UNIVERSITÉ DE MONTRÉAL

Towards Distributed Processing in Videotex

by

Jan Gecsei

PUBLICATION 447

DÉPARTEMENT D'INFORMATIQUE
ET DE RECHERCHE OPÉRATIONNELLE

Faculté des arts et des sciences
Université de Montréal
C.P. 6128, Succursale "A"
Montréal, P.Q.
H3C 3J7

Towards Distributed Processing in Videotex

by

Jan Gecsei

PUBLICATION 447

This report is prepared under contract 2 OSU.36100-1-0169 for the Departement of Communications, Canada.

The opinions expressed in this report are those of the author. They do not necessarily imply any position of the Department of communications.

Département d'informatique et de recherche opérationnelle
Université de Montréal
Août 1982

1. Intelligent terminals

In the opinion of many, videotex has not quite fulfilled the expectations characteristic for its initial development. This is exemplified by Prestel [Arnold, 1981], where the actual number of users (around 20,000) is far below the projected numbers. Also, one hears all too often statements such as "videotex is a technology in search of applications", etc. Part of the problem seems to be the familiar chicken and egg phenomenon: people will not buy videotex terminals before a sufficient varienty of useful and amusing applications are available, and service providers will not venture into providing such applications before a sufficient number of customers are in sight.

Another factor is that technical characteristics imposed by early systems and inherited by many later models (frequently mentioned examples are the numerical keypad, access by menu selection and limited interactivity) effectively hamper and complicate the implementation of more imaginative services. For example, consider the awkward ways in which selection from alphabetical system indexes must be done with numerical keypads, and compare it with the simplicity of keyboard operation. The same applies to keywords, which can be entered through keypads (in a manner similar to using the letters accompanying numbers on most telephone dials), but at the price of unnecessary ambiguities and other complications. Note that these difficulties cannot be uniquely blamed on insufficient hardware, but they are also anchored in the system software which is often difficult or impossible to change.

There are many ways to improve the functionality of videotex, and thus preparing the ground for breaking the chicken and egg vicious circle. According to the Telidon philosophy the key is in improved graphic capabilities. While geometric graphics are certainly superior to CEPT-like mosaics in an absolute sense, it is far from certain that they alone are enough to make videotex fly. Another argument goes that most useful information on videotex is in character form; graphics are a useful complement, but the possibilities offered by mosaics are sufficient for that purpose. This is the view of most European administrations which consider accessibility to external applications through gateways more important than better graphics.

A third way towards improvement lies in gifting the terminal controllers with more intelligence. More precisely, one can open up the resources already present in most modern terminals for uses other than just interpretation of user commands and incoming character strings, namely for telesoftware and local programmability.

The general trend, pushed by rapid progress of fabrication technology, is that videotex systems are moving towards public computer networks, where subscribers can choose from a large number of available services. There is no reason (except for cost, discussed below), why such videotex networks shouldn't benefit from notions of functional distribution, well known from other systems [Bochmann, 1979].

Intelligent videotex terminals come by in three ways. (a) Personal computers such as Apple can be fitted with hard or soft modules to interface with videotex. (b) Personal or small business computers designed integrally with videotex capability (such as Nabu), and (c) augmented terminal controllers such as mentioned above. A significant example of this approach is MUPID (Multipurpose Universally Programmable Intelligent Decoder) [Maurer, 1982] developed in Austria and available for lease to videotex subscribers in that country.

Before discussing in more detail the functional possibilities offered by such terminals, let us compare briefly the above three alternatives from and economic and psychological viewpoint. While personal computers under (a) and (b) certainly offer much functionality (maybe too much for the average user), their cost ($2 000-15 000) exceeds several times the cost of videotex decoders (in the hundreds of dollars). This puts an a priori limit on the number of potential customers. Further, the buyer of a personal <u>computer</u> is buying a computer, and not just an enhancement to his television as in case (c); this psychological barrier should not be underestimated. Back to cost considerations: augmented terminal controllers cost less because

- they are based on a volume - fabricated product

- offer less function than a t~~ro~~pical personal computer (but just enough for the purpose).

Further, as mentioned above (and by many others before), modern decoders already contain memory and microprocessors which are the essential components of a personal computer. Therefore the differential cost is (roughly) the price of the keyboard ($20-30) and additional memory (say 32kB @ $4/kB = $128). This argument is especially valid for augmenting Telidon - like terminals which already incorporate large amounts of memory; therefore the above cost differential of about $~~150~~ 200 would be relatively smaller than with cheaper mosaic terminals. Remark that intelligent terminals, being neither personal computers nor standard videotex decoders, fully qualify as "special purpose hardware".

2. Distributed processing

It is commonly recognized that videotex is now evolving towards public computer networks and there is no reason why they should not be treated as such (i.e. computer networks) and why the principles, lessons and experience learned from such distributed systems should not be applicable to videotex. Chances are that videotex systems will become even frontline computer networks, due to unique characteristies like (a) extremely large numbers of users (in nationwide systems); (b) the unifying influence of a presentation protocol (whichever from the many existing) and of videotex databases (as opposed to external) which can be used as a common repository of information and <u>also</u> as a communication medium; and (d) the possibility to link up with external computers through well-defined procedures and standards (gateways).

What is clearly missing so far in this picture, and what is commonly present in many other networks, is terminal intelligence (take for example office automation systems). We recall the cost argument from the preceding section which puts videotex in an advantageous situation compared to other systems where typically the point of departure towards intelligence are ordinary VDU's.

A network with (say) a service computer and a number of programmable terminals can be also likened to a <u>computer hierarchy</u> where the highest level (terminal) is capable of autonomous execution of some simple tasks, but for others it has to rely on life support provided by the service computer (such as storage of programs and data, gateway functions, etc.)

One can envisage further levels in this hierarchy in the form of hand-held keyboard/calculators, and/or deeper levels of service computers.

We cannot discuss here ~~his~~ in further detail this way of organizing videotex systems; however, two points must be mentioned: first, it provides a natural solution to the much debated problem of overloading the service computer by large numbers of simultaneous users. Second, it reconciles and combines the advantages of data processing based on time sharing and stand-alone small systems.

3. Telesoftware

Telesoftware should play a central role in the design and systematic application intelligent terminals in videotex networks.

Telesoftware has received so far much less attention than its potential would merit; this statement is heard quite often, but few seem to go further than that. Remark that telesoftware is not even mentioned in Chapter 8 (Technology forecast for User Terminals) in a new and well informed book [Tydeman, 1982]. In my opinion, most telesoftware projects reported to date are rather ad hoc applications, without aspiration to fully employ all potential of this technique. Examples are the CET (Council of Educational Technology) project [Thompson, 1981] in England and the Playcable system in the U.S. The former merely uses Prestel as a storage and delevery medium for educational software. Once the programs are down-loaded, they are executed on a variety of computers in presentation formats generally incompatible with Prestel. Similar "software banks" are found in home computer (or terminal) networks such as the Source. Playcable is an example of a "closed" system, where the language, hardware, transmission system and applications are under the strict control ofthe company and can be used only in a limited number of ways. Cooperation with other systems is practically impossible.

One should mention in this context the BBC microcomputer [Moir, 1982] aimed to be sold to more serious viewers of a series of programs on computers on BBC Television. This microcomputer system is supposed to be able to receive telesoftware via Prestel or Ceefax.

What is needed in order to exploit all possibilities of telesoftware is an <u>integrated network architecture</u> (to be developed in the OSI context) that provides a convenient framework for service development (using telesoftware as a tool), and not just individual applications.

Such a system with intelligent terminals would have a unique characteristic: the programmability of some nodes by others in order to enable meaningful and flexible (reprogrammable) cooperation between nodes of the system. Note that this is a far cry from simply "downloading" programs to a terminal and then disconnecting from the service computer, as in the case of most presently known applications. While the generalized distributed operation may seem weird and unapplicable to videotex in its full generality (cf. the "targeting" capability of the ADA language), it is quite applicable in a more restricted framework of videotex applications. An already implemented example is the simulation by means of telesoftware of keyword search on Prestel (its central software having only numerical selections and an alphabetic system index). Other applications will be given later in this report. In most cases, the complexities of telesoftware loading, initiation, etc. are completely hidden to the user, who doesn't even know whether he is dealing with the terminal or the service computer.

Networks with processing capability as described above will be called Distributed Processing Videotex Networks (DPVN).

4. Functional characteristics of DPVN

The following brief list is a first approach to what might be a reasonable starting point to the design of a DPVN.

a) Terminals capable of executing telesoftware written in a general purpose programming language.

b) Programmability of the terminal (preferably) in the same language.

c) Programs and displayable data can come to a terminal from any source (database, external computer EC, local user or remote user).

d) More generally, files (programs and data) can be moved between any nodes of the system.

e) Programs can communicate with other programs (in the database, EC) and with local and remote users.

f) Consequently, provision must be made for system-wide addressability and naming of objects such as programs, displayable files and terminals.

The outstanding characteristic of such DPVN is that its components can dynamically and automatically adapt (by receiving and executing appropriate programs) to new situations and applications. An application can be seen as the cooperative execution of several programs at several sites preceded (and possibly interleaved) by the loading of appropriate programs. All this is happening (typically) automatically and invisibly to the user (but under the implicit control of his choices and commands). To refer again to the example of simulating keyword access in a Prestel-like system, the user has simply to type a keyword (after selecting that mode of access) and wait until the appropriate information is displayed to him. The program to do the tedious search in the alphabetic index is loaded and initiated without his knowledge.

The most obvious place where such cooperative execution takes place in a DPVN is the terminal-service computer pair. But it would be shortsighted to exclude from the design other possibilities such as cooperation between terminal - service computer - external computer (gateway), between groups of terminals and a service computer (conferencing) or between service computers.

Thus the design of DPVN fully capable of exploiting intelligent terminals could benefit from the experience from other advanced distributed database or transactional systems. In particular, a number of technical issues have to be resolved:

1. The language problem: which language to use for telesoftware and for system implementation? There is no agreement in sight for an eventual telesoftware standard; proposals range from different machine languages through Basic and Pascal up to Cobol. The solution might be a

"metastandard" consisting in a standard international registration procedure for different languages similarly to the registration of different character sets by ISO. Loading a telesoftware program would be preceded by and option negotiation phase.

On a different level, one will have to provide for a set of tools (in effect an implementation language) for the service providers in order to easily apply the facilities offered by DPVN.

2. Reliable methods for transmission of programs. Evidently, the requirements here are more stringent than for the transmission of displayable information. A transport protocol (as defined in OSI) is needed for that purpose, possibly for combined use with a messaging service.

3. Naming and adressing conventions.

4. Control the ownership of telesoftware and other files, and the right to modify them.

5. Where lies the control for a given application? Is there always need for a unique locus of control? (E.q. in the Bildschirmtext/Prestel Gateway, the control is in the service computer).

5. Applications

Within the above outlined framework, many classes of applications would be possible. Some of them are (refer to Fig. 1 for the data paths involved):

- Classical videotex data retrieval (path a).
- Classical telesoftware: downloading of programs and local execution after disconnection from the source (b and then c).
- Local processing of displayable material (images) by telesoftware or locally written programs (d). Example: automatic animation. Note that in such cases telesoftware programs need not be separate entities stored on distinct pages or files in the service computer. Strings of displayable data and instructions to manipulate them can be freely mixed (separated by control characters). This feature is present in embryonic form already in the PDI concept; PDI code <u>is</u> a form of telesoftware. PMI's (Picture Manipulation Instructions), often mentioned as a possible extension to PDI's seem so be designed for a similar purpose.

  Telesoftware designed to interpret different presentation protocols would fall in this class of applications.
- Image manipulation under user control (that is, by a (tele-)program interacting with the user); examples: games, CAI (b, then c and d).
- Interaction between a program in the terminal controller

and <u>both</u> the user and service computer; examples are various enhancements and automations of database search procedures, as in the example in sec. 3. In such interfacing applications, terminals are often called <u>user agents</u> (b and c).

- Interaction between user terminals (through messaging or virtual circuit mechanism); ex. games, conferencing, electronic mail (c, b, e or c, f).
- Sending files (data or programs) to distant locations for storage, eventually execution (b, e, g). This amounts to what might be called "reverse telesoftware".

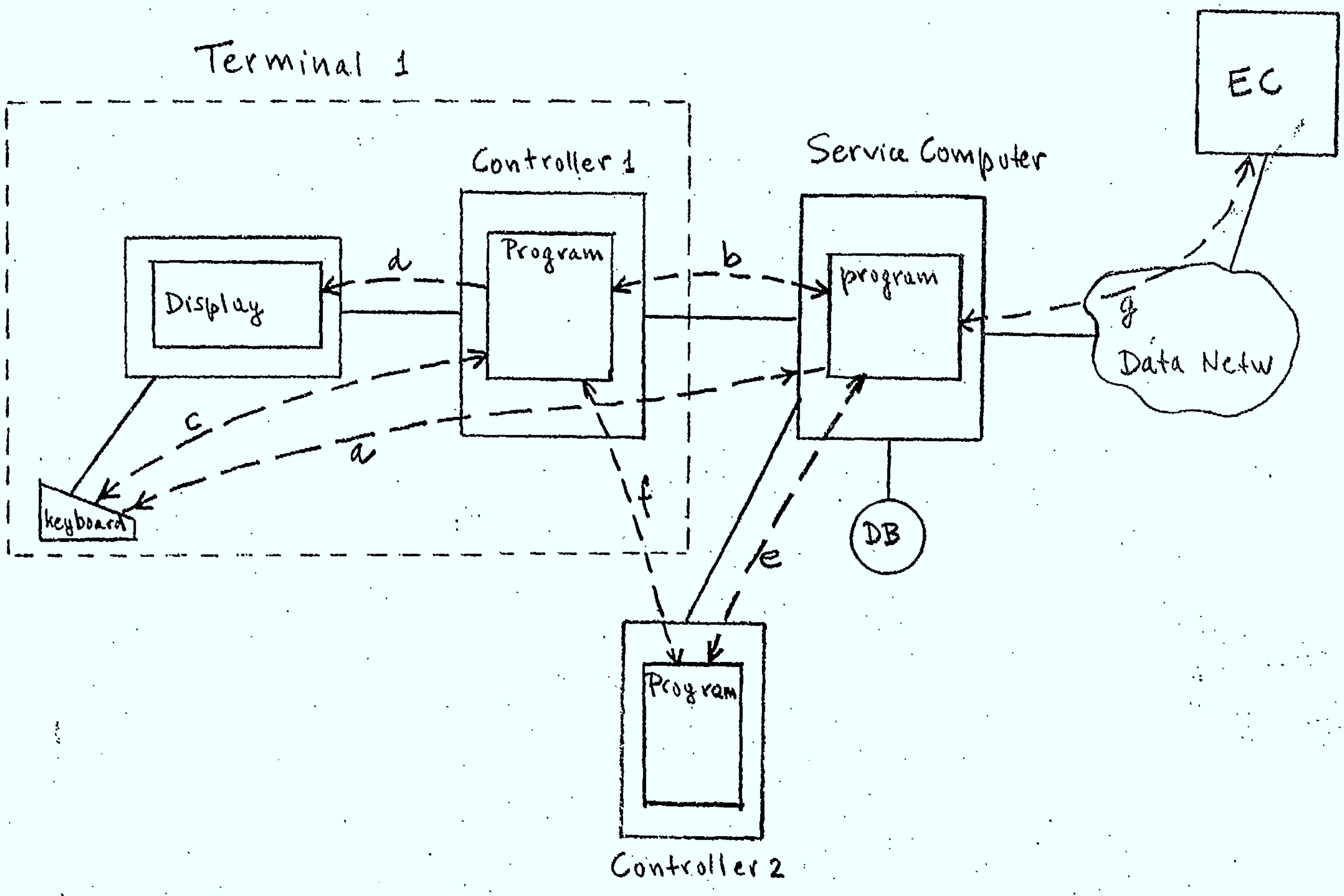

Figure 1.

Finally we include an illustrative list of concrete applications possible in a DPVN environment with intelligent terminals and telesoftware.

- Complex keyword search (querying) into videotex or other databases, including scanning the text (e.g. news stories) for given arguments. This is especially applicable to teletext.
- Text and graphic editing (e.g. for messaging, conferencing or construction of private databases).

- All variants of game playing (user-terminal, user-service computer, user-user with the terminal executing bookkeeping and arbitration tasks, group games).
- Program development in a variety of languages (with telesoftware compilers or interpreters).
- Educational applications.
- Accomodation of terminals to changes in presentation standards.
- Personal filing of data and programs in private pages of a public database. This makes it possible for the terminal to function without local secondary storage, contributing further to the economic attractivity of DPVN.

  From the point of view of such applications the reverse baud rate of 75b/s in some (mostly European) systems is a serious drawback. Current 1200/1200b/s transmission provides and acceptable minimum, while future broadband ISDN-like systems [Handler, 1982] will permit more breathing space for all applications, as frequently discussed in the literature.
- Data collection and form filling assisted by telesoftware. In this context we remark that "form management" [Tsichritzis 1982] seems to have a central role in office automation, one of the potential applications of videotex.

6. Summary and Conclusions

1. The use of intelligent terminals points to an economically and functionally very attractive way to upgrade videotext (and maybe to break the chicken and egg vicious circle hampering its development).

2. Telesoftware and local programmability (but not necessarily local mass storage) are the key elements in this process. the incremental cost for the additional functions is around $1~~0~~50-200.

3. In order to draw full benefit from the new terminal features, a videotex system should be designed as a Distributed Processing Videotex Network (DPVN). Its main characteristic is the cooperative execution of concurrent programs (e.g. in the terminal and service machine, or in several user terminals).

4. An important task is the systematic design of such a DPVN, drawing upon the experience from similar systems. This will assure that the necessary protocols and standards will be identified and that new applications with telesoftware will not be ad hoc as is presently the case. In this context one should keep an eye on the development of the Architel system in France, aiming at a unified architecture and interworking for all telematique and similar applications to be developed in the OSI (Open Systems Interworking) context.

5. The most important technical problems to be resolved are schemes for naming and adressing objects (programs, files, terminals etc.) on a systemwide scale, enabling orderly communication between objects.

6. DPVN should be designed in such a way that intelligent terminals (a suitable name might be Intelidon) and new applicatives they caser for are a superset of a standard system. Intelidon is a top-of-the-line terminal.

7. For those to whom the above proposals might seem far-fetched, complicated or expensive, we recommend to read [Maurer 1982]. In fact, most of the sample applications listed in the Report (and many others) have already been implemented, an intelligent terminal is available for $7,00/month (!), and Austria might be well on its way to have the first DPVN.

<u>References</u>

Arnold, E.: "Prestel-a Lemon in the Living Room?", Viewdata '81, p. 537.

Bochmann, G.v.: "Architecture of Distributed Computer Systems", Springer Verlag Lecture Notes, 1979.

Handler, G.J., "Networking - Bit by Bit", Videotex '82, p. 453.

Maurer, H.A.: "Will Mupid revolutionize Austria's Videotex?" Videotex'82, p. 187.

Moir, C.: "BBC Micro", Practical Computing, Jan. 1982.

Thompson, V.: "Educational Trial of Prestel", Viewdata'81, p. 303.

Tsichritzis, D.: "Form Management", CACM, July 1982.

Tydeman, J., et al. : "Teletex and Videotex in the United States", McGraw Hill, 1982.