Microprocessor Systems Development Laboratory
Department of Systems Engineering and Computing Science
Carleton University

MSDL Tech Note 80-1 (DOC)
March 31, 1980

WORD TELEPROCESSING INTERFACE

by

R.J.A. Buhr - Principal Investigator
D.A. MacKinnon - Associate Researcher

Prepared for the Department of Communications
under Contract No. OSU79-00174
"Word Teleprocessing Interface"

Microprocessor Systems Development Laboratory
Department of Systems Engineering and Computing Science
Carleton University

MSDL Tech Note 80-1 (DOC)
March 31, 1980

WORD TELEPROCESSING INTERFACE

by

R.J.A. Buhr - Principal Investigator
D.A. MacKinnon - Associate Researcher

Prepared for the Department of Communications
under Contract No. OSU79-00174
"Word Teleprocessing Interface"

Acknowledgement

This report would not have been possible without the cooperation of the companies whose products have been considered herein. We are sincerely grateful for the help of Dr. Umamaheswaran of AES Data Ltd.; of Keith Wilson, IBM Sales Representative; of Hans Rosenstein, MICOM Communications Product Manager; of Paul Kelly, MICOM Sales Representative; of Adele Saulnier, MICOM Marketing Support Representative; of Miles Kennedy, WANG Marketing Representative; of Dave Duquette, WANG Office Systems Analyst, and of Kerry Hawkins, XEROX Branch Product Specialist.

The efforts of Elaine Carlyle and Inez Hoey, who typed this report and who patiently suffered through its many revisions, are also greatly appreciated.

R.J.A. Buhr,
D. MacKinnon.

Table of Contents

Chapter 1: Introduction

1.1 Background
1.2 Terms of Reference of the Study
1.3 Outline of the Report

Chapter 2: An Overview of Communicating Word Processors

2.1 Introduction
2.2 Logical Components of a Communicating Word Processor

Chapter 3: Functional Profile of a Number of Commercially Available Word Processors

3.1 Introduction
3.2 AES Plus

3.2.1 Character Set
3.2.2 Control Functions
3.2.3 Format Information

3.3 IBM Office System 6

3.3.1 Character Set
3.3.2 Control Functions
3.3.3 Format Information

3.4 MICOM 2000/2001

3.4.1 Character Set
3.4.2 Control Functions
3.4.3 Format Information

3.5 WANG Word Processor 5

3.5.1 Character Set
3.5.2 Control Functions
3.5.3 Format Information

3.6 XEROX 850

3.6.1 Character Set
3.6.2 Control Functions
3.6.3 Format Information

3.7 Summary

Chapter 4: Communications Profile of a Number of Commercially Available Word Processors

4.1 Introduction
4.2 AES Plus

4.2.1 TTY Emulation
4.2.2 Batch Terminal Emulation
4.2.3 AES Point-to-Point

4.3 IBM OS/6

4.3.1 Batch Terminal (2770) Emulation

4.4 MICOM 2000/2001

4.4.1 TTY Emulation
4.4.2 Batch Terminal Emulation
4.4.3 WP-to-WP

4.5 WANG Word Processor 5

4.5.1 TTY Emulation
4.5.2 Batch Terminal Emulation
4.5.3 WP-to-WP

4.6 XEROX 850

4.6.1 TTY Emulation
4.6.2 Batch Terminal (2780) Emulation
4.6.3 Batch Terminal (2770) Emulation
4.6.4 WP-to-WP

Chapter 5: Current Interworking Capabilities and Problems

5.1 Introduction
5.2 TTY Level

5.2.1 AES Plus
5.2.2 MICOM 2000/2001
5.2.3 WANG Word Processor 5
5.2.4 XEROX 850

5.3 BSC Level

5.3.1 AES Plus
5.3.2 IBM OS/6
5.3.3 MICOM 2000/2001
5.3.4 WANG Word Processor 5
5.3.5 XEROX 850

5.4 Problems

5.4.1 Code Translation
5.4.2 Control Function Representation
5.4.3 Format Information Representation
5.4.4 Protocol Deficiencies

Chapter 6: Possible Solutions

6.1 Introduction
6.2 Translation Centre

6.2.1 Introduction
6.2.2 Code Translation
6.2.3 Function Representation
6.2.4 Format Information Representation
6.2.5 Communication Protocols
6.2.6 Remarks

6.3 Virtual Word Processor

6.3.1 Introduction
6.3.2 Guidance from International Standards
6.3.3 Code Translation
6.3.4 Control Function Representation
6.3.5 Format Information Representation
6.3.6 Communication Protocols
6.3.7 Remarks

6.4 Comparison of the Two Approaches

Chapter 7: Conclusions and Recommendations

7.1 Summary and Conclusions
7.2 Recommendations

APPENDIX A: ASCII Code Set
APPENDIX B: EBCDIC Code Set
APPENDIX C: EBCDIC/WP Code Set
APPENDIX D: Detailed Discussion of CWP Interconnection

D.1 Introduction
D.2 TTY Level

D.2.1 AES Plus - AES Plus
D.2.2 AES Plus - MICOM 2000/2001
D.2.3 AES Plus - WANG Word Processor 5
D.2.4 AES Plus - XEROX 850
D.2.5 MICOM 2000/2001 - MICOM 2000/2001
D.2.6 MICOM 2000/2001 - WANG Word Processor 5
D.2.7 MICOM 2000/2001 - XEROX 850
D.2.8 WANG Word Processor 5 - WANG Word Processor 5
D.2.9 WANG Word Processor 5 - XEROX 850
D.2.10 XEROX 850 - XEROX 850

D.3 BSC Level

D.3.1 AES Plus - AES Plus
D.3.2 AES Plus - IBM OS/6
D.3.3 AES Plus - MICOM 2000/2001
D.3.4 AES Plus - WANG Word Processor 5
D.3.5 AES Plus - XEROX 850
D.3.6 IBM OS/6 - IBM OS/6
D.3.7 IBM OS/6 - MICOM 2000/2001
D.3.8 IBM OS/6 - WANG Word Processor 5
D.3.9 IBM OS/6 - XEROX 850
D.3.10 MICOM 2000/2001 - MICOM 2000/2001
D.3.11 MICOM 2000/2001 - WANG Word Processor 5
D.3.12 MICOM 2000/2001 - XEROX 850
D.3.13 WANG Word Processor 5 - WANG Word Processor 5
D.3.14 WANG Word Processor 5 - XEROX 850
D.3.15 XEROX 850 - XEROX 850

APPENDIX E: International Standards Relevant to Word Processing

Chapter 1

Introduction

1.1 Background

This report presents the results of a study performed by Carleton University under a DOC University Research Contract entitled "Word Teleprocessing Interface".

This report is a contribution to the Canadian Government's efforts to gain insight into the problems associated with communication between word processors from different manufacturers. This insight will assist the Government Telecommunications Agency (GTA) to fulfil its advisory and operational mandates with respect to communicating word processors.

Recent years have witnessed widespread use of word processing machines in offices in both government and industry. A wide variety of word processing products has become available to service this flourishing market. As business office automation continues and with the increasing trend towards organizational decentralization, it is expected that the need for communications between such devices will grow. In answer to this perceived need, many manufacturers have added a communications capability to their products.

Unfortunately, no industry-wide standards exist at this time for defining communications protocols appropriate to the word processing environment. As a result, each manufacturer has developed its communications packages on an ad hoc basis, using only its own perceptions of the requirements of the market. In general these perceived requirements fall into three categories:

1) to provide a low-speed asynchronous communications capability for talking to mainframe computers in an interactive fashion;

2) to provide a medium-speed synchronous communications capability for batch-type exchanges with mainframe computers, and

3) to provide a file transfer capability between two machines from the same vendor.

Not included in this list is the requirement that word processors from different manufacturers communicate effectively. This requirement may be perceived by a manufacturer as destroying a market advantage with respect to an established customer base. Or it may be perceived as too difficult to satisfy at this time in the absence of appropriate standards. The latter difficulty arises from the differences in internal structures (both hardware and software) among devices. In any case, most of the current communications offerings have concentrated on enabling contact between two identical word processors or between a word processor and a large computer. One exception is the effort made by XEROX to have its word processor communicate with IBM word processing products.

This report investigates to what extent the current communications packages are suitable for the task of interconnecting differing word processors. It also examines ways of resolving the major incompatibilities that do exist so that it becomes possible for different machines to communicate effectively for the transfer of files.

The specific experience the authors of this report bring to bear on this investigation is their involvement with several projects of Carleton's Microprocessor Systems Development Laboratory which successfully implemented intelligent terminal systems in the TRADEX and Electronic Mail areas. A significant part of this effort has been involved with the adaptation of existing word processing terminals for these functions.

In addition to this practical experience, a parallel study for the Department of Communications involving one of the authors (Buhr) under a Research Contract entitled "Open Systems Interconnection: Issues Associated with ISO Layered Model" investigates the appropriateness of the ISO layered communication model for the communicating word processor environment.

The product information in this report is based on written documents and verbal presentations provided by representatives of the companies involved. Each company has different policies regarding the release of this type of information, so that there may be inconsistencies in the amount of detail contained in the product descriptions found in this report. Further, as actual testing of the interworking capabilities of the devices mentioned herein was outside the scope of this project, the discussions dealing with this topic represent the authors' best efforts at interpreting the available data. Under the circumstances, comments that have been made regarding the behaviour of the various communications packages should not be taken as definitive; however, they can be considered as a fair indication of current interworking capabilities.

1.2 Terms of Reference of the Study

The terms of reference are best stated by quoting Sections 5 and 7 of the project proposal (which is part of the contract):

"5. Purpose of Research

The ability of dissimilar word processing machines to communicate through, for example, a general multi-layered protocol or a network administrative machine could provide considerable stimulus to business office automation in two main respects:

i) consequent cost savings through streamlining the whole process of document exchange within and between organizations by eliminating the "hand carried" labour intensive part of this operation;

ii) permitting reasonable freedom of choice in selection of word teleprocessing equipment when such devices are able to communicate."

"7. Brief Description of Research Method to be Used

The research will be conducted in two steps:

i) In the first step, a survey of existing word processing machines, either manufactured in Canada or widely available in Canada, will be conducted. The survey will examine the internal structure of each machine to the extent that this is possible based on information in the public domain in order to determine the essential similarities and differences that will affect interfacing these machines (e.g., internal code sets, control characters, file subsystems, existing communications facilities, etc.).

ii) Aided by the results of the survey, several conceptual alternatives for interconnecting word processing machines will be examined to determine their feasibility (e.g., implementation requirements, generality, ease of use, etc.) and limitations."

Five word processors have been selected for this report. They are (in alphabetical order):

1) AES Plus
2) IBM Office System 6
3) MICOM 2000/2001
4) WANG Word Processor 5
5) XEROX 850

All of these are stand-alone machines. As such, they all consist of the following components: video display with a keyboard; magnetic recording unit capable of easy manual handling, insertion, extraction and filing; internal memory; and a logic unit for control of the processing. They do not require data processing or other support in order to function. They are self-contained systems which allow an operator to make keystroke entry of text, to display text, to edit text and to print text. With a communication capability, the operator is additionally able to transmit and receive text. For the purposes of this report, only the transmission and reception of text from and to file storage will be considered.

1.3 Outline of the Report

Chapter 2 provides a brief overview of communicating word processors and describes the general characteristics of communications packages that are currently available. It then presents the general methodology to be used in the remainder of the report to describe the logical components of a communicating word processor (CWP).

In Chapter 3, a functional profile of each machine is provided. This profile describes what information can be entered into the terminal, and includes the range of textual material and control codes available for each device as well as a mention of how format information is stored.

Chapter 4 deals with communications profiles for each CWP. The nature and performance of the various communications protocols available for each terminal are discussed and the types of mapping operations which take place presented.

These profiles serve as input to the discussion of Chapter 5 where the interworking capabilities are studied. Among the issues reviewed are the problems involved in the translation of text characters, in the representation of control functions and format information and in the behaviour of the communications protocols.

In Chapter 6, two possible solutions to the problems outlined in Chapter 5 are proposed, evaluated, and compared. The two approaches considered are a central translation facility and a virtual word processor.

Chapter 7 summarizes the findings of this report and presents recommendations regarding the short and medium term directions for communicating word processors.

Chapter 2

An Overview of Communicating Word Processors

2.1 Introduction

The logical components of a communicating word processor remain the same no matter what category of communication option is supplied by different manufacturers. The purpose of this chapter is to present the logical view which will be used throughout this report. To lay the groundwork, we first review in this section the different categories of communication options. Then in Section 2.2 we present and discuss the logical components.

The communications options currently available for word processors fall into three categories:

(i) "dumb terminal" emulation (TTY mode)
(ii) remote batch terminal emulation
(iii) word processor to identical word processor

The objectives, capabilities and limitations are different for each option, as discussed below:

Dumb Terminal Emulation: This option makes the WP appear as a dumb terminal to the remote device. This remote device is normally a mainframe, although this need not always be the case. It is intended to provide an interactive communication capability between the WP operator and the mainframe, generally at low speeds (110 to 1200 bps). There is usually also a facility for pseudo-batch operation, whereby quantities of data may be transferred to/from the local storage medium. The communication protocols used (TTY or IBM 2741) are simple asynchronous ones with little or no protection, and operate normally in half-duplex fashion. Because a variety of mainframes are to be accommodated by the one communications package, a series of options are usually available to tailor the operating characteristics of the terminal to particular mainframes.

Remote Batch Terminal Emulation: This option configures the Word Processor as a batch terminal able to communicate with a mainframe or any other machine supporting the appropriate protocol. The three protocols in common use are the IBM 2780, 3780 and 2770 protocols. All three use IBM's Binary Synchronous Communications (BSC) protocol for controlling the data link and operate at speeds up to 2400 bps. With this option, the WP user can transfer textual material to/from the word processor's local storage medium in transparent or non-transparent EBCDIC or ASCII. During 2780 or 3780 emulation, the WP terminal appears to the remote device as a card reader for input and as a line printer or card punch for output. The 2770 protocol has additional functions for handling a display terminal. These protocols impose code translation and data formatting restrictions which may result in data loss.

WP to WP: This option allows two machines from the same manufacturer to exchange complete document information. Whereas the other two communications options allow the transmission of only a subset of the information describing a document, this option permits exact duplication of a document from one terminal to another. This is possible because the actual codesets used internally in the terminal are transmitted. In addition, the manufacturers may use private protocols which satisfy all information transmission requirements. No attempt is made to emulate other devices and no code translation takes place, so no restrictions are imposed and no information need be lost. Although each manufacturer's protocol has its unique aspects, most are based on the BSC data link protocol and so operate in synchronous, half-duplex fashion.

Each manufacturer offers with its communications options various features such as unattended operation, multiple outstanding transmission requests, activity monitoring etc... These will be summarized in Chapter 4. However, every communicating word processing terminal, no matter what communications option it uses, has the same basic set of components. These components are discussed next.

2.2 Logical Components of a Communicating Word Processor

A Word Processor (WP) allows the operator to perform three basic activities :

i) enter and save a set of text characters,
ii) give structure to this set (headings, paragraphs, etc.) and
iii) alter with ease both the set of characters and its structure.

The first two activities can be performed effectively with a typewriter. The principle virtue of the word processor is the third activity. In order to perform this activity successfully, the WP must add control structure to the set of text characters over and above the text structure seen by the operator on the printed page. An example of this is the storage of the set of characters in a disk file. This involves arranging the characters in a disk compatible form, including format information such as margin settings, tab stops etc., and giving the whole a filename. Also, various features available for manipulating the set of characters require the insertion of special characters into the textual set for subsequent interpretation by the system. Examples of such special codes are justification information, end-of-paragraph symbols, and printer control codes such as bold face and superscripts. The set of functions for manipulating data and the associated set of embedded control codes is different for each machine.

When textual data is stored on disk, it is completely described by the data characters in the file and by the associated formatting information. The data characters consist of both text characters and control codes, while the formatting information may take the form of file header(s) or special embedded text sequences. These three elements, namely format information, text characters and control codes, make up the logical components of the file.

Thus, the file is a fundamental unit of information in the word processing environment. Note that although the file is a self-contained unit of information, it need not exist in the system as a totally independent entity; there may be external structures used to define associations of files. Typically, this is achieved through a disk directory which may define a group of files as a document where each file corresponds to a page of text.

When a communications option is added to a word processor, the objective is to give the system the capability of transferring file contents from one machine to another. However, the communications environment introduces two possible sources of limitations: the communications protocol and the functional capability of the remote machine. Either or both may not be able to handle all the available information without loss. If such is the case, the result of communications activity is a file on the receiving end which contains a subset of the information in the file at the transmitting end. Each of the logical components of the file, i.e., the format information, the text characters and the control characters, may be affected differently by the constraints of the communications protocol and of the remote machine.

The Communicating Word Processor (CWP) in essence performs a file transfer function. This activity may be summarized briefly as follows. As shown in Figure 2.1, each information component undergoes a mapping operation to transform it into a form suitable for both the communications protocol and the remote machine. It is then transported to the other end and is mapped there into useable form. It is during these mapping operations that information loss may take place. The extent of this loss is a function of the communication protocol and of machine compatibility. The nature of the information mapping for each of the file information components is discussed below:

format information: the format information mapping is affected by the remote machine, by the type of communications protocol and by the nature of the internal representation of the format information itself. Where the information is stored in a file header, the extent of information loss may range from zero for a transfer between identical machines to total for a transfer taking place in TTY mode. In the latter case, the transmitted information might consist solely of a string of text characters with all formatting information absent. Where formatting information is stored as a special embedded block of text, the amount of information loss during transfer is determined by the code translation facility of the communications protocol and by the remote machine's ability to interpret the received formatting information.

text: the principal mapping in this case is a code translation function, usually from the internal 8-bit representation of the machine to the 6, 7, or 8-bit representation of the communications line. The typical communications code sets are correspondence (6-bit) used with the 2741 protocol, ASCII (7 or 8-bit) and EBCDIC (8-bit). Also, when communicating between identical machines, the internal 8-bit code set is sometimes used. Other translation functions may also be in effect to handle special characters such as underscores and accents. For example, an underscored character may be translated into a character-backspace-underscore sequence. Finally, there may be a structural mapping whereby the organization of text in the file is altered (perhaps irrevocably) to suit the protocol requirements. An example of this is the segmentation of a line to suit the blocking requirements of the protocol (e.g. 80 character records). If the line boundaries are not preserved, or if new ones are introduced, then the original structure is altered.

control codes: the mapping in this case is affected by both the remote machine and the communications protocol. Some protocols only allow the transmission of 6 or 7 bit codes or the non-transparent transmission of 8-bit codes, all of which preclude the transmission of special characters. When the protocol does allow the transparent transmission of 8 bit codes, the use of different code values for special functions by different machines and the fact that much mainframe software will not accept unusual code values makes the transfer of control codes difficult. For these reasons, most communications packages translate control codes into either space or null characters on transmission.

The logical components of a CWP are those that are involved in the information transfer process and include those elements that effect the transfer and those that are affected by it. As already mentioned, the components affected by the transfer process are the format information, the text data and the embedded control codes. The components which effect the transfer process are the Code Translation and Communications Protocol Modules, both of which are part of the communications package.

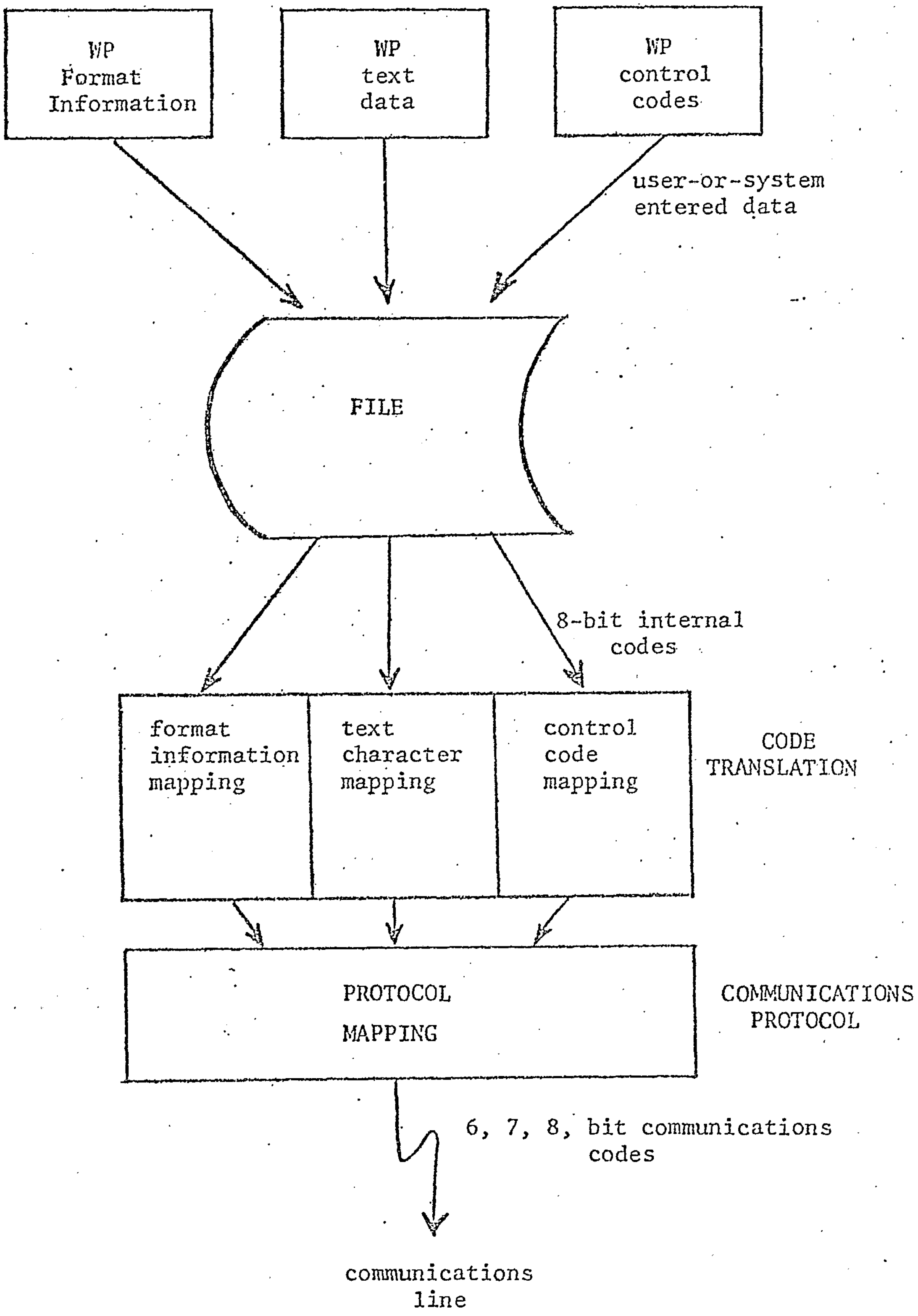


Figure 2.1: Logical Components of a Communicating Word Processor

Chapter 3

Functional Profile of a Number of Commercially Available Word Processors

3.1 Introduction

The logical components of a Communicating Word Processor are divided into two types: the information bearing and the information transfer components. It was shown in Chapter 2 that since the principal function of a communications option is to provide a file transfer capability, then the file contents become the main information bearing elements of the system. Further, three distinct data groups inside the file were identified: the textual data, the embedded control characters and the format information. These data groups are considered as distinct because each may be affected differently by the communications processing. The purpose of this chapter is to provide a functional profile of these data groups for a number of commercially available word processors. This information, together with the communications profile of Chapter 4, is required input for the discussions of Chapters 5, 6 and 7.

In the next five sections of this Chapter, a profile is developed for each of the five machines under consideration. This profile includes the following items: 1. The range of textual characters that can be generated by the machine; 2. a list of the control functions that result in embedded characters, and 3. the way that format information is stored.

The range of textual characters is important because it points to the lack of code set standardization among machines. The level of code set standardization available in current communications packages is not sufficient for the word processing environment.

Similarly, the list of control functions is important because it points out the similarities and differences in functionality among machines. Even where two or more machines have similar functions, there is no agreement on the internal representation of these functions, so a resulting information loss occurs during transmission. This is another area where the level of standardization is insufficient. Table 3.6 provides a composite list of all control functions identified for the 5 CWPs in this study. It shows which functions exist in which machines. Finally, the way that format information is stored in the disk file and processed during communications is also different for each machine and thus is another source of incompatibility.

3.2 AES Plus

3.2.1 Character Set

Each AES Plus machine contains a single character generator capable of displaying 128 characters. There are four different character generators to handle different language requirements. Each handles a different subset of the total set of text characters available. As a result, there is a possibility of data loss when communicating between machines with different character generators.

3.2.2 Control Functions

The set of control functions which have corresponding embedded control codes is shown in the table 3.1. Function descriptions, where helpful, are included in brackets. The end-of-page and end-of-line functions are the only ones which do not have a visual representation. The visual representation of the other functions is achieved through single or multiple character strings.

3.2.3 Format Information

Format information for each page of text is stored on diskette in the form of a file header. There is only one such set of information per page. Included in the file header is such information as page size, margin settings, tab stops and printer horizontal and vertical spacing.

TABLE 3.1 Control Functions for the AES Plus

- space (normal spacebar)
- unrequired space (indicates areas of screen where no characters have been entered)
- required hyphen
- unrequired hyphen (may be deleted during reformatting)
- tab space (result of tab function)
- connecting space (not affected by justification)
- end-of-line
- required end-of-line
- end-of-page
- justification
- merge with insert (allows text insertion at selected position)
- merge without insert (allows text entry at selected position)
- merge with numeric (allows insertion of numeric data at selected position)
- ignore during merge (selected text is not included in merge operation)
- backspace overwrite
- print commands:
  - bold type
  - down 1-99 increments
  - return to baseline
  - end bold type
  - force new page
  - character spacing, change to 1-9
  - character spacing, return to original
  - indent 1-99 character positions
  - stop indent
  - inhibit line feed
  - start of protected table
  - inhibit return/line feed
  - stop printer
  - return to top of form
  - up 1-99 vertical increments
  - up 2 vertical increments
  - change line spacing to 1-99 increments
  - reset to original line spacing
  - make next character superscript
  - make next character subscript
  - print current page number
  - end of protected table
  - begin non-proportional printing
  - end of non-proportional printing
  - align text when printing proportionately
  - change character spacing table dynamically
  - start inhibit printing
  - stop inhibit printing

- start of underline
- end of underline
- character compression
- double underline

3.3 IBM Office System 6

3.3.1 Character Set

The IBM OS/6 has three (and optionally five) character generators available per machine. One of these can handle up to 256 characters, of which 231 are currently in use. Other character generators include a 94 character one and a 96 character symbol generator useful for greek letters and equations.

3.3.2 Control Functions

The set of control functions which have corresponding embedded control codes is shown in Table 3.6. Function descriptions where helpful are included in brackets.

3.3.3 Format Information

The IBM OS/6 allows multiple format specifications within a document. There are two types of specifications; one type affects all subsequent text in the document (unless directed otherwise) and the other type affects only a specified body of text. Format information includes margin and tab settings, interline spacing, keyboard ID and printer pitch and font. Format specifications are identified in the text by control codes or sequences of text characters.

Table 3.2 Control Functions for IBM OS/6

backspace	
unit backspace	(backspace providing character alignment in proportional spaced printing)
numeric backspace	(fixed increment backspace)
index	(line feed)
index return	(same function as carrier return plus performs device control)
carrier return	(end-of-line)
required carrier return	(required end-of-line)
space	
numeric space	(fixed increment space)
required space	(not affected by justification)
subscript	
superscript	
end-of-page	
required end-of-page	
tab	
indent tab	
decimal tab	
syllable hyphen	(normal end-of-line hyphen)
required hyphen	(not affected by reformatting or justification)
word underscore	
switch	(switch device)
stop	(tells printer to stop; useful for changing printwheels)
repeat	(allows repetition of character sequences)
center	
format change	
return format	
envelope feed	
typestyle change	
return typestyle	
keyboard change	
prefix	
delete	(invalid character)

3.4 MICOM 2000/2001

3.4.1 Character Set

The MICOM machine has a single character.generator which can generate 256 different characters.

3.4.2 Control Functions

The set of control functions which have corresponding embedded control codes is shown in Table 3.2. Function descriptions, where helpful, are included in brackets.

3.4.3 Format Information

The MICOM machine allows a single format specification per page of text. This specification is displayed at the top of the page. When a page is first created, a default specification is associated with it.

Table 3.3 Control Functions for the MICOM 2000/2001

- space
- required space
- end-of-line
- required end-of-line
- tab
- indent tab
- numeric tab
- unrequired hyphen
- required hyphen
- end-of-page
- subscript
- superscript
- center
- print commands:
    - start bold type
    - end bold type
    - up $\frac{1}{4}$ increment
    - down $\frac{1}{4}$ increment
    - change horizontal spacing
    - change vertical spacing
    - set vertical offset (defines top margin)
    - set horizontal offset (defines left margin)
- merge with insert (new text is inserted at selected location)
- merge with overwrite (new text is placed at selected location; old text is overwritten)
- ignore during merge (selected text is not included in merge operation)
- end-of-variable (delimits new text to be merged)

3.5 Wang Word Processor 5

3.5.1 Character Set

Each Wang Word Processor 5 machine contains a single character generator capable of displaying 128 characters.. Underlined characters are represented by a different internal code.

3.5.1 Control Functions

The set of control functions which have corresponding embedded control codes is shown in Table 3.4. Function descriptions, where helpful, are included in brackets. The end-of-page and end-of-line functions are the only ones which do not have a visual representation.

3.5.3 Format Information

The Wang machine allows multiple format specifications within a document. The scope of each specification is from the current line to the next format specification. When a document is first created, a default specification is associated with it. Included in the format specification is such information as margin settings, tab stops and printer horizontal and vertical spacing. Each format specification is identified in the text by a special control character.

Table 3.4 Control Functions for the Wang Word Processor 5

- center
- tab
- end-of-line
- required end-of-line
- end-of-page
- indent
- decimal tab (useful for aligning columnar data)
- format block
- stop (tells printer to stop; useful for changing printwheels, etc.)
- note (delimits section of text to be treated as special, e.g., suppress printing)
- merge
- superscript
- subscript
- space
- hyphen

3.6 XEROX 850

3.6.1 Character Set

The XEROX 850 has a single character generator which is capable of displaying 256 different codes.

3.6.2 Control Functions

The set of control functions which have corresponding embedded control codes is shown in Table 3.4. Function descriptions, where helpful, are included in brackets. All functions in the XEROX 850 have a displayable counterpart.

3.6.3 Format Information

The XEROX machine allows multiple format specifications within a document. The scope of each specification is from the current line to the next format specification. When a document is first created, a default specification is associated with it. Format information includes justification selection, margin and tab settings, page size and numbering, and header and trailer texts to be included at the top and bottom of each printed page. Each format specification is identified in the text by a special control character.

3.7 Summary

A composite list of control functions for the five CWPs is shown in Table 3.6. The size of the table indicates the large number of available functions while the fact that only six functions are common to all machines reveals the extent of the disparity in functionality among devices. Where the same function has different terminology on different machines (e.g., the carrier return on the IBM and XEROX machines is the same function as end-of-line on the others), only a single description has been used in Table 3.6.

Table 3.5 Control Functions for XEROX 850

- index	(subscript)
- reverse index	(superscript)
- format block	
- lower tab	(paragraph indent)
- upper tab	(normal tab)
- unrequired space	(normal space)
- required space	(forced space - not affected by justification)
- unrequired end-of-line	(normally entered by system when word wraparound occurs)
- required end-of-line	(same as end of paragraph)
- unrequired end-of-page	(normally entered by system when page is full)
- required end-of-page	(user entered to force end of page)
- center	
- stop	(stops printer; useful for changing printwheels)
- required backspace	
- required half-unit backspace	
- unrequired hyphen	(may be removed during reformatting)
- required hyphen	
- flush right	(forces subsequent characters to be entered to the left of symbol)
- column center	(useful for centering text within columns)
- column double underscore	
- begin underscore	
- end underscore	
- revision mark	
- non-reproducing stop	

Table 3.6 Composite List of Control Functions

	AES	IBM	MICOM	WANG	XEROX
space	X	X	X	X	X
required space	X	X	X		X
numeric space		X			
end-of-line	X	X	X	X	X
required end-of-line	X	X	X	X	X
index		X			
index return		X			
backspace	X	X			X
unit backspace		X			
numeric backspace		X			
half-unit backspace					X
tab	X	X	X	X	X
indent tab	X	X	X		X
numeric tab		X	X	X	
unrequired hyphen	X	X	X	X	X
required hyphen	X	X	X		X
end-of-page	X	X	X	X	X
required end-of-page		X			X
subscript	X	X	X	X	X
superscript	X	X	X	X	X
center		X	X	X	X
column center					X
flush right					X
start underline	X				X
end underline	X				X
word underscore		X			
double underscore	X				X
stop printer		X		X	X
non-reproducing stop					X
note				X	X
justification	X				
end-of-variable			X		
merge				X	
merge with insert	X		X		
merge with overwrite	X		X		
merge with numeric	X				
ignore during merge	X		X		
revision mark					X
format block				X	X
print commands:					
- start bold type	X		X		
- end bold type	X		X		
- down 1-99 increments	X				
- return to baseline	X				
- force new page	X				
- change character spacing	X		X		
- up ¼ increment			X		
- down ¼ increment			X		
- set vertical offset			X		
- set horizontal offset			X		

Table 3.6 Composite List of Control Functions (cont'd)

	AES	IBM	MICOM	WANG	XEROX
- reset character spacing	X				
- indent 1-99 character positions	X				
- stop indent	X				
- inhibit line feed	X				
- inhibit return/line feed	X				
- start of protected table	X				
- end of protected table	X				
- return to top of form	X				
- up 1-99 vertical increments	X				
- up 2 vertical increments	X				
- change line spacing	X		X		
- reset line spacing	X				
- print current page number	X				
- begin non-proportional printing	X				
- end non-proportional printing	X				
- align text when printing proportionately	X				
- change character spacing table	X				
- start inhibit printing	X				
- stop inhibit printing	X				
switch		X			
repeat		X			
format change		X			
return format		X			
envelope feed		X			
typestyle change		X			
return typestyle		X			
keyboard change		X			
prefix		X			
delete		X			
character compression	X				

Chapter 4

Communications Profile of a Number of Commercially Available Word Processors

4.1 Introduction

This chapter summarizes the characteristics of the information processing components of the five CWP machines involved in this study. For each machine, a profile is presented in terms of the categories of communications options available. Recall from Chapter 2 that there are three categories of communication options, namely TTY Emulation, Batch Terminal Emulation and WP-to-WP. For each category, the following information is provided:

- protocol type (identifies the type of protocol used in a particular communications package)
- type of transmission (identifies transmission mode, whether synchronous, asynchronous, half or full duplex and the type of low-level line protocol, if any)
- error protection (identifies types of error protection available with a particular communications package)
- line speeds (lists the range of available line speeds)
- code sets (lists the possible ways in which data can be represented on the communications line)
- handling of text characters (describes how text characters are processed when moving from file storage to communications line)
- handling of control characters (describes how control characters are processed when moving fromm file storage to communications line)
- handling of format information (describes how format information is processed when moving from file storage to communications line)
- protocol behaviour (describes briefly the characteristics of a particular protocol as implemented by each manufacturer; there are differences between each machine which can lead to potential data loss)

- special features (lists some of the general characteristics of the communication package as seen by the operator)

Of the above items, those concerning error protection, line speeds and special features do not affect in any significant way the chances of two different machines talking to each other. They are included for the sake of completeness. The remaining items will serve as input to the discussions of the following three chapters.

Before proceeding with the profiles, it is appropriate to define some of the terms appearing in the text:

XON/XOFF refers to the use of special characters at the communications line level to control the flow of data across the link. Typically, a received XOFF tells the transmitter to temporarily halt transmission while a received XON notifies it to resume transmission activity. There are a number of variations in the use of these flow control characters.

Echoplex refers to a mode of transmission where transmitted characters are expected to be echoed back by the receiver and received characters are echoed back to the transmitting station. This mode is used typically in interactive terminal to mainframe communications.

Transparency refers to a mode of operation where transmitted and received characters are not interpreted by the low-level communications protocol. Link control characters are identified by a preceding DLE (data link escape) character. This mode allows transmission of 256 different 8-bit data characters.

Non-transparency refers to a mode of operation where certain characters are reserved for use as low-level communications control characters and cannot be used as data.

CRC-16 refers to 16-bit cyclic redundancy check, which is used to detect errors in a character stream. It consists of two bytes appended to the end of a character stream and is obtained as the remainder of a division of the character stream (treated as a dividend) by a standard polynomial $(X^{16} + X^{15} + X^2 + 1)$

LRC refers to Longitudinal Redundancy Check which is used to detect errors in a character stream. It consists of a single byte appended to the end of a character stream and is obtained in the same manner as the CRC-16 using the polynomial $X^8 + 1$

VRC refers to Vertical Redundancy Check which is an odd parity check performed on each character.

BSC refers to IBM's Binary Synchronous Communications link protocol which provides an error-free synchronous half-duplex communications facility.

FILE HEADER a string of binary data (as opposed to text) stored with file; it generally contains information descriptive of file contents as a whole.

4.2 AES Plus

There are three communications options available with this machine. They are as follows:

4.2.1 TTY Emulation

1) Protocol type: XON/XOFF optional
2) Type of transmission: Asynchronous, half-duplex, full-duplex, echoplex
3) Error protection: Odd, even, no parity
4) Line speeds: 50 to 9600 bps
5) Code sets: Typically ASCII-7; however a code translation table is available which converts internal 8-bit codes into output characters between 5 and 8 bits long; transparent and non-transparent operation is possible.
6) Handling of text characters: The underscore is transmitted as a character-backspace-character sequence. The handling of the remaining characters is a function of the translation table. For ASCII-7, characters outside the standard 94-character set are normally converted to the null character.
7) Handling of control characters: This is a function of the translation table and of the selection of transparent versus non-transparent operation. In typical operation, i.e., non-transparent ASCII-7, the end-of-line character is represented either as CR or CR-LF; the end-of-page character becomes Form Feed; the horizontal tab character becomes a sequence of spaces; most other control codes are converted to the null character. In transparent 8-bit operation, all control codes can be preserved.
8) Handling of format information: All format information, because it is stored in the file header, is lost except when transmitting in transparent 8-bit mode between two AES Plus machines. As a result the received data is stored with header information corresponding to the format settings at the received terminal.

9) Protocol Behaviour: The file data is transmitted as a stream of characters with possible XON/XOFF flow control exerted between lines. Variations on the XON/XOFF protocol is possible in that the sequence and representation of flow control characters is user selectable. All data flowing between disk and communications passes via video memory; this step requires an additional code translation and may result in data loss.

10) Special features: Foreground operation, automatic overflow from diskette to diskette on reception, parameterization of communications package (this allows the user to easily tailor the package to suit local requirements), interactive and pseudo-batch operation possible.

4.2.2 Batch Terminal Emulation

1) Protocol type: IBM 3780
2) Type of transmission: Synchronous half-duplex, BSC (Binary Synchronous Communications) line protocol
3) Error protection: CRC-16 (8-bit), VRC, LRC (7-bit)
4) Line speeds: 300, 600, 1200, 1800, 2000, 2400 bps
5) Code sets: EBCDIC and USASCII, transparent and non-transparent
6) Handling of text characters: Translation tables allow the user to specify which characters can be transmitted and how. The user may select one of three possibilities for each character:

   (i) transmit as a single character (1 to 1 conversion)
   (ii) transmit as an overstrike sequence (1 to m conversion)
   (iii) do not transmit

   Lines of text containing underlined non-space characters or composite characters may be handled on transmission in four possible ways:

   (i) send only first character of sequence; remaining characters in sequence are lost (e.g., underscore)
   (ii) transmit backspace sequence
   (iii) transmit separate lines
   (iv) transmit separate lines preceded by printer line skip suppress sequence

7) Handling of control characters: Both the EBCDIC and ASCII code sets in transparent mode allow the unrestricted transmission and reception of all 256 possible codes. The transmission of control codes is not possible with the ASCII 7-bit code set in non-transparent mode. When the AES machine is receiving as a line printer, control codes affecting printer behaviour may selectively be treated as commands or data.

8) Handling of format information: There is no provision in the 3780 protocol for the transmission of formatting information apart from the printer horizontal control feature. This feature allows the transmission of tab stop information as a special escape sequence. However, this sequence must be embedded in the textual material to be transmitted and is used only as a convenience feature to reduce the number of blank spaces that must be transmitted. Received data is stored on disk with the header information corresponding to the format settings at the receiving terminal.

9) Protocol behaviour: The 3780 package allows point-to-point connection over dedicated or switched lines. When transmitting, it emulates a card reader; in non-transparent mode, it transmits records up to 80 characters in length (this may or may not be fixed). In transparent mode, it may transmit either one or six 80-character records per block. Space compression is available for non-transparent text.

For reception, the machine may be selected to act as a line printer or a card punch. If selected as a line printer records up to 256 characters long can be received. In transparent mode end-to-end control characters (part of the BSC protocol) and vertical format commands (escape sequences) may optionally be treated as commands or data. In card punch emulation, an end-of-line is inserted in the file after every received 80-character record. An optional required carriage return may be appended as the last character of each received line. Other characteristics of the protocol are conversational mode, auto-restart after reception of a Reverse Interrupt (RVI) and the acceptance of terminal identification sequences. Finally, all data flowing between disk and communications passes via video memory; this requires a code translation with potential data loss.

10) Special features: The 3780 emulation package is available only with systems having an installed extended disk board; features include foreground operation, unattended operation with automatic disconnect, and parameterization of the package.

4.2.3 AES Point-to-Point

1) Protocol type: AES private (modified BSC)
2) Type of transmission: synchronous half-duplex, modified BSC line protocol
3) Error protection: CRC-16
4) Line speeds: 300 to 2400 bps on switched network 300 to 4800 bps on private line
5) Code sets: AES internal 8 bit
6) Handling of text characters: Because the AES internal 8-bit code is transmitted unaltered, all text characters can be transmitted and received successfully
7) Handling of control characters: All control characters can be transmitted and received successfully
8) Handling of format information: The private protocol permits the transmission of all format information
9) Protocol behaviour: The protocol is based on the BSC line protocol but is altered to suit the requirements of a point-to-point protocol between similar machines. It allows complete exchange of informationn between two AES Plus machines and a limited exchange facility between the AES Plus and the AES 100 P/B or AES 90 machines. All data flowing between disk and communications passes via video memory; this requires a code translation with potential data loss.
10) Special features: Foreground operation, unattended operation with auto-answer, password protection, remote access to files, operator messages.

4.3 IBM OS/6

The IBM Office System 6 provides a single communication package, as described below:

4.3.1 Batch Terminal (2770) Emulation

1) Protocol type: IBM 2770
2) Type of transmission: Synchronous, half-duplex, BSC line protocol
3) Error protection: CRC-16
4) Line speeds: 1200, 2000, 2400 bps
5) Code sets: 7-bit ASCII, EBCDIC, EBCDIC/WP, transparent and non-transparent
6) Handling of text characters: The EBCDIC/WP code set table has a fixed number of positions for representing text characters. However, some of these positions may represent more than one character. Identification of the proper characters to be associated with these positions is provided by a keyboard ID. There are 53 different keyboard layouts available for the OS/6. Therefore, it is essential that keyboard identification information be transmitted with each file. Assuming that all text characters in a file have been generated using the associated keyboard, then it is possible to transmit all the text characters. It is possible to change keyboard ID and thus the character set within a file.
7) Handling of control characters: Most of the internal control codes have unique EBCDIC/WP representations and can thus be transmitted. Others (typestyle and keyboard ID changes) are represented by a collective control code (the STOP code). The EBCDIC and 7-bit ASCII sets are limited to the standard characters. Control codes which are not among these standard characters are either mapped to similar codes or to the null character.

8) Handling of format information: Text formatting information stored with a file may optionally be transmitted in the form of special text commands which form part of an "Operator Control Language" (OCL). Thus a transmitted text may consist of both text characters and special commands which control the formatting of the text. Among the available commands are instructions defining tab stops, line spacing, keyboard ID and justification. These commands may affect all or just a portion of a transmitted file. It is therefore possible to incorporate format changes within the body of a text. The OCL commands are available with all code sets, although mixing of code sets within a file may lead to unpredictable results.

9) Protocol behaviour: The 2770 emulation is designed to allow communications with IBM mainframes and other OS/6 machines as well as the Mag Card II. There are a number of user-selectable options which govern the behaviour of the protocol. Among these are:

- block sizes of 128, 256 or 512 characters
- transmission of file in media image (i.e., as it appears on disk) or in page image (i.e., as it would appear when printed)
- the transmission of format information
- output device selection
- transparency
- record length; this is related to transparency and block size. In non-transparent mode, each block will consist of an integral number of complete lines (as determined by line ending codes) up to the block size limit. If transparency is selected, all records are 80 characters long and line ending codes may or may not be transmitted
- transmission of page end code

The emulation does not handle horizontal format control and only a limited number of vertical format control escape sequences. It does not perform space compression during transmission but will receive compressed spaces without difficulty.

10) Special features: Foreground operation, levels of security, multiple outstanding transmission requests.

4.4 MICOM 2000/2001

The MICOM 2000/2001 currently has three communications packages available, as discussed below.

4.4.1 TTY Emulation

1) Protocol type: TTY; XON/XOFF optional
2) Type of transmission: Asynchronous, half or full duplex
3) Error protection: Odd, even or no parity
4) Line speeds: 50 to 1200 bps
5) Code sets: ASCII, 2741 Correspondence, 2741 EBCD; 6, 7 or 8 bits
6) Handling of text characters: Only those characters which are members of the appropriate code sets are transmitted successfully. All other characters are transmitted as question marks (?). Underlines are transmitted as a character-backspace-character sequence.
7) Handling of control characters: Only those control characters which are members of the appropriate code sets (i.e., horizontal tab) are transmitted. The end-of-line character is translated into CR plus an optional additional character. The horizontal tab character may be transmitted as is or as an appropriate sequence of spaces.
   The remainder are transmitted as question marks (?).
8) Handling of format information: Format information is not transmitted. The output consists strictly of a string of text characters.
9) Protocol behaviour: This package is designed primarily for interactive communications with mainframes. In addition, it provides a file transfer capability. The use of a user-modifiable communications profile allows the user to tailor the behaviour of the package to a particular environment. Among the characteristics which are modifiable are:

   - operation of flow control (XON/XOFF)
   - nature of transmitted and received end-of-line character(s)
   - representation of rubout character
   - ability to record in a disk file all transmitted and received text
   - wrap-around of received characters at right column limit
   - right column limit
   - representation of tab key (HT character or spaces)
   - tab settings

10) Special features: Foreground/Background Operation, line monitoring capability, emulation of Digital Equipment Corporation's VT52 terminal, transmission/reception directly from/to diskette.

4.4.2 Batch Terminal Emulation

1) Protocol type: 2780/3780

2) Type of transmission: Synchronous half-duplex, BSC line protocol

3) Error protection: CRC-16

4) Line speeds: 600, 1200, 1800, 2000, 2400 bps

5) Code sets: EBCDIC

6) Handling of text characters: The translation table provided with the package allows the transmission of the standard EBCDIC set. However, it is possible to define a supplementary table for defining characters outside the standard set. Characters which are not defined in either set are converted to spaces on output.

7) Handling of control characters: A supplementary translation table can be defined by the user to enable the transmission of control codes having a visual representation. Otherwise, control codes are converted to spaces on output.

8) Handling of format information: The 2780/3780 protocols have no provision for the transmission of format information apart from the horizontal format control feature. The format information stored with a received file defaults to the format settings at the receiving terminal.

9) Protocol behaviour: The 2780 version of the emulation supports the following product features:

   auto answer
   auto turnaround
   component selection
   EBCDIC transparency
   horizontal format control
   144 character print line

   Options not supported are: multipoint line control, synchronous clock, multiple record transmission, dual communication interface, ASCII and transcode character sets.

The 3780 version of the emulation supports the following features:

component selection
conversational mode
EBCDIC transparency
processor interruption
space compression
switched network control

Options not supported are multipoint data link control, ASCII character set and synchronous clock.

During transmission, the terminal behaves as a card reader. Text must be formatted with a column limit of 80 characters. In 2780 emulation, two 80-character records are transmitted per block in both transparent and non-transparent modes. In 3780 emulation, one 80-character record is sent per block in transparent mode and up to six records in non-transparent mode.

During reception, the terminal behaves either as a line printer or a card punch. In either case, received data goes to disk. During printer emulation, the column limit is anywhere from 80 to 250 characters and is specified by the receiver. Required carriage returns are placed at the end of each received line. When a line wider than the column width is received, characters at the end of the line overwrite each other. Horizontal format control using tabs is supported. All printer control escape sequences are accepted by the 2780/3780 emulation; they are interpreted on the basis of a standard 66 line by 80 column page.

4.4.3 WP-to-WP

1) Protocol type: MICOM private (modified BSC)
2) Type of transmission: Synchronous half-duplex, modified BSC line protocol
3) Error protection: CRC-16
4) Line speeds: Up to 2400 bps
5) Code Sets: MICOM internal
6) Handling of text characters: Because the MICOM internal 8-bit code is transmitted unaltered, all text characters can be transmitted and recovered successfully
7) Handling of control characters: All control characters can be transmitted and received successfully
8) Handling of format information: The private MICOM protocol permits the transmission of all format information

9) <u>Protocol behaviour</u>: The protocol is based on the BSC line protocol but is altered to suit the requirements of MICOM-to-MICOM transfer of text files. No information is lost during this transfer.

10) <u>Special features</u>: Background and unattended operation, auto-answer, remote access to files.

4.5 Wang Word Processor 5

The Wang Word Processor 5 provides three separate communications options. These are as follows:

4.5.1 TTY Emulation (called Asynchronous Communications Option by Wang)

1) Protocol type: TTY or IBM 2741

2) Type of transmission: Asynchronous, half-duplex

3) Error protection: Parity checking

4) Line speeds: 110 to 1200 bps for TTY
134.5 to 1200 bps for 2741

5) Code sets: 7-bit ASCII for TTY, 6 bit IBM correspondence code for 2741

6) Handling of text characters: Because only 7 and 6-bit codes are available for transmission, only a limited set of characters can be handled. Other characters outside these code sets are translated into similar or space characters on output. Underlines and accents are stripped off. However because the translation table is user-accessible it may be modified to suit the user's needs.

7) Handling of control characters: All control codes not part of the line code sets are translated into spaces during transmission. However, the translation table is user modifiable.

8) Handling of format information: The Wang asynchronous communications software does not allow the transmission of format information. The output consists strictly of a string of text characters.

9) Protocol behaviour: This protocol is designed principally for interactive communications with mainframes. However, it does permit a limited file transfer capability with little protection and no error recovery procedures.

10) Special features: Selectable line speed, interactive operation.

4.5.2 Batch Terminal Emulation

1. Protocol type: IBM 2780/3780
2. Type of transmission: Synchronous half-duplex, BSC line protocol
3. Error protection: CRC-16
4. Line speeds: Up to 2400 bps
5. Code sets: EBCDIC transparent and non-transparent
6. Handling of text characters: The standard code translation is designed for the EBCDIC character sets. However, it can be altered to handle additional special characters. Accents and underlines are normally stripped off.
7. Handling of control characters: In normal operation, internal control codes are translated to spaces on output; however, the modifiable code translation table and the transparency option may permit the reception of control codes. Control codes outside the EBCDIC set cannot be transmitted.
8. Handling of format information: The 2780/3780 protocols have no provision for the transmission of format information apart from the horizontal format control feature. The format information stored with a received file defaults to the format settings at the receiving terminal.
9. Protocol behaviour: The Wang 2780/3780 emulation closely resembles the original IBM specifications. There is no difference in the behaviour of the emulation in transparent and non-transparent modes apart from the use of the "DLE" character for identifying link control characters in transparent mode. In either case, all output records consist of 80 characters with end-of-line characters removed. When behaving as a line printer during reception, line widths of 80 or 132 characters are acceptable. End-of-line characters are placed at the end of every received line. Space compression is implemented while the horizontal format control feature is not. Only a limited set of vertical format control sequences are accepted. These include the single space, double space, triple space and form feed escape sequences. Switched network control is included and terminal identification sequences are accepted. Format information stored within text is not transmitted.

10) Special features: Foreground operation, multiple outstanding transmission requests, auto-dial, unattended and auto-answer operation

4.5.3 WP-to-WP (called WPS Protocol by Wang)

1) Protocol type: WANG private (modified BSC)
2) Type of transmission: Synchronous, half-duplex, modified BSC line protocol
3) Error protection: CRC-16
4) Line speeds: Up to 2400 bps
5) Code sets: WANG internal
6) Handling of text characters: Because the WANG internal code set is transmitted unaltered, all text characters can be transmitted and received successfully
7) Handling of control characters: All control characters can be transmitted and received successfully
8) Handling of format information: The private WANG protocol permits the transmission of all format information
9) Protocol behaviour: The protocol is based in the BSC line protocol and preserves format lines, internal codes and passwords during transmission
10) Special features: Password protection, foreground operation, multiple outstanding transmission requests

4.6 XEROX 850

The XEROX 850 offers the widest variety of communications packages, which include TTY emulation, two batch terminal emulations plus synchronous and asynchronous versions of a private point-to-point protocol.

4.6.1 TTY Emulation

1) Protocol type: TTY with optional XON/XOFF
2) Type of transmission: Asynchronous, half-duplex or echoplex
3) Error protection: Odd, even or no parity
4) Line speeds: 110 to 1200 bps
5) Code sets: 7-bit ASCII
6) Handling of text characters: Only characters which are members of the ANSI 68 ASCII set can be transmitted. Among the other characters, some are translated where possible into similar characters and others are not transmitted at all. Underlined characters are transmitted as character-backspace-underline sequences.
7) Handling of control characters: Those control characters which cannot be folded into the standard ASCII set are not transmitted. On reception, special escape sequences are interpreted as control characters.
8) Handling of format information: Format information in format blocks cannot be transmitted. On reception special escape sequences delimit received format information.
9) Protocol behaviour: Asynchronous link to a mainframe for interactive operation; includes a file transfer capability.
10) Special features: Selectable parity and line speed, line monitoring capability, interactive operation, unattended operation, automatic disk switching on overflow during reception, emulation of GE Terminet and Hazeltine terminals.

4.6.2 Batch Terminal (2780) Emulation

1) Protocol type: IBM 2780
2) Type of transmission: Synchronous, half-duplex, BSC line protocol
3) Error protection: CRC-16
4) Line speeds: Up to 2400 bps
5) Code sets: EBCDIC, XEROX internal
6) Handling of text characters: When the EBCDIC code set is used only those characters which are part of the standard set can be transmitted. Underlines are discarded. When the XEROX internal code set is used, all text characters can be sent and received. The translation tables are not user accessible.

7) Handling of control characters: When the EBCDIC code set is used in transparent mode, only those control characters which are part of the standard set will be transmitted. When the XEROX internal set is used, all control characters can be sent and received.

8) Handling of format information: The 2780-protocol has no provision for the transmission of format information apart from the horizontal format control feature. When a horizontal sequence is received in non-transparent mode, a new format block is generated in the file which reflects the specified tab stops. However, the remaining formatting information is lost. When transmitting in transparent mode, the format information contained in the "format control block" is preserved. This format control block is part of the file text and contains such information as tab settings, character settings and justification selection. Because this information can be transmitted, very little information is lost during file transfers in this mode. The only information which is lost is page numbering data which is contained in the actual file header.

9) Protocol behaviour: The 2780 emulation includes all features of the 2780 protocol except for multipoint line control and the ASCII and correspondence character sets. When transmitting it behaves as a card reader and on reception it accepts vertical format control escape sequences. In non-transparent mode, every record is up to 80 characters long with end-of-line characters stripped and every received record is considered as a line of text. In transparent operation, the system performs a straightforward dump from one machine to another; no data is lost and the format of the text is unaltered. The maximum line length is 156 characters and is acceptable for both transmission and reception.

10) Special features: Background and unattended operation, auto-answer, multiple outstanding transmission requests, automatic disk switching on overflow during reception.

4.6.3 Batch Terminal (2770) Emulation

1) Protocol type: IBM 2770

2) Type of transmission: Synchronous, half-duplex, BSC line protocol.

3) Error protection: CRC-16

4) Line speeds: Up to 2400 bps

5) Code sets: 7 bit ASCII, EBCDIC, EBCDIC/WP

6) Handling of text characters: When the ASCII, EBCDIC and EBCDIC/WP code sets are used, only those text characters which are members of these sets may be transmitted. When the ASCII code set is used, underlined characters are represented by backspace sequences. Underlines are represented by a special character when the EBCDIC and EBCDIC/WP code sets are used. The translation tables are not user accessible.

7) Handling of control characters: When the ASCII and EBCDIC code sets are used, only those control characters which are members of these sets may be transmitted. The EBCDIC/WP code set is an extended set with an additional 15 values defined to represent word processing control codes. Thus, when this code set is used, fewer control characters are lost, although some still are.

8) Handling of format information: Format information may optionally be transmitted as an IBM Mag Card II format. On reception a Mag Card II format line is converted internally to an 850 format block. If this option is not selected, format information is not sent.

9) Protocol behaviour: The 2770 emulation is designed to allow communications with IBM mainframes and with IBM OS/6 and Mag Card II machines. It has a user selectable block size of 128, 256 or 512 characters. User options allow the transmission in card image (i.e., 80 character records with line ending codes removed), reception in card image (i.e., Required Carriage Returns are inserted at 80 character intervals), transmission of format blocks (some of the information therein may be lost) and transmission of page end codes.

10) Special features: Background and unattended operation, auto-answer, multiple outstanding transmission requests, automatic disk switching on overflow during reception.

4.6.4 WP-to-WP

1) Protocol type: XEROX private
2) Type of transmission: Asynchronous, half and full duplex; synchronous, half-duplex
3) Error protection: CRC-16
4) Line speeds: 300 to 1200 bps asynchronous; up to 2400 bps synchronous
5) Code sets: XEROX 8-bit internal
6) Handling of text characters: All text characters can be transmitted and received successfully
7) Handling of control characters: All control characters can be transmitted and received successfully
8) Handling of format information: All header and format information can be transmitted and received successfully
9) Protocol behaviour: The protocol is based on the BSC line protocol and allows the transmission of entire file contents
10) Special features: Remote requests, background and unattended operation, auto-answer, multiple outstanding transmission requests, automatic disk switching on overflow during reception.

Chapter 5

Current Interworking Capabilities and Problems

5.1 Introduction

In this chapter the extent to which the various word processors can interconnect is explored. It was pointed out in Chapter 4 that all available communications packages fall into two general categories: 1) Asynchronous, TTY-level communications packages for low-volume, interactive applications, and 2) Synchronous, BSC-based communications options for higher-volume batch-type situations. This latter category includes both batch terminal emulations and machine-to-identical-machine packages. In this chapter, the level of interconnection for the five CWPs under consideration will be discussed for each category of communications facility. Each CWP will be examined in turn, in Section 5.2 for the asynchronous category and in Section 5.3 for the synchronous one. The general characteristics of communication with other CWP's will be highlighted with an emphasis on areas and causes of information loss. This will lead to a general discussion in Section 5.4 of the problems currently facing anyone who is contemplating the possibility of exchanging files among word processors. A more detailed discussion of CWP interconnection will be found in Appendix D.

5.2 TTY Level

All machines except the IBM OS/6 provide an asynchronous communications package. The four machines which do are capable of transmitting the ASCII code set; in some cases they can also emulate the IBM 2741 protocol. Although all asynchronous packages are designed primarily for interactive communications with remote machines (usually a mainframe), they include as well a capability to transmit and receive disk files. It is this latter capability which is of interest here. In the following sub-sections, the extent to which file information can be transferred from machine to machine using asynchronous communications will be reviewed.

Table 5.1 summarizes the behaviour of asynchronous links between various combinations of machines. Further details are provided in Sections 5.2.1 through 5.2.4 below.

5.2.1 AES Plus

The AES Asynchronous Communications Package is very flexible in that it can transmit up to 8-bit character codes and its translation table is user modifiable. This means that all internal codes can be transmitted and represented on the communications line by whatever bit combinations the user chooses.

Further, this package has a special provision for communicating with another AES Plus such that the internal AES code set can be transmitted untouched. As a result, all text and control codes are preserved, although format information cannot be transmitted because it is not stored as a text string. However, a source of data loss is the use of 7-bit memory for temporary storage during communication (refer to Appendix D, Section D.2.1 for more details).

Communication with the MICOM, WANG and XEROX machines is possible because all of these support the 7-bit ASCII code set. However, the ASCII communications packages offered by these manufacturers are less flexible than the AES offering. With minor variations, the AES package can be made to behave in a manner suitable to each of these machines. Then interconnection between the AES and other machines all have similar characteristics: text characters are limited to the standard 7-bit ASCII set, underlines are sometimes preserved (MICOM, XEROX) and sometimes lost (WANG), characters outside the ASCII set and control characters are discarded or changed to some other character, and format information is lost. Also, text reorganization may take place if there are page width incompatibilities between machines.

Thus, apart from AES to AES communication, interconnection between AES and other word processors is most useful when the files to be exchanged contain only ASCII characters and when the two machines have identical page width settings. In that case, it should be possible to obtain on paper at the receiving end an exact image of the original file. If editing of the received file is to be performed, the first step should be the manual insertion of end-of-paragraph symbols to ensure that the layout of the text is preserved.

5.2.2 MICOM 2000/2001

The MICOM asynchronous package allows the transmission of the 7-bit ASCII set of characters only; all other characters found in a file are transmitted as the question mark (?) character. Nor is there any provision for transmission of format information.

The MICOM package is parameterized which enables it to adapt to many configurations; however, many of the parameters have greatest significance when the MICOM CWP is communicating with mainframes. File transfers between MICOM and other machines are most effective when only ASCII characters are involved and when page widths are compatible. As with asynchronous communications involving the AES Plus, preliminary editing of received files is recommended if any processing beyond printing is envisaged.

5.2.3 WANG Word Processor 5

Of the two available code sets (ASCII and 2741 correspondence code), the 7-bit ASCII code set contains more characters and it is therefore the preferred choice. However, it does not allow the transmission of many text and control characters or of any format information. As a result, its suitability is restricted to the transfer of print images of file contents where no special text or control characters are involved (including underscores). If a received file is to be modified, preliminary editing to define paragraph endings should be done in order to avoid destroying the layout of the text.

5.2.4 XEROX 850

XEROX has implemented a private asynchronous protocol designed for the transfer of files between two XEROX machines. All information is transmitted in the XEROX internal code set using a BSC-based protocol. All text, control and format information is preserved.

When communicating with CWP's of other manufacturers, an ASCII communication package is available which behaves in much the same way as the packages available from the other vendors. Thus, it allows the successful transfer of print image versions of files containing only ASCII text. Any additional information such as control or non-ASCII text characters is either lost or altered in transit.

TABLE 5.1: Characteristics of Interconnections Using Asynchronous (TTY-Level) Protocols

N.B. IBM OS/6 has no asynchronous communications capability

	Text Characters	Control Characters	Format Information	Comments
AES/AES	preserved if compatible character generators; otherwise some characters may be lost	preserved	lost	internal AES code is transmitted
AES/ MICOM	underlines preserved; special characters lost	most lost; tabs sent as spaces	lost	7-bit code
AES/WANG	underlines lost; special characters lost	most lost; tabs sent as spaces	lost	WANG does not transmit underlines 7-bit ASCII code
AES/XEROX	underlines preserved; special characters lost	most lost; AES sends tabs as spaces; XEROX as HT	lost	7-bit code
MICOM/ MICOM	underlines preserved, special characters lost	most lost; tabs sent as spaces or HT	lost	7-bit ASCII code

TABLE 5.1: Characteristics of Interconnections Using Asynchronous (TTY-Level) Protocols (Continued)

	Text Characters	Control Characters	Format Information	Comments
MICOM/ WANG	underlines lost; special characters lost	most lost; tabs sent as spaces	lost	WANG does not transmit underlines; 7-bit ASCII code
MICOM/ XEROX	underlines preserved; special characters lost	lost; TAB sent as HT by XEROX or spaces by MICOM	lost	7-bit ASCII code
WANG/ WANG	underlines and special characters lost	most lost; tabs sent as spaces	lost	WANG does not transmit underlines; 7-bit ASCII code
WANG/ XEROX	underlines and special characters lost	most lost; tabs sent as HT by XEROX, as spaces by WANG	lost	WANG does not transmit underlines; 7-bit ASCII code
XEROX/ XEROX	preserved	preserved	preserved	internal XEROX code transmitted using private protocol

5.3 BSC Level

All five CWPs under consideration offer at least one communications option based on IBM's Binary Synchronous Communications (BSC) line protocol [1]. This protocol is designed to provide a synchronous half-duplex error-free link between two stations connected in a point-to-point or multipoint configuration. It is suitable for both private line and switched network environments. The BSC protocol has served as the basis for device-to-computer communication facilities for a number of IBM products. The IBM 2780, 3780 and 2770 batch terminals each have an individual BSC-based protocol designed to suit their particular needs. The reader is referred elsewhere for descriptions of these protocols [2,3,4]. Over the years, the dominance of these devices in the marketplace has led to the acceptance of their associated protocols as de facto standards for batch terminal interaction with mainframes.

For this reason, one or more of these protocols was adopted by each CWP manufacturer to provide an error-free file transfer capability between its word processors and mainframes. In this way, it became possible for CWPs to hook into large computers and their networks without requiring any special new software on the part of the big machines. However, the 2780/3780/2770 protocols were not designed for the CWP application and have deficiencies which limit their usefulness in CWP to CWP situations. This led some manufacturers, in the absence of standards, to develop their own private protocols to allow them to do whatever they pleased when communicating among their own machines. Such is the all-pervasiveness of BSC that it again formed the basis for each of these private protocols. Although all current such protocols are BSC-based, they are all sufficiently different from each other to make them incompatible. Thus, the powerful private protocols are useful only between identical CWPs; communication between differing CWPs is left to the 2780/3780/2770 emulations.

It is fortunate that the 2780/3780/2770 protocols are similar to each other; in fact, the 2780 protocol is a subset of 3780 and 3780 is in turn a subset of 2770. This allows one machine running a 3780 emulation to exchange files successfully with a machine running the 2770 protocol, as long as the latter machine does not use any non-3780 features.

Not only have all manufacturers implemented their private protocols differently, they also have come up with unique point-to-point emulations of the 2780/3780/2770 protocols. Fortunately, these differences are not large enough to preclude compatibility among machines, but they do affect the degree of information transferability between different CWPs. In Chapter 4, the characteristics of each manufacturer's BSC-based communications packages were described. In the following subsections, the five CWPs under study will be investigated to determine the extent to which interconnection is possible. In most cases, this will demand extensions to manufacturer-supplied translation tables which in turn will require access to manufacturers' internal code sets; if this is not possible, then most synchronous CWP-to-different-CWP communication is reduced to the level of restricted print image transfer of files, with an effectiveness similar to asynchronous communications packages.

Table 5.2 summarizes the characteristics of synchronous interconnections between various combinations of machines. Details are provided in Sections 5.3.1 through 5.3.5 below.

5.3.1 AES Plus

AES has developed its own private BSC-based protocol for synchronous point-to-point communication between two AES Plus machines. When this protocol is used, the entire file contents are transferred successfully. Table 5.2 summarizes this fact. The only possible instance of data loss occurs when two machines with different video character generators are communicating. This can lead to a potential loss of textual information (typically character accents). This protocol cannot be used when communicating with non-AES machines.

When an AES Plus is to communicate with a dissimilar CWP, its 3780 emulation package is used. This package enables it to communicate with all four of the other CWP's in this study, although each combination has its own particular characteristics. In general, it is possible to exchange files containing compatible control and text characters between machines having compatible page widths (80-character lines is the universally accepted size). It is not possible to transfer format information.

The extent to which control characters beyond those contained within the standard EBCDIC set can be exchanged depends on the connected machine. IBM and XEROX allow up to 15 control characters as defined in the EBCDIC/WP code set. MICOM and WANG give the user access to their translation tables but impose restrictions which limit the number of such characters that can be transmitted.

Access to the translation activity allows the inclusion of additional text characters during file tranfers involving the WANG and MICOM WP's. The IBM and XEROX terminals do not give the user such access to their translation tables and therefore limit the range of transmission codes available for textual data.

The use of a 2770 emulation in the IBM and XEROX emulations puts no restraints on the organization of transmitted text, unlike the 3780 emulation available with WANG and MICOM.

Communication with other CWP's is generally more effective using BSC-based protocols than is possible using asynchronous protocols because more control and text characters can usually be exchanged. However, it is not possible to transfer format information, nor all possible control and text characteters; in addition, some protocols impose restrictions on text layout. As a result, there is only a limited probability of being able to transfer a file that is an exact duplicate of the original. Some editing is usually required.

5.3.2 IBM OS/6

Unlike the other vendors in this study, IBM has not implemented a special protocol for communication among its machines. Instead it uses the 2770 protocol with an extended code set, the EBCDIC/WP set (see Appendix C). This set has 15 additional control codes defined for it. This enables the machine to transfer most of its internal control functions. Thus it is possible for two IBM OS/6 machines to exchange in most cases file contents without data loss; all format information, all code set information and almost all control codes can be transferred.

Of the other manufacturers, only XEROX offers a 2770 emulation package. An IBM-XEROX link is able to exchange most control information using the EBCDIC/WP code set; text characters are limited to the EBCDIC character set and some format information can be exchanged using the Mag Card II format; there are no restrictions on the layout of transmitted files. Thus, it is possible in many cases to transfer exact file images between these two systems because typical text characters, control codes and format information can all be transmitted.

The other three vendors would use a 3780 emulation when talking to an IBM machine. This protocol limits the range of text and control characters that can be transmitted, precludes the possibility of exchanging format information and imposes restrictions on the layout of transmitted text. The AES and MICOM machines allow the transmission of additional control codes beyond the basic EBCDIC set such that most if not all of the 15 additional codes of the EBCDIC/WP set can be handled. Further, the AES implementation of the 3780 protocol is less restrictive in its data layout requirements so that transmission and reception of variable length lines is possible. The result is that varying degrees of performance are obtained when an IBM CWP communicates with dissimilar CWP'S. The common level of functionality is the exchange of files in 80-column form which contain only standard characters and compatible control codes.

5.3.3 MICOM 2000/2001

MICOM has developed its own private BSC-based protocol for synchronous point-to-point communication between two MICOM machines. It provides a complete file transfer capability with no information loss. This protocol is not compatible with any other.

When communicating with dissimilar CWP's, either a 2780 or 3780 emulation may be used. When the WANG and AES machines are involved, a 3780-3780 link would be used, while a 3780-2770 connection would be the most attractive choice when IBM and XEROX are considered. In all cases, the ability to increase the number of control codes that can be handled by making additions to the translation table will augment the information transfer capability of these connections. A further improvement is achievable when the AES and WANG machines are involved by increasing the number of text character codes that the translation table can process correctly. This second improvement is not possible with the IBM and XEROX machines because these translation tables are not user accessible.

The MICOM 3780 implementation places restrictions on the format of the text that it transmits and receives. This reduces the effectiveness of all communications involving this machine. Further, it is not possible to exchange any format information. The result is that file transfers between MICOM and a dissimilar CWP are most successful when the files are in 80-column form and contain only compatible text and control characters. When these conditions are adhered to, then a synchronous BSC-based communication link can be more powerful than an asynchronous one.

5.3.4 WANG WORD PROCESSOR 5

WANG has developed its own private BSC-based protocol for synchronous point-to-point communication between two WANG machines (called WPS protocol by WANG). It provides a complete file transfer capability with no information loss between these machines. This protocol is not compatible with any other.

WANG offers both a 2780 and 3780 emulation for communicating with dissimilar word processors. As with the AES and MICOM emulations, the translation table can be modified to suit particular requirements. However, special control characters are stripped off by the WANG system before they reach the transmit translation table so that it is not possible to transmit more than the basic set of control characters available with the EBCDIC code set. Thus on the transmission side, the translation table can be modified only to increase the number of text characters that can be transmitted. On the reception side however, the translation table could be expanded to correctly interpret additional text and control characters.

Communications with the AES and MICOM CWP's would involve a 3780 to 3780 link while IBM and XEROX would use their 2770 emulations to talk to the WANG machine. In all cases, there are restrictions on the layout of transmitted and received text and no format information can be exchanged. The result is that file transfers between WANG and other CWP's are most successful when the files are in 80-column form and contain only compatible text and control characters. In these circumstances, more information can be exchanged with a WANG machine using the 3780 emulation than is possible using with the asynchronous option.

5.3.5 XEROX 850

XEROX has developed its own private BSC-based protocol for synchronous point-to-point communication between two XEROX machines. It provides a complete file transfer capability with no information loss. This protocol is not compatible with any other.

XEROX offers two different packages for communication with other CWP's; they are a 2780 and a 2770 emulation. The 2770 emulation was designed primarily for purposes of compatibility with the IBM OS/6 but it turns out to be applicable to communicate with the products of other vendors as well. In fact, it is more suitable than the 2780 emulation for talking to the CWP's involved in this study. This is because it allows additional control functions to be transmitted and is more flexible than the 2780 emulation in terms of text layout requirements.

Communication with the IBM OS/6 is most effective of all combinations because of the use of compatible control codes (from the EBCDIC/WP set) and because some format information can be exchanged in Mag Card II form. No text characters beyond the standard EBCDIC set can be exchanged.

No other machine can exchange format information with the XEROX 850.

Both the AES and MICOM machines can exchange some or all of the control codes in the EBCDIC/WP set while the WANG CWP should be able to receive some. In all cases, text characters are restricted to the EBCDIC set.

In terms of text layout requirements, XEROX and IBM are identical and XEROX is more flexible than the other three. In general, 80-column files are guaranteed to be transferred successfully; exchanges involving files with different text layouts are less predictable.

Table 5.2 Characteristics of Interconnections Using Synchronous (BSC-based) Protocols

	Text Characters	Control Characters	Format Information	Protocol
AES-AES	preserved	preserved	preserved	private AES protocol
AES-IBM	underlines and special characters lost	many preserved	lost	AES uses 3780 IBM uses 2770 (non-transparent)
AES-MICOM	underlines lost; compatible special characters preserved	many preserved	lost	both use 3780 (transparent)
AES-WANG	underlines lost; compatible special characters preserved	many preserved	lost	both use 3780 (transparent)
AES-XEROX	underlines preserved; compatible special characters preserved; possible contamination of text	compatible control codes preserved	lost	AES uses 3780 XEROX uses 2780 (transparent)
IBM-IBM	preserved	most preserved	preserved	2770 (non-transparent)
IBM-MICOM	underlines lost; special characters lost	some preserved	lost	IBM uses 2770 MICOM uses 3780 (transparent)
IBM-WANG	underlines lost; special characters lost	some preserved	lost	IBM uses 2780 WANG uses 3780 (transparent)
IBM-XEROX	underlines preserved; special characters lost	most preserved	most preserved	both use 2770 (non-transparent)

Table 5.2 Characteristics of Interconnections Using Synchronous (BSC-based) Protocols (Continued)

	Text Characters	Control Characters	Format Information	Protocol
MICOM-MICOM	preserved	preserved	preserved	private MICOM protocol
MICOM-WANG	underlines lost; compatible special characters preserved	some preserved	lost	both use 3780 (transparent)
MICOM-XEROX	underlines lost; special characters lost	some preserved	lost	MICOM uses 3780 XEROX uses 2770 (non-transparent)
WANG-WANG	preserved	preserved	preserved	private WANG protocol
WANG-XEROX (1)	underlines lost; compatible special text characters preserved; possible text contamination	compatible codes preserved	lost	WANG uses 2780 XEROX uses 2780 (transparent)
WANG-XEROX (2)	underlines lost; special characters lost; no text contamination	some preserved	lost	WANG uses 3780 XEROX uses 2770 (non-transparent)
XEROX-XEROX	preserved	preserved	preserved	private XEROX protocol

5.4 Problems

The foregoing makes it clear that while communications between identical machines is satisfactory, that between different CWPs is far from ideal. The problems that currently exist in this environment stem from two principal causes: (1) Inherent machine differences, and (2) Inappropriate communication protocols.

While all machines involved in this study are based on 8-bit microcomputers, and all provide similar word processing functionality, there exist significant differences in the approaches adopted by the various manufacturers in pursuit of their similar goals. These differences are reflected in each of the three logical components of a CWP: the text characters, the control functions and the format information. Each manufacturer has dealt with these items in unique ways such that there is no direct compatibility for any of these components among the various machines. Clearly, this does not make the task of exchanging information between two machines a simple one; the problem is compounded by the fact that the protocols available at the present time for achieving this goal are as a rule inappropriate for the desired function. They can limit the range of data that can be communicated and enforce data grouping restrictions such that both the content and organization of transmitted information is adversely affected.

The problems that have come to light in the discussions of Sections 5.2 and 5.3 are summarized and discussed further in Section 5.4.1 through 5.4.4 below. They have been grouped according to four principal factors: Code set translation, control function representation, format information representation and protocol deficiencies.

5.4.1 Code Set Translation

There are two sources of difficulty in this area: the total set of text characters involved in the word processing environment, and their representation on the communication line.

Each of the five CWPs considered here handles a different set of characters. Each terminal, because of the 8-bit limitation of the machine, can represent internally up to 256 different characters. However, a certain number of these 8-bit codes is allocated to the representation of control functions. This number varies from manufacturer to manufacturer; thus a variable number of codes are available for text characters. In some cases the set of different characters is limited (e.g., WANG has fewer than 128 different characters because each character underlined is represented by a different code); in others, it is quite extensive (AES, XEROX, MICOM). In at least one case (the IBM OS/6) the set of available codes is not sufficient to represent all the possible printing characters. In this case, IBM has adopted a scheme whereby font information is stored with the text to identify the proper interpretation of stored character codes. Thus the character 'V' in one font may correspond to the character 'X' in another.

It is obvious then that two machines which have different sets of text characters are doomed to some information loss when they communicate. This problem cannot be resolved until all manufacturers agree on a standard set of characters and on a standard way of representing them.

Another problem involving possible loss of text information is peculiar to the AES machine. The video memory inside each AES Plus is capable of storing only 7-bit codes and therefore is restricted to 128 different characters. However, the total number of text characters that are part of the internal AES code set is greater than 128. Each AES machine contains one of four character generators, each capable of displaying 128 different characters; each text code that is part of the internal code set can be displayed by at least one of these character generators. For each character generator, there is an appropriate conversion table which identifies which codes can be displayed correctly. Unfortunately, during communications activity, data that is transmitted/received is buffered in video memory on its way from/to the disk. As a result, a conversion is required from the internal 8-bit code representation to the 7-bit one of video memory. The result is a potential information loss. As mentioned in Section D.2.1 of Appendix D, this problem does not occur in local word processing because each machine has a matched keyboard and character generator such that all characters generated from the keyboard can be displayed correctly. In a communications environment, however, the remote terminal could easily transmit characters that are inappropriate for the local machine. At present, there is no solution to this problem.

When two machines communicate, one condition for avoiding a loss of information is to ensure that all characters to be transferred are compatible to both machines. However, this is not sufficient. Most ocmmunication protocols limit the number of characters that can be transmitted., This is true for the ASCII (7-bit) and EBCDIC (8-bit) code sets which are the ones in prevalent use. For example, the standard ASCII set has only 94 printing characters which is a far cry form the more than 256 characters possible with the IBM machine.

Differences between and limitations of the various protocol implementations further limit the amount of information which can be transferred. All communication code sets include the backspace character which should permit the transmission of composites such as underlined and accented characters. However, few protocol implementations take advantage of this fact. Some ASCII TTY packages provide a backspace capability for underlined characters (e.g., AES, MICOM) but only the AES 3780 emulation package implements a general composite character translation facility. Since no other CWP provides the same service, it is of limited usefulness in this particular environment.

The code translation problem can be alleviated somewhat (at least with 8-bit code sets such as EBCDIC) by transmitting in transparent mode; in that case, more codes are available to represent additional characters; however, this approach requires access to the manufacturer's internal code sets and translation tables, either or both of which may not be available.

5.4.2 Control Function Representation

Every file in the word processing environment includes both text characters entered by the operator and control characters entered by either the operator or the system. These control characters are interpreted as special by the word processing software and affect the processing of the textual material by the system. They are crucial components of a word processing file.

While there are some control functions which exist in all systems (e.g., end-of-line), many are unique to a particular CWP. Tables 3.1 through 3.5 listed the control functions for each of the five machines and Table 3.6 provided a composite list of all control functions in different machines. It is clear from these tables that there is a significant disparity among the various manufacturers' products in the range of available control functions. In an environment where different machines are communicating, incompatibilities in control function implementation are a serious cause of information loss. If a file on one machine contains embedded characters representing control functions unique to that system, then it becomes impossible to communicate that information; any attempts to transmit that file will result in a received file which is no longer an exact image of the original. In some cases the information loss may be minor - the loss of commands to change character spacing may alter the readability of a text but won't alter its meaning; in other situations it may be severe - the loss of commands to inhibit printing of portions of text may significantly alter the content of the printed product.

This type of information loss is not recoverable as long as there exist differences in functionality among CWPs.

One current source of information loss that should be surmountable is the occasional inability to exchange compatible control function information. The root of this problem lies in the code sets used during communication. The ASCII and EBCDIC code sets define only a limited number of functions such as horizontal tab (HT), form feed (FF), etc.; this is insufficient for the CWP application. The problem is alleviated somewhat when communication takes place in 8-bit transparent mode as is possible with most BSC-based protocols. This allows up to 256 different characters to be transmitted. However, most compatible control functions are represented internally by different codes on different machines so that a translation is required to suit the needs of the individual systems. This necessitates access to the internal translation tables and code sets of each machine; this is not always possible.

A second source of difficulty in this area is the differing representation on different machines of similar control functions. For example, the AES and XEROX terminals represent underline sequences by a "start of underline" character followed by an "end of underline" character at the end of the sequence. IBM on the other hand uses a single control character at the end of each underlined word to signify the underline function. This incompatibility cannot be resolved using a simple one-to-one translation facility. A considerably more intelligent mapping function is required. Another example in the same vein is the use of two control characters by AES to represent a function (e.g., merge) that is represented by a single character in the WANG machine.

Another problem related to function representation is peculiar to the AES machine. It represents certain printer-related functions by a sequence of characters, the first of which is a control character and the remaining are text. This control character is not common to any other system, and so would be lost during communication. However, the remaining characters in the string would be transmitted and considered as data by the receiving terminal. Thus, the original textual information may become corrupted.

5.4.3 Format Information Representation

There is no standardization at all among CWPs as to the nature and representation of format information. This information describes units of text in a general way and may contain such items as

- page size
- margin settings
- tab stops
- special tab stops (e.g., indent)
- keyboard identification
- printer font information
- language identification
- justification selection
- line spacing
- character spacing
- page numbering
- page header text
- page trailer text
- general comments

The units of text affected by a particular format specification may range from an entire page for the AES and MICOM machines to anywhere from a single line to an entire page for the others.

No two manufacturers maintain the same set of items for describing the formatting of text and no two manufacturers represent this information in the same way. For example, the AES Plus places its format information in a file header, i.e., as a non-textual string of data stored as part of a file. The XEROX 850, on the other hand, maintains its format information in embedded blocks of textual data delimited by a special control character.

Because of the wide discrepancies in the representation of this information, no attempt is usually made to transmit it. This of course means that the received file has lost some vital information. Every stored file must have associated with it some information so most machines store a set of default settings with each received file. The exceptions to the above rule are machine to identical machine communications and XEROX to IBM communication using the 2770 protocol. In the first case, all format information can be transferred successfully usually with the help of a private protocol, while in the second, only a subset of format information is transferred. This is achieved by a mapping of the internal format representation to an IBM Mag Card II format which can be interpreted by both systems.

5.4.4 Protocol Deficiencies

The private WANG, MICOM and XEROX protocols provide a full file transfer capability with no information loss for communication between identical machines. When communicating between two AES Plus CWPs, the private AES protocol suffers only from possible text character modification as a result of the use of 7-bit video memory; when communicating between an AES Plus and the AES 100, additional information loss may result due to character set incompatibilities. Communication between two IBM OS/6 machines can be totally successful as long as a few control characters do not exist in the transmitted text.

In all other cases, communication between two different CWPs is fairly severely restricted by protocol deficiencies. When two machines talk at the TTY level using a 7-bit ASCII code set, many special text characters, most control characters and all format specifications are lost. The best possible outcome of such a file transfer is a print image of a file that contains no special text or control characters and that was formatted in the manner expected by the receiver (usually 80 columns and 66 lines).

It is evident that the translation tables used with the various communications packages play a key role in determining the amount of information loss taking place during file transfer. This affects principally the transmission of text and control codes, as discussed in Sections 5.4.1 and 5.4.2. However, there are other limitations with protocol implementations which also hinder the file transfer process.

One of the more obvious deficiencies is the inability to handle format information. The only feature of the 2780/3780/2770 protocols suitable for the transmission of general format information is the horizontal format control feature. This enables one terminal to pass on information about the current tab stop settings. However, the only system to take advantage of this is the XEROX 2780 non-transparent implementation. It will create a new format block with appropriate tab settings whenever a horizontal format control sequence is received. Apart from this, the only other systems which attempt to exchange format information are the XEROX and IBM 2770 packages. These will transmit some of their format settings in Mag Card II format so that they can be interpreted by the other terminal.. Outside of these two situations, no format information is transferred between different machines.

More subtle problems arise from the fact that the 2780/3780 and to some extent the 2770 protocols are designed for communication from a card reader to a mainframe and from a mainframe to a line printer or card punch. As a result, during transmission, 80-character records are to be sent with end-of-line characters stripped off. All emulations can do this, and some force it (MICOM, WANG, XEROX non-transparent 2780). Similarly, on reception, the card punch emulation should place an end-of-line at the end of every received 80-character record. This again is possible with all emulations. The line printer emulation allows reception of wider lines and accepts certain escape sequences as carriage control functions. Problems occur if the transmitted text is not organized in the manner expected by the receiver. For example, a MICOM machine acting as a line printer might be expecting 80-character lines; if a file is received that has lines greater than 80 characters, then characters beyond the 79th will be lost as they all overwrite the last character position. A WANG machine in the same situation will simply insert an end-of-line character after 80 characters and proceed to the next line. The consequence is that the layout of the original text is altered in the received file.

In general, the 2780/3780/2770 protocols were designed to operate with 80-character records and so are most suitable for the exchange of data in 80-column form. When line lengths go beyond this, difficulties usually appear with the result that the received file is no longer an exact image of the original.

Chapter 6

Possible Solutions

6.1 Introduction

It is clear from the discussion of Chapter 5 that the communications packages currently available from the various manufacturers impose considerable limitations on the amount of information that can be transferred directly between dissimilar CWPs. Problems exist in each of the areas of text character handling, control functions, format information representation and communications protocol behaviour. These difficulties in most cases reduce the effectiveness of file transfers between different CWPs to the level of a print image exchange of data formatted in a rigid manner (80 character lines) and containing few control characters and a limited range of text characters. This present state of affairs is far from ideal and there is much room for improvement. To define possible areas where useful changes can be made is the principal goal of this chapter.

That problems do exist when attempting to have different CWPs communicate is well known. Already, some companies have implemented systems that at least partially solve the problem. A report entitled "Communicating Word Processors" prepared by International Resource Development Inc. [5] describes some of the activity in this area. One company, Graphic Scanning Corporation, has developed software interfaces for a large variety of different terminals including word processors, teletypewriters and facsimile machines. These interfaces allow inputs from one terminal to be delivered in printable form to almost any other type of terminal. The network, in effect, performs a translation service for different terminals. The above report also mentions that AT & T is in the process of doing the same thing for its Advanced Communications Service (ACS), but on a much larger scale. Its intention is to interface this message switched service to all terminals which exist in sufficient numbers to constitute a demand. The objective is to handle both data and text terminals, and to provide a communications "highway" thereby giving everyone access to everyone else. The National Bureau of Standards in Gaitersburg, Maryland, is developing an in-house network with the objective of interconnecting its 700 electronic information processing machines, including several text processing devices [6]. Compatibility among devices is achieved by specially-developed interface units which perform required protocol conversions.

A study done by Tymshare Inc., entitled "Communicating Word Processors - An Integration Study" [7], investigated the possibility of having various word processors communicate using an electronic mail program, called Interface, implemented on the Tymnet network. In this environment, each CWP only had to worry about establishing contact with the network's host computer. The host took care of communication with the remote CWP. All communication was asynchronous in nature using TTY-type protocols.

It was found that it was possible to exchange simple text successfully but that problems arose when special text or control characters were part of the transmitted data. These results corroborate the findings of Chapter 5 of this report. The Tymshare report proposes a possible solution to the mismatched code problem, suggesting that the host computer perform a translation function between the sender's and receiver's respective codes. This could be achieved by converting all input characters to a universal code set using a table lookup technique. On output, these universal codes would be translated into the values appropriate to the receiver.

One company, Telesystems Network Inc. (TNI) [8], has already developed a product which performs the functions envisaged by Tymshare. The TNI 303 Protocol Translator is a "black box" which allows two CWPs using different communications protocols to talk to each other. When TTY-level ASCII protocols are used, data transferred is restricted to the standard ASCII character set; characters outside of this are converted into special character sequences which allow easy identification and editing by the recipient. When BSC-based protocols are used, compatible control codes are preserved where possible; this is achieved by converting all incoming data into an internal representation followed by a second conversion to the receiver's code set. Nevertheless, communication is most effective when simple print image file transfers are desired. The protocol translator handles the communications protocols of at least fourteen different manufacturers and software for a total of 37 different machines is currently in preparation.

Section 6.2 of this chapter will deals in more detail with the general characteristics of a translation centre approach to the communicating word processor problem.

The various methods discussed above to interface CWPs have all been based on a desire to take existing products and communications packages and somehow make them communicate. The advantage and disadvantages of this approach will be discussed later but it may be safely said at this time that it may not be realistic to provide a central translation facility for all possible machines. There are currently over 100 different word processors on the market, of which more than 75 have a communications capability; and new CWPs are appearing regularly. The complexity of a central translation centre to handle all of these would quickly take on overwhelming proportions. A more satisfactory long-term solution is required, one that can be adopted by each manufacturer so that successful communications between CWPs becomes possible without the help of a central translation facility.

One such solution involves the creation of a "virtual word processor". This pseudo-device would incorporate the characteristics of all existing word processing machines so that each can be described as a subset of the virtual one.

The act of communicating then involves an internal translation from local to virtual data representation. All information passing over the communication line is in this virtual format and can be interpreted successfully by any receiving machine. An obvious prerequisite of this type of solution is international agreement on standards for representing and for communicating word processing information. If these standards can be agreed upon and if manufacturers implement them, then true compatibility between CWPs becomes possible.

The characteristics of the virtual word processor approach are discussed more fully in Section 6.3.

6.2 Translation Centre

6.2.1 Introduction

It was shown in Chapter 5 that it is possible with the currently available communications packages to achieve successful transfer of print image text material via direct connections between two CWPs. It was also shown there that most CWP pairs had their own particular characteristics which made them in some sense unique. This was true especially for the BSC-based protocols where attempts to maximize information flow usually required changes or additions to code translation tables. This quickly becomes a major nuisance if a CWP is to communicate with many different types of word processing equipment; profiles and translation tables are required for each possible connection. A translation centre which would relieve each CWP of this task by assuming responsibility for maintaining all required information about various machines is certainly a desirable asset to a communications network.

Also in favour of this approach is that it allows two machines with incompatible protocols to communicate. This may not appear at first glance to be very significant as it was made clear in Chapter 5 that all CWPs in this study can already communicate directly with each other using existing packages. However, with a translation centre, there would no longer be any restrictions on the choice of protocol so that each machine's most effective protocol could be used to communicate with the translation centre with total disregard for the nature of the chosen destination. Therein lies the true power of this approach, for we have seen in Chapter 5 once again that four of the five machines have private protocols which guarantee successful file transfers between identical terminals. The fifth machine, the IBM OS/6, suffers only from a minor limitation in its ability to communicate.

Thus, a translation centre capable of handling each CWP's most effective protocol and of performing all necessary conversions between each pair has for all intents and purposes solved the word processor interface problem. In such an environment only minor sources of information loss would remain (e.g., IBM midline keyboard changes). Figure 6.1 shows how such a centre could be configured for a network which includes the five CWPs involved in this study. The AES, MICOM, WANG and XEROX machines all would use their private BSC-based protocols while IBM would use the 2770 protocol. It appears then that a translation centre would be an effective solution to the communicating word processor problem. However, there are serious implementation difficulties which complicate matters somewhat. These are discussed in the next four subsections.

6.2.2 Code Translation

Every word processor handles a different range of text characters and the same character may have different internal representations on different machines. In order to handle this range, a translation centre would maintain its own code set consisting of all possible text characters that the centre would ever process. Each incoming character would be converted to its appropriate internal code via a table lookup process. The activity would be a function of the input device. Similarly, on output, another table conversion appropriate to the output machine is used to produce the correct outgoing character. Characters which do not exist on both word processors are treated in some predefined manner (e.g., substitute a space character). Because there certainly are more than 256 possible text characters, the internal representation would require either a code containing more than 8 bits or a multiple character sequence.

The principal source of difficulty in the code translation area is the requirement for translation functions other than one-to-one (on a character basis). Although none of the five machines under consideration here generate composite character sequences in their "best" protocols, it is conceivable that other CWPs would. In that case, provision would have to be made in the conversion to and from the internal code set to decode and generate multiple character sequences, respectively. Another difficulty which is of immediate concern to the network of Figure 6.1 is the fact that the IBM system uses a keyboard identifier to determine the true identity of certain code values in the EBCDIC set. This requires that the code translator on input recognize keyboard identification command sequences and utilize the appropriate conversion table to generate the correct internal code value. Similarly, on output to an IBM machine, the proper command sequences must be generated automatically to identify the characters being transmitted. It is clear then that the handling of text characters in the translation centre is not a straightforward one-to-one process.

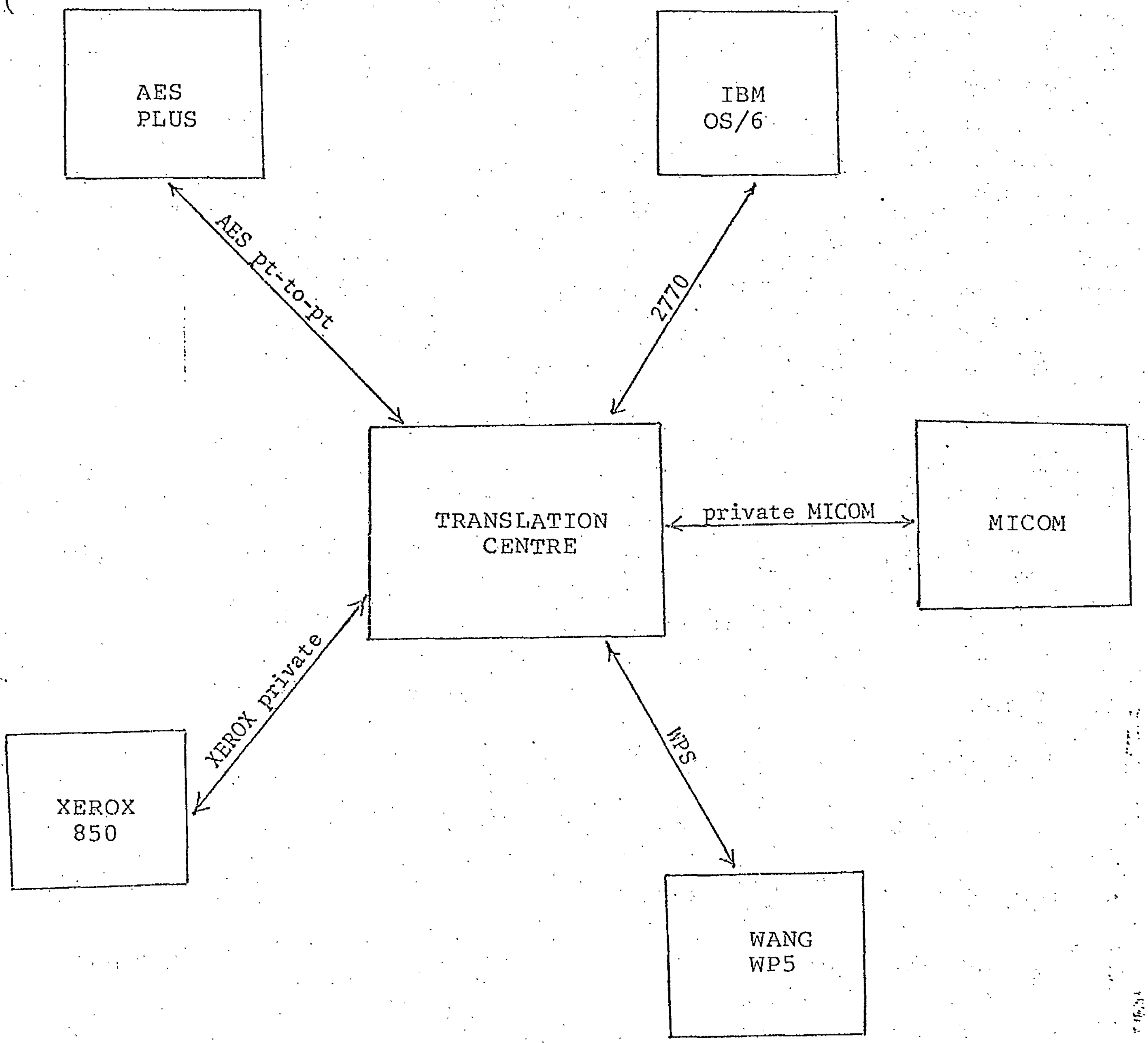


Figure 6.1: Network of CwPs using Translation CENTRE FACILITY

6.2.3 Function Representation

As is the case with text characters, each word processor supports a unique set of control functions with a corresponding unique set of internal codes to represent them. The first step in the development of a control code translation facility is to define a complete list of all possible control functions and to represent each of these with a unique internal code in the translator. Table lookup operations on input and output perform the required conversion to and from the internal code. Once again, control functions from one machine which do not have counterparts on another are treated in some consistent manner. This scheme works well for one-to-one translations. However, there are two other types of translations which do present some difficulty. One is the one-to-n translation involving only control characters; the other is the handling of control functions represented by a mixture of control characters and text characters (e.g., some print commands on the AES Plus).

The fact that some control functions require a sequence of control characters to represent them (e.g., merge on the AES Plus) signifies that a more general conversion process is needed for the translation centre. This process should handle the n-to-1 case on input and 1-to-n on output. In all such cases, the internal representation would still be a single code.

The second type of translation is trickier. It means that some control functions cannot be represented by a single internal code in the translator. This suggests that the translation process must be generalized even more to handle n-to-m translations to and from the internal representation.

All of the translation requirements discussed so far have involved simply the processing of the current input or output character(s) on a character-by-character basis. A more elaborate processing scheme is yet required to deal with the situation where the same control function is represented in radically different ways on different systems. An example is the representation of underlines. On the AES and XEROX machines, underlines are represented by a "start-of-underline, characters-to-be-underlined, end-of-underline" sequence. On the IBM terminal, a word underscore control character is used to determine that the preceding word should be underlined. The handling of this difference in function representation requires that the input and output character streams be processed on a word basis as opposed to the character basis that is possible otherwise. Other differences in control function representation might necessitate further changes in the translation process. This means that no clear picture of the final translation process is possible until all possible control function representations have been identified.

6.2.4 Format Information Representation

Format information is probably the most difficult item for a central translation facility to deal with. There is significant variation in the organization and content of this information such that an internal representation for the translation centre becomes difficult, although it is probably feasible. In this case, a general n-to-m translation facility would likely be required.

It seems that it might be possible to process the format information which accompanies every page of text, but blocks of format information embedded within a page pose a more serious problem. Not all machines possess the capability of placing these blocks of information randomly in the text (the AES Plus is one case in point). Does this mean that information is lost? Perhaps, but not necessarily. Embedded blocks of format information exist to effect changes in the format of the succeeding text. They have the effect of embedded commands; thus they could conceivably be represented as well by sequences of control characters. This is exactly what AES does; it represents justification information and changes in printer characteristics by control character sequences. This same result is achieved on other systems through embedded format blocks.

As a consequence of this, it is no longer possible to draw a definite boundary between enbedded format information and control functions. Perhaps all format information should be considered as sequences of control functions, which in some systems happen to be conveniently bundled together. However, some elaborate mapping functions would still be required to deal with situations such as justification where on one system (AES) this information is represented by a sequence of characters on every line while on others (XEROX, IBM) it is announced in embedded format blocks at the desired intervals. The entire area of format information representation is for further study.

6.2.5 Communication Protocols

The choice of communication protocols for use in a network having a translation centre is straightforward. Every machine talks to the centre using its most effective protocol. In this way, each CWP supplies the maximum amount of information to the translation centre. The onus is then on the latter to perform the necessary protocol translations between the input and output devices. Because many machines (e.g., AES, MICOM, WANG, XEROX) have implemented private protocols which do not cause any information loss, it should be possible to do the required conversions without affecting data contents. That these protocols are all BSC-based should simplify matters. Where some terminals have only protocols which are prone to information loss, there necessarily will be a possibility of imperfect communication when such terminals are involved. The IBM OS/6 is not seriously affected in this way so that most communication activity involving this machine should be successful

6.3 Virtual Word Processor

6.3.1 Introduction

The virtual word processor represents a much bolder approach to improving word processor interconnections. It requires a thorough rethinking of word processor communication requirements. This rethinking would have to move away from the restriction of IBM mainframe compatibility and specialized, limited-use protocols and towards a universal environment where information flows smoothly, unhindered by the communication process itself. Each machine in such a network of different CWPs would present the same image to the other members of the network, and this image would be that of a virtual word processor which can receive and understand all information directed its way. Each word processor is then responsible for making the necessary conversions between this general data representation and the appropriate internal form. Figure 6.2 illustrates a network of five different CWPs.

A CWP successfully incorporating the virtual word processor approach would have the following characteristics:

1) It would be able to translate all relevant data stored in a local disk file to a virtual representation understood by all connected terminals.

2) It would be able to receive and interpret all incoming data from the network. That information which has a local counterpart is translated accordingly; the remainder is discarded.

3) It would have a communications capability providing error-free transmission and reception with no restrictions on the data itself.

4) It would incorporate a file transfer protocol for coordinating the transfer of information between CWPs.

Every virtual word processor machine must be capable of receiving all possible types of information and interpreting it correctly; however, it need be capable of generating only that subset of total virtual data sufficient to represent its own internal information.

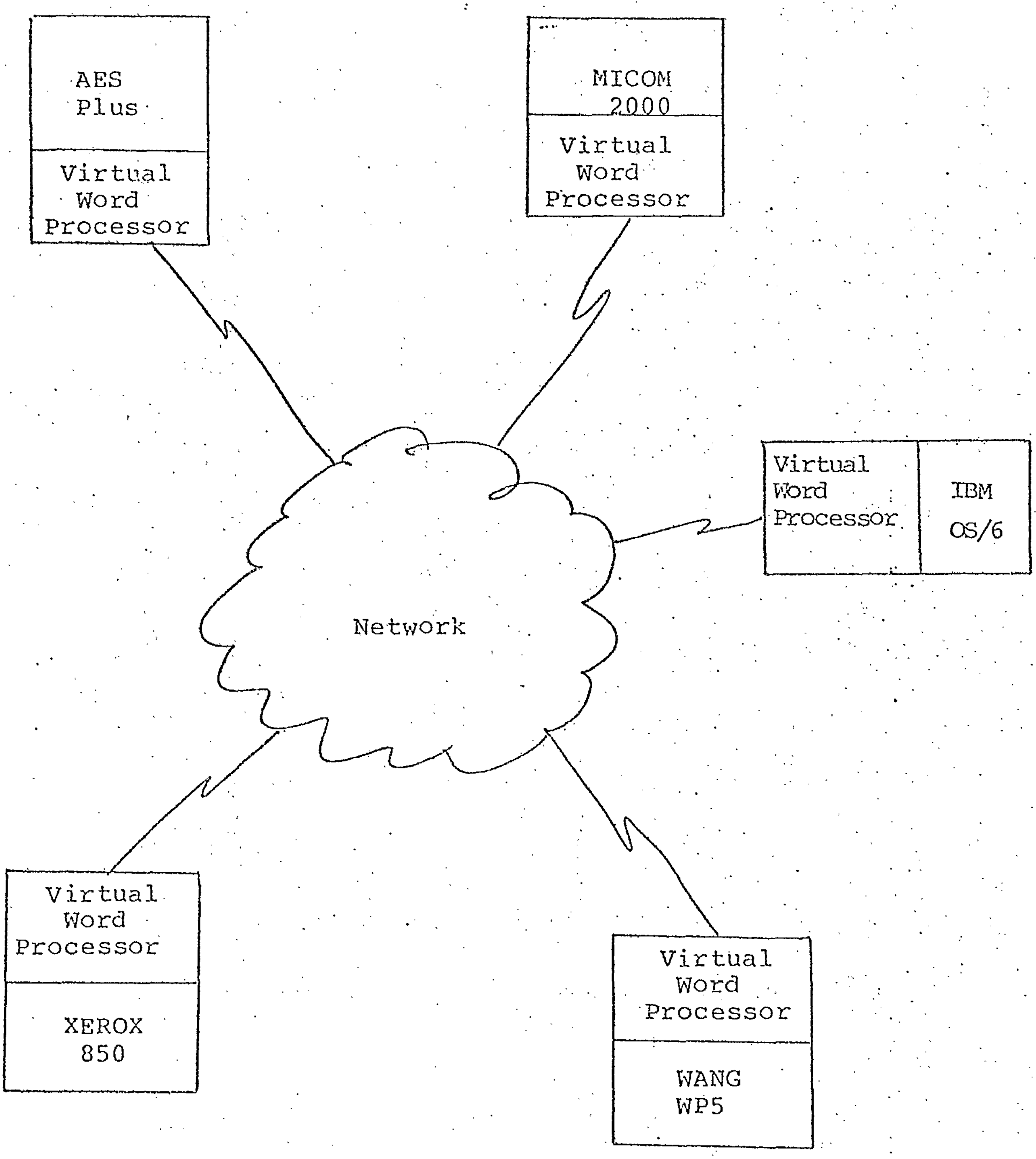


Figure 6.2: Virtual Word Processor Network

If this approach is to be totally successful, every manufacturer in the industry must develop a new communications package that satisfies the above requirements. It would then be possible for the maximum amount of information to be exchanged between CWPs, with the only remaining limitation being the inherent functional differences among machines, where functions on one machine have no counterpart on another.

Although this solution is in theory the best possible, short of having all word processors adopt the same functionality and the same internal data representation, it is not easily attainable. The previous chapters of this report have made it abundantly clear that there currently exist major discrepancies among the various word processing products available at this time.

A very powerful virtual machine will be needed to satisfy the requirements of today's CWPs. In addition, it will have to be flexible and expandable to adapt to new offerings in the CWP area. A large international effort will be required to come to agreement on standards that are satisfactory to everyone. A number of standards will be needed to cover all the aspects involved in the communication of word processing information. Fortunately, work is already underway in some of these areas, as discussed in the next section.

Once the status of current standardization efforts has been reviewed, it will then be possible to examine the implications of the virtual word processor approach for the exchange of text, control and format information and also to investigate the requirements for the communications protocol.

6.3.2 Guidance from International Standards

There is considerable standardization activity taking place at this time in the international arena in areas relevant to text communication. Attention is currently being focused on the development of standards for open systems interconnection, covering the control of a physical communication link, the flow of data in a network, virtual terminal and file transfer protocols, code sets and specific applications such as Teletex and Videotex. Appendix E summarizes some of the standards relevant to word processing. Some of these standards are well developed and in some cases have been accepted and put into use in actual systems. One example of this is the X.25 standard which has been implemented in the Canadian Datapac, the American Telenet and the French Transpac packet-switching networks. Other standards such as virtual terminal and file transfer protocols are still very much at the preliminary draft stage. The two standardization activities which have the greatest impact on the communicating word processor environment are the development of a model for open systems interconnection [9, 10] and the development of standard coded character sets for text communication [11]. These activities are discussed in turn below.

(a) Open Systems Interconnection

Work on a model for open systems interconnection is ongoing within both ISO and CCITT. This model defines seven different layers relevant to communications activity, as shown in Figure 6.3. Each layer performs a specified set of functions and provides certain services to the layer above it. The exact nature of the functions and services appropriate to each layer is still under study at this time, although the general features of each layer have been defined. The bottom four levels of this model, viz. the physical, link, network and transport levels, are of concern to this report only in that they must provide error-free and context-free transport of file data between CWPs. Appropriate standards covering the bottom three layers exist already and are in current use. These are the EIA RS-232C standard which deals with the physical level functions, the HDLC protocol for the link level (this protocol is the counterpart to the BSC protocol available in current CWP packages), and the X.25 protocol which controls interactions with a packet-switching network. The functions performed by the Transport layer in a communicating word processor environment are not yet clear; they might include providing a network-independent interface to the higher levels. At any rate, there appears to be little standardization activity in this area at this time apart from a draft proposal for the format of heading information added to the data stream at this level [12].

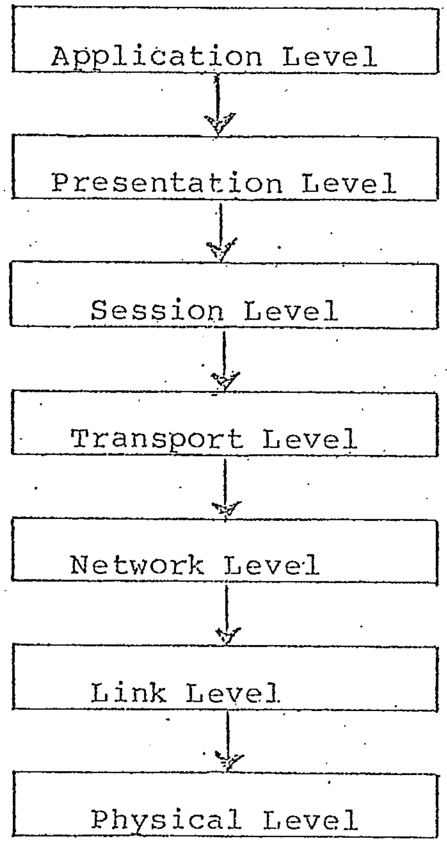


Figure 6.3: Seven Levels of Open System Interconnection Model

The arrows indicate that higher levels use services of the next lower level.

Of greater interest are the three highest levels of the model, namely the session, presentation and application layers. The companion report to this one [13] discusses in some detail the nature of the functions and services for each of these layers with respect to various applications such as communicating word processors, electronic mail, Teletex and Videotex. The general conclusion is that the proposed model for open systems interconnection (OSI) is indeed valid. However, the requirements of each of these applications are diverse enough to require different functions and services at the various layers. Thus, the implementations of each of these applications may be very different while still remaining within the framework of the model. Further, there is some question as to which layer certain functions should be allocated. One example of this is the negotiation on code sets to be used during file transfer; this activity could possibly take place at the session level as part of the session startup procedure or at the application level as part of the file transfer protocol.

The general validity of the OSI model strongly suggests its use for defining the virtual word processor terminal that is to provide the basis for interconnection among differing word processors. In fact, there is within ISO the beginnings of an effort to specify the properties of an even more general type of virtual terminal, one that supports a variety of terminal classes, one of which is word processing [9]. The Virtual Terminal Service (VTS) envisaged by ISO would be a service of the presentation layer and would exist to provide device independence for a particular application. In the words of reference 9 the VTS "is based on the concept of a <u>logical model</u> which is viewed by the correspondent [application] entity as a <u>unique presentation image</u>". In terms of the communicating word processor application, this means that the file transfer software on each terminal has the same logical view of the remote machine regardless of the actual nature of that device. This is illustrated in Figure 6.4. The VTS performs all necessary translations between local and global data representations. The inclusion of the VTS as part of the Presentation layer reflects the current thinking of ISO. This corresponds to the "Virtual Application" approach discussed in the companion report [13].

Another approach discussed in [13], termed the "Functional Separation" approach, suggests that any formatting or translation function which may require understanding of Application-level data semantics should be performed at the Application level while any formatting or translation functions performed solely for communications purposes should be performed within the Presentation level. If this view is adopted, then the Virtual Terminal Service, at least in the communicating word processor case, would be allocated to the Application level because the required translations between internal and virtual terminal data representations are non-trivial and do require understanding of Application-level data semantics.

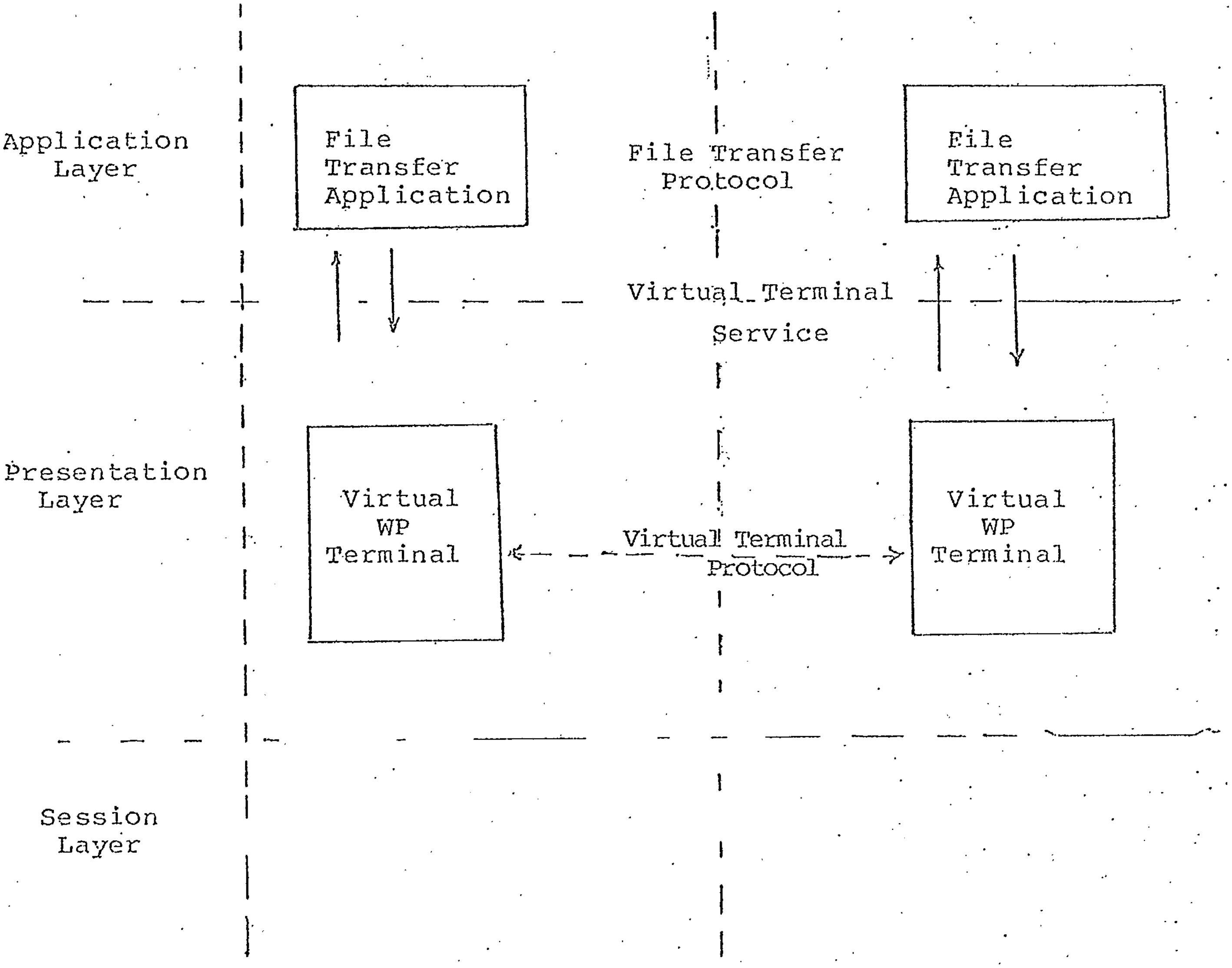


Figure 6.4: Virtual Word Processor Service as part of Presentation Level

It is beyond the scope of this report to determine which of these approaches is the more suitable. Both have their advantages and disadvantages. The important issue at this time is whether a workable virtual word processor is realizable, not where such a virtual terminal service should be located in terms of the OSI model. Subsections 6.3.3, 6.3.4, 6.3.5 and 6.3.6 will deal with the possibilities and problems of the virtual word processor.

We have seen so far that international standardization activity has focused at least to some extent on two of the required functions of a CWP incorporating the virtual terminal concept, namely a solid communications transport facility and a virtual terminal representation for word processing data. The third element, the file transfer protocol, has also been identified as an area requiring standardization [9]. However, at this stage, little has been done beyond stating that the requirement exists. Only when such a protocol has been defined and accepted can a complete view of the functions and services of all the layers of the OSI model be obtained. This is because the high-level application tends to determine the contents of the Transport, Session and Presentation levels as well as the Application layer itself [13].

(b) Coded Character Sets:

A great deal of effort has gone into the development of international standards for character codes for text communication. The major part of this effort has been directed toward achieving standards for textual material in general and the latin alphabet in particular. Attempts to define code standards for items outside the latin alphabet, such as greek and arabic characters or special application-specific control functions, are more recent phenomena and therefore are at a less mature stage of development.

Both CCITT and ISO are active in this area but as yet there is little firm agreement on character set contents. One standard that appears to be generally accepted is ISO 2022 [14] which partitions the layout of 7-bit and 8-bit code sets into fixed areas of control and data characters. In a 7-bit code, columns 0 and 1 (32 characters) are reserved for control characters while columns 2 through 7 (94 characters) are used for graphic characters. The exceptions are the first position of column 2 and the last position of column 7 which are reserved for the "space" and "DEL" characters, respectively. In an 8-bit code, the first eight columns are identical to the 7-bit structure and the last eight mimic the first eight, with different characters of course. Many sets of control characters are possible and each set is identified by the letter "C" followed by a number, so that the primary set of control characters is termed the CØ set. Similarly, each set of graphic characters is denoted by the letter "G" followed by a number. Thus the primary set of graphic characters is known as the GØ set. A variety of mechanisms using escape character sequences have been defined for switching from one code set to another, both on a per character or entire set basis. These are described in reference 14 which is currently being revised [15].

The main thrust of standardization efforts so far has been towards defining character sets appropriate to the representation of all characters in the latin languages. Two character sets have been proposed for text communication, consisting of a 7-bit primary set (the GØ set) containing alphabetic and numeric characters plus a few special graphics plus a supplementary set (the G2 set) containing more special graphics plus accents and diacritical marks. However, these sets have not yet been finalized. The most recent ISO draft proposal for the GØ set (Figure 6.6) [11] is only a subset of the GØ set of the International Reference Version (IRV) of ISO 646 (Figure 6.5) [16]. This latter set is the one proposed for use with Teletex. Similarly, the proposed G2 set for Teletex use (Figure 6.7) [17] is not the same as the latest ISO draft proposal for text communication (Figure 6.6) [11]. However, either version is sufficient for latin-based characters. Until these basic sets are finalized, there is little hope for the development of further code sets for special graphic characters for use in particular applications such as word processing.

The development of standards for control characters for text communication is less advanced than that for graphic characters. The latest ISO draft proposal for control functions for document interchange [11] defines twenty-one different functions in four groups:

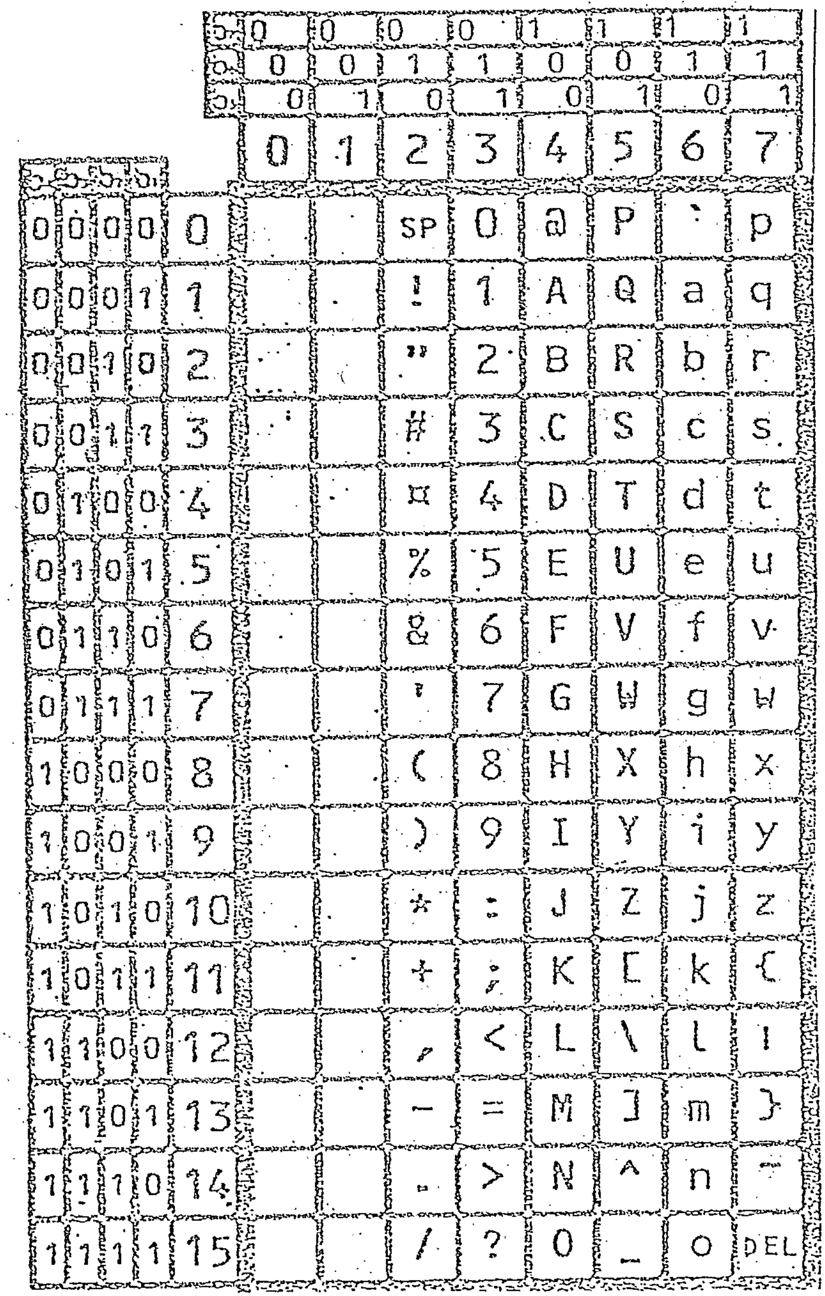
a) Format effectors: control functions which influence the layout and positioning of text on a presentation device (e.g., screen or printer).

b) Presentation control functions: These influence the appearance of the text on a presentation device in a uniform way, e.g., colour or page format. They may apply to all or part of a page or screen.

c) Code extension control functions: These are used for defining the repertoires of graphic characters and control functions in both 7-bit and 8-bit codes.

d) Miscellaneous control functions: These functions do not fit into any of the preceding categories.

Table 6.1 lists the proposed functions along with a brief description of each. Figure 6.8 shows the allocation of functions to character codes in the primary 7-bit CØ and supplementary C1 code sets. Note that some functions are not represented by individual character codes. These, such as "page format selection", are defined by a character sequence headed by the "control sequence introducer" character (CSI). This particular character is very useful for defining control functions which include parameters.



				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7
0	0	0	0	0			SP	0	@	P	`	p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2			"	2	B	R	b	r
0	0	1	1	3			#	3	C	S	c	s
0	1	0	0	4			¤	4	D	T	d	t
0	1	0	1	5			%	5	E	U	e	u
0	1	1	0	6			&	6	F	V	f	v
0	1	1	1	7			'	7	G	W	g	w
1	0	0	0	8			(	8	H	X	h	x
1	0	0	1	9			)	9	I	Y	i	y
1	0	1	0	10			*	:	J	Z	j	z
1	0	1	1	11			+	;	K	[	k	{
1	1	0	0	12			,	<	L	\	l	|
1	1	0	1	13			-	=	M	]	m	}
1	1	1	0	14			.	>	N	^	n	‾
1	1	1	1	15			/	?	O	_	o	DEL

Figure 6.5: International Reference Version (IRV) Graphical (GØ) part

				b.	0	0	0	0	1	1	1	1
				b.	0	0	1	1	0	0	1	1
				b.	0	1	0	1	0	1	0	1
b.	b.	b.	b.		0	1	2	3	4	5	6	7
0	0	0	0	0	▨	▨	▨	0	@	P		p
0	0	0	1	1	▨	▨	!	1	A	Q	a	q
0	0	1	0	2	▨	▨	"	2	B	R	b	r
0	0	1	1	3	▨	▨		3	C	S	c	s
0	1	0	0	4	▨	▨		4	D	T	d	t
0	1	0	1	5	▨	▨	%	5	E	U	e	u
0	1	1	0	6	▨	▨	&	6	F	V	f	v
0	1	1	1	7	▨	▨	'	7	G	W	g	w
1	0	0	0	8	▨	▨	(	8	H	X	h	x
1	0	0	1	9	▨	▨	)	9	I	Y	i	y
1	0	1	0	10	▨	▨	*	:	J	Z	j	z
1	0	1	1	11	▨	▨	+	;	K	[	k	
1	1	0	0	12	▨	▨	,	<	L		l	
1	1	0	1	13	▨	▨	-	=	M	]	m	
1	1	1	0	14	▨	▨	.	>	N		n	
1	1	1	1	15	▨	▨	/	?	O		o	▨

1 - Primary set of graphic characters for text communication

Coding when used as the G0 set of a 7-bit or 8-bit code (in an 8-bit code, the additional bit b_8 = 0).

				b.	0	0	0	0	1	1	1	1
				b.	0	0	1	1	0	0	1	1
				b.	0	1	0	1	0	1	0	1
b.	b.	b.	b.		0	1	2	3	4	5	6	7
0	0	0	0	0	▨	▨	▨	°		―		ĸ
0	0	0	1	1	▨	▨	¡	±	`	|	Æ	æ
0	0	1	0	2	▨	▨	¢	²	´	Ω	Đ	đ
0	0	1	1	3	▨	▨	£	³	^	ª		ð
0	1	0	0	4	▨	▨	$	×	~	º	Ħ	ħ
0	1	0	1	5	▨	▨	¥	µ	¯			ı
0	1	1	0	6	▨	▨	#	¶	˘		Ĳ	ĳ
0	1	1	1	7	▨	▨	§	·	˙		Ŀ	ŀ
1	0	0	0	8	▨	▨	¤	÷	¨		Ł	ł
1	0	0	1	9	▨	▨	SQL	SQR			Ø	ø
1	0	1	0	10	▨	▨	DQL	DQR	˚		Œ	œ
1	0	1	1	11	▨	▨			¸			ß
1	1	0	0	12	▨	▨	←	¼			Þ	þ
1	1	0	1	13	▨	▨	↑	½	˝		Ŧ	ŧ
1	1	1	0	14	▨	▨	→	¾	˛		Ŋ	ŋ
1	1	1	1	15	▨	▨	↓	¿	ˇ		ŉ	▨

2 - Supplementary set of graphic characters for text communication

Coding when used as the G2 set of a 7-bit or 8-bit code (in an 8-bit code, the additional bit b_8 = 0).

Figure 6.6: GØ and G2 Sets from ISO Draft Proposal 6937

				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7
b_4	b_3	b_2	b_1									
0	0	0	0	0						ʿ		
0	0	0	1	1			¡		`		Æ	æ
0	0	1	0	2			ƒ	°	´	¸	Đ	đ
0	0	1	1	3			£	²	^	˛		ð
0	1	0	0	4			$	³	~		Ħ	ħ
0	1	0	1	5			¥	µ	¯			ı
0	1	1	0	6			¢	π	˘		Ĳ	ĳ
0	1	1	1	7			§	·	˙			
1	0	0	0	8			½		¨	_	Ł	ł
1	0	0	1	9			¼		¨		Ø	ø
1	0	1	0	10			¾		˚		Œ	œ
1	0	1	1	11			«	»	ʼ			
1	1	0	0	12			±				Þ	þ
1	1	0	1	13			ª		˝			
1	1	1	0	14			º					ß
1	1	1	1	15			ℓ	¿	ˇ			

Figure 6.7: The G2 Set of the Teletex Proposal

				b.	0	0	0	0	1	1	1	1
				b.	0	0	1	1	0	0	1	1
				b.	0	1	0	1	0	1	0	1
b.	b.	b.	b.		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL		▨	▨	▨	▨	▨	▨
0	0	0	1	1			▨	▨	▨	▨	▨	▨
0	0	1	0	2			▨	▨	▨	▨	▨	▨
0	0	1	1	3			▨	▨	▨	▨	▨	▨
0	1	0	0	4			▨	▨	▨	▨	▨	▨
0	1	0	1	5			▨	▨	▨	▨	▨	▨
0	1	1	0	6		SS2	▨	▨	▨	▨	▨	▨
0	1	1	1	7			▨	▨	▨	▨	▨	▨
1	0	0	0	8	BS		▨	▨	▨	▨	▨	▨
1	0	0	1	9		SS3	▨	▨	▨	▨	▨	▨
1	0	1	0	10	LF	SUB	▨	▨	▨	▨	▨	▨
1	0	1	1	11		ESC	▨	▨	▨	▨	▨	▨
1	1	0	0	12	FF		▨	▨	▨	▨	▨	▨
1	1	0	1	13	CR		▨	▨	▨	▨	▨	▨
1	1	1	0	14	SO		▨	▨	▨	▨	▨	▨
1	1	1	1	15	SI		▨	▨	▨	▨	▨	▨

1 - Primary set of control functions for document interchange

Coding when used as the C0 set of a 7-bit or 8-bit code (in an 8-bit code, the additional bit $b_8 = 0$).

				b.	0	0	0	0	1	1	1	1
				b.	0	0	1	1	0	0	1	1
				b.	0	1	0	1	0	1	0	1
b.	b.	b.	b.		0	1	2	3	4	5	6	7
0	0	0	0	0	▨	▨	▨	▨			▨	▨
0	0	0	1	1	▨	▨	▨	▨			▨	▨
0	0	1	0	2	▨	▨	▨	▨			▨	▨
0	0	1	1	3	▨	▨	▨	▨			▨	▨
0	1	0	0	4	▨	▨	▨	▨			▨	▨
0	1	0	1	5	▨	▨	▨	▨			▨	▨
0	1	1	0	6	▨	▨	▨	▨			▨	▨
0	1	1	1	7	▨	▨	▨	▨			▨	▨
1	0	0	0	8	▨	▨	▨	▨			▨	▨
1	0	0	1	9	▨	▨	▨	▨			▨	▨
1	0	1	0	10	▨	▨	▨	▨			▨	▨
1	0	1	1	11	▨	▨	▨	▨	PLD	CSI	▨	▨
1	1	0	0	12	▨	▨	▨	▨	PLU		▨	▨
1	1	0	1	13	▨	▨	▨	▨			▨	▨
1	1	1	0	14	▨	▨	▨	▨			▨	▨
1	1	1	1	15	▨	▨	▨	▨			▨	▨

2 - Supplementary set of control functions for document interchange

Coding when used as the C1 set of a 7-bit code.
The coded representations shown are those of the final bit combinations of the Escape sequences of the form ESC Fe.

Figure 6.8: CØ and Cl Sets of ISO Draft Proposal 6937

Table 6.1 Proposed ISO Repertoire of Control Functions

Format Effectors

Name	Abbrev.	Definition
backspace	BS	active position moved one character backwards on same line
line feed	LF	active position advanced to corresponding position on next line
form feed	FF	active position advanced to corresponding position of first line of next page
carriage return	CR	active position moved to first position of current line
partial line down	PLD	perform partial vertical shift downwards
partial line up	PLU	perform partial vertical shift upwards
horizontal position relative	HPR	has one numeric parameter indicating the number of character positions to be advanced horizontally
vertical position relative	VPR	has one numeric parameter specifying the number of lines to be advanced vertically

Presentation Control Functions

Name	Abbrev.	Definition
page format selection	PFS	specifies page format; numeric parameter specifies whether size A4 or A4L is selected
select graphic rendition	SGR	specifies presentation attribute for subsequent text; numeric parameter selects one of - current default rendition - bold or increased intensity - italics - underlined
select horizontal spacing	SHS	specifies character spacing according to numeric parameter
select vertical spacing	SVS	specifies vertical spacing according to numeric parameter

Table 6.1 Proposed ISO Repertoire of Control Functions (cont'd)

Code Extension Control Functions

Name	Abbrev.	Definition
shift out	SO	used to extend graphic character set of the code according to ISO 2022
shift in	SI	used to extend graphic character set of the code according to ISO 2022
escape	ESC	used to extend the set of control functions of the code according to ISO 2022
single shift 2	SS2	used to extend the graphic character set of the code according to ISO 2022
single shift 3	SS3	used to extend the graphic character set of the code according to ISO 2022
control sequence introducer	CSI	used to extend the set of control functions of the code, in particular for control functions with parameters according to ISO 6429

Miscellaneous Control Functions

Name	Abbrev.	Definition
null	NUL	used to accomplish media-fill or time-fill
substitute character	SUB	used in place of a character that has been found invalid or in error
identify graphic subrepertoire	IGS	identifies subrepertoire of the total set of graphic characters, numeric parameter identifies the particular subrepertoire selected

6.3.3 Code Translation

While it seems definite from the discussion of the previous section that it may be some time before international agreement is reached on code representations for all text characters that are possible in the word processing environment, it is equally certain that time is the only obstacle that remains in the path of such agreement. The code extension techniques of ISO 2022 [14] provide a powerful mechanism for representing as many different characters as are required. The only difficulty is deciding on the range of characters appropriate to word processing and on the allocation of characters to code sets. The current proposed GØ and G2 code sets are not sufficient for word processing as they are missing a number of special graphics. At least one additional set would be required.

6.3.4 Control Function Representation

Control functions may be classified into four groups on the basis of the way they are represented on different machines:

1) Functions which are always represented by a single special character (e.g., end-of-line). This group contains the majority of control functions.

2) Functions which may be represented by more than one special character (e.g., merge).

3) Functions which may be represented by one or more special characters plus a parameter (e.g., some AES Plus print commands).

4) Functions which may be represented in totally different ways in different terminals. One example of this is underline which is represented by AES and XEROX as "start-of-underline, character sequence, end-of-underline", while IBM uses a word underscore character to identify underlines. Another example is justification which is represented on a line-by-line basis as a special sequence of characters in the AES machine while in the XEROX and IBM case, this information is represented by a single command which affects an arbitrary range of text.

The question then is: Can these four groups of functions be mapped successfully into a virtual terminal representation? To answer this question, two possible approaches to representing control function information will be investigated. They are the control code mapping and general text language approaches. Each is discussed below.

Control Code Mapping: This approach follows the current ISO proposal for the representation of control functions [11] as presented in the previous section. Currently only 21 control functions have been identified (Table 6.1); this clearly is not sufficient for the word processing application (refer to Table 3.6). Representations for all possible control functions would have to be defined by creating additional control character sets beyond the CØ and C1 sets proposed in reference 11 and/or by defining control character sequences starting with the Control Sequence Introducer (CSI) character.

In this way, the first three groups of control functions can be handled by a direct mapping function. Note that Group 3 functions are handled easily through the facility for parameter inclusion in control sequences preceded by the CSI character. The fourth group can also be handled, although in this case semantic knowledge of internal data representation is needed. This may complicate the mapping function somewhat. This approach is not only workable but has the advantage of being consistent with the current ISO proposals.

General Text Language: Another possible approach to describing control functions is to use a language as opposed to special character codes. This language would be agreed upon internationally and would consist of language primitives using only simple text characters. Such a language approach has been adopted in the UNIX text processing environment [18] for describing mathematical equations. Another occurrence of the language approach is in the TEX word processing system [19].

This technique has the desirable feature that only one or two special control characters would be needed to delimit the language representation of control functions from true text. Also, the language approach provides considerable flexibility in function representation; this in turn enhances the chances of speedily reaching international agreement. However, it is not clear whether there is sufficient interdependency among control functions to allow clear compact function descriptions to be achieved with a small set of primitives as is possible in the applications discussed in [18] and [19]. If this is not the case, then the value of this approach diminishes as the number of required primitives increases. In addition, this approach has the further disadvantage of requiring a more elaborate translation facility than the control code mapping approach. It needs to have all of the intelligence of the latter as far as understanding the semantics of internal control function representation is concerned and it must include a language generator and interpreter. Finally, it is not the direction in which ISO is moving.

6.3.5 Format Information Representation

In the preceding chapters of this report, format information has been treated differently from control functions because of its different representation in existing systems. However, format information can be viewed as a set of control functions which happen to be grouped together and perhaps represented differently. When this information is translated into a virtual word processor format, the distinctions between format information and control functions vanish, so that only two distinct items of information remain: text characters and control functions. This is exactly what has taken place in the current ISO proposal for control function representation, where some of the proposed standard functions, e.g., Page Format Selection, Select Horizontal Spacing, correspond in current terminals to format information.

With this breakdown of the separation between format and control information, the comments in the preceding subsection on the representation of control functions in the virtual word processor apply as well to format information. The only distinction that remains is the potentially different mapping between virtual terminal representation and internal machine representation for different types of functions.

6.3.6 Communication Protocols

International standardization efforts have had the most success with the lower levels of the OSI model. Working versions of standards for the physical (RS-232), link (HDLC) and network (X.25) layers are already in existence, and together they provide the desired type of communication facility, one that provides error-free transport of data in a network environment with no constraints on the nature of the data transmitted. The use of these standard protocols has no potential for information loss due to communication protocol behaviour when CWPs exchange files.

6.3.7 Remarks

It is evident at this point that the virtual word processor is a viable mechanism for having differing CWPs communicate, and will become more so as international standards evolve and become more established. Already it is possible to get a glimpse of the possible contents of most of the layers of the OSI model. Figure 6.9 shows that workable standards exist for the physical, link and network levels while the transport layer remains a question mark.

Layer	Content
Application	- file transfer protocol
	- Virtual Word Processor
Presentation	- Transfer control - Encryption
Session	Teletex-like
Transport	?
Network	X.25
Link	HDLC
Physical	RS-232C

Figure 6.9: Possible Contents of OSI Layers for Communicating Word Processor Application

Similarly, the functions of the session layer are to some extent dependent on the nature of the expected interactions between CWPs, e.g., whether dialogue is one-way or two-way; also the question of making session functions applicable to a variety of uses, e.g., electronic mail, CWP, Teletex, will also affect the makeup of this layer. In the simplest case, a session layer similar to the one proposed for Teletex use [17] would probably be appropriate.

Candidates for inclusion at the presentation level are transfer control (checkpointing, recovery and flow control) of application data units (normally file records), encryption control and possibly the virtual word processor terminal translation service.

The application layer contains the file transfer protocol and possibly the virtual word processor terminal translation service. This latter function is shown in Figure 6.9 as spanning both the presentation and application layers to indicate the uncertainty surrounding its actual placement.

6.4 Comparison of the Two Approaches

The Central translation centre has the following advantages:

1) It requires no change to existing CWP equipment or software.

2) It can resolve to the greatest possible degree differences in protocol behaviour and in data representation.

3) Varying levels of performance are achievable depending on the amount of effort invested in the translator.

4) It can be used for communication between devices other than CWPs. Possibilities range from mainframes down to simple "dumb" terminals.

5) It is the approach currently in vogue.

Its disadvantages include the following:

1) It requires at least one additional processor in the network to perform the translation duties.

2) It is dependent on the communications protocols of the individual machines; if a particular machine has only an inferior protocol available, then the translation centre cannot compensate for any related deficiencies.

3) The complexity of the translator increases enormously when large numbers of different CWPs are to be interconnected.

4) The requirement to perform translations on all of text, control functions, format information and communication protocols makes the software development effort a complicated and arduous one.

5) Significant maintenance overhead would be associated with a central translator to keep up with new entries in the CWP field.

6) Intimate knowledge of vendors' private protocols and data formats is required.

7) Obsolescence is inevitable.

The advantages of the <u>virtual word processor</u> are as follows:

1) No additional intermediate processor is needed, differing CWPs may interconnect directly.

2) It resolves all discrepancies between machines apart from inherent functional differences.

3) There is a one-time cost associated with adapting to this approach; there is no maintenance cost in adapting to new terminals as these will incorporate the virtual word processor standard.

4) Each manufacturer may maintain the privacy of his internal data representations as he alone develops the translation software to adapt to the virtual WP service.

5) The virtual word processor may become but one class of a range of virtual terminals. The intelligent terminal of the future may take on many disguises; it may at any particular time take on the role of word processor, data processor, or Teletex machine, etc. The ability to communicate with other dedicated, less intelligent terminals will be a valuable asset.

Its disadvantages include the following:

1) It is not available now; much intensive standardization effort will be required before the virtual word processor becomes a reality.

2) It requires a new development effort on the part of each manufacturer to adapt to the virtual terminal.

3) The software to provide the required amount of intelligence may not fit into existing machines.

4) It would not be possible to communicate with less intelligent machines; only terminals which understand the virtual word processor can interconnect. (Note that this drawback can be overcome if an intelligent terminal implements more than one virtual terminal service).

Both approaches have their good points and their weaknesses. The main point in favour of the central translation facility is that it is possible now, albeit at the cost of considerable complexity in the translator and with the prospect of much more to come. However, it is inevitable that standardization will come to the area of communicating word processors and it is just as inevitable that the virtual word processor approach will become the accepted means for having CWPs interconnect because it provides a much cleaner path between two machines. Work is already underway in this area but it may take some years yet before it comes to fruition.

Chapter 7

Conclusions and Recommendations

7.1 Summary and Conclusions

This report has investigated the communicating word processor (WP) problem as it affects five stand-alone word processors: the AES Plus, the IBM OS/6, the MICOM 2000/2001, the WANG Word Processor 5 and the XEROX 850. The general categories of communications packages currently available for stand-alone word processors have been identified and a methodology for describing the logical components of a CWP presented. A functional and a communications profile of each of the five CWPs under consideration has been prepared and used as input to a discussion of current interworking capabilities for these machines. Finally, two possible solutions to the problems that were thus identified have been proposed and considered in some detail.

The communications packages that are currently available for stand-alone word processors fall into three categories:

i) dumb terminal emulation using asynchronous protocols such as TTY or IBM 2741; these are designed mainly for low-speed, unprotected communications with mainframes.

ii) remote batch terminal emulation using synchronous BSC-based protocols such as the IBM 2780, 3780 and 2770 protocols; these are intended principally to provide protected, synchronous, medium speed access to mainframes.

iii) word processor to identical word processor using private BSC-based protocols; these give terminals from the same manufacturer a full file transfer capability with no information loss.

Only the third category was designed specifically for direct interconnection among word processors and is the only one well suited to the application. Unfortunately, it is restricted to communications between identical machines.

There are three logical components of a word processing file which may be differently affected by the communication process. These are as follows:

i) text: this consists of all data characters entered by an operator that have no special meaning for the system.

ii) control codes: these are embedded characters in a file which are interpreted as special by the system.

iii) format information: this information defines the general structure of the text. It includes such items as page size, margin settings and tab stops.

The above components are those which are affected by the file transfer process. The logical components which effect the transfers are the code translation and communications protocol modules, as shown in Figure 2.1.

In chapter 3, a functional profile has been prepared for each of the five CWPs which includes the range of textual characters that can be generated, a list of control functions that are represented by embedded codes and a mention of the way in which format information is stored. For each of these components, there is significant variation from machine to machine. This is illustrated for control functions in Table 3.6.

In chapter 4, a communications profile for each CWP has been presented in terms of the three categories of available communications packages. These profiles indicate that all CWPs except the IBM OS/6 have an asynchronous TTY-level communications capability plus a private terminal-to-identical-terminal protocol for error-free file transfers. The IBM machine has only the BSC-based 2770 protocol available for communications. The other four CWPs in this study also have a BSC-based batch terminal emulation which does allow some interconnection. However, no two communications packages are identical.

Evaluation of the interworking capabilities of these terminals indicates that communication is possible using both asynchronous TTY-level and synchronous BSC-level protocols.

Asynchronous communications is possible among all but the IBM CWPs. In general, the range of text characters is limited to the ASCII set and few control characters and no format information can be exchanged.

Also, page width incompatibility among machines may lead to text layout changes. As a result, there are only limited uses for such a communication capability. The principal one is the ability to obtain a printed copy of the original file if it did not contain any non-communicable text or control characters. As a rule, the significant amount of information loss which occurs when files are transferred in this way severely inhibits further processing of the received data. File transfers between two AES Plus or two XEROX machines do not suffer from these limitations because special provisions exist in their communications packages for the special case.

Synchronous communications among all five CWPs is possible because they all support at least one batch terminal emulation (2780, 3780, or 2770) and these emulations are to a great extent compatible with each other. However, communication between dissimilar machines remains far from ideal because of protocol limitations. It is worth repeating that perfect communication between identical machines is possible for all terminals save the IBM OS/6 because each vendor has a private protocol designed specifically for this environment. Communication between two IBM OS/6 machines suffers from only minor deficiencies. Difficulties which arise when machines from two different manufacturers attempt to communicate with these emulation packages are due to the following factors:

i) code set translation: each word processor handles a different set of text characters and the communications packages limit the number of characters that can be represented on the communications line. Normally, this corresponds to the EBCDIC code set.

ii) control function representation: differences in functionality between machines result in control functions on one terminal which have no counterparts on others or which are represented differently. Also, the code sets used for communication limit the number of control codes which can be transmitted.

iii) format information representation: no two CWPs maintain the same set of items for describing the formatting of text nor do they represent this information in the same way. Further, the communications protocols in most cases have no provisions for transferring such information.

iv) protocol deficiencies: in addition to the code set limitations and the restrictions on the transfer of format information, the 2780, 3780 and 2770 protocols work best when dealing with 80-character records which correspond to individual lines of text. Otherwise, the layout and content of the text may be altered during the file transfer.

The net effect of these limitations is that communication between dissimilar CWPs is not very effective when files to be transferred contain other than the standard text and control characters found in the ASCII and EBCDIC code sets. More success is sometimes possible with 8-bit BSC-based protocols than with 7-bit ASCII ones because the vendor-supplied code translation tables in some instances include additional characters for representing control functions (the EBCDIC/WP set available with IBM and XEROX) and in other cases (AES, MICOM, WANG)may be extended to handle additional characters. Even then, the possibility of totally successful file transfers remains slim. A more realistic expectation is the transfer of print image text material between CWPs.

Two possible schemes to solving the above problems have been proposed in Chapter 6. They are the translation centre and the virtual word processor approaches.

The translation centre would provide a central switching service which allows different machines with different protocols to communicate. Each individual word processor would communicate with the translation centre which would in turn perform all translation functions required to communicate with a remote machine. This approach is currently popular because it requires no change to existing CWPs and it can be very powerful if each individual CWP uses its most effective protocol (typically its own private one). However, it would require an enormously complex translation facility if it is to support many CWPs.

The virtual terminal approach requires that each word processor emulate a standard terminal for purposes of communication. Thus, all CWPs would appear the same to each other, all internal information would have a valid virtual representation on the communication line and all received data could be received and interpreted correctly in terms of the local environment. Also, there would be a restriction-free and error-free communications capability along with a common file transfer protocol. An essential ingredient of this scheme is international agreement on all the various aspects involved in the communication of word processing information. Appendix E summarizes some of the international standards relevant to this activity. Another standard of importance is the proposed model for Open Systems Interconnection [9, 10]. The virtual terminal approach is consistent with this model, although it is unclear as yet at which level the virtual terminal protocol should sit, whether it should be at the Application or Presentation level (see section 6.3.2). The virtual terminal approach has the major advantage that it can resolve in a clean fashion all discrepancies between machines apart form inherent functional differences. Its principal disadvantage is that it is not available now and may not be for a while yet. Many difficult issues remain to be resolved at the international level.

The advantages and disadvantages of the translation centre and virtual terminal approaches have been discussed in greater detail in section 6.4.

7.2 Recommendations

Based on the results of this investigation, the following short and medium-term view of the status of word processor communication is presented.

1) It is recommended that existing communications packages be used in situations (such as electronic mail) where the objective of communication is the transfer of file copies for the sole purpose of printing or viewing at the remote station. In such cases, if the requirements that files contain only standard text and control characters and that they be organized in 80-column form can be adhered to, then, useful, uniform communication is possible.

2) It is recommended that if the above conditions are too stringent, a general translation facility be considered to improve the performance of the communications environment. The degree of performance improvement is dependent on the amount of effort invested in such a facility and on the amount of currently proprietary information describing private protocols and code sets that can be obtained from vendors.

3) Further study is recommended of the virtual terminal approach. This is the ultimate solution to the communicating word processor problem. However, efforts to standardize the various components of a virtual terminal system are not sufficiently advanced to permit the development at this time of a terminal incorporating this approach. While international agreement on the representation of the basic latin character set and on mechanisms for extending it is imminent, much work remains to be done. The following areas are for further study:

   - definition of additional code sets for describing word processing text characters not included in the current international sets.

   - definition of code sets for describing all possible control codes used in word processing; these would serve to represent both control and format information.

- further clarification of the proposed model for Open Systems Interconnection as it applies to the CWP situation. In particular, the functions and services of the Transport, Session, Presentation and Application layers remain to be precisely defined.

- specification of a virtual terminal protocol that can be used by all CWPs.

- specification of the file transfer protocol required to move files from one terminal to another.

Once these items have been dealt with, it will be possible for all vendors to incorporate the virtual terminal into their products, threby assuring a full interconnection capability among CWPs.

Bits

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1	COLUMN ROW	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Appendix A: ASCII Code Set

[redacted]

MSB LSB HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE*	DS		SP	&	HY						{**	}**	**	0
1	SOH*	DC1*	SOS				/		a	j	~**		A	J	NSP	1
2	STX*	DC2*	FS	SYN*					b	k	s		B	K	S	2
3	ETX*	DC3*							c	l	t		C	L	T	3
4	PF	RES	BYP	PN					d	m	u		D	M	U	4
5	HT	NL	LF	RS					e	n	v		E	N	V	5
6	LC	BS	EOB/ETB*	UC					f	o	w		F	O	W	6
7	DEL*	IL	PRE/ESC*	EOT*					g	p	x		G	P	X	7
8		CAN							h	q	y		H	Q	Y	8
9	RLF	EM						**	i	r	z		I	R	Z	9
A	SMM	CC	SM		¢**	!**	¦**	:						**	**	
B	VT				.**	$**	,**	#**								
C	FF	IFS		DC4	<**	***	%	@**								
D	CR	IGS*	ENQ*	NAK*	(	)	—	'								
E	SO	IRS	ACK		+	;	>**	=								
F	SI	ITB*	BEL	SUB	|**	¬**	?	"								

* BSC LINE CONTROL CHARACTERS
** LANGUAGE VARIABLE CODES - USO GRAPHICS ARE SHOWN

Appendix B: EBCDIC Code Set

MSB LSB HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE*			SP	&	RHY						2**	3**	¼**	0
1	SOH*	DC1*			RSP		/		a	j	o**		A	J	NSP	1
2	STX*	DC2*		SYN*					b	k	s		B	K	S	2
3	ETX*	DC3*	WUS	IRT					c	l	t		C	L	T	3
4									d	m	u		D	M	U	4
5	HT	NL	LF						e	n	v		E	N	V	5
6	RCR	BS	ETB*	NBS					f	o	w		F	O	W	6
7	DEL*		ESC*	EOT*					g	p	x		G	P	X	7
8				SBS					h	q	y		H	Q	Y	8
9	SPS			IT				±**	i	r	z		I	R	Z	9
A	RPT	UBS	SW	EOP	[**	]**	½**	:					SHY	**	**	
B			CU2*		.**	$**	,**	//**								
C	FF				§**	***	%	@**								
D		IGS*	ENQ*	NAK*	(	)	_	'								
E		IRS*			+	;	¶**	=								
F		ITB*	BEL		!**	¢**	?	"								

* BSC LINE CONTROL CHARACTERS
** LANGUAGE VARIABLE CODES - USO GRAPHICS ARE SHOWN

EBCDIC/WP CONTROL CHARACTER DEFINITION

BS	BACKSPACE
CRE	CARRIER RETURN
EOP, RPE	REQUIRED PAGE END
HT	HORIZONTAL TAB
INX	INDEX
IRT	INDEX RETURN (REQUIRED)
IT	INDEX TAB
NBS	NUMERIC BACKSPACE
NSP	NUMERIC SPACE
PE	PAGE END
RCR	REQUIRED CARRIER RETURN
RHY	REQUIRED HYPHEN
RPT	REPEAT
RSP	REQUIRED SPACE
SBS	SUBSCRIPT
SHY	SYLLABLE HYPHEN
SP	SPACE
SPS	SUPERSCRIPT
STP	STOP
SW	SWITCH
UBS	UNIT BACKSPACE
WUS	WORD UNDERSCORE

Appendix D

Detailed Discussion of CWP Interconnection

D.1 Introduction

In this Appendix, the level of interconnection for the five CWPs under consideration will be investigated for two categories of communications facility: asynchronous TTY-level and synchronous BSC-level. The procedure will be the same in each case. All possible combinations of interconnection will be looked at; for each combination, a set of protocol parameters for optimum performance will be suggested and the specific areas and causes of possible information loss studied.

D.2 TTY Level

All machines except the IBM OS/6 provide an asynchronous communications package. The four machines which do are capable of transmitting the ASCII code set. In the following sub-sections, the extent to which file information can be transferred from machine to machine using asynchronous ASCII communications will be reviewed.

D.2.1 AES Plus-AES Plus

It is possible for two AES Plus machines to communicate in transparent mode using 8-bit data. In this case, the code translation step is bypassed and the internal AES code set is transmitted. As a result, all text and control codes are preserved. However, the format information, because it is not stored as a text string, cannot be transmitted and is therefore lost.

Another source of information loss is the fact that during transmission and reception, all data moving to/from disk and communication line passes via video memory. This memory can store only 7-bit characters; during transmission a conversion takes place from the internal 8-bit representation of the data on disk to the 7-bit representation of video memory and then back again to the 8-bit representation used on the communication line. Those codes which cannot be represented directly in video memory are converted to their closest counterpart. The consequence is a potential data loss. In actual fact this data loss does not occur frequently. This is because each machine is capable of generating a maximum of 128 characters from the keyboard. Since the video memory character generator in a particular machine can handle all characters generated from the local keyboard, there is no data loss during local processing. A problem only occurs when two machines which have different character generators attempt to communicate. In that case, those characters which are not available on both machines get converted to a compatible character. Character accents are the items of information most frequently lost as a consequence of this process. Table 5.1 summarizes the behaviour of an asynchronous AES Plus - AES Plus link.

Parameter settings for optimum information transfer are listed below:

Parameter	AES Setting
code set	AES internal
number of data bits	8
mode of operation	transparent, AES Plus to AES Plus
modem type	full duplex
XON/XOFF	no

D.2.2 AES Plus - MICOM 2000/2001

The AES asynchronous communications package is very flexible in that it can transmit up to 8-bit character codes and its translation table is user modifiable. This means that all internal codes can be transmitted and represented on the communications line by whatever bit combinations the user chooses. On the other hand, the MICOM asynchronous communications package does not give the user access to the translation function, although 8-bit codes can be transmitted. In fact, only characters which are part of the standard 7-bit ASCII set are transmitted untouched; all others, including special text and control characters, are transmitted as the question mark (?) character.

Thus the MICOM machine is the more restrictive one with the result that asynchronous communication between an AES Plus and a MICOM 2000/2001 may be characterized in the following way:

Text characters are limited to the standard 7-bit ASCII set; underlines are represented by character-backspace- underline sequences; tabs are sent as a series of spaces; all text characters and control codes outside the ASCII set are transmitted as the question mark character by the MICOM machine; the AES machine normally would transmit these as the null character. Neither machine is capable of transmitting format information. However, there are differences in how each machine formats received text. With the MICOM machine, the received text is stored and displayed according to the margin settings set at the receiver. If an incoming string of characters reaches the right margin, two possibilities can occur: (1) Wrap-around to the next line will take place in the same manner as during local word processing character entry, or (2) The remaining characters in the incoming line overwrite each other at the right margin. In contrast, the AES machine stores the format settings of the receiving terminal but preserves the line lengths of the transmitting one (as long as the page widths match). For example, if the transmitting terminal is sending a file consisting of lines all 70 characters long and the receiving terminal has its margin set for 50 character lines, the received file on the AES machine will contain 70 character lines. The corresponding file on the MICOM machine would contain 50 character lines. Therefore, if maximum fidelity in transmission is to be achieved, both terminals should have identical margin and page width settings. Even so, the loss of critical control characters such as end-of-paragraph reduces the usefulness of this type of link to the print image transfer of files, where the files contain no special text or control codes. In such cases, the printed version of the received text will be identical to the original. However, if the received file is to be modified, preliminary editing to insert end-of-paragraph symbols should be done in order to preserve the layout of the text. Table 5.1 summarizes the behaviour of an asynchronous AES-MICOM link.

Parameter settings for optimum information transfer are listed below:

Parameter	AES Setting	MICOM Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	nothing	CR
end-of-paragraph representation	CR	N/A
mode of operation	non-transparent	N/A
modem type	full-duplex	full-duplex
XON/XOFF	no	no
respect margins	yes	N/A
right margin wrap-around	N/A	yes

D.2.3 AES Plus - WANG Word Processor 5

These machines can communicate because both are capable of transmitting and receiving the 7-bit ASCII code set. The WANG machine behaves in the following way: text characters are limited to the ASCII set; underscores are stripped off underlined characters; the remaining text and control characters are converted either to a similar character (e.g., accented character to unaccented counterpart) or to spaces; each line of text is delimited by a carriage return character; no format information is transmitted.

The AES machine can be made to behave in a manner similar to the WANG except that the AES machine transmits underlined characters as backspace sequences.

Although both machines boast user accessible conversion tables, the fact that the WANG machine is restricted to 7-bit output characters reduces the usefulness of this feature. It could be helpful though in situations where a small set of special characters is to be transmitted and an equal-sized subset of the ASCII code set is not to be included in the text to be communicated. In such a case, the conversion tables could be modified to suit these special requirements. Because the ASCII character set restricts the amount of information that can be exchanged, a link between these terminals is most suitable for the print image transfer of files containing no special text or control codes. Even then, incompatibilities in page widths may cause reorganization of the received text. If a received file is to be modified, preliminary editing to define paragraph endings would be required in order to avoid destroying the layout of the text. Table 5.1 summarizes the behaviour of an asynchronous link between these two machines.

The parameter settings for optimum information transfer are listed below:

Parameter	AES Setting	WANG Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	CR	CR
mode of operation	non-transparent	N/A
modem type	full duplex	full duplex
XON/XOFF	no	N/A
respect margins	yes	N/A

D.2.4 AES Plus - XEROX 850

The XEROX 850, like the AES Plus, is capable of transmitting and receiving the 7-bit ASCII code set. It does not handle 8-bit data. Its behaviour may be summarized as follows:

Text characters are limited to the ASCII set; underlined characters are transmitted as character-backspace- character; the remaining text characters are translated to a similar ASCII counterpart or are not transmitted at all; control characters are converted where possible to an ASCII equivalent (e.g., end-of-line to CR, end-of-page to FF, tab to HT) or are not transmitted. Format information is not transmitted. The AES Plus can be made to behave in a similar fashion except that horizontal tabs are transmitted as a sequence of spaces.

The net result of this is that file transfers between AES and XEROX machines have the typical characteristics of all 7-bit ASCII asynchronous communication links. The received file would contain only ASCII text characters; given the right circumstances (i.e., the original file contained only ASCII characters and both machines had compatible page width and tab settings), an exact copy of the original file could be printed out at the receiving terminal with no editing required. Otherwise the received file is different from the original and normally would require editing. The first step in this process is usually to delimit paragraphs in order to avoid destroying the layout of the text.

Table 5.1 summarizes the behaviour of an asynchronous link between these two machines.

Parameter settings for optimum information transfer are listed below:

Parameter	AES Setting	XEROX Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	CR	CR
mode of operation	non-transparent	N/A
modem type	full-duplex	full-duplex
XON/XOFF	no	no
respect margins	yes	N/A

D.2.5 MICOM 2000/2001 - MICOM 2000/2001

The MICOM asynchronous communications package allows the transmission of the 7-bit ASCII set of characters only. As a result, it is not ideally suited to MICOM-to-MICOM communication; many text and control characters and all format information is lost. Thus, MICOM-to-MICOM transmission is suitable only if a printed image of a file containing only simple text is the desired outcome of file transfer activity. Even then, both terminals must have identical margin and page width settings if the format of the received text is not to change. If the received file is to be edited and modified, then some preliminary editing is required to define paragraph endings; this is to prevent the layout of the text from changing radically during subsequent editing.

The behaviour of an asynchronous MICOM-to-MICOM link is summarized in Table 5.1.

The parameter settings for optimum information transfer are listed below:

Parameter	MICOM Setting
code set	ASCII
number of data bits	7
parity	odd
end-of-line representation	CR
modem type	full duplex
XON/XOFF	no
right margin wraparound	yes

D.2.6 MICOM 2000/2001 - WANG Word Processor 5

The WANG and MICOM asynchronous communications packages are very similar in that the 7-bit ASCII code set defines the upper limit on the amount of information the machines can transfer. As a result, many text and control characters may be lost along with all format information. Consequently, the exchange of information between these two terminals can be most successful when the desired outcome is a printed version at the receiving terminal of the file contents at the transmitting station. This requires that the file contain simple text (i.e., only ASCII characters) without underlines or control characters, and that the margin and page width settings be compatible. Otherwise, editing of the received file will be needed. If the original text contains characters other than ASCII characters, the MICOM-to-WANG transfer will be more painful because the received file will have to be purged of unwanted question marks. A WANG-to-MICOM transfer is less troublesome because inserted spaces are less messy and can often be left in.

Table 5.1 summarizes the behaviour of an asynchronous MICOM-WANG link.

The parameter settings for optimum information transfer are listed below:

Parameter	MICOM Setting	WANG Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	CR	CR
modem type	full-duplex	full-duplex
XON/XOFF	no	N/A
right margin wrap-around	yes	N/A

D.2.7 MICOM 2000/2001 - XEROX 850

These two terminals can communicate asynchronously using the 7-bit ASCII code set. There are only minor differences between these two systems, so a link between them is suitable for the exchange of print image file data where no special text or control codes are included. Both machines transmit underlined characters as a backspace sequence and both convert where possible internal control codes to ASCII equivalents (this includes the Horizontal Tab character). One difference between the terminals is in the way they treat unexpected characters. The MICOM machine transmits all unexpected characters (i.e., all those which do not have an ASCII counterpart) as question marks. The XEROX machine is more discriminating, converting some characters to a similar ASCII counterpart (e.g., accented to non-accented character) and ignoring the remainder. Since format information is not transmitted by either machine, care must be taken to ensure that page width, tab and margin settings correspond at both sites if the layout of the text is to be preserved. As in other cases, preliminary editing to define ends of paragraphs should be done to a received file before proceeding to any activity apart from printing.

Table 5.1 summarizes the behaviour of an asynchronous MICOM-XEROX link.

The parameter settings for optimum information transfer are listed below:

Parameter	MICOM Setting	XEROX Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	CR	CR
modem type	full-duplex	full-duplex
XON/XOFF	no	no
right margin wraparound	yes	N/A

D.2.8 WANG Word Processor 5 - WANG Word Processor 5

Of the two available code sets, the 7-bit ASCII code set provides the maximum amount of information transfer for the WANG machine in asynchronous communication. It is therefore the preferred choice. However, it does not allow the transmission of many special text and control characters or of any format information. As a result, its suitability is restricted to the transfer of print images of file contents where no special text or control characters are involved (including underscores). If a received file is to be modified, preliminary editing to define paragraph endings should be done in order to avoid destroying the layout of the text.

Table 5.1 summarizes the behaviour of an asynchronous WANG-WANG link.

The parameter settings for optimum information transfer are listed below:

Parameter	WANG Setting
code set	ASCII
number of data bits	7
parity	odd
end-of-line representation	CR
modem type	full duplex

D.2.9 WANG Word Processor 5 - XEROX 850

These two terminals can communicate asynchronously using the 7-bit ASCII code set. There are only minor differences between these two systems, so a link between them is suitable for the exchange of print image file data where no special text or control codes are included. Underlined characters will have their underscores stripped off because the WANG machine cannot handle them. Other differences between machines are related to the handling of unexpected characters. The WANG CWP converts them to an ASCII equivalent or to spaces (including the Horizontal Tab) while XEROX either converts them to an ASCII equivalent or does not transmit them at all (in that case, the Horizontal Tab is transmitted as the ASCII character HT). Since format information is not transmitted by either machine, care must be taken to ensure that page widths correspond at both sites. Otherwise the layout of the text may be altered. If the received file is to be modified, preliminary editing to define paragraph endings should be done in order to prevent destruction of text layout.

Table 5.1 summarizes the behaviour of an asynchronous WANG-XEROX link.

Parameter settings for optimum information transfer are listed below:

Parameter	WANG Setting	XEROX Setting
code set	ASCII	ASCII
number of data bits	7	7
parity	odd	odd
end-of-line representation	CR	CR
modem type	full-duplex	full-duplex

D.2.10 XEROX 850 - XEROX 850

XEROX has implemented a private asynchronous protocol designed for the transfer of files between two XEROX machines. All information is transmitted in the XEROX internal code set so all text and control characters are preserved. All format information is transferred successfully as well so no information loss takes place, as revealed in Table 5.1. This protocol is an asynchronous implementation of the BSC line protocol; it is not compatible with any other.

D.3 BSC Level

In the following subsection, the characteristics of each manufacturer's BSC based communications packages will be analyzed to determine which CWPs can talk to each other and how well. All combinations of the five CWPs under study will be investigated to determine the attainable limits to which communication between two machines may strive.

D.3.1 AES Plus - AES Plus

AES has developed its own private BSC-based protocol for synchronous point-to-point communication between two AES Plus machines. When this protocol is used, the entire file contents are transferred successfully. Table 5.2 summarizes this fact. The only possible source of data-loss occurs when two machines with different video character generators are communicating. As described in Section D.2.1, this leads to a potential loss of textual information (typically character accents). This protocol cannot be used when communicating with non-AES machines.

D.3.2 AES Plus - IBM OS/6

AES offers a 3780 emulation while IBM has a 2770 emulation package. Because 3780 is a subset of the 2770 protocol, communication between these two machines is feasible.

Both machines communicate with the EBCDIC code set. IBM actually uses an extended version of this code set, one that includes 15 additional control characters. Appendix C contains a list of the control codes of the EBCDIC/WP set. The code translation table for the AES machine is user modifiable (the IBM one is not) and so can be adapted to suit the EBCDIC/WP set. In this way most important control functions (eg., end-of-paragraph) can be exchanged between these machines. However, some functions have different representations which makes exchange difficult. For example, AES represents underline sequences by control character-text characters-control character while IBM uses a single control character after every underlined word. This difference cannot be resolved using a simple translation table.

The IBM terminal has a fixed number of positions in its line code set for text characters. Some of these positions always represent the same character (e.g., A, B, 1, 2, etc.), but other positions are variable, depending on the particular keyboard layout in use. This permits the transmission of a greater number of characters than the number of positions allocated for text characters in the EBCDIC code set would normally allow. When a file is transmitted from the IBM machine, identification of the current keyboard selection is sent via special commands embedded in the text. These commands form part of an Operator Control Language (OCL) which also is used to determine the format specifications for a text. These commands are of course foreign to AES and other non-IBM machines. If they are transmitted by the OS/6, they are interpreted by the AES Plus as text and thus contaminate the actual textual data.

Fortunately, the transmission of OCL text can be inhibited. If this is done, then the transmitted text may consist solely of textual material with embedded EBCDIC/WP control functions. An AES machine receiving such text would interpret the characters according to the standard EBCDIC set (i.e., keyboard identification information is lost). This may result in misinterpretation of text.

Because the IBM formatting information would not be transmitted in this environment, the format information stored at the AES machine is derived from local settings at the time of reception. The IBM machine behaves in the same manner when receiving AES text.

The IBM OS/6 is capable of transmitting files in two formats: media image and page image. Media image text is sent as it appears on disk. Page image text is sent as it would appear on the printer; this might include the merging of top and bottom margin text with the main body of page material. Because this latter option transmits more information, it is the preferred choice.

Although it is possible for both machines to operate in transparent mode, non-transparent operation is also workable as all possible codes that can be handled by both systems can be sent successfully in non-transparent fashion. Also, both CWPs perform blocking and unblocking of data in compatible fashion in non-transparent mode.

In summary, communication between these two systems is useful for the exchange of files having variable length lines and which contain only compatible text and control codes. In this case, the received file would be an exact image of the original except that format information is not preserved. However, no preliminary editing of the file would be required before modifying it further. Table 5.2 summarizes the behaviour of a synchronous AES-IBM link.

Parameter settings for optimum performance are listed below:

Parameter	AES Setting	IBM Setting
protocol	3780	2770
code set	expanded EBCDIC	EBCDIC/WP
number of data bits	8	8
mode of operation	non-transparent	non-transparent
block size (bytes)	512	512
handling of underlines	transmit special character	transmit special character
handling of text characters	transmit single character	N/A
number of characters/ record	variable	variable
receive pagewidth	user selectable	N/A
page image	N/A	yes
send format	N/A	no
data option	N/A	no

D.3.3 AES Plus - MICOM 2000/2001

Both of these machines offer 3780 emulations which allow them to communicate with each other. The AES offering is restrictive in that it is available only with machines having a certain hardware configuration (extended disk option).

The compatible code set for these two machines is EBCDIC. Both machines allow the user to define additional codes for characters which are outside this set and all characters may be transmitted transparently or non-transparently. Thus, it is possible for all text characters which are common to both machines to be transmitted successfully; this, however, requires access to the AES internal code set. While the AES machine gives the user the choice of four ways of dealing with underlined and composite characters, the MICOM machine does not handle underlines or any character mapping apart from one-to-one. Thus any such information is lost. Also, if an AES machine is transmitting a file generated on another AES machine with a different character generator, then there is a potential data loss as characters are moved from disk to video memory to communication line. Similarly, the range of characters that can be received is limited by the range of the video character generator.

In normal non-transparent operation only a small number of control codes (e.g., horizontal and vertical tab) are transmitted. Because the translation tables are user accessible and transparent operation is available, it is possible to also transmit control codes that are outside the standard EBCDIC set. Unfortunately, the MICOM machine restricts this translation process to codes which have a visual representation (e.g., end-of-paragraph) and AES requires access to the internal code set. Given that this is possible, though, it means that a significant amount of functional information can be transferred between the AES and MICOM machines. This is because most important control functions exist in all machines and these tend to have visual representations.

Even if it becomes possible to exchange important control codes, the limitations of the 3780 protocol emulations still restrict the usefulness of this venue. For example, the MICOM implementation forces all files that are to be transmitted to have 80-character lines; similarly, on reception, it will insert a required CR (i.e. end-of-paragraph) after every received line. Line lengths on reception are predefined and range from 80 to 250 characters. This works fine if all the text to be transmitted and received conforms to this format; otherwise, the layout of the original text is not preserved. The AES machine is not so restrictive: input files are not compelled to be in 80-column format and required CRs need not be appended to received records.

Another restriction of these 3780 emulations is that they do not preserve format information. Although both implement the horizontal format control feature, the associated command sequence is not generated automatically, and the tab stop information is not preserved with the received file; this feature is used solely to allow the transmission of the HT character in lieu of sequences of spaces. Because format information is not exchanged, the format information such as margin and tab settings that is stored with the received file corrresponds to the settings in effect at the receiving terminal.

Thus, communication between these CWPs is most effective when transmitting 80-column files in transparent operation with the translation tables set up to handle the greatest number of text and control characters. In this configuration, it is more powerful than an asynchronous link between these machines, although it is not normally possible to transfer an exact image of the source file. Operation in non-transparent mode may be slightly less powerful because less information can be transferred. Table 5.2 summarizes the behaviour of a synchronous AES-MICOM link.

Parameter settings for optimum performance are listed below:

Parameter	AES Setting	MICOM Setting
protocol	3780	3780
code set	expanded EBCDIC	expanded EBCDIC
number of data bits	8	8
mode of operation	transparent	transparent
80-char records/block	one	one
handling of underlines	transmit single character	N/A
handling of text character	transmit single character	N/A
interpret printer commands	no	N/A
receive page width	80	80

D.3.4 AES Plus - WANG Word Processor 5

Both of these machines offer 3780 emulations so there is no problem having the two machines communicate. The AES offering is restrictive in that it is available only with machines having a certain hardware configuration (extended disk option).

The compatible code set for these two terminals is EBCDIC. In both cases, the translation tables are modifiable, thereby permitting the transmission and reception of codes which do not normally form part of the EBCDIC set. Because transparent operation is available, a greater number of codes can thus be accommodated. In this way, given access to internal code set representations, it should be possible to transmit all character codes common to both machines. Further, the fact that the machines do not have the same internal set of character representations (AES has a more extensive one), that only one-to-one character mappings are accepted by WANG, that WANG does not handle underlines and that the AES video memory cannot store all possible characters indicates that full interchange of all text information is not possible.

As far as control functions are concerned, WANG does not allow the transmission of special control characters outside the standard EBCDIC set. As a result, file transfers from WANG to AES are limited to textual information only. In the other direction, it should be possible to adapt the WANG translation tables to accept compatible control codes so that file transfers from AES to WANG could include both text and control information.

As with AES-MICOM communication, the protocol behaviour has significant impact on the organization of the received file. The WANG machine transmits 80-character records with end-of-line characters stripped off (it does not force input files to be in 80-column form); on reception, it accepts line lengths of 80 or 132 characters and places an end-of-line character at the end of each received line. This works fine as long as files to be transmitted happen to have the appropriate line length. Otherwise, the layout of the transmitted and received text is altered. The AES machine in line printer emulation expects to receive end-of-line characters to delimit each line of received text. Failing this, it will truncate all text past the expected line width. This is inappropriate for communication with a WANG machine because WANG never sends end-of-line characters. Therefore, card punch emulation is preferred in this case, such that an end-of-line character will be placed at the end of each received 80-character record. This of course could cause reorganization of received text.

Neither machine exchanges format information. Because of this, the format information such as margin and tab settings that are stored with the received file correspond to the settings in effect at the receiving terminal.

Thus, communication between these CWPs is most effective when transmitting 80-column files in transparent operation with the translation tables adopted to provide maximum exchange of text and control information. It is then more powerful than an asynchronous link between these machines, although it is not usually possible to transfer an exact image of the source file. File transfers from WANG to AES are less effective than in the reverse direction because of WANG's inability to transmit special control characters. Table 5.2 summarizes the behaviour of a synchronous AES-WANG link.

Parameter settings for optimum performance are listed below:

Parameter	AES Setting	WANG Setting
protocol	3780	3780
code set	expanded EBCDIC	expanded EBCDIC
number of data bits	8	8
mode of operation	transparent	transparent
80-char records/block	one or six	one or six
handling of underlines	transmit single character	N/A
handling of text characters	transmit single character	N/A
interpret printer commands	no	N/A
receive page width	80	80

D.3.5 AES Plus - XEROX 850

The XEROX 850 offers 2780 and 2770 emulations while AES offers a 3780 emulation. Of the XEROX emulations, only the 2780 emulation currently operates in transparent fashion. For non-transparent operation, the XEROX 2770 emulation is preferred because more information can be transferred.

The XEROX 2780 emulation is unique in that in transparent mode, no code translation takes place and all internal codes are transmitted; in this case, the 2780 protocol serves simply to block the data into 80 character records with no significance attached to this process. Similarly, all received codes are not translated and are stored as is on disk. In this way, a complete file image can be transferred successfully. The receiving terminal must be able to convert received characters into the local machine's representation. With the AES Plus and its user accessible conversion table, this is possible as long as internal code set information is available.

Thus, a transparent link between an AES running 3780 and a XEROX running 2780 will be totally successful in transferring all compatible text and control codes. Those that do not match are converted into similar ones or are discarded. However, there is a problem with this approach. Format information for the XEROX machine is embedded in the file as a text string delimited by a control character. This information is always transmitted by the XEROX in transparent mode; the AES machine can discard the special control character but will treat the format information as text. Such format information would have to be edited out if the original file data is to be preserved intact. Further, with the AES machine, the format information stored with a received file is taken from the local settings at the time the file was received and may not be consistent with the layout of the received data. In the other direction, from AES to XEROX, compatible text and control codes can also be transferred successfully, but once again, the format information stored with the received file may not correspond to that in the source text. It may be possible in this case to obtain a printed version of the original text without performing any preliminary editing.

If the user is willing to trade off the ability to send all compatible control codes and text characters, with the limitation of having to edit the received file at the AES end, against the ability to send most compatible control codes and some text characters, with the possibility that the received file will not have to be edited, then the 2770 protocol in non-transparent mode should be adopted for the XEROX machine.

In this configuration, the XEROX CWP is capable of transmitting the EBCDIC/WP code set, i.e., the standard EBCDIC set plus an additional set of 15 control codes which happen to be used by the IBM OS/6 system. The net effect then is that many of the common control codes can be exchanged between the AES and XEROX machines but no special text characters. Format information stored in a received file would correspond to the current settings at the receiving terminal. Page width incompatibilities between machines may cause reorganization of received text.

Thus, communication between these two machines via a 3780/2780 link does allow the exchange of much textual and control information, and as such is more powerful than an asynchronous link. Yet, the transfer of format information is a problem; in XEROX to AES transmission, the received file must be edited to remove embedded format-related text strings. An alternative approach, using a 3780/2770 link, which limits the amount of textual and control information which can be exchanged, has the advantage that editing of received files may not be required. Table 5.2 summarizes the behaviour of a synchronous AES-XEROX link.

Parameters for maximum transfer of information are listed below:

Parameter	AES Setting	XEROX Setting
protocol	3780	2780
code set	expanded EBCDIC	XEROX internal
number of data bits	8	8
mode of operation	transparent	transparent
80 char records/block	one	one
handling of underlines	transmit control characters	transmit control characters
handling of special characters	transmit single character	transmit single character
interpret printer commands	no	N/A
receive page width	operator selectable	156

D.3.6 IBM OS/6 - IBM OS/6

Unlike the other vendors in this study, IBM has not implemented a special protocol for communication among its machines. Instead, it uses the 2770 protocol with an extended code set, the EBCDIC/WP set (see Appendix C). This set has 15 additional control codes defined for it. This enables the machine to transfer most of the internal control functions. Those that are not transmitted (compare table 3.6 with Appendix C) are mostly related to printer functions and do not affect editing capability. The typestyle and keyboard change control codes are converted to a stop code on transmission. This multiple-to-one code translation makes it impossible for the receiver to identify the original code. However, these codes occur only when midline changes are desired. If changes are to be made between lines of text, this can be achieved through Operator Control Language (OCL) commands which are transferred correctly.

Thus it is possible for two IBM OS/6 machines to exchange in most cases file contents without data loss; all format information, all code set information and almost all control codes can be transferred. This is summarized in Table 5.2.

Parameters for optimum information transfer are listed below:

Parameter	IBM Setting
protocol	2770
code set	EBCDIC/WP
number of data bits	8
mode of operation	non-transparent
block size	512
media image	yes
send format	yes
record length	variable
data option	no

D.3.7 IBM OS/6 - MICOM 2000/2001

IBM offers a 2770 emulation while MICOM has a 3780 emulation package. Communication between these machines is possible because 3780 is a subset of 2770.

Both machines communicate with the EBCDIC code set; the IBM OS/6 uses an enhanced version, called EBCDIC/WP which includes 15 additional control codes. The IBM translation table is not user modifiable while the MICOM one is. It enables the user to specify additional code translations beyond the basic EBCDIC set but restricts them to codes with a visual counterpart (i.e., they can be entered from the keyboard). This allows some of the compatible IBM control functions (e.g., end-of-paragraph) to be transferred successfully.

There are only a limited number of positions available for text characters; in the IBM system, some of these positions are variable and may represent different characters according to the keyboard selection currently in force. The MICOM CWP has no such capability and transmits and receives only the standard EBCDIC configuration. As a result, misrepresentation of text characters may occur.

The OS/6 system embeds formatting information as text strings inside the file through its Operator Control Language (OCL) facility. When transmitting from IBM to MICOM, these OCL strings would be interpreted as data by MICOM and would corrupt the textual material. It is better then to have this information suppressed during transmission (which can be done). In this case, the format information stored with the received file on the MICOM machine is taken from default settings at the receiving terminal. Similarly, in the reverse direction, MICOM does not transmit format information so the IBM machine uses default settings to determine the format information stored with received files.

Protocol considerations limit the effectiveness of a MICOM-IBM link. The MICOM machine forces all transmitted files to be in 80-column format. End-of-line characters are not transmitted and a required CR is inserted at the end of every received line. The IBM machine can be made to behave in a similar fashion, if transparent operation with 80-character records is selected. In this case, successful file transfers are possible as long as they are in 80-column format and don't include incompatible text and control characters. Otherwise, the layout and content of the received file may not match the original. The behaviour of a synchronous IBM-MICOM link is summarized in Table 5.2.

Parameter settings for optimum performance are listed below:

Parameter	IBM Setting	MICOM Setting
protocol	2770	3780
code set	EBCDIC/WP	expanded EBCDIC
number of data bits	8	8
mode of operation	transparent	transparent
number of characters/-record	80	80
receive page width	80	80
page image	yes	N/A
send format	no	N/A
data option	no	N/A

D.3.8 IBM OS/6 - WANG Word Processor 5

The WANG CWP running a 3780 emulation can communicate successfully with an IBM machine with 2770 emulation because the former protocol is a subset of the latter.

Both machines communicate with the EBCDIC code set; the IBM OS/6 uses an enhanced version, called EBCDIC/WP, which includes 15 additional control codes. The IBM translation table is not modifiable while the WANG one is, but the WANG CWP does not allow the transmission of special function codes. Thus, it is possible for compatible control codes to be sent from the IBM to the WANG machine provided that access to the internal WANG codeset is possible.

There are only a limited number of positions reserved for text characters in the EBCDIC code set; in the IBM system, some of these positions are variable and may represent different characters according to a currently selected keyboard configuration. The WANG machine has no such capability and handles only the standard EBCDIC codeset. As a result, misrepresentation of characters may occur.

The IBM formatting information embedded in files as Operator Control Language (OCL) commands would be interpreted as text by the WANG system with a resulting contamination of the original text. It is better than to inhibit the transmission of such information (which can be done). In this case, the format information stored with the received file in the WANG system contains default values generated by the receiving machine.

Protocol considerations limit the effectiveness of an IBM-WANG connnection. The WANG machine does not transmit end-of-line characters and inserts them at 80 or 132 character intervals in the received text. The IBM machine can be made to behave in similar fashion if transparent operation with 80-character records is selected. In this case, successful transmission is possible as long as files are in 80-column format and don't include incompatible text and control codes. Otherwise, the layout and content of the received file may not match the original. The behaviour of a synchronous IBM-WANG link is summarized in Table 5.2.

Parameter settings for optimum performance are listed below:

Parameter	IBM Setting	WANG Setting
protocol	2770	3780
codeset	EBCDIC/WP	expanded EBCDIC
number of data bits	8	8
mode of operation	transparent	transparent
number of characters/-record	80	80
receive pagewidth	80	80
page image	yes	N/A
send format	no	N/A
data option	no	N/A

D.3.9 IBM OS/6 - XEROX 850

Both the IBM and XEROX CWPs offer 2770 emulations and both can use the EBCDIC/WP code set. Further, both machines are capable of taking advantage of the variable text character positions in this code set. However, this latter feature cannot be shared between the two systems because the relevant information is stored differently and is not transmitted. It is possible then for these two systems to transmit files containing compatible text and control characters. All IBM control characters which are transmitted have a XEROX counterpart; insofar as control characters are concerned, an IBM to XEROX transfer is identical to an IBM to IBM one.

Unlike other combinations of differing machines, the IBM and XEROX terminals are capable of exchanging a limited amount of format information, including tab stops, right margin setting and line spacing. This is achieved by transmitting this information in Mag Card II format which both machines can generate and receive. When format information is transmitted in this way, the IBM machine does not transmit any other of its format information (OCL commands).

The 2770 protocol implementations do not have any negative effects on the layout of transmitted files. In summary, it is possible for these machines to communicate successfully and in many cases to transfer exact file images from one system to the other because typical text characters, control codes and format information can all be transmitted. The behaviour of a synchronous IBM-XEROX link is summarized in Table 5.2.

Parameter settings for optimum information transfer are listed below:

Parameter	IBM Setting	XEROX Setting
protocol	2770	2770
code set	EBCDIC/WP	EBCDIC/WP
number of data bits	8	8
mode of operation	non-transparent	non-transparent
number of characters/-record	variable	variable
block size (bytes)	512	512
media image	yes	N/A
send format	Mag Card II	Mag Card II
data option	no	no
send card image	no	no
receive card image	no	no
send end-of-page	yes	yes

D.3.10 MICOM 2000/2001 - MICOM 2000/2001

MICOM has developed its own private BSC-based protocol for synchronous point-to-point communication between two MICOM machines. It provides complete file transfer capability with no information loss as reflected in Table 5.2. This protocol is not compatible with any other.

D.3.11 MICOM 2000/2001 - WANG Word Processor 5

Both of these CWPs offer 3780 emulation packages, communicating with the EBCDIC code set. In both cases, the code translation table is user accessible and transparent operation is possible. This makes it possible to exchange all single text characters which are common to both machines; in the case of the WANG machine, this requires access to the internal code set. Neither machine is capable of transmitting or receiving multiple character sequences representing underlines or other composite characters.

In normal non-transparent operation, only a small number of control codes is available in the standard EBCDIC code set. This can be supplemented by extending the code translation tables of the MICOM machine to handle compatible control functions. The MICOM machine limits this extension capability to characters with a visual representation. Fortunately, most important control codes do have a visual representation. The WANG terminal does not allow the transmission of special control codes so WANG to MICOM transmission is more restrictive than MICOM to WANG.

The 3780 protocol implementations impose a number of restrictions which limit the amount of information which can be transferred. For example, the MICOM machine will only allow the transmission of files having 80-character lines; similarly, on reception, the line length is predefined (in the range of 80 to 250 characters) and a CR is inserted after every received line. In similar fashion, the WANG machine strips off all end-of-line characters during transmission and inserts them on reception at 80 or 132 character intervals. These restrictions do not pose any problem if all text to be transmitted and received conforms to this format; otherwise, text layout may be altered.

In addition, these 3780 emulations do not have any provision for the exchange of format information. As a result, format information stored with received files corresponds to the settings in effect at the receiving terminal.

In summary, communication between these CWPs is most effective when transmitting 80-column files in transparent operation with the translation tables set up to handle the greatest number of text and control characters. In this configuration, it is more powerful than an asynchronous link between these machines, although it is not usually possible to transfer exact file images. File transfers from WANG to MICOM are less effective than transfers in the reverse direction because of WANG's inability to transmit special control characters.

Table 5.2 summarizes the behaviour of a synchronous MICOM-WANG link.

Parameter settings for optimum information transfer are listed below:

Parameter	MICOM Setting	WANG Setting
protocol	3780	3780
code set	expanded EBCDIC	expanded EBCDIC
number of data bits	8	8
mode of operation	transparent	transparent
receive pagewidth	80	80

D.3.12 MICOM 2000/2001 - XEROX 850

MICOM offers 2780 and 3780 emulations while XEROX has 2780 and 2770 packages. All communicate with the EBCDIC code set. While the MICOM machine allows its standard EBCDIC translation table to be extended to enable transmission and reception of additional codes, the XEROX translation table is not user accessible. Thus, in non-transparent 2780 mode, only standard EBCDIC codes could be exchanged successfully between these two machines. In transparent 2780 mode, the XEROX machine transmits its internal code set with no code translation. This code set is an extension of 7-bit ASCII. If this feature is to be useful, the MICOM machine should be able to translate all compatible text and control codes and bypass the standard EBCDIC code translation table. However, the supplementary user-definable translation table can only handle characters with visual representations. Thus, control codes without visual counterparts such as end-of-line, tab, etc., are lost or translated incorrectly (as a result of codes not in the user-defined translation table defaulting to the standard EBCDIC one). Depending on the nature and quantity of such control codes in the source text, the amount of information loss in a file transfer may range from none to severe. In the general case, irrecoverable data loss is the likely result of attempts to transfer files in transparent mode.

Thus, a transparent MICOM-XEROX link is probably not optimal. The choice then is between a 2780-2780 or 3780-2770 configuration. With a non-transparent 2780-2780 link, only standard text and control codes can be transferred. Both machines implement the horizontal format control feature. When the XEROX machine receives the appropriate command sequence, it generates a new format block in the received text to preserve the tab stops; thus some format information can be preserved. XEROX is the only machine which has this capability.

The MICOM terminal only allows transmission of files with 80-character lines; the XEROX 850 does not enforce this. MICOM places required CRs at the end of received lines (preset by receiver in the range 80-250) while XEROX places them at the end of each received record (80 characters).

Thus, 2780-2780 communication between these terminals works best in non-transparent mode when files containing 80-character lines with no embedded special text or control codes are to be exchanged. In this case, a print image of the original file can be obtained.

With a 3780-2770 connection, the MICOM machine behaves as in the 2780 case. The XEROX machine behaves differently. It is now capable of transmitting some control codes through the use of the EBCDIC/WP code set. By expanding the MICOM translation table, it becomes possible to exchange some control codes. In this way, this approach offers an improvement over the 2780-2780 link. However, the XEROX machine is no longer able to interpret horizontal format control sequence, so no format information can be transferred. It can send and receive 80-character card images as desired by the MICOM machine. This approach is considered the most desirable as the ability to exchange control information is felt to be more significant than the loss of some format information. Table 5.2 summarizes the behaviour of such a synchronous MICOM-XEROX link.

Parameters for optimum performance are listed below:

Parameter	MICOM Setting	XEROX Setting
protocol	3780	2770
code set	extended EBCDIC	EBCDIC/WP
number of data bits	8	8
mode of operation	non-transparent	non-transparent
receive pagewidth	80	80
send card image	yes	yes
receive card image	yes	yes
receive pagewidth	80	80
send end-of-page	N/A	yes

D.3.13 WANG Word Processor 5 - WANG Word Processor 5

WANG has developed its own private BSC-based protocol for synchronous point-to-point communication between two WANG machines (called WPS protocol by WANG). It provides a complete file transfer capability with no information loss between these machines, as reflected in Table 5.2. This protocol is not compatible with any other.

D.3.14 WANG Word Processor 5 - XEROX 850

Both of these machines implement 2780 protocols, operating in transparent and non-transparent mode. The WANG implementation uses the EBCDIC code set in both modes; the code translation table is modifiable to add additional text codes. The XEROX machine uses the EBCDIC code set in non-transparent operation, but transmits its internal ASCII-based codeset without translation when transparency is in effect. This characteristic can be taken advantage of because the WANG translation table can be adapted to the particular situation. In this case, the translation table on the receive side would convert all XEROX codes to their WANG representation. In this way, it is possible to transfer all compatible text and control characters from XEROX to WANG, given that internal code set information for each machine is available. Because WANG does not transmit special control codes, the WANG to XEROX file transfer is less powerful.

While it is possible to exchange character information successfully, format information is a different matter. In the XEROX machine, this information is embedded in files as a text string delimited by a special control character. This information is always transmitted in transparent mode, with the result that the WANG machine will consider this information as text. Such information would have to be edited out of the received file if the original file data is to be preserved. Another problem with XEROX to WANG transmission is that the WANG machine inserts the end-of-line character after every received line (either 80 or 132 characters long). This can result in an altered layout of the text. In addition, the WANG machine will store default format information such as margin settings with the received file. In the other direction, from WANG to XEROX, compatible text can be transferred successfully, but the organization of the received text may differ from the original because the WANG machine does not transmit the end-of-line character. Also, format information such as margin settings stored with the received file will be default ones generated at the receiving terminal.

With the above approach it is unlikely that a print image of a text can be transmitted successfully between these two machines. If the objective is to obtain printed versions of original text without requiring editing, then WANG 3780 to XEROX 2770 link in non-transparent mode is probably preferable. In this case, many of the common control codes can still be transferred from XEROX to WANG through the use of the EBCDIC/WP code set on the XEROX machine and a suitably adapted WANG translation table. This configuration works well in both directions with files having 80-character lines with no incompatible text or control characters. With such files, no preliminary editing is required before printing. Otherwise, pagewidth incompatibilities may cause reorganization of text. No format information would be transferred in this environment. The behaviour of a synchronous WANG-XEROX link is summarized in Table 5.2.

Parameter settings for the two configurations discussed above are listed below:

Parameter	WANG Setting	XEROX Setting
protocol	2780	2780
code set	expanded EBCDIC	XEROX internal
number of data bits	8	8
mode of operation	transparent	transparent
80-char records/block	one	one
receive pagewidth	80 or 132	156
protocol	3780	2770
code set	expanded EBCDIC	EBCDIC/WP
number of data bits	8	8
mode of operation	non-transparent	non-transparent
send card image	yes	yes
receive card image	yes	yes
receive pagewidth	80	80
send end-of-page	N/A	yes

D.3.15 XEROX 850 - XEROX 850

XEROX has developed its own private BSC-based protocol for synchronous point-to-point communication between two XEROX machines. It provides a complete file transfer capability with no information loss, as reflected in Table 5.2. This protocol is not compatible with any other.

Appendix E

International Standards Relevant to Communicating Word Processors

BASIC CODE STANDARDS

ANSI X3.4-1977 (Code for Information Interchange)

This standard defines a 7-bit coded character set (called ASCII) to be used for the exchange of information among text processing equipment, communication systems and associated equipment. ASCII consists of 32 control characters, 94 graphic characters, SPACE and DELETE.

FIPS Pub 1 (Code for Information Interchange)

FIPS Pub 1 adopted ANSI X3.4-1968, with the exception of the "New Line" concept. "New Line" is the use of a single keystroke to accomplish the two character sequence CR-LF. The adoption of ANSI X3.4-1977 to update this FIPS Pub is currently being proposed.

ISO 646-1973 (7-Bit Coded Character Set for Information Processing Interchange)

This standard defines a 7-bit coded character set exactly like ANSI X3.4-1977. Two code tables are included in the standard: the Basic Code Table and the International Reference Version (IRV) Table. The Basic Code Table lists the same control characters as ANSI X3.4. However, it provides several undefined graphic character positions to be used as options for specific applications. The IRV differs from ASCII in one graphic position; namely, it replaces the $ symbol with the international currency symbol.

ECMA-6 (7-Bit Input/Output Coded Character Set)

This standard is identical to ISO 646.

CODE EXTENSION STANDARDS

ANSI X3.41-1974 (Code Extension Techniques for Use with the 7-Bit Coded Character Set)

This standard defines two techniques of extending ASCII (it does not define the character sets). First, if only a few characters are needed, extension can be accomplished by substituting unneeded ASCII characters with other characters which are needed. Second, escape sequences can introduce additional single characters or sets of characters to be used in conjunction with ASCII. Escape sequences identify and invoke the control character sets, C0 and C1, and the primary graphic character sets, G0, in cases where multiple G0 sets are used. However, if supplementary graphic character sets, G1, are used in combination with G0 sets, an escape sequence simply identifies the subsequent G1 set, and the control characters shift-out (SO) and shift-in (SI) are used to invoke the G1 set and return to the G0 set, respectively.

ANSI X3.41-1974 also defines an 8-bit code which simply adds a bit to all the ASCII characters and produces an additional 128 characters. The extension techniques for the 8-bit code are the same as for 7-bit ASCII.

When control sets and graphic sets are established for text processing, they will be introduced and invoked by escape sequences and control characters as described in this standard.

FIPS Pub 35 (Code Extension Techniques in 7- or 8-Bits)

FIPS Pub 35 adopts in whole ANSI X3.41-1974.

ISO 2022-1973 (Code Extension Techniques for use with the ISO 7-Bit Coded Character Set)

ISO 2022 is identical to ANSI X3.41-1974. However, it is undergoing revision. The revision identifies two additional elements of extension of graphic sets; namely, the G2 and G3 sets. They will provide up to 94 additional characters per set and will be introduced by escape sequences as defined in 2022. The G2 and G3 sets will be invoked with the control characters single-shift 2 (SS2) and single-shift 3 (SS3), respectively.

ECMA-35 (Extension of the 7-Bit Coded Character Set)

ECMA-35 is identical to ISO-2022-1973.

ADDITIONAL CONTROL FUNCTIONS

dpANS X3.64 (Additional Controls for Use With ASCII)

This standard builds on ANSI X3.4-1977 and ANSI X3.41-1974 to provide additional control codes for CRT display terminals, including communicating word processors, and printers. ANSI X3.64 defines control functions for software and device control strings, editing functions, formatting functions and control sequences with numeric or selective parameter values.

These control functions are defined by independent control characters or by the final character of control strings. Each independent control character (identified by the 2-character escape sequence ESC Fe) is assigned a unique position in a C1 set and is treated as a single additional control to the C0 set.) Control strings are of the form: introducer, parameter string, optional intermediate character, and final character. The function of the control string is determined by its final character, which is assigned a position in columns 4-7 of the 7-bit or 8-bit code table. This standard defines 25 independent control characters and 50 final characters for control strings.

The standard also defines 19 modes which alter the meaning of subsequent control functions. Each mode has two states, set and reset. The modes are divided into 4 classes, according to whether they apply to the terminal and/or display locally or to the data stream being transmitted. For example, in the Format Effector Action Mode the reset state causes format effectors to be performed immediately when received in a data stream, and the set state causes them to be stored in the data stream but not performed when received.

ISO DP 6429 (Additional Control Functions for Character-Imaging Devices)

This standard differs from X3.64 in the number of modes and the number of control functions it defines. DP 6429 has 17 modes; the Line Feed New Line Mode is not defined and Select Editing Extent (SEE) is considered a control function rather than a mode as in X3.64. Because of SEE and Select Size Unit (SSU), this standard lists and defines 80 control functions compared to 78 in X3.64.

ECMA-48 (Additional Control Functions for Character-Imaging I/O Devices)

ECMA-48 is identical to ISO DP 6429.

ADDITIONAL PROPOSED STANDARDS

ISO DP 6937 (Coded Character Set for Text Communication)

This standard is concerned with the control and graphic character sets needed for communicating typewriters and CRT terminals, including communicating word processors. The coding scheme is 7- or 8-bit ASCII as described in ISO 646 and the techniques of extension are in accordance with ISO 2022. ANSI participates in the ISO Working Group responsible for generating DP 6937 and plans to adopt it as an ANSI standard when it is finalized.

References

1. "General Information - Binary Synchronous Communications", IBM Systems Reference Library Publication No. A27-3004-2

2. "IBM 2780 Data Transmission Terminal - Component Description", IBM Systems Reference Library Publication No. GA27-3005-3.

3. "Component Information for the IBM 3780 Data Communication Terminal", IBM Systems Reference Library Publication No. GA27-3063-0.

4. "System Components: IBM 2770 Data Communication System", IBM Systems Reference Library Publication No. GA27-3013-2.

5. "Communicating Word Processors", International Resource Development Inc., April 1979.

6. reported in "Word Processing", January 1980, p.34.

7. "Communicating Word Processors - An Integration Study", Tymshare Inc., June 1979.

8. "User's Guide, TNI 303 Protocol Translator", Telesystems Network Inc., 1980.

9. "Reference Model of Open Systems Interconnection (Version 4, June 1979)", ISO/TC97/SC16/N227., Aug. 1979.

10. "Draft New Question VII/XX - Structure for and Use of a Reference Model for Public Data Network Applications", CCIT/Study Group 8/Com VIII - No. 394E, October 1979.

11. ISO Draft Proposal 6937/I, "Coded Character Set for Text Communication", ISO/TC97/SC2/N1003-1005.

12. "Second Draft Proposed Communication Heading Format Standard", ISO/TC97/SC6/N1948.

13. R.J.A. Buhr, S.A. Mahmoud, K.C. Toth, "Open Systems Interconnection: Issues Associated with ISO Layered Model", DOC Contract Report, March 1980.

14. ISO 2022, "Code Extension Techniques for Use with the ISO 7-bit Coded Character Set", July 1973.

15. see document ISO/TC97/SC2/N991

16. ISO 646,"7-bit Coded Character Set for Information Processing Interchange".

17. "Proposed Text for Draft Recommendation S.D.", CCITT Study Group VIII, January 1980.

18. B.W. Kernighan, L.L. Cherry, "A System for Typesetting Mathematics", Commun. Assn. Comp. Mach., 18(March 1975), pp. 151-157

19. D.E. Knuth, "Mathematical Typography", Bulletin of the American Mathematical Society, March 1979.