OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

This document has been produced without formal United Nations editing. The designations employed and the presentation of the material in this document do not imply the expression of any opinion whatsoever on the part of the Secretariat of the United Nations Industrial Development Organization (UNIDO) concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries, or its economic system or degree of development. Designations such as "developed", "industrialized" and "developing" are intended for statistical convenience and do not necessarily express a judgment about the stage reached by a particular country or area in the development process. Mention of firm names or commercial products does not constitute an endorsement by UNIDO.

FAIR USE POLICY

Any part of this publication may be quoted and referenced for educational and research purposes without additional permission from UNIDO. However, those who make use of quoting and referencing this publication are requested to follow the Fair Use Policy of giving due credit to UNIDO.

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

RESTRICTED

17516

DP/ID/SER.A/1196
9 May 1989
ORIGINAL: ENGLISH

MICROPROCESSOR APPLICATION ENGINEERING PROGRAMME

DP/IND/84/030

INDIA

Technical report: Standardization of interfaces and communications software*

Prepared for the Government of India
by the United Nations Industrial Development Organization,
acting as executing agency for the United Nations Development Programme

Based on the work of David Rowe, expert in standardization of interfaces

Backstopping officer: V. Smirnov, Engineering Industries Branch

United Nations Industrial Development Organization
Vienna

1/58

* This document has not been edited.

V.89-55417

ABSTRACT

This report describes the expert's mission to India between 25 February 1989 and 24 March 1989. The visit was primarily to the Southern Regional Centre of the Microprocessor Applications Engineering Programme, based at Indian Telephone Industries at Bangalore.

The main activities consisted of:

- Detailed technical discussions about specific projects with the MAEP personnel involved.
- Presentation of a series of lectures on Data Communications Standards and the implementation of Communications Software.

The report reviews activities at the Southern Regional MAEP Centre and gives the assessments and specific recommendations of the expert.

In general, the conclusion is that a good start has been made at the Centre and good work has been done in projects already completed. However, as more complex projects are attempted, the narrowness of the skills-base of the team is proving to be an increasing difficulty.

LIST OF CONTENTS

1. Purpose of Visit. 4

2. Review of Specific Projects. 5

   2.1 The Projects 5

   2.2 The Current Status of the Projects . . . 5

   2.3 Specific Technical Advice concerning the Projects 6

      2.3.1 Statistical Multiplexor 6

      2.3.2 Local Area Network 8

3. Evaluation and Recommendations 11

   3.1 Overall Evaluation 11

   3.2 Recommendations - Software Engineering . . 12

   3.3 Recommendations - Specific Design Areas . . 13

   3.4 Recommendations - Planning and Business Strategy 16

4. References 19

APPENDICES

APPENDIX I: Itinerary 22

APPENDIX II: List of Personnel with whom the expert met 23

APPENDIX III: Lecture Series Outline 24

1. PURPOSE OF VISIT.

The purpose of the visit, as defined in the job description was to:

1. Apprise myself of the current status of microprocessor applications in the Indian Industry.

2. Apprise myself of the objective status and the results of various system engineering development projects going on in various centres.

3. Help project personnel in hardware and software development for various projects.

4. Train project personnel as well as centres on new methodologies for microprocessor-based system engineering systems.

On 13 July 1988, a telex was received from Mr. S. Rajaram, the then coordinator for the MAEP project at Bangalore. This telex described the work being carried out on developing a statistical multiplexor and a local area network. It stated that my expertise in software for data communications would be extremely useful and went on to say:

> During your visit, it would be most helpful if you could kindly bring technical material relevant to the hardware as well as the software aspects of data communications systems, particularly as related to local area networks. It would be extremely helpful to us if you could bring appropriate material which would give us full insight into application software modules of computer networks, such as file transfers . . . electronic mail, virtual terminal facilities . . . and remote file sharing.

As a result of this telex, I also prepared myself to advise on the technical aspects of data communications software and identified suitable reference material to take with me.

On arrival in New Delhi, I was briefed by Dr Krishna Kant, Chief Project Coordinator, MAEP, at the Department of Electronics. While stressing that I should review the situation generally, Dr Kant confirmed that the area in which I could be of most help was that of software for data communications.

On arrival in Bangalore, further discussions confirmed that the area where the MAEP team were currently having the greatest problems was in designing and developing the software for various microprocessor-based data communications devices. That is therefore where I devoted my effort.

2. REVIEW OF SPECIFIC PROJECTS

2.1 THE PROJECTS

The two projects with which I was involved were:

Statistical Multiplexor

An microprocessor-based device to multiplex up to eight streams of data from asynchronous devices over a single data communications link. The protocol used over the link is based on LAPB (HDLC Asynchronous Balanced Mode) which has mechanisms for synchronous transmission, error detection and recovery and overall flow control.

The multiplexor is designed around a free-standing Intel 8088 (or similar) microprocessor, a Signetics 2698 octal UART and a Siemens HDLC controller chip.

Local Area Network for IBM-compatible PCs

This consists of hardware and software to provide communications between a network of IBM-compatible PCs.

The hardware consists of a PC board containing a Standard Microsystems COM 9026 Local Area Network Controller which implements the ARCNET token-passing bus protocol.

The software will provide file transfer, electronic mail and the direct exchange of messages between operators.

2.2 THE CURRENT STATUS OF THE PROJECTS

Statistical Multiplexor

Hardware:
The hardware has been designed and is awaiting fabrication. No serious problems are anticipated.

Software:
This is currently at the design stage. Difficulty has been encountered in understanding how to structure the software, particularly in the area of achieving concurrency of input, output and processing in the two different directions of transmission.

Local Area Network for IBM-compatible PCs

Hardware:
Prototype boards have been constructed and appear to be working correctly. These boards are external to the PC and

work is now underway on a more compact version which can plug directly into a slot in the PC bus.

Software:
Some simple software is working, providing a convenient user interface and simple message passing between operators. However, difficulties have been encountered in developing the software to provide remote access to discs.

2.3 SPECIFIC TECHNICAL ADVICE CONCERNING THE PROJECTS.

No particular problems had been reported concerning hardware, so attention was concentrated on the software for the two projects.

2.3.1 STATISTICAL MULTIPLEXOR:

(a) **Problem**: How to model a concurrent system such as this during the program design stage.

Advice: For initial design, use a data-flow based model, such as MASCOT or DARTS; see ALLWORTH & ZOBEL, PRESSMAN and WARD & MELLOR [references are given in Section 4].

(b) **Problem**: How to structure the software so that it is driven by events such as input and output transfers on the 8 asynchronous ports and on the single synchronous port.

Advice: Interrupt service routines must be used for the time-critical parts of I/O device handling, but the amount of work done in the interrupt routines must be minimised.

The major work of assembling and disassembling packets for synchronous transmission must be done in a 'background activity'. There are two ways of structuring this:

(i) Use a process scheduler to support multiple processes and decompose the problem into a number of independent processes. Separate processes are used for inbound data and for outbound data. Further processes may be used for handling supervisory activities such as time-outs and operator commands.

This approach is a traditional way of structuring communications software. However it requires the use of a real-time operating system and suffers from process-scheduling overheads.

(ii) Use an 'event queue' together with a single background activity. The interrupt service routines place 'request blocks' on an event queue; each request block contains information identifying what work needs to be done on what data. The single background task processes the requests serially. High priority events may go to the head of the event queue, but cannot preempt the event currently being processed. This approach has lower overheads and avoids the need for a real time operating system. It is less general that Approach (i); it would not be useful if any background activity had to wait (e.g. for an I/O transfer), but it is quite adequate for this application.

Approach (ii) is recommended for implementing the statistical multiplexor.

(c) Problem: How to organise the buffers? Existing plans were to use fixed length buffers and to use pointers to chain together the buffers constituting each queue; empty buffers would be kept in a common pool.

Advice: There are two groups of buffers in each multiplexor:

The UART input and output buffers; there is one set per asynchronous channel, i.e. up to 8 in total.

The synchronous input and output buffers, i.e. for supporting the HDLC protocol; there is only one set.

The UART interrupt service routines transfer data between the UARTs and the UART buffers; the interrupt service routines for the HDLC controller chip transfer data between the chip and the synchronous buffers; finally the background task transfers data between the two sets of buffers.

My advice is that the synchronous buffers should be organised as chained fixed-length buffers, as originally planned.

The UART buffers could also be organised in the same way, but it would be better to use simple dedicated pair of circular buffers for each UART. This is easier to implement and avoids the problems of 'buffer-hogging' by very busy channels.

There must be a range of flow-control mechanisms sufficient for all circumstances; e.g. there should be a separate flow control mechanism for each direction of every asynchronous channel; this will prevent an overload on one asynchronous channel from locking up the flow on all channels. It is also essential to prevent the two multiplexors being unable to exchange control messages because the synchronous link has been saturated by the aggregate data flow.

(d) Problem: How to communicate control information between the multiplexors; e.g. to configure ports in and out, to change port speed, etc.

Advice: It is necessary to design a simple high-level protocol to sit on top of the HDLC protocol to distinguish between data and the different types of control message being exchanged between the multiplexors.

2.3.2 LOCAL AREA NETWORK FOR PCs:

The current software has severe structural limitations in its current form:

(a) **Problem:** When one PC sends a request to another, the software at the receiving end does not work if an MS-DOS service is invoked.

Advice: The fundamental problem is that the receiving software has been written to run entirely within the interrupt service routine for the COM 9026 local area network control chip. This leads to two problems if the receiving software tries to call an MS-DOS function:

- The MS-DOS function will be running with interrupts either wholly or partly masked off. If the MS-DOS function relies on the use of interrupts, e.g. for synchronising I/O, then it cannot work correctly.

- The code for MS-DOS is not reentrant. Consider the following circumstances:

    - An ordinary application program is running.
    - The application program calls an MS-DOS function.
    - While the MS-DOS function is executing, a LAN interrupt occurs as a result of an incoming message.
    - The interrupt service routine happens to call the same MS-DOS function.

  MS-DOS is implemented in such a way that there is only one copy of working variables for MS-DOS functions. Thus, the second call of the MS-DOS routine corrupts the data set up by the first call, so when the first call is resumed, it misoperates.

Even if the above problems did not occur, it is undesirable to invoke a lengthy operation, such as an I/O transfer, within an interrupt service routine, since while an interrupt service routine is running, the system is unable to respond to fresh interrupts of the same class. Suppose there is a network for at least three PCs; what happens if PC2 sends a request to PC1 and while this is being serviced, PC3 also sends a request to PC1? There is a danger of the second request being lost.

The solution to this problem is to move the servicing of in-bound requests out of the interrupt service routine into a background server process. This means implementing a simple process scheduler with a mechanism to prevent one process preempting another during execution of an MS-DOS function. An example of such a system is given in BIGGERSTAFF, Chapters 6,7 & 8 [See references in Section 4]

(b) **Problem:** The set of protocols currently used is extremely simple, consisting of only two layers:

- Data Link Layer: the ARCNET protocol as implemented directly by the COM 9026 chip.
- Application Layer: a very simple protocol to distinguish between different types of request, e.g display message, transfer file, send mail, etc.

While the use of simple protocols is not necessarily a problem, there are limitations which the designers should clearly understand.

- The implementation of ARCNET relies entirely on the COM 9026 chip to provide a **reliable** data-link service. This is largely satisfactory, but there is one case in which software intervention is necessary: if one station tries to send a message to another and the target station is operational, but its software has not allocated a buffer, then the sender will try to send the message indefinitely. This will prevent the sender from sending messages to any other station. There must be a software timeout mechanism to abort a transmission attempt if it is not successful within a reasonable time.
- Stations are addressed by physical addresses. Thus in a network of many PCs, every station must know the physical addresses of all the stations with which it needs to communicate. The alternative is to use an address resolution protocol together with a mechanism to derive physical addresses from symbolic addresses.
- No logical link protocol is used (e.g. IEEE 802.2). The main result is that it is difficult to multiplex system management messages together with data transfers. [See Item (ii) of Section 3.3]
- No network protocol is used, thus making inter-networking difficult to implement.
- No transport protocol is used. This is acceptable if multiplexing is not needed and if the data link service

is truely reliable. However, an alternative LAN protocol is ever used (e.g. Ethernet) then a full transport service will be needed.

- In the application protocol, there are no mechanisms for acknowledgements and flow control. If a file is to be transferred to another station, then the only form of acknowledgement is the ARCNET acknowledgement that the block has been received; there is no acknowledgment that the data has been successfully written to disc. [see Item (i) of Section 3.3]

- Currently, when a file is to be transferred to another system, then the complete file is first copied across the network to memory in the target and then it is written from memory to disc. Consequently, the amount of free memory in the target limits the maximum size of file which can be transferred. This is undesirable.

- It is not clear how the application level protocol handles the case where one station is in the middle of servicing a series of requests (e.g. file transfer) from another station and a third station starts sending new requests. Can the two sets of requests be interleaved, or does the second set have to wait until the first has finished.

- There seem to be no facilities for network management [see Item (ii) of Section 3.3].

Advice: A careful evaluation needs to be made of exactly what capabilities the system should have and consequently, what protocols are needed. This is particularly important for the application level.

(c) Problem: All the services currently planned for the system (e.g. message passing, file transfer, electronic mail) are provided to the human operator. Commercial local area networks also provide services to application programs, e.g. access by an application program to remote files. This level of service is considerably more difficult to implement, as it means, in effect, producing a distributed operating system. Requests from an application program to its local operating system must be intercepted and turned into network requests if the request involves remote data.

Advice: The implementation of a distributed operating system is probably beyond the current capabilities of the team. If it is planned to make this level of services available, then a commercially available product should be bought-in, e.g. Microsoft's MS-NET/MS-Redirector or Novell's Netware

3. EVALUATION AND RECOMMENDATIONS

3.1 OVERALL EVALUATION

A good start has been made at the Southern Regional MAEP Centre. A competent team has been assembled, adequate equipment has been acquired and significant projects have been completed.

However, serious difficulties are now arising. More ambitious projects are now being attempted and these require a wider variety of skills than before; the inexperience of the team in areas outside of traditional Electronic Engineering is proving a severe handicap.

These problems are not unique to the team at Bangalore. For five years, I was responsible for running a post-graduate course in Australia and I continually encountered exactly the same problems with Australian engineers and computing professionals. Data Communications is an interdisciplinary subject involving Communications Engineering, Computing and also Business Studies and Management. Very few individuals have had the opportunity to qualify or gain experience in more than one of these areas.

The projects now being undertaken are intended to develop products for possible manufacture and general sale in India, although it is understood that the actual marketing of such products (and possibly also the manufacture) may be undertaken by another organisation.

The products of earlier projects were either for in-house use or were components of larger systems. The new products must be commercially viable in their own right. It was merely sufficient for the earlier products to work, whereas the new products must not just work, but they must work better than competing products; that is, they must be better at satisfying the requirements of a significant sector of the market.

Designing products for a competitive market places greater demands on the development team; in effect, the team must out-perform the competitors' development teams. Furthermore, the marketplace demands an on-going commitment to product improvement, since as soon as a successful product is introduced, then competitors will develop new products to erode its competitive advantages.

There are considerable benefits in adopting a policy of developing products for the open competitive market; e.g. reduction of imports, possible future exports, incentive for skilled personnel not to emigrate, etc. There is no reason to think that the MAEP team at Bangalore could not be as successful at developing competitive products as teams elsewhere in the world. Overseas teams have no monopoly of ability, education and enthusiasm, and similar basic components are available in India to those available to everyone else. However, if it is to be done, it must be done properly; *in order to be successful, the Bangalore team must be strengthened in a number of areas:*

(1) The team must acquire expertise in software engineering, particularly for system and program design.

The team also needs to make stronger use of project management techniques to identify problems early and to ensure that products are developed on time and within budget.

(2) There are also a number of specific technical areas in which the team needs more expertise, e.g. data communications applications, database, system management and software reusability.

(3) A strong market-orientation must be developed so that products produced really meet customer needs and are competitive.

These points are developed below

3.2 RECOMMENDATIONS - SOFTWARE ENGINEERING.

Currently, the members of the MAEP team at Bangalore are all qualified as electrical/electronic engineers. Unfortunately, the only software training they received in their degree courses consisted of microprocessor programming in assembly language. In most engineering degree courses, there simply is no time for a greater coverage of software.

In the last few years, a new disciple referred to as 'Software Engineering' has been developed to deal with the problems of developing large-scale and complex software. Such software is typically developed by large teams and may be in use for periods of ten to twenty years and so effective management and coordination of the development team is vital.

Large-scale software usually has a well defined 'life cycle' consisting of the following stages:

- Requirements specification
- System design
- Program design
- Program implementation
- System integration and testing
- Maintenance

A range of methodologies have been developed for each stage in this life cycle. These methodologies are especially important in the early stages of a project since incorrect design decisions made at this stage are the most expensive to correct later.

There are also established standards for the documentation necessary for adequate communication both within the development team and with customers and other concerned parties.

Software Engineering is usually used in conjunction with project management techniques to monitor progress and take early remedial action when problems arise.

It is strongly recommended that urgent steps be taken to ensure that the MAEP team at Bangalore acquires more expertise in Software Engineering, and puts that expertise into use. As it is highly likely that the other MAEP centres in India are encountering similar problems, this action should be coordinated nationwide. A strong commitment must be made at all levels of management, since Software Engineering must be introduced within an organisation from the top down; it is not a purely technical matter which can be allowed to percolate upwards from the junior members of a team.

Possible actions to acquire expertise in Software Engineering are:

(a) Hire graduates in Computing to join the MAEP team. Most courses in Computing now include a strong component of Software Engineering. It must be recognised that Data Communications is an inter-disciplinary area and a team with a variety of skills is necessary to handle all the problems which might arise.

(b) Encourage existing members of the team to broaden their experience, e.g. by reading books and journals, by attending courses, etc.

(c Make contact with existing centres of expertise in Software Engineering. These are likely to be in Computing departments of universities and in groups developing large-scale software systems, e.g. government, banks, airlines, software contractors, avionics.

Once identified, suitable activities could be arranged, e.g:
- Visits & discussions.
- Short term secondment of staff.
- Conferences.
- Short courses by guest speakers.

(d) Arrange help from UNDP experts in Software Engineering.

3.3 RECOMMENDATIONS - SPECIFIC DESIGN AREAS

As well as the general area of Software Engineering, there are some other areas to which the development team have paid insufficient attention, largely because of lack of experience. These areas are:

(i) Computer Applications, Distributed Processing and Databases.

Data communications is not an end in itself, it is a component of a complete computer application. One cannot appreciate the full significance of data communications without understanding the context in which it is used.

This applies particularly to the role of data in applications. In most communication-based applications, the whole reason for the existence of the system is to enable people to have access to accurate data which may be in a different location. Database systems are the technology used to store and retrieve the data and although database technology is different to communications technology, the two are frequently used together. Anyone working seriously in data communications must therefore have an appreciation of database technology.

In many applications the only copy that may exist of data may be inside a computer, i.e. the may be no copies on other media such as paper. However things will go wrong; hardware will fail, software will contain errors, data communications links will break. It is therefore vital to design systems so that no matter what failures occur, no critical data can be lost permanently.

This is particularly important in the design of application level protocols in data communications. It is not sufficient to merely ensure that the data is **transmitted** successfully; the protocols must ensure that it is **processed** successfully.

In the case of the two current projects, these issues are of most importance in the PC Local Area Network rather than the Statistical Multiplexor.

(ii) System Management.

There are always two groups of people involved with computer-based systems:

- The **users** of the system.

- Those responsible for managing and maintaining the system.

Any system must provide facilities to make the system manageable. In the case of data communications such facilities include:

- **Configuration Management**: Most computer networks are in a constant state of flux; new devices are added, old one are removed, devices are moved to new locations, new types of device replace old ones, etc. There must be facilities for reconfiguring the network without disrupting work unnecessarily.

- **Fault Detection Diagnosis & Recovery:** Faults will occur in networks and it is often difficult to locate them, partly because the equipment is dispersed geographically and partly because equipment is often supplied by multiple vendors. There must be adequate facilities for diagnosing and recovering from faults.

- **Performance Monitoring:** As the usage of a network expands, it eventually overloads and users experience unacceptable delays or inadequate throughput. There must be adequate facilities for identifying where in the network the bottlenecks are occurring.

- **Security:** Recent events have graphically illustrated the dangers of unauthorised access to computer networks. There must be adequate facilities for identifying who is using a computer network and for controlling what facilities they have access to. In some cases, there are also auditing requirements for permanent records to be kept of who has done what on a network.

- **Accounting:** Where it is necessary to charge users for the costs of resources utilised (e.g. line charges) then there must be facilities to record the usage of the resources.

Data communications equipment must be designed to provide appropriate levels of the above facilities.

(iii) Software Reusability

A large number of the seeming-different data communications devices currently available actually share the same basic structure: a microprocessor inputs data according to one communications protocol, processes it (e.g. to perform multiplexing, data compression, encryption, etc.) and then outputs it, possibly using a different protocol. Thus although different devices will have different overall functions they may well share common hardware and software components.

The concept of hardware reusability is well understood, but there are also a number of techniques for software reusability. This concept was strongly emphasised in the work which resulted in the real-time programming language Ada, but the techniques may be applied in other contexts.

If it is intended to produce a range of data communications equipment at Bangalore, then software reusability techniques should be used in order to minimise duplication of effort.

As with the previous recommendation concerning Software Engineering, it is strongly recommended that urgent steps be taken to ensure that the MAEP team at Bangalore acquires expertise in the areas of Applications & Database, System Management and Software Reusability, and puts that expertise into use. Again, since it is highly likely that the other MAEP centres in India are encountering similar problems, particularly in the area of software reusability, this action should be coordinated nationwide.

Possible actions to acquire the necessary expertise would be the same as for the previous recommendation.

3.4 RECOMMENDATIONS - PLANNING AND BUSINESS STRATEGY

The planning of products for manufacture and sale must be done in the context of a realistic assessment of the commercial situation. In the discussions which I had with team members, I was not aware that such an assessment had been made for the current projects.

The following are examples of the type of activity which should be undertaken:

(a) *Establish the primary objectives:* These might include:

- Short term profits.
- Acquisition of experience, with a longer term expectation of profits.
- Import replacement, with consequent savings of hard currency.
- Development of a possible export industry.
- Dissemination of an 'enabling technology', with the aim of improving the efficiency of indian industry.

Of course, it may be possible to satisfy more than one objective at once, but should the objectives conflict, it is important to assess their relative priority.

(b) *Establish the Strengths and Weaknesses of the Organisation:* Before entering a competitive situation, it is important to assess realistically the strengths and weaknesses of the organisation relative to the competition. E.g.:

- Capabilities of design team, in terms of both size and expertise.

- Manufacturing costs.

- Access to capital (i.e. what sort of cash flows can be tolerated).

- Knowledge of local marketplace.

- Ability to respond quickly to new developments.

(c) *Establish market requirements:* This should be established in terms of 'what are the problems which the customer needs to have solved', rather than 'what products are required'.

Market requirements cannot be determined by sitting in a laboratory and thinking, nor can they be determined from overseas information since there is no guarantee that the problems facing indian organisations are identical to those facing overseas organisations.

Regular contact should therefore be established with:

- *a range of typical users.*

- *local dealers in data communications equipment.*

- *computer and telecommunications users associations.*

- *some of the larger suppliers of computer equipment* (since they will be aware of the applications used by their customers, but may not represent competition over communications equipment)

- *planners within the domestic and overseas telecommunications authorities* (for future developments in network services).

- *local representatives on international communications standards committees* (e.g. CCITT, ISO & IEEE).

(d) *Establish a market strategy:* This will largely follow from the points considered previously. Issues which should be considered include:

- Should the approach be to go into head-on competition with existing suppliers or to find a currently unsatisfied 'niche market'?

- Are there any special local requirements which are not satisfied by imported products; e.g. ability to handle multiple (human) languages and writing systems.

- Should the aim be to produce high-volume low-margin products, or low-volume high-margin products?

- Many end-users wish to deal with a single supplier for all needs (rather than having to deal with many suppliers). In this case, what can be done to fill the gaps in the product line? E.g. import or manufacture overseas products under licence.

To summarise, ***every effort should be made to establish an attitude of customer-orientation and market-orientation as part of the philosophy of the MAEP teams, at all levels of seniority.*** Only this approach can ensure that the right products are produced, i.e. those which meet users' needs more successfully than competitors' products. This is not something which engineers can just leave for the sales force to worry about; everybody must be involved (see PETERS & WATERMAN - references are given in Section 4).

Finally, ***it is strongly recommended that maximum use of data communications is made within the MAEP organisation itself,*** even though it may not be justifiable on pure economic grounds. The team members will gain a better appreciation of customers' needs through being users themselves.

4. REFERENCES

Books are no substitute for practical experience and guidance from those with practical experience; however, they can supply information when no other source is available. The following may be useful for the particular problems currently being encountered at Bangalore:

Data Communications General:

HALSALL, Fred,
'Data Communications, Computer Networks and OSI (Second Edition)',
Addison-Wesley, 1988.

A good overall introduction to the basic principles behind Data Communications.

TANENBAUM, Andrew S.,
'Computer Networks (Second Edition)',
Prentice-Hall International, 1988.

An excellent description of the principles of the more recent types of data communications network.

KNOWLES, T., LARMOUTH, J. & KNIGHTSON, K.G.,
'Standards for Open Systems Interconnection',
BSP Professional Books, 1987.

A detailed examination of the ISO standards for Open Systems Interconnection.

Data Communications Software:

Unfortunately, there is no single good book covering the whole of this topic. Useful information can be gleaned from the following:

BIGGERSTAFF, Ted J.,
'System Software Tools',
Prentice-Hall International, 1986.

A useful description of programming systems software for IBM-compatible PCs. Includes descriptions of simple communications software and of a basic multiprogramming system. Contains sample code in 'C'.

COMER, Douglas.,
'Operating System Design - The Xinu Approach',
Prentice-Hall International, 1984.

COMER, Douglas.,
'Operating System Design - Volume II - Internetworking with Xinu',
Prentice-Hall International, 1987.

A description of a model operating system (Xinu) together with a listing of the source code of Xinu in 'C'. Volume II includes software for a very simple local area network for Xinu. Rather over-simplified, but useful for those with limited experience of multiprogramming operating systems and how they are implemented.

LANE, Malcolm G.,
'Data Communication Software Design',
Boyd & Frazer Publishing Co (Boston), 1985.

A rather elementary description of data communications software.

PURSER, Michael.,
'Data Communications for Programmers',
Addison-Wesley, 1986.

Mainly a general introduction to data communications, but with one useful chapter on communications software.

<u>Data Communications Components:</u>

HELD, Gilbert.,
'Data Communications Networking Devices (Second Edition)',
Wiley, 1989.

A useful survey of available devices.

Proprietary catalogues containing useful data on the variety of available data communications components are issued by:

Black Box Corporation,
Mayview Rd at Park Drive, Box 12800, Pittsburg, PA 15241, USA.

Glasgal Communications Inc,
151 Veterans Dr, Northvale, NJ 07647, USA.

(This is not intended as a recommendation for the products of these companies, but just for the catalogues).

Journal:

'Data Communications', published by McGraw-Hill, USA. (monthly)

Software Engineering - General:

BROOKS, Frederick P.,
'The Mythical Man-Month - Essays on Software Engineering',
Addison Wesley, 1975.

The classic book on the subject, based on hard experience.

SOMMERVILLE, Ian.,
'Software Engineering (Second Edition)',
Addison-Wesley, 1985.

An excellent introduction to the basic principles of software engineering. A 3rd edition is believed to be imminent.

PRESSMAN, Roger S.,
'Software Engineering - A Practitioner's Approach (Second Edition)',
McGraw-Hill International, 1987.

More detailed coverage than Sommerville.

Journals:

'IEEE Transactions on Software Engineering', published by IEEE, USA (monthly).

'IEEE Software', published by IEEE Computer Society, USA (bimonthly).

'Software Engineering Notes', published by ACM Special Interest Group on Software Engineering, USA (approx quarterly).

Real Time Software Design:

ALLWORTH, S.T. and ZOBEL, R.N.,
'Introduction to Real-time Software Design (Second edition)',
Macmillan, 1987.

WARD, Paul T. and MELLOR, Stephen J.
'Structured Development for Real-time Systems
Volume 1: Introduction & Tools
Volume 2: Essential Modeling Techniques
Volume 3: Implementation Modeling Techniques'
Yourdon Press, 1985.

Business Strategy:

PETERS, Thomas J. and WATERMAN Robert H.
'In Search of Excellence - Lessons from America's Best-Run Companies',
Harper & Row, USA, 1982.

APPENDIX I

Itinerary: David Rowe - 25 February to 24 March 1989

Sat	25 Feb	Leave UK.
Sun	26 Feb	Arrive New Delhi.
Mon	27 Feb	Briefing with Mr M. Ramachandran, UNDP.
Tue	28 Feb	Briefing with Dr Krishna Kant, Dept of Electronics
Wed	1 Mar	Travel from New Delhi to Bangalore. Introduction to MAEP team.
Thu to Sat	2 Mar \ ! 11 Mar /	Briefing on individual projects. Technical discussions with project members. Preparation of conclusions for final report.
Mon to Sat	13 Mar \ ! 18 Mar /	Preparation and delivery of technical lectures on Data Communications Standards and the Implementation of Communications Software.
Mon	20 Mar	Travel from Bangalore to New Delhi.
Tue	21 Mar	Debriefing with Mr. N. Saratchandra Babu, Dept of of Electronics and Ms. V. Sukuntha, UNDP.
Wed	22 Mar	Travel New Delhi to Vienna.
Thu	23 Mar	Debriefing with Mr. V. Smirnov, UNIDO.
Fri	24 Mar	Return home.

APPENDIX II

List of Personnel with whom the Expert Met

UNDP New Delhi:

Mr. M. Ramachandran.	Senior Programme Officer.
Ms. V. Sukuntha.	UNDP Advisor.

Department of Electronics - Government of India - New Delhi:

Dr. Krishna Kant.	Chief Coordinator MAEP.
Mr. N. Saratchandra Babu.	Scientist/Engineer MAEP.

MAEP Southern Regional Centre - Bangalore:

Dr. A. Prabhakar.	Executive Director & Acting Project Coordinator.
Mr. Chidambara.	Senior Engineer.
Mr. M.V. Roopchandar.	Engineer Executive.
Mr. M.S. Mohan.	Engineer Executive.
Mr. K. Nalinakshan.	Assistant Executive Engineer.
Mr. K.J. Somashekar.	Assistant Executive Engineer.
Mr. Edward Rodrigues.	Assistant Executive Engineer.
Mr. K. Haridoss.	Senior Technical Assistant.
Ms. Kanaka Nayak.	Clerk-cum-typist.
Mr. R. Ravichandran	Clerk-cum-typist.
Mr. Kuppuswamy.	Helper.

UNIDO - Vienna:

Mr. Valery Smirnov.	Industrial Development Officer.

APPENDIX III

Lecture Series Outline.

Lecture 1:

Review of basic principles of communications protocols:
- Framing
- Addressing & multiplexing
- Error detection and recovery
- Flow control
- Message fragmentation & reassembly.

The need for layers of protocols. The principle of enveloping of protocols

A detailed examination of LAPB (HDLC ABM) as example of a low-level protocol.
An introduction to high-level (application) protocols.

Lecture 2:

The development of proprietary network architectures.

The differences between closed and open networks.

The ISO Reference Model for Open System Interconnection. The purposes of the 7 layers.

Service Specifications and Protocol Specifications.

The Transport Service.

Lecture 3:

The use of OSI Standards over Local Area Networks.

The 802.2 Specifications for Logical Link Control.

The need for a Network Layer for Internetworking. Connection-oriented and Connectionless Internetworking.

The Transport Service and Classes of Transport Protocol.

The relationships between Data Communications and Database Management.

Review of Session Layer Synchronization and its use in the secure updating of Databases.

The Application Layer facilities for Commitment, Concurrency and Recovery. Their use in Distributed Databases.

Lecture 4:

Communications software as a Real-Time problem. Making sure that the right thing is done at the right time.

Review of the principles of Process Scheduling in Operating Systems.
The role of:
- Application programs
- Supervisor Service Routines
- Interrupt Service Routines
- System Processes.

and the circumstances under which each is executed.

Process coordination: Mutual exclusion; synchronization between producers and consumers.
The use of Semaphores for Process Coordination.
Coordination between Interrupt Service Routines and Processes.

Lecture 5:

The Relationship between a Communications Protocol and the structure of the mechanisms used to implement it.
For each of the following time intervals:
- Bit level
- Byte level
- Frame level
- Logically related group of frames level

identifying:
- What work has to be done?
- What mechanism can be used to the work?

The need for queues to decouple:
- Physical I/O to and from the communications line
- Logical I/O to and from the application program.

Implementation of simple asynchronous/TTY-like communications.

Finite State Machines and their use for modeling problems.
A table-driven approach to implementing Finite State Machines.
Finite State Machines in the specification and implementation of communication protocols.

Implementation of a half-duplex, polling protocol; e.g. HDLC Normal Response Mode.

Implementation of a full-duplex, balanced protocol; e.g. LAPB.