CENTRALIZED DATA PROCESSING: AN APPROACH TO SYSTEMS DEVELOPMENT

BY

JAMES G. OLIVERIO
L. DAVID SALYER
GTE DATA SERVICES INCORPORATED

BACKGROUND

GTE Data Services Incorporated is a wholly-owned subsidiary of General Telephone & Electronics Corporation (GTE). The Company provides data processing services to GTE telephone operating subsidiaries and to independent telephone companies throughout the country. The 19 GTE domestic and international telephone companies employ some 116,000 persons and serve more than 16.6 million telephones.

Prior to the formation of GTE Data Services, GTE realized that increased demands for information within the GTE telephone companies were causing data processing costs to soar. Even short-range forecasts by the telephone companies indicated dramatic rises in data processing personnel and computer rental costs to meet their ever-increasing needs.

Consequently, GTE conducted studies to find an economical approach toward development and usage of data processing that could meet the challenges of the future. As a result of these studies, GTE Data Services was formed to:

1) Provide computer systems, programming and processing services to the GTE telephone companies.

2) Develop and implement a series of central, computer based information systems to provide all GTE telephone companies with timely data on a broad range of specialized subjects. This information enables the companies to operate more efficiently and greatly aids their management in decision-making and planning.

3) Provide the telephone industry, through our time-sharing library, a highly reliable means of solving long range and daily problems in every area of operation including finance, budgeting, marketing, engineering and revenue requirements.

ORGANIZATION

GTE Data Services employs more than 2,000 persons throughout its 12 data centers. These 12 data centers are divided into three regions, designed to efficiently meet the processing requirements of the three GTE telephone company regions.

The western region includes data centers in Los Angeles, California and Everett, Washington. The northern region contains data centers in Fort Wayne, Indiana; Bloomington, Illinois; Muskegon, Michigan; Johnstown, New York; Marion, Ohio; Erie, Pennsylvania; and Sun Prairie, Wisconsin. The southern region includes data centers in San Angelo, Texas; Durham, North Carolina; and Tampa, Florida.

MAJOR ACHIEVEMENTS

GTE Data Services was organized primarily to meet several challenges facing the data processing operations of the GTE telephone companies. These include: slowing the rapid rise of data processing costs, reducing the number and variety of computer systems in use, developing generalized computer programs for use by all telephone companies, and handling the tremendous volume of information.

The Company met these challenges head on. Costs were reduced by purchasing computer hardware, rather than leasing or renting it; by using fewer but more powerful computers; by eliminating duplication of programming efforts; and by centralizing the purchasing and control of equipment.

The Company has formed regional data centers in each part of the country, where large-scale computers are available to handle the needs of one large telephone company or several smaller companies.

Regional and divisional data centers using standardized computers and procedures, coupled with centralized systems and programming development, enable GTE Data Services to provide its customers with efficient and timely services.

SYSTEMS DEVELOPMENT

Our centralized Telephone Systems Group, comprised of some 400 data processing and telephone professionals, plays a leading role in our continuing challenge to meet the growing needs of the GTE telephone companies. Aside from developing major data processing systems for the GTE companies, an increasingly important goal of this staff is the development and implementation of a Business Information System (BIS) to assist management with the control and decision-making process while minimizing data processing costs and optimizing data handling.

BIS represents a long range concept of development, with its various systems designed to fit together to provide a comprehensive data base for telephone company management.

In addition to BIS, the development of Special Products, Interim Systems, and Time Sharing are three other key items for which we are responsible. Special Products are administrative computer systems which study equipment usage and facilities administration by interfacing the output of electronic central office switching equipment to existing management or business systems, such as customer billing.

When the developmental time factor is critical, Interim Systems are developed in a brief time frame using standard computer languages. These systems stand alone and are not necessarily dependent on BIS systems or information.

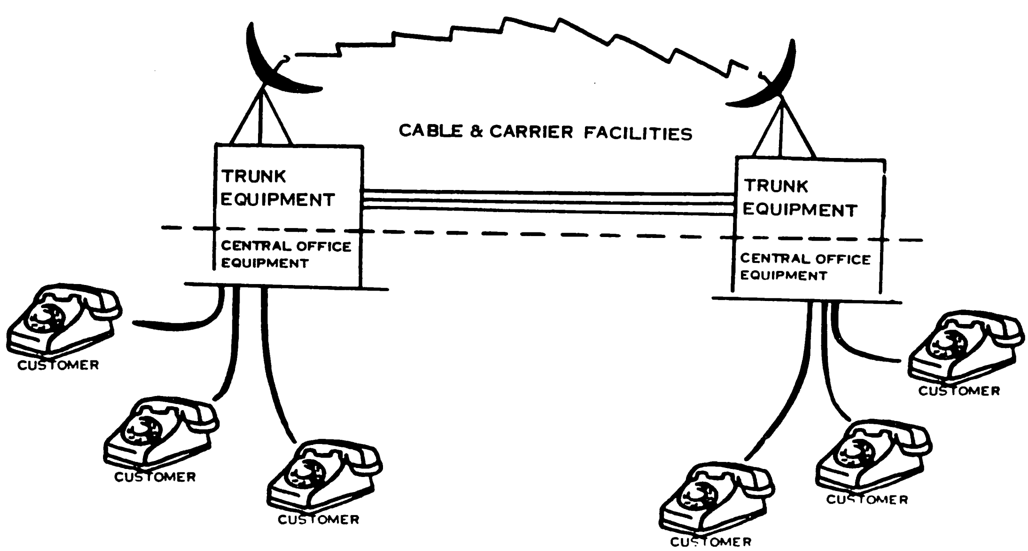
When processing response time is critical or there is a need for the user to interact with the computer in a problem-solving mode of operation, systems are developed by the Time Sharing group.

THE APPLICATION

The application that we will be discussing deals with the record keeping associated with the design, selection of the equipment and facilities (or assignment), and maintenance of Telephone Trunk networks. Customer telephones are served through individual Central Offices. When a customer calls someone connected to a different central office than his own, his call is carried via the trunk network to the receiving central office and then to the phone being called. Whenever you make a long distance telephone call or local call that must go through more than one central office, you are using a circuit that is commonly called a "Message Trunk". This "Message Trunk" is composed of items of equipment in the central offices and cable pairs or carrier channels (facilities) which go between central offices. Also, if you have any kind of telephone service other than your ordinary home or business phone, you probably have what is referred to as a "Special Service Circuit". Examples of Special Service Circuits are WATS service, private line data circuits, or direct voice circuits which bypass the telephone switching mechanisms. As with "Message Trunks", these special service circuits must use equipment and facilities to perform their prescribed function.

EXHIBIT I

TRUNK NETWORK



The following is an oversimplification of how these records are maintained manually.

1) When a circuit is to be installed, an engineer will design the circuit on a paper, specifying the central offices through which the circuit is to go and the type of equipment to be used at each central office.

2) The engineer will then forward this design to an assignment clerk who maintains work books on the facilities and equipment installed throughout the telephone company. The clerk will go through these books to find available facilities and equipment which meet the designers requirements. Once found, the clerk will write the circuit identification in the workbook next to the appropriate items, then, post the assignment information on the design form.

3) Several copies of the design form will be made and a copy sent to each location where work is to be performed in order to install the circuit as well as to various administrative locations in the company where these records are required. Often times this information will also be sent to other telephone companies which might be involved in providing the service.

4) When a circuit is removed from service, the files will have to be purged and erasures made in the work books in order to free the facilities and equipment for future use.

5) Frequently, changes will have to be made after the design forms are distributed and these changes will have to be communicated to all locations where these records are stored even though the change may not necessarily affect them.

As you can see, this creates a massive procedural problem. When you consider that an average telephone company may have 65,000 circuits, 150,000 facilities, and 200,000 items of equipment, the problem becomes acute and accuracy of the records can deteriorate to the point where they are no longer reliable.

The first step in mechanizing this application was to develop a system which would capture the manual records and point out to the users where inconsistencies existed in their records. They could then resolve these differences and return their records to a high degree of accuracy. The second step would be to provide an operational system to handle daily activity that would prevent these records from deteriorating in accuracy again.

The first step was accomplished with the Trunk Administration inventory and reconciliation system, which is a batch system that runs on an IBM 360/370 computer. This system was enhanced with a mini-computer based front end system called the Trunk Administration Record Support System (TARS). Originally written for the HP21MX, TARS has been converted to the HP3000. These systems, in tandem, provide an economical and relatively fast method of collecting, validating, and reconciling the manual records. Five telephone companies currently are using the Trunk Administration System and three of those companies are also using TARS. Additionally, we have requests to install both TA and TARS in two more companies.

The second step of mechanization, that of providing an operational system, is primarily what we are here to talk about today. The way we established the requirements, developed the system, and implemented it on the Hewlett Packard 3000 will be discussed in some detail.

ESTABLISHING THE REQUIREMENTS

The requirements for the operational system were written by a "requirements team". The team was composed of representatives from six telephone companies, GTE Service Corporation, and GTE Data Services. This team was chartered to document the requirements for an operational system, recognizing that each company had many differing procedures and requirements. The requirements were to address those features that were common to all companies and make provisions for enabling each company to meet its own unique requirements.

The requirements team produced a "Requirements Document" (RQD) for the Circuit Network Administration System (CNAS) which was submitted to GTEDS for evaluation. GTEDS responded by proposing a modular development approach utilizing mini-computer technology and networking capabilities. In formulating its proposal, GTEDS compared the alternatives of putting this system on line to an IBM/370 or using mini-computers.

EXHIBIT II

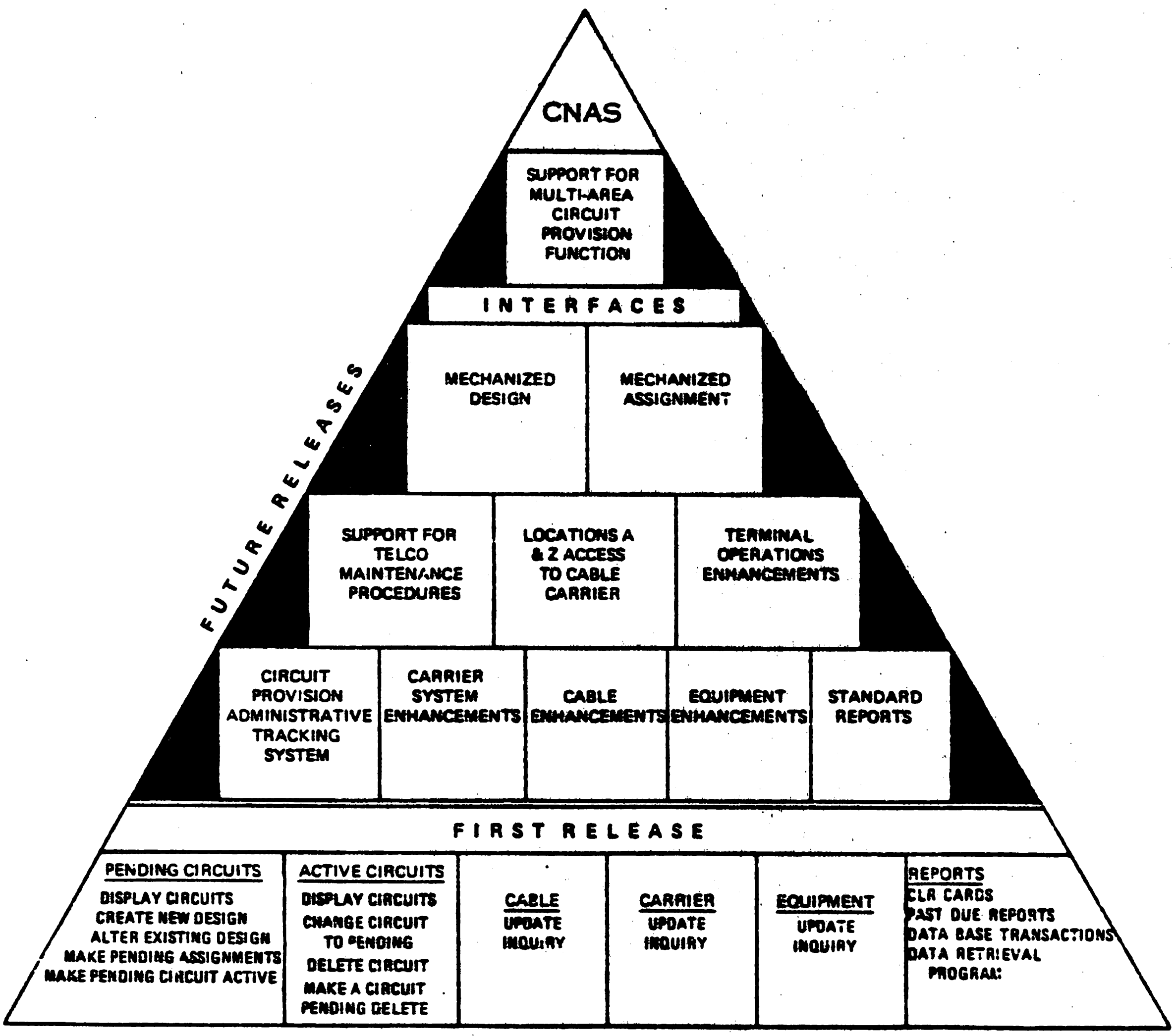


FIGURE 1 – NEW TRUNK ADMINISTRATION SYSTEM MODULAR STRUCTURE

Before doing the economic study, it was necessary to determine if the HP3000 had the capability of handling a system of this size (i.e. the largest module). To make this determination, a data base was designed, volumes established, and scenarios written for typical system activity. These scenarios were described in the form of "scripts" for the IDEA processor. We then made several runs of IDEA varying the amount of activity against the data base. The results of these runs showed that the HP3000 had the capability of handling the transaction volumes required by the system and that with DS3000 the modular approach could be implemented in a logical manner.

The economic evaluation, using five telephone companies as the basis, showed that processing the system on an IBM computer was much more expensive than processing on an HP3000 even when you considered the sharing of the IBM resources with other systems. This information was the primary basis for the GTEDS proposal.

SYSTEMS DEVELOPMENT

The proposal was accepted and authorization for development of the first module of the Circuit Network Administration System (CNAS) was given on July 29, 1977. A staff of eight Data Processing professionals was budgeted for this project. The project was organized into three functional areas headed by a Systems Supervisor;

1) A "programming" team headed by a chief programmer. The team was responsible for system design, programming, documentation, and testing of the system.

2) System Management - This area consisted of the HP3000 system manager and a general clerk. They were responsible for the HP3000 operation, data entry, and utility software requested by the programming team.

3) Conversion - Initially, this area consisted of one person and was later expanded to three. The responsibility of this area was to develop the methodology and programs to convert the data base from the Trunk Adminstration System to CNAS.

The "programming" team designed the system using structured techniques. Included in these techniques were:

1) Tops down design

2) Hierarchical Input Process and Output (HIPO) charts

3) Structured walk-thrus, and

4) The use of a librarian

During the design phase, the data entry philosophy was established, data element definitions were entered into a data dictionary system, and the data base schema was established.

Hardware requirements for each of the telephone companies were developed and it was determined that each company would require between three and six 120 mb. disc drives depending on their size, 512 kb. main memory, and a tape drive.

When the detailed design was completed, a System Specification report was prepared and distributed to the requirements team. In addition, simulated demonstrations of the system were prepared on cassettes for the HP2645 terminal. The requirements team was then called together and they were given a demonstration of the system on the terminal. By focusing a television camera on the terminal and displaying the video on a seven foot television screen, they were able to get a good understanding of how the system was designed and would work in a live environment.

As a result of this presentation, the requirements team was able to identify a number of changes or additions that they wanted in the system. It was relatively easy to incorporate the requested changes, since, up to this point, no programming had taken place.

Once the requested design changes were made, program development began in earnest. Detailed program specifications were written; again, using the "tops down" technique. Programs were oriented around user functions and thus, were modular almost as a by-product of the development method instead of by design. Program "mainlines" which incorporated several functions were developed. By using this "mainline"/"subprogram" technique for the on-line programs, maximum flexibility for system tuning purposes was maintained.

Program functions which were going to either require a significant amount of time to execute or utilize a lot of the system resources, were designed to be initiated by on-line transactions and then processed in a batch mode.

The first release in its final form contained approximately 90,000 lines of code. Of that total, 85,000 were COBOL and 5,000 were SPL. The data base schema contains 177 elements and 64 data sets. There are 67 discrete transactions in the on-line portion of the system and eight batch processes. Exhibit III on the following three pages is a representation of the data base structure that exists today.

In developing a system of this size on a mini-computer, as you might expect, many interesting problems developed. Some of the constraints of the HP that we continually had to deal with were:

Problem:	Maximum Code Segment Size
Constraint:	System doesn't allow for a code segment to be larger than what is specified in system generation parameter. The maximum size specified in our system is 16,384 bytes.
Resolution:	Used COBOL sections. Attempted to establish uniform section sizes. Also, frequency of execution and logically related process considerations were applied in establishing the content of a section.
Problem:	Maximum number of segments per program.
Constraint:	System allows only 63 code segments per program. Each COBOL program generates an initialization segment.
Resolution:	Established a mainline subprogram concept which allowed us to move subprograms from mainline to mainline. First consideration was to group programs with logically similar functions. Second consideration was the aggregate number of segments in the program. Where necessary, subprograms could be in more than one mainline.
Problem:	COBOL initialization segment size
Constraint:	If you have a large number of value clauses in your data division, the code generated for the initialization segment can exceed the maximum code segment size.
Resolution:	Initialize data areas from within the programs procedure division.

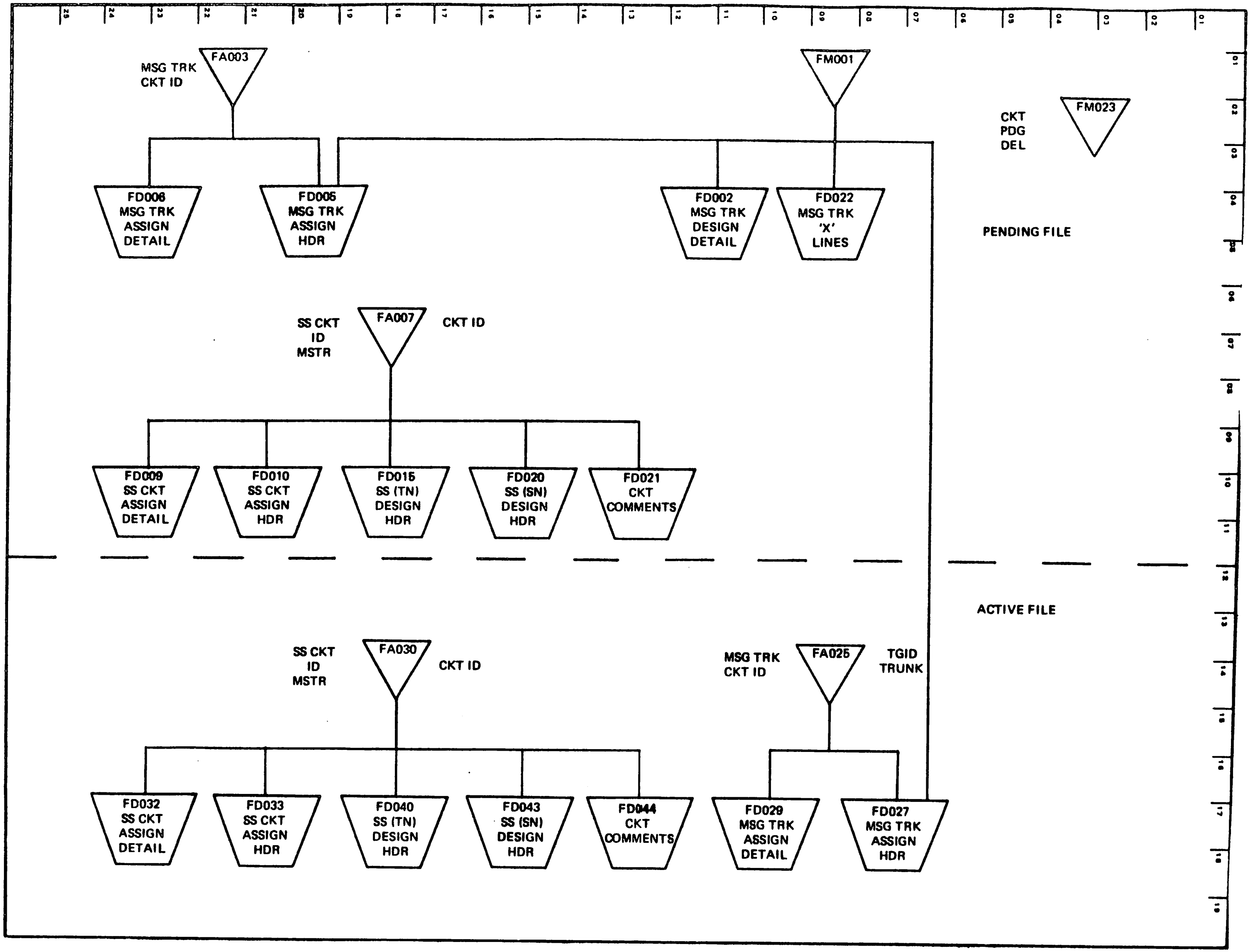


EXHIBIT III - I

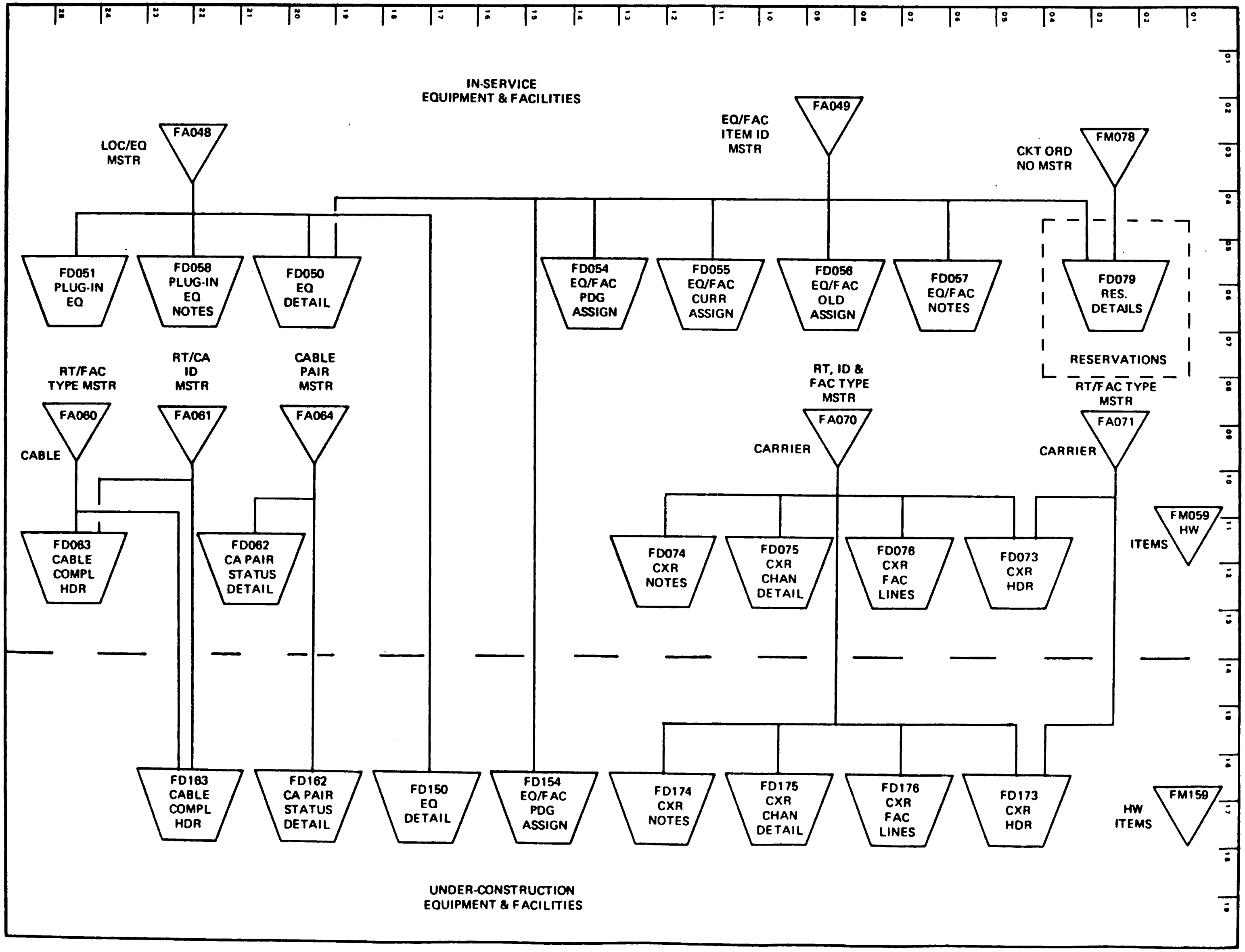


EXHIBIT III -2

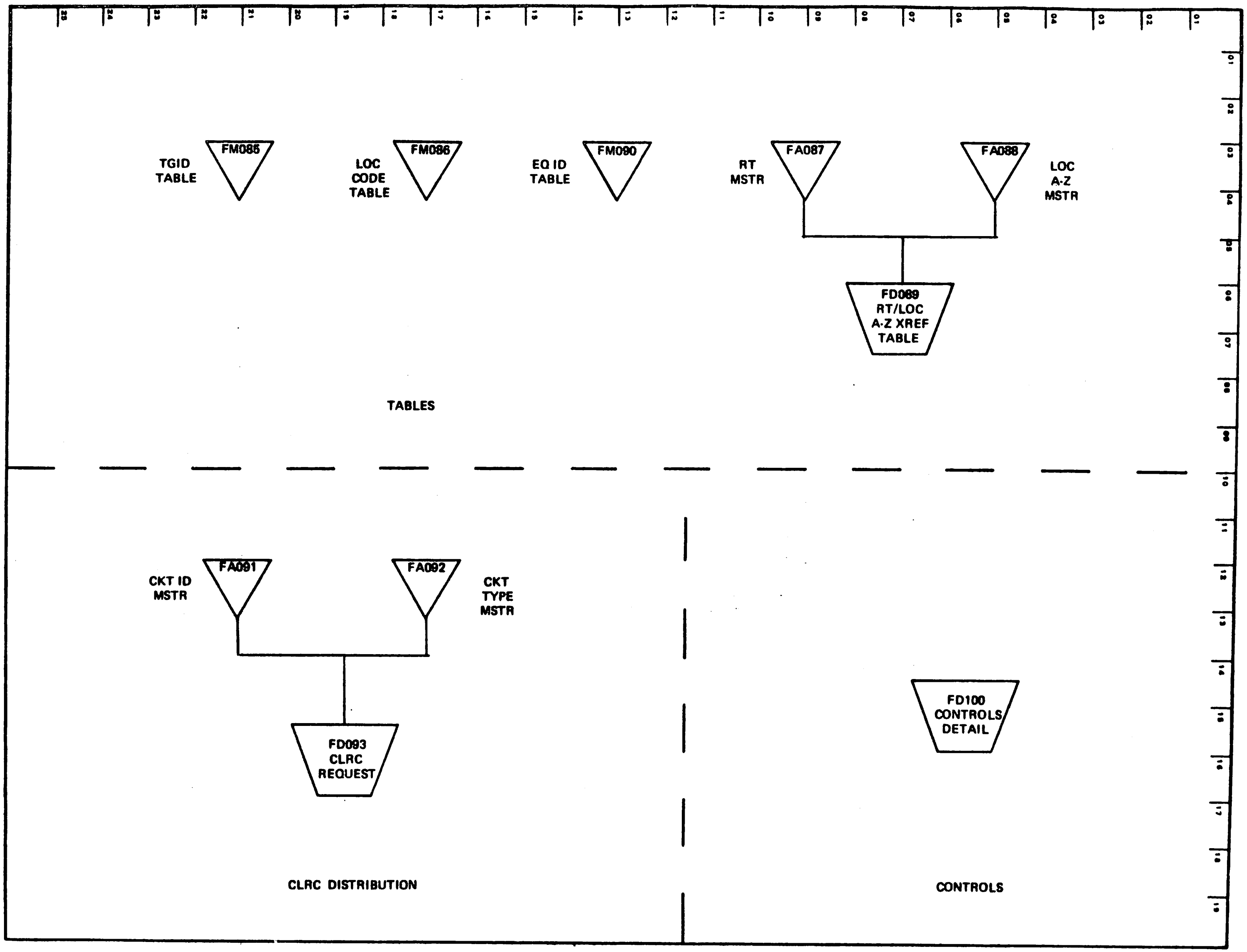


EXHIBIT III - 3

Problem:	Maximum number of shared files open at one time.
Constraint:	Up until the release of the 1906 MIT, only 63 shared files could be open at one time.
Resolution:	This problem was brought to the attention of the HP Lab. We pointed out that a shared data base could contain 99 data sets but that only 63 of them could be open at any one time. This problem was resolved by the lab before it became a serious problem in a production environment.
Problem:	Maximum number of data sets in a data base
Constraint:	The maximum number of data sets in a data base is 99.
Resolution:	Several attempts at data base design ended up with 75-85 data sets. We felt that this left too few data sets for future enhancements. Therefore, we included the number of data sets as a criteria for design. Other criteria used was: 1) System performance 2) Disc requirement 3) Query capability
Problem:	Maximum number of entries in a chain.
Constraint:	IMAGE only allows 64,000 items in a chain. We had one situation where a longer chain length would have been very beneficial for some report retrieval requirements.
Resolution:	A good resolution of this problem was not found. A partial solution was to separate certain records into different data sets.
Problem:	Data Base Locking Schemes
Constraint:	Item level locking, as it was implemented, presented us with a difficult situation because of our requirement to change the locking table several times during the processing of a single transaction.
Resolution:	This problem is still under review. There does not appear to be a foolproof way of implementing item level locking. The best we can hope for is a method which has the least probability of creating a problem and the problem must have a minimal consequence to the user. Locking at higher levels, such as data set or data base, will obviously create an intolerable performance problem.

Problem:	System Back-up Methods
Constraint:	DB stores and DB unloads can be very time consuming for a data base this size, particularly when the back-up medium is tape.
Resolution:	Each system will be configured with an extra disc drive. This drive will be used as a serial disc for back-up and as a private volume for logging.

Problem:	Lack of multiple data set retrieval capabilities from Query.
Constraint:	Query can only access a single detail data set in a find command. With the relationships that exist among data elements in the various data sets, it is often necessary to access several data sets for reporting purposes.
Resolution:	We are currently reviewing software packages from vendors other than HP to satisfy this requirement.

Problem:	Inability to group data elements without losing their element identity.
Constraint:	Sometimes data elements need to be combined to form an IMAGE item. In particular, in order to construct certain paths into data sets this capability was required.
Resolution:	For IMAGE purposes, we combined the elements. If one or more of the elements could be identified as a requirement for Query purposes, then it was repeated in the data set as a single element.

Despite the problems mentioned above, the productivity rate was nearly three times the company average for a large development project. We think that this is attributable to three things:

1) The team concept and its associated techniques.
2) The fact that we had a computer dedicated for use by the development team, and
3) The reliability of HP hardware and software.

IMPLEMENTATION

Once the system was tested by the project team, it was installed in a "pilot" company for a production test. This activity began with two weeks of detailed training on the use of the system. Also, the data processing people who will be involved in providing local support were given a week of training in the DP aspects of the system. The documentation of the system consists of a system design manual (7 volumes) containing program specifications and related documentation and a users manual (3 volumes) giving specific instructions on how to use the system.

Concurrent with the training, the conversion programs were being run to convert the data base from the IBM system to the Hewlett Packard system. Only a portion of the data base was loaded for pilot purposes. This consisted of 13,000 circuits, 50,000 items of equipment, and 30,000 facilities.

During the conversion, we had verified what we had expected. Namely, that loading a large IMAGE data base is a very time consuming operation.

The cumulative processing time for the conversion was 77 hours. This time includes the periodic back-ups that were taken between jobs. Since this was a pilot area, these times were acceptable. However, when projected for converting a full company, these times would be too long. Therefore, the project team is looking at ways to improve the time. Hewlett Packard has been contracted to do a performance analysis on the conversion programs. Also, some changes in the design of the conversion method are being contemplated. We hope to obtain about a 50% improvement in the performance of the conversion programs.

At the time of this writing, the pilot activity is continuing. Very few technical problems have been identified (only one program bug) at this point. There are some administrative problems with the system that have been identified and these are being addressed as high priority enhancements for the system.

During the pilot activity, the requirements team was again called in to observe the system in operation. They then provided additional input for determining the priorities for future modules of the system.

THE FUTURE

The current plans for the system are to continue developing new modules for the system. The various telephone companies will implement the version of the system that is current when they are ready for the system, while other companies are choosing to wait until certain features that they consider as essential are available. The sequence in which the various modules will be added will be determined by an executive committee based on cost/benefit statements that are being prepared for each feature. At this point in time, ten companies have requested installation of the system.

For support of the system, we intend to have a communications capability (DS3000 on a switched network) which will enable us to access any user system for the purpose of problem determination, error corrections, and program changes. When this capability is in place, we will be able to provide very efficient service from a highly trained group. This system has many more opportunities to evolve and it should evolve in a way that is consistent with evolving technology.

CONCLUSION

This paper has touched on many aspects of system development. To pursue any of these aspects in any greater detail would require a document far beyond the scope of this paper. We feel that our methods are very traditional, but what makes this situation unique, is the way that these traditional methods were applied in today's technology.

If any reader of this paper wishes to discuss any aspect of this effort in greater detail, please feel free to contact us.