PROTEO SYSTEM: SOFTWARE STRUCTURE AND SOFTWARE SUPPORT FOR TANDEM EXCHANGE RELIABILITY

C. Carrelli
SIP-Società Italiana Per l' Esercizio Telefonico p. a.
Roma, Italy
R. Galimberti and G. Valbonesi
SITS-Società Italiana Telecomunicazioni Siemens p. a.
Milano, Italy

ABSTRACT

This paper discusses software efforts needed in developing and managing PROTEO, a modern telecommunication system, in order to assure the required service features, high availability and maintainability and self-recovery. Particular emphasis is laid on rigid interdependence among these features, hardware and software system architecture and support software.

1. INTRODUCTION

PROTEO digital exchange is designed for tandem, tell and trunk applications and is intended to serve, in connection with Peripheral (local) Exchanges (CT), a switching area up to 30000 subscribers (1,2).

The digital exchange is basically composed of two main blocks (fig. 1): the Transit Network (RT) and the Central Control (CC).

The Transit Network includes the digital switch with its access devices (UCT), plus all additional units dedicated to signalling preprocessing (UCS, UCM..). The digital switch is a three stage TST non blocking network with a capacity up to 512 2Mbit/s PCM links. Each PCM link is interfaced with a single line unit (UL). Both T stages of the switch are made up of 64 "group units", each one collecting eight line units. Each group unit switches 256 time slots and incorporates its own "speech memory" with the corresponding address memory. The S stage is a 64 x 64 single plane serial transmission matrix with its own address memory. The nom blocking condition is achieved,according to Clos, with a number of 512 internal switching times. The UCT block, as mentioned above, interfaces the digital switch to the central control.

The preprocessing units are specialized according to the signalling system they handle, namely line signalling (UCS), MFC signalling (UCS/MFC) and common channel or remote control signalling (UCM). Line signalling time slots (16th time slot of each PCM link according to CCITT Rec. G 732) as well as common channel signalling time slots are switched through the digital network and then connected to preprocessing units. Each preprocessing unit has a microprogrammed control and is designed to serve a given number of channels (e.g. 900 channels for the UCS); the quantity of required units is therefore dependent on exchange configuration.

The central control is basically made up with a pair of specially designed computers; processors synchronization procedure is controlled by two synchronization and updating units (UAS), while all the check and test functions are monitored by a double surveillance system (SS). The 24 bit processor has microprogrammed CPU with 200 nsec microcycle time. Core memory has 24 bit work length and 1.2 µsec cycle time; it uses 16 K words modules and is expandable up to256 K words. The CC has a powerful input/output system, and for a fast operation up to 60 Direct Memory Access channels may be used. The envisaged traffic capacity the Central Control can handle is about 150000 busy hour call attempts. The Central Control is connected, via a data channel, to a service computer (ES) which carries out operational maintenance and administrative functions e.g. subscriber class modifications, system configuration changes, meter charging read out, test and diagnosis program aids.

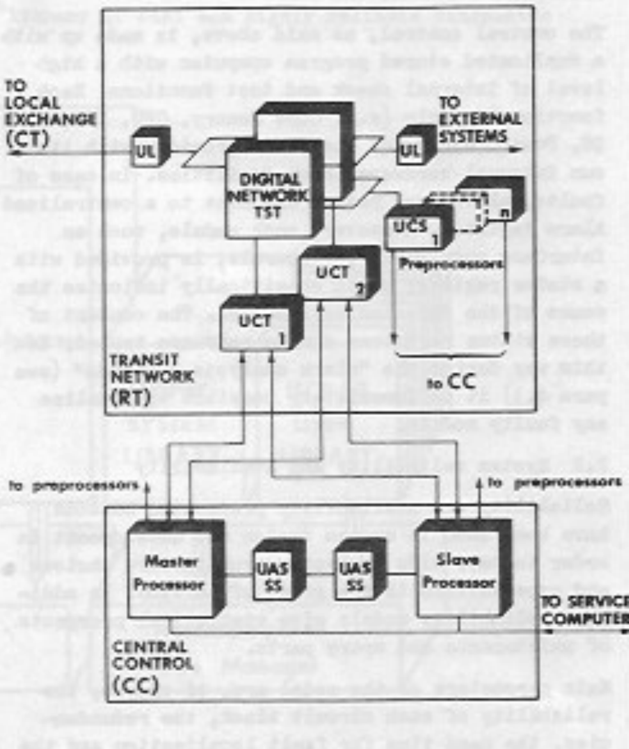

Fig. 1 – PROTEO Tandem Exchange

Section 2 presents an overview of system redundancy, recovery and diagnostics procedures. Section 3 describes the main software supports in designing and developping PROTEO system. Section 4 shows the on-line software structure.

2. SYSTEM REDUNDANCIES AND DIAGNOSIS

2.1 System redundancies

To enhance system reliability and service availability, important checking functions have been created. Redundancies have been provided at block, circuit, function and information level. Duplication, according to master/slave philosophy, has been the redundancy policy throughout all the system design. Each slave unit is to be intended as "hot" and continuously ready to be switched to the master status.

The digital TST switching network is completely duplicated from the input to the output section of the line unit UL. The two parts are synchronously addressed and synchronously work so that the outgoing bits from the two paths are compared channel by channel to allow a continuous check. Should a mismatch occur between the two sections it will immediatly be detected. The same duplication criteria have also been applied to UCT blocks as well as to preprocessing units; however for the last units alternative configurations may be realised with a n+1 redundancy philosophy.

All control messages, as well as data messages flowing through the system (e.g. word transfers within the computer, information exchange between different blocks) have appropriate redundancy bits to check the correctness of information transmission.

The central control, as said above, is made up with a duplicated stored program computer with a high level of internal check and test functions. Each functional module (e.g. Core Memory, CPU, I/O, UAS, SS, Power Supply) of the CC is provided with its own internal hardware check facilities. In case of faults, alarms are raised and sent to a centralized Alarm Register. Moreover, each module, such as interface modules to peripherals, is provided with a status register which specifically indicates the cause of the detected malfunction. The content of these status registers can be software tested; in this way during the "alarm analysis subphase" (see para 4.3) it is immediately possible to localise any faulty module.

2.2 System reliability and availability

Reliability and availability prediction methods have been used in system design and development in order to have aids in system architecture choices and expectations in the grade of service. In addition reliability models give statistical prospects of maintenance and spare parts.

Main parameters of the model are, of course, the reliability of each circuit block, the redundancies, the mean time for fault localisation and the mean time to repair. The first parameters must be satisfactorily improved in design while the last one in planning and carrying out maintenance. The redundancy structure itself gives the main indication in the policy to cut off the faulty parts.

In the central control, one processor is cut off by being put in "slave" status, in such a manner that operating peripheral equipments do not receive its commands. In the digital switch the selection of the master section is made, upon CC command, enabling the UL (Line Unit) to receive the switched bits only from the "master" network.

In case of more than one existing faults particular care is needed in selecting the master networks; fault classification programs have been carefully designed in order to optimise the system efficiency. The same philosophy applies to a faulty preprocessor; it is simply cut off blocking its operational outputs. The CC disables itself to receive any messages from the faulty peripheral and does not route through the digital switch the time slots coming from the faulty signalling preprocessor. When a fault occurs in parts working in load-sharing basis (e.g. the trunks) the CC provides normal procedures to put them in "unavailable status".

2.3 Diagnosis phases

The first task in fault handling is to detect any malfunction and in addition to collect all available fault information; this "check and alarm phase" is performed by hardware circuits, by software traps or software cyclic test programs.

The alarms activate the "first level of diagnosis" phase in order to accomplish the proper recovery procedure identifying the correct part to be isolated. Particular diagnosis programs, performed by CC, if necessary complete tests to find out which function is wrong and which new system configuration is to be used. System reaction time of this phase has to be rather short to have as little service trouble as possible.

Fault localisation to identify the faulty block (only one or a few boards) for maintenance is carried out during the "second level of diagnosis". Proper software programs resident in the Service Computer (ES) for sending and/or activating test routines in the CC and analysing the results, carry out this accurate diagnosis. There is no strict time problem here,because of the necessary human intervention needed for the repair work. Before putting the repaired unit back into service again a complete check is carried out and its memory is updated.

2.4 Tandem exchange system tests

Testing of factory produced units has been organised with a view to optimising fundamental points:
a) test procedure automation
b) coverage of "all" possible causes of malfunction
For the furtherance of the objective above systems controlled by minicomputers have been adopted. Programs directly obtained from functional tests or by diagnostic programs normally used in on-line operation (see 4.2 and 4.3) are sent to the modules under test. We ought to make it clear beforehand that all electronic components, all printed circuits

and all racks are tested with automatic test equipment during the factory assembly process.

3. SOFTWARE SUPPORT FOR TANDEM EXCHANGE

A considerable effort has been made for implementing PROTEO tandem exchange software. Briefly, we can say that software already developed or under development consists of: 60 K (words of 24 bits) for operational software; 80 K (words of 24 bits) for diagnostic software; 400 K bytes for support software.

A team of about 100 programmers and software experts is currently working in SITS for the realisation of such programs. In order to put together a team of this size, interesting problems concerning organisation and training have been treated. Results and conclusions are outlined in (5).

Structured programming techniques have been used since the beginning, to ensure a high degree of productivity and "good quality" software. Furthermore, software aids have been developed in order to facilitate programmers work; these aids will be discussed in sections 3.1 and 3.2.

3.1 ISOFAC: Integrated Software Factory

In order to implement and test operational and diagnostic software for PROTEO exchange a specific software tool has been developed, namely the Integrated Software Factory (ISOFAC). ISOFAC is essentialy a program library consisting of the following:

a) Translators (Macroassembler and, in development, LPL compiler)
b) Generators (Linkage Editors)
c) Simulators for program test and validation
   i) PREDEB – used for sub-programs level test
   ii) OVERALL – used for program test, consisting of a track simulator
   iii) FLOOD – used for program set test, consisting of a dynamic simulator.
d) Program Administrators to handle PROTEO software and related documentation.

The ISOFAC structure (5,6) is shown in Fig. 2. A few Job Control Language macroinstructions were devised (for the host computer) for easy handling of program libraries. It is organised in a unique system library and there are as many private libraries as users. Each user is allowed read and write accesses to its own library but only read access to those of the others: a good information exchange is realized without losing security of data. The system library is accessible and changeable only by the Program Validation Group (see (5)).

During ISOFAC design phase two important techniques were chosen: integration and simulation. "Integration" is the implementation process wich can contain and handle all necessary modules to produce and test software is a simple and orderly way. "Simulation" is an exhaustive testing technique of operational and diagnostic software under normal as well as failure modes. The decision to use simulation techniques implied that several "linear with respect to the time" simulators had to be implemented e.g. for the CPU, for Central Control and for PROTEO environment. All those efforts are justified by the benefits derived in PROTEO software production, above all with respect to correctness, reliability and productivity. ISOFAC is now currently used on IBM 370/145 under OS/VS1.

3.2 Software aids for hardware design and diagnosis

Early and easy identification of faults is one of the major factors that contributes to continuity of service. It therefore becomes important to posses a library of fast and highly reliable diagnostic

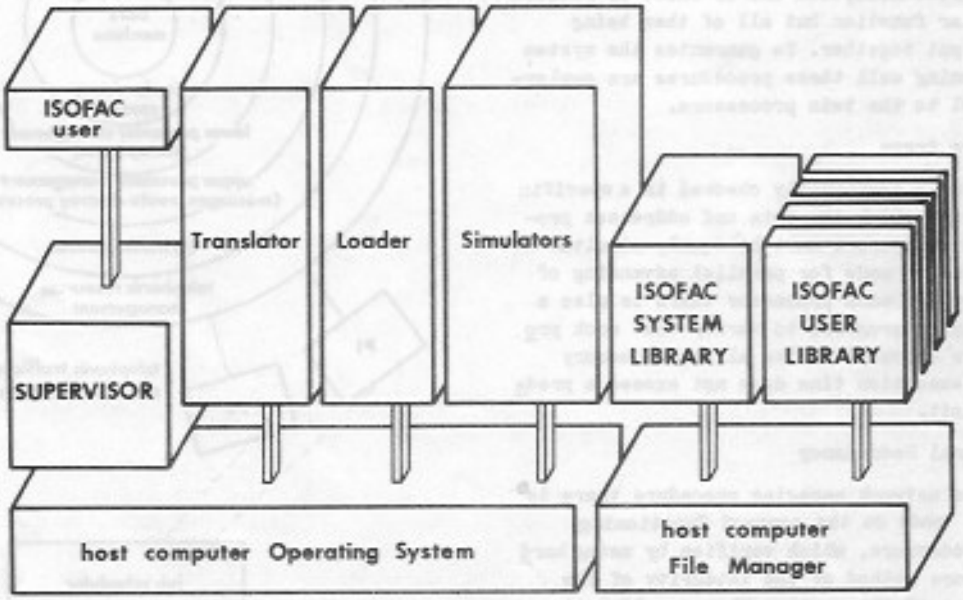

Fig. 2 – ISOFAC structure

programs. A suitable software structure should therefore be devised to facilitate diagnostic programs development and to make available an instrument which provides design guidance in order to facilitate diagnosis procedures.

A software system for the simulation of logic networks has been designed and implemented in the context of PROTEO project (7).

The three basic blocks of the simulation system are:

a) a module for structuring software model for the simulated circuit;
b) a module operating on the principle of "exclusive activity" simulation, utilising a micrologic circuits library;
c) a module for analysing and formating of data produced by simulation according to given directives.

The characteristics of the system are:

- about 20 K FORTRAN statements
- about 4000 micrologics as the maximum size of the circuit to be simulated
- a library of 80 micrologic functions with possibility of inclusion of new function. For the time being only TTL circuits are considered and include SSI components as well as MSI functions such as PROMS, ALU's, Shift-rotates etc.

4. ON-LINE SOFTWARE

We present now the software organisation for managing a tandem exchange. Particular attention is paid to the reliability of the software system obtained by using firmware traps as well as suitable functional redundancy. We also illustrate how the software helps in guaranteeing the correct functioning and eventual diagnosis of the entire network.

4.1 Functional Safeguards

Network management procedure is composed of individual simple procedures,each one of these is related to a particular function but all of them being harmoniously put together. To guarantee the system to be functioning well these procedures are employed in parallel to the twin processors.

4.1.1 Firmware traps

Each operation is continuosly checked in a specific time-slot during which the data and addresses processed by the processors must be equal; simultaneously a check is made for parallel advancing of the processors. In each processor there is also a check for program progress to verify that each program is always executed in its allocated memory zone and the execution time does not exceed a predefined time limit.

4.1.2 Functional Redundancy

Along with the network managing procedure there is also a check , made on the correct functioning of the two processors, which verifies by using hardware or software method of the integrity of the various modules of the system. When a malfunction is detected the processor responsible for the alarm is immediately excluded from network management procedures whereas the other processor carries on regardless. A quick automatic diagnostic procedure is carried out to localise the fault in the processor concerned. Once the faulty processor has been repaired it is brought back in the system by updating it so that it can continue its network management activities. An external logic module decides which one of the processors is to become the Master and henceforth is to continue in its role whether the system is complete or degraded and all points of reference is made to it. The switchover from one state to the other is checked by that part of the operating system which manages the control of the system redundancy.

4.2 Operating System and Application programs

The software structure,according to a multilevel hierarchical structure,is shown in Fig. 3 where all the functions assigned to each level is pointed out.

At redundancy level the strategy for recovery from a malfunction due to an alarm is applied.

The alarms detected consist of:

a) mismatch in network management procedure
b) hardware malfunction detected by checking circuits at the time it is verified
c) hardware malfunction detected by checking software before utilising the resources.

The two successive levels implement the basic moni-

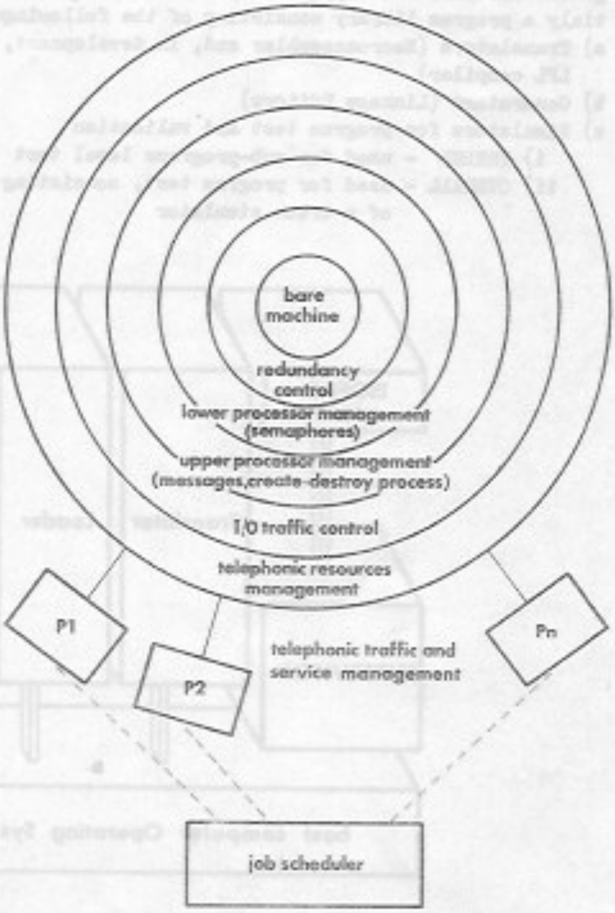
bare machine
redundancy control
lower processor management (semaphores)
upper processor management (messages,create-destroy process)
I/O traffic control
telephonic resources management
P1
P2
Pn
telephonic traffic and service management
job scheduler

Fig. 3 - Multilevel on-line software

tor of a normal real-time operating system. The first level implements the mechanisms for suspension, switch and activation of a process utilising semaphor techniques. In the second level the orderly information interchange between two processes for realising a function is implemented. The processes are arranged in ten levels of priority and is guaranteed to have a switch over time less than 80 μsec (max execution time of the Dispatcher).

The last two levels implement the strategy for processing switching peripherals comprising of remote-controlled check and diagnosis. The level also has the task of checking redundancies of the switching peripherals which are logically regarded as a single peripheral.

From the last level a virtual machine, expressly oriented to control telephone networks, is generated from which the characteristics of an exchange itself can be easily derived by specific operational software.

The operating system software is entirely realised in macroassembler language. The programs occupy about 6 K words (24 bits/word).

4.3 Software tools for on-line CC diagnosis

The first phase of on-line CC diagnosis ("check and alarm") includes hardware checks and surveillance programs. Such programs are resident in the CC and are periodically activated in order to submit to test those portions of the CC not completely covered by hardware alarms. The surveillance programs amount to about 5K instructions.

The second phase of on-line CC diagnosis ("first level") is very simple. As far as CC is concerned, the procedure involves excluding from service the processor which has shown up a fault in the previous phase; while the other processor takes over control of the network, diagnostics continue to be executed on the faulty one as shown below.

The third phase ("second level") is, in turns, split into 3 subphases, namely alarm analysis, external test, internal test.

An alarm activates the alarm analysis subphase. In this subphase the alarm registers' state word is examined in order to single out the faulty portion of the CC and to accelerate the execution of the subsequent subphases. Programs dedicated to the execution of this subphase are resident in the CC, are executed in the CC itself, and amount to about 4 K words. Therefore, the execution of this subphase is feasible only if the fault has not completely invalidated the processor. The external test subphase is activated either by an alarm analysis subphase or directly by an alarm. In this subphase the soundness of a minimum hardcore needed for setting off the internal test subphase is asserred; information derived from the alarm analysis subphase as well as the soundness of the lines interconnecting the service computer (ES) and the CC are also checked. In this subphase the console of the processor under examination is activated by the ES which executes all the necessary actions. The relevant programs are resident in ES itself: then task is to load the wanted through the console into the processor under diagnosis and to fetch, also through the console, the information needed for the completion of this subphase. Programs running in the processor under diagnosis, but normally resident the ES, are devided into 7 modules and amount to 6K instructions.

The internal test subphase is triggered on as soon as the external test subphase is completed. This subphase attempts to single out the faulty printed board; occasionally the fault may be localised on two pointed boards rather than one. Programs dedicated to this subphase are divided into modules in the ES. Only one module at a time is sent by the ES to the processor under investigation; by way of a loading program resident in a ROM, the module is loaded into the CC from ES. The relevant software is composed of 54 modules and amounts to 45K instructions: each module does not exceed 1.5K instructions.

4.4 Software tools for on-line RT diagnosis

The alarms that activate the diagnosis procedures, giving all available fault information to the CC, can be produced as follows:

a) deterministic alarms
   - by self-checking circuits;
   - by check-bits in control busses;
   - by negative output of test routines over check, alarms and master-slave switch circuits;
   - by continuity check over connections not duplicated because of previous fault.

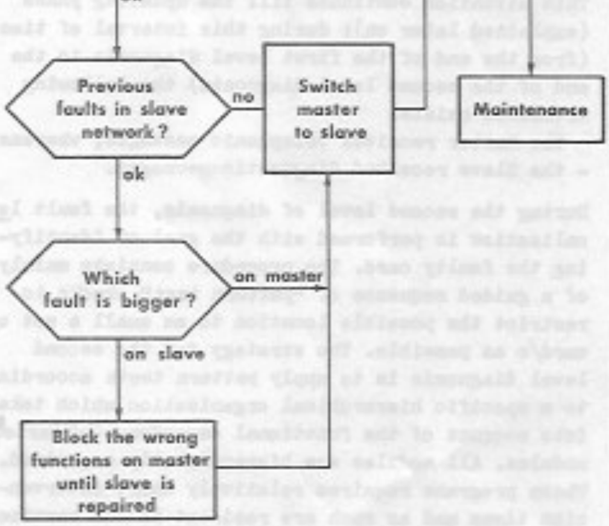
START - received alarm
Test pattern checked through slave network ?
no
2nd diagnosis level on slave network
ok
Previous faults in slave network ?
no
Switch master to slave
Maintenance
ok
Which fault is bigger ?
on master
on slave
Block the wrong functions on master until slave is repaired

Fig. 4 - RT diagnosis procedure

b) non-deterministic alarm

- by comparators at the output of the two redundant TST networks.

For deterministic alarms there is no need for any further tests because the information already available is sufficient for identifying the faulty network. Diagnosis for the non-deterministic alarms is made by software programs but needs tools to investigate and to check the network.

The fundamental tool in PROTEO Digital Switch is the test pattern circuit that checks the connections through each TST switch network. A test pattern sender sends, upon CC command, a fixed bit sequence over an input channel of a previously established connection and a test pattern receiver at the output channel raises an alarm if the bit sequence is incorrect.

The first-level diagnosis phase quickly identifies and puts the faulty network out of service also in case of the alarm being non-deterministic. A typical procedure, for an alarm raised by the output comparator, is shown in the simplified flow-chart. The comparator alarm contains also the output channel number. This information is used by the CC to localise the input channel involved in the particular connection under consideration. The result of the pattern test done between the two channels, as indicated above, allows to identify the faulty TST network.

The programs for implementing this level of diagnosis, by virtue of their necessity for quick intervention, must be resident in CC. Typical time for this procedure is 10 ms. The program dimension, written in Assembler, is about 5K instructions.

At the end of the first level diagnosis the following situation exists:

- the good TST (or the one "less" faulty) becomes the master;
- operationally it becomes a single one.

This situation continues till the updating phase (explained later on); during this interval of time (from the end of the first level diagnosis to the end of the second level diagnosis) the following situation exists:

- the Master receives telephonic messages, whereas
- the Slave received diagnostic messages.

During the second level of diagnosis, the fault localisation is performed with the goal of identifying the faulty card. The procedure consists mainly of a guided sequence of "pattern test" checks to restrict the possible location to as small a set of card/s as possible. The strategy for the second level diagnosis is to apply pattern tests according to a specific hierarchical organisation which takes into account of the functional dependence of various modules. All modules are hierarchycally organised. These programs requires relatively short intervention times and as such are resident in the service computer (ES). The program language is a high-level language of the machine concerned and memory occupation is envisaged to be around 20K words . Typical fault location time is few minutes. After the repair of the faulty block and before it is put in service again, a check on repaired block is performed.

Since the diagnosis causes the loss of the contents of the address memory in the slave TST network, it is necessary to update the addresses of all present connections. The Updating phase needs a maximum of 4 seconds.

5. CONCLUSIONS

Both CC and RT are undergoing the final stages of the test program, PROTEO system trials will start thereafter.

By "PROTEO switching area" we mean all subscribers and equipment, sparselly installed but controlled by one CC. Many functions primarily concerning with switching, management and maintenance are features of the PROTEO switching area. The Tandem exchange forms a part of the whole PROTEO switching area.

PROTEO system trials are scheduled according to the following four phases:

phase Ø : testing the availability of the Tandem Exchange will be the prime objective; CC and RT will be connected together to check the correct flow of dialogue; procedures to evaluate the efficiency of the 3 diagnostic phases will be activated.

phase 1 : a first block of switching features are tested; basically the Tandem Exchange processes the incoming traffic to local exchanges and associated signalling and routing procedures.

phase 2 : the Tandem exchange processes both incoming and outgoing traffic; in particular, routing procedures and associated charging and signalling will be tested.

phase 3 : the PROTEO switching area is completed by connecting the Tandem Exchange to the Peripheral Exchanges; CT's Control and subscriber switching features are tested.

Two identical systems will be installed,system RØ in SITS's laboratories and R1 in the Italian Public Network. Simulated traffic and network will be used in the trial phases for RØ, as soon as one phase is successfully completed, the same phase will be applied to R1 under equivalent field traffic. During phase 1 and 2, the R1 Tandem Exchange and related trunks and routing facilities are added to the public network,thus achieving an extra capability in traffic handling. Therefore R1 can be tested with live trunks and traffic without affecting public service; as a matter of fact, any exclusion from operation of the Exchange on account of modifications, improvements, tests, faults etc. will cut out only the extra traffic capability of the network. During phase 3 trials, users will be connected to the exchange but utmost care will be taken to ensure full service availability. Having this in mind operational field trials for the Peripheral Exchange (8) are into public service since April 1975.

REFERENCE

1. de Ferra F., Martinelli S. "Basic features of the time division switching system PROTEO", Int. Conf. on Comm., San Francisco, June 1975.

2. Lucantonio P. "Preliminaries to the introduction of electronically switched networks", Revue F.I.T.C.E. n° 6, 1975.

3. Valbonesi G., Formenti F., Musumeci L., Bovo A. "PROTEO System: Transit Network and Central Control". Telecomunicazioni, 1975, 54.

4. ITALTEL, "PROTEO", Special issue, No. 359-1-E01, Dec. 1975.

5. Carrelli C., Galimberti R., Rosci G., Taroni P. "The production of software for the PROTEO system", Software engineering for telecommunication switching systems,Salzburg, Feb. 1976.

6. Husu E., Martucci R., Rosci G. "Sistema PROTEO: gestione della fabbrica del software", Congresso AICA, Genoa, Italy, 1975.

7. Alleva I., Corti M.G., Galimberti R., Pescarolo F. "A simulation system for implementation and evaluation of diagnostic programs of a special processor", 12ª Design Automation conference, Boston, June 1975.

8. Arrigoni G., Di Stefano G.B., Dal Monte S., Magnolfi G. "PROTEO system: field experience of PROTEO peripheral exchange", ISS 1976.