ALPHABETIC SEARCHING IN VIDEOTEX SYSTEMS

H.A. Maurer, W. Rauch, and I. Sebestyen
International Institute for Applied Systems Analysis, Laxenburg, Austria

RR-82-11
March 1982

Reprinted from *Electronic Publishing Review*, volume 1(3) (1981)

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
Laxenburg, Austria

Research Reports, which record research conducted at IIASA, are independently reviewed before publication. However, the views and opinions they express are not necessarily those of the Institute or the National Member Organizations that support it.

Reprinted with permission from *Electronic Publishing Review* 1:217–223, 1981.

FOREWORD

In 1981 the International Institute for Applied Systems Analysis began a program of research on the impacts of information technology. This work was planned as a cluster of related tasks, rather than a unitary whole; and, indeed, the various activities were intended to explore various possibilities, and therefore were not necessarily predicated on the same set of technological and societal assumptions.

One of these tasks dealt with the applications and social impacts of Viewdata (Videotex) systems – and the three authors of this report were the research team that carried out the work.

This is only one of a number of papers they have written. Too, its content intersects that of papers from another task concerned with computer-based messaging (or conferencing) systems. An appendix listing related publications appears at the end of this report.

The potential social impacts of both Viewdata and computer-based messaging systems are immense – the basis for the inquiry whose results are reported here.

ALEC M. LEE
Chairman
Management and Technology Area

Alphabetic searching in videotex systems

H.A. Maurer, W. Rauch and I. Sebestyen

Abstract: Of the four major types of interactive videotex systems currently being tested (Telidon, Télétel, Captains and Prestel-like) only one (Télétel) permits the use of alphabetic key words for searching. It is claimed that alphabetic keyword searching should be incorporated into future videotex systems. Methods of alphabetic keyword seaching in the absence of alphanumeric keyboards are then discussed. A novel technique is proposed which has been implemented recently on Prestel-like systems as an interim solution whenever genuine alphabetic searching is not available.

1. Introduction

This paper is only concerned with a minor aspect of interactive videotex systems. For a broader perspective of VTX the reader is referred to the literature, e.g. ref. [6].

1.1 *Alphanumerical searching (keyboard) versus numerical searching (keypad)*

The basic idea of videotex systems originated in the United Kingdom and is now known under its trademark Prestel. It involves 'upgrading' of colour TV equipment with some additional electronics that converts it into a simplified computer terminal, hooked up to a videotex center via a dialup phone line. To keep modifications of existing TV sets to a minimum, it appeared reasonable to use the standard remote control unit of the TV set as input keypad. However, the use of such a keypad with a fairly limited number of symbols leads to serious limitations, namely that, basically, only digits and a few special characters are available for input. For this reason, Prestel was originally designed as simple-minded information retrieval system (see below), ignoring the huge potential of making genuine interactive use of the videotex-center computer and the knowledge and experience in information retrieval systems already existing.

Following Britian's lead, other countries started to develop videotex systems too: the Canadians with improved graphic facilities (Telidon); the Japanese using a facsimile-type approach (Captains), forced on them by the complexity of Japanese letters; and the French (with Télétel and their Electronic Phone Directory) as the only ones to take a substantially different approach as far as keypad and searching techniques are concerned, by introducing alphanumeric keypads and full alphabetic searching. All major systems were still designed with purely numeric keypads and, consequently, rather rudimentary searching techniques.

It is our contention that an approach involving alphanumeric keypads such as the French one is most promising and should be pursued in the future. This is not only because keyword searching as such is important (as will be demonstrated in Section 2, it can be handled fairly efficiently even with purely

The authors are with the International Institute for Applied Systems Analysis, 2361 Laxenburg, Austria.

numeric keypads!) but because, without alphabetic input, videotex must remain a simple retrieval-only system, whereas with alphanumeric keypads it can turn into a versatile omnipresent multipurpose retrieval and transaction system. Properly equipped videotex terminals will become easy-to-use and (due to mass production) inexpensive computer terminals, for the mass consumer (business and residential) market, to be used for a broad variety of tasks [4] such as information retrieval, transactions (bookings, reservations, fund transfers), electronic mail, entertainment, education, personal and telecomputing etc.

To appreciate the difference between what Prestel or Prestel-like systems (as they are used in, e.g., Great Britain, Switzerland and Austria) can do and what videotex systems as outlined above could do, a *brief description of how Prestel† works is necessary*.

Information is stored on so-called pages or frames. Each page is identified by an integer number. Each frame can point to up to 11 further pages, to be reached by typing either #, or one of the digits 0 to 9 (see Table 1). A page with number n can be accessed by either typing $*n\#$ or, if currently a page m is displayed and m points via x to page n (where x is #, or 0,1,2,...9), by typing the symbol x.

Table 1. *Uses of Prestel-type keypad.*

*0#	Return to start
n#	Jump to page n
*#	Go back one page
*00	Repeat same page
**	Correct keying error

When a user dials up the videotex center, he obtains automatically page 0 of the system. This page (like most others) offers a number of choices to the user, a so-called 'menu'. By selecting the appropriate alternative and typing the corresponding symbol, a further page can be obtained, and so on.

†From the point of view of information retrieval, Prestel, Telidon and Captains are quite similar.

For the discussion in Section 2, it is important to understand two points:

(a) typing $*nm\#$ (where n and m are positive integers) does not (necessarily) result in the same page as typing $*n\#m$ (e.g. input *12# gives page number 12, but *1#2 can give a page with arbitrary number m, if page 1 is made to point to page m using digit 2).
(b) no page can point to more than 11 further pages, but if a page m permits a choice k, that choice can be keyed in before page m has completely appeared on the screen. In this case, the full page will never become visible. (Thus, suppose page m leads via four pages obtained by consecutive choices d_1, d_2, d_3, d_4 to a page n, then typing d_1, d_2, d_3, d_4 in rapid succession will immediately give page n. The intermediate pages will not be shown, and no time for building them up will be consumed.)

The significance of the above rather technical points will only become apparent in Section 2. However, from what has been explained it should be clear that the search for information based only on the menu approach and numerical input will sometimes be rather cumbersome. The possibility of typing-in a keyword x if information on an item x is desired is certainly an attractive feature (and currently only available in Télétel). Beyond that, an alphanumeric keyboard is essential for using videotex systems for sending messages, for ordering from some catalogues offered on videotex, for carrying out a reasonable dialog in a teaching application of videotex, for being able to use videotex as a terminal for simple programming tasks etc. Not equipping videotex terminals with alphanumeric keyboards reduces the value of videotex tremendously and permits only rudimentary services. The fact that Prestel in Britain, after two years of intensive campaigns by the British Post Office, has attracted only some 10,000 users is party due, in our opinion, to the absence of alphanumeric keyboards and the possibilities they offer, and to the lack of real interaction possible in the British system.

In the past a number of arguments have been put forward against the use of alphabetic

"Most videotex users today are from a commercial environment."

keypads. One such argument is the higher price for such somewhat more sophisticated keypads. Observing that most videotex users today are from a commercial environment, the very small (and only initially existing) price difference seems insignificant when compared with the loss of applications otherwise incurred. Another argument often voiced is the claimed increased complexity of using an alphanumeric board. We do not believe in this line of thought. Indeed, noting the circuitous ways which are used to try to overcome the lack of alphanumeric keys the impression is quite the opposite; the lack of alphanumeric keys makes using videotex more complicated, not easier. However, the design of the keyboard may have to be modified from the QWERTY arrangement of letters found on usual typewriters to an arrangement easier to use for the naive (= nontyping) use. This is done, for instance, on the French terminals, where letters are arranged in alphabetic order. Yet another argument against alphanumeric keypads which is sometimes mentioned is the size and weight of the keypad. Observing that current remote control units of modern TV sets have up to 25 keys, and noting that hand-held boards with 20 'genuine' keys (to be operated with the pointing figure of the right hand) and 3 'escape' keys (to be operated with fingers of the left hand while holding the keypad) allow a total of 80 characters (i.e. small and capital letters, digits and 18 special symbols) do exist (e.g. Dynaflex), it is clear that this argument is not valid either.

Summarizing, alphabetic keypads are of crucial importance for videotex systems, both for searching and other applications, and there are no good reasons why they should not be used in the future instead of purely numerical keypads.

However, as long as only numeric keypads are available, a kind of 'pseudo-alphabetic search', as explained in Section 2 may alleviate the problem of having no letters available to some extent.

2. Alphabetic searching with numeric keyboards

2.1 *The basic idea*

As has been explained in Section 1, searching for information based on a given keyword is an important asset to the user of a videotex system. Despite the fact that many videotex systems are currently only offering numeric keypads the above mentioned problem is recognized and admitted by providing an 'alphabetic index' using a menu-type search in all of them. The user interested in a certain keyword selects one of a number of choices depending on the first one or two letters of this keyword (e.g. *Aa-Al* ... 00, *Ag-Ba* ... 01 etc.), and repeats the process with the next few letters until the desired entry is located.

This process of 'alphabetic searching by narrowing down' turns out to be a reasonably cumbersome process if the alphabetic index at issue is of any size at all. The user has to refocus his attention a number of times between keypad and screen, and each time has to wait

"The design of the keyboard may have to be modified from the QWERTY arrangement of letters found on usual typewriters to an arrangement easier to use for the naive (= nontyping) user."

"It is our contention, and the main point of this paper, that even with current systems and purely numeric keypads a more clever way of organizing an alphabetic search is possible."

for the screen to be filled (typically 5–10 seconds) and to enter the appropriate choice.

Since Prestel-like systems and Telidon do not support any searching beyond the menu technique, it is generally agreed that this is evidently the only way to perform an alphabetic search.

It is our contention, and the main point of this paper, that even with current systems and purely numeric keypads a more clever way of organizing an alphabetic search is possible. The basic idea is to associate with each of the digits 1,2,...,9 a group of letters. By typing in a string of letters, a string of digits is obtained (in fact, creating a hash-code for each string of letters) which, by organizing the data appropriately, leads to the desired information.

More specifically, we associate the letters *A,B,C* to the digit 1, the letters *D,E,F* to the digit 2, ..., the letters *YZ* and the symbol '.' (period) to the digit 9 (most conveniently by putting little stickers with the appropriate letters on or below the numeric keys on the keypad) (see Table 2). In this fashion, we associate with each string of symbols w composed of letters and the period-symbol a sequence of digits which we denote by $d(w)$.

Suppose that the set of keywords used for some database is W and that each key $w \in W$ has a number of frames of information associated with it, the first of which we denote by $f(w)$. We choose a page with number n as start-page of our 'pseudo-alphabetic index' (as we call it henceforth), i.e. keying in $*n\#$ results in the display of that start-page. The main idea is to assign the frame $f(w)$ to the page whose number is obtained by typing $*n\#w$. (However, because of the way we have associated letters and digits, this, of course, amounts to inputting $*n\#d(w)$.)

It should be clear that searching using such a pseudo-alphabetic index is exactly the same as searching an ordinary alphabetic index, providing 'everything works out'. Evidently, there are a number of problems which may arise and have to be taken care of. As we demonstrate below this is easily done in each case and does not lead to serious obstacles in using the proposed technique.

Table 2. *Proposed assignment of letters to digits on videotex keypads.*

1	=	*A,B,C*
2	=	*D,E,F*
3	=	*G,H,I*
4	=	*J,K,L*
5	=	*M,N,O*
6	=	*P,Q,R*
7	=	*S,T,U*
8	=	*V,W,X*
9	=	*Y,Z*

2.2 *Problems*

Problem 1: Different keywords with same hash-code

It is clearly possible that different words w and w' (of same length) yield the same hash code $d(w = d(w')$. This problem is easily solved by using the page accessed by $*n\#w$ ($\subset *n\#w'$) not to store the frame $f(w)$ or $f(w')$, but to store a frame offering two choices leading to either $f(w)$ or $f(w')$.

Problem 2: Input of a word which is not a valid keyword

If the user types $*n\#z$, where z is a word not in W, one of two things may occur:

(a) if a word $w \in W$ with $d(w) = d(z)$ exists the

user is led to information for the keyword w. To avoid difficulties with respect to this each first frame $f(w)$ associated with a word w should contain w in a clearly visible fashion. In this way, a user who keys in $*n\#z$ and obtains information concerning a word $w \neq z$ realizes that $z \notin W$.

(b) If no word $w \in W$ with $d(w) = d(z)$ exists, inputting $*n\#z$ would ordinarily give a systems message 'page non existent'. If desired, this can be replaced by a special frame with a more elaborate message such as 'no keyword with the code $d(x)$ exists. To return to startpage of index press 0, to ...' To access such a special frame F the pointers must be organized as follows: let $z = uav$, where u is the longest prefix of z such that $d(u)$ is the prefix of some $d(z)$, $z \in W$, and where a is a single letter. By definition of u, $d(ua)$ does not occur as prefix of any $d(z)$, $z \in W$. Hence the choice $d(a)$ on the frame $*n\#d(u)$ can be used to point to F.

Problem 3: Unnecessarily long input

Suppose $z = ua_1a_2 \ldots a_m \in W$ (where a_i are individual letters for $1 \leq i \leq m$, $m \geq 1$) is a word such that u is as short as possible and $d(u)$ does not occur as prefix of any $d(z)$, $z \in W$. Thus, when the user has typed in $*n\#u$ it is already clear that he intends to input $*u\#z$ and should be saved the trouble to type the remaining m symbols. Hence, it is reasonable to associate the frame $f(z)$ already with $*n\#u$ rather than with $*u\#z$. Without further adjustments this, however, is not a good solution; the user who types in rapidly $ua_1 \ldots a_i (i \geq 1)$ will get a message 'page not existent', and will get the wrong impression that $z \in W$. To avoid this dilemma and yet to permit access by short prefixes it is sufficient to ensure that the page $*n\#u$ points to itself with each of the digits, 1,2,...,9(!).

Problem 4: The prefix problem

This problem only arises due to the fact that in Prestel-like systems only one digit is processed at a given time. Thus, if two keywords w and z (but z longer than w) have the property that $d(w) = d(z')$, where $z = w'z'$, then slowly keying in $*n\#z$ yields, on the way to the desired information, the page $*n\#w$ representing $f(w)$. This is quite apt to confuse the user. To avoid such confusion it is probably preferable to insert an extra frame F as the page $*n\#w$ which offers the choice to either reach $f(w)$ or else to continue to complete the desired keyword by typing in further letters.

Problem 5: Long prefixes

A number of different keywords $w_1, w_2, \ldots, w_t$ may have the property that $d(w_1), d(w_2), \ldots, d(w_t)$ have a long common prefix u. To shorten the input process for the user, one may choose the shortest v, such that v is prefix of $d(w_1), d(w_2), \ldots, d(w_t)$ (but of no other keyword $w \in W$) and one could insert an extra frame F as the page $*n\#v$ which allows t choices leading to $f(w_1), f(w_2), \ldots, f(w_t)$, respectively. To prevent the user from going beyond the page $*n\#v$ that page should point to itself for every digit 1,2,...,9.

Problem 6: Probability of collision

In above fashion, a purely numeric keypad and a system designed only for numeric menu choices can be used for alphabetic searching by typing in the keywords almost as if a full alphabetic search facility were available. Of the five problems mentioned above the last three (and problem 2 to some extent) arise in any alphabetic keyword system. Only problem 1 arises merely because of the hashing technique proposed. Although 'collisions' are easily resolved as explained, the user might well find it annoying that (rather than getting the desired information directly) he is forced to make one additional choice at the end. Hence it is important to have some feeling for how often a 'collision' (different keywords w, w' with same code $d(w) = d(w')$ will occur.

This is a classical problem from the theory of hashing and data structures, see for example, ref. [3], a uniformly distributing hashing function will give roughly alpha collisions per key, where alpha is the loading factor, and

will give alpha collisions in total. Assuming that we consider words of length five only (for a rough estimate) we have an address space of 9^5, i.e. roughly 60,000. For a list of, for example, 240 keywords, this gives a loading factor $\alpha = 240/60{,}000 = 0.004$. The probability of a collision is thus less than 0.5%.

This is still a fairly pessimistic estimate since only short words were considered. The address space for actual English words is somewhat higher, resulting in a still smaller probability of collision.

A first experiment with such a pseudo-alphabetic index was carried out in the Austrian videotex pilot-trial, see, for instance, ref. [4] [MHA2]. Information on 260 Styrian† towns and villages was prepared in this fashion. No single collision occurred, in good agreement with above calculations. Experimental results with untrained personnel showed that the time required to find a specific keyword with the pseudo-alphabetic index is about half of that when using the narrowing-down approach.

A final remark concerning the above-mentioned Austrian experiment may be of interest. As start-page of the pseudo-alphabetic index the number 35228 was chosen, since d(INDEX) = 35278. Thus, in the Austrian videotex trial, typing *INDEX#z, where z is the name of any Styrian town or village will give information on that location.

2.3 *Conclusion*

The pseudo-alphabetic index is a possible 'clutch' (to be used only as long as necessary), but not a substitute for a full alphabetic keypad and for genuine alphabetic searching, which, as outlined above, seems to be essential. Not only is it fairly hard to manually structure the data correctly (hence requiring software for that purpose [1]), but also it does not allow real alphabetic input for, for example, messages, despite the fact that it can be used in this fashion in a somewhat cumbersome way: to type a letter one first lists the corresponding key (i.e. 1 for *A,B,C*, 2 for *D,E,F* etc.) and then 1,2 or 3 depending on whether the desired letter is the first, second or third on the key just used. Thus, *A* would be encoded as 11, *B* as 12, *C* as 13, *D* as 21 and so on; the end of words is indicated by typing a zero.

As clumsy as the method sounds, if keys are equipped with the appropriate lettering, both encoding and decoding is possible directly (without any memorizing or pencil and paper). It is being used for sending messages in the Austrian VTX pilot trial (e.g. allowing people to register for certain events) and is used in the field trial in the BRD in a still somewhat clumsy version for sending 'number letters' to, say, Axel Springer Pub. Co. [5].

3. Indexing versus search-trees

Even if this method of 'numerical coded alphabet' turns out to be of limited practical value (the way we actually hope, if the general trend should be towards alphanumeric keyboards) it has some conceptual aspects for videotex, since it bridges the gap between the two main information-access schemes, namely search-trees and indexing.

In the case of search trees, all accessible elements are organized into sets and subsets of increasing specificity and are ordered into a 'tree-structure'. Examples are Dewey's Decimal Classification which covers the whole universe on such principles. Less sophisticated systems are familiar to us from most organizational schemes; the videotex access trees are one of the most recent applications. A search tree is simple to understand, to construct and almost self-explanatory in use. It is suitable for specific applications as well as for global attempts.

However, there are basic limitations to the system of search trees: the decisions taken by building up the tree are irreversible and restricted to only one dimension; the structure of the tree is rigid and cannot be adjusted to changes in the systems environment; browsing through the system is almost impossible; in complex cases, a high number of branches

†Styria is one of the provinces of Austria.

has to passed before ending up at the 'leaves' of the trees.

Recent information retrieval systems therefore use a different approach for accessing information: indexing in this method, brief descriptions of data are themselves organized in a file. The elements (= data) may be indexed with a varying number of descriptors and according to different aspects; the index-terms may be chosen 'free' or from a controlled vocabulary (= thesaurus); for retrieval purposes they can be arranged according to the rules of Boolean logic. With the use of computer technology even very large index files (at present up to ten billion words) [2] can be organized as inverted databases; information science invented very 'sharp' retrieval instruments operating such inverted index files (including natural language access).

The method of 'numeric coded alphabet' for videotex systems introduced combines elements of search-trees as well as of indexing. The mode of access strictly follows a decimal tree and is therefore fully suitable for videotex logic. But this tree structure follows the digital construction of the alphabet, not the structure of the reference material. So, even if this method at the first glance shows characteristic elements of a tree-structured information-retrieval system, it definitely is of index-type, since it operates on inverted files.

With the combination of a tree-structured access path and an inverted index file, the described method opens the possibility of an index-oriented retrieval approach for videotex for the system user, even without alphanumeric keyboards. To make all other advantages of index-oriented information retrieval systems available for videotex systems, it will become necessary to introduce an appropriate software for the information supplier function of the system. This could be realized by using videotex as gateways for external computer capacity already equipped with such possibilities. In that way, videotex can be developed from simple information distribution networks into sophisticated information retrieval systems.

References

[1] F. Aurenhammer: 'Bildschirmtext Alternativindex'. Diplomarbeit, Institut für Informationsverarbeitung, TU Graz, 1981.

[2] C. Burns: 'Information storage and display', *J. Amer. Soc. Information Sci.*, 1981, **32** (2), p. 145.

[3] H.A. Maurer: *Datenstrukturen und Programmirungsverfahren*. Teubner, Stuttgart, 1974.

[4] H.A. Maurer: *Bildschirmtextaehnliche Systeme*. Studie für des BM fFu W, 1981.

[5] J. Hoefele: *Bildschirmtext — der Kurze schnelle Weg zum Leser*. Bildschirmtext Seminar, Schloss Laxenburg, 1981.

[6] R. Woolfe: *Videotex*. Heyden, London, 1980.

RELATED PUBLICATIONS

Maurer, H., W. Rauch, and I. Sebestyen (1981) Videotex message sending systems. Electronic Publishing Review 1(3):267–296.

Sebestyen, I. (1982) The videodisc revolution. To appear in Electronic Publishing Review.

Maurer, H. and I. Sebestyen (1982) Unorthodox videotex applications: teleplaying, telegambling, telesoftware and telecomputing. To appear in Information Services & Use.