternative is to provide an implicit "link to information provider" from each page; for example, the entry "#0" or "↑" may be reserved to designate a user's request to access the appropriate information provider's entry page. (The videotex system can retrieve the appropriate page identifier from the information provider's accounting entry as described in Section 2.3.3.)

4.2. Eliminating users' awareness of page identifiers

Having found that the tree structure is of little value, it may be quickly realized that the page identifiers themselves are virtually meaningless. It is ironic that the videotex information medium requires the extensive use of printed material to inform users of page identifiers (see, for example, the *Prestel Business Guide* published periodically by The Financial Times Ltd.).

It has long been realized by programming language, operating system, and data base system designers that users' exposure to absolute memory addresses or address-dependent data base keys is highly undesirable. Such exposure inhibits the system's flexibility to reorganize the data (for example, to increase efficiency) because at any time any user may request a piece of data by an address that may no longer be valid. Similar criticisms can be levelled at page identifiers: they are non-mnemonic labels that inhibit the reassignment of pages to information providers or their re-use by an information provider within a service. † Again it needs to be shown that the useful roles of page identifiers as page labels can be better served by alternative means. (Naturally some form of page identifier must still be used by the system itself; the criticism here is directed at their visibility to users.)

Page identifiers serve as a means to traverse the set of pages in a videotex system. One role of the information provider is to develop access paths that anticipate users' needs; these are incorporated by means of multiple-choice displays and cross-links, for which absolute page identifiers are not required by users. This is, of course, enhanced when keywords are permitted, since the cross-linking becomes mnemonic.

Often users do not wish to traverse long access paths to retrieve often-requested pages, and absolute page identifiers allow direct access. The problem is easily circumvented by the information provider constructing an arbitrarily large set of quick-access absolute page labels (unrelated to the *system's* page identifiers) that are stored in an information provider's label map, similar to a page's cross-link map, and are usable from any page. For example, whereas entering "3" uses the current page's cross-link map to determine the target page, entering "#3" would use the information provider's map. The system could reserve "#0" to access the entry page (as mentioned in the previous section) or to serve as a complete system reset in order to insure that a user is not unwillingly trapped within an information provider's access structure. (In fact, a user *could* always

† The British Post Office is contemplating increasing the number of recognized information providers by using four-digit instead of three-digit identifiers for their roots; this may well cause problems for current information providers and users.

disconnect from the system by unplugging the receiver or hanging up the telephone, but this is inelegant.)

It is impossible for information providers to supply absolute labels for all their users' needs, and there is therefore still some demand for each user to access other arbitrary pages conveniently. With the use of numeric page identifiers, each user can keep a private, written list of commonly used identifiers for reference when using the system. A more satisfactory method, and a necessity if there is no direct access to page identifiers, is to have the system maintain for each user a private, relatively small absolute label map similar to that of the information provider. For example, the command "*3" could access the user's label map to determine the target page to be retrieved, and "**3" could be used to indicate that the current page (typically retrieved by following some access path) should be entered opposite label 3 in the user's label map.

Although absolute labels have been described in terms of numeric directives, their use is even more appealing with keyword labels. The availability of keywords would, of course, allow the absolute page labels to be mnemonic and, thus, easier to recall. Furthermore, the information providers' maps could contain arbitrarily many entries and each user's map may contain up to some fixed number of entries. † When a user issues a keyword directive, either the desired map could be indicated explicitly (e.g., "INDEX" *vs.* "#INDEX" *vs.* "*INDEX") or, similarly to Teletel/Star's indices, the maps could be searched in some prespecified order (e.g., *first* the current page *then* the user's map *then* the information provider's map). Such a system may eventually reduce the need for printed catalogs of information providers' pages.

Having removed page identifiers from the users' realm, it remains to remove them from the information providers' as well. In fact, information providers access pages in a manner similar to their users; thus the absolute labels together with relative cross-links should suffice. The only remaining use for page identifiers is to build the maps in the first place. Again the experience from programming language design could be applied: for example, similarly to conventional linked list manipulation, when a page is created, it becomes the current page and it must be linked into some access structure before proceeding to another page. Using the proposed facilities, it must either be entered into a label map as explained for users above, or it must be entered into some other page's cross-link map by the use of a suitable notational convention, e.g., "5@#3" could mean to assign it as the fifth cross-link on the page denoted by absolute label 3 and "7@2@INDEX" could indicate that the new page is to be linked as the seventh cross-link on the page accessed from the second cross-link on the page labelled INDEX.

† Depending on the system's cost to maintain label maps, the information providers and users could be charged for their maps according to the number of entries, as they are now charged for page storage and page accesses, respectively.

4.3. Providing conventional record-oriented facilities

It seems strange that although data base 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.

Consider, for example, a real estate application for which pages have been constructed to contain photographs, floor plan diagrams, and textual descriptions of houses available from some agent acting as an information provider. Current systems are not designed to service typical user requests from such a service. All the current videotex systems provide page access through unique identifiers only; that is, each user directive identifies at most one target page. However, the typical search for houses is specified by a boolean expression over a set of secondary keys that are not unique identifiers, e.g., "three bedrooms and a formal dining room and priced between $50000 and $70000." In fact, this is exactly the sort of request best handled by conventional data base systems.

The application of conventional data base technology to videotex systems can be recognized as soon as it is realized that "videotex page" may be made to be a valid domain in a record-oriented system. In particular, using the relational model[4] as an example videotex pages could be used for one or more attributes in a relation. For the real estate example, a relation could be defined in which the attributes are *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* = 3, *dining room type* = "formal" and $50000 \leqslant price \leqslant 70000$.

Rather than completely rewriting the videotex page management systems and the conventional record processing systems in order to merge the two, a distributed architecture similar to that depicted in Figure 7 may be beneficial. In the real estate example, 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 then be filled in by the user in some convenient notation and format (the use of Query-by-Example[17] or some other forms-oriented language may well lend itself to such specification), after which it is passed to the conventional data base system for processing. The response is then passed back to the videotex access machine, which either requests the target pages from the videotex page depository, requests more information from the user, or displays a dynamically constructed index page for further selection by the user. † Further discussion of this architecture can be found elsewhere.[2]

† This sort of capability already exists in many bibliographic retrieval systems as well as in Teletel/Star's electronic directory, for which the responses are displayed directly if they are few enough in number or else further information is requested from the user.

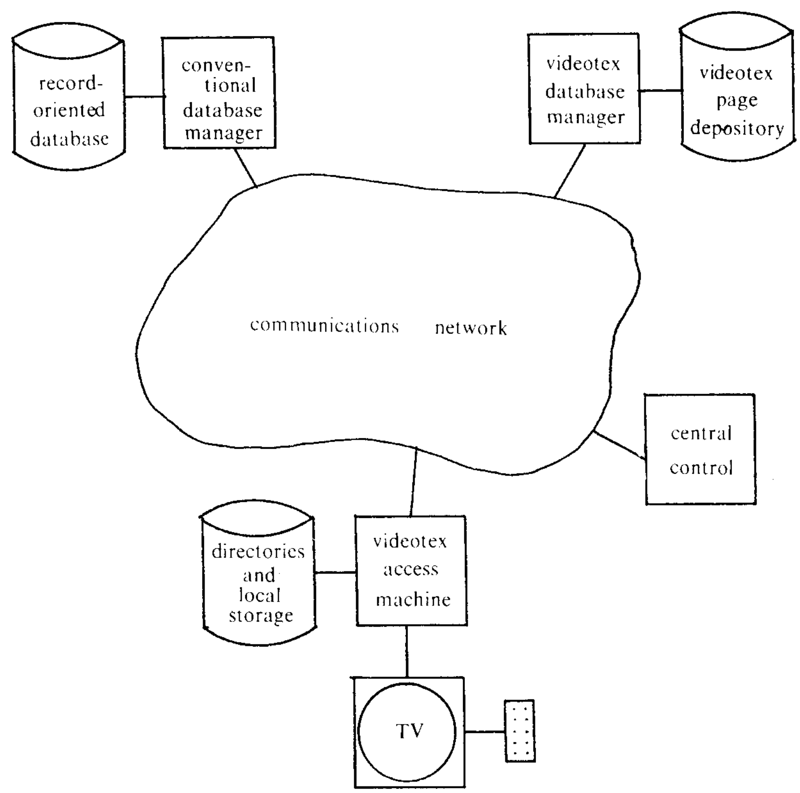


Figure 7: Schematic for an architecture combining videotex page management and conventional record processing

5. CONCLUSIONS

In this paper we have shown that the data structuring facilities available in current videotex systems are still relatively primitive and need more attention. The Teletel/Star experimental system has shown that simple concepts, the use of keywords and the association of an executable program with each page, can lead to far more sophisticated and user-oriented facilities. Whereas the provision of keywords does not expand the functionality of videotex systems, the program facility provides an opportunity for user interaction that goes beyond the simple selection of pages. Much more research and development is still warranted.

We feel that the experience gained by those who have dealt with conventional programming languages, graphics, operating systems, and data base systems should be applied to the future design of videotex systems. For example, we have proposed replacing the underlying tree structure by an array of pages and removing the visibility of page identifiers. In addition the marriage of page-oriented videotex technology with conventional record-oriented retrieval systems seems very promising.

Acknowledgments

Most of the material in this paper was collected or developed during the First Montréal Workshop on Videotex Technology sponsored by the Université de Montréal and the Canadian Natural Sciences and Engineering Research Council under grant G0363. We wish to acknowledge the contributions of the other attendees, especially A. Ball, B. Botten, D. Leahy, D. Le Moign, and A. Turpin, as well as the contributions of our colleagues who could not attend. We are particularly indepted to Daniel Le Moign for his careful explanation of Teletel/Star. The financial support of the Canadian Natural Sciences and Engineering Research Council under grants A9292 and G0363 and the Canadian Department of Communications under project 908-01 are appreciatively acknowledged.

References

1. M. Aysan, "Project IDA: home of the future," *Inside Videotex*, Infomart, Toronto, Canada, 1980, pp. 60-75.
2. A. S. Ball, G. V. Bochmann, and J. Gecsei, "Videotex networks," *IEEE Computer*, Vol. 13, No. 12, December 1980.
3. R. D. Bright, "The télématique program in France," *Viewdata 80*, Online Conf. Ltd., London, UK, 1980, pp. 19-24.
4. D. D. Chamberlin, "Relational data-base management systems," *ACM Computer Surveys*, Vol. 8, No. 1, March 1976, pp. 43-66.
5. J. J. Coyne, "OMNITEL™ – an integrated broadband distribution system for the eighties," Coyne Associates Systems Consultants, Ltd., Winnepeg, Canada, 1980, 15 pp.
6. D. Godfrey and D. F. Parkhill (eds.), *Gutenberg 2*, Press Porcepic, Toronto, Canada, 1979, 231 pp.
7. A. Henriot and J. Yclon, "Langage de programmation des bases de données Star," CCETT Note Technique RSI/41/443/79, Rennes, France, December 1979, 39 pp.
8. G. T. Hopkins, "Multinode communications on the Mitrenet," *Proc. of the Local Area Comm. Networks Symp.*, Mitre Corp. and NBS, May 1979, pp. 169-177.
9. E. Lee and S. Latrémouille, "Evaluation of tree-structured organization of information on Telidon," Telidon Behavioural Research I, Dept. of Communications, Ottawa, February 1980.
10. D. Le Moign, "Presentation de Star," CCETT Note Technique RSI/NT/23/17/80, Rennes, France, March 1980, 13 pp.
11. P. McFarland, "Videotex 1980: state of the art in Britain," *Inside Videotex*, Infomart, Toronto, Canada, 1980, pp. 20-24.
12. D. F. Parkhill, "Videotex 1980: state of the art in Canada," *Inside Videotex*, Infomart, Toronto, Canada, 1980, pp. 12-18.

13. D. A. Phillips, "Telidon and the human factors of videotex data bases," *Proc. of the Sixth Inter. Conf. on Very Large Data Bases*, IEEE, ACM, and CIPS, October 1980, pp. 330-331.

14. E. Sigel (ed.), *Videotex: The Coming Revolution in Home/Office Information Retrieval*, Knowledge Industries Publ., White Plains, N.Y., 1980, 154 pp.

15. K. H. Taylor, "On-line business databases and viewdata," *Proc. Online 79*, Online Conf. Ltd., London, UK, 1979, pp. 273-282.

16. K. Yasuda, "Conception of Captain system — background, experiment, and future plans," *Viewdata 80*, Online Conf. Ltd., London, UK, 1980, pp. 107-111.

17. M. M. Zloof, "Query-by-Example: a data base language," *IBM Systems J.*, Vol. 16, No. 4, 1977, pp. 324-343.