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

FAIR USE POLICY

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

20916

Final report

on UNIDO Contract 90/128 Project No. DP/ALG/87/027

Activity code: J13208

"Computerized Production and Maintenance Management System"

for El Hadjar Metallurgical Combinate, Sider, Annaba, Algeria

SFW Software Anstalt

1993

TABLE OF CONTENTS

1. Introduction.

2. The phases of the subcontract.

- 2.1. Phase 1.
- 2.2. Phase 2.
- 2.3. Phase 3.
- 2.4. General remarks concerning courses 3.1 - 3.5.
- 2.5. Course 3.6 not delivered yet.

3. Financial overview of the subcontract.

- 3.1. Cost schedule of the original subcontract
- 3.2. Proposed financial arrangement

4. Concluding remarks

Appendix 1 (Copy of Phase 1 Report)

Appendix 2 (Terms of Reference Page 5)

Appendix 3 (Compte Rendu 11/10/90 Page 2)

1. Introduction.

The main purpose of our subcontract is to support the establishment of a computerized Production and Maintenance Management System, by transferring state of the art know-how in the application of local area networking in a real-time industrial environment, i.e. OS/400, OS/2 and DOS based equipments, connected by token ring.

2. The phases of the subcontract.

2.1. Phase 1.

Target:

Specification of additional software components required to operate the above network, and to carry out the development of CPMMS.

Status:

The report containing recommendations for the purchase of the additional software components have been made available in November 1990, and accepted in 1991 (Appendix 1).

2.2. Phase 2.

Target:

AS/400 training courses, covering the topics specified in Section 2/a of the Terms of Reference (Appendix 2).

Status:

This phase had been executed by ABIS, our subcontractor, residing in Leuven, Belgium. The courses delivered to SIDER are summarized in the following table:

Basic AS/400 course	2 days	Nov. 90
(additional course on request of SIDER for 30 persons)		
OS/400 System management	2 days	Nov. 90
Control language	4 days	Dec. 90
Database implementation	1.5 day	Dec. 90
Database and device file handling	4 days	Dec. 90
SQL	2 days	Dec. 90
PC support	2 days	Jan. 91
Phase 2 total	17.5 days	

According to the subcontract minimum 15 days of OS/400 training had to be delivered. The courses included all topics specified in the TOR, but has been packaged slightly differently, due to practical reasons (i.e. certain time and practice is required for a student to digest a course material before she/he can attend the next course).

Certain further details of the courses had been included in our first interim report. The fulfilment of Phase 2 has been accepted by UNIDO and the NC on 26/02/91.

2.3. Phase 3.

Target:

OS/2 training courses, covering the topics specified in Section 3/a of the Terms of Reference (Appendix 2). The detailed tematics of the six course to be delivered has been agreed in Annaba in October 1990 (Appendix 3).

Status:

Five courses has been delivered and accepted by the NC:

Course identifier:	**3.1.**
Course title:	Introduction to System Application Architecture (SAA)
Place of delivery:	Annaba
Time of delivery:	December 1990.
Instructor:	Dr. Istvan Gergely
Remarks:	Accepted by NC

Course identifier:	**3.2.**
Title:	OS/2 Operating System (General Concepts)
Place of delivery:	Annaba
Time of delivery:	December 1990.
Instructor:	Dr. Istvan Gergely
Remarks:	Accepted by NC

Course identifier:	**3.3.**
Title:	Practical C programming on IBM PC
Place of delivery:	Annaba
Time of delivery:	January 1991.
Instructor:	Dr. Peter Nagy
Remarks:	Acccepted by NC, but additional two days of practical training has been requested. This training has been completed by Dr. Istvan Gergely in May 1991 as part of Phase 4 (Consultation).

Course identifier:	3.4.
Title:	OS/2 Application Programming Interface (API)
Place of delivery:	Annaba
Time of delivery:	May 1991.
Instructor:	Dr. Istvan Gergely
Remarks:	Accepted by NC

Course identifier:	3.5
Title:	OS/2 Database Manager
Place of delivery:	Leuven, Belgium
Time of delivery:	December 1991.
Instructor:	Abis Staff
Remarks:	Accepted by NC

2.3.1. General remarks concerning courses 3.1 - 3.5.

The attendance of the courses varied from 8 (Database Manager) to 30 (OS/2 introductory courses).

Every course has been attended at least by 3-4 students, who succesfully understood the course material to be able to apply the content of the courses in practice, and are capable to participate in the development of the CPMMS system.

This is an internationally acceptable figure, and compares favourably to the number of participants planned in the TOR (4-6). Our instructors had been very satisfied with the motivation and participation of these students. It should be noted, that we are extremely impressed by the positive educational attitude experienced from our lady-students.

As indicated in previous notes, as well as during our personal meetings, the air travel to/from Annaba had been problematic, partially due to the unavailability tickets, as well as due to air traffic strikes.

Until the beginning of the Gulf War the project was basically running according to the original schedule. Afterwards, due to the change of circumstances problems had been experience in various fields related to the completion of the subcontract.

2.3.2. Course 3.6 not delivered.

After several date changes the OS/2 LAN Manager Course - originally scheduled for February 1991 in Annaba - has been agreed to be delivered in Budapest in January 1992 (with the kind support of IBM Hungary, providing equipment and software). Unfortunately, the NC had to cancel the trip in the last moment (very likely due to vis major). SFW made serious effort to arrange cancellation of reservations without payment of penalty, though all parties - including SFW - experienced serious financial losses due to the late cancellation of the course.

SFW suggested two replacement courses in Paris during Spring/Autumn 1992. In our meeting in Vienna in early 1992 the NC agreed that Sider will specify a new time schedule by 15 April; later this date had been postponed to summer 1992. We indicated five times in the last 6 month our willingness to discuss this matter, as soon as the NC is ready.

By 1993 IBM extended its activity in Alger providing locally certain courses and know-how, to be delivered for Sider under this subcontract. Therefore in mid-1993 in a tri-party meeting UNIDO, the NC and the subcontractor agreed to delete course 3.6 from Phase 3 of the subcontract (along with the appropriate reduction of the costs of Phase 3).

2.4. Phase 4.

Target:

Consultations

Status:

In mid-1993 during a tri-party meeting UNIDO, the NC and the subcontractor agreed to delete Phase 4 from the subcontract, due to the change of circumstances.

3. Financial overview of the subcontract.

3.1. Cost schedule of the original subcontract.

Phase 1	$20.000
Phase 2	$30.000
Phase 3	$36.000
Phase 4	$20.000
Subcontract total	$106.000

Phase 1 and Phase 2 has been completed and paid in 1991.

Following a tri-party meeting in early 1992 the subcontract has been amended and the cost schedule have been increased by $10.350. This has been paid in 1992.

No payment has been made toward Phase 3 and Phase 4 yet.

3.2 The financial arrangement agreed in 1993

Since courses 3.1 - 3.5 has been delivered in 1991, and course 3.6 as well as Phase 4 have been deleted from the subcontract, the original agreement has been amended as follows: SFW shall present a brief final report to UNIDO. Upon the acceptance of the final report UNIDO shall pay $30.000 for SFW. This is to reflect the fact, that from the 6 planned courses 5 courses have been succesfully completed.

4. Concluding remarks.

The coworkers of SIDER succeeded to gain an insight to OS/400, OS/2 and local area networks to serve as a proper base of their further activity. Considering the fact, that the SIDER configuration is the first AS/400 installation on the African continent, the staff of SIDER should be congratulated to their pioneering activity.

In fulfilling the subcontract, our coworkers have spent a couple of manmonths working together with the SIDER staff. The human relations between the coworkers of our companies ranged from good to excellent (with very few exceptions). This working atmosphere definitely contributed to the success of the subcontracted activity.

The attendance of the courses varied from large (8) to very large (30). The number of students, who succesfully understood the course material ranged from 2 to 5. This assures that the content of the courses can be applied in practice in the development of the CPMMS system. We were especially impressed by the positive attitude of our lady-students.

Certain problems have been detected during the fulfilment of the subcontract.

Keeping the schedules proved to be difficult; dates had to be modified in the last moment (or later), due to problems of vis major type (airline strikes, ticket problems, political unstability etc).

Furthermore, it should be noted, that the organization of software development projects of CPMMS size require special technologies. Due to the lack of the appropriate system design and project management skills a significant proportion of such projects fails or misses the target both in Europe and in the USA. It would be worthwile to consider to include such topics into the curriculum, or delegate the leading coworkers of the NC to such courses. Alternatevely, external specialists in system design and project management could be hired for certain periods.

C. THE SCOPE OF CONTRACTING SERVICES

Phase 1 : Selection of additional CPMMS software components

1a) Task description : To advise in selection of local area network software and database managment software for non dedicated servers operating under OS/2.

The network software should permit the following functions :

- OS/2 interprocess communications
- Industry standard multitasking
- File transfer to/from IBM AS/400
- Remote program execution
- Remote network monitoring
- Network activities scheduling
- User access control to server resources
- Coexisting of DOS and OS/2 applications on the network

The database management software for the non-dedicated server should have the following attributes :

- Concurent user access with file and record level locking.
- Access according to multiply key field.
- Programming interfaces from C-language.

Sophisticated database user interfaces on server level are not requested. Database volume is approx. 40 MB per server, divided into a great number of small data domains.

Furthermore, the Contractor should advise on :

- selection of program development tools in C under OS/2 and MSDOS

- Other OS/2 utilities (API) needed for server functions outlined in chapter A.

1b) Output : Specifications for additional software selection, containing justification of selected components, sources of purchase and estimated price.

1c) Implementation: Consultation with Project Chief Technical Adviser (CTA) and with the National Project Director (NPD) of CPMMS directly at SIDER or at Contractor's Home Office. Purchase of specified software components will be made separately by UNIDO and installed by CTA before commencing Phase-3.

In his proposal the Contractor should briefly outline his approach for SW selection.

Concernant la Phase-2, le programme de formation est le suivant

2.1 OS/400 Opérating System

a) Database Implémentation du 04 au 07.11.90
avec Introduction à l'AS/400

b) System Management du 10 au 11.11.90

c) Database and Device File Handling
du 12 au 14.11.90

d) Control Language du 17 au 21.11.90

Les cours a) et b) seront assurés par une personne et

les cours c) et d) par une autre personne.

2.2 PC.SUPPORT du 03 au 04.12.90

2.3 SQL/400 du 01 au 02.12.90

Toutes ces formations seront assurées par ABIS.

Mr GERGELY de SFW doit assurer une journée de pratique sur PC.SUPPORT le 05.12.90.

Concernant la Phase-3 le programme sera le suivant :

3.1 Introduction à S.A.A du 08 au 09.12.90

sera assuré par Mr GERGELY

3.2 OS/2 Operating System du 10 au 16.12.90

sera assuré par Mr GERGELY

3.3 Practical C programming on IBM PC
du 07 au 13.01.91

sera assuré par Mr PETER NAGY

3.4 Application Programming Interface
du 19 au 27.01.91

sera assuré par Mr GERGELY

3.6 LAN management du 16 AU 24.02.91

sera assuré par Mr GERGELY

3.5 OS/2 Data Management du 16 au 24.03.91

sera assuré par Mr GERGELY

Durant les périodes de formation de 3.5 et 3.6 , deux(02) jours de chaque période seront consacrés à la consultaion.

Selection of additional CPMMS software components

for the development of the

"Computerized Production and Maintenance Management System"

at SIDER, Annaba, Algeria

Proposal of SFW Software Anstalt in fulfillment of Phase 1 of

Terms of Reference (UNIDO Project DP/ALG/87/028 Contract 90/128)

Table of Content

1. Network Architecture and Functional Network Requirements 3

1.0 General considerations 3
1.1 Network architecture 4
1.2 Functional software requirements in LAN environment 6
1.2.1 AC01 Production control 6
1.2.2 LAC Production Control 12
1.2.3 Maintenance management 13
1.2.4 Production and Maintenance Management 13
1.2.5 System LAN 14

2. Software methods/tools to meet the demands of application programs 17

2.1 Communication tasks grouped by station types 17
2.2 Overview of the proposed tools. 19

3. Description of the software components proposed for CPMMS. 20

3.1 The OS/2 Extended Edition Version 1.2 20
3.2 OS/2 LAN Server 20
3.3 OS/2 Programming Tools and Information 21
3.4 OS/2 program development tools 21
3.5 DOS program development tools 21
3.6 Additional software for APPC support 22
3.7 Bridge Software for Token-Ring Local Area Networks 22
3.8 LAN Command for LAN management and error analysis 22
3.9 The Norton Guides for C language and OS/2 22

4. Software to be purchased 23

4.1 Software development department 23
4.2 Production departments 26

Figures 28

Appendix 1.

Network architecture and functional network requirements

Appendix 2.

Terms of Reference

1. Network Architecture and Functional Network Requirements

SIDER El-Hadjar Complex computer specialists and UNIDO's Chief Technical Adviser Mr. Attila Toth, during the visit of SFW Chief Executive Officer Dr. Peter J. Bruck and Team Leader Dr. Istvan Gergely in Annaba (09. 10. 1990 - 12. 10. 1990) proposed a hardware architecture for the UNIDO/SIDER network and compiled a list of functional requirements as an additional basis for specification of system software (see Appendix 1).

1.0 General considerations

The Terms of Reference (May 1990, see Appendix 2) gives a clear definition of the required functionality. We considered two alternatives, since both can meet all the requirements cited above:

- a Novell-based network, using OS/2 and Oracle as a database server, including Oracle's support toward dBASE on DOS workstations

- a solution based on OS/2 Extended Edition, including its own Database and LAN management components and DOS requester.

As a separate product, until 1990 the Novell/Oracle based solution definitely had significant advantages, both in performance and in user interface.

With the new (1.2) version of OS/2 the two alternatives came much closer: in the view of the recent (1990 September) reviews, the two alternatives for networking are approximately equivalent, both in performance and in capabilities.

Version 6. of the Oracle database management software in itself is better than the current release of OS/2 Database Manager; however, in an environment, where AS/400 and IBM 4381 also will be used, the high level of compatibility between the three IBM Database Management product in our view is of higher importance.

In general, OS/2 and its components are integral parts of a new, unified architecture of IBM (Systems Application Architecture - SAA), resulting in a high level logical consistency and portability. Therefore our current proposal is based on the IBM edition of the OS/2 operating system. IBM commited itself toward this direction, and considers OS/2 as the decisive factor in the next decade of the computer industry.

In this report we first analyze the software components to be purchased starting from the requirements, imposed by the functional specification, then describe a consistent set of software components to be purchased which can meet the above expectations.

1.1 Network architecture

SFW analyzed the hardware architecture of the SIDER network from the viewpoint of the SOFTWARE architecture, i. e.

- suitability to apply industry standard software components
- effectivity of operation (expectable performance)
- maintenance and management aspects
- software education/training aspects

Based on the above aspects SFW proposes to consider the following alternative solutions (changes) in the architecture of the network:

a.) The laboratory workstation (T6, Appendix 1) should be directly connected to the local area network (i.e. should be integral part of the LAN, and not connected as a remote station). Since the return of analysis data from the lab to ACO1 is time-critical, and the loss of information may result in costly damages, modem-based solutions seems to be unjustified.

The alternative hardware solution is to use in T5 the Asynchronous Connection Server Program (requiring in a dedicated DOS station) to connect the laboratory workstation via a pair of modems; this solution causes no change for the application programmer. (A third alternative solution, which requires significant additional investment on the application programming side, will be discussed briefly under F53 a.)).

If T5 runs the Asynchronous Connection Server Program, the same gateway station could be used for async connection toward D.O.I).

b.) The gateway station (T5) - designed to connect remote devices - is planned two have two functions: connect asynchronous workstations (located in the laboratory and in D.O.I.) as well as the IBM 4381.

For remote workstations the Asynchronous Connection Server Program is recommended. This software runs under DOS, and requires a dedicated station.

For the mainframe connection the appropriate interface board and the 3270 support of the OS/2 can be used, i.e. this function requires a different operating system then the async connection, therefore the two gateway functions should run in separate PCs. For the 3270 connection we propose to use Management Station (TSM).

With the above remarks SFW observed no contradiction between the proposed hardware architecture and the targets of the project.
Application of standard OS/2 and DOS components is possible using the proposed architecture. No obvious bottlenecks are indicated to cause performance degradation of the network. Segmentation contributes to the ease of maintenance and to the managability of the network. The planned training is adequate to introduce the SIDER team to the utilization of the main software components.

1.2 Functional software requirements in LAN environment

1.2.1 Domain: Production control. LAN: ACO1

F10: Local processing of dBASE databases on DOS workstations (T1)

The technique of the processing is unchanged in comparision with the standalone database management; however, the memory overhead due to the workstation components of the operating system should be taken into account. Therefore the use of an interpreter-based database management system in the production phase is not recommended; even if an interpreter will be used in the development phase, it is suggested, that in the production environment a compiled (e.g. Clipper) version should be used on the workstations, in interest of the reduction of its memory requirement.

It is recommended, that only the common subset of the interpreter and compiler version of the dBASE language is used during the development phase, and the final tests are made using the compiled version. We note, that the index file formats of certain dBASE compatible systems differ from that of dBASE itself (e.g. Foxbase uses multi-level indexes, while Clipper utilizes a dBASE-compatible format).

Furthermore, it will be necessary to read/write dBASE files from C programs, which have to use simultaneously various network resources as well. For C-callable dBASE file access the CodeBase packages of Sequiter Software is recommended.

F11: Communication between DOS workstations within the same domain (T1 - T1)

a.) Messages

Three simple mechanisms can be used to send messages betweeen workstations:

I. Using the OS/2 LAN Server command

net send destination-adress

The text of the message is read from the standard input (i.e. could be entered via the keyboard).

II. Alternatively, using redirection, the command input could be read from a file:

net send destination address < message-file

Limitations: messages are assumed to be text. The size of the message is limited by the buffer size at the receiving end

(default 4 Kb, max 64 Kb).

III. From programs, using the OS/2 NetMessageFileSend API; for a sample code listing please refer to Figure 2. This solution is more flexible, e.g. enables the specification of more than one destinations; size limitation: 64Kb text files.

This illustrates an important facility of OS/2: the functionality
of the command processor is available for programmers from high level languages, e.g. C in the form of Application Program Interfaces (APIs) calls.

b.) Transfer of files

I. Methods II. and III. above, used for message transfer, can be used for file transfer as well, with similar size limitations. In certain cases it is possible, however, to use workarounds, e.g. in case of dBASE databases, in a single step only certain number of records are converted into ASCII form and transferred to the other workstation (since in numerous applications only the changes after a given event contain relevant information).

II. An alternative solution (proposed by Microsoft) is the following:
The sending workstation tranfers the file to the server, using e.g. the
net copy workstation-filename server-filename
command, and sends a message to target workstation, that the file is available on the server, and could be copied to the target workstation. This copy operation can be initiated from the target workstation using another "net copy" command, before the file is to be accessed locally.

In this case the equivalent APIs are available as well for execution from programs.

III. The most flexible way of information exchange between DOS workstations is the Advanced Program-to-Program Communications (APPC), which is a common method between members of the IBM Systems Network Architecture (SNA): IBM PC (DOS-OS/2), AS/400 and IBM mainframes.
APPC defines a set of inter-program communication services permitting conversation between distributed processing programs. The program, running in the sending workstation can transmit data to the program, activated in the receiving workstation.

The flow of a sample OS/2 APPC program is explained on Figure 4: a file is transferred to another station, and after arrival it is printed; the source code is listed on Figure 5.

F12: Communication of DOS workstation with OS/2 server

a.) Messages

Same as under F11 a.) above

b.) Exchange of files

The OS/2 LAN Server command

net copy source-filename destination-filename

can be used to copy files between a workstation and the server.

c.) Record-level exchange

The most general method of transmitting and receiving records is the application of user-written transaction processing programs using APPC [see under F11 b.) III. above]; alternatively, a simpler solution is to copy the selected records to a temporary file, and transfer it from the workstation to the server (this is the only feasible solution in case of dBASE files, located on the workstation). For the other direction, the solution is even simpler: to access files/records on the server file transfer (or APPC) is not necessary, since the workstation can see the resources of the server as they were local to the workstation.

d.) Inquiries

Inquiry means the display of data (stored on the server) using the screen of the workstation. Server files are accessible as they were reside on the workstation.

- ISAM files
Records can be accessed using application programs (e.g. in Cobol) running under DOS in the workstation's memory

- dBASE files
Records can be accessed using dBASE application programs running under DOS in the workstation's memory; due to the larger memory, required by the workstation software itself, it is advisable to use compiled version of dBASE programs (this results in faster program loading times as well); alternatively, dBASE records can be accessed from C programs, using CodeBase calls.

- Database Manager files
Database Manager is the relational DBMS system of OS/2, which is compatible to AS/400 and 4381; therefore in case of data to be shared with the AS/400 this file format can be of importance. OS/2 databases can be accessed using the Remote Data Access (RDS) facility, which provides support for DOS as well as for OS/2 environments (Figure 1). Under DOS, an application program (e.g. in Cobol) can use (precompiled) high level SQL commands to access the remote databases on the OS/2 server. These programs are portable and can be used under OS/2 and OS/400 as well.

F13: Communication of DOS workstations with AS/400 server (T1-T4)

IBM - as part of the OS/400 operating system - delivers a PC software, called PC SUPPORT, to access the AS/400 from DOS and OS/2 workstations. The memory requirement of PC Support depends on the configuration but is relatively high (usually 200-300 Kbytes), therefore it should only be activated on the DOS workstations upon demand (IBM promised to reduce the memory requirement in the next - end of 1990 - release by automatic unloading of currently unnecessary functions of PC Support).

a.) File transfer/exchange

I. PC Support should be used to exchange native (EBCDIC) AS/400 files with (ASCII-oriented) DOS workstations; it is advisable to utilize the built-in character conversion facility of PC Support. Furthermore, it is very convenient to use PC

Support to convert and transfer - even selectively - records of AS/400 Database Manager databases to ASCII DOS files. (The other direction is more tedious, since the database and record/field description have to be specified for the conversion of an ASCII file to a relational database.)

II. The AS/400 can store PC type (ASCII) files as well; these can be accessed using PC Support as they were on the local drive of the workstation (resembling to the "virtual disk" facility of System/36).

III. As soon as an error correction of OS/2 will be released (promised for 1990), the OS/2 server can handle PC compatible AS/400 files as "remote volumes"; therefore these files will be accessible for all DOS workstations, as any other file on the OS/2 server, using only OS/2's DOS workstation software (50-80 Kb of code), without using PC Support.

Our AS/400 Special Consultant recommends, that for workstation access native OS/400 files or databases should be converted to OS/400 PC-TYPE files (residing on the same drive as the original files) by programs running in the AS/400, but initiated from the Workstation, and then these could be accessed similarly to local DOS files. In many cases this solution can replace the application of DOS-OS/400 APPC, which is a more general, but also more laborious solution for such problems.

b.) AS/400 database inquiries

Workstations on the network - using PC Support - can log into the AS/400 and access databases as they were local terminals.

F14: Communication between OS/2 Servers and AS/400 Server (T2-T4)

a.) File exchange

All the methods, listed under F13 a.) are available for the OS/2 server as well. File transfers to/from the OS/2 server can be initiated either from a workstation, or - if the server is not dedicated (i.e. available for user programs as well) - from the OS/2 server. Due to the multi-tasking capability of OS/2 - together with the lack of the 640 Kb memory limitation - it can be advantageous for certain applications to run PC Support on the server, as an OS/2 task.

b.) Record exchange

The general mechanism available for record exchange is APPC (Advanced Program-to-Program Communication): a transaction processing program, running in the requestor computer organizes a handshake with an appropriate program on the server side, posts a request for certain records, which will be retrieved and transferred to the requestor by the program running in the server.

APPC support is available from OS/2 Cobol (which is recommended for application programming). It should be noted, that the Cobol program has to take care of the ASCII-EBCDIC conversion as well.

In case of database manager files, the high level SQL query language can be used to retrieve the records to be transferred; SQL commands are available within Cobol under both operating system.

It is planned by IBM, that from 1993 OS/400 databases will be accessible from networked OS/2 systems, as they were local.

It should be noted, that the design, development and testing of OS/2 - OS/400 APPC is an advanced task, requiring certain experience. We suggest to acquire special documentation (for both operating systems) and sample programs (e. g. AdaptSNA/2 from Network Software Associates).

It is possible to use interim solutions until the APPC-based solution becomes available: e.g. initiate the execution of an AS/400 program which copies the relevant records into a file, then transfer this file to the OS/2 server.

F15: Communication between DOS Workstations and Laboratory Workstations (T1 - T6)

Wether the workstation in the laboratory is connected directly to the token ring, or via asynchronous server, for the application programmer this situation is identical to the case of F11.

F16: Communication between DOS workstations and workstation dedicated to charge calculation (T1 - T1C)

The solutions proposed for F11 /e.g. Method a.) II/ can be used; however, the problem can be solved using APPC as well. It is noted, that under DOS the charge

calculation station can service only one request simultaneously; if this is inadequate, the station should be configured as an OS/2 workstation.

F17: Communication between DOS workstations attached to different OS/2 servers (T1 - T2 - T2' - T1')

If the server stations are in different rings, the rings should be connected by bridge. The bridge is a dedicated PC running such software (e.g IBM Token Ring Bridge Version 2.2) which hides the difference: T1 can communicate with T1' as it were in the same ring.

a.) Record exchange
The exchange of record is possible using APPC. It should be noted, that before writing an APPC-based transaction program it is worthwhile to investigate the use the combination of file transfer and message exchange.

b.) Message exchange
Please refer to F11 a.).

1.2.2 Domain: LAC Production Control. LAN: LAC

F20 - F24 functions are equivalent to F10 - F24 under 1.2.1

F25: Communication between DOS workstation located at PPL and OS/2 server located in a different network domain (T3 - T2')

Supposing that the domain, containing the workstation and the one, containing the server are properly connected through a bridge, the solution is equivalent to the case of F12. For file transfer, a possible solution would be for larger files to copy via a temporary file on any of the server stations; for message exchange the use of the "net send" command is recommended.

F26: Communication between DOS workstation located at PPL and another DOS workstation located in a different network domain (T3-T1').

Supposing that the two domains are properly connected via bridge, the methods available for message exchange are the same as in the case of F11 (the simplest solution is the use of the "net send" command).

1.2.3 Domain: Maintenance management. LAN: ACO1 + LAC + PPL

F30: Local batch processing of plan of maintenance on OS/2 server station (T2) with printing on the AS/400 (T4).

Under OS/2 the PC Support can be used to access the AS/400; PC Support makes available all features of the AS/400 printer as it were a local printer dedicated to the station, where PC Support is running.

F31: Communication between DOS workstations and OS/2 server (T1 - T2)

The situation is equivalent with F12.

F32: Communication between DOS workstations within different domains (T1 - T1')

In case of properly configured bridge between the domains this case is equivalent to F11.

F33 - F34: Inquiries from DOS workstations to the AS/400 server (T1 - T4 and T3 - T4)

The situation is equivalent to F13 b.); the use of AS/400 PC Support is recommended.

F35: Inquiries from DOS workstations at PPL to the OS/2 server (T3 - T2)

This situation is equivalent to F12 d.)

1.2.4 Domain: Production and Maintenance Management. LAN: PPL

F40: Local processing on standard DOS workstations in PPL (T3)

(See under F10).

F41: Communication of DOS workstations with AS/400 server (T3-T4)

a.) AS/400 command execution

It is recommended to use the workstation as an AS/400 terminal applying PC Support (see also under F13).

b.) Inquiries (PC program - AS/400 program)

The standard method of program-to-program communication within the SNA architecture is the use of APPC; this facility is available also from high level languages, e.g. OS/2 Cobol.

F42: Communication between DOS workstations in different rings (T3 - T1).

Supposing proper connection between the rings using bridge(s) the problem is equivalent to F11; the use of the "net send" command is recommended.

F43: Communication between DOS workstation and OS/2 server (T3 - T2)

This situation is equivalent with the case of F12; the simplest method of file transfer is the use of the "net copy" command.

1.2.5 Domain: System LAN

F51: Communication between the Network Management Station (T5M) and any other station.

The Network Management Station is a station, running a program to support LAN management/administration under OS/2 EE. The "net send" command can be used to transfer messages to any other station in the network.

F52: Network monitoring from the Network Management Station (T5M)

Functionality required for effective network management, e.g.:
- error management
- configuration management
- performance management

- accounting management
- safety management

An advanced and economic solution to support the network management and error analysis is the LAN Command package of Dolphin Software. It is recommended to use color VGA display.

F53: Remote communication functions

a.) To/from the Laboratory

The "net send" command (or the equivalent API) can be used to send short text files to other workstations from the laboratory workstation. This should preferably be connected directly to the token ring; if the station is connected asynchronously, the Asynchronous Connection Server Program (running under DOS on T5) could be used; however, this solution has lower reliability. Since this connection is crucial for the usability of the ACO1 subsystem, we recommend the token-ring based connection.

To communicate with the laboratory workstation without token-ring or async server, it can be connected to the serial line of T5M, and the communication is programmed using the Asynchronous Communication Device Interface program (ACDI) of OS/2; T5M then should exchange the received information with the DOS workstations in ACO1 using OS/2 APPC and APPC/PC.

b.) To/from the IBM 4381 (3270 emulation)

OS/2 facilitates 3270 emulation, including printer and file transfer support. It should be noted, however, that the connection requires the presence of the appropriate interface hardware for the gateway, and the emulator function of OS/2 is only available on the OS/2 stations; each DOS workstation would require the addition of the IBM PC/3270 emulator software (Figure 3). Since SIDER was not yet in the position to provide detailed information on the 4381 hardware interface surface available upon the delivery of the mainframe, this point requires clarification at a later stage.

c.) Furthermore, it is foreseeable, that D.O.I shall require certain access to the network, and it is likely that some kind of connection will be desirable with and between the AS/400 and the 4381. However, we feel that the detailed hardware and software specification should be made separately, at a later stage, since this primarily affects mainframe/minicomputer hardware and software (the software support on the LAN side is already included in the current proposal).

F54: Controlling the AS/400 (T4)

Every DOS and OS/2 station, where PC Support is installed, can log into the AS/400 using terminal support. Furthermore, OS/2 Communication Manager contains the IBM 5250 Workstation Feature, available from OS/2 stations.

F55: File exchange between servers (T2, T2' and T4) via the Network Management Station (T5M).

The purpose of this function to enable the network administrator to "force" file transfer between servers. This task can be executed using the "net copy" command of the LAN Server.

2. Software methods/tools to meet the demands of application programs

In this chapter we wish to categorize the communication requirements, and arrive to a set of logically consistent software tools which can be used to satisfy the requirements.

2.1 Communication tasks grouped by station types

Communication requirements between:		Proposed methods:
DOS workstation - DOS workstation		
message exchange	(F11,F15,F17,F21, F26,F32,F42,F53)	"net send" command or NetMessage APIs
file exchange	(F11,F21,F32)	"net send" command for small ASCII files; "net copy" command via temporary server file; APPC
record exchange	(F15,F16,F17,F53)	APPC
DOS workstation - OS/2 workstation/server		
message exchange	(F12,F22,F25,F51)	"net send command or NetMessage APIs
file exchange	(F12,F22,F25,F31,F43)	"net send command for small ASCII files; "net copy command via temp. server file; APPC
record exchange	(F12,F22,F31)	APPC
inquiry	(F12,F22,F31,F35)	Cobol/dBASE programs; RDS for Database Manager files
OS/2 server - OS/2 server		
file exchange	(F55)	"net copy" command
DOS workstation - AS/400 server		
file exchange	(F13,F23)	"net copy" command
record exchange	(F41)	APPC
inquiry	(F13,F23,F33,F34)	PC Support
AS/400 command execution	(F41,F54)	PC Support

OS/2 workstation/server - AS/400 server

file exchange	(F14,F24,F55)	"net copy" command
record exchange	(F14,F24)	APPC
printing on AS/400	(F30)	PC Support; APPC

2.2 Overview of the proposed tools.

a.) OS/2 and LAN commands (e.g. "copy", "net send", "net copy")
b.) AS/400 PC Support
c.) Application programs (in Cobol, dBASE)
d.) Remote Data Services (RDS) of OS/2 Database Manager
e.) Advanced Program-to-Program Communication (APPC)
f.) OS/2 Application Programming Interfaces
(APIs, e.g. NetMessageFileSend)

Remarks:

1. The list reflects the increasing complexity of the tools. In most cases it is advantageous to use the simplest appropriate tool.

2. We propose a relatively small set of tools with broad applicability, to minimize the effort in system programming.

3. RDS and APPC is supported from Cobol under OS/2. The use of C language is recommended for API calls under OS/2 and DOS; for APPC/PC (under DOS) some assembly language background is required.

Alternative solutions:

1. If the AS/400 have to be accessed for inquiries also from user programs, instead of PC Support the Emulator High Level Language API (EHLLAPI) should be used; this contains 3270 and 5250 Workstation emulation (this software is integral part of OS/2; the required optional documentation is listed in Chapter 4).

2. For communication between DOS workstations instead of APPC the NETBIOS interface is also applicable. This software is already available at SIDER as part of the LAN Support Program; the documentation is in the IBM Local Area Network Technical Reference manual (see Chapter 4). From strategic viewpoint the main difference between APPC and NETBIOS is that APPC support is available on all SNA family members (including AS/400 and 4381), while NETBIOS is PC specific.

3. Description of the software components proposed for CPMMS.

3.1 The OS/2 Extended Edition Version 1.2

OS/2 is an advanced single-user multitasking operating system, featuring
- a graphical user interface (Presentation Manager), providing a multi-window environment to view and control multiple application programs and manipulate files
- a Communication Manager, to handle information exchange with workstations and hosts, supporting terminal emulations and asynchronous communication, AS/400 connection (5250 Workstation Feature), SNA gateway
- full-function relational database management system (Database Manager), compatible with larger IBM computers (e.g. AS/400), with Structured Query Language (SQL) support; access to data without programming (using the Query Manager); supports database management functions across LAN with the Remote Data Services; file/record locking in case of concurrent access.

An important characteristic of OS/2 is the accessibility of all operating system features for programmers, even from high level languages, using the Application Programming Interfaces (APIs).

3.2 OS/2 LAN Server

IBM's LAN Server (equivalent to Microsoft's LAN Manager) provides resource sharing across local area networks. Application program calls to access resources (files, printers, serial devices etc.) originating at workstations are redirected by the LAN Requester onto the network. These requests are received by the LAN Server, the required resource are provided, and the responses are passed back to the requesting application program as if the call were satisfied locally.

Main functions:

- resource sharing
- application program sharing (program runs in the workstation)
- remote program execution (program runs in the server)
- message service (message can be displayed, redirected or saved)
- extensive file transfer support
- LAN access control
- logging and statistics

3.3 OS/2 Programming Tools and Information

- Contains the Dialog Manager for simple development of text-oriented window applications, callable from Cobol, C etc.
- source code of sample programs to access important OS/2 services
- technical reference for OS/2 APIs

3.4 OS/2 program development tools

- The IBM C/2 compiler is recommended for system programming, especially when OS/2 calls (APIs) should be used; a significant part of OS/2 itself has also been written in C

- The IBM Cobol/2 compiler is recommended for programming non-computational applications e.g. information systems and transaction processing; Cobol/2 permits imbedded SQL statements and supports APPC calls for distributed processing applications.

- The Live Parsing Editor is a new IBM tool to facilitate the fast development of high level languages under OS/2.

3.5 DOS program development tools

- The Microsoft C Compiler (Version 6) is recommended for programming system-related tasks, especially when DOS APIs should be used. (This software can also be used installed under OS/2, and complements the IBM C/2 compiler, which is a modified version of this compiler).

- The IBM Cobol/2 compiler is recommended for programming non-computational applications e.g. information systems and transaction processing;

- IBM Macro/2 is the assembly language of the Intel 86 family processors

CodeBase of Sequiter Software is a dBASE file access facility, callable from C language. Multi-user, 85k dBASE compatible applications can be built, which are portable to OS/2. Source code is included, which enables SIDER to select only the minimum of the code required for its applications

- IBM's LAN Support Program contains drivers and NETBIOS support for PCs

- The Norton Editor is a simple to use but powerful screen editor to support program development.

3.6 Additional software for APPC support

APPC/PC

This is the DOS version of the Advanced Program-to-Program Communication software facility, required for the support of record exchange of DOS workstations with any other computer in the SIDER LAN (OS/2, AS/400 and 4381 supports APPC, since it is part of the Systems Network Architecture /SNA/).

AdaptSNA/2 APPC

This package - written by Network Software Associates (California) - implements IBM's Advanced Program-to-Program Communication for OS/2-OS/2, OS/2-DOS, OS/2-Mini and OS/2-Host applications, including an applications sub-system and a system tutorial package (ASSIST).

3.7 Bridge Software for Token-Ring Local Area Networks

This software runs in a standalone PC, which has two interface boards to connect two, otherwise separated rings.

3.8 LAN Command for LAN management and error analysis

This package - written by Dolphin Software Inc. - is one of the most advanced local area network management/diagnostic softwares available today. In its database store the description and the complete error history of each node. Monitoring functions generate real-time activity graphs and charts. Performance and diagnostic tools are available to test any segment of the network. When problems occur, recommendations are printed in textual form.

3.9 The Norton Guides for C language and OS/2

An excellent, indexed and cross-reference source of information; can be stored on the server, saving the purchase of multiple copies of handbooks.

4. Software to be purchased

4.1 Software development department

OS/2 based systems (2 computers)

Suggested configuration: iAPX 386 CPU, min 8MB memory
and 80 Mb hard disk
VGA color monitor suggested
Token Ring Card with LAN Support Program

DOS based systems (8 computers)

Suggested configuration: XT or AT with 640 KB memory
and 40 MB hard disk; DOS 3.3
Token Ring Card with LAN Support Program

Since the DOS based systems can access the OS/2 server, only a single copy of each application program should be purchased, since it can be loaded into the memory of the workstation from the shared disk of the server.

It is supposed, that DOS 3.3 is available on each workstation.

IBM Products

Please note, that IBM plans to release OS/2 V1.3 soon, to fix errors in V1.2. It is possible, that the French version of some manuals only will be available for V1.3; therefore we always specify one copy of the English version, followed or replaced by a French version (which may be delivered with some delay).

Part Number		Approx. price (USD)
45F0056	OS/2 Extended Edition V1.2 or higher (French)	1.400
45F0101	Upgrade of OS/2 Standard Edition V1.1 to Extended Edition V1.2 or higher (English) Price dependent on date of purchase; Proof of License (from V1.1 manual) is requested	?
45F0791	OS/2 LAN Server V1.2 (or higher) (English)	1.900
70F1821	Upgrade of OS/2 LAN Server V1.2 or higher English to French	400
45F0480	OS/2 Programming Tools und Info. V1.2 or higher	1.300
6280284	C/2 compiler V1.1 or higher	750
95X3071	Cobol/2 compiler	1.500
95X3068	Macro/2 assembler	350
44F6977	OS/2 Live parsing Editor V1.0 or higher	350
95X3075	APPC/PC V1.11 or higher	300
70F4131	IBM PC/3270 V1.01 or higher (French) One copy of the emulator - stored in the server - should not be used by more than one workstation, connected to IBM 4381 simultaneously (for copyright reasons)	500

Additional documentations required for software development:

01F0267 Database Manager Administrator's Guide
01F0269 Database Manager Programming Guide and Reference
01F0265 Database Manager SQL Reference
01F0261 Systems Administrator's Guide for Communications
01F0264 ACDI Programming Reference
01F0263 APPC Programming Reference
01F0266 EHLLAPI Programming Reference
01F0259 Programming Services and
Advanced Problem Determination for Communication
01F0260 Problem Determination Guide for Service Coordinator
01F0282 Commands Reference
01F0283 Procedures Language/REXX User's Guide
01F0284 Procedures Language/REXX Reference
75X1084 APPC/PC V1.11 Programming Guide
C218247 OS/2 EE APPC to AS/400 APPC Programming Example
G243466 OS/2 SE V1.2 Internals and Evaluation
G243550 OS/2 EE V1.2 Communications Manager Cookbook 1.
G243556 OS/2 EE V1.2 Communications Manager Cookbook 2.
C218162 AS/400 PC Support Tips Newsletter
G243255 AS/400 PC Support Redbook

Approximate total cost of additional documentation: 2.000

Source: IBM Austria
Obere Donaustrasse 95
A-1020 Wien, Austria
Contact person: Dipl. Ing. Peter Werzer
Tel: 43-222 21145 2771 Fax: 21145 3145

Alternative sources: Comicro AG
PC Division
Zypressenstrasse 76
CH-8004 Zurich, Switzerland
Contact person: Peter Erb PC product manager
Tel: 41-1 242 9855 Fax: 242 4062

IBM Algerie
6 Bd Mohammed V, Alger
Contact person: M. Dahmane
Tel: 213-2 630961 to 64 Telex: 67127 DZ

Masterbyte Computer
101 East Peddie
Newark, NJ 07114 USA
Contact person: Morton Green
Tel: 1-201 565 9855 Fax: 565 0984

It is possible to purchase all proposed software via Masterbyte (including Nantucket, Norton etc products).

Nantucket products

Clipper V5.0 dBASE language compiler 500

Source: ComputAbility

P.O. Box 17882
Milwaukee, WI 53217
Tel: 1-414 357 8181 Fax: 1-414 357 7814

Please note, that ComputAbility sells also Microsoft and Norton products (as well as hardware) at competitive prices

Microsoft products:

Microsoft C Compiler Version 6 350

Source: Microsoft Corporation
16011 NE 36th Way
P.O.Box 97017
Redmond, WA 98073-9717
Tel: 1-206 882-8080

Alternative source: ComputAbility
P.O. Box 17882
Milwaukee, WI 53217
Tel: 1-414 357 8181 Fax: 1-414 357 7814

Norton Products

The Norton Guides (C Language) 100
The Norton Guides (OS/2) 150
The Norton Editor 50

Source: Peter Norton Computing
2210 Wiltshire Boulevard
Suite 186
Santa Monica CA 90403
Tel: 1-213 319 2000

Alternative source: ComputAbility
P.O. Box 17882
Milwaukee, WI 53217
Tel: 1-414 357 8181 Fax: 1-414 357 7814

Sequiter products

CodeBase - the dBASE file access from C 300

Source: Sequiter Software Inc
P.O.Box 5659, Station L
Edmonton, Calgary
Canada T6C 4G1
Tel: 1-403-448 0313 Fax: 1-403 448-0315

Dolphin products

LAN Command 400

Source: Dolphin Software Inc.
6050 Peachtree Parkway
Suite 340-208
Norcross, Georgia 30092, USA
Tel: 1-404 339 7877

Network Software Associates Products

AdaptSNA/2 APPC **300**

Source: Network Software Associates
22982 Mill Creek
Laguna Hills, CA 92653 USA
Tel: 1-714 768 4013

4.2 Production departments

OS/2 based systems (3-6 computers)
T2 stations in ACO1 and LAC;
T5M network management station;

Suggested configuration: iAPX 386 CPU, min 8MB memory
and 80 Mb hard disk
VGA color monitor suggested
Token Ring Card (3Com TokenLink #3C603)

DOS based systems: 50 T1 and T3 computers in ACO1, LAC and PPL;
1 T5 Async Connection Server
1 T6 Laboratory workstation
1 T1C Computational station (with FPP)
3 B Bridge computer

Suggested configuration: XT or AT with 640 KB memory
and min 20 MB hard disk; DOS 3.3
Token Ring Card e.g. 3COM TokenLink
adapter #3C603 (this board is specified
until 55 Celsius !)

It is supposed, that MS-DOS 3.3 will be ordered with each workstation (significantly cheaper than separately).

IBM Products

Please note, that IBM plans to release OS/2 V1.3 soon, to fix errors in V1.2 (it is possible, that the French version of some manuals only will be available for V1.3). It is advisable to order the software only as soon as it is required for system integration. (It is possible that some part number should be reviewed as new releases become available).

Part Number		Number of copies	Approx. price (USD)
45F0161	Right to copy OS/2 Extended Edition V1.2	3 - 6	700
45F0792	OS/2 LAN Server V1.2 (or higher) French	3 - 6	1.900
or, ALTERNATIVELY:			
1 copy of OS/2 LAN Server V1.2 + Right to Copy:		2 - 5	?
96X5678	LAN Support Program V1.2 (French)	59 - 65	100
or, ALTERNATIVELY (in addittion to 10 copies in Development):			
Right to copy LAN Support Program V1.2		59 - 65	?

53F7724 Token Ring Network Bridge Program V2.2	3	2.500
or, ALTERNATIVELY		
1 copy of Bridge Program + Right to Copy	2	?

70F4131 IBM PC/3270 V1.01 or higher (French) 500
One copy of the emulator - stored in the server - should not be used by more than one workstation, connected to the IBM 4381 simultaneously (for copyright reasons);

*

It is country-dependent, which products are sold also with "Right to Copy" allowance; from the above products, in Austria generally OS/2 itself and PC/3270 can be copied with permission. It is suggested, that UNIDO officially request this right for the above products from IBM's International Licensing Officer, who is in charge of UN purchases. He is authorized to determine, which IBM office should deliver OS/2 software for the UNIDO/Sider project.
Please contact: M. Jan Ferez, IBM Suisse, 48 av G. Motta, P.O.Box 1211 Geneve 2 Switzerland; tel: 41-22 791 5555 fax: 791 5200.
If the Right to Copy is not granted, the indicated number of copies should be ordered.

Source: IBM Austria
Obere Donaustrasse 95
A-1020 Wien, Austria
Contact person: Dipl. Ing. Peter Werzer
Tel: 43-222 21145 2771 Fax: 21145 3145

Alternative sources: Comicro AG
PC Division
Zypressenstrasse 76
CH-8004 Zurich, Switzerland
Contact person: Peter Erb PC product manager
Tel: 41-1 242 9855 Fax: 242 4062

IBM Algerie
6 Bd Mohammed V, Alger
Contact person: M. Dahmane
Tel: 213-2 630961 to 64 Telex: 67127 DZ

Masterbyte Computer
101 East Peddie
Newark, NJ 07114 USA
Contact person: Morton Green
Tel: 1-201 565 9855 Fax: 565 0984

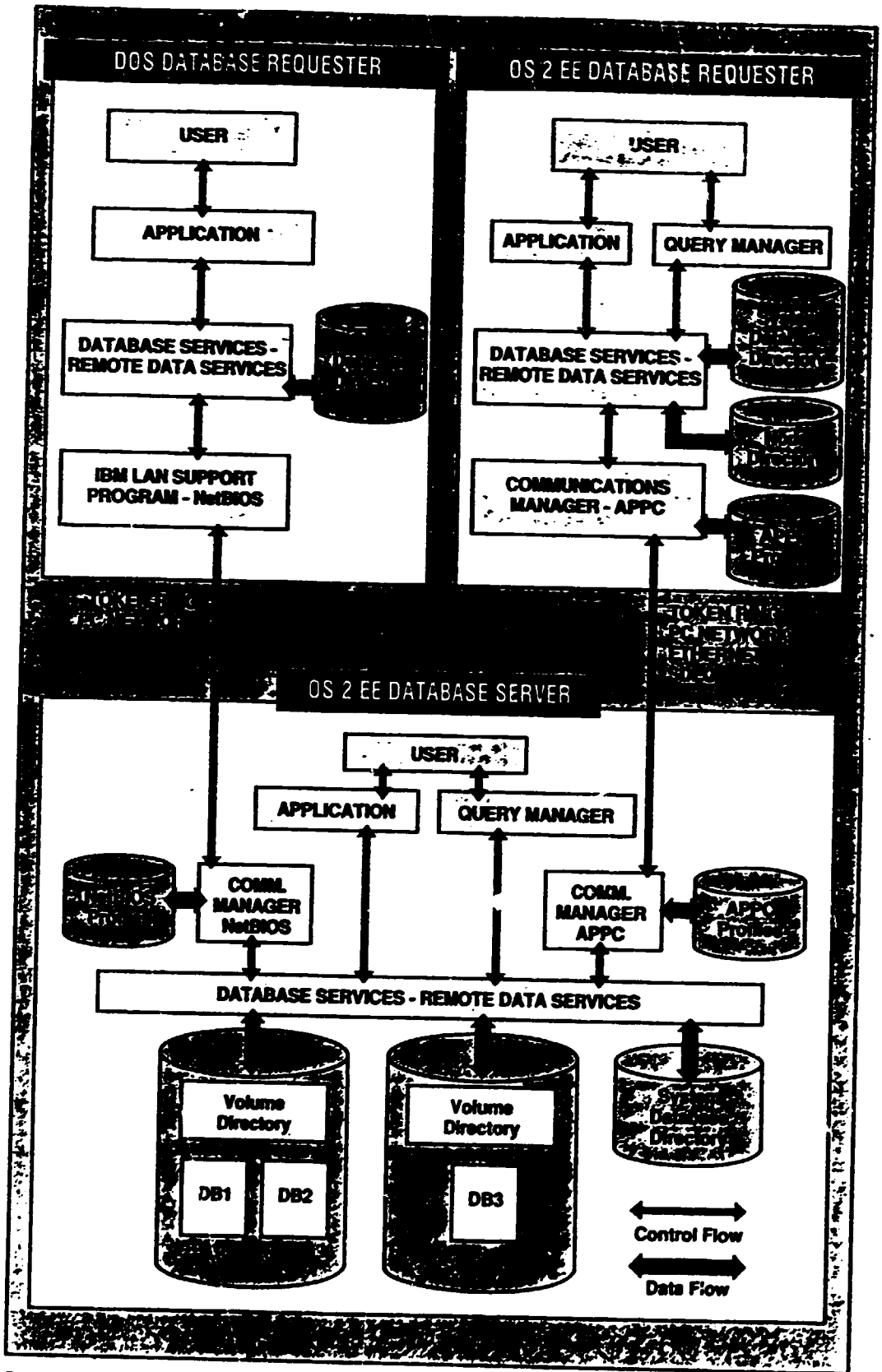


Figure 1. Three Workstation Types

```
/*
 * MSGFILE.C -- This program uses the NetMessageFileSend API
 * to send a copy of the specified file to each name in a
 * list of network users.  Each message name is converted to
 * uppercase for compatibility with the LAN Manager interface.
 *
 * Compile with:  C> cl msgfile.c netoem.lib
 *
 * Usage:  C> msgfile username(s) filename
 *
 */

#include <stdio.h>
#include <string.h>
#include <netcons.h>
#include <message.h>

main(argc, argv)
int argc;
char **argv;
    {
    unsigned len, err;
    int i, exitcode;

    if (argc < 3)
        {
        printf("syntax: msgfile username(s) filename\n");
        exit(-1);
        }

    exitcode = 0;
    printf("%-20s%-20s\n", "NAME", "STATUS");
    printf("%-20s%-20s\n", "----", "------");
    for (i = 1; i < argc-1; ++i)
        {
        strupr(argv[i]);
        err = NetMessageFileSend(NULL, argv[i], argv[argc-1]);
        if (err == 0)
            printf("%-20sOK\n", argv[i]);
        else
            {
            printf("%-20s%d\n", argv[i], err);
            ++exitcode;
            }
        }
    exit(exitcode);
    }
```

Figure 2.
The MSGFILE.C program.

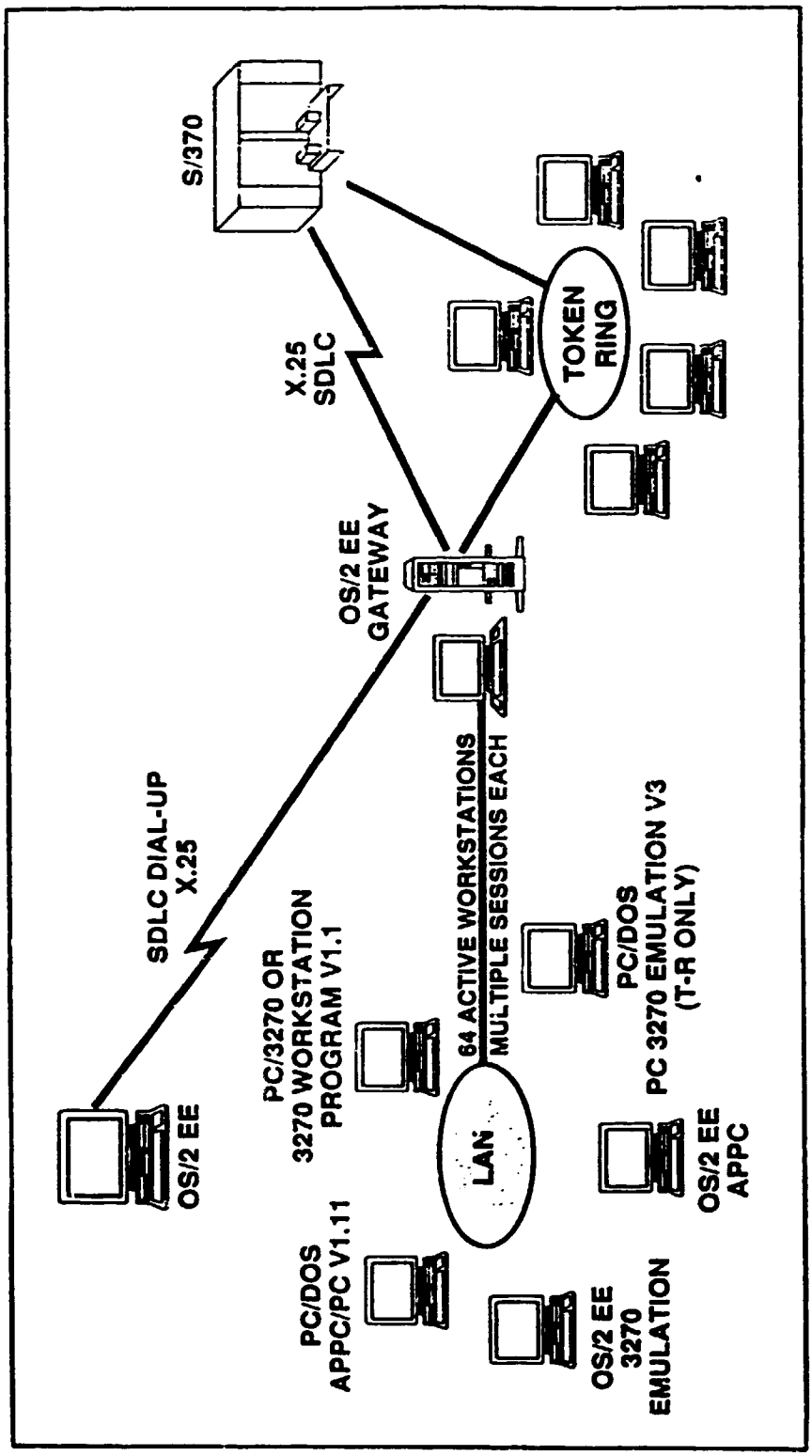


Figure 3. SNA Gateway Provides Session Pooling for More Than 64 Active Workstations

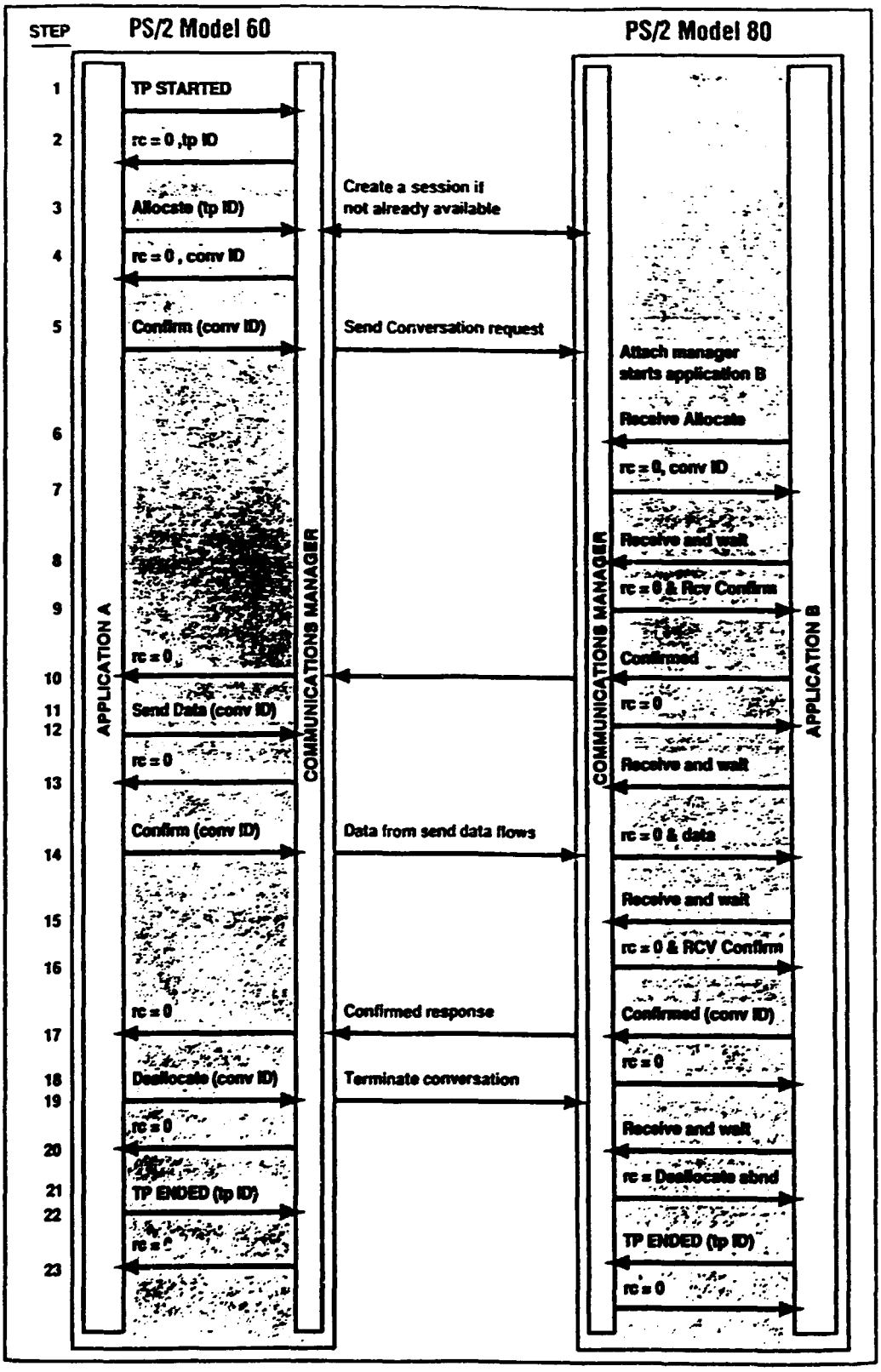


Figure 4. Steps to Communicate with a Remote Application (Continued)

Step	Description
1 and 2	Call APPC to start a Transaction program. This will return a TP ID which all verbs must contain when sending verbs to this TP.
3 and 4	Application A requests a conversation with the remote application B. A conversation ID is returned but the actual request does not flow. The logical record length has not been exceeded.
5	Application A issues a confirm to both force the conversation request to flow to the partner and wait for a verification that the remote application has started. The Attach Manager on the PS/2 Model 80 starts Application B. Application B now issues APPC verbs.
6 and 7	Application B issues the RECEIVE ALLOCATE verb which will create the conversation.
8 and 9	Application B issues a receive command waiting for data or status indicators. A RECEIVE CONFIRM status is received which tells Application B that a confirm must be issued.
10 and 11	Application B issues the confirm and Application A gets the successful completion of the confirm verb.
12 and 13	Application B issues another RECEIVE AND WAIT, waiting for data or more status information. Application A sends a data buffer.
14	Application A issues a CONFIRM to force the data buffer to flow to Application B and also wait for a confirmation from Application B. Application B receives the data that was sent.
15 and 16	Application B issues another RECEIVE AND WAIT and receives the RECEIVE CONFIRM indicating that Application A is waiting for confirmation.
17 and 18	Application B issues the CONFIRM and Application A gets a successful completion of the confirm verb.
19 and 20	Application B issues another RECEIVE AND WAIT. Application A issues a DEALLOCATE to terminate the conversation with Application B.
21	Application B gets an unsuccessful completion of the RECEIVE AND WAIT verb. The return code indicates the conversation has been terminated by a DEALLOCATE verb from Application A. Application A terminates the TP by issuing a TP ENDED verb. This closes the TP ID.
22 and 23	Application B also terminates its TP ID by sending the TP ENDED verb.

Figure 4. Steps to Communicate with a Remote Application

```
/*****************************************************************/
/*    This routine shows how to use APPC to start an application on   */
/*    a Partner LU and then transfer some data to that Partner.       */
/*    The following APPC verbs are used :                             */
/*                                                                    */
/*        TP_STARTED, ALLOCATE, CONFIRM, CONVERT                      */
/*        SEND DATA, DEALLOCATE, TP ENDED                             */
/*                                                                    */
/*    The code example assumes the partner LU has been configured     */
/*    with a remote TP.  This TP has been configured to accept        */
/*    mapped conversations and a sync level that accepts confirms.    */
/*                                                                    */
/*    The following string is sent to the partner LU :                */
/*                                                                    */
/*        "PC6::C:\REPORTS\APRIL10.TXT"                               */
/*                                                                    */
/*    This string instructs the remote application to get the file    */
/*                                                                    */
/*        "C:\REPORTS\APRIL10.TXT"                                    */
/*                                                                    */
/*    from another partner LU named PC6 and print that file.          */
/*****************************************************************/

#include <dos.h>
#include <memory.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "appc_c."
#include "acssvc.h"
#include <doscalls.h>

union verb_blocks {
       struct tp_started tp_started_block;
       struct allocate allocate_block;
       struct send_data senddata_block;
       struct confirm  confirm_block;
       struct tp_ended tp_ended_block;
       struct deallocate deallocate_block;
};
typedef union verb_blocks verb_control_blocks;

struct convert far *I_CONVERT(
      struct convert far *call_block,
      char direction,
      char char_set,
      unsigned short length,
      char far *src_ptr,
      char far *dest_ptr);

#define  tp_id_size       8
#define  tp_name_size    64
#define  userid_size     10
#define  password_size   10
#define  mode_name_size   8
#define  lul_name_size    8

char far *lu_alias = "PC1";
char far *plu_alias = "PC2";
char far *remote_tp = "TPSERVER";

/*****************************************************************/
/*                                                               */
/*                          MAIN ROUTINE                         */
/*                                                               */
/*****************************************************************/

void main(argument_count,arguments,envp)
int argument_count;
char *arguments[];
```

Figure 5. Sample Code for Application A in Figure 4 (Continued)

```
char *envp[];
{
  verb_control_blocks far *control_block = 0L;
  char tp_id[tp_id_size];
  unsigned long conv_id;
  struct convert far *convert_buffer = 0L;
  char far * data_ptr = 0L;
  char *print_report = "PC6::c:\\reports\\april10.txt";

  if (argument_count > 1) lu_alias = arguments[1];
  if (argument_count > 2) plu_alias = arguments[2];
  if (argument_count > 3) remote_tp = arguments[3];
  printf("\x0A\x0D Local LU: %s  Partner LU: %s Remote TP: %s\x0A\0xD",
          lu_alias, plu_alias,remote_tp);

/*********************************************************************/
/* Get shared memory segment for verb control blocks and data buffers */
/*********************************************************************/

  DOSALLOCSEG(512,&FP_SEG(control_block),1);
  DOSALLOCSEG(sizeof(struct convert),&FP_SEG(convert_buffer),1);
  DOSALLOCSEG(512,&FP_SEG(data_ptr),1);

/*********************************************************************/
/*    Start a Transaction Program (TP) using the TP_STARTED verb     */
/*                                                                   */
/*    First build the required information for a tp_started verb     */
/*                                                                   */
/*                      Set the tp started opcode                    */
/*********************************************************************/

  control_block->tp_started_block.opcode = AP_TP_STARTED;

/*********************************************************************/
/*                    Set the name to call this TP                   */
/*           Initialized to blanks; then fill in the TP name         */
/*********************************************************************/

  memset(control_block->tp_started_block.tp_name,' ',64);
  memcpy(control_block->tp_started_block.tp_name,"TEST TP",6);

/*********************************************************************/
/*       Set the local LU name with which to associate this TP       */
/*      Ensure the lu alias field contains only ASCII characters     */
/*********************************************************************/

  memset(control_block->tp_started_block.lu_alias,' ',8);
  memcpy(control_block->tp_started_block.lu_alias, lu_alias,strlen(lu_alias));

/*********************************************************************/
/*                  Call the APPC system to start a TP               */
/*********************************************************************/

  APPC_C((long)control_block);

/*********************************************************************/
/*                             Check for errors                      */
/*********************************************************************/

  if (control_block->tp_started_block.primary_rc) {
     printf("The TP_STARTED verb failed.  Primary RC: %X Secondary rc: %lX",
              control_block->tp_started_block.primary_rc,
              control_block->tp_started_block.secondary_rc);
     return;
  }

/*********************************************************************/
/*                             Save the tp id                        */
/*********************************************************************/
  memcpy(tp_id,control_block->tp_started_block.tp_id.tp_id_size);
```

Figure 5. Sample Code for Application A in Figure 4 (Continued)

```
/*****************************************************************/
/*          Allocate a conversation with a Remote application    */
/*****************************************************************/

/*****************************************************************/
/*    First initialize all required EBCDIC fields to EBCDIC blanks;*/
/*****************************************************************/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  memset(control_block->allocate_block.tp_name,0x40,tp_name_size);
  memset(control_block->allocate_block.plu_alias,' ',lui_name_size);
  memset(control_block->allocate_block.mode_name,0x40,mode_name_size);
  memset(control_block->allocate_block.user_id,0x40,userid_size);
  memset(control_block->allocate_block.pwd,0x40,password_size);

/*****************************************************************/
/*       Initialize a convert verb buffer to convert the mode name */
/*       to EBCDIC.  Place the converted name in the allocate verb */
/*       buffer.                                                   */
/*****************************************************************/

  I_CONVERT(convert_buffer,SV_ASCII_TO_EBCDIC,SV_A,5,
                       "MODE1",
                       control_block->allocate_block.mode_name);

/*****************************************************************/
/*     Call the Service verb to convert the Mode name to EBCDIC  */
/*****************************************************************/

  ACSSVC_C((long)convert_buffer);

/*****************************************************************/
/*                        Check for errors                       */
/*****************************************************************/

  if (convert_buffer->primary_rc) {
     printf("Converting Mode name failed.  Primary RC: %X Secondary rc: %lX",
              convert_buffer->primary_rc,
              convert_buffer->secondary_rc);
     return;
  }

/*****************************************************************/
/*      Initialize a convert verb buffer to convert the remote TP */
/*      to EBCDIC.  Place the converted name in the allocate verb */
/*      buffer.                                                   */
/*****************************************************************/

  I_CONVERT(convert_buffer,SV_ASCII_TO_EBCDIC,SV_A,strlen(remote_tp),
                       remote_tp,
                       control_block->allocate_block.tp_name);

/*****************************************************************/
/*      Call the Service verb to convert the TP name to EBCDIC   */
/*****************************************************************/

  ACSSVC_C((long)convert_buffer);

/*****************************************************************/
/*                          Check for errors                     */
/*****************************************************************/

  if (convert_buffer->primary_rc) {
     printf("Converting Remote TP name failed.  Primary RC: %X Secondary rc: %lX",
              convert_buffer->primary_rc,
              convert_buffer->secondary_rc);
     return;
  }
```

Figure 5. Sample Code for Application A in Figure 4 (Continued)

```
/*-------------------------------------------------------------------*/
/*          Finish initializing the other required fields            */
/*-------------------------------------------------------------------*/

  control_block->allocate_block.opcode = AP_M_ALLOCATE;
  control_block->allocate_block.opext = AP_MAPPED_CONVERSATION;

/*-------------------------------------------------------------------*/
/*     Move the tp id returned from the tp_started verb into the     */
/*     the allocate verb                                             */
/*-------------------------------------------------------------------*/

  memcpy(control_block->allocate_block.tp_id,tp_id,tp_id_size);

/*-------------------------------------------------------------------*/
/*                      Set the remote TP name                       */
/*-------------------------------------------------------------------*/

  memcpy(control_block->allocate_block.plu_alias,plu_alias,strlen(plu_alias));

/*-------------------------------------------------------------------*/
/*      The flag AP_WHEN_SESSION_ALLOCATED will cause APPC to        */
/*      return control to this task only after a session has         */
/*      been created with the remote partner                         */
/*                                                                   */
/*      Use the flag AP_WHEN_SESSION_FREE to return control quicker  */
/*      with return codes that indicate whether sessions are         */
/*      available or not.                                            */
/*                                                                   */
/*      NOTE: conversations are created on top of a session          */
/*-------------------------------------------------------------------*/

  control_block->allocate_block.rtn_ctl = AP_WHEN_SESSION_ALLOCATED;
  control_block->allocate_block.sync_level = AP_CONFIRM_SYNC_LEVEL;

/*-------------------------------------------------------------------*/
/*                      Send the allocate request                    */
/*-------------------------------------------------------------------*/

  APPC_C((long)control_block);

/*-------------------------------------------------------------------*/
/*                           Check for errors                        */
/*-------------------------------------------------------------------*/

  if (control_block->allocate_block.primary_rc) {
     printf("Allocate failed.  Primary RC: %X Secondary rc: %lX",
              control_block->allocate_block.primary_rc,
              control_block->allocate_block.secondary_rc);
     return;
  }

/*-------------------------------------------------------------------*/
/*                      Save the conversation ID                     */
/*-------------------------------------------------------------------*/

  conv_id = control_block->allocate_block.conv_id;

/*-------------------------------------------------------------------*/
/*    Send a confirm verb to verify that the Remote TP has started   */
/*                                                                   */
/*                   Build the confirm verb buffer                   */
/*-------------------------------------------------------------------*/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  control_block->confirm_block.opcode = AP_M_CONFIRM;
  control_block->confirm_block.opext = AP_MAPPED_CONVERSATION;
  memcpy(control_block->confirm_block.tp_id,tp_id,tp_id_size);
  control_block->confirm_block.conv_id = conv_id;
```

Figure 5. Sample Code for Application A in Figure 4 (Continued)

```
/*********************************************************************/
/*                        Send the confirm verb                      */
/*********************************************************************/

  APPC_C((long)control_block);

/*********************************************************************/
/*                          Check for errors                         */
/*********************************************************************/

  if (control_block->confirm_block.primary_rc) {
     printf("CONFIRM  failed.  Primary RC: %X Secondary rc: %lX",
              control_block->confirm_block.primary_rc,
              control_block->confirm_block.secondary_rc);
     return;
  }

/*********************************************************************/
/*          Send the pathname of the file to be printed to the       */
/*          remote application                                       */
/*                                                                   */
/*                   Build the send data verb buffer                 */
/*********************************************************************/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  control_block->senddata_block.opcode = AP_M_SEND_DATA;
  control_block->senddata_block.opext = AP_MAPPED_CONVERSATION;
  memcpy(control_block->senddata_block.tp_id,tp_id,tp_id_size);
  control_block->senddata_block.conv_id = conv_id;
  control_block->senddata_block.dlen = strlen(print_report);
  control_block->senddata_block.dptr = data_ptr;
  memcpy(data_ptr,print_report,strlen(print_report));

/*********************************************************************/
/*                       Issue the Send Data verb                    */
/*********************************************************************/

  APPC_C((long)control_block);

/*********************************************************************/
/*                          Check for errors                         */
/*********************************************************************/

  if (control_block->senddata_block.primary_rc) {
     printf("Send Data failed.  Primary RC: %X Secondary rc: %lX",
              control_block->senddata_block.primary_rc,
              control_block->senddata_block.secondary_rc);
     return;
  }

/*********************************************************************/
/*        Send a confirm verb to verify that the Remote TP has       */
/*        received the print request                                 */
/*********************************************************************/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  control_block->confirm_block.opcode = AP_M_CONFIRM;
  control_block->confirm_block.opext = AP_MAPPED_CONVERSATION;
  memcpy(control_block->confirm_block.tp_id,tp_id,tp_id_size);
  control_block->confirm_block.conv_id = conv_id;

/*********************************************************************/
/*                        Send the confirm verb                      */
/*********************************************************************/

  APPC_C((long)control_block);
```

Figure 5. Sample Code for Application A in Figure 4 (Continued)

```
/*****************************************************************/
/*                          Check for errors                     */
/*****************************************************************/

  if (control_block->confirm_block.primary_rc) {
     printf("Confirm of Print request failed.  Primary RC: %X Secondary rc: %lX",
              control_block->confirm_block.primary_rc,
              control_block->confirm_block.secondary_rc);
     return;
  }

/*****************************************************************/
/*          Deallocate the conversation with the Partner         */
/*****************************************************************/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  control_block->deallocate_block.opcode = AP_M_DEALLOCATE;
  control_block->deallocate_block.opext = AP_MAPPED_CONVERSATION;
  memcpy(control_block->deallocate_block.tp_id,tp_id,tp_id_size);
  control_block->deallocate_block.conv_id = conv_id;
  control_block->deallocate_block.dealloc_type = AP_FLUSH;

/*****************************************************************/
/*                     Send the deallocate verb                  */
/*****************************************************************/

  APPC_C((long)control_block);

/*****************************************************************/
/*                 Terminate the Transaction Program             */
/*****************************************************************/

  memset((char *)control_block,'\0',sizeof(verb_control_blocks));
  control_block->tp_ended_block.opcode = AP_TP_ENDED;
  memcpy(control_block->tp_ended_block.tp_id,tp_id,tp_id_size);

  APPC_C((long)control_block);

}

struct convert far *I_CONVERT(call_block,direction,char_set,length,
                              src_ptr,dest_ptr)
struct convert far *call_block;
char direction;
char char_set;
unsigned short length;
char far *src_ptr;
char far *dest_ptr;
{
  if (call_block == 0) return(0);
  call_block->opcode = SV_CONVERT;
  call_block->direction = direction;
  call_block->char_set = char_set;
  call_block->len = length;
  call_block->source = src_ptr;
  call_block->target = dest_ptr;
 return(call_block);
}
```

Figure 5. Sample Code for Application A in Figure 4

FUNCTIONAL NETWORK REQUIREMENTS

Domain: Production control LAN: AC01

Typ	Function	Volune/Freq
F10	Local processing on each station type T1 Local DB: DBASE III Plus	N/A
F11	Peer-to-peer communication between T1 - T1 a) Messages b) File transfer	 200 B/min 10 KB/min
F12	T1-T2 communication a) messages b) exchange of files c) record-level exchange d) inquires	 300 B/hour 100 KB/30 min 500 / hour Random
F13	T1 -) T4 communication a) File transfer / exchange b) AS/400 DB inquires	 1 MB/8 hours 20 Q/hour
F14	T2 -) T4 communication a) file exchange b) record exchange	 20 KB/hour 60 KB/hour
F15	T1 - T6 communication bidirectional (laboratory) a) send block to T6 b) receive block from T6	 200 B/min 200 B/min
F16	T1 - T1C program-to-program communications (charge calculations)	2x10x500B/10min
F17	T1 - T2 - T2- T1 communication (transport of brames) a) record exchange b) message exchange	 50 x 1 KB/hour 5 x 100 B/hour

Domain: LAC Production control LAN: LAC

Typ	Function	Volume/Freq
F20	= F10	
F21	= F11	
F22	= F12 as AC01 functions	
F23	= F13	
F24	= F14	
F25	T3 - T2 communication a) file fransfer (b) message exchange)	 1x2 BM/8 hour 10 x 500 B/hour
F26	T3 - T1 message exchange	10 x 500 B/hour

Domain: Maintenance management LAN: AC01+LAC+PP2

Typ	Function	Volume/Freq
F30	Local processing (batch) on T2 (plan of maintenance) with printing on T4	1 x /Week
F31	T1 - T2 communication a) fill transfer b) record exchange c) inquires	 2 MB/hour 300x500B/hour 300x500B/hour
F32	T1 - T1 communication a) messages b) files	 random random
F33	T1 to T4 inquires	random
F34	T3 to T4 inquires	random
F35	T3 to T2 inquires	random

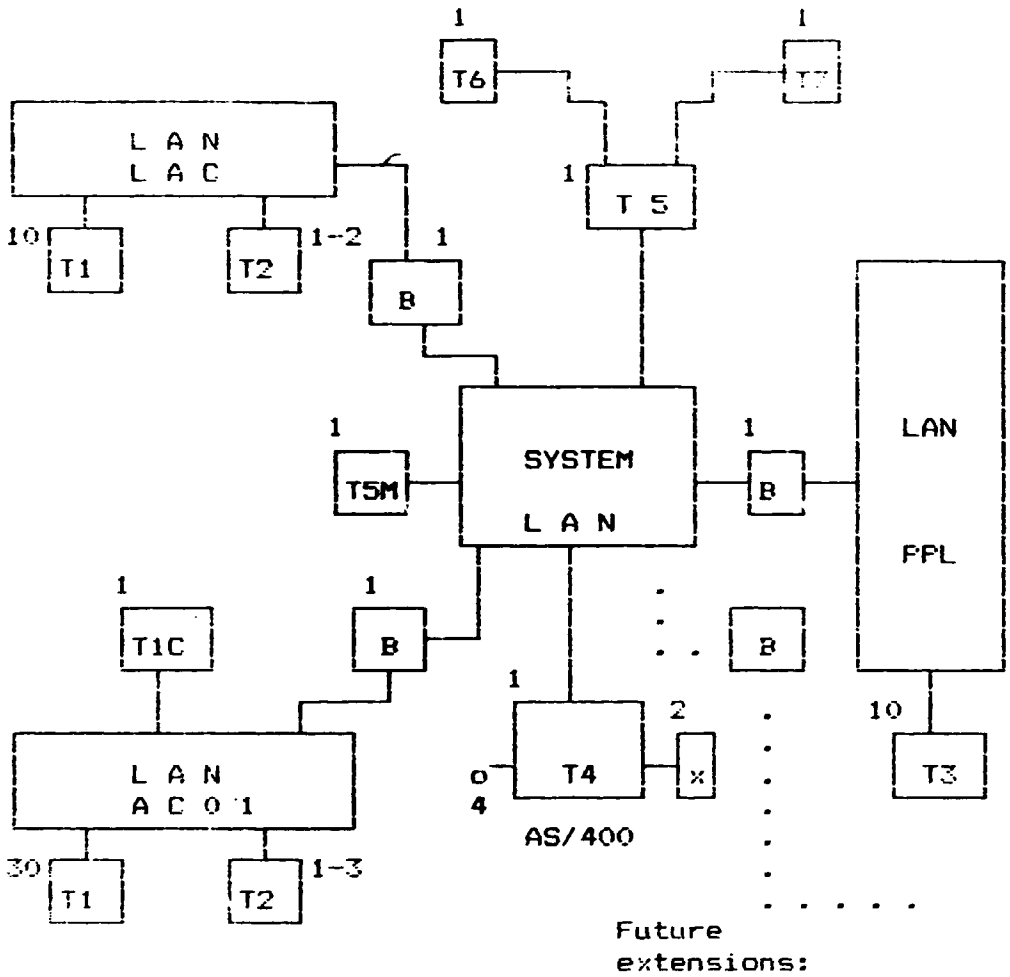
Domain: Production and Maint. management LAN: PPL

Typ	Function	Volume/Freq
F40	Local processing on each T3 type function	N/A
F41	T3 to T4 communication a) AS/400 command execution b) Inquires (Program -Program)	Random
F42	T3 to T1 message exchange	
F43	T3 to T2 communication a) file transfer	2 MB/day

Domain: System LAN functions

Typ	Function	Volume/Freq
F51	TSM to any message exchange	
F52	Network monitoring from TSM	
F53	Remote communication functions a) to/from laboratory b) to/from IBM 4381	200 B/min 5 MB/day
F54	Controlling T4 (AS/400)	
F55	File exchange between T2, T2 and T4 via TSM	

UNIDO/SIDER SGPM - NETWORK ARCHITECTURE



LEGENDA: Network node types

T1	-	Standard Workstation	(XT/640 KB, 20 MB HD)
T2	-	Server	(/386, 4 MB, 80 MB HD)
T3	-	Standard T1 at PPL	
T4	-	AS/400	(8 MB, 930 MB HD)
T5	-	Gateway station	(/286, 40 MB HD ...)
T5M	-	Network manamegement station	(/286, ...)
T6	-	Remote workstation	(laboratory)
T7	-	Host IBM 4381	
B	-	Token Ring Bridge	
o	-	IBM Infowindows	
T1C	-	Computational station	
[x]	-	Network printers	

C. THE SCOPE OF CONTRACTING SERVICES

Phase 1 : Selection of additional CPMMS software components

1a) Task description : To advise in selection of local area network software and database managment software for non dedicated servers operating under OS/2.

The network software should permit the following functions :

- OS/2 interprocess communications
- Industry standard multitasking
- File transfer to/from IBM AS/400
- Remote program execution
- Remote network monitoring
- Network activities scheduling
- User access control to server resources
- Coexisting of DOS and OS/2 applications on the network

The database management software for the non-dedicated server should have the following attributes :

- Concurent user access with file and record level locking.
- Access according to multiply key field.
- Programming interfaces from C-language.

Sophisticated database user interfaces on server level are not requested. Database volume is approx. 40 MB per server, divided into a great number of small data domains.

Furthermore, the Contractor should advise on :

- selection of program development tools in C under OS/2 and MSDOS
- Other OS/2 utilities (API) needed for server functions outlined in chapter A.

1b) Output : Specifications for additional software selection, containing justification of selected components, sources of purchase and estimated price.

1c) Implementation: Consultation with Project Chief Technical Adviser (CTA) and with the National Project Director (NPD) of CPMMS directly at SIDER or at Contractor's Home Office. Purchase of specified software components will be made separately by UNIDO and installed by CTA before commencing Phase-3.

In his proposal the Contractor should briefly outline his approach for SW selection.

Selection des composants additionnels de logiciel CPMMS

Proposition de SFW Software Anstalt pour accomplir Phase 1 du "Terms of Reference (UNIDO Project DP/ALG/87/028 Contract 90/128)"

1. Architecture de réseau et exigences fonctionnelles de réseau

Les experts d'informatique de SIDER El-Hadjar Complex et consultant technique en chef d'UNIDO, Mr. Attila Toth, pendant le visit de l'Officier exécutif en chef de SFW, Dr. Peter J. Bruck et chef d'équipe, Dr. Istvan Gergely à Annaba (de 09. 10. 1990 à 12. 10. 1990) ont proposé une architecture materielle pour le réseau UNIDO/SIDER et ont compilé une liste des exigences fonctionnelles comme une base additionnelle pour la specification du logiciel système (voir Annexe 1).

1.0 Considérations générales

Les "Terms of Reference" (May 1990, voir Annexe 2) donne une définition claire des fonctionnalités demandées. Deux alternatives ont été considerées, car toutes les deux peuvent satisfaire tous les exigences citées ci-dessus:

- un réseau basé Novell, utilisant OS/2 et Oracle comme serveur de base de données, comportant le support d'Oracle pour dBASE sur station de travail basée DOS

- une solution basée OS/2 édition étendue, comportant les gérants base de données et LAN propres à OS/2 ainsi que le requérant DOS (DOS Requester).

Consideré comme un produit séparé, et jusqu'à 1990 la solution basée Novell/Oracle avait sans doute des avantages significatifs quant à sa performance ainsi qu' à son interface utilisateur.

Avec la nouvelle version d'OS/2 (1.2), les deux alternatives sont beaucoup plus proches l'un à l'autre: considérant les revisions récentes (de séptembre 1990), les deux alternatives sont à peu près équivalents quant à leurs performances ainsi qu'à leurs capabilités.

Version 6 du gérant base de données Oracle est meilleur que la version actuelle du gérant base de données d'OS/2; néanmoins, dans un environnement où AS/400 et IBM 4381 seront aussi utilisés, nous considéront que le haut niveau de compabilité entre les trois produits IBM pour gestion des bases de données est plus important.

En général, OS/2 et son composants sont des parties integrales d'une architecture nouvelles et unifiée d'IBM (Systems Application Architecture - SAA), résultant un haut niveau de consistance logique et de portabilité. En conséquence, notre proposition actuelle est basée sur l'edition IBM du système d'exploitation OS/2. IBM s'est engagé à suivre cette direction, et il considere OS/2 comme un facteur décisif de l'industrie de l'informatique dans le décade suivante.

Dans ce rapport, nous analysons d'abord les composants de logiciel à acheter départant des exigances imposés par la specification fonctionnelle. Après, nous allons décrire un jeu consistant des composants logiciels à acheter pouvant satisfaire les expectations ci-dessus.

1.1 Architecture réseau

SFW a analysé l'architecture materielle du réseau de SIDER du point de vue de l'architecture du LOGICIEL, c.à.d.

- possibilités à appliquer des composants de logiciel de norme industrielle
- efficacité de l'opération (performance à expecter)
- aspects de maintenance et de gestion
- aspects de l'education/training de logiciel

Basé sur les aspects ci-dessus, SFW a proposé à considerer les solutions alternatives suivantes pour l'architecture du réseau (changement):

a.) La station de travail "laboratoire" (T6, voir Annexe 1) doit être connectée directement au réseau local (c.a.d doit être une partie intégrale du LAN et pas connectée comme une station éloignée). Comme le délai du retour du laboratoire à AC01 des résultats de l'analyse des données est très critique, et une éventuelle perte d'informations peut causer un dommage important, une solution basée modem semble être unjustifiée.

La solution materielle alternative est d'utiliser dans T5 le Programme serveur de connection asynchrone (Asynchronous Connection Server Program) (demandant une station DOS dédiée) et de connecter la station de travail laboratiore à travers

une paire de modems; cette solution ne demande pas de changement au niveau de l'application. (Une triosème solution alternative, qui demande un investissement considérable au niveau de programmation de l'application, sera discutée en bref sous F53 a.).

Si T5 tourne le Programme serveur de connection asynchrone (Asynchronous Connection Server Program), la même station gateway peut être utilisée pour la connection asynchrone au D.O.I).

b.) La station gateway (T5) - désignée à connecter les appareils éloignés - est à avoir deux fonctions: connecter des station de travail asynchrone (se trouvant dans le laboratoire et dans D.O.I.) ainsi que IBM 4381.

Pour des stations de travail éloignées, le Programme serveur de connection asynchrone (Asynchronous Connection Server Program) est recommandé. Ce programme se tourne sous DOS et demande une station dédiée.

Pour la connection mainframe, la carte d'interface appropriée et le support 3270 de OS/2 peut être utilisés, c.a.d. cette fonction a utilise un autre système d'exploitation que la connection asynchrone, et par conséquent les deux fonctions gateway doivent tourner dans deux PCs séparés. Pour la connection 3270 nous proposons à utiliser la station de gestion (T5M).

Considérant les remarques ci-dessus, SFW n'a pas observé de contradiction entre l'architecture materielle proposée et les buts du projet.
Utilisant l'architecture proposée, l'application des composants standards d'OS/2 et de DOS est possible. On ne prévoit pas d'étranglements apparents qui porraient causer une dégradation de performance du réseau. La segmentation facilite le maintenance et la gestion du réseau. L'education prévue est adéquate pour l'introduction le groupe SIDER à l'utilisation des composants principals du logiciel.

1.2 Exigences fonctionnelles de logiciel dans l'environment LAN

1.2.1 Domaine: Control de production (LAN: ACO1)

F10: Traitement local des bases de données dBASE sur des station de travail DOS (T1)

Le technique du traitement n'est pas changé par rapport de la gestion autonome (standalone) de base de données; l'overhead mémoire dû aux composants de station de travail du système d'exploitation doit être pris en considération. En conséquence, l'utilisation d'un système de gestion de base de données de type interprétation dans

la phase de production n'est pas recommendée; même si un interprétair est utilisé dans la phase de développement, nous suggérons à utiliser une version compilée dans l'environment de production (par example Clipper) sur les stations de travail dans l'intérêt d'une réduction d'exigence mémoire.

Nous recommendons que seulement le sous-ensemble commun des versions interprétée et compilée du langage dBASE soit utilisé dans la phase de développement, et que le test final soit effectué en utilisant la version compilée. Nous notons que les formats des fichiers indices de quelque système compatible dBASE est différent de celui de dBASE (par example, Foxbase utilise des indices multi-niveau, tandis que Clipper utilise un compatible dBASE).

En outre, ce sera nécessaire à lire/écrire des fichiers dBASE it will be necessary to read/write dBASE à partir des programmes C, qui, en parallèle, doivent utiliser des resources réseau variés. Pour accès aux fichier dBASE à partir des programmes C, nous recommendons des paquets CodeBase de Sequiter Software.

F11: Communication entre stations de travail DOS de même domaine (T1 - T1)

a.) Messages

Trois Three mécanismes simples peuvent être utilisés pour envoyer des messages entre des stations de travail:

I. En utilisant la commande de LAN serveur de OS/2

net send adress-destination

Le texte du message est lu de l'entrée standarde (c.a.d. peut être entrée à travers du clavier).

II. En utilisant la facilité de redirection, l'entrée à la commande peut être lue dans un fichier:

net send adress-destination < fichier-message

Limitations: message est supposé d'être de texte. La mesure du message est limitée par la dimension du buffeur au côté de réception
(4 Kb par défault, 64 Kb au maximum).

III. A partir des program: en utilisant l'API NetMessageFileSend d'OS/2. Pour un listing examplaire, voir Figure 2. Cette solution est plus flexible; elle permet, par example, la specification de plus d'une destination; limit de dimension: fichiers textes de 64Kb.

Cette méthode illustre une facilité importante d'OS/2: les fonction du processeur de commandes sont accessible à partir des langages de haut niveau, par example C sous le format des appel à l'Interface de programmes d'application (API - Application Program Interfaces).

b.) Transfert de fichiers

I. Méthodes II. et III. ci-dessus, utilisées pour le transfert des messages, peuvent être utilisées aussi pour le transfert des fichiers avec une limitation de dimension pareille. Dans des cas certains, c'est possible à utiliser des "workarounds", par example, dans le cas des bases de données dBASE, un certain nombre d'enregistrements est converti à format ASCII dans chaque pas et transferé vers la station de travail éloignée (dans un grand nombre d'applications, ce sont seulement des changements après un évenement donné qui contien des informations essentielles).

II. Une solution alternative (proposée par Microsoft) est comme suit:
La station de travail envoyeur tranfère le fichier au saerveur utilisant par example la commande

net copy workstation-filename server-filename

et envoie un message à la station de travail de destination indiquant que le fichier est displonible sur le serveur et peut être recopier à la station de travail de destination. Cette operation de copie peut être lencée du station de travail de destination en utilisant une autre commande "net copy" avant que le ficher est à acceder localement.

Dans ce cas, les API équivalents sont disponibles pour exécution à partir des programmes.

III. La méthode la plus fléxible d'échange d'informations entre des stations de travail DOS est la Communication avancée programme à programme (APPC - Advanced Program-to-Program Communications) qui est une méthode commune entre des membres du IBM Systems Network Architecture (SNA): IBM PC (DOS-OS/2), AS/400 et IBM mainframes (Figure 4-5).
APPC definit un jeu de services de communication inter-program permettant la conversation entre des programmes distribués. Le programme s'exécutant sur la station de travail enoyeur peut transferer des données à un autre programme activé sur la station de travail récepteur.

F12: Communication entre une station de travail DOS et un serveur OS/2

a.) Messages

Identique à F11 a.) ci-dessus

b.) Echange de fichiers

La commande

net copy source-filename destination-filename

du serveur LAN OS/2 peut être utilisée pour copier des fichiers entre une station de travail et un serveur.

c.) Echange de niveau enregistrement

La méthode la plus génerale pour envoyer et recevoir des enregistrements est l'application des programme utilisateur utilisant APPC [voir F11 b.) III. ci-dessus]; une , a solution alternative et plus simple est de copier des enregistrements sélectionnés dans un fichier temporaire et de transferer ce fichier du station de travail au serveur (c'est la seule solution faisable pour les fichiers dBASE situés sur la station de travail). Pour l'autre direction, la solution est encore plus simple: pour avoir accès aux fichiers/enregistrements sur le serveur les services de transfert de fichier (ou APPC) ne sont pas nécessaires car la station de travail voit des resources du serveur comme s'ils était locals au station de travail.

d.) Enquêtes

Une enquête est la visualisation sur l'écran de la station de travail des données (stockées sur le serveur). Les fichier serveur sont displonibles comme s'ils résidaient sur la station de travail.

- fichiers ISAM

Des enregistrements peuvent être accédés par des programme d'application (par example des programme Cobol) s'exécutant sous DOS dans le mémoire de la station de travail

- fichiers dBASE

Des enregistrements peuve être accédés en utilisant des programme d'application dBASE s'exécutant sous DOS dans le mémoire de la station de travail; à cause des exigences importantes de mémoire du logiciel de la station de travail, il est proposé d'utiliser la version compilée des programmes dBASE (qui a aussi pour résultat la diminution du temps de chargement des programmes); alternativement, les enregistrement dBASE peuvent être accédés à partir de programmes écrits en C en utilisant des appel CodeBase.

- Fichiers du gestionaire de base de données

Le Gestionaire de base de données (Database Manager) est le système DBMS relationnel d'OS/2, et il est compatible avec AS/400 et 4381; en conséquence, en cas des données à partagées avec l'AS/400 ce format de fichier peut avoir d'importance. Les bases de données OS/2 peuvent être accédées en utilisant l'option Accès éloigné des données (RDS - Remote Data Access) qui support des environement DOS et OS/2 (voir Figure 1). Sous DOS, un programme d'application (par example un programme écrit en Cobol) peut utiliser des commandes (precompilées) SQL de haut niveau pour avoir accès aux bases de données éloignées se trouvant sur le serveur OS/2. Ces programmes sont portable et peuve être utilisés sous OS/2 et sous OS/400.

F13: Communication entre station de travail DOS et serveur AS/400 (T1-T4)

IBM fournit - comme une partie du système d'exploitation OS/400 operating system - un programme PC appelé PC Support, pour avoir accès au AS/400 à partir des stations de travail DOS et OS/2. L'exigance mémoire de PC Support dépend sur la configuration mais, en général, elle est considérablement haut (200 à 300 koctets). En conséquence, elle doit être activé sur la station de travail DOS seulement pour le temps de connection. (IBM a promis de réduir l'exigence mémoire dans la version suivante - fin d'année 1990 - par la suppression automatique des fonctions temporairement superflues de PC Support).

a.) Transfert/échange de fichiers

I. PC Support doit être utilisé pour l'échange des fichier propre à AS/400 (fichiers EBCDIC) avec des station de travail DOS (orientées ASCII); c'est recommendé d'utiliser la facilité intégrée de PC support: la conversion des caractères. Ensuit, c'est très convenable à utiliser PC Support pour convertir et transferer des enregistrements - même selectés - des bases de données de AS/400 Database Manager au fichiers ASCII DOS. (L'autre direction donne plus de travail car la description de la base de données ainsi que celles des enregistrement et des champs doit être données pour la conversion d'un fichier ASCII à une base de données relationnelle.)

II. L'AS/400 peut aussi stocker des fichier de type PC (ASCII); ces fichier peuvent être accédés en utilisant PC Support comme s'ils étaient sur un disque local de la station de travail (similaire au facilité "disque virtuel" du Système/36).

III. Dès qu'une correction d'une erreur de l'OS/2 sera relâcher (promis pour 1990), le serveur OS/2 sera capable à gérer des fichier compatible PC sur AS/400 comme des "volumes éloignés"; en conséquence, ces fichiers seront accessible pour toutes les stations de travail DOS comme tous les autre fichiers sur le serveur OS/2 en utilisant seulement le logiciel station de travail DOS d'OS/2 (50 à 80 koctets de code) sens utilisant donc PC Support.

Notre consultant spécial AS/400 recommends à convertir des fichiers propres à OS/2 ainsi que les bases de données en fichier de type PC-TYPE sur OS/400 avant d'être accédés d'une station de travail (les fichier convertis seront rengés sur des même volumes que des fichiers originals). Après la conversion performée par des programmes s'exécutants sur l'AS/400 et lancés d'une station de travail, ces fichiers peuvent être accédés de même façon que les fichiers DOS locals. Dans beaucoup de cas, cette solution peut remplacer l'utilisation de l'APPC DOS-OS/400, qui offre une solution plus générale mais aussi plus exigeant.

b.) Enquêtes des bases de données AS/400

Les station de travail du réseau peuvent se connecter (log in) au AS/400 - en utilisant PC-Support - et avoir accès aux bases de données comme si elles étaient des terminaux locaux.

F14: Communication entre serveur OS/2 et serveur AS/400 (T2-T4)

a.) Echange de fichiers

Toutes les méthodes citées sous F13 a.) sont disponibles pour des serveurs OS/2. Le transfert de fichier de du/au serveur OS/2 peut être initialisé d'une station de travail ou, si le serveur n'est pas dédié (c.a.d. il est disponible pour exécuter les programmes utilisateur), du serveur OS/2 lui même. Due à la capabilité multi-tâche de l'OS/2 - avec la suppression de la limitation de mémoire operationnelle à 640 koctets - ça peut être utile pour quelques applications d'utiliser PC-Support sur le serveur comme une tâche OS/2.

b.) Echange d'enregistrements

Le mécanisme général disponible pour l'échange d'enregistrement est l'APPC (Advanced Program-to-Program Communication): un programme s'exécutant dans l'ordinateur requérant accomplit une procédure d'handshaking avec un programme approprié s'exécutant à côté serveur, envoie une requête demandant certains

enregistrements qui, à leur tour, seront recherchés et transferés au requéramt par le programme s'executant dans le serveur.

Le support APPC est disponible dans Cobol est disponible dans Cobol OS/2 (langage recommendée pour programmation des application). Il est à noter que c'est le programme Cobol qui doit assurer la conversion ASCII-EBCDIC.

Dans le cas des fichier du gestionaire de base de données, le langage de haut niveau SQL (Structured Query Language) peut être utilisé pour retirer les enregistrements à transferés; les commandes SQL sont aussi disponible dans Cobol sous tous les deux système d'exploitation.

IBM projette, qu'à partir de 1993, les bases de données OS/400 seront accessible à partir des systèmes OS/2 connectés en réseau comme si elles étaient locales.

C'est à noter que l'étude, le développement, et la mise au point de programme APPC OS/2 - OS/400 est une tâche avancée demandant une certaine expérience. Nous suggérons d'acquérir une documentation spéciale (pour tous les deux système d'exploitation) et des programmes examplaires (par example AdaptSNA/2 de Network Software Associates).

Il est possible d'utiliser des solutions intermédiaires tandis qu'une solution basée APPC devient disponible: par example, initialiser l'exécution d'un programme AS/400 qui copie des enregistrements nécéssaires dans un fichier, et après, transfere ce fichier au serveur OS/2.

F15: Communication entre station de travail DOS et station de travail laboratoire (T1 - T6)

Soit la station de travail laboratoire connectée directement au réseau anneau à jeton où soit elle connecté à travers un serveur asynchron pour le programme d'application la situation est similaire au cas F11.

F16: Communication entre station de travail DOS et station de travail dédiée aux calculations (T1 - T1C)

Les solutions proposées pour F11 /par example méthode a.) II/ peuvent être utilisées; le problème peut aussi être résolu en utilisant APPC. Il est à noter que si la station de calculation travail sous DOS, elle ne peut servir qu'une seule requête simultanément; si cette solution est inadéquate, cette station doit être utilisée sous OS/2.

F17: Communication entre stations de travail DOS attachées aux serveurs OS/2 différents (T1 - T2 - T2' - T1')

Si les stations serveur sont dans les anneaux différents, les anneaux doivent être connectés par de ponts. Le pont est un PC dédié à exécuter un logiciel (par example IBM Token Ring Bridge Version 2.2) qui cache les difference: T1 peut communicer avec T1' comme si tous les deux étaient dans le même anneau.

a.) Echange d'enregistrements

L'échange d'enregistrements est possible en utilisant APPC. Il est à noter, qu'avant d'écrire un programme APPC, il est recommendé d'examiner l'utilisation combinée de transfert de fichiers et le service d'échange de messages.

b.) Echange de messages
Voir F11 a.).

1.2.2 Domaine: LAC control de production LAN: LAC

Les fonctions F20 à F24 sont equivalentes aux fonctions F10 à F24 discutées sous 1.2.1

F25: Communication entre station de travail DOS située à PPL et serveur OS/2 situé dans un domaine réseaue différent (T3 - T2')

Supposant que le domaine containant la station de travail et celui containant le serveur sont bien connectés avec un pont, la solution est equivalent avec le cas F12. Pour le service de transfert de fichiers, une solution possible pour des fichier plus important est de les copier à travers des fichiers temporaires à une station serveur; pour le service d'échange de messages, l'utilisation de la commande "net send" est recommendée.

F26: Communication entre station de travail DOS située à PPL et une autre station de travail DOS située dans un domaine réseau différent (T3-T1').

Supposant que les deux domaines sont bien connectés à travers de pont, les méthodes disponible pour l'échange de messages sont les même que dans le cas de F11 (la solution la plus simple est d'utiliser la commande "net send").

1.2.3 Domaine: Gestion de maintenance LAN: ACO1 + LAC + PPL

F30: Traitement local des travaux de maintenance sur station serveur OS/2 (T2) avec impression sur AS/400 (T4).

PC Support sous OS/2 peut être utiliser pour avoir accès au AS/400; PC Support rend disponible toutes les facilités de l'impriment de AS/400 comme s'il était un impriment local dédié à la station où PC Support est utilisé.

F31: Communication entre stations de travail DOS et serveur OS/2 (T1 - T2)

La situation est équivalente avec F12.

F32: Communication entre stations de travail DOS situées dans des domaines differents (T1 - T1')

S'il y a un pont bien configuré entre les deux domaines, ce cas est équivalent avec F11.

F33 - F34: Enquêtes à partir des stations de travail DOS au serveur AS/400 (T1 - T4 et T3 - T4)

La situation est équivalente avec F13 b.); l'utilisation du PC Support AS/400 est recommendée.

F35 - F34: Enquêtes à partir des stations de travail DOS de PLL à serveur OS/2 (T3 - T2)

La situation est équivalente avec F12 d.)

1.2.4. Domaine: Production et gestion de maintenance LAN: PPL

F40: Traitement local sur station standarde de travail DOS de PPL (T3)

(Voir F10).

F41: Communication des station de travail DOS avec serveur AS/400 (T3-T4)

a.) Execution de commande sur AS/400

Il est recommendé à utiliser la station de travail comme un terminal AS/400 en utilisant PC Support (voir aussi F13).

b.) Enquêtes (programme sur PC - programme sur AS/400)

La méthode standarde de la communication programme-à-programme dans l'architecture SNA est l'utilisation d'APPC; cette facilité est disponible aussi des langages de haut niveau, par example Cobol OS/2.

F42: Communication entre station de travail DOS situées dans des anneaux differents (T3 - T1).

Supposant une connection propre entre des anneaux en utilisant d'un ou plusieurs ponts, ce problème est équivalent avec F11; l'utilisation de la commande "net send" est recommendée.

F43: Communication entre station de travail DOS et serveur OS/2 (T3 - T2)

Cette situation est équivalente aved le cas F12; la simplest méthode la plus simple de transfert de fichier est l'utilisation de la commande "net copy".

1.2.5. Domaine: LAN système

F51: Communication entre la Station de gestion de réseau (Network Management Station) (T5M) et une station quelconque.

La Station de gestion de réseau (Network Management Station) est une station éxecutante un programme pour supporter ;a gestion/administration de LAN sous OS/2 EE. La commande "net send" peut être utiliser pour transferer des messages à n'import quel station dans le réseau.

F52: Surveillance du réseau de la Station de gestion (Network Management Station) (T5M)

Fonctionalités nécessaires pour une gestion efficace de réseau:
- gestion des erreur
- gestion de la configuration
- gestion du performance
- gestion de la facturation
- gestion de la sécurité

Une solution avancée et économique pour supporter la gestion de réseau et l'analyse des erreurs est d'utiliser le logiciel "LAN Command" de Dolphin Software. Nous recommendons d'utiliser un visu de type VGA couleur.

F53: Fonctions éloignées de communication

a.) Du/au laboratoire

La commande "net send" (ou des API equivalents) peuvent être utiliser pour envoyer des fichier court de texte de la station de travail laboratoire aux autres stations de travail. La station de travail laboratoire est recommendée d'être connectée directement à l'anneau de jetons; si la station est connectée à travers d'une ligne asynchrone, le Programme serveur de connection asynchrone (Asynchronous Connection Server Program) (executé sous DOS sur T5) est à utiliser; néanmoins, la fiabilité de cette solution est plus faible. Etant donné que cette connection est cruciale pour l'utilisation du sous-système ACO1, nous recommendons la connection basée d'anneau de jetons.

Pour communicer avec la station de travail laboratoire sens d'anneau de jetons ou serveur asynchron, la station peut être connectée au T5M à travers d'une ligne sériale et la communication gerée par le programme Asynchronous Communications Device Interface (ACDI) de OS/2; T5M peut alors communiquer les informations reçues aux autres stations de travail dans ACO1 en utilisant APPC OS/2 et APPC/PC.

b.) De/à IBM 4381 (émulation 3270)

OS/2 supporte l'émulation 3270, y compris les supports des impriments et de transfert de fichiers (Figure 3). Néanmoins, il est à noté que la connection demande la présence d'un interface materiel approprié pour le gateway, et que la fonction d'émulation de l'OS/2 n'est disponible que sur des stations OS/2; chaque station de travail DOS demandera l'addition d'un programme d'émulation PC/3270 (Version

1.01 ou plus avancée). Comme SIDER n'était pas encore dans la position à fournir des informations détaillées sur la configuration materielle de 4381, cette question demandra une clarification future.

c.) Ensuite, il est prévu que D.O.I demandera d'avoir quelque sorte d'accès au réseau, et il est probable qu'on aura besoin de queldue sorte de connection avec l'AS/400 et le 4381. Néanmoins, nous pensons qu'une specification detaillée du materiel et du logiciel doit être éffectuée séparement et dans un stage futur, car elle a des effets primairs sur le materiel et logiciel mainframe/minicomputer (le support logiciel côté LAN est déjà inclu dans la proposition actuelle).

F54: Contrôle de l'AS/400 (T4)

Chaque station DOS et OS/2 où PC Support est installé peut se connecter (log in) à l'AS/400 en utilisant le support terminal. Ensuite, le Communication Manager OS/2 contient le IBM 5250 Workstation Feature, qui est disponible des stations OS/2.

F55: Echange de fichiers entre serveurs (T2, T2' et T4) à travers de la Station de gestion de réseau (Network Management Station) (T5M).

Le but de cette fonction est à permettre à l'administrateur de réseau à "enforcer" le transfert de fichiers entre serveurs. Cette tâche peut être exécutée en utilisant la commande "net copy" du serveur LAN.

2. Méthodes/outils logiciels pour accomplir les demandes des programme d'application

Dans ce chapitre, nous voulons catégoriser les exigances de communication, et arriver à un jeu consistent des outils logiciels qui peut être utilisé pour satisfaire des exigences.

2.1 Tâches de communication grouper selon les types de station

Exigence de communication entre:	Méthode proposée:
station de travail DOS - station de travail DOS	
échange de messages (F11,F15,F17, F21,F26,F32,F42,F53)	commande "net send" ou des API NetMessage
échange de fichiers (F11,F21,F32)	commande "net send" pour des fichiers ASCII petits; commande "net copy" à travers des fichier temporaire serveur; APPC
échange d'enregistrements (F15,F16,F17,F53)	APPC
station de travail DOS - station de travail/serveur OS/2	
échange de messages (F12,F22,F25,F51)	commande "net send" ou des API NetMessage
échange de fichiers (F12,F22,F25,F31,F43)	commande "net send" pour des fichiers ASCII petits; commande "net copy" à travers des fichier temporaire serveur; APPC
échange d'enregistrements (F12,F22,F31)	APPC
enquêtes (F12,F22,F31,F35)	Programme Cobol/dBASE; RDS pour des fichiers u Gestionaire de base de données
serveur OS/2 - serveur OS/2	
échange de fichiers (F55)	commande "net copy"
station de travail DOS - serveur AS/400	
échange de fichiers (F13,F23)	commande "net copy"
échange d'enregistrements (F41)	APPC
enquêtes (F13,F23,F33,F34)	PC Support
exécution des commande AS/400 (F41,F54)	PC Support
station de travail/serveur OS/2 - serveur AS/400	
échange de fichiers (F14,F24,F55)	commande "net copy"
échange d'enregistrements (F14,F24)	APPC
impression sur AS/400 (F30)	PC Support; APPC

2.2 Résumé des outils proposés

a.) commandes OS/2 et LAN (par example "copy", "net send", "net copy")
b.) PC Support AS/400
c.) Programmes d'application (Cobol, dBASE)
d.) Services éloignés de données (Remote Data Services) (RDS) du Gestionaire de base de données OS/2 (OS/2 Database Manager)
e.) Communication avancée programme-à-programme (Advanced Program-to-Program Communication) (APPC)
f.) Application Programming Interfaces de l'OS/2
(des API, par example NetMessageFileSend)

Remarques:

1. Cette liste reflète la compéxité croissante des outils. Dans la plupart des cas, il est avantageux d'utiliser l'outil approprié le plus simple.

2. Nous proposons un jeu relativement petit des outils avec une applicabilité étroite pour minimiser l'effort de programmation niveau système.

3. RDS et APPC sont supportés de Cobol sous OS/2. L'utilisation du langage C est recommendée pour des appel API sous OS/2 et DOS; pour APPC/PC (sous DOS), quelque support en langage assembleur peut être nécessaire.

Solutions alternatives:

1. Si l'AS/400 est à être accédé pour des enquêtes à partir des programmes utilisateur, les Emulator High Level Language API (EHLLAPI) sont à utiliser au lieu de PC Support; Ils contiennent l'émulation de station de travail 3270 et 5250 (ce logiciel fait une partie intégrale de l'OS/2; les documentations optionelles nécessaires sont listées dans le Chapitre 4).

2. Pour la communication entre stations de travail DOS, l'interface NETBIOS peut aussi être utilisé au lieu de l'APPC. Ce logiciel faisant partie de LAN Support Program est déjà disponible à SIDER; la documentation se trouve dans le manuel de référence technique de IBM Local Area Network (voir chapitre 4). Du point de vu strategic, la différence principale entre APPC et NETBIOS est que le support APPC est disponible sur tous les membres de la famille SNA (y compris AS/400 et 4381), tandis que NETBIOS est specifique PC.

3. Description des composants logiciels proposés pour CPMMS

3.1 OS/2 édition étendue (Extended Edition) Version 1.2

OS/2 est un système d'exploitation avancé mono-utilisateur, multi-tâches, offrant
- un interface utilisateur graphique (Presentation Manager), assurant un environnement multi-fenêtres pour voir et contrôler plusieaurs programmes d'application simultanement et pour manipuler des fichiers
- le Communication Manager à gérer des échanges d'informations avec d'autres stations de travail et des machines host, et supportant les émulations variées des terminaux, communication asynchrone, connection avec AS/400 (5250 Workstation Feature), gateway SNA
- un sous-système de gestion de base de données relationnelle aux fonctions complètes (Database Manager), compatible avec des ordinateurs plus grands d'IBM (par example AS/400), avec support de Structured Query Language (SQL); accès aux donées sans programmation programming (en utilisant la gestion des enquêtes - Query Manager); supports des fonctions de gestion de base de données à travers de LAN avec des services éloignés de données (Remote Data Services); verrouillage de fichier/enregistrment en cas d'accès concurrant.

Une caractéristique importante d'OS/2 est l'accessibilité de toutes les fonctions du système d'exploitation à partir des langages de haut niveau de programmation avec l'utilisation des appel aux interfaces de programmation d'application (des API - Application Programming Interfaces).

3.2 Serveur LAN d'OS/2

Le serveur LAN d'IBM (équivalent à gestionaire de LAN de Microsoft - LAN Manager) offre le partage des rsources à travers des réseau locaux. Des appels des programme d'application demandant accès aux resources (des fichiers, des imprimant, des appareils sérials etc.) issus des stations de travail sont reroutés par le requérant LAN vers le réseau. Ces requêtes sont reçues par le serveur LAN, les resources demandés sont mis en disposition et les réponses sont rentrées au programme d'application ayant fait la requête comme si l'appel était satisfait locallement.

Fonctions principales:

- partage des resources
- partage des programmes d'application (des programmes exécutés dans la station de travail)
- exécution éloignée de programmes (des programmes exécutés dans le serveur)

- services de messages (des messages peuvent être affichés sur l'écran, reroutés ou stockés)
- support important de transfert de fichiers
- contrôle de l'accès au LAN
- journalisation et statistiques

3.3 Outils de programmation d'OS/2 et des informations (OS/2 Programming Tools and Information)

- Contient la gestion des dialogues (Dialog Manager) pour le développement simple des applications de fenêtre orientées textes; peut être appelée à partir de Cobol, C, etc.
- code source des programmes exemplaires utilisants des services importants d'OS/2
- référence technique des APIs d'OS/2

3.4 Outils à développement de programme sous OS/2

- Le compilateur C/2 d'IBM est recommendé pour programmation système, en spécial quand des appel OS/2 (des APIs) sont à utiliser; une partie aignificative d'OS/2 était aussi écrite en C

- Le compilateur Cobol/2 d'IBM est recommendé pour la programmation des application "non-computational" par example systèmes d'information et traitement des transaction; Cobol/2 permet l'utilisation des phrases SQL incorporées et supporte des appel APPC pour des applocations à traitement distribué.

- Le Live Parsing Editor est un outils nouvel d'IBM facilitant le développement rapide des programme en languages de haut niveau sous OS/2.

3.5 Outils de développement de programmes sous DOS

- Le compilateur C de Microsoft (Version 6) est recommendé pour la programmation des tâches systèmes, en spécial quand des API DOS sont à utiliser. (Ce compilateur peut aussi être installé sous OS/2 et il complète le compilateur C/2 IBM qui est une version modifiée du premier.)

- Le compilateur Cobol/2 d'IBM est recommendé pour la programmation des application "non-computational" par example systèmes d'information et traitement des transaction;

- IBM Macro Assembleur/2 est l'assembleur des processeur de la fammille Intel 8086

CodeBase de Sequiter Software est une facilité d'access aux fichiers dBASE est peut être appelée à partir des programme en C. Multi-utilisateur, 85k dBASE compatible applications portable vers OS/2 peuvent être construites. Code source est compris qui permet à l'utilisateur de n'inclure que le code minimum nécessaire pour ses applications

- Le LAN Support Program d'IBM contien le support NETBIOS pour des PCs

- Le Norton Editor est un éditeur éfficace orienté écran à utilisation simple supportant le développement des programmes.

3.6 Logiciels additionnels pour le support APPC

APPC/PC

C'est la version DOS de la facilité Communication avancée programme-à-programme (Advanced Program-to-Program Communication - APPC), nécessaire pour le support d'échange d'enregistrements entre une station de travail DOS et un autre ordinateur dans le LAN de SIDER (OS/2, AS/400 et 4381 supportent APPC, car ce dernier fait partie de Systems Network Architecture /SNA/ d'IBM).

AdaptSNA/2 APPC

Ce paquet - écrit par Network Software Associates (California) - réalise le Advanced Program-to-Program Communication d'IBM pour des applications OS/2-OS/2, OS/2-DOS, OS/2-Mini et OS/2-Host, y compris un sous-système d'application et un système tutorial (ASSIST).

3.7 Logiciel de pont pour des LAN anneau à jetom

Ce logiciel s'exécute dans un PC dédié (standalone) ayant deux cartes d'interface pour connecter deux anneaux séparés.

3.8 Commande LAN pour gestion de LAN et l'analyse des erreurs

Ce logiciel - écrit par Dolphin Software Inc. - est l'un des logiciels les plus avancés destinés pour la gestion/diagnostiques des réseaux locaux. Il stocke dans sa base de

données la description et l'histoire complete des erreurs de chaque noeud. Les fonctions de surveillance génerent des graph et charts d'activités en temps réel. Les outils de performance et de diagnostiques sont disponible pour tester n'import quel segment du réseau. En cas des problèmes, de recommendations sont imprimées sous forme de textes.

3.9 Le Norton Guides pour le langage C et OS/2

Un jeu excèllent d'informations indexéesd avec des cross-reference; il peut être stocker sur un serveur pour évirter la nécessité de plusieurs copies de manuels.