Telesoftware: adding intelligence to teletext

J. Hedger and R. Eason

Indexing terms: *Television applications, Data-communication systems, Computer software*

Abstract: The authors seek to show how, by using a microprocessor, teletext may be extended to provide many new and useful facilities, while remaining compatible with existing teletext transmissions. The design and implementation of an experimental telesoftware terminal are described, and a summary of current thinking is given, indicating the probable future software and hardware constraints of the system and its potential applications.

1 Introduction

Independent Television has been broadcasting Oracle as a service of broadcast teletext since 1975.[1] It was soon realised that teletext, as well as being an efficient means of disseminating textual and simple graphic data, offered a very convenient and cost-effective method of distributing software programs for home computer systems. Furthermore, an existing teletext decoder already contains many of the 'building blocks' necessary for a simple home computer, such as memory, colour visual display and numeric data entry. With the addition of a microprocessor, such a decoder may be transformed into a small, but potentially powerful, personal computer inside the television set itself.

By utilising the Oracle system to broadcast programs for such an enhanced teletext decoder (with the programs presented as pages of standard text), a user would need only to select the pages containing his desired program. Once received, this software could then be executed by the microprocessor, utilising the existing colour display and keypad, without the need for expensive storage peripherals or telephone lines.

Thus telesoftware[2] represents a very inexpensive form of personal computing, since once a user has purchased the necessary 'intelligent' television set, telesoftware, like teletext, is a free service.

2 An experimental terminal for telesoftware

To stimulate the interest of decoder manufacturers and broadcasters in the concept of telesoftware, Independent Television and Mullard Ltd. embarked, towards the end of 1977, on a joint project to design and construct a simple telesoftware terminal for evaluation, using the Oracle service to broadcast the trial software. The terminal hardware was built at the Mullard Applications Laboratory, Mitcham. The aim of the project was to produce a prototype terminal in a relatively short space of time, and so the design was intentionally made somewhat simple, far from optimum and not intended in any way to represent a definitive means of implementation. However, this simplicity was far outweighed by the subsequent value of the terminal for demonstration purposes.

2.1 Compatibility with teletext

At this early stage there existed no specification or guidelines for the broadcasting of telesoftware, nor for its implementation in hardware. It was realised therefore that parameters governing the experimental terminal, although fairly arbitrary in their choice, could not be permitted to interfere in any way with the existing teletext system. For this reason, design work was based entirely upon the use of normal teletext transmission techniques, using ordinary characters to represent bytes of computer software.

2.2 Overall design

The prototype terminal itself consists of three main parts: a domestic colour television receiver with ultrasonic remote-control; a teletext decoder card and purpose-built microprocessor card. For convenience, the two latter items were mounted outside the receiver together with their power supplies; although ideally the whole terminal would be housed within the receiver cabinet. Fig. 1 shows a block diagram of the experimental terminal. For clarity, only the bus lines and major interconnections are shown.

The teletext card contains a standard Mullard production chipset, and is of the viewdata-compatible type. This implies that the data and address lines to its page store are accessible via a connector, a facility which is essential for telesoftware operation. The microprocessor card contains a Signetics 2650 microprocessor and associated memory which together form the local microcomputer.

2.3 The telesoftware page

For the experimental project, telesoftware programs were broadcast as sequences of Oracle pages of text, using up to seven to form a complete program. When teletext pages are used in this way there is no means of determining which pages of the sequence will be received first, and so the terminal is designed to be able to load the pages in any order. Fig. 2 shows the format of the telesoftware page, the contents of which are as follows:

Row 00 is the Oracle page header and is not currently affected by telesoftware transmission.

Row 01 contains a special sequence of characters which identify the page as carrying telesoftware data. Since this sequence is short, and the remainder of the row is not scanned by the microprocessor, the space is used to carry the program title.

Row 02 contains control data for the program itself and must be highly secure. This data is protected by Hamming code,[3] with each byte of data being represented by two contiguous teletext characters. To avoid the use of teletext control characters, only six of the possible seven bits per character are used, giving a total of twelve usable bits. Eight of these make up the data byte, while the remaining four are used for error detection.

The control data are further protected by a block-check character (b.c.c.) and use 16 teletext characters in total. As a further security measure, the control data block is repeated

Paper 8451 E, first received 13th July and in revised form 9th October 1979

Mr. Hedger is with Independent Television, South Bank Television Centre, Kent House, Upper Ground, London, and Mr. Eason is with the Mullard Applications Laboratory, Mitcham, Surrey, England

0020-3270/79/121412 + 05 $01·50/0

in the second half of the row; this allows the terminal a 'second chance' if the first occurrence of the block is incorrectly received. The control data itself is made up as follows:

(*a*) Address field: a 2-byte address that specifies the loading address for the first byte of the program.

(*b*) Count field: a 2-byte value for the number of bytes of program data on the page.

(*c*) Page number: a single byte for page identification within the sequence in the range 0–7.

(*d*) Program b.c.c.: a single byte to verify overall program data.

(*e*) Control line b.c.c.: a b.c.c. for the control data itself. Remaining rows on the page carry the data representing the program.

The data represents machine-code instructions for the microprocessor rather than higher level instructions such as assembler instructions. As with the control data, each byte is represented by two teletext characters with Hamming-code protection. By using this method of encoding, the upper limit of the segment of program which may be carried on a single page is 420 bytes.

3 Description of terminal

The terminal has several distinct features which are now described. The page store consists of random-access memory (r.a.m.) addressed by two bytes of indirect address, the first pointing to a row and the second to a column address. In a conventional teletext decoder, this memory would be used exclusively for storage of a single teletext page for display.

The scratchpad is a further 256 bytes of r.a.m. and is used as a buffer store for data between the page store and secondary memory which is another 4K bytes of r.a.m. forming the main memory of the microprocessor.

The control program is held on 2K bytes of erasable programmable read-only memory (e.p.r.o.m.) and has the following basic functions:

(*a*) to accept and interpret commands on the I-bus

(*b*) to validate incoming data and transfer verified data to secondary memory

(*c*) to reread data incorrectly received

(*d*) to go to the start of the program when correctly loaded

The remote keypad is the ultrasonic command device normally used to control television set functions and which, for telesoftware, additionally acts as a data-entry device for the user.

3.1 Operation

The address and data lines from the microprocessor are connected to the teletext address and data bus via buffer circuits, permitting the microprocessor read/write access to the teletext page store. To avoid a conflict of addressing, the page store is accessed only within defined periods during field flyback. A window-generator circuit prevents access at other times and also sets up an interrupt at the start of the window.

Keypad commands received by the television set are decoded and are used for controlling receiver functions such as volume, brightness etc. In addition, a serial data pattern and timing waveform are generated to form the I-bus, which controls the teletext operating modes and page selection. The I-bus is also taken to an input port on the microprocessor card, enabling it to respond to keypad commands and data.

On the teletext card, the video processor (VIP) slices data and data-clock information from the television composite video signal. This is then fed to the timing chain

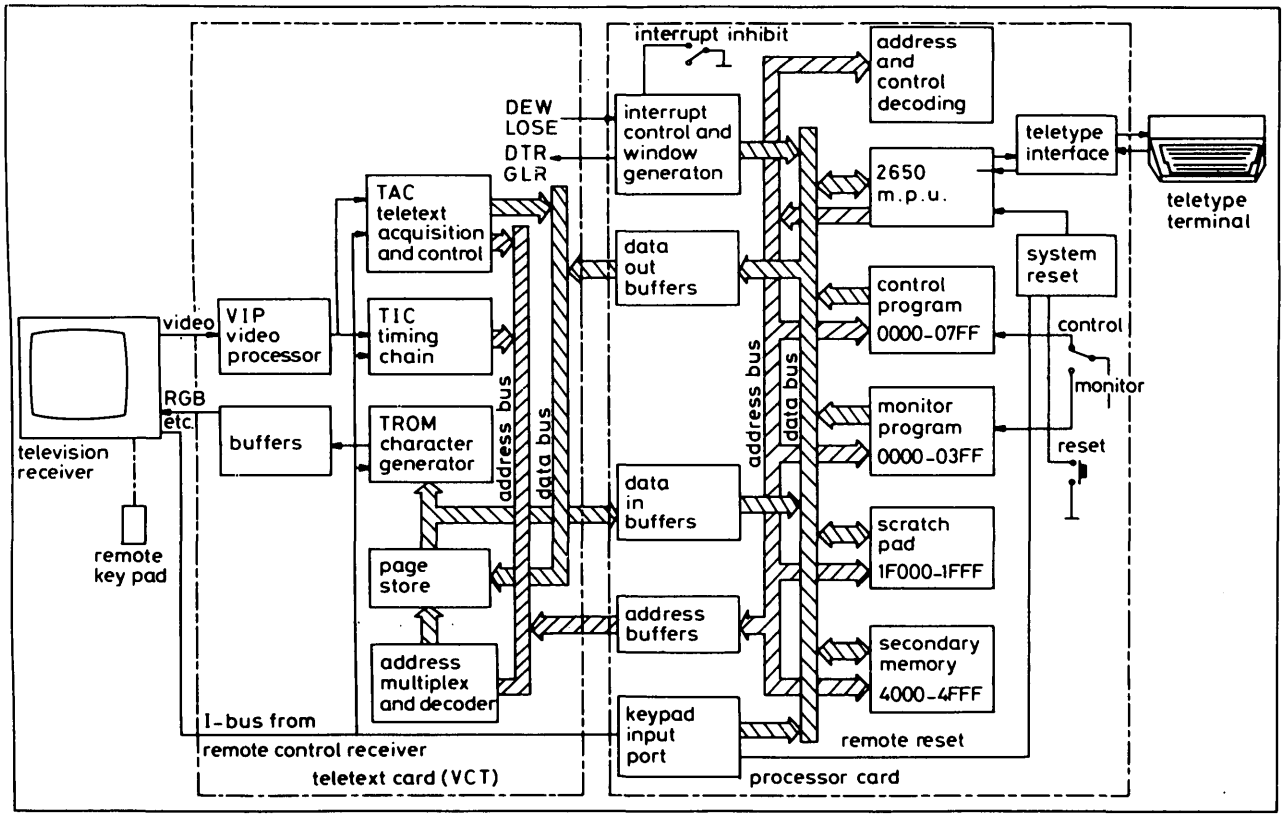


Fig. 1 *Telesoftware demonstration system: simplified block schematic*

(TIC) and to the teletext acquisition and control (TAC). The required page is selected by TAC and, when received, is parallel-loaded to the page store. The character generator (TROM) is fed with this information, enabling it to generate red, green and blue colour signals for page display. Because these colour signals drive the television colour amplifiers directly, fully saturated display colours are produced.

After the microprocessor is reset, either by power-up or by a major change of mode caused by a keypad command, the control program will be run. This program causes the microprocessor to scan the page store for the special identifying character sequence which it will recognise as telesoftware. Until this is found, the terminal behaves like a standard teletext decoder, receiving and displaying pages of text. However, if a valid telesoftware page is selected and loaded into the page store, the microprocessor will, upon recognition, perform error checks upon the data before loading verified data into the correct part of the secondary memory via the scratchpad. This process will be repeated until all pages which form the complete program have been received and loaded. If errors are detected, their location is noted and the erroneous bytes retested when that page is reread subsequently. Since errors are unlikely to occur a second time in exactly the same place on the page, the terminal stands a very good chance of accumulating a perfect program after two passes even under relatively poor decoding conditions.

Once the complete program has been loaded, a prompting message will appear on the screen requesting the user to key VIEWDATA on the keypad. This action sets the teletext card into viewdata mode so that no more teletext pages are received but the page store contents remain displayed; it also starts the program now loaded into memory. From this point, the program can run and will respond to inputs from the keypad. The television screen becomes a colour v.d.u., fed by data loaded by the microprocessor into the page store.

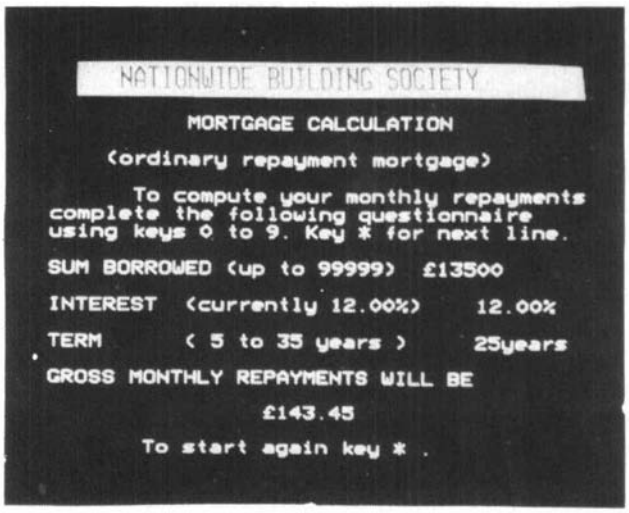


Fig. 3 *'Mortgage calculator' program display*

4 Programs for the experimental terminal

To evaluate the terminal in use, and also to provide the basis for active demonstration, four simple programs were devised for the terminal. These have been broadcast since 1978 on the Oracle service and are written in 2650 machine code. It is hoped that they serve to illustrate (albeit in a somewhat limited way) some of the main application areas for telesoftware. The four programs are as follows:

(*a*) 'Mortgage calculator' (Fig. 3) calculates monthly mortgage repayments based upon user-input variables of term, sum borrowed and interest rate.

(*b*) 'Insurance quotation' allows a user to obtain a quotation for car insurance by entering responses to multiple-choice questions with numerically oriented answers. The questions parallel those to be found in a 'traditional' proposal questionnaire.

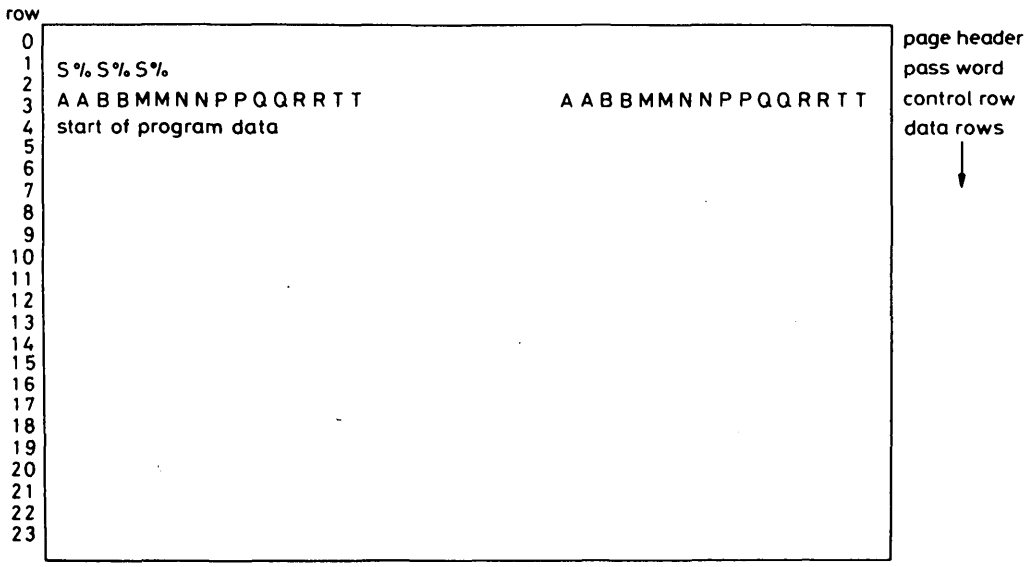


Fig. 2 *Page format*

Characters AABB → AB bytes = address field
MMNN → MN bytes = count field
PP → P bytes = page number
QQ → Q bytes = number of pages
RR → R bytes = program BCC
TT → T bytes = control hire BCC

(*c*) 'ESP' is a number guessing game incorporating a random-number generator within the program.

(*d*) 'First aid' is an example of a simple educational program, again structured around multiple-choice questioning techniques.

4.1 Software considerations

For practical reasons during the experimental period, all the telesoftware programs were written and transmitted in machine code. This method is compact and convenient when dealing with a single microprocessor and associated instruction set but, of course, allows no real portability of programs between one microprocessor and another. To adopt this system as a standard for telesoftware would impose severe restrictions in terms of having to define one machine which must be used in all terminals, effectively precluding competition between different manufacturers.

It would be more desirable to transmit programs in a high-level language and to employ a resident interpreter (held in r.o.m.) to translate this into machine-code instructions to be executed by the microprocessor. This would permit many different microprocessors to be used for telesoftware simply by writing interpreters appropriate to the specific machine in use in each case. The interpreter would also reduce substantially the amount of software per program which must actually be transmitted, because a degree of 'expansion' takes place with a high-level language when the source code is interpreted into machine instructions. Thus writing programs in such a high-level language results in effective compaction, which is crucial in telesoftware, because the longer the program, the longer the user will have to wait to load the software from the teletext system.

A compiler approach would be less desirable, since more memory is required and the program could not possibly start until it had been loaded in its entirety, resulting in undesirable delays. The teletext system itself incurs delays in that the most efficient way of broadcasting programs is using the 'rolling pages' method previously described, with finite times between each, but this would prove less of a problem if a program could commence execution before the whole of it had been received. With careful segmenting of the program, this is possible with the interpretive approach, but not with the compiler.

As well as deciding on whether a compiler or interpreter is most appropriate for the future implementation of telesoftware, a suitable high-level language must be identified. ITV's research to date has pointed to a modified BASIC as being most appropriate,[4] but no final conclusions have been reached.

It should be remembered that telesoftware is likely to be aimed, not at the home-computer enthusiast, but at the man in the street, as a television set with enhanced and highly versatile facilities. Thus the vast majority of users would not possess any knowledge of the traditional computing techniques associated with the current generation of personal computers. For this reason, accepted computer-type operating systems with their attendant error messages such as '***ERROR INVALID OPERATOR OR TERMINATOR' could not be tolerated, or understood! Instead, careful design of programs will have to take place to ensure that software is highly reliable in a runtime-only environment and to ensure the inability of the terminal to produce meaningless, obscure or incorrect information to the user, regardless of data input by him. However, since a real telesoftware will still permit the broadcaster full control of the programs that are broadcast (in the same way as he retains full control of the content of teletext pages). It will be his responsibility to ensure that strict quality-control procedures are carried out on the programs prior to their transmission on air.

5 Other telesoftware applications

It is difficult to assess how the use of telesoftware will evolve until it becomes established as an important and useful extension of teletext. However, work already carried out has pointed to a number of areas where applications seem most appropriate, and a few of these are now described.

5.1 Self-assessment programs

Examples of self-assessment programs are mortgage calculators, tax assessment programs, social-security benefits assessment etc. These applications would normally take the form of question-and-answer sequences with the terminal performing numerical and logical calculations upon data entered by the user interactively. In their simplest form, programs would need only the numeric keypad for data-entry, but others, of more complexity, would necessitate the use of a full alphanumeric keyboard as an extra peripheral. A characteristic of this area of applications is the large quantity of text needed to be held within a program.

5.2 Education

Education is a large and important area in which telesoftware could have a major role to play. It provides the educationalist with two significant advantages: first, it is completely cost free, once the relatively cheap hardware has been obtained; secondly, any number of simultaneous users may be supported (clearly, no peak-loading problem exists in the ether).

Many existing and proven techniques for computer-aided learning can be translated directly into telesoftware, such as the teaching of numeracy, some mathematics and some science subjects. However, to be really efficient, a significantly large portion of a teletext database needs to be devoted entirely to the broadcasting of educational programs to provide the necessary diversity and complexity required.

5.3 Video games

Video games can be viewed as falling into two groups; verbal-reasoning games and manual-dexterity games. The former are ideally suited to telesoftware implementation, since they involve manipulation of mainly textual data and will need only a keypad, or at most and alphanumeric keyboard for data entry. The latter, however, deal with factors such as hand-to-eye co-ordination and are nearer in concept to the 'traditional' (fixed) video games; as such they demand the use of paddles, joysticks and other control devices. A real-time clock might also be necessary.

5.4 Database broadcasting

An example of database broadcasting is a system proposed to use telesoftware to detect credit-card fraud. A telesoftware terminal located at the point of sale in a store could be loaded with a small program at the start of the working day. Throughout the day, Oracle would broadcast updated

list of lost or stolen credit-card numbers at intervals and these would be used by the terminal to maintain its own up-to-date lists. The sales assistant, when presented with a 'dubious' credit card, would simply key in on the keypad the number of the card and the terminal would attempt to match it with its latest list. If it succeeds, the assistant would be advised to take 'precautionary measures'. This system could permit terminals all over the country to be kept constantly up to date without the use of expensive computers or even a telephone.

5.5 Teletext enhancements

As well as providing a computer-type facility at low cost, current thinking suggests that telesoftware could represent an effective method of compatibly extending the facilities offered by the teletext display, particularly in the areas of high-definition graphics, vector graphics and dynamically redefined character sets. Furthermore, since telesoftware terminals will contain more r.a.m. than would be found in a single-page teletext decoder, the memory can be used to hold multiple pages, and perhaps even to store a succession of related pages and display them interactively as the user required. This could conceivably ease the access-time delay inherent in the existing teletext system.

6 Standardisation of telesoftware

At the time of writing, much work is being done to establish the technical standards for a future telesoftware system and these will obviously include recommendations for terminal design. However, the degree of specification will have to balance carefully flexibility and economics. A highly flexible system would tend to be uneconomic in the early days, possibly inhibiting the growth of the market itself, whereas a rigidly defined system could be very economic in the early days, perhaps leading to rapid market growth, but probably becoming increasingly restrictive as applications become more demanding and hardware even cheaper.

Perhaps the most important single factor in any such specification is the need to preserve compatibility between telesoftware and the earlier generation of simple teletext, so as not to render existing hardware obsolete overnight.

Consideration will have to be given to the range of peripherals which are bound to be demanded by telesoftware users, such as cassette recorders, discs, printers etc., particularly in terms of electrical isolation from the e.h.t. in the television set. Consideration will also have to be given to the hardware/software interfacing required.

7 Conclusion

The experimental work carried out so far shows telesoftware to be a practical and cost-effective means of extending teletext compatibly to offer facilities hitherto only possible with more sophisticated and expensive solutions. With mass production of l.s.i. (large-scale integration) units, the incremental cost of minimal telesoftware facility in a teletext-decoder/t.v. set would be of the order of £50, and this would be a once-and-for-all outlay with no appreciable running costs and a supply of free software.

However, certain limits have also to be recognised: telesoftware can never order a plane ticket or place an advertisement, which will be exclusive applications of two-way systems like viewdata, but its advantages of simplicity and low cost make it highly suitable for the domestic market.

In a year or two from now, perhaps, inside every television there will be a microprocessor trying to get out. Our hope is that telesoftware will provide the key.

8 References

1 'Broadcast teletext specification'. Joint IBA/BBC/BREMA publication, September 1976
2 OVERINGTON, W.J.G.: 'Telesoftware', *Computing*, 1977, **5**, p. 25
3 HAMMING, R.W.: 'Error detecting and error-correcting codes', *Bell Syst. Tech. J.*, 1947, **26**, pp. 147–157
4 GREEN, N.W.: 'Software for telesoftware'. Report under preparation by the Independent Television Companies Association