A COMPUTERIZED INFORMATION SYSTEM FOR PATHOLOGY

L. HERCZ

M.Eng. thesis March 1974

BMEU 1974-5

A COMPUTERIZED INFORMATION SYSTEM FOR PATHOLOGY

A COMPUTERIZED INFORMATION SYSTEM FOR PATHOLOGY

L. Hercz

A thesis submitted to the Faculty of

Graduate Studies and Research

in partial fulfillment of the requirements

for the degree of

Master of Engineering

Department of Electrical Engineering

McGill University

Montreal, Canada

March, 1974

ABSTRACT

This thesis describes a Computerized Information System for Pathology (CISP). The system is intended to process, store, retrieve and correlate information contained in pathology reports. Before the present system was designed, previously existing pathology information systems were reviewed and from this base the present approach has evolved.

Our aim is to provide the capability of dealing with a variety of correlative data retrieval procedures and it seems that a system based on a large hierarchical dictionary is most appropriate. Both ICDA (International Classification of Diseases (Adapted)) and the Topography section of SNOP (Systematized Nomenclature of Pathology) are used in forming the CISP's dictionary. Current computer technology makes the storage of such dictionaries feasible

The objectives set forth by pathologists were translated into a set of practical specifications for CISP. The formulation of these specifications was guided by: (a) our design approach which is based on a large hierarchical dictionary, (b) the information flow environment which the pathology system must satisfy and, (c) the constraints imposed by the available computer systems on the McGill University Campus. Following this procedure, CISP was programmed and its data base, consisting of the system's dictionary and the collection of coded report files, was stored.

In the operational prototype system the report is entered in text form and is coded utilizing the system's dictionary. In the coded form, any diagnostic item in the report may be directly retrieved. The entire report may also be retrieved in a decoded text form which is tailored to be suitable for clinical use

The system is at present undergoing operational trials in the Montreal General Hospital.

RESUME

Cette thèse décrit un système de traitement d'informations par ordinateur en pathologie (STIOP) qui enmagasine, extrait et met en corrélation les protocoles d'anatomies pathologique. La conception de ce système fut précédée par une revue de l'évolution historique des systèmes conçus et réalisés dans le passé. Nous avons conclu que plusieur approches de conception étaient possibles. Notre but etant de realiser un système capable d'exécuter une variété de procédés de corrélation, nous avons decidé de concevoir un système basé sur un dictionnaire hiérarchique extensif. Le dictionnaire ICDA (International Classification of Diseases - Adapted) et la section de Topographie de SNOP (Systematized Nomenclature of Pathology) ont été employés dans la formation du dictionnaire de STIOP. L'enmagasine de tels dictionnaire est possible à l'heure actuelle grâce aux toutes dernières techniques en informatique.

Les objectifs exposés par les pathologistes ont été traduits dans une serie de spécifications pratiques pour STIOP. La formulation de ces spécifications fut guidée par: (a) notre approaches de conception qui est basée sur un dictionnaire hiérarchique extensif, (b) les filières que doit suivre le cheminement des informations en pathologie et, (c) les contraintes imposées par le système du centre d'informatique de l'université McGill. Après cette phase, STIOP a été programmé et la rassemblement des données, se composant du dictionnaire du système et la collection des dossiers de protocoles codés, a été enmagasiné.

Dans ce système prototype en fonctionnement le protocole est entré dans une forme narrative et il est codé utilisant le dictionnaire du système. Dans sa forme codée, n'importe quel élément diagnostique du protocole peut être extrait directement. Le protocole complet peut être récupéré dans une forme narrative decodée qui est approprié a l'utilisation clinique.

Actuellement le system est experimenté au Montreal General Hospital.

ACKNOWLEDGEMENTS

The work reported in this thesis was carried out under the guidance of Professor Charles A Laszlo, Acting Director of the Biomedical Engineering Unit, McGill University and Dr. Michael Reesal, Assistant Professor, Department of Pathology, McGill University. I wish to express my gratitude to Dr. Laszlo, my supervisor, for the invaluable advice and encouragement he offered during the execution of this project. Dr. Reesal introduced me to pathology and pathologists. I am indebted to Dr. Reesal for his efforts in preparing the dictionaries and the pathology reports for storage in the computer and for the valuable criticism he offered during the writing of this thesis.

I wish to acknowledge the assistance of Dr. W.H. Mathews, former Head of the Department of Pathology at the Montreal General Hospital, who was responsible for instituting the Termatrex system and who made it possible to initiate within his department the present study for a computerized system. Since Dr. Mathews' departure, I have received every cooperation and assistance from the present Head, Dr. W.P. Duguid. I thank the members of the Department of Pathology and Administration of the Montreal General Hospital and the Department of Pathology of the Reddy Memorial Hospital for their understanding and consideration. And in particular, I am grateful to Dr. R. Abbott for his efforts spent on preparing the system's

dictionaries and Ms. C. Grant for demonstrating the Termatrex system and providing me with pathology report samples.

The assistance received from the members of the McGill University Computing Centre has facilitated the programming of the system described in this thesis. Professor A.M. Valenti has advised me on numerous occasions on matters of data base storage and brought to my attention the "round-robin" method used in copying-and-extensions of data sets. At one time or another, I received advice from most programmers-on-duty, especially David McCaffrey and Alex Cameron.

I am especially grateful to

Eleanor Bedford who typed this thesis,

Serge Lafontaine who assisted me in translating the abstract, and

Garry Bernstein and Gavril Herez for proof reading the thesis on very short notice.

I especially wish to thank my wife for the many months of expert work that she spent on producing the flowchart diagrams for this thesis, and my parents for their encouragement and support during these many years of study.

This work was supported by the Montreal General Hospital.

Table of Contents

LIST OF ABBREVIATIONS

CD	Clinical Diagnosis
CISP	Computerized Information System for Pathology
CN	Clinical Notes (symptoms)
CPU	Central Processing Unit
CRJE	Conversational Remote Job Entry
HIS	Hospital Information System
ICDA	International Classification of Diseases (Adapted)
ID	Patient Identification
IS	Indexed-Sequential
JCL	Job Control Language
MGH	Montreal General Hospital
MUCC	McGill University Computing Centre
NT	Note
OP	Operative Procedures
OS	Operating System
PD	Pathology Diagnosis
PIS	Pathology Information System
PL/I	Programming Language/One
SNOP	Systematized Nomenclature of Pathology
SP	Specimen Description
TSO	Time Sharing Option

See also Tables 6.2,6.3,6.4 and 6.5

Chapter 1

INTRODUCTION

For more than a decade now, computers have been playing a significant role in the delivery of medical care. Still, the impact of computers has not been felt as radically in medicine as it has been in other areas notably in science, business, and engineering. Intensified research, system analysis and engineering is needed to reach the stage where the potential of computers will be readily translated into applications yielding improved treatment of the patient.

Nevertheless there are many areas of medical care where computers have been successfully used. For example:

- In clinical laboratories the computer is employed to accummulate the various test results and to print the updated test report for each patient;

- In intensive care units the physiological functions of critically ill patients are monitored by computerized systems which issue warnings whenever abnormal parameter variations are detected;

- The computer was found to be very promising in automating some patient screening methods;

- Computerized scheduling of operating rooms enables a more efficient use of these facilities; and

- The traditional use of computers in billing and accounting has been successfully extended to the hospital.

During the treatment of any one patient, data are accummulated which can be of great use later in the life of the same patient, in research and teaching. While a summary of this data forms the patient's chart (or patient's medical record), more specific data about the patient's illness are gathered in various documents generated by various hospital departments. Such departments are typically clinical chemistry, radiology, pathology and others concerned with the application of laboratory methods to diagnosis. Thus exhaustive study of information generated from even one patient entails a significant amount of document collection and cooperation among departments. This need, together with the difficulty of analysing these documents, discourages interested hospital personnel from undertaking analyses which would require collections of documents on a large scale, from several departments, and possibly from several hospitals. In the past 15 years several attempts were made to alleviate at least part of this problem through the use of computers.

The objective of the project described in this thesis is to automate the information handling procedures in the Department of Pathology of the Montreal General Hospital (MGH). This automation was to cause no interruption in the operation of the Department of

Pathology. The primary resources available to us in this process consisted of the data base existing in the Department of Pathology and any readily available information processing facilities.

In Chapter 2, the role of pathology in treatment, the pathology report and the pathology information system are discussed in general. Termatrex, the present mechanized system at the Montreal General Hospital is also described. Chapter 3 deals with medical coding schemes used in pathology. The subject of choosing a coding scheme for a pathology system is introduced briefly.

Chapter 4 focuses on the evolution of computerized pathology information systems. This discussion is illustrated with specific system descriptions. The basis for the design of our system is also explored. In Chapter 5 the design concepts of the new computerized system, called CISP, are described. The objectives specified by the pathologists are interpreted to specify the basic components of the system.

The storage of the dictionaries and the coded reports and the programming of CISP are described in Chapter 6. The results obtained with the new system are shown in Chapter 7. In the same chapter an evaluation of the system is presented followed by recommendations for improvements and extensions.

Chapter 2

PATHOLOGY, PATHOLOGY REPORT AND PATHOLOGY INFORMATION SYSTEM

2.1 Introduction

The material in this chapter is intended primarily for the non medical reader, to provide a brief survey of the field and to focus on the problem of this study in its setting. The role of the speciality of pathology in patient treatment is reviewed and the nature of the data accumulated by the pathologist is examined.

The pathology report itself and the data base and structure of the information system which handles these reports are important aspects of this analysis.

Terminology specific to the pathology report and to the pathology information system as used throughout this thesis is defined in this chapter.

Finally a description of the present mechanized system existing in the Montreal General Hospital for coding, storing and retrieving pathology reports is given.

2.2 The Role of Pathology in the Treatment of the Patient

Pathology is that speciality of medicine which studies the nature of diseases, their causes and manifestations. The role

of pathology in the treatment of a patient is illustrated in Fig. 2.1.

Upon admission to a hospital the patient is examined by a clinician. From an evaluation of the patient's complaints, assessment of alterations in his physical state and a review of the patient's past history, the clinician often formulates a clinical diagnosis. Based on this clinical diagnosis treatment may be initiated and/or further tests ordered. If a biopsy is taken or the patient undergoes corrective surgery, tissue removed during the operation is sent to the pathologist for analysis. After both gross and specialized microscopic study, the pathologist arrives at a pathological diagnosis which often provides a specific diagnosis of the disease process. During his analysis, the pathologist makes use of the tissue submitted to him, the available clinical information, and any previous diagnostic records that may bear relevance to the case, before he arrives at a diagnosis. (Fig. 2.2).

The pathological diagnosis may be coupled with recommendations by the pathologist to assist in further clinical management.

2.3 The Pathology Report

The findings of the pathologist are summarized in the pathology report. Such reports serve in the study of disease and as references for the pathologist in future cases. There are three

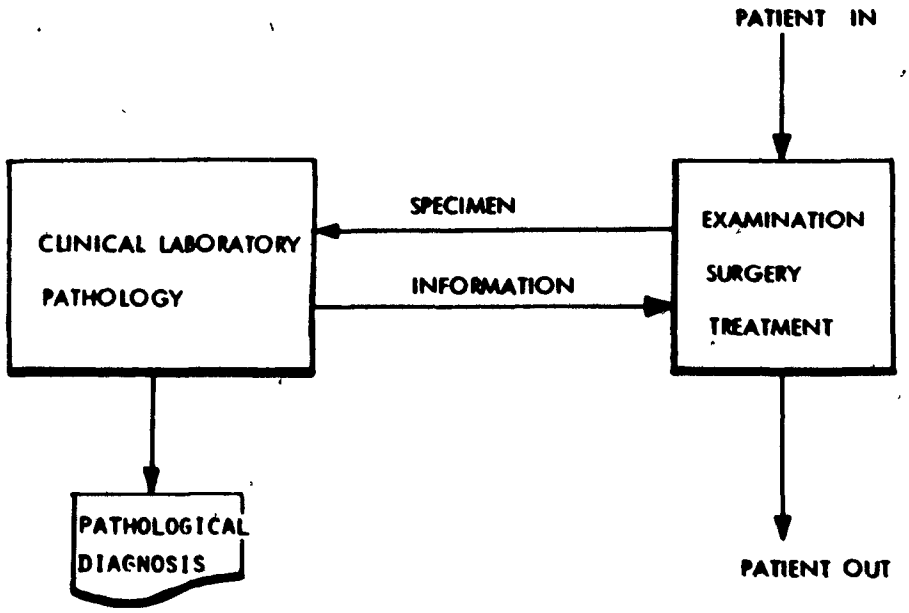


Fig. 2.1 - The relationship between Pathology and the hospitalized patient.

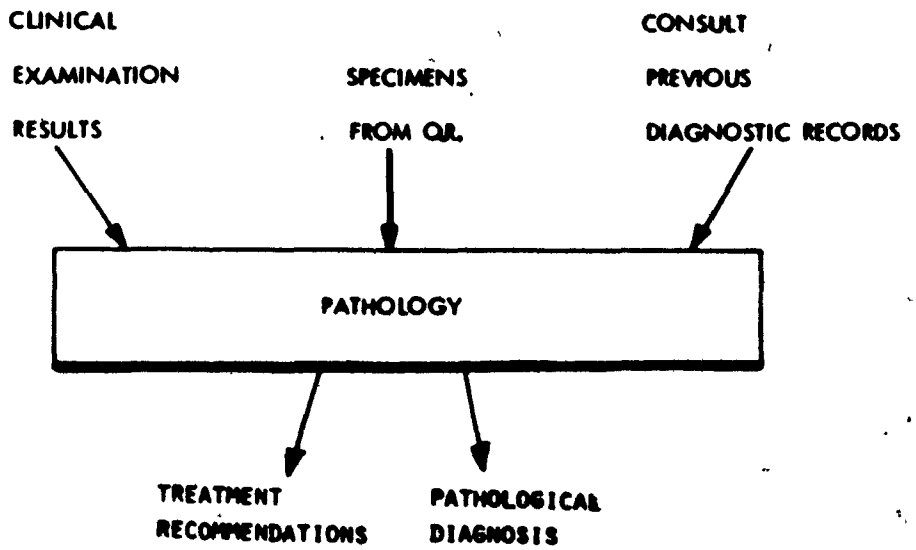


Fig. 2.2 - Major sources of material and information input and output. O.R. = Operating Room.

types of pathology reports.

The autopsy report is a report compiled on the basis of dissection of the body after death and the gross and microscopic examination of organs and tissues.

The cytology report summarizes the study of cells exfoliated or scraped from organs or tissues. Sputum, for example, will contain cells exfoliated from the lungs.

The surgical pathology report summarizes findings obtained from analysis of tissues removed by biopsy or during surgery. This is the only type of report discussed in this thesis and hence surgical pathology report should be assumed whenever "pathology report" is stated.

It is convenient to divide the surgical pathology report (Fig. 2.3) into 6 sections. The first section of the report is the Patient Identification (ID) section containing a variety of demographic data. The second and third sections consist of the clinician's observations. They include the Clinical Notes (CN), or symptoms, and the Clinical Diagnosis (CD). The OP (for Operative Procedure) section of the report describes the surgical procedure used (not shown in Fig. 2.3) and the surgical diagnosis. The specimen sent to the pathological laboratory, and the sections made from this specimen are described in the SP (for Specimen) section. The last section contains the Pathology Diagnosis (PD). In this particular report, the PD section also contains the operative procedure "total gastrectomy". Such inconsistencies occur

THE MONTREAL GENERAL HOSPITAL
DEPARTMENT OF PATHOLOGY

Name: Sickman ,Mr. John Room No.: 923 Unit No.: 388786
Sex: male Age: 89 Service Dr.: PUB Service No.: S.71-6147
Date received: June 7/71

CN: weight loss,anemia . Positive gastric washing for CA -confirmed with gastroscopy.

CD : CA stomach

Op. Findings: same

Specimen: stomach(total),greater omentum ,spleen and mesenteric nodes.

Specimen consists of the stomach, duodenum and spleen. The stomach includes the proximal portion of the duodenum and measures altogether 24 cm. along its greater curvature and 16 cm. along the lesser curvature . On opening there is a very firm irregular raised area on the mucosal surface of the stomach along the lesser curvature and measures 10x6cm. in diam. The raised irregular mass is found to be directly infiltrating the muscle coat and reaches up to the serosal surface.... ....etc.....etc.... ....

Sections: 2 rep sections sub. across this area extending up to the serosal coat and labelled as no.1 .
1 rep.section subm. from prox. resection margin as no. 2 .
1 from distal resection margin as no.3 .
2 lymph nodes submitted as no. 4 .

The spleen is attached to the mesentery and weighs altogether 150 gms. , and measures 10 x 8 x 5.4 cm. in dim. The dark red homogeneous cut .....etc....etc.....etc.

No enlarged lymph nodes could be palpated in the attached mesentery .

1 rep. section across the mesentery sub. as no. 6 .

DIAGNOSIS:

1. TOTAL GASTRECTOMY: WELL DIFFERENTIATED ADENOCARCINOMA OF THE STOMACH INFILTRATING THE SEROSA . TWO REGIONAL LYMPH NODES AND THE RESECTION MARGINS ARE FREE OF TUMOR .
2. SPLEEN: PASSIVE CONGESTION .
3. OMENTUM : HYPEREMIA.

Code 1

Date: June 8/71 Pathologist: .....
Surgical Pathology

Fig. 2.3 - An example of a pathology report.

frequently in the formulation of a report.

Although one is presumably dealing with one disease process, several diagnostic statements -- clinical, surgical, pathological may be made. The reason for this is that various diagnostic approaches often result in the expression of several opinions and not infrequently the same disease process may be differently described by physicians representing allied interests. For example, a clinician often specifices a CD (Clinical Diagnosis) based on the physical examination of the patient, chemical laboratory tests and health history. The clinician is interested in diagnosis mainly as a specific guide to treatment. On the other hand the pathologist is interested in categorizing the nature of the underlying disease process.

The pathological diagnosis, based on laboratory tests and tissue analysis, tends to be an objective finding based on "hard" data. It is not available in every patient but when a diagnosis is rendered by pathology, the nature of these findings are vitally important in any future review of the patient's medical outlook.

2.4 The Pathology Information System (PIS)

Some of the uses of the pathology report have already been mentioned. Other uses include the utilization of the report as a document of communication between departments, and with

clinicians. Upon repeated return of a patient to the hospital, the pathology report can also serve as a precise record of health history. In order to provide these services the Pathology Department must file its reports and create patient and disease indexes. The Pathology Information System (PIS) consists of the totality of reports of a pathology department together with all the facilities, manual or automatic, for the storage, recall and analysis of the reports. The communication channels that connect Pathology to other hospital departments are also part of this system.

These information channels, together with the internal structure of files, documents and their retrieval mechanisms are shown in Fig. 2.4. This information flow diagram shows the environment whose information needs the PIS should satisfy. Fig. 2.5 depicts the major classes of information which are based on the same data set but differ in the way the information is extracted, processed and aggregated and in the way the information is used.

In discussing Pathology Information Systems, we use the terminology that is generally used in describing information systems. Thus the pathology report becomes a pathology record (or just a record) which consists of a number of data items. For example, the term "age" in the demographic section of the report (Fig. 2.3) is a data item. Each of the diagnostic statements within the CN, CD, OP, and PD sections contain either single or multiple data items. The SP section, however, always

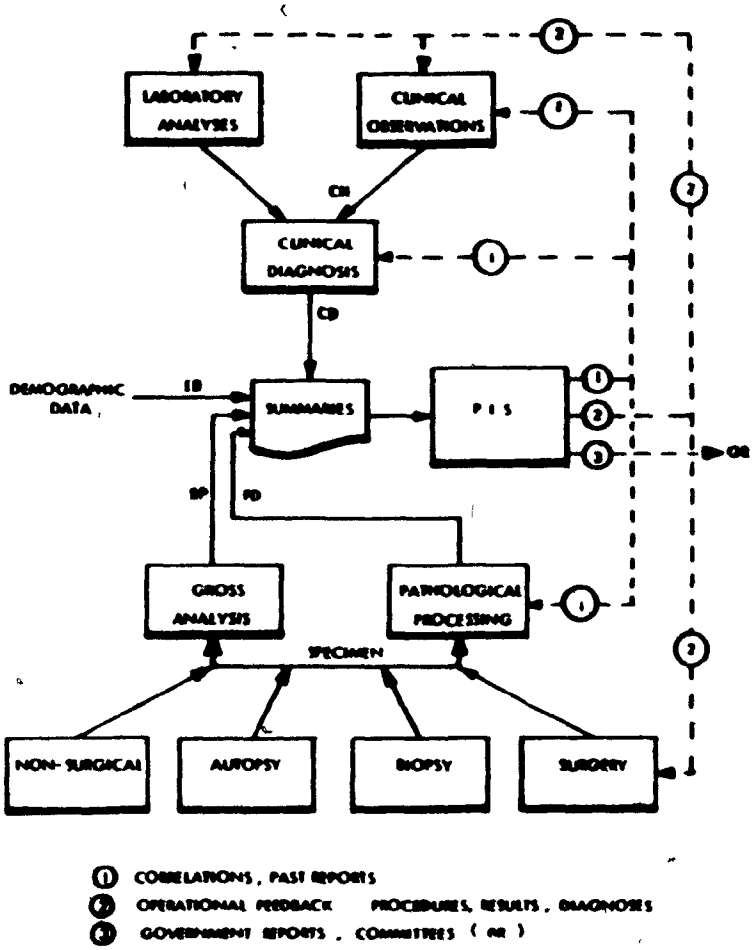


Fig. 2.4 - The information flow environment of the PIS

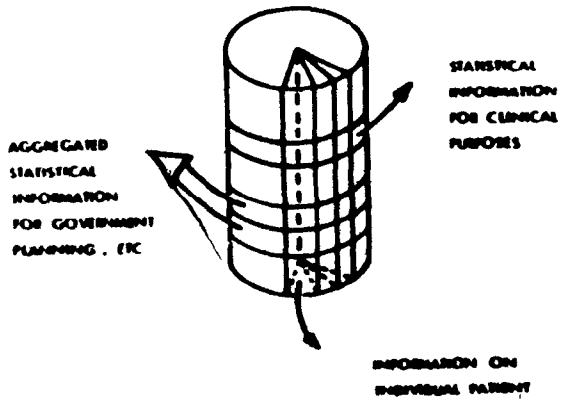


Fig. 2.5 - Data base utilization subdivided by major classes.

consists of a single data item.

A collection of records is called a file. Thus one speaks of an autopsy file, or of a file of all records of 1969, or of the file of neuropathology microscopic slides. The collection of files of the pathology department forms the data base of the PIS.

The PIS is only one of the information systems that can exist in a hospital. There may also exist a Radiology Information System (RIS), etc. All of these disjointed information systems may be interconnected to form a Hospital Information System (HIS) which assures a smooth information flow.

2.5 Present Pathology Information Processing at the Montreal General Hospital (Reesal et al., 1970)

The Montreal General Hospital (MGH) is a McGill University affiliated teaching hospital. With its 1,000 bed capacity it is considered a large size hospital. The pathology department generates 16,000 surgical pathology reports a year increasing approximately 8-10% annually. In addition about 420 autopsy reports are also processed yearly.

About 10 years ago the Pathology Department adopted a mechanized information system called Termatrex. The information storage medium for this system consists of cards with holes drilled in them. The drilling is done with a J-301 Jonkers Encoding

Machine which can accommodate 40 cards at a time. The holes in the cards are read with the aid of a backlighted board.

Ten thousand holes can be drilled in each of the 24 cm. x 23 cm. cards. The x-y coordinates of a hole are converted to a number which is the surgical pathology number of the patient. A surgical pathology number is assigned to each specimen processing request starting with surgical number 00001 which is assigned to the first specimen of the year. If the same patient has several specimens analysed during the same year then he is assigned the corresponding number of surgical numbers. A list which cross indexes surgical numbers with medical record numbers (unit numbers) is maintained in order to link the surgical pathology reports with the patient's chart.

Each card corresponds to one type of item that can be stored. For example there is an appendicitis card, a card for all females, a card of all patients between the ages of 41 and 50, etc. There is a number associated with each card which relates to the position of the card within the deck. This number also appears in the dictionary which lists all terms codable by this system. In this system the drilling process is the coding process. During this process a particular data item in a report is translated into a hole on the card carrying that data item. By reading-off the holes from a card, one may find all patients sharing the same data item in their report. The size of the card deck used depends on the number of items

one wishes to code in a system such as this. This usually includes all significant diagnostic terms a pathologist is interested in.

The MGH pathology department uses a dictionary of approximately 1400 terms. Different sections of the deck are differently coloured so as to group the term describing cards according to common criteria of interest such as different anatomic parts of the body.

The process of coding a pathology report (either surgical or autopsy) commences with the scanning of the text of the report to identify significant terms which are to be coded. A hole drilled in a card only means that that particular term is present. One has no way of coding any modifier to that term unless a separate card exists for that term. For example, the term "subacute appendicitis", cannot be exactly coded if there is only a card for "appendicitis". Alteration of information through coding is not uncommon even among systems whose dictionary is more sophisticated than the Termatrex dictionary.

The easiest type of retrieval possible with the Termatrex system is to find the surgical number of all patients with a certain diagnosis. Thus all patients with cholecystitis between 1965 and 1970 can be found by retrieving the cholecystitis card for each of the years from 1965 to 1970 and reading-off the hole numbers with the aid of the backlighted board. Questions containing the boolean operator AND can be answered. The cards corresponding to the operands in question are

retrieved, superimposed and read on the backlighted board. Wherever the light shines through a hole, a surgical number exists which meets all criteria expressed by the operands. A retrieval involving OR is similar to the simple retrieval of cholecystitis cases described above. We should point out that while these retrievals are possible they are time consuming and can become quite error prone depending on the conscientiousness of the Termatrex system operator.

Because of the nature of information storage in this system, questions involving the boolean operator NOT (e.g. patients who are female AND between 41-50 and did NOT have appendicitis) cannot be answered directly. Also it is not practically feasible to retrieve all data corresponding to a surgical number or all data corresponding to the same patient. In addition, pathologists often request the retrieval of all cases where clinical diagnosis A was present and pathological diagnosis B was reported. The Termatrex system cannot accommodate these requests.

There are also signs which indicate that the system has outgrown its efficient operating size. The 10 years of accumulated data have resulted in a sizable number of card decks which are becoming difficult to store. The large number of cards that must be read in order to scan all these years, results in slow retrieval time and the hospital cannot truly profit from the accumulated large data base. For example, to read-off the surgical numbers of the 6800 patients suffering from cholecystitis takes a whole day. This represents

50% more time than would have been needed in 1970 when only 40000 cases were accumulated. Furthermore, since one set of cards can only accomodate 10,000 patients at least another set is required for the 16,000 or more patients yearly (cost per set $111.) This means that the "number" of a hole does not correspond anymore to a surgical number and additional cross indexing tables are needed to correlate the hole numbers with the actual surgical numbers. This further slows the coding process. At present there is a delay of four months in the coding of reports.

In summary, an increasing number of reports are stored at a slower rate resulting in a decrease of at least 50% in the system utilization efficiency.

In spite of these drawbacks, the pathology department has maintained the system in operation because it was capable of satisfying many of the informational needs of the pathologist. Requests came into the Termatrex system not only from Mill sources but also from various outside bodies such as the United Nations and the World Health Organization. To satisfy these demands with the limited capability of the Termatrex system is clearly no longer possible. To serve the present demand for sophisticated report analysis and to provide the basis for the full utilization of the existing pathology data base, the conversion of the Termatrex system to a computer based automated system is indicated. The analysis, design and demonstration of the feasibility of such a system is the subject matter of this thesis.

Chapter 3

CODING SCHEMES IN PATHOLOGY

3.1 Introduction

Almost all pathology information systems use a dictionary or coding scheme. These dictionaries may be stored in the system or may be used as an accessory during manual coding. The terms in the dictionary serve as a reference "standard" for disease names which can be coded by the system. At the same time the dictionary establishes a hierarchical structuring of the disease names it contains. Therefore the coding scheme of a pathological information system has a great bearing on the storage, retrieval and correlation of reports.

Some coding schemes came into existence as a result of terms accumulated from the various reports which are entered for storage into an information system. Other schemes may have come into existence through the efforts of many workers who were interested in various studies that needed a systematized usage of disease names. Several of these schemes are internationally known and used in various fields of medicine. Some attempts were recently made in using them in information systems as well.

In this chapter we will discuss in more detail the content and structure of coding schemes. We will illustrate this with a detailed description of two internationally used schemes - International

Classification of Diseases (Adapted) and Systematized Nomenclature for Pathology. In the concluding section, we shall touch briefly on the subject of choosing a coding scheme suitable for a pathology information system.

3.2 The Content and Structure of Coding Schemes

Diagnostic statements contain one or more disease names. A disease name, such as 'malignant neoplasm of larynx', may contain one or several of diagnostic terms, anatomic terms and modifier terms. When a disease name is located in a dictionary, or when it is put in a form in which it can be entered into the computer, it is referred to as a disease entry or as a diagnostic entry. This can be either a text or a numeric entry depending on the nature of its content. Appendicitis, for example, is a general disease name. A dictionary may only contain the more specific diagnostic entry "appendicitis fulminating" and "appendicitis obstructive". Any one of these two entries is a text entry. Symptoms of a disease, syndromes* and eponyms** constitute special types of disease names.

Dictionaries formed by systematically assembling disease names can facilitate their study, avoid duplication of names and

---

*The aggregate of signs and symptoms associated with any morbid process and constituting together the picture of the disease (Anon, 1966a).

**The name of a disease, structure, operation, or procedure, supposedly derived from the name of the person who discovered or described it first (Anon, 1966a).

establish a relationship between them. This relationship may be that of a synonym or one in which certain diseases appear as sub-categories of others. These latter types of dictionaries which impose a structural relationship of several levels upon the diseases, are called hierarchical dictionaries, hierarchical thesauri or classification schemes. These classification schemes are usually assembled to carry out certain studies (e.g. study of tumour), or to increase the accuracy of disease names associated with a diagnosis being observed (Bohrod, 1971). The assembled dictionary should be sufficiently dynamic to allow for continuous changes that reflect the newest developments in the medical field. The assembly of totally satisfactory dictionaries is hindered by the lack of agreement among physicians on the specific names to be used for many diagnoses, or the exact relationship that exists among them.

If a code is assigned to every disease entry in a dictionary, then it changes the dictionary into a coding scheme. The code is usually numeric. Codes are useful in compressing disease names for the purposes of computerization. Their alphanumeric structure often reflects the structural organization of the dictionary. For example, in the ICDA dictionary (discussed below), all diseases of the digestive system have a code starting with the number 5.

One of the first comprehensive coding schemes to appear was the Standard Nomenclature of Diseases (SNOD) whose first edition was

published in 1932. During later years, the revised editions of SNOD (Jordan, 1947) also contained card indexing methods for implementation of information systems in the various hospital departments. SNOD is not in use any more, but the experience gained in using this dictionary was very valuable in the formation of two of the most widely used classification schemes today: SNOP and H-ICDA.

3.3. The Systematized Nomenclature of Pathology (SNOP) (Anon., 1965).

The Systematized Nomenclature of Pathology (SNOP) is intended primarily as a coding scheme to be used in Pathology Information Systems. A pathological diagnostic statement is coded by specifying it in terms of 4 different types of information. These consist of topography (T), morphology (M), (the structural changes produced compared to what is normal (Pratt and Thomas, 1966)) , etiology (E), and function (F) (physiological or chemical disorders within the body caused by the disease). A 4 digit numeric field is used for coding each of these 4 types of information. An example of the coding of a pathological diagnostic statement is (Pratt and Thomas, 1966):

T	M	E	F
Descending colon	Acute inflammation	due to Salmonella typhosa	with associated diarrhea
6760	4100	1361	7225

Through this method of coding, systems based on SNOP allow for a wide range of correlations to be carried out. This is further enhanced by the systematic use of codes. For example, the second digit of all morphology codes stands for the following attributes:

0 - NOS (not otherwise specified)

1 - Acute

2 - Subacute

3 - Chronic

4 - Granulomatous

Hence in order to retrieve all cases in which the morphology term contains the attribute "acute", one has to retrieve all cases with a morphology code of the type: *1**.

SNOP is a one-to-one correspondence coding scheme; that is, given a text entry and its type (that is either T,M,E, or F) one, and only one, code will be found. Any one numeric code may correspond to several text entries which are either synonymous or equivalent. Criticism of SNOP tends to be centered on the fact that while it is too specific for everday coding in the Pathology Department of a hospital, it does not contain sufficient clinical oriented data to enable the coding of the entire pathology report. An international effort is presently in progress to update SNOP to be more convenient for use in hospitals.

3.4 The International Classification of Diseases (Anon., 1972a)

The International Classification of Diseases (ICD) first appeared in 1950 and was later revised specifically for hospital use. In this new form it was given the name H-ICDA (ICD specifically adapted (ICDA) for use in hospital) (Anon., 1972a). H-ICDA (or ICDA, as we shall refer to it) is meant to be generally used in the hospital in contrast to SNOP's restricted use in pathology only. It has been realized that ICDA cannot only serve as a coding scheme but also as an educational tool which encourages conformity in the use of diagnostic terms. This served as motivation for the continuous updating of ICDA resulting in several revised editions.

The ICDA dictionary is divided into sections, some of which apply to major topographic regions of the body. Thus one finds a chapter on "Diseases of Digestive System", another on "Diseases of the Circulatory System" and so on. There are also chapters on "Infective and Parasitic Diseases", "Neoplasms", "Physical Signs, Symptoms and Ill-Defined conditions", "Injuries and Adverse Effects", and a separate section on "Classification of Operations and Treatments". Volume I contains a listing of entries arranged by the code numbers with the above mentioned chapter divisions. Volume 2 is an alphabetical listing of the entries for the purposes of coding. This listing often contains several versions of a particular diagnostic entry so that it can be readily found in whichever version it appears

in the medical record. For example, both "loss of weight" and "weight loss" are present in the alphabetical list.

ICDA is a hierarchical dictionary with its entries arranged on three levels proceeding from the general to the specific. Two examples of typical entry structures are;

level 1 - 564 Functional Disorders of Intestine

level 2 - 564.1 Irritable colon

level 3 - 564.1A Enterospasm

and,

level 1 - 568 Peritoneal adhesions

level 2 - 568E Adhesions (of) omentum.

The latter type is less frequently encountered and consists only of a general disease entry with a more specific subcategory added on. To code a diagnostic statement in most cases, it is sufficient to look up the alphabetic list and locate the statement irrespective of the level to which the statement corresponds. A diagnostic statement can contain terms at any, or all three of these levels depending on the specificity we wish to assign to a diagnosis. A code of an entry at a more specific level always implies the entries at the more general levels. The converse of this is not true and should be remembered when coding the diagnostic statement. If 564 was coded, the code does not imply the patient also suffered of enterospasm. Conversely 564.1A implies a complete diagnostic statement containing the

entries of 564, 564.1, and 564.1A.

The look up of an entry such as "fistula" may yield several codes. This happens because an entry term can be part of several diagnostic statements appearing in ICDA. Therefore ICDA is not a dictionary having a one-to-one correspondence between a text entry and its code. "Fistula" corresponds to 3 codes: 527.4, 543F, 537.1. The complete diagnostic statements within which it appears are:

527 - Diseases of the salivary glands

527.4 - Fistula

537 - Other diseases of stomach and duodenum

537.1 - Fistula

537.1A - Fistula gastrocolic

543 - Other diseases of appendix

543F - Fistula.

When multiple codes correspond to the same entry, correct coding can be assured if the entry is specified with an associated entry at another level. Thus, if we were interested in coding fistula as it applies to salivary glands we would search for the code that corresponds to the joint entries:

"Diseases of the salivary gland" and "Fistula".

These coding complexities of ICDA are on the whole not more serious than those of other dictionaries. For example, though

one benefits in the coding procedure from the one-to-one correspondence of the SNOP dictionary, one can only code with SNOP if, previously, the diagnostic statement was quite artificially split up into the 4 entry types accepted by SNOP.

We must also point out that the entries and their classification in ICDA are nearer to those disease names which are generally used by physicians.

3.5 Choosing a Coding Scheme to be Used in PIS

Coding schemes, whether large or small, differ greatly in their content, specialization, structure and completeness. One factor that causes differences in the completeness of dictionaries is the existing practical limitations on the size of a dictionary. Both the content and completeness of a dictionary are affected by the nature of the nomenclature it embodies. Some dictionaries may list only one form of the disease name, synonyms and equivalents being disregarded. Some dictionaries completely delete the use of eponyms, while others make use of them interchangeably with disease names.

Disagreement exists among physicians on the appropriate disease name to be used for certain diagnosis. This disagreement may be the cause of the differences that exist in the nomenclature and structure of various coding schemes. Doctors in the various specialties of medicine often compile dictionaries whose structures reflect their differing interests.

Rarely will members of a pathology department be satisfied with any one of the available dictionaries. Therefore, the designer of a pathology information system in conjunction with the pathologists of the department will have to assemble a suitable dictionary.

Both the completeness and the structuring of the adopted dictionary has to be evaluated. Two types of problems may sometimes occur while trying to code a report using this dictionary. One is that a disease entry is not present in the dictionary and an equivalent entry has to be chosen. The other is that an entry may be present but the hierarchical context in which it is present in the dictionary does not correspond to that intended by the pathologist. In both of these cases, if coding is completed, some alterations in the meaning of the coded report occurs. The fewer alterations occur, the better the assembled coding dictionary. Alterations in the meaning of the information stored should not be interpreted as an inherent fault of the coding scheme. Coding schemes can only be as good as the consistency with which disease names are used in describing diagnosis. In an information system based on a coding scheme, weaknesses of the scheme can be pinpointed and through updating, the weaknesses can be eliminated. Thus an unexpected benefit accrued from the use of these systems may very well be a higher degree of standardization in the use of disease names.

Chapter 4

PATHOLOGY INFORMATION SYSTEM (PIS) IN PERSPECTIVE

4.1 Introduction

The history of PIS extends over no more then 15 years. During this time there were many attempts of PIS implementation few of which culminated in a satisfactorily operating system. The critical review of the literature on PIS is difficult since it is often impossible to determine from the published material whether a system is only being planned, is already being worked upon or has been actually implemented. In addition, "follow up" articles which report on a system's perfotmance over a period of time are rare. This is very regrettable because this performance depends not only on the excellence of technological design and physical implementation but also on the acceptance of the system by its users. Many organizational, sociological, professional and emotional factors come into play during the introduction and operation of the system affecting its ultimate success or failure.

Finally, system evaluation and comparison on the basis of published material is often found to be misleading. Understandably it is not often possible to include detailed technical descriptions in an article published in a scientific journal. Consequently, one often finds that only the basic ideas and the system philosophy are discussed without the all important details which often are crucial in overcoming limitations of computer

facilities, storage, personnel and finances. Thus it is very easy to misjudge the actual capabilities of the systems described in the literature.

The approach I have used in describing the evolution of PIS consists of the discussion of a number of computerized PIS in the chronological order of appearance. Due to the limitations of this thesis only those systems are mentioned which embody interesting and new concepts and/or represent important stages of development. If several such systems appeared simultaneously, only one representative system is described. Therefore, I must apologize to all whose work I did not mention here.

4.2 Computerized PIS: A General Background

The evolution of computerized pathology information retrieval systems was predicated by the evolution of information retrieval systems for the military, industry and government and followed the development of computer utilization in hospitals.

The computers of the 50's were bulky and expensive. They operated at a slow speed with memory cycle times in the range of several microseconds, a limited core-memory of perhaps a few tens of kilobytes, external tape storage that could be used only for the simplest sequential data organization and input facilities consisting almost exclusively of punched card readers. These computers were used in simple business applications, in accounting and sorting, by the military in air defence control and science.

Scientific users were attracted mostly by the fast calculating ability of the computer. At this time the computers were not yet used for information processing and dissemination.

The computers of the 60's showed many advancements over their predecessors of the 50's. These newer computers operated at higher speeds, benefited from memory storage of hundreds of kilobytes, had an improved internal organization and occupied less space. System software and high level languages increased the versatility and ease of the operation of computers. A number of new peripheral storage devices appeared enabling random access of data. Soon afterwards the much needed storage and retrieval software was developed to allow the programming of information storage applications utilizing these peripheral devices. Also, developments in the field of telecommunications enabled the remote use of computers through terminals.

Although computers were used in hospitals since their commercial introduction, early computer applications were restricted to the business office and to certain research purposes. The widespread introduction of computers had to await the developments of the 60's. Among these developments, of particular significance was the appearance of minicomputers and time-sharing.

The minicomputer was developed primarily as a laboratory tool. In particular, it is widely used to collect laboratory results from laboratory instruments in numerical form, to process the data, to store them in appropriate files and to print-out a variety of

documents. In intensive care patient monitoring applications, the minicomputer monitors and processes physiological signals for purposes of display and patient health warning.

In general, minicomputers perform extremely well in applications where they are directly interfaced with instruments and where they are called upon to perform only a limited number of functions. They are exceptionally useful in "closed loop" applications where data collection must be accompanied by near instantaneous data processing and decision making followed by an appropriate control action.

Since the information processing and storage capability of minicomputers is limited, it is rare to encounter minicomputers in PIS. Nevertheless the minicomputer had a significant effect in shaping the attitudes of pathologists towards automation because many minicomputers are being used in clinical chemistry and laboratory systems. In fact there were even some attempts to extend such laboratory systems to include pathology. However these never succeeded because of the limited storage capacity and system software of minicomputers.

Time-sharing grew out from the need to work with the computer on an interactive basis. Ordinarily this kind of operation would be prohibitively expensive because the computer's central processing unit (CPU) is capable of performing millions of operations per second while the speed at which a human can input data or absorb output from a computer is much slower. Consequently it was necessary to design a

scheme which on one hand would provide people with an ability to communicate with a computer and at the same time make the computer operation economical by allowing its CPU to operate at full speed and without any idle periods.

In a time-shared system several computer users are connected to the computer simultaneously. While data is entered or is being printed at the terminal of one user, another user's data is processed in the computer. This way a user is always able to input data, and the output of processed data is almost instantaneously available. Each user is serviced by the CPU in a "round robin" fashion for a period of a fraction of a second. Meanwhile, the impression remaining with any one of the users is that he is the sole user of the system.

Modern telecommunication technology enables hospitals to communicate with a distant computer using ordinary telephone lines. Through time-sharing facilities the input and the output of data is not interrupted by a waiting period. Thus, both practically and financially, remote interaction with computer based data becomes feasible.

4.3 The University of California Hospital Laboratory Medicine System

We will start now with the discussion of pathology information systems. One of the first systems was built by B.G. Lamson (Lamson, 1965), whose goals were similar to those of other medical information systems designed at about the same time (e.g. Smith and Melton, 1963; Lindberg, 1965; Korein, 1963). These goals were:

The computerization of the hospital information system. The recommended approach was to carry this out in a stepwise fashion, the implementation of a pathology system being one of the necessary steps;

Devising a computerized system for medical diagnosing or a tool to aid in the diagnostic research.

Lamson's system gathered both clinical laboratory and pathology data. He designed the system to provide better communication between the laboratory and the clinicians on the ward via display terminals. He hoped that the data accumulated on patients in such a computerized system would be easier to disseminate for medical research purposes.

The operation of the system is shown in Fig. 4.1. An IBM 1410 computer was used with a 1301 Disk file serviced by rudimentary random access software. The system was operated by specially trained personnel. The punched card input interface and the equipment of the time needed much expertise from those who operated it. Because of equipment limitations the operational procedures shown in Fig. 4.1 include seemingly unnecessary and complex routines. This operational system however, facilitated a smoother flow of the laboratory results to the clinician than was previously possible. It also resulted in extensive monitoring of the performance of the laboratory which proved to be very useful for quality control and management.

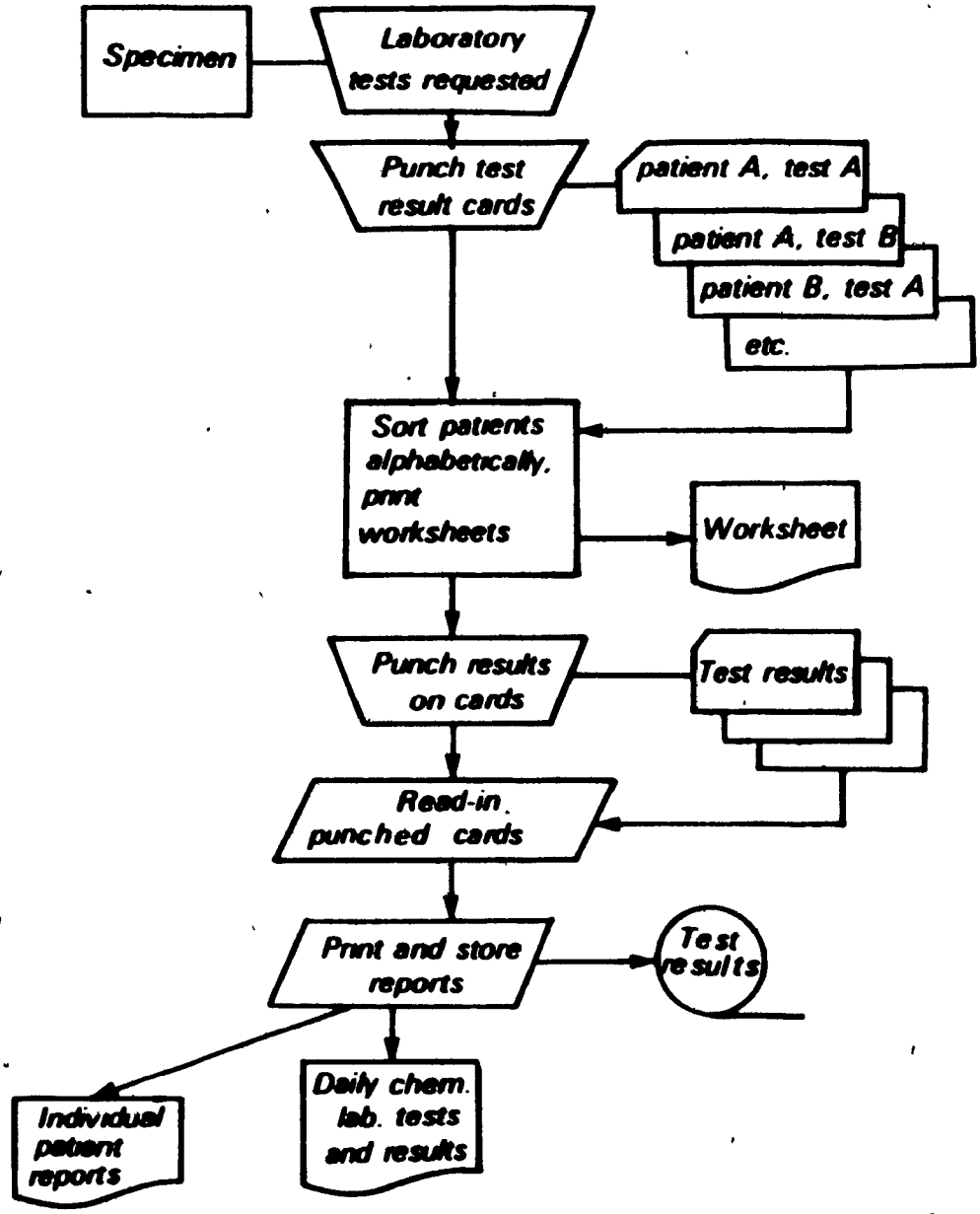


Fig. 4.1 -- System chart for the operation of the University of California Hospital System

Since more than 6000 pathological diagnoses had to be processed each year, Lamson designed his system so that it would accept these diagnoses in free text form. The coding of the text was carried out through a table-look-up procedure. Unfortunately, very little information was provided about this coding process or the dictionary. The scant description though, suggests that Lamson's dictionary was constructed similarly to that of Smith and Melton (1963, 1964). In their approach the entries of the dictionary were accumulated by scanning every new report until a new diagnostic term was found Terms which were added to the dictionary during previous scans were of course ignored. A code was assigned to each new entry which was then entered into the dictionary.

Based on their experiences, Lamson, Melton and Smith, Lindberg, Korein and others pointed out the difficulties associated with this kind of work. This was very significant since there was a great wave of enthusiasm for employing computers in the hospital. In fact, there were many optimistic predictions that in a very few years hospital information systems would become a functional reality. This did not prove to be so since the application of computers alone is not enough to solve the problem. It is necessary that system analysis precede automation; in particular, the analysis of the procedures in the hospital, the analysis of the systems that are being used to code and classify information, and so on. In fact -as for example, Lamson pointed out , even such a tractable subsection of the

total hospital information system as the clinical laboratory system was far from satisfactorily implemented. In particular, cost and complexity of the system had to be carefully considered. And, the problem of free text processing and retrieval has remained to this day.

4.4 An Interactive Clinical Pathology Data Analysis System

The SMOLDS system (Krieg et al., 1968) was completed by 1966 for the Department of Pathology, Upstate Medical Centre, SUNY, Syracuse, New York. This system came into existence to automate the reporting of laboratory generated data whose volume was increasing by 10% to 20% a year. The data base generated was meant to be queried by the clinician in the hope of obtaining feedback on the treatment procedures used. Therefore the system was designed to allow the physician to communicate with the computer without interposing an intermediary data handler.

In order to make such an arrangement functional it is imperative that the system allow easy, direct and realtime communication between the physician and the computer. In particular, for the physician the primary activity is that of treating patients and therefore he will quickly be discouraged from the use of even the most sophisticated system, whatever its potential, if he has to pay a large time penalty. For example, if he has to prepare questions on punched cards, submit them to a computer operator and then has to wait for the answer he will be rapidly discouraged from using the computer as a daily tool. The system designed by Krieg,et al.

overcame this problem by using a dedicated computer (IBM 1410) with a typewriter terminal. In addition, the operating program was designed to accept English language commands which the clinician could learn with a minimal amount of instruction. Such querying is feasible if an appropriate data base is designed. A two-level structure (Fig. 4.2) was found to be satisfactory.

The user has to have a general knowledge of the data base structure in order to phrase his query. The original question is broken down into the three basic steps of: a) data access, b) processing and c) result display. Each of these steps can be entered using such commands as FIND, AVERAGE, COMPRESS, PRINT, GRAPH, etc. Boolean operators are accepted by the system and data obtained in retrieval can be used in subsequent steps. The article describing the system contains examples of queries which yielded clinically significant findings. It is not often that one finds a system whose implementation is completed to this extent before it is actually reported in the literature !

4.5 The Radiology System at the University of Arkansas Medical Centre

Further developments of interactive systems were handicapped by problems of processing narrative text. Many experimental systems have been devised in the quest of a system which would:

- Accept at the input a narrative text as written by the physician;
- Extract from this narrative text the medically significant information;
- Store the information in either coded or text form but in an efficiently retrievable manner; and

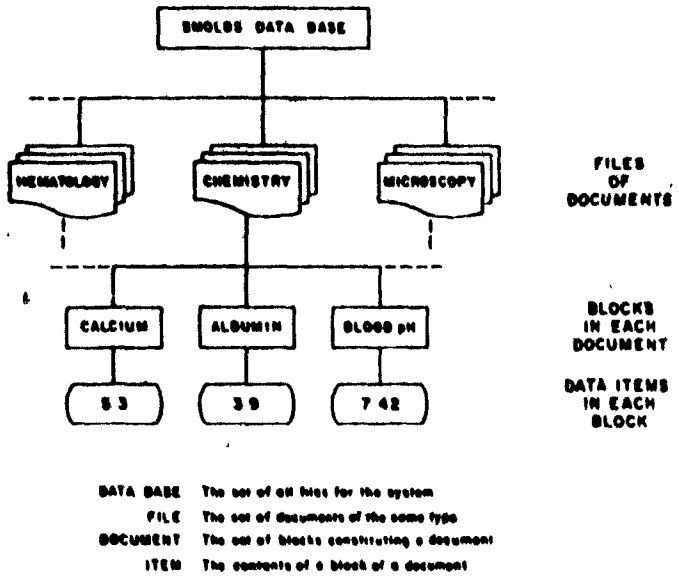


Fig. 4.2 - Structure of SMOLDS data base (Krieg et al., 1968).

- Have the ability of recognizing the hierarchical ordering of the informational items and of automatically grouping entries belonging to the same class. It is particularly important that the system have the ability to recognize synonyms and expressions which contain essentially the same idea and the same information.

A system which satisfies at least some of these aims, was implemented by Barnhard et al. (1966) at the University of Arkansas Medical Centre. This system processes radiology reports which present the same text processing problems as pathology reports.

The approach taken by Barnhard et al. is illustrated in Fig. 4.3. While the report is typed, the text is also punched on paper tape or cards. The text is copied onto magnetic tape which serves as the input to the processor. In effect, this section of the operation is an early version of today's key-to-tape data entry.

The text is screened by the system to locate the keywords which carry the medical information. A word is labelled as a "keyword" if it is found in the keyword dictionary of the system. Two other classes of words are also identified: " discard" words and "unknown" words. Discard words are those which can be skipped since they are not meaningful. Included in this class are articles, connecting words, etc. A separate dictionary contains these words. Words which cannot be found in either the keyword dictionary or the

discard word dictionary are labelled as unknown words and are investigated separately by the operator. Some of these unknown words may contain misspelled keywords, new discard words or a not-yet-encountered keyword. Hence from this stream of unknown words, keywords may be extracted and added to the dictionary. The entire keyword dictionary was built up through this method. However, the authors of the system found that the dictionary would not stabilize even after the scanning of 100,000 words and continuous additions were necessary (Barnhard et al. 1968). Furthermore, the frequency distribution of word occurrence did not warrant the reorganization of the dictionary into a stack which contained the more frequently used words at its top. In retrospect, it seems that the size of the dictionary used was far too small, compared to ICDA or SNOP (see sections 3.3, 3.4), to achieve stability.

A more detailed illustration of the text-processing steps followed by this system is shown in Fig. 4.4. The numeric characters heading the text are patient identification data. In the first processing run, illegal words and findings following a negation are deleted from the text. The choice of deleting negative diagnostic findings is questionable since in many cases the absence of diagnostic signs is highly significant. During the second run the discard and the unknown words are deleted. The remaining words contain either an anatomical term, a pathological (diagnosis) term, a term indicating size, degree of positivity or location side.

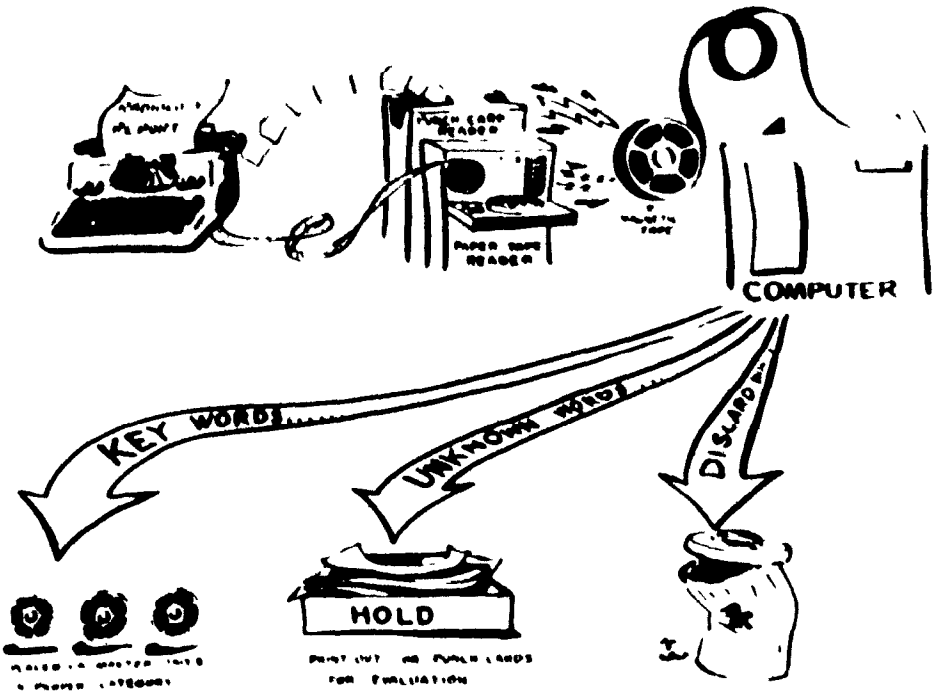


Fig. 4.3 - The words of the radiology report are typed (upper left) and simultaneously punched on paper tape or cards in machine readable form. Following transfer to magnetic tape the computer compares the words of the report with its "dictionary" and treats them as key, discard or unknown words (Barnhard and Long, 1966).

ORIGINAL RECORD

25745 16905112362 CHEST THE HEART IS SLIGHTLY ENLARGED PREDOMINANTLY IN THE REGION OF THE LEFT VENTRICLE. THE LUNG FIELDS ARE CLEAR AND NO OTHER THORACIC ABNORMALITIES ARE SEEN. IMPRESSION MILD CARDIOMEGALY. ETIOLOGY UNDETERMINED.

RESULTS AFTER NEGATION AND DELETION OF MOST COMMON WORDS AND WORDS WITH ILLEGAL CHARACTERS.

00043 HEART SLIGHTLY ENLARGED PREDOMINANTLY REGION LEFT VENTRICLE LUNG FIELDS CLEAR. MILD CARDIOMEGALY ETIOLOGY UNDETERMINED

RESULTS AFTER REJECTING DISCARD WORDS AND WORDS NOT FOUND IN DICTIONARY

00043 HEART = A SLIGHTLY = M ENLARGED = 0 PREDOMINANTLY = M REGION = A LEFT = M VENTRICLE = A. LUNG = A. MILD = M CARDIOMEGALY = 0

RESULTS AFTER APPLIST

A25745 HEART SLIGHTLY ENLARGED PREDOMINANTLY REGION LEFT VENTRICLE MILD CARDIOMEGALY.

Fig. 4.4 - Computer print-out of the text processing steps (Barnhard et al., 1968).

The type is identified by looking up the term in the keyword dictionary. An attempt was made to impose some hierarchy upon the keywords but in this system this could not be done successfully. In the last processing run, the so called AP test eliminates those sentences which contain only an anatomical term. When all the keywords are identified, the patient identification number is added to the storage areas of each recognized keyword. Each keyword has such a storage area which serves as a "linkage" of patient data.

This method of building up the dictionary is again similar to that used by Smith and Melton (1963, 1964). Barnhard, et al. hoped that the improved computer facilities, storage devices, software and text input devices would allow the implementation of a system that processes narrative text with a higher degree of success than was possible before.

Although conceptually it seems possible to carry out correlation of data items involving comparison and boolean operators, the authors only illustrate data retrieval involving the use of the AND operator. Output of stored reports is not truly possible with this system. Only a random listing of keywords found within the text of the report is feasible. Thus it is difficult to check for correctness of the entered report and updating becomes meaningless.

4.6 The Johns Hopkins Hospital System for Autopsy Report Analysis

Paplanus et al. (1969) aimed to implement a system that would aid clinicians in their attempt to correlate data contained in the autopsy report. Since a working system was desired, the designers of this system stayed away from any approach that would have required long experimentation and unproven techniques. The system was to be built so that clinicians could operate it with ease and use it as an analytic tool.

The approach adopted was to input the diagnostic terms in full English, not in the narrative text form of the autopsy report but in a pre-edited form. The pre-editing process extracted the diagnostic terms of the report for storage. This manual operation overcame the text entry problem that Barnhard et al. could not solve reliably using the computer.

The pre-editing process consists of rewriting the autopsy report in a form that contains the patient's identification data and a listing of all diagnosis found in the original report. The relevant anatomy term must appear ahead of each diagnosis listed. If no anatomy term is present in the diagnosis, or the existent anatomy term is not satisfactory for correlation purposes, then a more satisfactory one is added. For example, a diagnosis of "arteriosclerosis" would be entered as "blood vessel, arteriosclerosis". The dictionary of the system is built up as new terms are encountered within the entered reports. This again reminds us of the early approach used by Smith and Melton (1963, 1964).

In the system implemented by Paplanus et al. the reports are entered in a key-to-tape operation using an IBM 1401 computer. The tape serves as the input medium to the system's software which resides on an IBM 7094. As in the radiology system mentioned in the previous section, each keyword has its own storage area containing the patient identification numbers of the reports with this keyword in them.

A book is printed out regularly containing an alphabetical list of all diagnostic entries encountered by the system together with the respective report numbers. This book is used in manual searches of the reports.

Two types of automated searches were possible in this system. One was a search of reports using a search key containing boolean and comparison operators. The search key was used in looking up the reports residing on the input tape. If a report satisfied the key, it was retrieved in its entirety. Another search could look up classes of diagnoses, each of which contained the same given word. For example, a number of diagnoses containing the word "abscess" could be found in the dictionary. Then all reports containing at least one of these diagnoses would be retrieved. This search still does not allow, for example, the retrieval of all cases with "malignant neoplasm" diagnosis because not all names of malignant neoplasms contain this word in their name. The external storage media of this system was tape and consequently the reports were stored sequentially. Thus the cost of these retrievals was high.

This autopsy system was already in operation for 18 months at the time it was reported and the evaluation of the system design concepts was possible. Two reasons were cited for not using numeric coding: (a) the possibility of errors occurring during coding, and (b) the alteration in the meaning of the original diagnosis in the process of coding and decoding. The authors recognized the possibility of misspelling a diagnosis and thus automatically creating a new entry in the dictionary, but they pointed out that this is easier to spot than a misspelled numeric code. The fact that the data base did not allow a true hierarchical retrieval was also mentioned by the authors.

Since the system did not recognize synonyms or diagnoses which are the same but slightly differently worded, effective searches could only be carried out if a consistent diagnostic entry was used. In view of this, diagnostic entries had to be sometimes reworded before they were entered into the system. The alteration in meaning induced through rewording is comparable to that of searching a coding dictionary for the nearest diagnostic entry to fit a term. Therefore the choice of using Paplanus' approach or the coding dictionary approach, becomes a matter of personal preference.

It is perhaps worth mentioning that most of the systems attempted till this time tried to cope first of all with the problem of narrative text entry. Extensive programming efforts resulted only in a partially satisfactory system. In the end, the

designers of these systems summarized the difficulties they encountered with narrative text entry and added that a truly satisfactory system should also be able to perform hierarchical retrievals. However, it would have been possible around 1968, to design a system which would have performed satisfactorily in carrying out hierarchical searches. Such a system would have used a numeric coding scheme and would have yielded English text output. The input would have consisted of the manually coded diagnostic terms. Organization of the stored report could have been identical to that used in the system of Paplanus et al. The patient identification would have been placed in the storage area allowed for each code. The output would have been obtained by translating the codes back into text using a disk stored numeric code to text dictionary. The hierarchy built into the dictionary would have enabled the desired hierarchical retrieval. The system could have been programmed in FORTRAN.

4.7 Investigations of Narrative Text Input in PIS

During the past 15 years, many attempts were made to communicate with the computer through both narrative text and specially designed computer languages. Some of the so called high level computer languages were designed for the specific purpose of easing the programming task and to eliminate the need to understand the "inner workings" of the computer. These computer languages were designed to contain many ordinary words,

such as SKIP, EDIT, WRITE and LIST. The design of such languages was made possible through studies of the organization of ordinary and artificial languages and the manner in which they communicate information. Especially well known is Chomsky's work (Chomsky, 1965).

Some of these new computer languages, such as PL/I, resemble the spoken human language. Current computer systems can easily translate these languages into machine language. Is it not reasonable then, to attempt to automate the translation of the human language as well? The dictionaries of current high level languages such as PL/I contain English words exclusively. Compared to the human language, however, the size of these dictionaries is significantly smaller, the rules for expressing a thought are much stricter and the range of thoughts expressible is much more limited. Therefore, the translator of the human language is bound to be more complex than any computer language translator.

Before we answer the above question we must define a few concepts which will serve as criteria for evaluating whether it is possible to build a human language translator. A language enables information to be communicated in specific increments called statements (in written English language these are the sentences). A statement contains a number of elements, each consisting of a word or a group of words. An element has a function. It is this function that determines whether an element in the English language becomes

a subject, a predicate, a complement, etc. The elements and their functions occur within a statement according to specific structural rules which are sometimes referred to as syntax. The totality of these rules forms the grammar of the language.

Syntactic analysis is the analysis which inquires into the structure of the statement. The process of finding a sequence of elements within a statement and identifying it as a rule of the grammar is called parsing. Syntactic analysis of a correctly written statement in a computer language always yields an equivalence between the string (a collection of alphanumerics) being analysed and one, and only one, rule of the grammar. In such a case the parse is said to be successful. In the English language, often more than one rule can be found to be equivalent to the same string, giving rise to syntactic ambiguity. Added to this, English grammar contains a large number of complex rules.

A statement can be analysed not only syntactically, but also semantically. Semantic analysis inquires into the meaning of a word and if more than one meaning is found semantic ambiguity arises. The information communicated by a statement is determined by both the syntax of the statement and semantics of the words.

The above introduced concepts were essential in the analysis and synthesis of computer languages and the implementation of the respective translator. In terms of these concepts our original question can now be rephrased to ask: Can we find a set of rules for the English language, or a subset of it, enabling us to carry

out syntactic and semantic analysis of any statement in this language. A subset, in this case, is the medical language used in formulating the pathology report. Medical language has a limited dictionary and has strong ties to Latin word structure and grammar. The many diagnostic statements expressed in this language resemble each other in structure. Therefore, it seems feasible to attempt an analysis leading to a translator. Such work has been carried out by Lamson and Dimsdale (1966) and Wong and Gaynon (1971).

While working on his laboratory medicine system (section 4.3), Lamson decided to investigate further the entry of pathological statements in natural text form. To this end he implemented a natural language information retrieval system specifically adapted for pathology (Lamson and Dimsdale, 1966). In this implementation, he reduced the complexity of his task by replacing the general syntactic structure of the diagnostic statements with a simpler one. Within this simple structure he assumed that it was not essential to carry out an exact syntactic and semantic analysis of the statement. Only the appearance and identification of certain words such as the terms (called descriptors by Lamson) denoting anatomy and diagnosis became essential. These terms were inserted in a directed graph which established the hierarchical relationship among them. This graph could also handle the synonymous relation existing

between certain terms.

Input to the system consists of the pathology report in an unedited form, of requests to make changes within the directed graph structure or of search questions also in unedited prose form. A dictionary of 7000 terms was accumulated. This continually expanding dictionary was kept functional with the aid of a logic structure maintenance program.

Search requests yielded copies of the original document input. Retrieval success was reported to be 100% while 7% of all documents retrieved were not relevant to the search request.

The system implemented on an IBM 7040/7094 was cumbersome. Significant complications in the maintenance of the logic structure resulted. Only questions involving NOT and AND operators could be used in searches. Lamson intended to expand this system on an IBM 360 for a definitive evaluation of this method. The results of this expansion have not yet been reported.

A more recent experiment to syntactically analyse surgical pathology diagnostic statements was attempted by Wong and Gaynon (Wong and Gaynon, 1971) at the University of Illinois. Rather than replace the complex structure of the diagnostic statement, as Lamson did at the expense of a complex dictionary, Wong and Gaynon decided to accept its full complexity and devise a grammar whose rules would enable parsing any diagnostic statement.

The grammar devised is shown in Fig. 4.6. The first step in the syntactic analysis process is the scanning process. The first scan looks for the statement delimiter, a period ('.'), and isolates from the text the diagnostic statement which will be analyzed structurally. Such a statement contains elements which can be one of three types: site, diagnosis and modifier. The word groups are separated into elements by the identification of delimiters existing between element types. The 13 groups of delimiters are shown in Fig. 4.5. Altogether 62 delimiters are present in these 13 groups. These element delimiters are searched for during the second scan of a statement (see example in Fig. 4.7). The element type is determined by investigating the delimiter preceding and following the element. Once the sequence of element types and delimiters is established, comparison with the grammar rules can be carried out. If the sequence matches one of the rules the parse is successful. If not, either the sequence represents a new rule not yet present in the grammar, or the statement did not parse successfully because of syntactic ambiguity. After the parsing is completed, the elements are stored in their full text form together with an element identification "tag".

Parsing test runs with this system resulted in a 10% error rate. The authors suggest that further improvements of the parsing routine can reduce this unacceptably high error rate. In addition, some arrangement that would establish a hierarchical relationship among the stored diagnostic statement elements would also be needed.

Group of Delimiters	*Example*
1 Conjunction	and
2. Diagnostic suffixes for transformation	-icitis of »appendicitis«
3 Garbage word	fragment, portion
4 Site pointer with garbage word	fragment of
5 Diagnostic delimiter	-oma, -osis, ulcer
6 Ambiguous pointer	consistent with
7 Site adjectival delimiter with transformation	-ial, renal
8 Diagnostic adjectival delimiter	-ating, -ed
9. Diagnostic pointer	with
10 Site pointer	of, involving
11 Comma	»,«
12 Semicolon	»;«
13 Period	» «

Fig. 4.5 - List of delimiters used in the parsing routine in 13 groups (Wong and Gaynon 1971)

Diagnostic Statement Type	*Example*
1) Dx of SITE (default)	Papillary transitional cell adenocarcinoma of urinary bladder, deeply penetrating into the muscular layer.
1a) Dx, MOD, *involving* SITE	Poorly differentiated adenocarcinoma, possible metastatic, involving skin of forehead.
1b) Dx, Dx and Dx	Squamous metaplasia, ulceration and acute inflammation.
1c) *Fragment* of Dx of Site	Fragment of poorly differentiated epidermoid carcinoma of cervix.
2) Dx-ive SITE (default)	Essentially normal vermiform appendix and jejunum.
2a) Dx-ive Dx of Site	Recurrent mixed tumor of parotid region.
3) SITE-al Dx (default)	Renal amyloidosis, advanced.
3a) SITE-al Site	Uterine content with hydatidiform mole.
4) Dx Site-itis	Acute appendicitis with perforation.
5) SITE with Dx (default)	Uterus with leiomyoma.
5a) *Consistent with* Dx of SITE	Consistent with Hodgkin's disease of breast.
5b) Dx with MOD of SITE	Neurilemmoma with central hemorrhage and necrosis of triceps region, right arm.
Note Dx = diagnosis, MOD = modifier	

Fig. 4.6 - Types of diagnostic statements used in surgical pathology reports acceptable by the parsing routine (Wong and Gaynon, 1971)

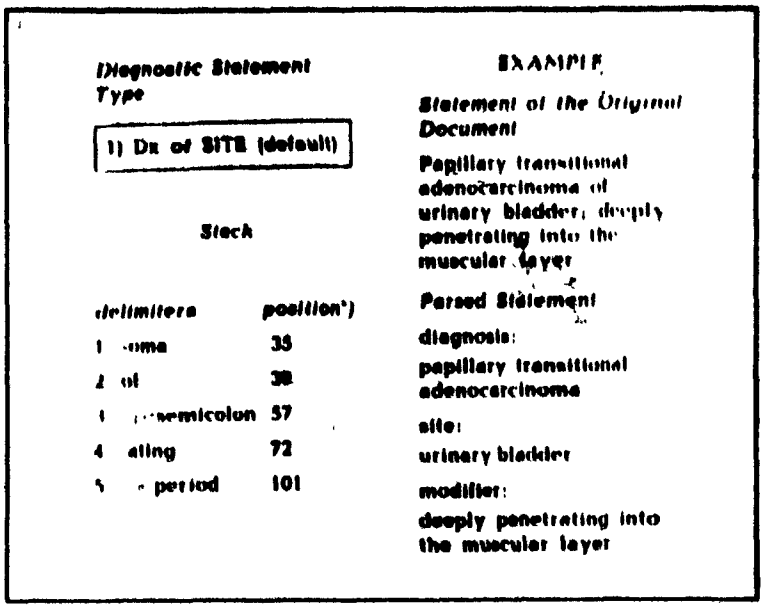


Fig. 4.7 - Example illustrating the operation of the parsing routine.
The above statement is scanned. First the delimiter "-oma" is encountered which signals that the preceding element is likely to be a "diagnosis". Then the next delimiter "of" is found indicating that the element up to and including "oma" is indeed a "diagnosis". Following this, delimiter ";" is encountered, suggesting the possibility of a "site" element following the "diagnosis". Delimiter "-ating" is found later, its presence confirming that the element up to the semicolon is a "site". The last delimiter, the end-of-statement symbol allows the parsing routine to conclude that the entire string following the semicolon is a "modifier", (Wong and Gaynon, 1971).

In their conclusion, Wong and Gaynon elaborate on this aspect by presenting a list of characteristics which a free text analyser must possess. This includes the ability to recognize synonymous or hierarchical relations among elements, interpret negative results and accept multiple word search keys. A later implementation of a PIS by the same authors (Gaynon and Wong, 1972) did not include this parsing routine. It seems that the above recommendations cannot be readily put into practice in a system using the parsing routine.

4.8 Present Directions in the Implementation of PIS

The predominant issue in the systems discussed in the foregoing has been the extraction of information from the pathology report entered in a narrative text form. A totally satisfactory method has not yet been found and at this time, we cannot even speculate when a truly successful system will be designed . A possible future solution may consist of devising a medical English language which would have a simpler syntax than the one currently used (Martin, 1973). This language would be widely taught and used by physicians to converse with systems containing medical data. Statements expressed in this language would be similar to natural language statements. The limitations would mainly be in the variety of sentence structures and the size and the specificity of the medical vocabulary. The effort of learning this more formal English language would probably be

accepted by clinicians if, in return, they would have the reward of effortless interaction with large medical data bases.

There are,however additional problems to be solved before total medical information systems serving the population of an entire region become functional. These problems and the relation between them are shown in Fig. 4.8. The investigation of these problem areas can be carried out independently. Thus a small medical information module, such as the pathology module, can be implemented even before the "physician-computer" conversation problems are solved. These modules can be used on their own until it becomes feasible to incorporate them into the larger medical information system (Grams, 1971).

We shall now proceed to investigate the various PIS modules that are either being "built" or could be "built" at present.

A new system was recently reported by Gaynon and Wong (1972), the authors of the parsing routine discussed in section 4.7. Their current system was designed for report retrievals, statistical inquiries and linkage. In particular, the system was meant to link statements from reports with the corresponding microscopic and photographic slides. The input to the system, read in as free text, consists of the tissue examination request form and the surgical pathology report. After statements and words are delimited, the identification of the disease entries is achieved through a look-up procedure of the system's dictionary, called

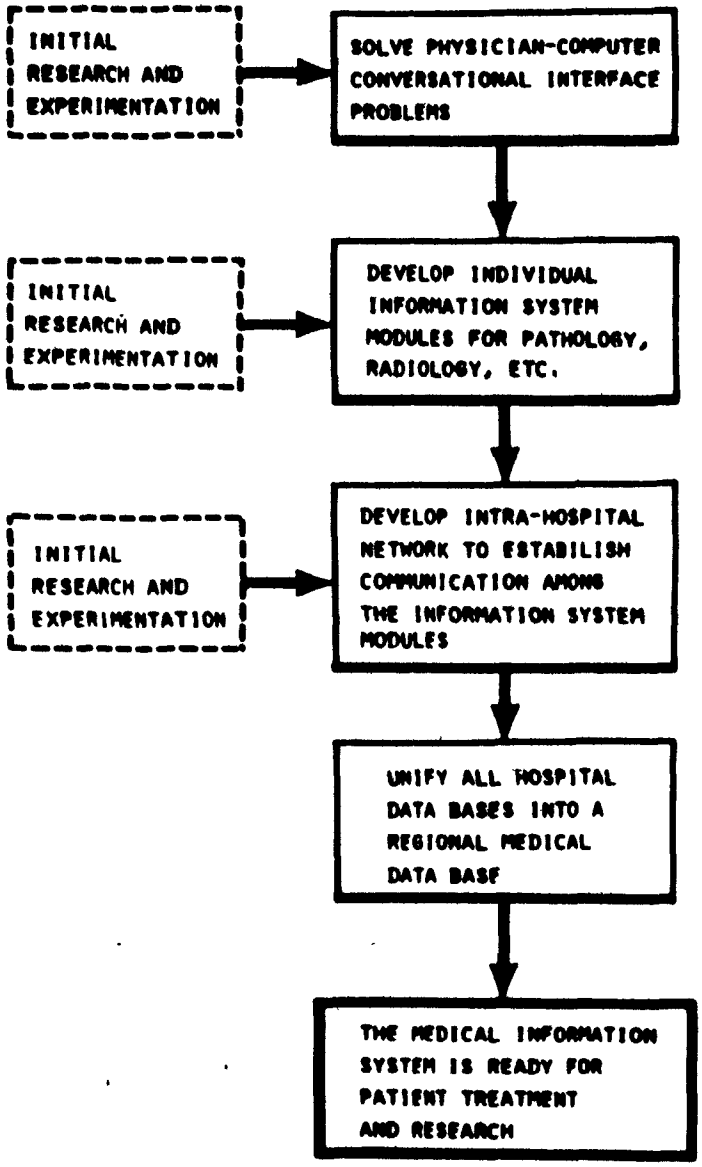


Fig. 4.8 - The sequence of problems that have to be solved for the implementation of a medical information system for a given region.

Lexicon. Hence, no structural analysis takes place and only the presence of certain words is investigated (context-free analysis). Lexicon changes some of the words into the system's standard form. It also enables the recognition of negation within statements, the recognition of synonyms, and the implication of certain diagnostic terms. It does not recognize any hierarchical relation among disease entries.

Each statement read in is rotated so that several versions of it exist, each starting with one of the identifiable keywords (this is similar to IBM's KWIC process). These resulting statements and the corresponding surgical number are entered in one of the 3 catalogs for clinical diagnosis, operative procedures and pathological diagnosis. These catalogs make possible the desired linkage operation and the retrieval of the original report, given any one of the statements it contains. The wording of a retrieved report is usually different from the original, because of the word standardization changes performed by Lexicon. Due to the same reason the report is also difficult to read.

The system was programmed on an IBM 370/155 with a 250K partition using the PL/I language. A general purpose information retrieval software package was also adapted for use in this system. Large amounts of external storage are used by the various files. Lexicon was reported to occupy 8.5 million bytes and it was still accumulating new words from the reports entered.

While an attractive feature of the system is that it allows narrative text input, the system suffers of high faulty retrieval rate. In addition the system has a limited correlation capability which is due to the lack of hierarchical organization of the stored diagnostic terms. Of course a system with such a vast storage capacity and software package is expensive to run. What is most discouraging though is that even such a large scale system as this cannot offer a working combination of free text-input and satisfactory correlative retrievals.

Alternative automated pathology systems concentrate on solutions in which the emphasis is on retrieval and correlation effectiveness, rather than conversational effectiveness. Therefore, emphasis is on a data base in which the hierarchical relationship among the diagnostic entries is recognized. In addition, extensive software is available to enable the answering of a variety of questions.

The practical approach to hierarchical retrieval capability lies within the use of coding schemes. Traditionally, pathology information systems based on coding schemes used either the SNOP dictionary or some smaller dictionary developed in the hospital (for example, the Termatrex dictionary used in the Montreal General Hospital) The popularity of the SNOP dictionary is not surprising because it was specifically designed for pathology. One of the earliest computer based systems using SNOP was implemented by Pratt and Thomas (1966) for the U.S. National Cancer

Institute. This system demonstrated the use of SNOP in coding diagnostic entries, and in carrying out retrieval operations.

All three types of reports - autopsy, cytology and surgical pathology - are processed in this system. The reports entered are manually pre edited, separating the diagnostic statement into the 4 types of codable information existing in SNOP - topography, morphology, ettology and function. The report is entered using punched cards which contain the patient identification and the pre edited diagnostic statement. The procedure for the latter is to enter the codes and the text coded on the same card. Essential information which is not codable can be added immediately following the diagnostic entry on the card. While Pratt and Thomas claim that computer look up of the dictionary is feasible in their system further informa tion on this is not provided and thus no evaluation can be made. They also claim that SNOP's organization allows most diagnostic statements to be entered into the computer in text form without having to resort to complex syntactic analysis. However, on two occasions, Wong and Gaynon found that the complexity has been underestimated. This was shown first in the implementation of the parsing routine (Wong and Gaynon, 1971), and later in the imple mentation of their system which used "dictionary look up" for syntactic analysis (Gaynon and Wong, 1972).

Several computerized searches are feasible with the system devised by Pratt and Thomas (1966). Included among these are correlations involving the AND, OR and AND/NOT operators.

Retrieval of a report yields only a collection of diagnostic statements appearing in the entered report.

Within the past 4-5 years few manually coded systems were implemented because the process of manual coding based on such large dictionaries as SNOP is time consuming and the handling of number codes can often result in error. Nevertheless, a number of systems exploring the use of SNOP were reported in 1972 (Van der Esch, 1972).

The recent appearance of new data management software and the continuous decrease in both the cost per bit storage and the cost of computer processing indicate that a more sophisticated approach towards pathology system implementation based on large coding schemes may prove fruitful. In this approach both the coding and decoding of pathology reports would be automated. Current high level languages such as PL/I allow the programming of such a system with reasonable effort.

This newer approach was used in the implementation of the Computerized Information System for Pathology (CISP) at the Montreal General Hospital. ICDA was the dictionary chosen by the pathologists for this system. This dictionary allows coding not only of the PD but also of the CN, CD and OP sections of the report. Thus, ICDA allows a more complete coding of the report than SNOP would. The resulting large data base in conjunction with ICDA's complex hierarchical organization makes the complex

correlations of stored data items feasible.

The pathology reports are entered in a pre-edited form. This pre-editing is necessary to assure that diagnostic statements correspond to the entries in the ICDA dictionary. Orthographic errors in the report are easily detected, either by the comptorized system - when it fails to find the misspelled entries in the dictionary - or by the system operator - when he compares the retrieved report with the original one.

The problems of information alteration through the pre-editing and coding process (see section 3.5) are not acute in this ICDA based system. This can be clearly seen from the resemblance of the wording of an original report and the wording of the same report rewritten in terms of the ICDA dictionary. The suitability of ICDA for CISP could be further improved through changes in the terminology of the stored version of this coding scheme.

Chapter 5

DESIGN CONCEPTS OF THE COMPUTERIZED INFORMATION SYSTEM FOR PATHOLOGY (CISP) (Hercz et al., 1972)

5.1 Introduction

The failure of the Termatrex system to satisfy the Montreal General Hospital provided the motivation to investigate a computerized system (section 2.5). Review of the literature suggested a number of possible approaches to this problem (Chp. 4). At the current level of development of computer technology, the utilization of large dictionaries with complex hierarchical structures seems both feasible and cost-effective. Since this approach has the potential to provide the data processing and correlation capabilities we seek, we have decided to make it the basis of our system design.

In this chapter the design concepts of the planned system are discussed. The system objectives specified by the pathologists of the MGH are translated into a series of working specifications. These specifications were reconsidered in view of the limitations of computer systems, storage devices, etc. A compromise practical approach for the implementation of CISP is developed and, thus, the basic components of the system can be described. This chapter also discusses some useful system implementation principles (e.g. system modularity principle) and the man-machine interaction in CISP.

5.2 User's Objectives, System Specifications and Constraints

The primary objective of the Montreal General Hospital's Pathology Department (the future users) has been the creation of a computerized system that would enable the storage and retrieval of pathology reports with greater speed, ease and detail than is presently possible with their Termatrex system. Their secondary objective has been to implement a system that would profit from the capabilities of the computer in extending the scope of the analysis of pathology information.

The Pathology Department's objectives for the computerized system were reinterpreted into specifications for this system. To this end, we carried out a detailed analysis of the Pathology Department's informational requirements. Our findings on the information flow environment were already summarized in Fig. 2.4. The following working specifications were derived to be used in the design and programming of the prototype system:

1) The data to be entered into the system consists of the pathology report which includes the clinically generated data. The data entered should be kept as near to its original free-text form as it is feasible. The input mode should be "on-line conversational" using a keyboard terminal. Provision should be made to detect, and wherever possible, to correct any typing or contextual error in the entered text. It should be remembered that the operator using the system will neither be

knowledgeable about computer data processing, nor will he be motivated, as a casual user, to remember complicated operating instructions. Therefore, the use of the system should be simple, employing only a limited number of English-like commands which could be easy to memorize;

2) The emphasis in the specification of the data base is on the subsequent analysis of the stored data. The structural classification of the stored diagnostic items should enable the retrieval of items that correspond to the level of entries in a large coding scheme. On the recommendation of the Pathology Department it was decided to code data using the ICDA dictionary. The coding of the report text should be automated;

3) The inquiries addressed to the system will require direct access to data on any specific patient by surgical number, name or unit number (hospital number). They will also demand statistics on any type of data item present within the data base, correlation involving these data items, or a combination of both. The correlative inquiries will be formulated using a combination of both comparison and boolean operators;

4) The output of the system will consist, first of all, of copies of reports entered. Quick and easy production of copies of past reports will enhance the communication with clinicians and with other hospital departments. The results of inquiries should be available in the form of text, table or graph;

- 5) The updating of the data base should be convenient and under the control of the operator. This will include updating of any section of the stored report or of any entry present in the stored dictionary;

- 6) As these specifications are for a prototype system, it is most likely that changes within the system will be necessary during the implementation phase. The design of the system should allow for these changes to be made at little expense and at a small programming effort; and,

- 7) The design of the system must recognize the importance of the system's reliabilities. The stored data should not be alterable in any way and this data should be retrievable. For example, simultaneous coding and decoding of a particular data item should result in an output equivalent to the input. Also, no accidental erasure of stored data is allowable either by the system internally or by unintentional operator input. In addition, no input by the operator should affect the future operation of the system.

Due to the experimental nature of this system it is not possible to specify à priori more stringent working specifications such as the average system response time. While we recognized the importance of such specifications, it was apparent that these could be given only after an evaluation of the completed CISP prototype system was made.

The working specifications detailed above can be implemented only if the constraints imposed by the limitations of the computing systems available to us are also taken into consideration. In the next sections we will consider these constraints and further modify our working specification to obtain the best possible compromise guidelines for the implementation of CISP.

5.3 The Computer System

The choice of the computer system for CISP is governed by the stated objectives which define a minimum required capability and by the available resources. An in-house independent computer system must be ruled out because of cost considerations. In particular, minicomputers which may be within a reasonable purchase price or rental range (say $50,000 or $1000/month respectively) do not have the required core memory, external storage facilities, programming languages and data management software (see also section 4.2).

The range of large-scale computing facilities available to us in principle includes the service bureaus, regional shared hospital computers and the McGill University Computing Center (MUCC). Our decision was to utilize the MUCC services.

There are two systems available on the campuswide terminal network. The first system, based on an IBM 370/155, provides on-line conversational services through time-sharing.

Unfortunately, this system is geared mostly to scientific work and its range of languages does not include any which are particularly suitable for text processing applications. The software available for data management was also judged to be too primitive for our purposes. In addition, it was estimated that the maximum effective usable core of 120K may be too small, and the maximum time slice (of the order of a few seconds) alloted to each user may be too short. Since searches in our CISP would conceivably require several such time slices, the resulting long system response time would destroy the advantages accrued from the on-line conversational capability.

The second system, an IBM 360/75 and its Operating System (OS), has a large core of up to 800 kilobytes and provides software which includes a large array of subroutines, sophisticated data management routines and several language compilers, including PL/I. The only difficulty with the adoption of this system for CISP lies in meeting the conversational requirements. In particular, the IBM 360/75 OS system offers only a multiprogramming environment. In this environment all programs are placed into a queue according to their priority and core requirement and executed in sequence (batch processing). It is not possible to "communicate" with the program while it is being executed to establish a "conversational mode".

It is possible, however, to enter from a remote terminal a data set containing programs and data and also to obtain the

output at the same terminal. This operational mode is called the Conversational Remote Job Entry (CRJE). CRJE allows the "interactive" editing of the entered data set. During such editing the data set can be updated, typing errors can be corrected, etc. These editing activities are carried out in a conversational manner with response times comparable to that of time shared systems.

In CISP the CRJE editor can be used to enter and manipulate the pathology report that is to be processed. The actual processing of the report is carried out in batch mode, with a turn-around time of less than 10 minutes. While waiting for the processing of this report, the operator may type-in the next report, and so on. Therefore, batch mode processing coupled with interactive report entering is acceptable .

A similar approach is taken in the programming of the querying routines. The retrieval, correlation and statistical programs will be tested and operated in batch mode. By that time, advancements in computer technology may allow the operation of these tested programs in the desirable conversational mode. If such a conversational mode will not be available, then CISP queries will have to be formatted according to stricter rules and entered using CRJE.

5.4 Data Base Organization

The data within a data base must be organized to make them suitable for the various analyses and retrieval operations specified

by the system's objectives. This organization is achieved by arranging the records in a file forming suitable data structures. Dodd (1969) has shown that no matter how complex the data structure is, it can be built up from three basic organizations: sequential, random and list. In a sequential organization the records are assembled according to a specific sequence. In random organization the physical arrangement of the records is at random and every record may be accessed directly by knowing their location on the storage device. In a list organization the records are coupled through the use of pointers which indicate the location of the next record irrespective of where that record resides physically.

Unfortunately, very little information is available in the literature (Carville et al., 1971) on how to choose a suitable data structure for an information system. In particular, no readily available rules exist which translate working specifications for an information system into specifications for an appropriate data base structure. Furthermore, although complex file systems are used extensively today, only scant information is available in the computer literature on the use of different languages for the implementation of file structures. The documentation which can be found mainly describes simple sequential organization. This situation is similar to the literature of computer sorting. In this area many computer developments occurred in the 50's and the various sorting methods became common knowledge among programmers. Nevertheless, not one article on sorting

methods appeared in the literature until 1956 (Knuth, 1972).

The structure of the CISP's data base was chosen by analysing the relationship that has to exist between any one record and the other records in the data base. The form of this relationship is governed by the various analyses and retrieval requirements that the data base will have to fulfill. The scanning paths that have to exist to fulfill these requirements are shown in Fig. 5.1 (see also Fig. 2.5). It is essential to be able to retrieve any record directly so that a patient's report may be retrieved easily. For analytical purposes the ability to scan the same type of data item (e.g. name) in every report is essential. This scanning is feasible if the data items can be treated as a list. Aside from the ability to scan data items "among" many records, it is often also essential to be able to scan the data items "within" a record.

A data organization that meets these criteria of data structuring is the indexed-sequential organization. In this data organization a record can be looked up directly (a) using a key which is translated into a record location in the data set * random organization or (b) in the alphabetic sequence order of the keys (sequential organization). Hence a particular organization of a data set may allow several methods of access of the data within the

---

* A collection of data items (e.g. a file) residing on an external storage device is referred to as a data set.

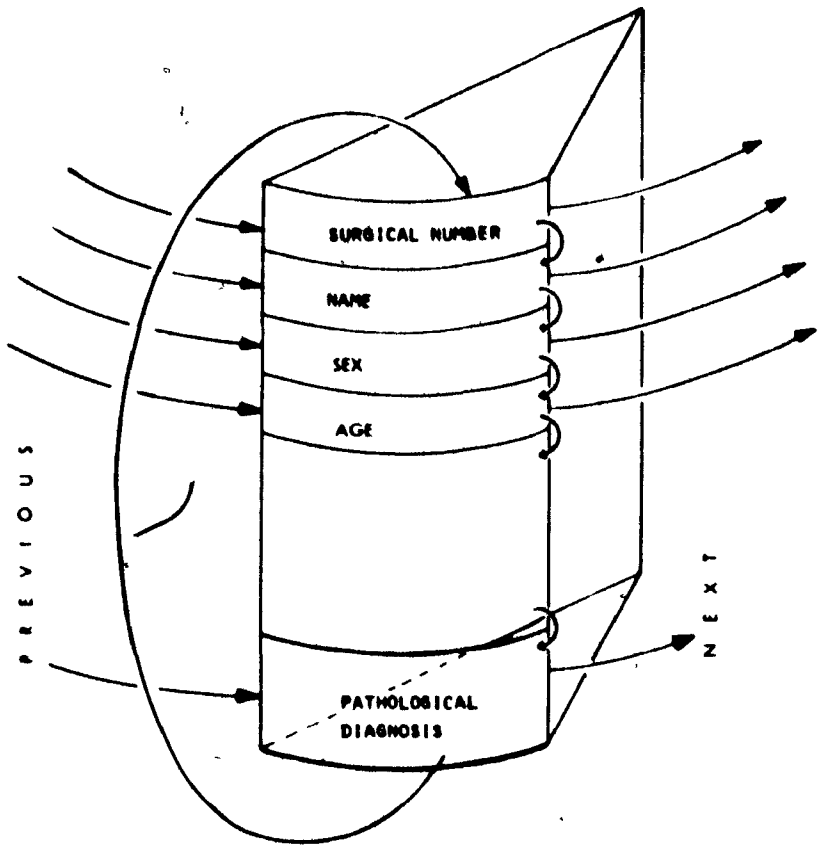


Fig. 5.1 - The scanning paths crossing a record in the data base. Arrows indicate the direction of the scanning paths. Any path that follows a sequence of such arrows is a feasible scanning path. The parallel arrows connect the same data item (e.g. age) in all records. The other arrows allow scanning of items "within" the record.

data set.

CISP's data base will also include the medical nomenclature dictionaries to be used in coding and decoding operations. In these dictionaries only the ability to look up a record directly is essential. On the computer system chosen, direct access is feasible through the use of keys, Since keys cannot be of variable length to accommodate the various text entries in a dictionary, we had to resort to a compromise dictionary look up method, which would allow, when necessary, the combination of both direct and sequential access. Again the indexed-sequential organization was found to be the most suitable for our purposes.

The software for the implementation of the indexed-sequential organization is available as part of the OS software on the IBM 360/75 computer. Because this software is widely used, it has the added benefit of being thoroughly tested and refined. The storage size of data sets using the indexed-sequential organization may extend over several disk packs*, assuring that large dictionaries such as ICDA can be stored in completeness.

Maintenance operations that keep a data set functional despite the many extensions, deletions and replacements of records can be readily programmed for the indexed-sequential data set.

*A disk pack consists of several disks mounted on the same vertical shaft. Usually both surfaces of the disk are used for recording.

5.5 Modularity and the Management of Modules in a System

A system which is expected to undergo modifications and expansions has to be implemented in a modular fashion. A system is modular if it is made up of a number of modules which contain a section of the system's program, usually in the form of a subroutine.

On the IBM 360/75 OS program modules are stored in data sets called partitioned libraries. These differ from the data sets discussed earlier in that instead of storing data, the processing programs are stored in them. Every partitioned library has a directory with the names of the programs present in the library. Whenever reference is made in a program to a subroutine stored in a specified library, the computer system automatically looks up the directory of the library, locates the program and reads it into core.

Program modules stored in the partitioned library are usually kept in either object or loadable code form (object module and load module, respectively)*. If a module is not executed very often, it is better to store it in object code form to save on the larger storage cost that would be incurred by the equivalent

---

* The compilation of the source program results in the object program. The object program is link-edited to obtain the load module which is in executable form.

load module. However, if the module is executed often, the module should be in loadable code form because the computer time saved in linkage editing will more than compensate for the larger storage cost.

The calling of subroutines, and hence the accessing of modules, is performed by a supervisory program (or a "driver"). This program interprets the commands to determine the type of processing needed, calls in the appropriate modules for execution and monitors their performance. A specific illustration on how this is performed in the context of CISP is seen in Fig. 5.2.

Originally it was hoped that some commercially available modules could be readily adopted for CISP and thus shorten the time required for implementation. Save for one exception, no such module was found.

5.6 PL/I: The Language Chosen for Programming CISP

Until recently the implementation of the general software for computer systems or user-oriented systems was carried out almost exclusively by using assembler language. This situation is now changing and higher level languages are being used increasingly for these purposes (Corbato, 1969). The main reason for using high level languages is that they allow a more direct translation of the problem into a program than lower level languages do. Also, to quote Corbato (1969), a high level language "forces one to design, not to fiddle with code".

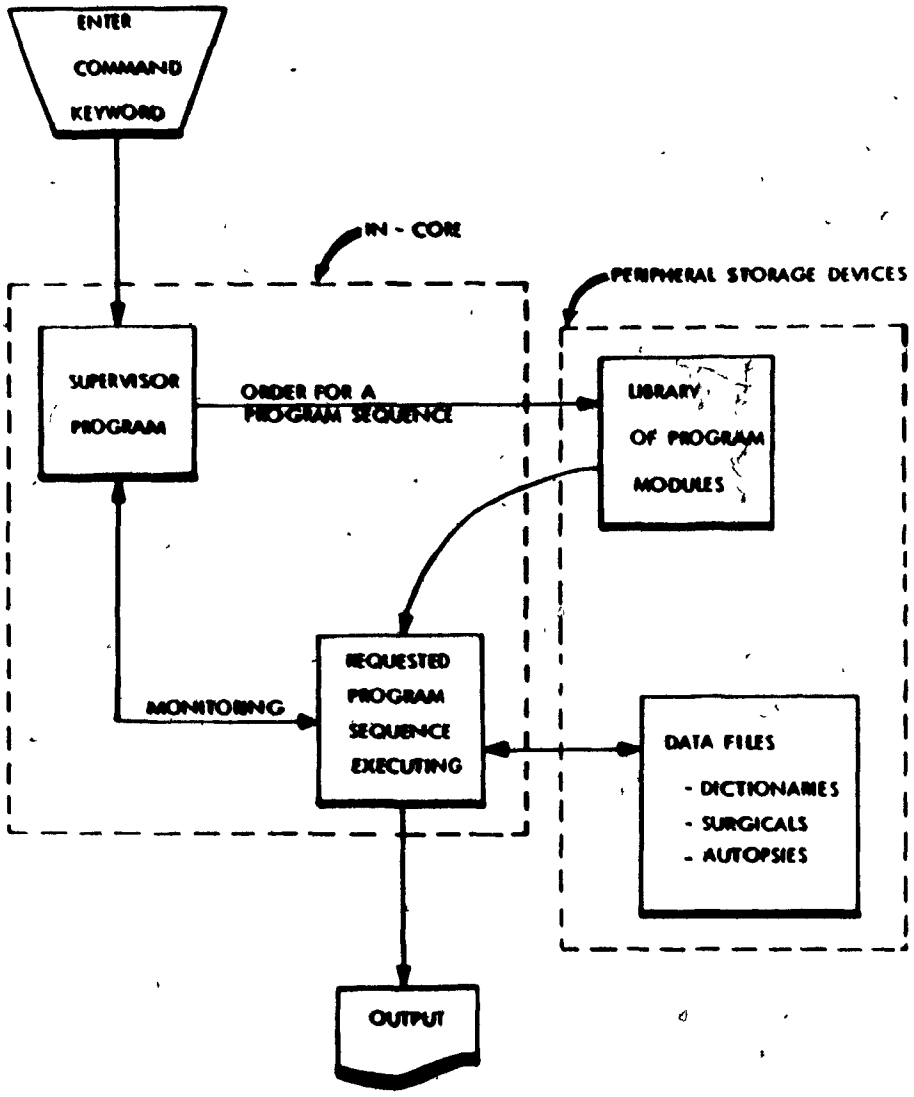


Fig. 5.2 - Processing with a modular system.

Because of the complexity of CISP, and because the programming of this system was to be a one-man effort, the use of a high level language becomes an absolute necessity. Although this may incur execution time and storage requirement penalties, it has been hoped that the use of a high level language will speed up programming and an operational prototype system could be implemented within the time limitations of this project. Once the system is operational, inefficient modules can be reprogrammed in assembler.

Programming language/One (PL/I) was the language chosen for the programming of the CISP. PL/I has a number of advantages that make it especially attractive. First, the language allows for modularity in programming. Thus, subroutine programs can be compiled without having to compile the main program as well. The resulting object or load module may then be stored in a partitioned library. Also the transfer of different data types (varying length strings, dynamically allocated variables, file names, etc.) between programs is well defined in this language, though not always simple to accomplish.

The second important asset of the language is its powerful facility for text processing. A number of variable attributes, operators and functions suitable for text manipulation are present in the language.

Third, the language is superior to other commonly used high-level languages in data management programming (Rubey, 1968). Data contained in a data set is accessed by defining a symbolic file in the program. The correspondence between the organization of a data set and that recognized in the program is established through the environment specification of the file in the program. The data access mode is specified through the attributes of the file. Examples of these specifications appear in the next chapter.

Fourth, the language incorporates many features associated with the manipulation of binary variables, boolean and comparison operators and list processing. These features are useful in interpreting queries and carrying out the requested processings. In addition to this, the Scientific Subroutine Package available to PL/I users contains a comprehensive set of scientific subroutines. These modules can be incorporated into those CISP programs which carry out statistical processing.

The construction of the PL/I language is not as systematic as that of Algol, for example. This, together with PL/I's complexity, inevitably results in a slow, large and costly compiler which translates into a significant system development cost. Surprisingly, this is not equivalent to saying that implementation of a system such as CISP would be cheaper if a language like assembler would be used

because the rapidity of programming in PL/I compensates for the extra costs involved (Corbato, 1969). At present an optimized version of the PL/I compiler is available for IBM computers. The hope is that this new compiler may produce object modules whose efficiency will approach that of assembler programs (Anon., 1973).

By programming CISP in a high level language such as PL/I we make it machine independent. This means that the CISP's programs can be easily run on another computer system which has a PL/I compiler and supports the data organization used in CISP (see Fig. 6.2). This allows CISP to be transfered to other hospitals which may want to use it.

In the IBM 360/75 OS environment, Job Control Language (JCL) is used in conjunction with PL/I in programming CISP. JCL statements are used to connect the symbolic file defined in the PL/I program with the actual data set residing on an external storage device. Maintenance of these data sets are carried out through statements, or utility programs, written in JCL. The creation, maintenance and management of partitioned libraries is also done through JCL. Unfortunately, JCL has a very strict syntax, has very poor error messages and descriptions of the language are complex, making programming in JCL a difficult and lengthy task.

5.7 The Basic Software Components of CISP

The consideration of the working specifications detailed in section 5.1 together with the computer system constraints and facilities described in this chapter enable us at this stage to specify the basic components of the working system. Fig. 5.3 illustrates the two basic phases of processing within the CISP. The first is the interactive phase in which a question or a report is typed-in and readied for the second phase in which the actual processing resulting in the storage, retrieval or correlation of records is carried out in batch mode.

The operation in the interactive phase is facilitated by the interactive editor of the system. This editor is implemented using CRJE's library of data sets and through programs which consist of CRJE editor commands.

The batch processing phase uses a large number of PL/I programmed modules and the system's data base. The text input is first investigated to determine the exact processing operation that has to be carried out. This information is relayed to the batch text editor program through command keywords embedded in the text and which this editor will identify. The investigation is carried out using the scanning method as shown in Fig. 5.4.

The command keywords identified by the batch text editor are transmitted to a supervisor program which calls in the module sequence that will carry out the requested operation.

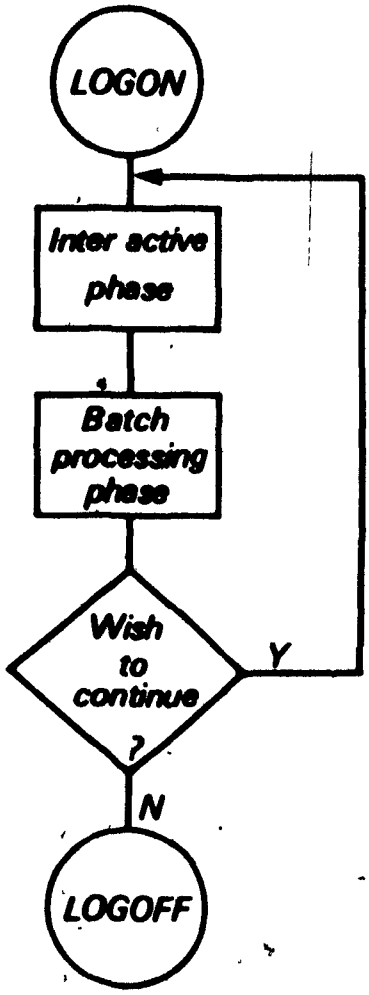


Fig. 5.3 - The interactive and batch processing phases in CISP.

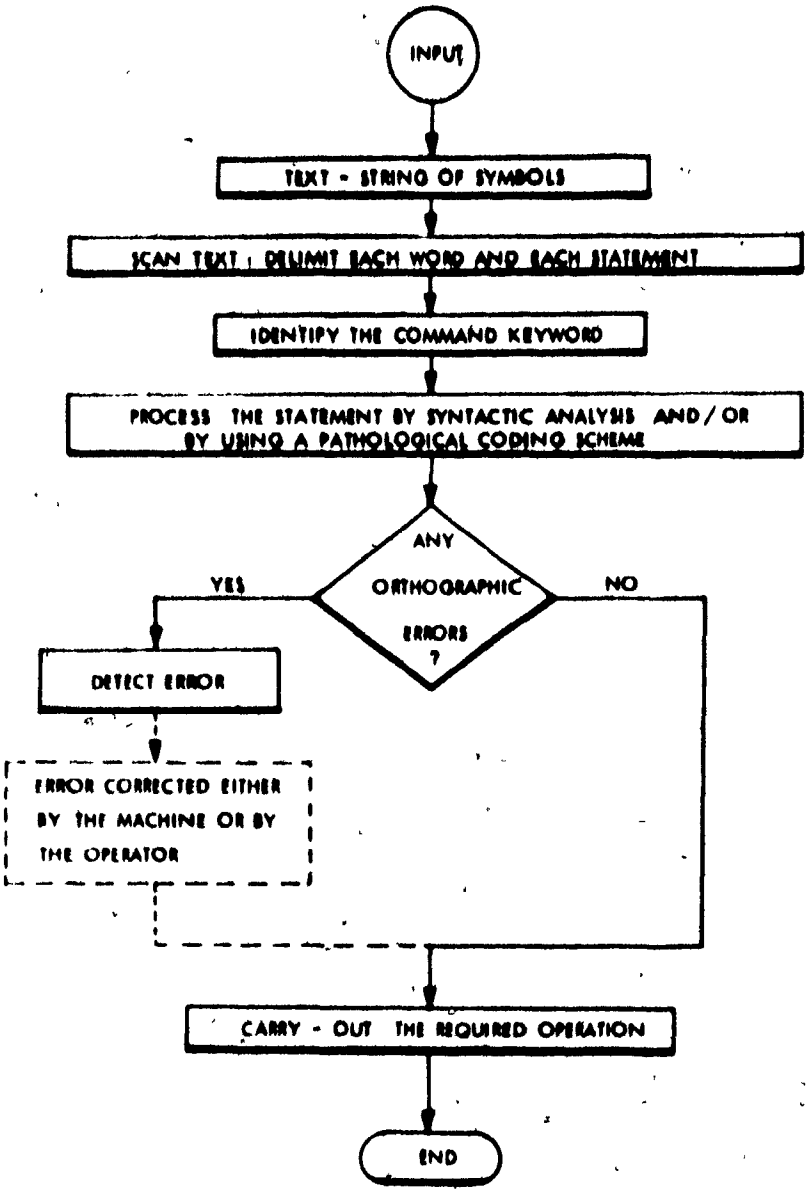


Fig. 5.4 - The scanning method as used to process the input text in CISP.

The supervisor also monitors the performance of these modules. The manner in which this is done has already been shown in Fig. 5.2.

Several types of information processing modules exist in the system. Some analyse and transform information which is not dictionary coded, others are specific file processing modules which look up entries in dictionaries and store the coded reports in the pathology report file. Among information processing modules we can also include those that print the reports either in coded, or in decoded form and those that interpret the questions addressed to the system and process the reply.

The data base consists of dictionary files for both coding and decoding operations and a variety of report files such as the surgical pathology and autopsy reports. The relation between the data base and the processing modules is shown in Fig. 5.2.

The last remaining component of CISP is the maintenance program collection. These programs are executed to reorganize the program library and/or the data base which have become cluttered as a result of updatings and expansions. Thus, maintenance operations assure that CISP's functioning continues to be faultless and efficient.

5.8 The Man-Machine Interface of CISP

The operator using CISP interacts with an automated system based on a computer. This interaction takes place across the man-machine interface of CISP. The ease and convenience of the man-machine interaction determines to a great extent the

degree of success of the automated system.

There are a number of factors influencing the quality of man machine interaction in CISP. Some of these factors are directly dependent on the behaviour of the working system. For example, a system which is highly interactive, has a short response time, uses easily memorized English word commands and issues clear and helpful error messages, possesses desirable system behaviour qualities. These qualities are within the control of the system designer while the system is being programmed. Other factors influencing man-machine interaction relate to the physical environment in which the system is used. In this category we can include the choice of the terminal and the noise and lighting of the room in which the terminal is used. Finally, the training of the operator will greatly determine the extent of the utilization of the system and hence the satisfaction derived from the existence of the system.

The importance of CISP's man-machine interface has been recognized early in the design stage. The design concepts detailed in this chapter have all been formulated with the aim of achieving an optimal man-machine interface.

Chapter 6

THE DATA BASE STORAGE AND PROGRAMMING OF CISP

6.1 Introduction

In this chapter the storage of the dictionaries and the programming of CISP are described. It is assumed that the reader is familiar with the PL/I programming language.

Parts of three medical dictionaries were used to make up CISP's dictionary. In addition to ICDA, sections of SNOP and the Termatrex dictionary were employed. It was found that it is not very practical to use the same dictionary for both coding and decoding. Hence, two versions exist: a coding dictionary and a decoding dictionary. To expedite changes in the dictionaries it was practical to keep sections of these dictionaries separated. Thus there is a coding dictionary for symptoms and diseases (CODE), one for operative procedures (OPERC), one for topography (CTOPOG) and one for neoplasms (CNEOP) (See Fig. 6.2). The corresponding four decoding dictionaries are DECODE, OPERD, DTOPOG and DNEOP, respectively.

In addition to the discussion of these dictionaries in this chapter the batch processing modules and the interactive phase of CISP are also described with the aid of flowcharts. The form in which a report is entered for coding and storage, the organization

of the coded pathology report and the maintenance procedures employed in this system are also presented.

6.2 The Storage and Access of the Dictionaries in CISP

6.2.1 Historical Note on Dictionary Storage. As mentioned in Chp. 5.3., the indexed-sequential (IS) organization was chosen for the storage of the dictionaries. At the time when we started programming CISP, the computer system software only enabled an IS organization with fixed length records. This was not suitable for the storage of the variable length entries of the ICDA dictionary. Variable length records could only be accommodated in data sets sequentially organized or those with the REGIONAL (3) organization. The need for direct access immediately ruled out the sequential organization, (Chp. 5.3). In the REGIONAL (3) organization records are stored in storage regions. Both the number of a "near-by" storage region and the key of the record need to be specified in order to retrieve a particular record. If the region specified is not "near" enough to the desired record, a significant amount of sequential search has to be carried out and the record may not be found though it is in the data set. Due to this reason, and to the difficulty of keeping track of the region number of updated records, we decided to abandon this organization. Instead we decided to use the IS organization with a record size of a few

thousand bytes (1 byte = 1 character = 8 bits). We designed a program that would fill every record with as many variable length dictionary entries as could be fitted in without truncation. Data regarding the number of entries and their individual length was also stored in the record to facilitate the search for the desired entry. Even after two months of programming effort a reliable and efficient program to store, update and delete entries could not be attained. This serves as an example that programs which should be part of the computer systems software cannot be implemented cost-effectively within the limited scope of a project such as CISP. Fortunately, however, IS data set organization with variable length records did become available on the IBM 360/75 OS computer system in the autumn of 1972.

6.2.2 IS Data Sets. The IBM 2314 type disk packs were used for the storage of IS data sets. The storage capacity per disk track is 7294 bytes. There are 200 tracks per disk surface with 20 surfaces in a pack forming the equivalent of 200 cylinders per pack (Anon., 1966b).

A minimum of one cylinder has to be allocated to an IS data set with its 3 data storage areas: prime, index and overflow. The records are stored in the prime area. The pointers that allow direct access of the records are stored in the index

area. Records that do not fit into the prime area are stored in the overflow area. The advantage of keeping all of these areas in the same cylinder is that access of data is very quick because no time is wasted on the positioning of the access head. However, if any of these areas need to be large, they have to be kept on separate cylinders.

Records stored on disk are grouped into blocks in order to minimize the number of empty gaps that would otherwise exist between records. The specific form of a block in an IS organization with variable length records is shown in Fig. 6.1. In this illustration the key of the record is embedded within the record. With this type of record organization less storage space is used and the record key is retrieved every time the record is retrieved.

6.2.3 Dictionaries. Two types of dictionaries are used by CISP: coding and decoding dictionaries. It is feasible, however, to store only one dictionary which could be utilized for both coding and decoding purposes. In particular, a set of pointers embedded in the dictionary would allow scanning the dictionary entries in an alphabetic order for text coding purposes, while another set of pointers would allow scanning it in the increasing numeric order for decoding purposes. The savings in storage space is not decisively significant, however, since the new space required by

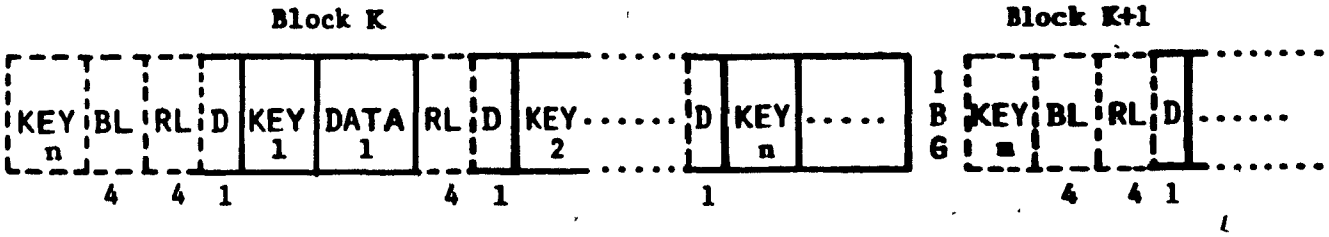
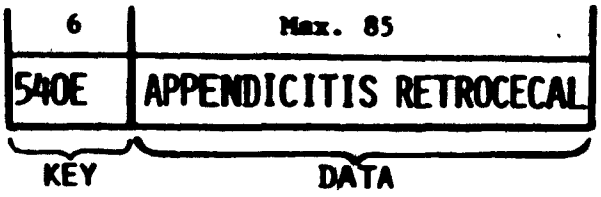


Fig. 6.1 - The structure of an IS block (Anon, 1972b; Brown, 1970) The dashed line indicates data inserted by the system software while the continuous line indicates data inserted by the programmer. The first item in the block is a copy of the key of the last record in the block (key n in block K and key m in block K+1). This item is followed by the block length count (BL). Each record starts with the record length count (RL) which is usually followed by the dummy field (D). This field contains either (8)'1'B if the data will be entered later in the record or (8)'0'B if the data in the record is not valid. The D field may be written by either the system program or the programmer. Separating the blocks is the Inter-Block-Gap (IBG). Numbers indicate the field length in characters.

the pointers nearly equals the space vacated by one dictionary. In addition, the scanning process increases the access time of a particular record and complicates the programming of the dictionary look up module. Therefore we decided on a "two" dictionary implementation which consists of storing a coding dictionary and a separate decoding dictionary. Since the coding dictionary can be recreated from the equivalent decoding dictionary, and inversely, the reliability of the data base is increased through this approach.

6.2.4 Decoding Dictionaries. The structure of the record in the decoding dictionary is the following (without the D-field - see Fig. 6.1):



The record key must be of constant length. A length of 6 characters was chosen, corresponding to the longest dictionary entry code. The data field can have any length up to 85. It contains the text of the dictionary entry corresponding to the code in the key field. Including the D and RL fields the maximum record length is 96 bytes (characters).

By choosing a certain block size we fix the amount of storage space wasted by inter-block-gaps, the time needed to read into core a block and the size of the buffer needed to hold a block. A small block takes little time to read in and needs only a small buffer to hold it but it incurs a great waste of storage space. To decide on the best compromise a few block sizes and the respective percentages of the total track space effectively occupied were considered. Remembering that the storage capacity per disk track is 7294 bytes, the following figures were obtained (Anon, 1966b):

1) 96 bytes - $\left(\frac{30 \text{ rec./tr} \times 96 \text{ bytes}}{7294 \text{ bytes/tr}}\right)$ - 39.7%

2) 1640 bytes - $\left(\frac{4 \times 1640}{7294}\right)$ - 90.5%

3) 3470 bytes - $\left(\frac{2 \times 3470}{7294}\right)$ - 95.8%

Examples 2) and 3) are the maximum block sizes for 4 and 2 blocks per track, respectively. While in example 1) (the unblocked case) more than half of the track is wasted, in examples 2) and 3) efficient use is made of the storage space.

In conjunction with above storage efficiency calculations, the following table of data retrieval times was considered (Anon., 1966b):

Block size to be read-in (bytes)	Transfer Time (msec)	Rotational Delay (1) (msec)	Total (1) (msec)	Rotational Delay (2) (msec)	Total (2) (msec)
200	0.64	3	3.64	12.5	13
400	1.28	3	4.28	12.5	13.3
800	2.56	3	5.56	12.5	14.5
1200	3.84	4.5	8.34	12.5	15.8
1600	5.12	6	11.12	12.5	17.6
3200	10.24	12.5	22.75	12.5	22.5

- Transfer time: time to transfer one block from storage into core; calculated on the basis of 3.2 nanosec/byte

- Rotational Delay (1): time to access the next record

- Total (1): Transfer time + Rotational Delay (1); time to retrieve records sequentially

- Rotational Delay (2): time for one-half of a full rotation; an average rotation time.

- Total (2): an average direct-access retrieval time.

This table was compiled by assuming that the dictionary occupies only one cylinder and hence access head positioning time (access time) is 0.

It would be desirable to use a block size of 3470 bytes because it allows the efficient utilization of the storage space. However, as the above table indicates, by reducing the block size to 1640 bytes, we reduce the direct-access retrieval time from about 23 msec to 18 msec and the buffer storage requirement by nearly 2 Kilobytes. Since this reduction in block size increases the storage waste by less than 6%, we decided to adopt the block size of 1640 bytes for the storage of the dictionaries.

The data set that is to contain a decoding dictionary is created by declaring in the program a FILE of the form

DI-CODE FILE RECORD KEYED ENVIRONMENT (INDEXED V(1640,96)).

DECODE is the file name, V indicates variable length records, and 1640 and 96 are the block and record sizes, respectively. The file is created in a sequential access mode (OPEN FILE (DECODE) SEQUENTIAL OUTPUT). The records to be written onto this file have to be supplied with their keys in ascending (numeric or alphabetic) order. Program REC which creates this file is shown in Appendix A. The decoding dictionaries created are: DECODE, DNEOP, DTOPOG, OPERD (Fig. 6.2).

A peculiarity of PL/I programs interacting with IS files is that character variables of varying length cannot be used in some record transmission statements such as WRITE (writes a record on disk). Hence every time a variable length record is to be written on disk, a fixed length variable equal to the length of the record, has to be used. In practice this is accomplished by using CONTROLLED variables and always ALLOCATE-ing a fixed length variable of the desired length. The structure containing the record is:

```
1  REC CONTROLLED,
2  DUMMY BIT (8),
2  NO CHAR (6),
2  TEXT CHAR (V),
```

where V is the length of the text entry we wish to store next. As soon as V is known, REC can be dynamically allocated. DUMMY contains the 8 bits of the dummy field while NO contains the numeric code of the entry. After the record is written, the variable REC is freed (FREE REC)

until the next record to be stored becomes available.

The association between the file DECODE declared in the program, and the data set on disk which is created through this program, is accomplished through these JCL statements:

```
//GO.DECODE DD UNIT=ONLN,DISP=(NEW,CATLG,DELETE)
//          DSN=B.BE15.DECODE,SPACE=(CYL,L,,CONTIG),
//          DCB=(DSORG=IS,KEYLEN=6,RKP=5,OPTCD=LY,
//          CYLOFL=1,RECFM=VB)
```

Briefly, the following is the meaning of the parameters. The file DECODE in the program (GO.DECODE) is associated with a data set named B.BE15.DECODE, existing on a device named ONLN. After its creation this NEW data set is to be catalogued (CATLG) unless the program fails in which case the data set is to be DELETE-d. One cylinder is requested for this data set whose organization (DSORG) is IS, keylength is 6, relative key position (RKP) in the record is 5, overflow area is 1 track within the requested cylinder, and records are of variable length and blocked (VB). The OPTCD parameter, L denotes that deletion of records is done through the D field and Y denotes that the overflow area within the cylinder is to be used.

There are two ways of expanding an already created IS data set (Anon., 1970b). If the records to be added have their keys arranged in increasing alphanumeric order, the data set is

expanded by accessing it in a SEQUENTIAL OUTPUT mode. Instead of specifying DISP=OLD, as is usually done for data sets already created, DISP=(OLD, MOD) is specified. This has the effect of removing the end-of-file (EOF) mark in the data set and allowing records to be added in continuation at the end of the file rather than over writing already existing records. The danger is that if the computer fails ("goes down") while this operation is in progress, the data can never be retrieved from this data set. To prevent this from happening a copy is made of the data set and then the copy is expanded. Several such copies may exist, with copying-and-expansion being carried out in a "round-robin" fashion.

In the second method, records can be added even if their keys are not arranged in increasing order. The data set is accessed in the DIRECT UPDATE mode. However, since most records added through this method end up in the overflow area, the number of records that can be added is limited by the size of this overflow area.

By specifying the numeric code of a dictionary entry as a key, the record with the corresponding text entry is retrieved. In IS data sets not only direct access but sequential access is also feasible. Sequential reading can start from anywhere in the data set by first positioning the reading head through a direct

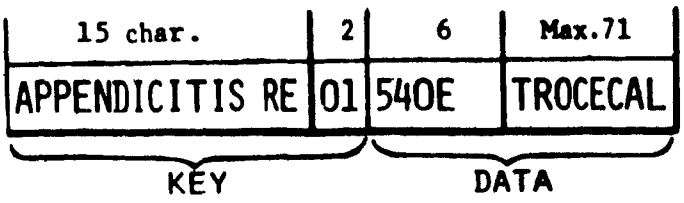
access operation.

The content of the data field in a record may be changed. The file is open in the DIRECT UPDATE mode and the record may be updated using REWRITE. The content of the key cannot be changed. If this is desired, the record has to be deleted and a new record with the new key and the old data field content must be written.

6.2.5 Coding Dictionaries. To create the coding dictionaries, the text entries have to be arranged in alphabetic order. First the records of the decoding dictionary are copied into a sequential data set (program SEQU, Appendix A). Then program SORTA sorts the records of this new data set over their text entry field. SORTA uses an IBM 360/OS sort/merge module which can be called from within a PL/I program (CALL IHESTRA.....) (Anon., 1972b). The sorted records are placed in a sequential data set called SORTOUT.

Program TOKA (Appendix A) creates the coding dictionary by reading the records from data set SORTOUT. Since the key of a record in an IS data set cannot vary in length, a key length equivalent to the longest text entry could have been chosen. However, more than 50% of the storage space within the key field would have been empty and wasted. Therefore the key field of the coding dictionaries contains only the first 15 characters of a text entry. The remaining part of the text entry and its corresponding numeric code are inserted in the data field. Since several records may exist whose first 15 characters are identical, the last 2 characters in the key field contain

a number (from 01 to 99) distinguishing between these otherwise identical keys. This is essential because the keys of an IS data set must all be different. The resulting record structure is the following:



IS data sets allow the access of records even if only a part of the key is specified. Thus it is not essential to specify the numeric part of the key. The result is that the first record is accessed in a class of entries which have in common their first 15 characters. Of course, this class may contain as few as one record. If the first record accessed does not contain the desired full text entry then one of the records sequentially following it will, if the record is to be found at all in this data set (see module PATSEK , section 6.3).

The JCL statements used in the creation of the decoding dictionary can be also used here with the exception of the KEYLEN parameter which in this case is 17. The coding dictionaries created are: CODE, CNEOP, CTOPOG and OPERC (Fig. 6.2).

6.3 The Pre-edited Pathological Report

After CISP's dictionaries were created, the programming of the batch processing modules was started. Before these modules can be described, however, it is essential to describe the form in which reports are entered.

The pre-edited report entered for coding and storage closely resembles the original pathology report. Every report entered starts with the word ID followed by the text of the report (Table 6.1 - col.A). Embedded within the report text there are a number of keywords. The most important of such keywords are called commands. At the head of each report section there is a command. The various report sections and their names have already been discussed in Chp. 2.3. Because they are easy to remember, the abbreviated names of these sections were adopted as commands (e.g. ID, CD, OP, etc). A seventh command, denoted NT (note), was added (tables 6.1, - col. B, 6.2). This command enables a non-codable observation to be entered with the report for storage.

Several keywords called headings follow the ID and PD commands. These headings denote the type of data entered within these sections. In the ID section, they specify the type of demographic data entered (Table 6.1 - col. C). The names adopted for these headings are also taken from the pathology report (Table 6.3).

The text following the heading is called the heading-text. A colon (:) is used to separate them (e.g. AGE:73).

In order to enhance the correlative retrieval capability of the system, the pathological diagnosis is entered in a specific form. First the topographic region of the body, then the actual pathological diagnosis followed by the category of the diagnosis, if applicable, are indicated. Accordingly, three PD headings exist: TOPOGRAPHY, DIAGNOSIS, CATEGORY (Tables 6.1-col.C,6.4). Each of these headings may be followed by several heading-texts. This is necessary, for example, when more than one topographic region of the body must be specified.

To code the topography entries, the 4 digit SNOP-Topography dictionary was added to the ICDA dictionary creating the DTOPOG and CTOPOG dictionary data sets. To code the category entries, nomenclature from the Termatrex dictionary was added to the Neoplasm section of ICDA. A code starting with "1C" was assigned to each of these entries.

Commands and headings must be present in the entered text so that the processing programs can determine the type of the data that are processed. Both full-word and abbreviated headings may be used. An operator entering several reports a day will probably opt to use the

abbreviated headings.

A subcommand is usually present following the command ID. The subcommand informs the supervisory program about the type of processing requested for this report. There are 4 subcommands altogether (Table 6.5). They may specify that the report following is a NEW one or that it is to REPLACE, EXTEND or DELETE an already stored report. If there is no subcommand following ID, it is assumed to be NEW.

Very few restrictions exist in entering keywords and text. Except for the command ID, no restriction exists in the number and order of commands used, including the number of times the same command is used. If a particular command (except ID) is used several times, the section contents are fused together before the data is stored. The ID headings can be used in any order. If such a heading is used more than once, only the last heading-text is retained. Since pathology reports are accessed by surgical numbers, SURGICAL NO must be one of the ID headings entered. If another command is encountered after ID but before the surgical number was read-in then the entire report is skipped.

The first PD heading must be TOPOGRAPHY and the order of headings - TOPOGRAPHY, DIAGNOSIS, CATEGORY - must be maintained.

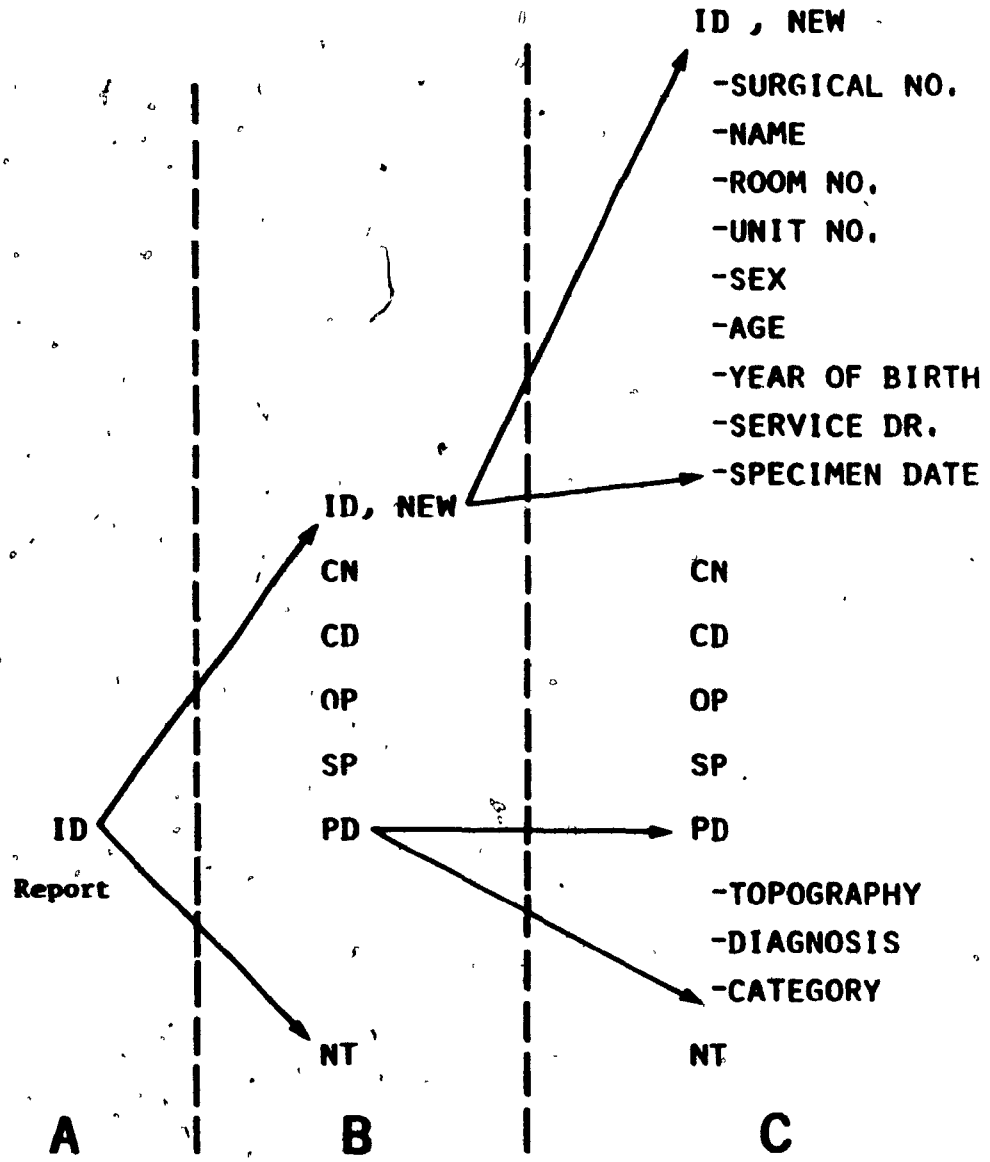
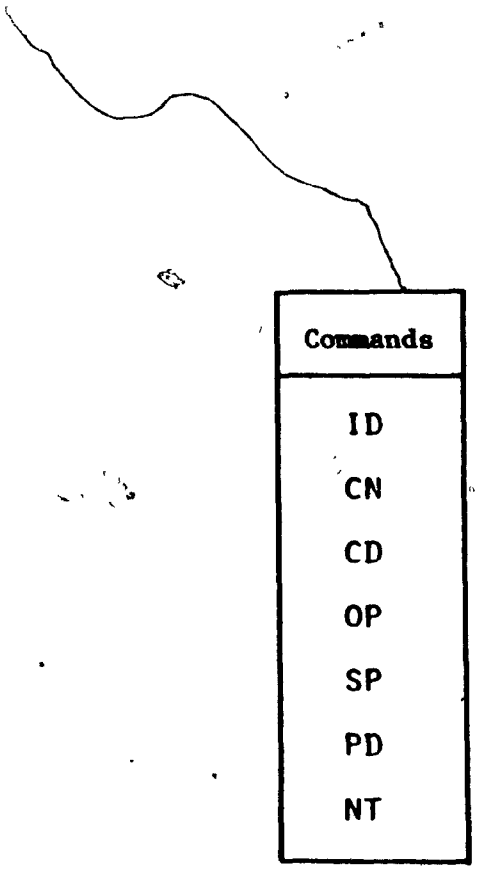

ID , NEW
-SURGICAL NO.
-NAME
-ROOM NO.
-UNIT NO.
-SEX
-AGE
-YEAR OF BIRTH
-SERVICE DR.
-SPECIMEN DATE
ID, NEW
CN
CD
OP
SP
PD
NT
CN
CD
OP
SP
PD
-TOPOGRAPHY
-DIAGNOSIS
-CATEGORY
NT
ID
Report
A
B
C

Table 6.1



Commands
ID
CN
CD
OP
SP
PD
NT

Table 6.2

ID Headings	
Full-word	Abbreviated
SURGICAL NO.	SNO
NAME	NM
ROOM NO.	RNO
UNIT NO.	UNO
SEX	SEX
AGE	AGE
YEAR OF BIRTH	YB
YR. OF BIRTH	
SERVICE DR.	SRV
SPECIMEN DATE	SPD

Table 6.3

PD Headings	
Full-word	Abbreviated
TOPOGRAPHY	TPG
DIAGNOSIS	DX
CATEGORY	CAT

Table 6.4

Subcommands	
Full-word	Abbreviated
NEW	N
REPLACE	R
EXTEND	E
DELETE	D

Table 6.5

Text and command may not be entered in the same line. Each text must follow the appropriate command and it has to be preceded and trailed by the delimiter "/". E.g.:

CN

/Excessive sweating # hyperhidrosis/

The delimiter "/" is used so that entries extending over several lines can be recognized as such. If the diagnostic entry contains entries corresponding to more than one level of the ICDA dictionary then these are separated by the symbol "#". Entries are entered with the first level entries (the more general entry) first.

Entries may be accompanied by comments which are stored without being coded. For example, the entry / headache (daily) / is stored as "781(daily)". The comment field extends the range of information that can be stored by coding with either the ICDA or SNOP dictionary. It may be used to express negation, the side of the body involved, degree of an assessment (slightly, certainly), etc. Through the use of the comment field, CISP overcomes to some extent the rigidity of expression of the ICDA and SNOP dictionaries. Other uses of the comment field are described in section 6.4

Chp. 7.2 contains two examples of pre-edited pathology reports (Figs. 7.1b, 7.2b).

6.4 Description and Programming of the Batch Processing Modules of CISP

A considerable part of the effort in the implementation of CISP went into programming of the batch processing modules. A set of these modules processes, codes and stores an entered pathology report, while another set performs the requested retrievals. Some modules are shared by both sets.

The modules will be described with the aid of flow charts (App.B). Two types of charts are used: system charts and flow diagrams (Chapin, 1970). System charts represent the general aspects of the modules. In particular they focus on the nature of input, the type of processing carried out and the output generated. Specialized equipment, such as disk or core section, are clearly indicated. Only general references are made to programs and data structures. Flow diagrams, on the other hand, are more specific. They focus on the specific data transformations that occur together with the algorithmic processes involved. Individual operations within the program and changes in variables are explicitly shown. No reference is made to specific equipment used. Generally speaking, the flow diagram tells "how" a process is carried out, while the system charts only specify "what" is done.

The modules of the system consist of main programs and subroutines. Subroutines (or procedures) programmed in PL/I can be either internal or external. Modules can be formed only from the

main programs and the external procedures. Internal procedures cannot be compiled independently and, hence, they can only exist within a module.

First the modules involved in the processing of an entered report will be described. The general processing scheme is shown in Fig. 6.3. The description of the modules (see Fig. 6.4) and their flow charts will illustrate in more detail the various processing steps shown in Fig. 6.3. During the discussion of a module, the reader may find it helpful to refer back to Fig. 6.3. in order to relate the module to the overall processing scheme shown in this figure.

The batch editor and system supervisor for the report entry operation is contained in the main program PATREAD (Fig. 6.5, 6.6, 6.7). The recognition of commands and subcommands is performed by this program. Keywords such as the ID and PD headings are recognized in modules invoked by PATREAD after command recognition.

A line containing a command may not also contain text so that scanning can be quickly performed. In addition, this ensures that the entered report is neat in appearance. Fig. 6.6a shows the batch editor section of PATREAD which is involved in the recognition of commands and subcommands. Because of the importance of the ID command, more elaborate error detection and error correction exists for this command than for the other commands. If the line read in

does not contain a command, execution continues in one of the supervisory blocks of PATREAD labelled CLAB (...).

As shown in Fig. 5.2, supervisors not only order the appropriate processing but also monitor it. Monitoring in CISP is necessary to check on erroneous processing and to detect the occurrence (or non-occurrence) of events which may become significant later on in the processing. Binary variables (DECLARE MID BIT (1)) are used as monitors in CISP. Such monitors yield fast execution when they are tested in IF statements:

```
IF MID THEN....
ELSE.....
```

If MID has the value binary 1 ('1'B) then the THEN clause is executed; if it is '0'B then ELSE executes. Accordingly, a monitor can have two states: an <u>on</u> state and an <u>off</u> state. In some system charts, the monitoring performed by monitors is self evident. Where this is not so, the function of the monitor is explained in table 6.6. Usually a monitor is tested after a subroutine is invoked in order to find out the success of the subroutine execution.

Not only monitor values are passed from one module to another but contents of variables as well. Care had to be taken in the design of the modules so that variable contents were not altered while they were being passed. Such errors, due to the internal malfunctioning of the system, can greatly reduce its reliability.

There are two methods of passing variable contents between modules. The first method consists of variable passing through an argument-parameter relationship. For example, in the program:

```
        .
        .
        .
    CALL SUB (K)
        .
        .
        .
SUB: PROCEDURE (COUNT);
        .
        .
        .
        .
```

variable contents are passed between argument K in the invoking procedure and parameter COUNT in the invoked procedure. The second method consists of using variables declared with the attribute EXTERNAL. Variables in different modules which have the same name, and are declared EXTERNAL, are recognized as being the same. In CISP these types of variables are used as storage registers for the coded sections of the report. Thus, it is not necessary to return these code containing variables to the supervisor program (PATREAD) which originally invoked the coding operation. Only the monitor need be returned for checking the success of coding. Later on, when the report is ready for storage, the appropriate variables are declared in the storing module and thus their content becomes available. Fig. 6.4 shows the method chosen for passing variable contents between various modules.

The most complex supervisor block in PATREAD is CLAB (1) which is entered while command ID is in effect (Fig. 6.6b), and occurs when a new report is encountered or during the processing of the demographic entries of a report. The complexity of this block is due partly to the large number of monitors checked here. This is done, for example, when a coded report is prepared for storage. If n monitors have to be checked, 2^n-1 IF statements have to exist to test all monitor state combinations. Thus, 4 monitors need 15 IF statements to take into consideration all possible combinations.

Whenever block CLAB (1) is entered in the course of processing ID headings, module ID is invoked. This module supervises the recognition of headings and invokes the heading-text processing modules. First ID (Fig. 6.8a) prepares the string which ought to contain the heading. Recognition of the heading is performed by module IDHED (Fig. 6.9) which is basically a table look-up program. As mentioned in chp. 6.2 it is most likely that the abbreviated ID headings will be entered by the operator. Therefore, IDHED was programmed to execute fastest when abbreviations are used. IDHED looks up the table of fullword headings only after look up of the table of abbreviations fails. Certain headings such as "SPECIMEN DATE" contain more than one word (composite heading). The table in IDHED usually contains only the first word. The presence of the second word of the heading is investigated by module NO (Fig. 6.10).

NO is a text comparator module. Both the text and the text to be compared with (comparator) are passed to NO. NO carries out the comparison and issues error messages if necessary.

Identification of a heading is followed by the processing of the heading-text. To process a heading-text which contains the Surgical Number, the module MOSNO (Fig. 6.11) is invoked by ID. If the processing of the Surgical Number fails, the entire pathology report must be skipped. In order to avoid this whenever it is possible, extensive error detection and correction routines were built into MOSNO. From a typical surgical number such as S.72-9560 only the numeric fields are retained (at most 7 characters) since the rest are redundant. MOSNO will accept surgical numbers without the 'S.' or the '-' in it, or with blanks embedded between the digits. CISP's operator should not take advantage of these error correction and detection routines by introducing laxity in the entered report because the processing may prove costly.

The name text is processed by the module MONM (Fig. 6.12). An example of the accepted form of a name entry is: DOMEK, MR. JOHN K. Entering names in this form allows for the identification of the surname and christian names. Thus retrieval of a patient's report is possible even if only his last name is specified.

The texts following the headings RNO, UNO, SEX, AGE, YB and SRV are processed within module ID (Fig. 6.8c, 6.8d). The texts of UNO, AGE and YB are accepted only if they are numeric. The sex text is tested for the two conventional possibilities. If the SRV text contains the name of the physician, his last name is recorded first. As in all names, periods after initials are redundant and are left out.

The specimen date is transformed to a 6 character "date code" by module MOSPD (Fig. 6.13). The first 2 characters denote the month, the next 2 the day and the next 2 the year.

The PATREAD supervisory blocks labelled CLAB(2), CLAB(3) and CLAB(4) are very similar with respect to the nature of processing carried out within them (See Fig. 6.6c). CLAB(2) codes the CN entries, CLAB(3) the CD entries and CLAB(4) the OP entries. As soon as the execution is transferred to one of these supervisory blocks, the variables that will contain the coded entries are prepared, the dictionary to be looked up is chosen and PATSEK is called. PATSEK (Fig. 6.14) is the module invoked whenever a disease entry has to be coded through the look up of one of the externally stored dictionaries. After the entries are coded they are assembled in the respective EXTERNAL variables which are used again later during the storage of the report.

The file name associated with the dictionary that must be searched is passed as an argument to PATSEK. The EXTERNAL variables containing the coded entries are also passed to PATSEK as arguments rather than as EXTERNAL variables. This way only one variable has to be declared within PATSEK for holding codes, reducing variable storage requirements from 2000 bytes to only 400 bytes.

Although the dictionary look up operation is quite a simple one, PATSEK is a large module (300 PL/I source statements). Part of the program prepares the entry for look up. If the delimiter "#" is present in the text, the entries at different levels are separated. Also, the comment field, if present, is separated from the entry-text. Before the diagnostic entry is looked up in the dictionary it is reorganized by module SPLIT (Fig. 6.15). The operation performed by this module is shown in its system chart (Fig. 6.15a). The text of the entries stored in the dictionary are also first reorganized by SPLIT. This type of reorganization is necessary because a dictionary look up can only be successful if the dictionary entry and the diagnostic entry match character-by-character. Module SPLIT may also be used to compress text through the elimination of unnecessary blanks.

The subroutine LOOKUP (Fig. 6.14d), an internal sub-routine to module PATSEK, performs the coding dictionary look-up

operation. Using the PL/I file environment GENKEY feature (available only with IS files) an appropriate dictionary entry can be directly retrieved by specifying only part of the record key. If the full dictionary entry does not march the diagnostic entry, a few of the "near by" dictionary records are sequentially retrieved until the entries match (see section 6.2 and Fig. 6.14d).

If the entry consisted of more than one level then the codes obtained have to be matched (section 3.4). For example, if the level 1 entry yielded codes 5031 and 460 and the level 2 entry yielded 5071, 4601, 231 then only the codes 460 and 4601 form a matched set. This matching is performed by the internal procedure TWO - for 2 entry levels -, and THREE - for 3 entry levels. The more complex algorithm of THREE is shown in Fig. 6.14e.

CLAB(5) (Fig. 6.6d) is the block supervising the reading in of the specimen description (SP entries). These entries are not coded and are only retained in the report while the patient is in the hospital. In order to compress the amount of data stored, the trailing blanks are cut-off and a line skip delimiter is inserted instead.

The PD headings and texts are processed by module PAT which is similar to module ID. PAT edits the PD headings and supervises the coding of the entries (Fig. 6.16).

The SNOP dictionary is looked up to code the topography entries and the ICDA dictionary to code the diagnosis and category entries. In spite of the different hierarchical organization of SNOP and ICDA (see Chp. 3), it was found that with minor modifications the already existing PATSEK could be used to look up both dictionaries. An entry that is to be coded using SNOP consists of only a one level entry. Hence PATSEK skips the code matching process which is only necessary to accomodate ICDA's hierarchical structuring.

Since the three data items - topography, diagnosis and category form a complete pathological diagnosis, they have to be entered together and in the proper sequence. If an error occurs in processing any of these items, the processing of the report is skipped till the next topography item or the next report section is encountered. The corrected version of the PD section left out may be added to the already stored report using the subcommand REPLACE or EXTEND.

Some category entries contain not only the entries codable through the standard dictionary, but also a modifier entry. These modifiers are underlined in the following examples:

/ no lymph node involvement/

/ metastatic to liver/

/ size 2 mm/

PAT searches these statements in an attempt to find within them such category entries as "size", "metastatic to", "grade", etc. (Fig 6.16b, 6.16c, 6.16d). The adjoining modifiers are inserted into an internally created comment field which is attached to the code of the respective category entry. Thus /no lymph node involvement/ becomes 1C80(NO). The entry /metastatic to liver/ contains the topographic site modifier "liver". PATSEK looks up this site entry in the SNOP dictionary yielding the code 5600. The entire category entry is coded as 1C81(5600).

The coded pathology report is assembled and stored by module STORE (Fig. 6.17) in data set PATREP. If the report was entered with subcommand EXTEND or REPLACE then the previously stored report is read first. If an entry in a section is to be replaced, the entire corrected section has to be entered because individual entries cannot be replaced in this prototype version of CISP.

Printing of the report is carried out by module PRINT (Fig. 6.18). It was found that for checking purposes it is enough to print just the coded version of the report, report. If a more elaborate checking of the entered report is necessary, then the operator can print it out in the decoded form.

SUBSTRING, a PL/I built-in function was used in programming most of these modules. Great care had to be exercised in using SUBSTRING. If SUBSTRING was made to refer to a part of a string lying outside the boundaries of the variable concerned, damage could be caused to areas of the core where the program machine code instructions reside. This may result in malfunction which can damage the data base. Therefore, in the programming of CISP modules, the use of SUBSTRING was preceded by a test of the string length within the variable involved.

Since the modules of CISP will need to execute several times a day, it is important that execution cost be as low as possible. Therefore, these modules were compiled at the optimizing level 2 (OPT=2) which yields modules that execute fast at the expense of a larger storage area needed to hold them.

The subroutine modules described above are link edited to form separate special purpose load module programs. Thus, one load module program codes a report, another retrieves a report in decoded text form, a third retrieves the coded version of the report, and so on. Several of these programs may contain copies of the same CISP subroutine modules. For example, IDHED is used in the program which codes reports and in the retrieval programs where it recognizes the ID heading by which a report is to be retrieved.

The execution of a load module program is initiated through a JCL program of the form:

```
//EXEC PGM=REPCOD
//STEPLIB DD DSN=B.BE15.PATLIB,DISP=SHR
```

In this example, REPCOD is the program which codes a report. It consists of the main program module PATREAD and the sub-routine modules it invokes.

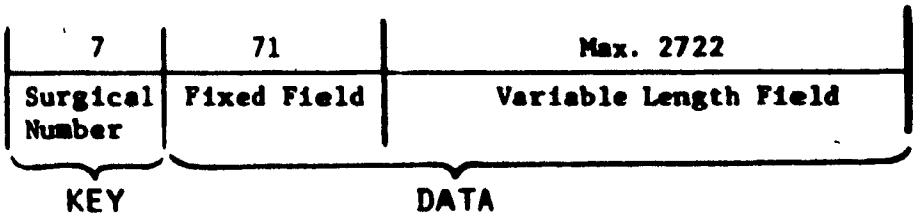
It would be feasible to simply have one program module containing all these special purpose program modules, together with a small batch editor that would choose the appropriate processing to be performed. However, this is not practical because of the large storage area needed by such a module and the large core area needed for execution. Therefore, in CISP these programs are kept separately and the program which is to be executed is selected during the interactive processing phase of CISP.

6.5 The Organization of the Coded Pathology Reports

PATREP, the data set containing the coded pathology reports, is the equivalent of the pathology report file in the pathology department. The data set organization adopted for PATREP is indexed-sequential with variable length records. The

creation of PATREP is similar to that of the dictionaries (Chp. 6.2)., except that the block size is 3470 and the record size is 2800 bytes. The cylinder overflow area extends to half the cylinder size (10 tracks) in order to accommodate the possible numerous updatings (rewritings) of the records.

Each record in the data set contains one pathology report in coded form with the surgical number serving as the record key. The record structure is the following:



The fixed length field consists of 8 fixed length subfields which accommodate the processed version of the texts following the ID headings. The following are the subfields, including the record key, and their respective length in characters:

SNO	7
NM	25
RNO	5
UNO	7
SEX	1
AGE	3
YB	4
SRV	20
SPD	6

Fixed fields were adopted for these entries because they are present in most of the reports entered and thus very little storage space is

wasted. A desired item can be retrieved from this field by knowing its location. Thus for example, patient age is found starting with the 39th character and extending till the 41st character of the field.

A number of delimiters are embedded in the variable length field of the record in order to separate one code from another and the various report sections. Thus, each section in the record forms a subfield which starts with a 4 character field of the form *KK*, where KK is the command of that section. As a result the general structure of the variable length field is:

CN...*CD*...*OP*...*PD*...*NT*...*SP*.....*

The codes in these subfields are delimited by the symbol #. A CN subfield may have the form:

CN 791(daily) #7843A

The delimiters T, D and C are used to delimit the topography, diagnosis and category codes of the PD section. An example of a PD section is:

*PD*T<5840>D<5540#2080>C<1C80(NO)#1C81(6387)

#1C10>T<5503>D<2301#503>C<1C82(3MM)>

Chapter 7.2 contains examples of coded pathology reports (Fig. 7.1c, 7.2c).

6.6 The Programming of the Interactive Processing Phase

The programming of the interactive processing phase of CISP was carried out by adapting the CRJE editor to the needs of the CISP's operator. Several sequences of CRJE editor commands were collected in CRJE data sets called CLIST (command list) (Anon., 1970a). The sequence of commands in a CLIST data set may be executed by typing in the command EXEC (or X) followed by the data set name. For example, when:

```
EXEC  CODE1
```

is typed in then the following commands will execute:

```
SUBMIT  REPCOD, ONE, FINISH
EDIT MSG
LIST, NONUM
END
```

The first command submits a program for execution. REPCOD contains the JCL program which calls for the execution of the load module program which codes the pathology report. The report to be processed is to be found in the CRJE data set ONE. FINISH contains the end-of-program delimiter /*. After the program is submitted, the message contained in data set MSG is printed for the operator:

```
REPORT (S) IN DATA SET "ONE" WERE SUBMITTED FOR PROCESSING.
```

Fig. 6.19 illustrates the various interactive processing activities which are at the disposal of an operator using CISP. The operating instructions do not have to be memorized by the operator

because on request they are printed at the terminal. (See Appendix C for the operating instructions).

6.7 Maintenance of the Data Base and Program Library in CISP

In the course of time, numerous alterations of CISP's program library and data base will occur. The effect of these alterations is to clutter the storage areas available for these data sets. Even the normal daily operations of CISP consisting of the addition of new pathology reports will necessitate the regular reorganization of the pathology report file. There are a number of maintenance programs which are run for this purpose.

Updatings of the dictionaries and of the stored pathology reports quickly fill the overflow area of these IS data sets. Unless the overflow records are inserted into the prime area leaving the overflow area empty, further updatings of the data set may not be performed. This is accomplished by re-copying the data sets.

Programs TRANS and TOTRANS (Appendix D)* carry out the recopying of the decoding and coding dictionaries, respectively. With minor parameter alterations, TRANS is also used to recopy

---

*All programs mentioned by name in this section are shown in Appendix D.

the data set containing the coded reports (PATREP). Program RENAME is used after recopying to change the name of the new data set back to that of the old one.

As the number of stored pathology reports becomes large, it may become impossible to keep all these reports on disk. Older reports may be stored on tape, off-line. There are IBM utility programs which copy the content of an IS data set onto tape (unloading), and, when needed, recreate the IS data set on disk from the information stored on tape (loading). Thus, old reports can easily become "active" again and available for on-line retrieval.

Often, it is necessary to consult the dictionaries used by the system. Programs BLOW and TOBLOW produce paper copies of CISP's dictionaries. Due to the bulk of these dictionaries, their content is printed on the high speed printer of the computing centre.

A different set of programs is needed to maintain and expand CISP's library of programs. Each new module is compiled using program JCOPT which produces modules optimized for fast execution. Program LIB creates a partitioned library, LIN adds new modules to the library and LINREP replaces an existing module with its improved version. The addition and replacement of modules is limited by the size of the library's directory which is fixed during the creation of the library. If the directory of the library

is full, the library can be recreated using program LOADCOPY. This program allows selective copying of the old library.

Since replaced modules are not overwritten, program COMPRESS should be used to eliminate unused spaces. Program RELEASE may also be used to release storage space assigned to the library but not utilized.

Those PL/I source programs which are not used are stored, in the source form, on an external data set using program SAVE. Should the need arise to update an old load module, the respective source program is read back into the CRJE library where it is altered and the load module produced is entered into the library as a replacement.

Table 6.6

PATREAD		
CONS	On signals that:	a surgical number has been encountered in this report
MONP		an error in coding PD entries; it initiates a search for the next TPG heading or command
MID		the line read in contains command ID
MER		subcommand could not be identified; print last report and skip this one entirely.
MFP		ID was not met at the head of the 1st report entered.
MRUN		error occured in the analysis of the SNO text, or error occured in identifying an ID heading before SNO was read in.
FPR		the PD command was already encountered but the PD codes were not yet collected in the PD variable
MV		see MON (PATSEK)
ID		
MRUN		see MRUN (PATREAD)
MTH		error occured in the identifying an ID heading before an SNO was read in
MON		the ID heading text was correctly analyzed.
CONS		see CONS (PATREAD)
IDHED		
MTH		see MTH (ID)
CONS		see CONS (PATREAD)
PATSEK		
MQ	Off signals that:	coding error is due to incorrect use of entry delimiters.
MON	On signals that:	an entry was correctly looked up in the dictionary.
MST		both coding and the matching of the different level entry codes was correctly performed
M1		the 1st level entry was correctly looked up in the dictionary
M2		the 2nd level entry was correctly looked up in the dictionary.
M3		the 3rd level entry was correctly looked up in the dictionary.
MTHREE		matching of the codes of a 3 level entry was correctly performed.

Table 6.6 contd

PAT

MONP	On signals that:	see MONP (PATREAD)
MP		see MON (PATSEK)
MQ		see MQ (PATSEK)

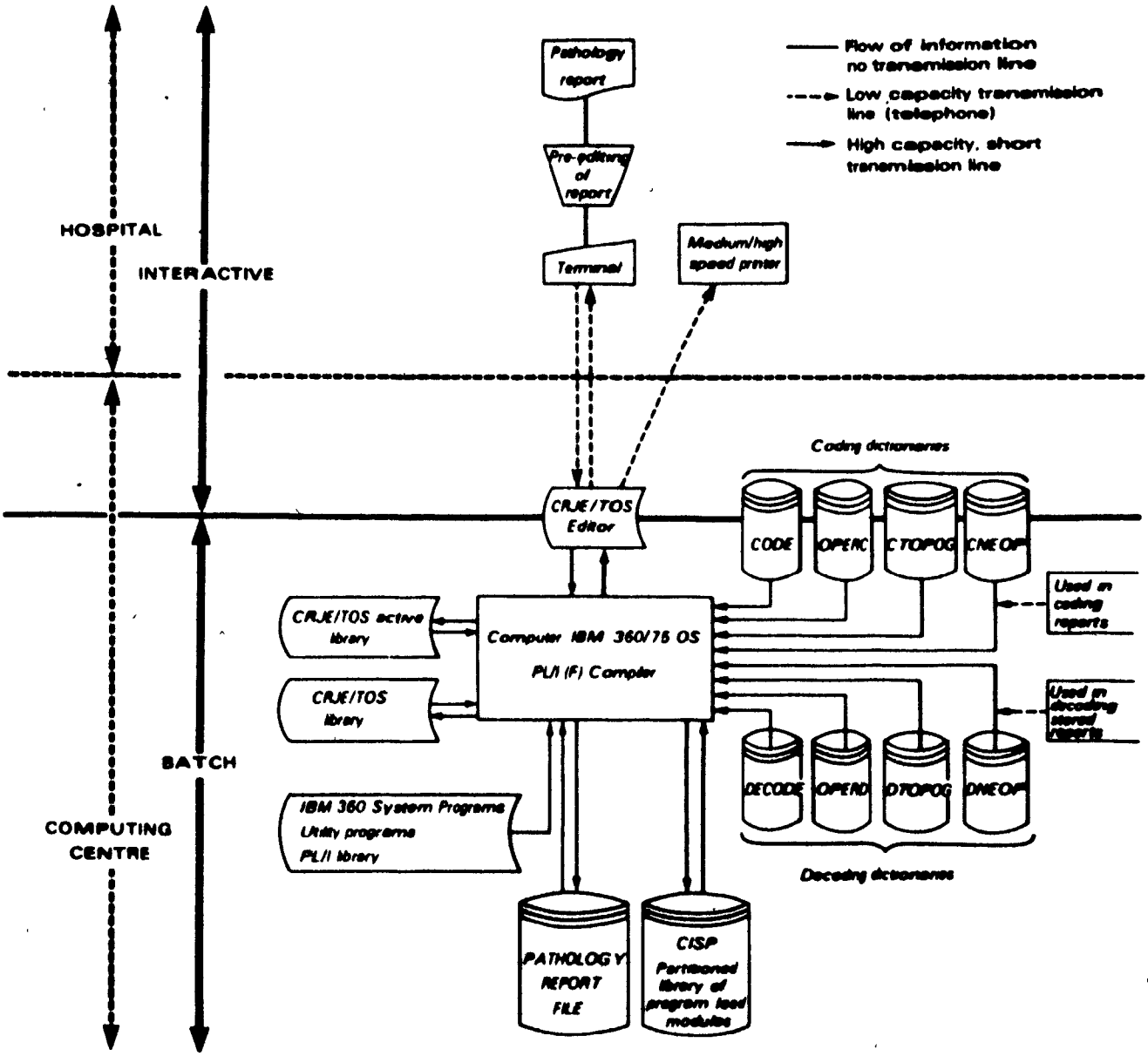


Fig. 6.2 - Diagram of the communication between the hospital, the computer system and CISP.

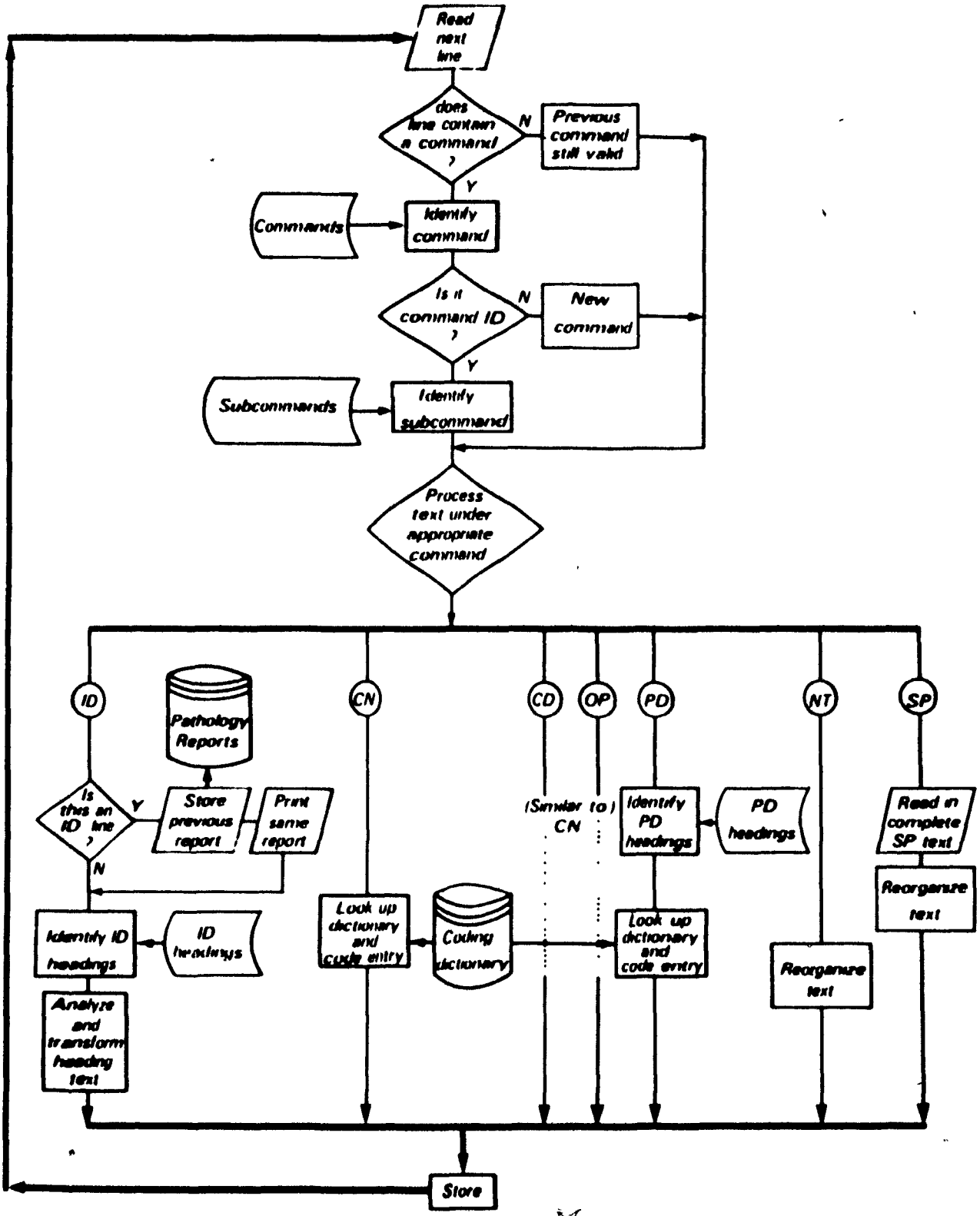


Fig. 6.3 - System chart of the processing of an entered report.

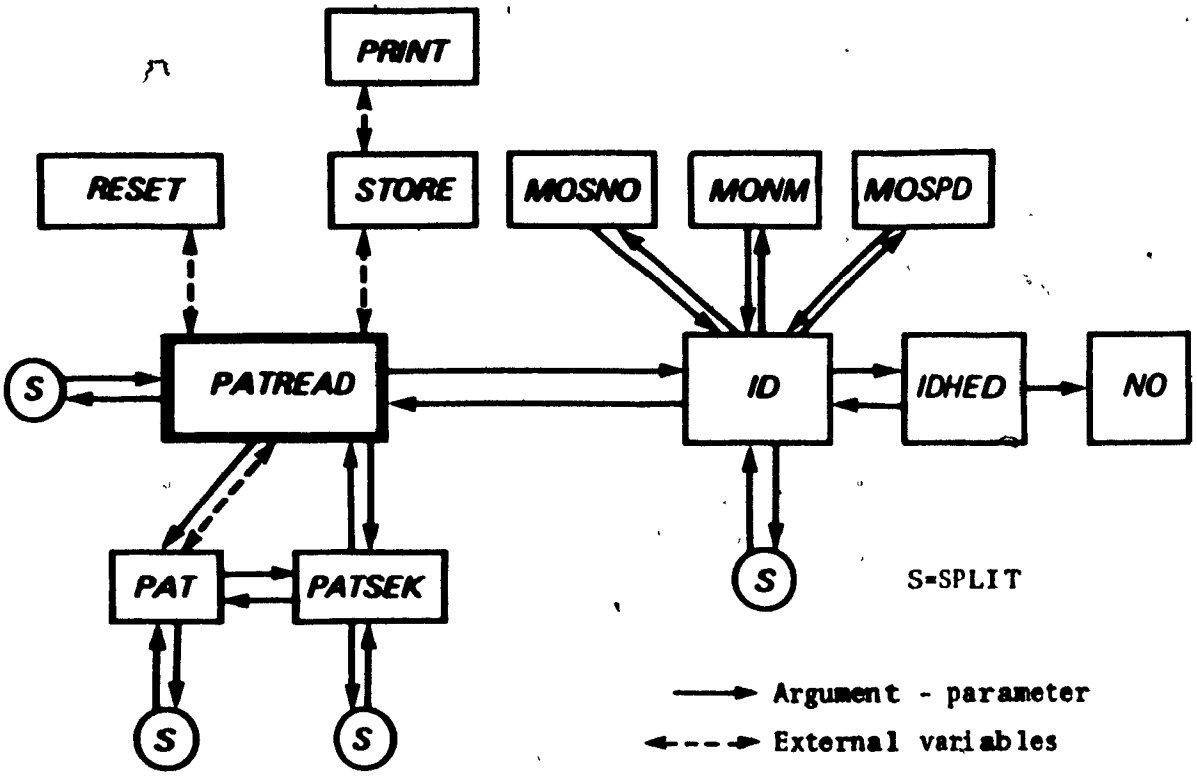


Fig. 6.4 - Diagram of communication between the modules involved in processing an entered report.

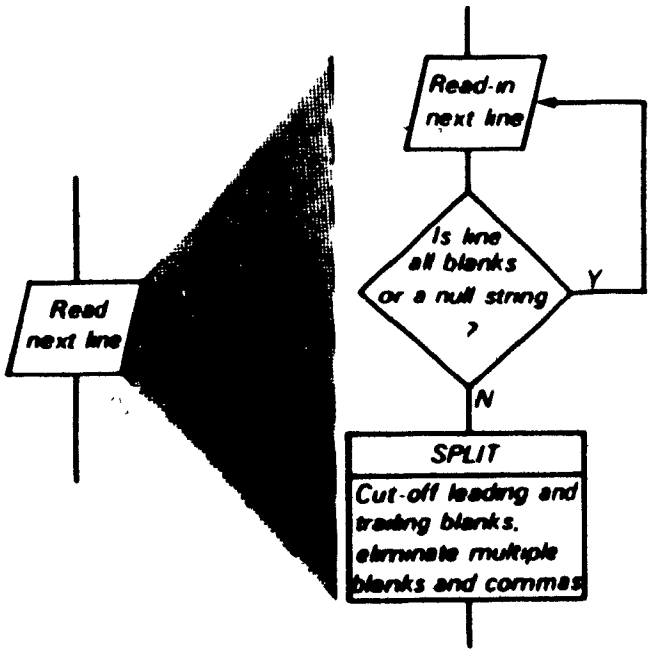
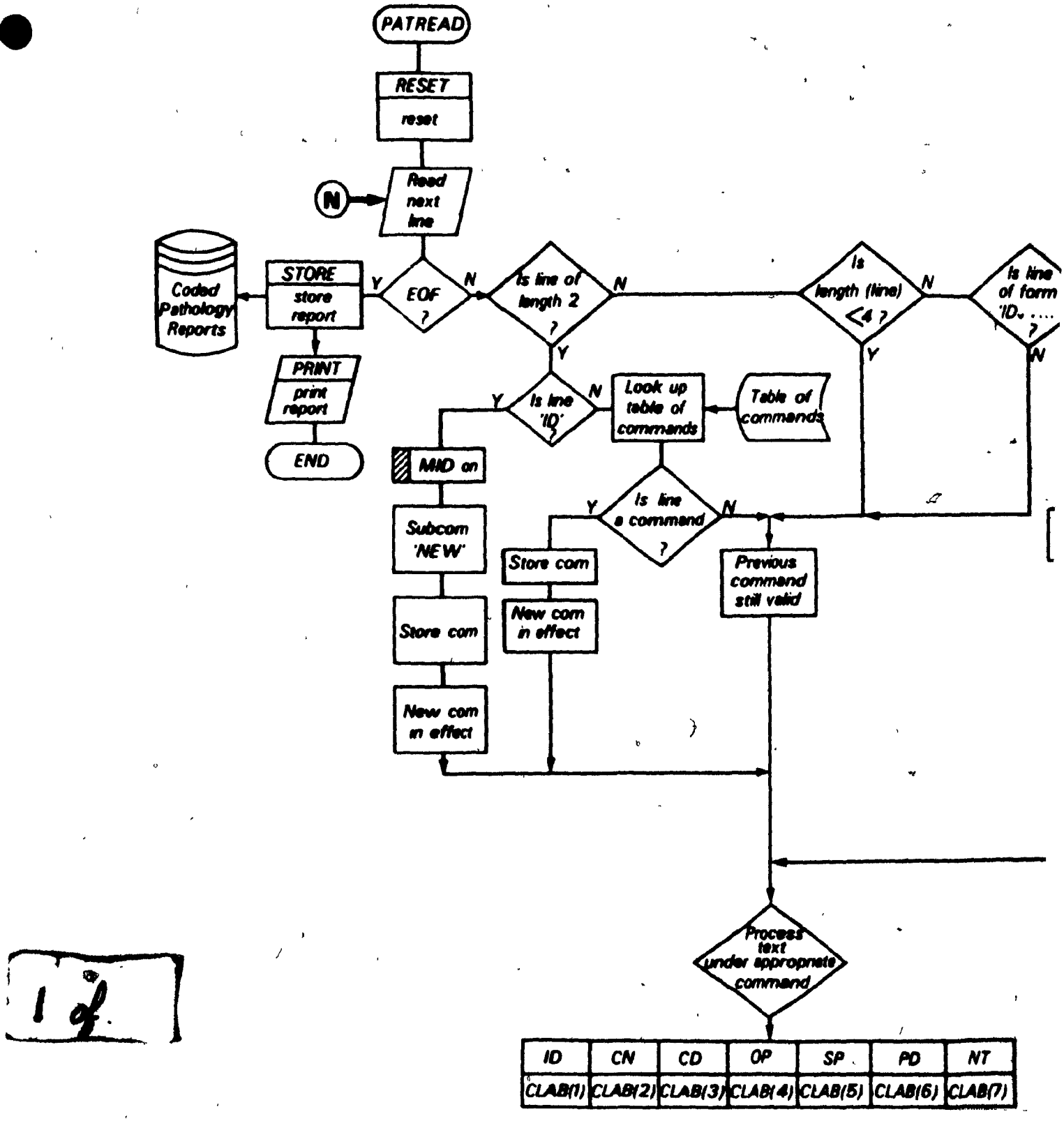
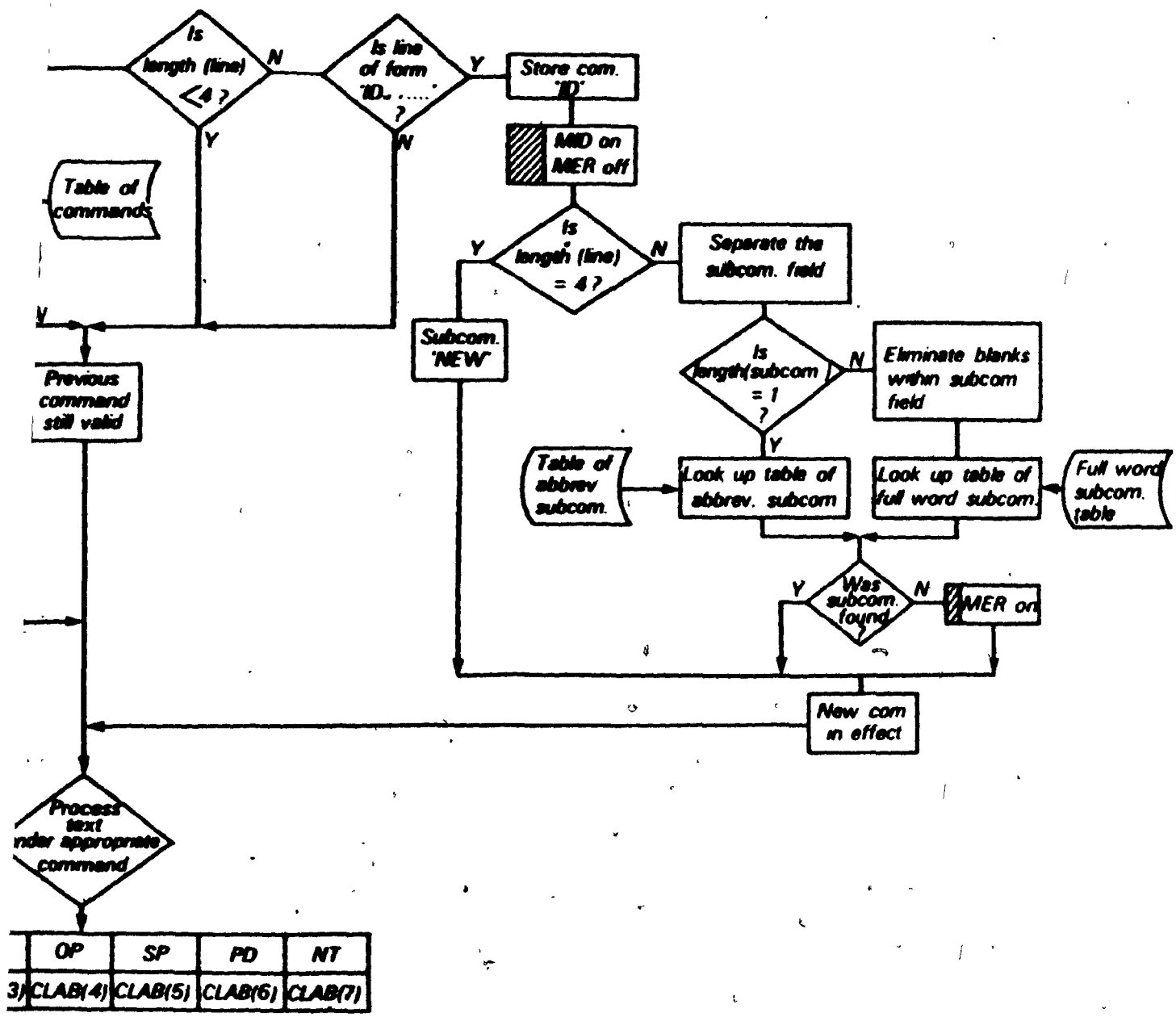


Fig. 6.5 - Convention for the meaning of "read next line" in a system chart.



1 of

Fig. 6.6a - System Chart for Module PATREAD


Is length (line) <4 ?
N
Y
Is line of form 'ID.....' ?
Y
N
Store com. 'ID'
MID on MER off
Table of commands
Is length (line) = 4 ?
Y
N
Separate the subcom. field
Subcom. 'NEW'
Is length(subcom = 1 ?
N
Eliminate blanks within subcom field
Previous command still valid
Table of abbrev subcom.
Look up table of abbrev. subcom
Look up table of full word subcom.
Full word subcom. table
Was subcom. found ?
Y
N
MER on
New com in effect
Process text under appropriate command
OP
SP
PD
NT
3)
CLAB(4)
CLAB(5)
CLAB(6)
CLAB(7)

[redacted]

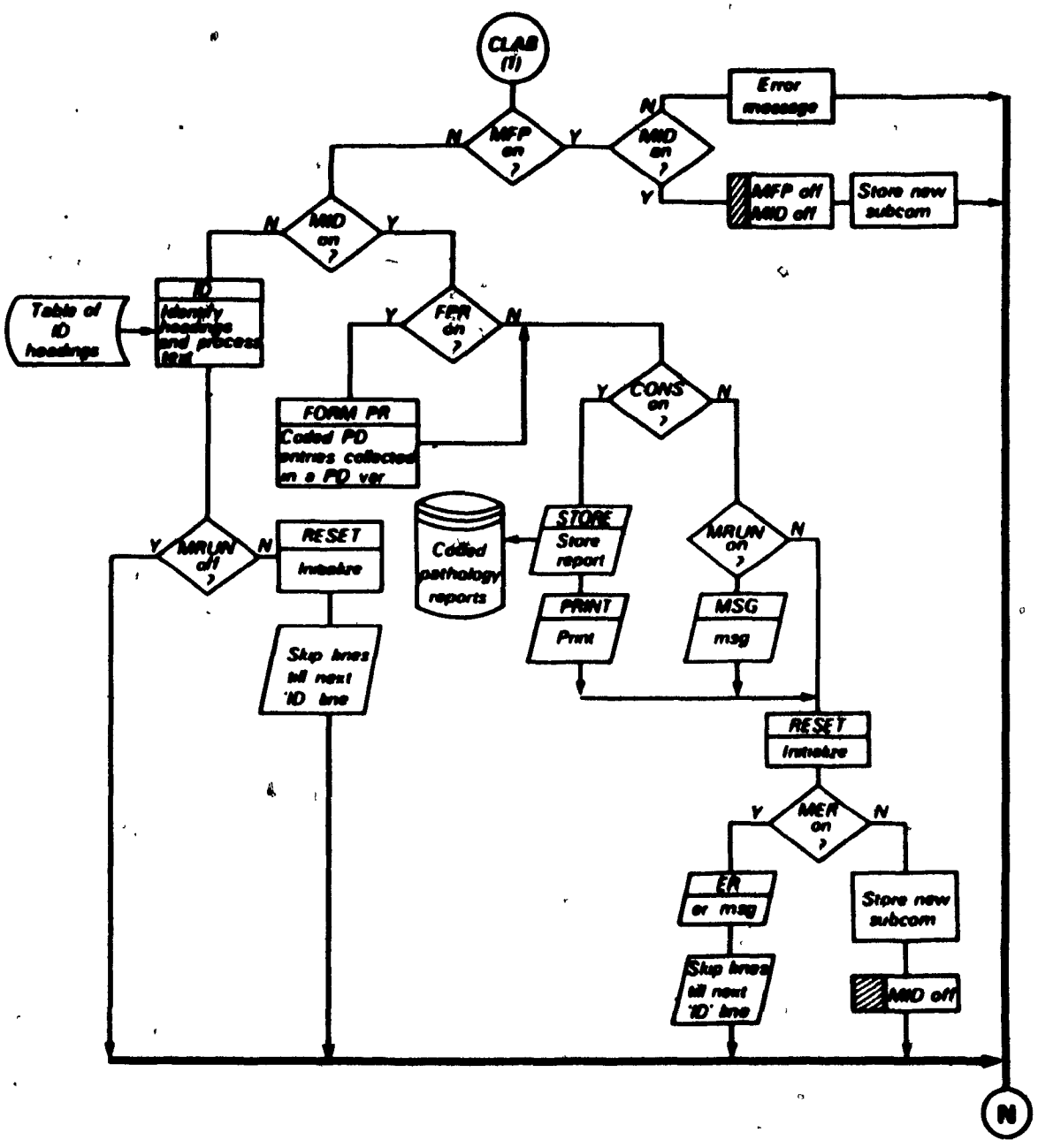
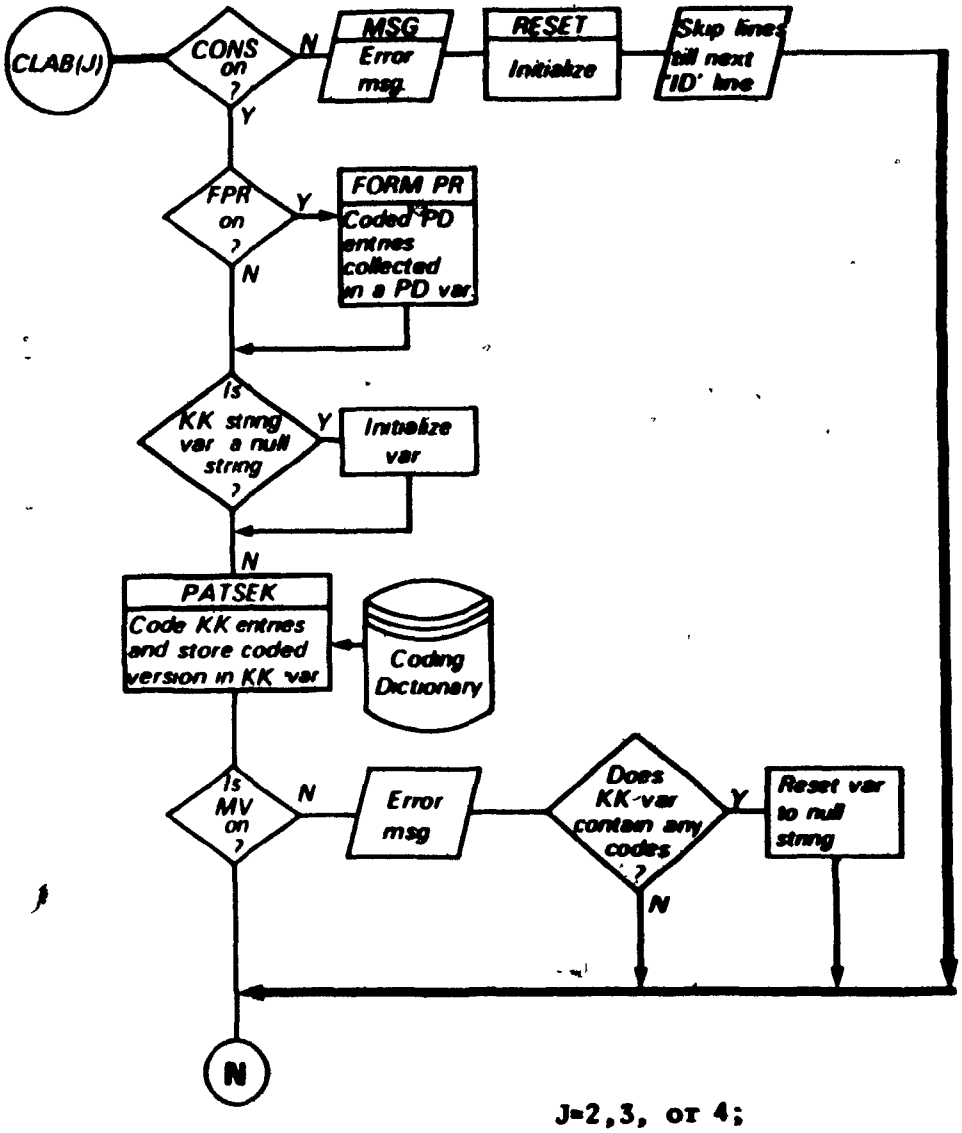


Fig. 6.6b - Module PATREAD



J=2,3, or 4;

KK=CN,CD, or OP, respectively

Fig. 6.6c - Module PATREAD

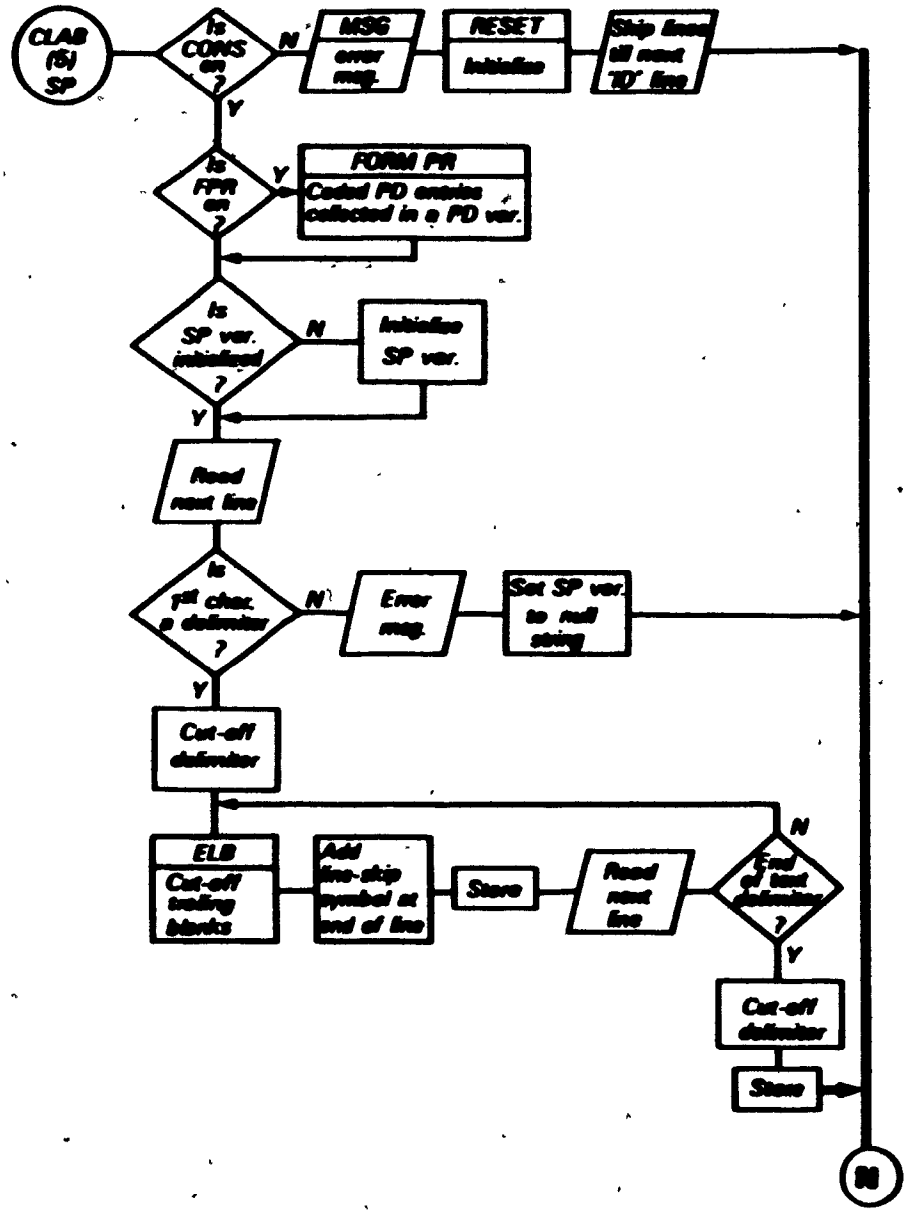


Fig. 6.6d - Module PATREAD

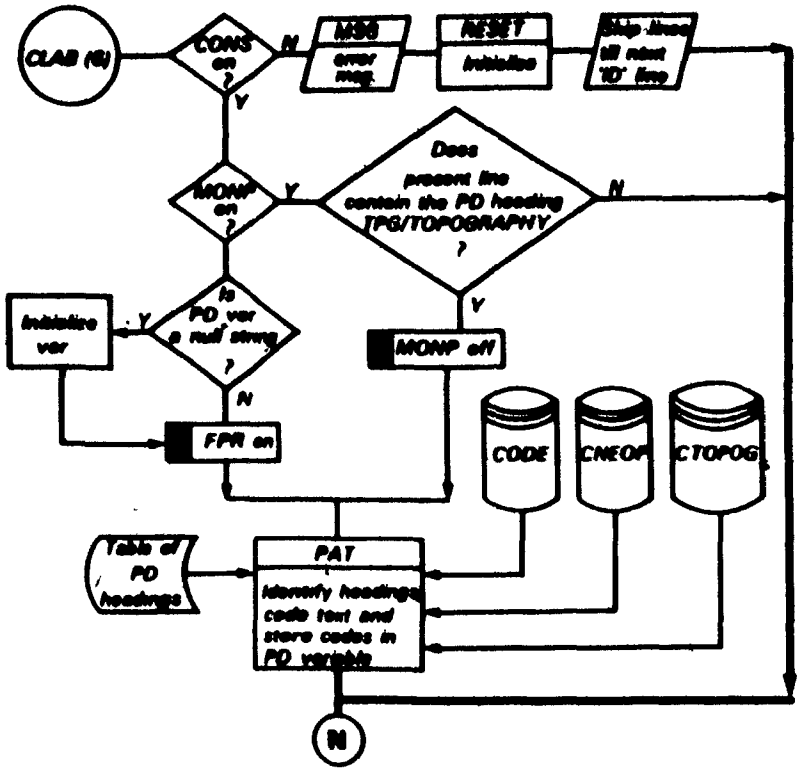


Fig. 6.6e - Module PATREAD

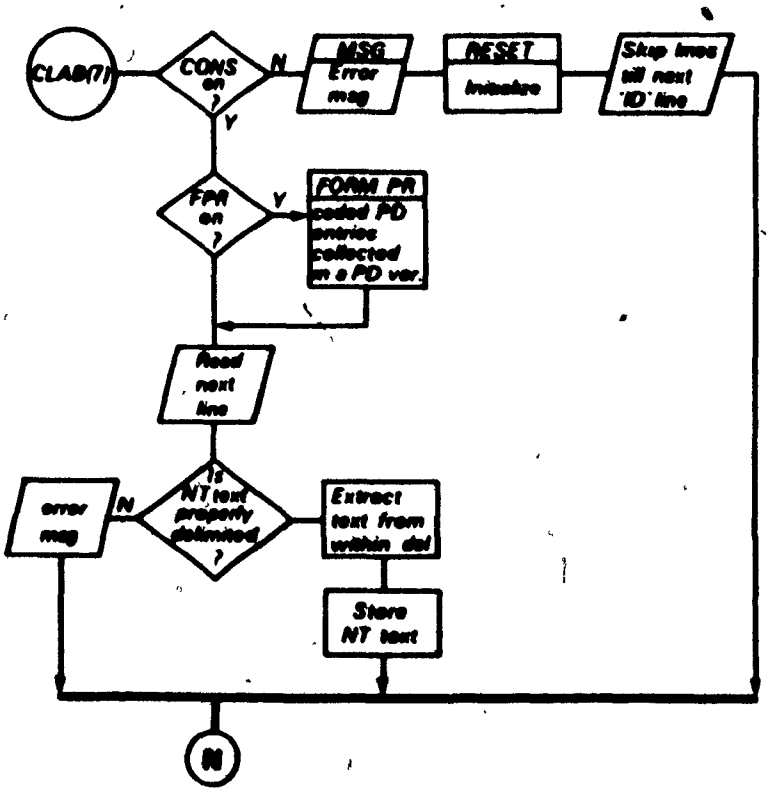
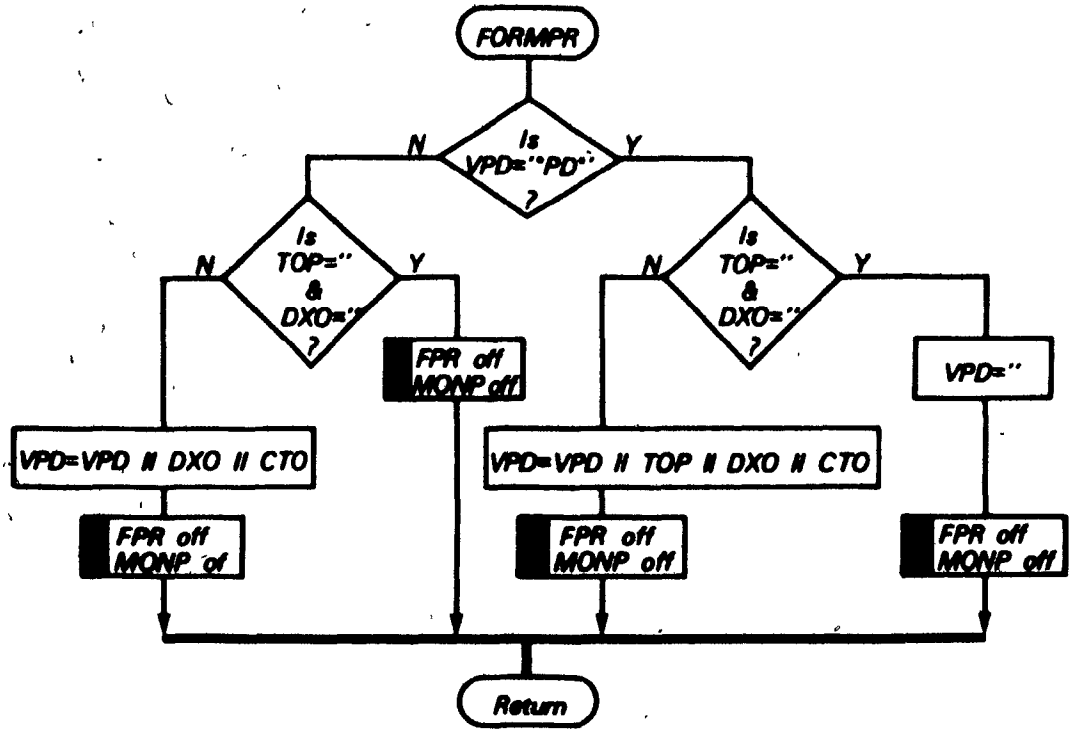


Fig. 6.6f - Module PATREAD



VPD = variable containing coded PD
TOP, DXO, CTO = variables containing the TPG, DX and CAT codes, respectively

Fig. 6.7 - Flow diagram of subroutine FORMPR, internal to PATREAD.

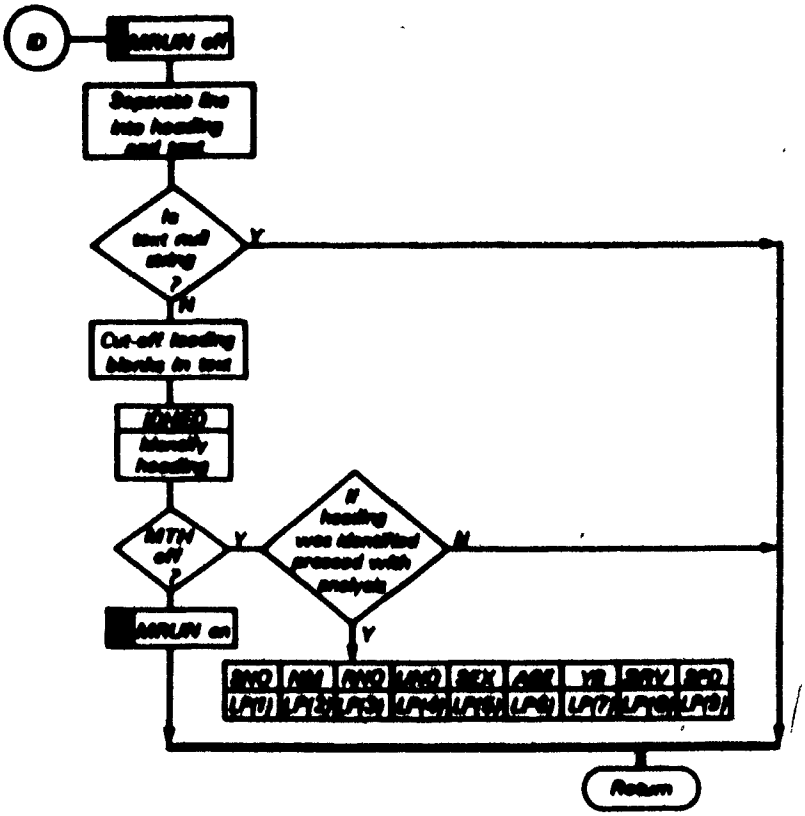
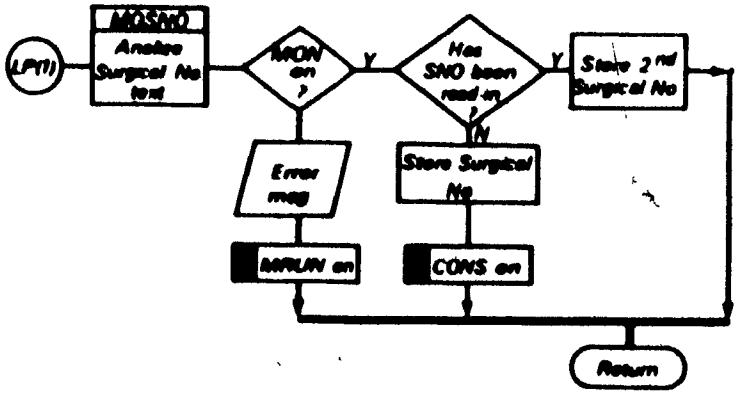


Fig. 6.8a - System chart of module ID



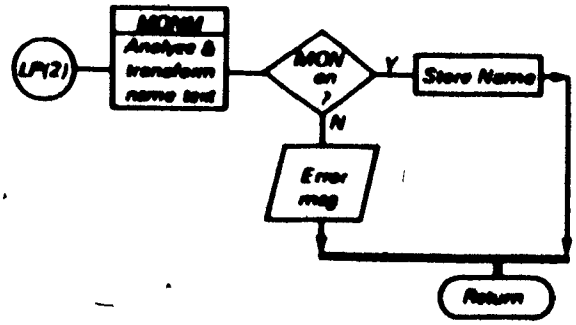


Fig. 6.8b - Module ID

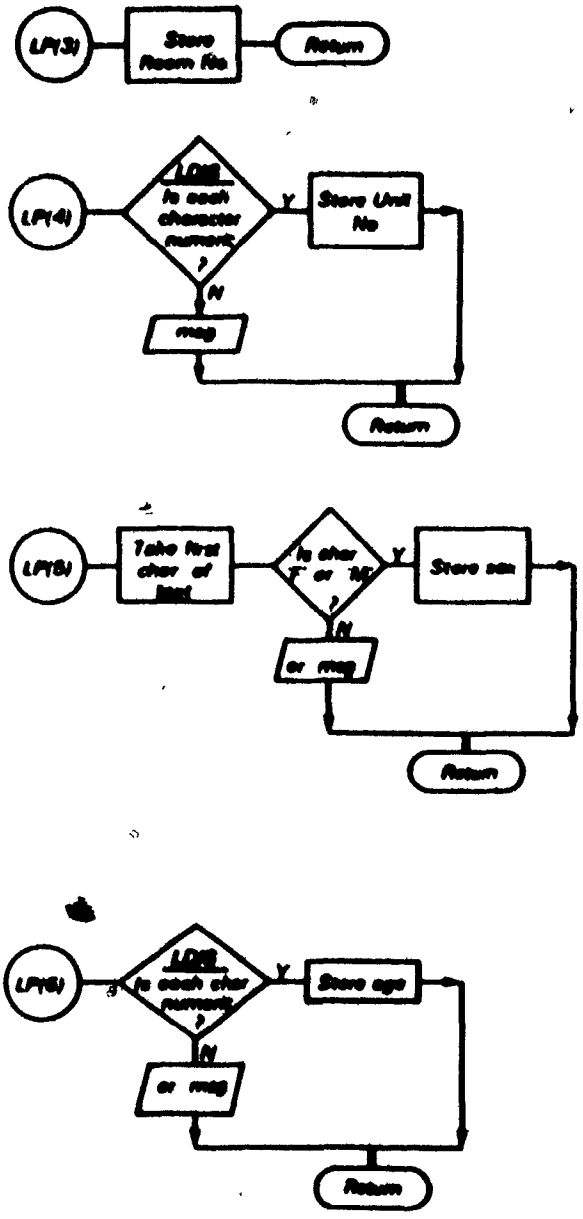
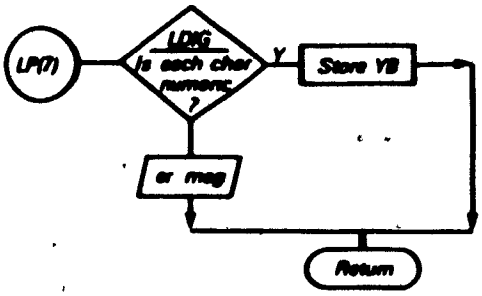
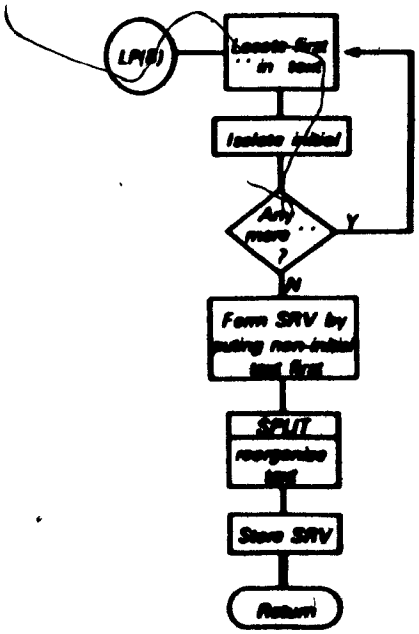


Fig. 6.8c - Module ID





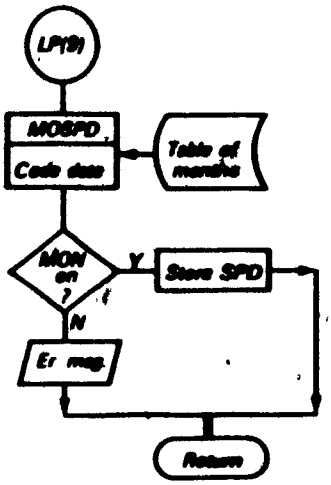


Fig. 6.8d - Module ID

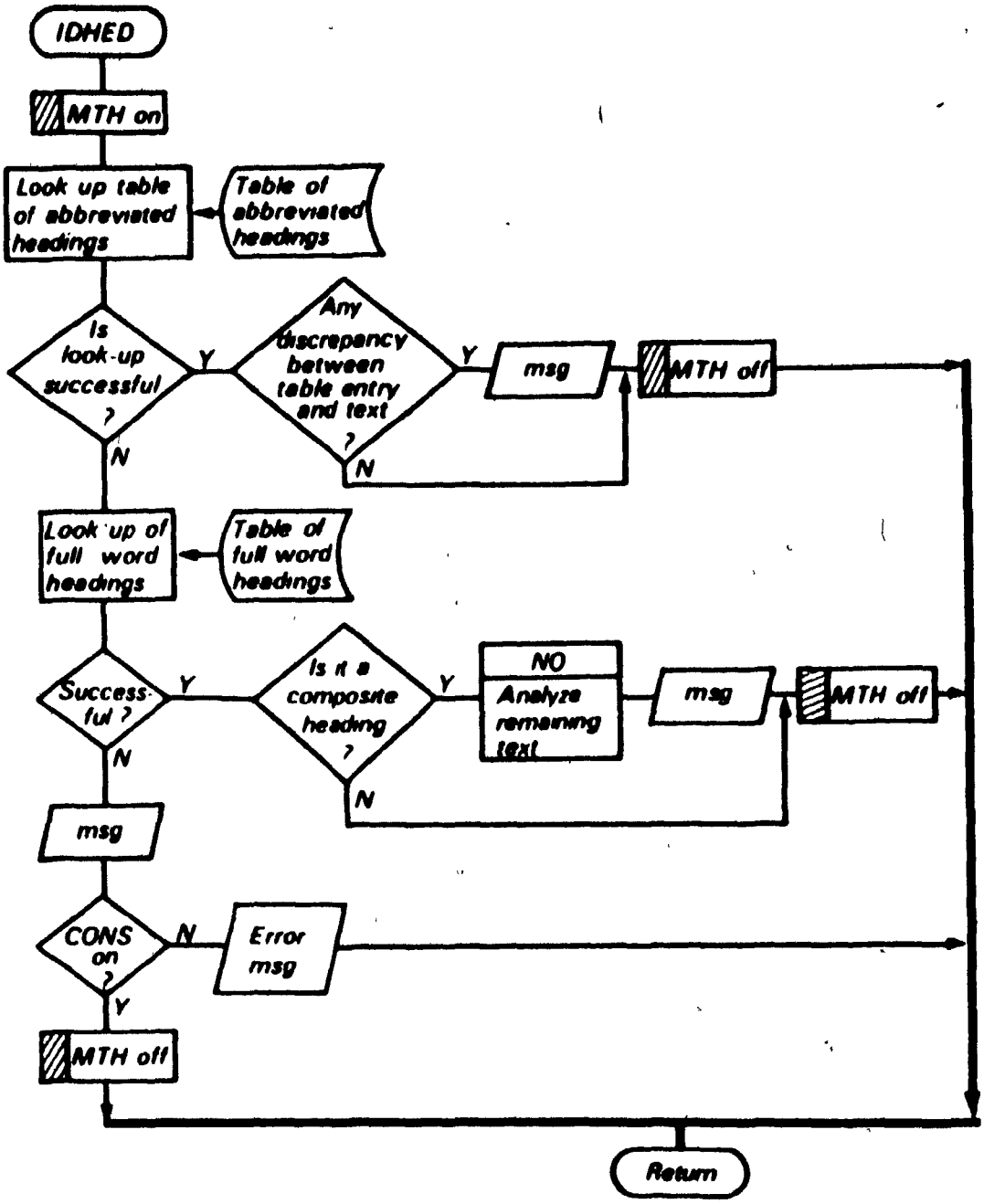


Fig. 6.9 System chart of module IDHED

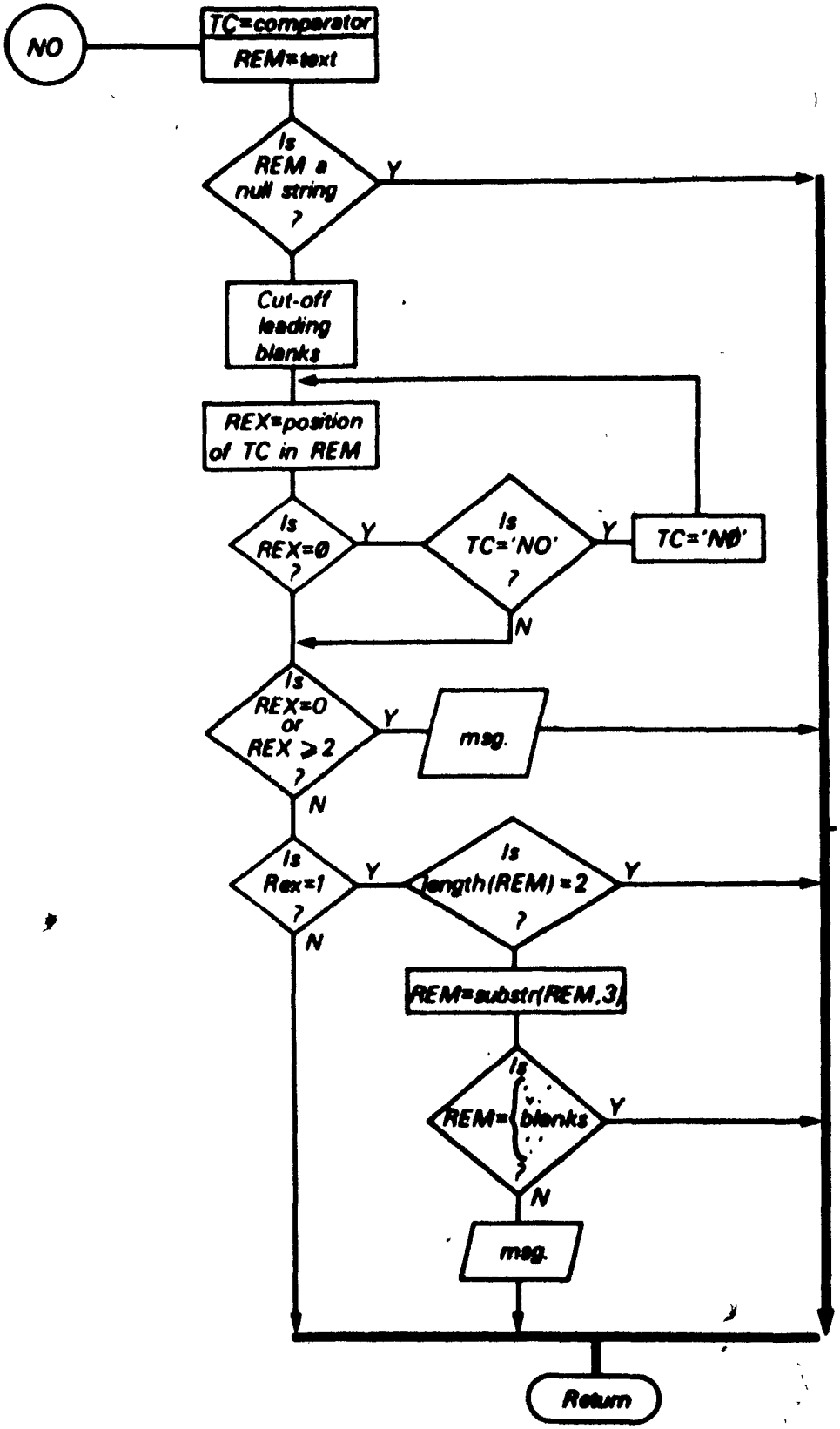


Fig. 6.10 - Flow diagram of module NO

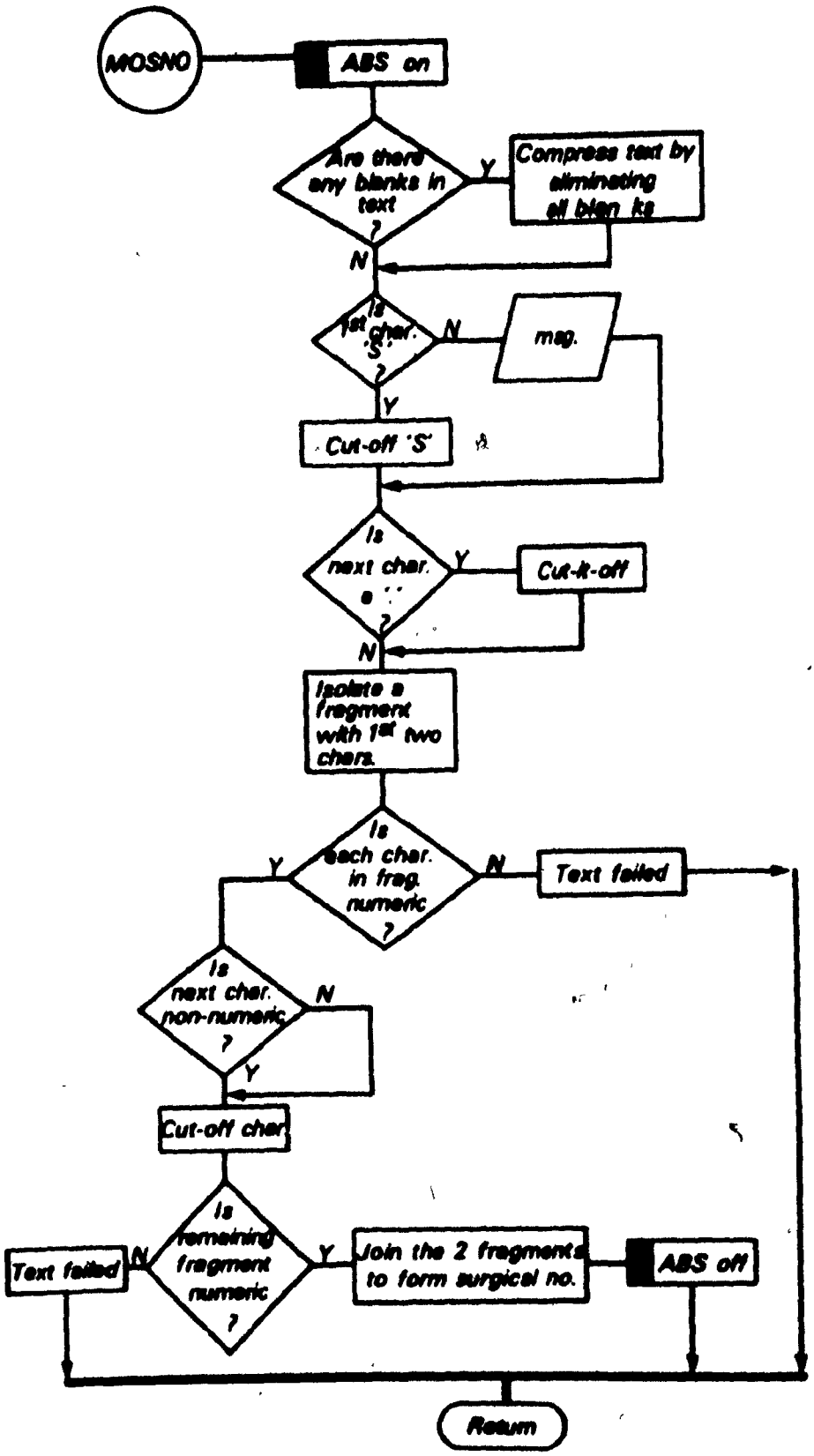


Fig. 6.11 - System chart of module MOSNO

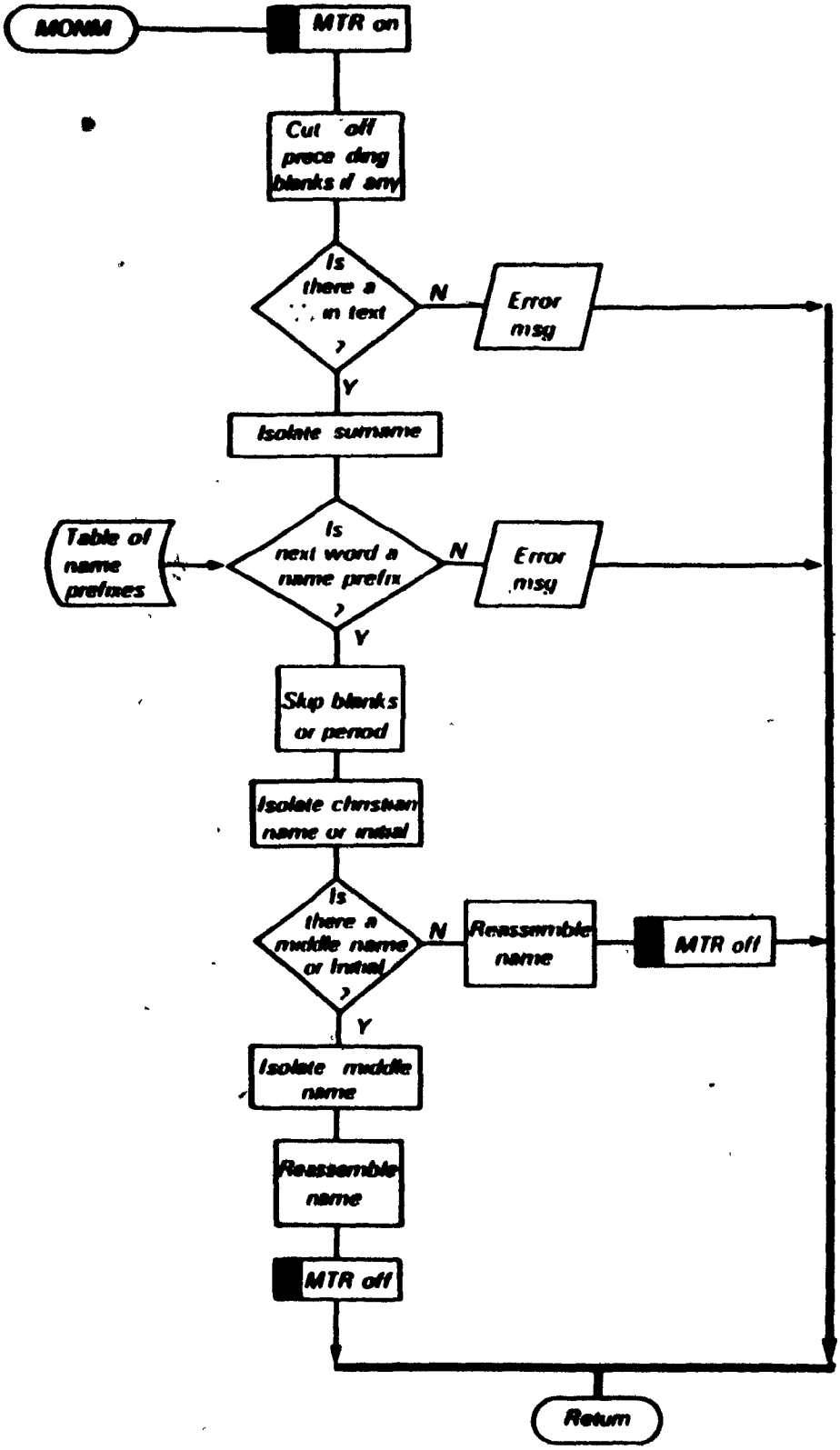


Fig. 6.12 - System chart of module MONM

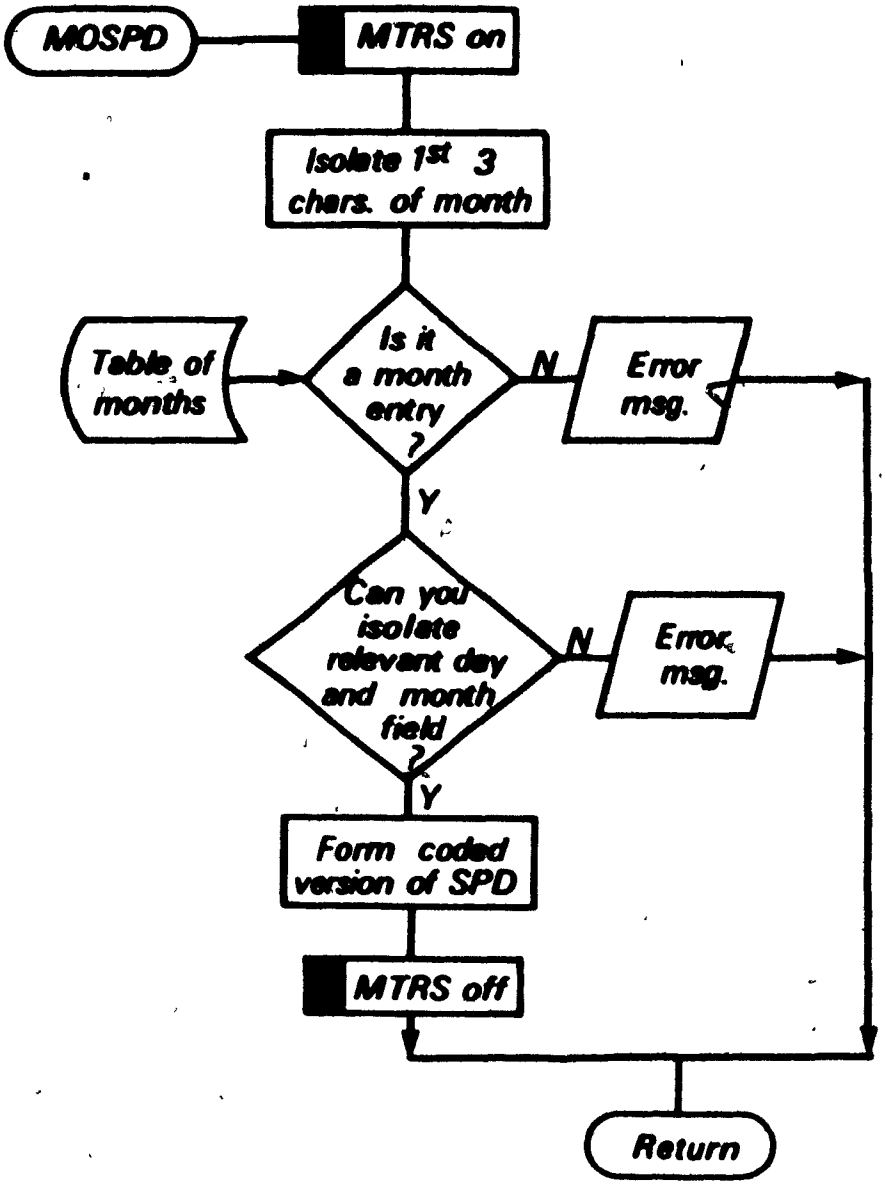


Fig. 6.13 - System chart of module MOSPD

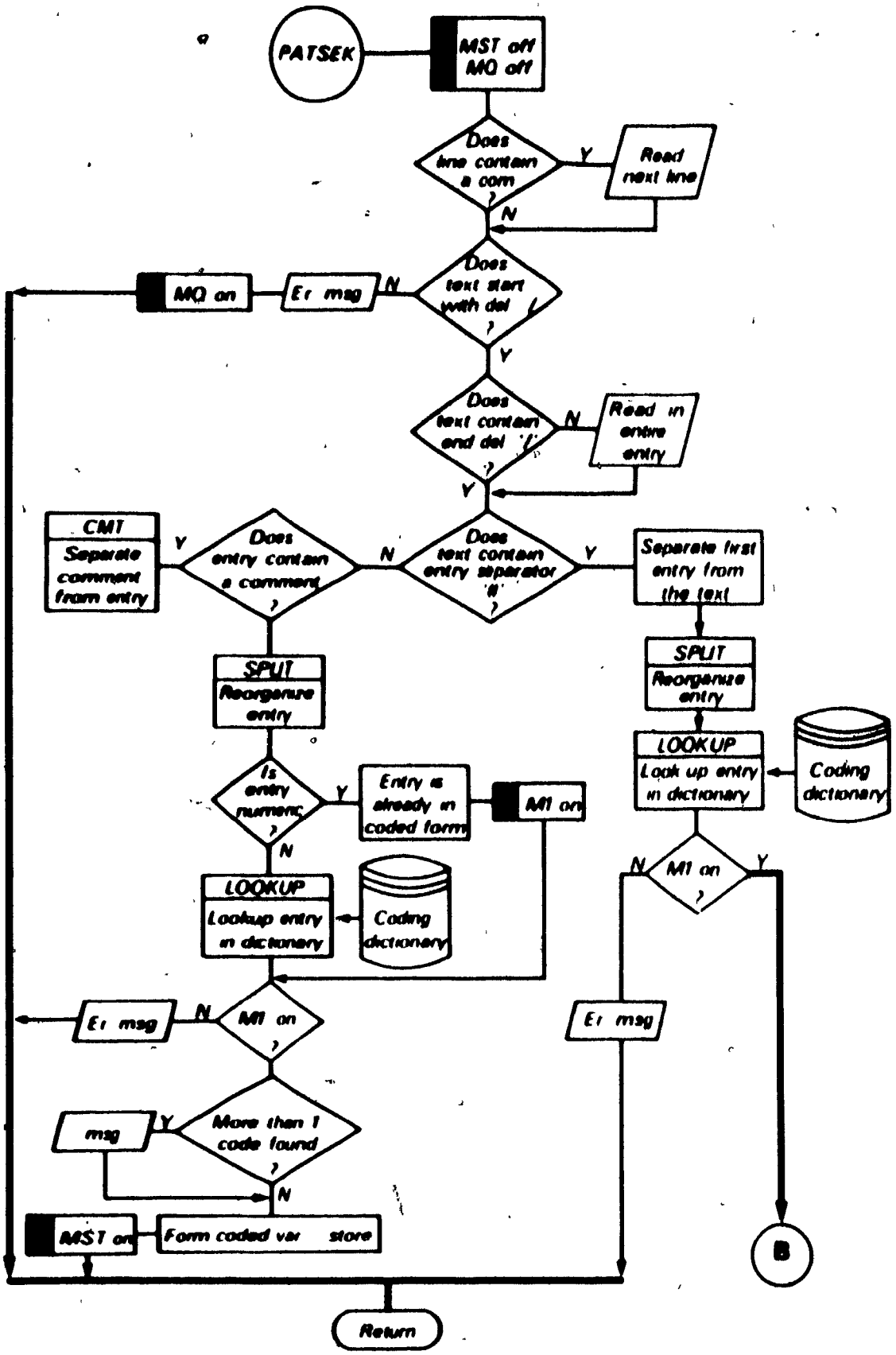


Fig. 6.14a - System chart of module PATSEK

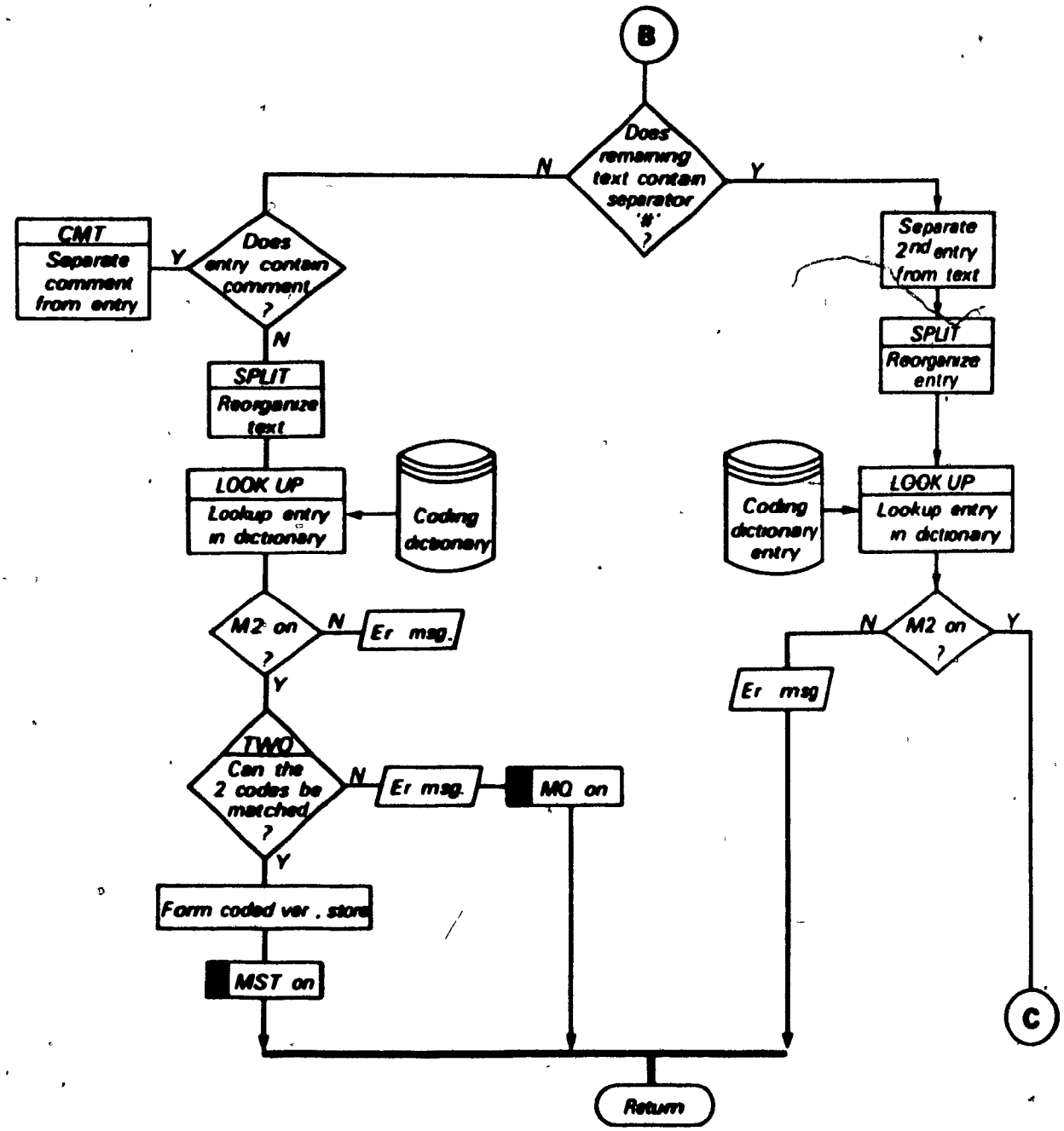


Fig. 6.14b - Module PATSEK

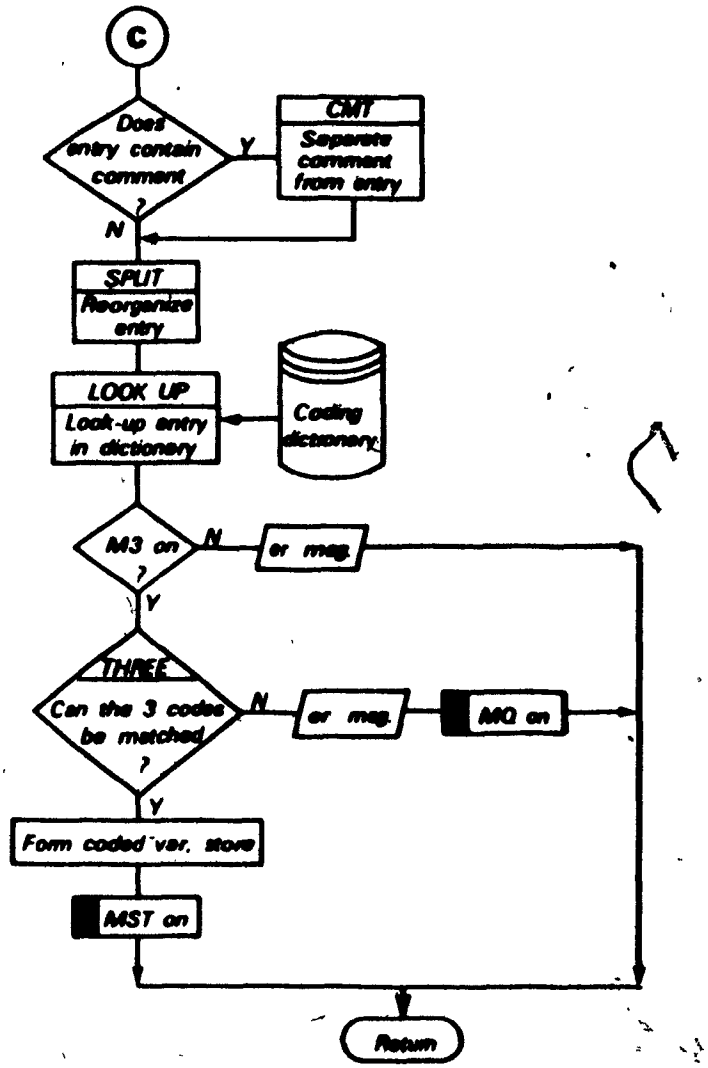


Fig. 6.14c - Module PATSEK

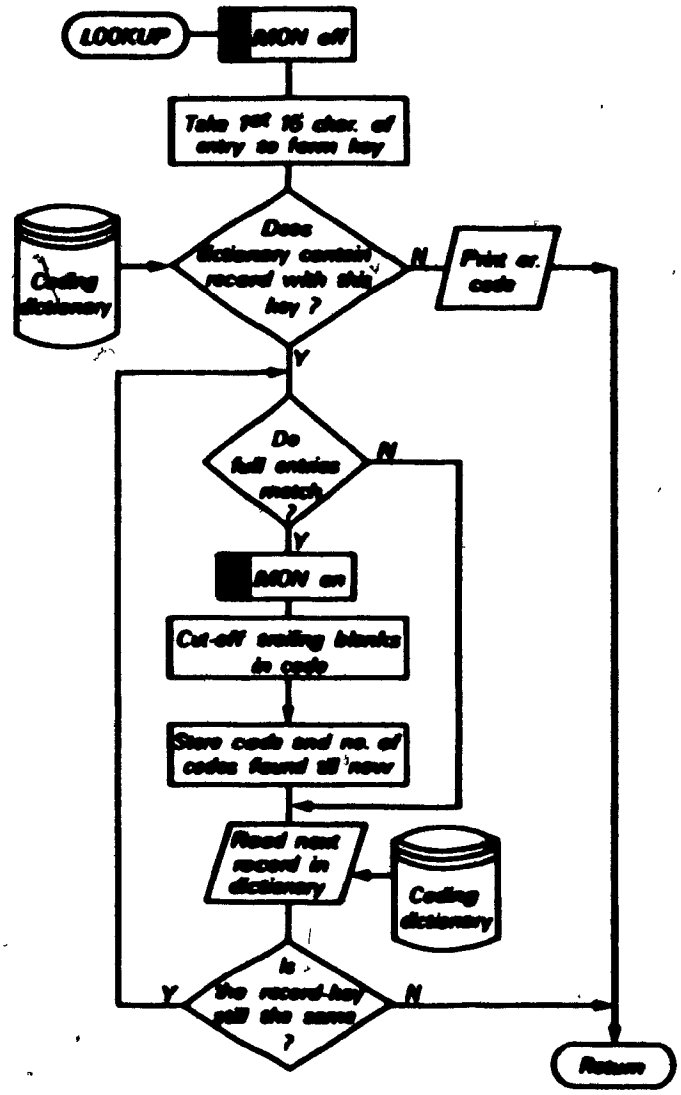


Fig. 6.14d - System chart of subroutine LOOKUP, internal to PATSEK

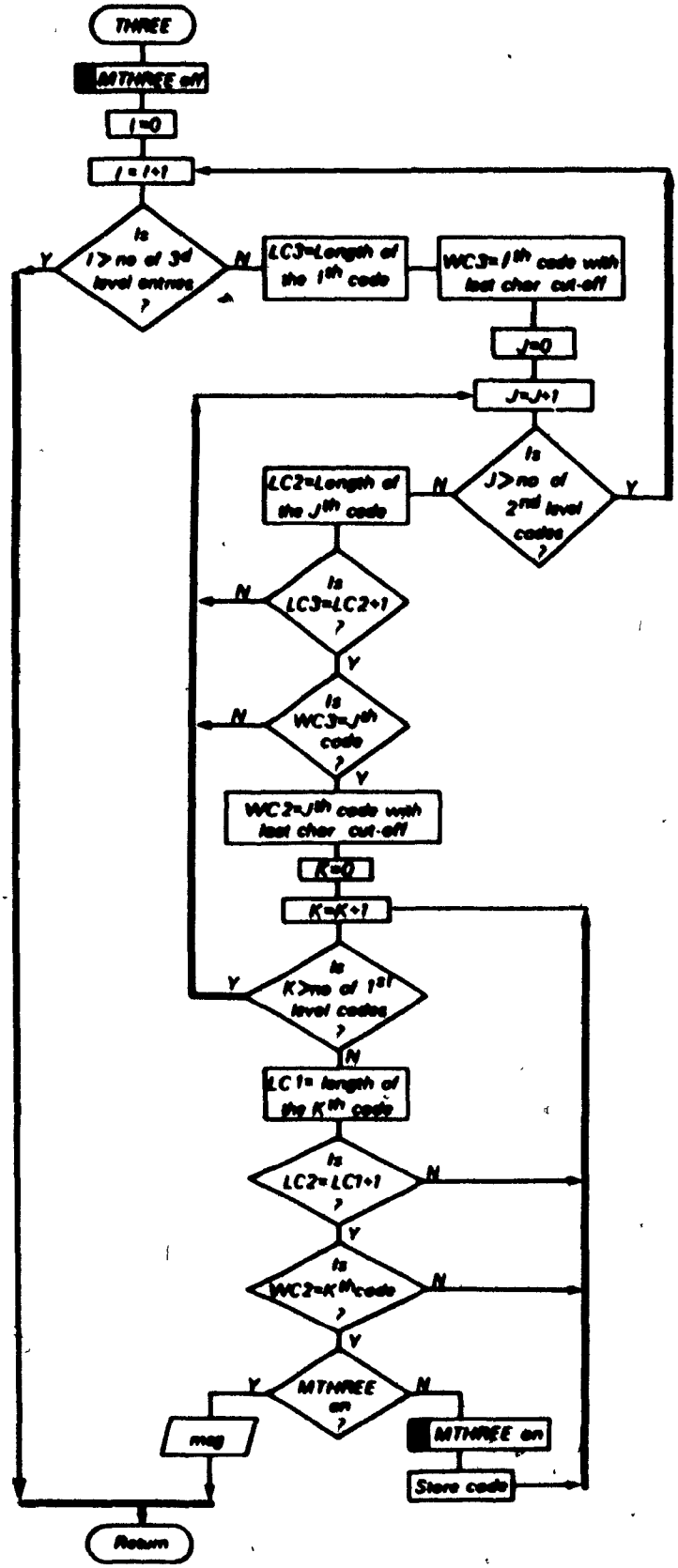


Fig. 6.14e - Flow diagram subroutine THREE, internal to PATSEK

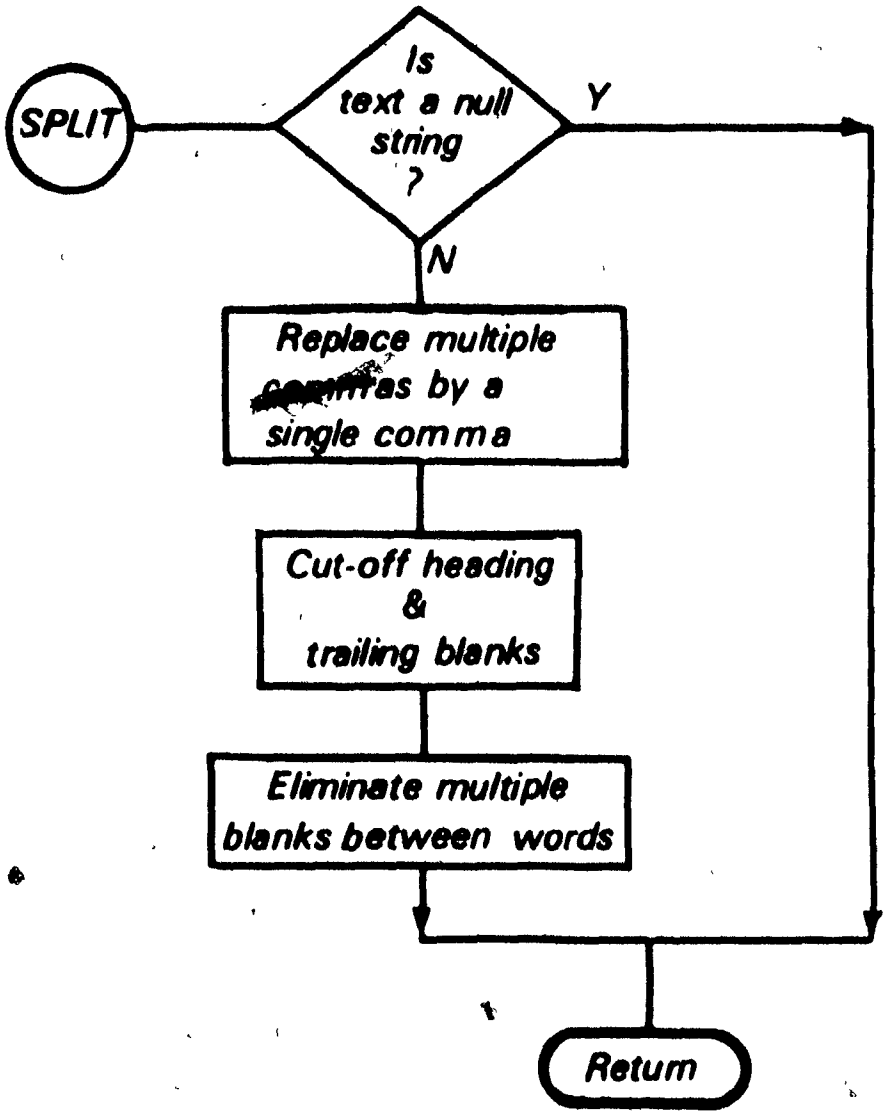


Fig. 6.15a - System chart of module SPLIT

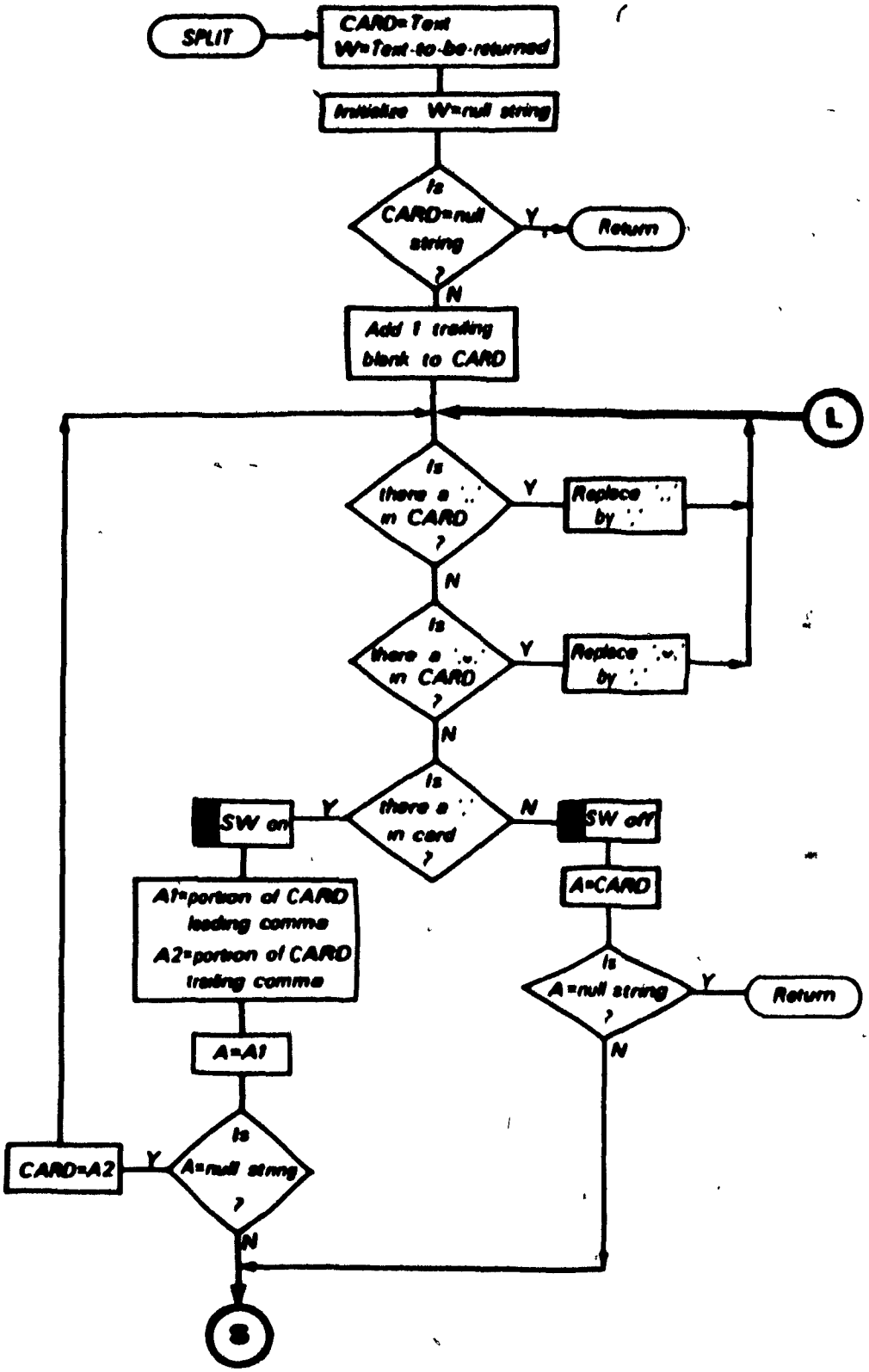


Fig. 6.15b - Flow diagram of module SPLIT

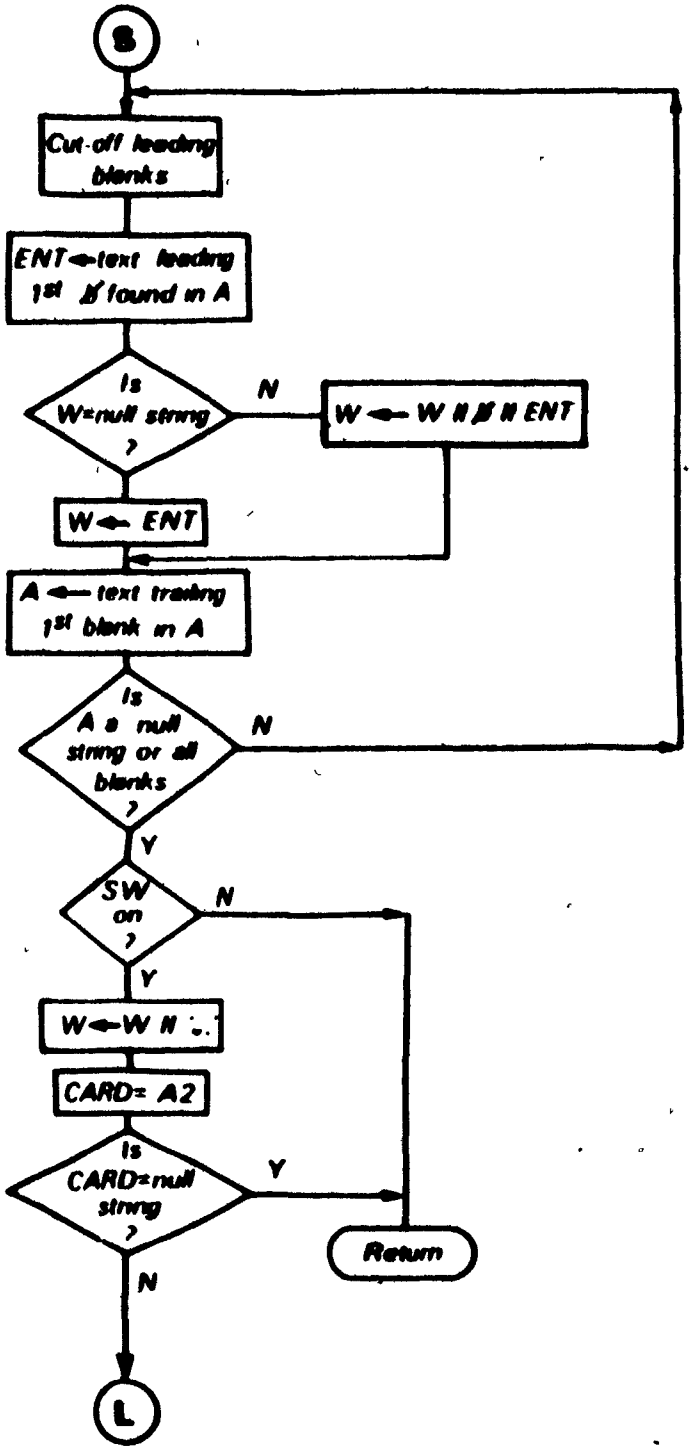
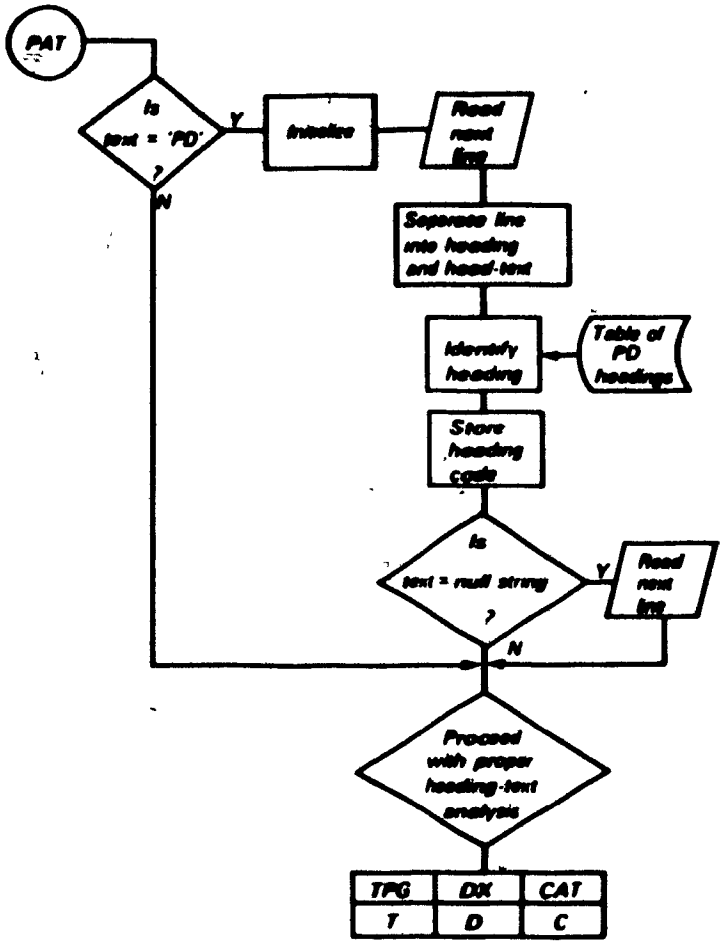


Fig. 6.15c - Module SPLIT



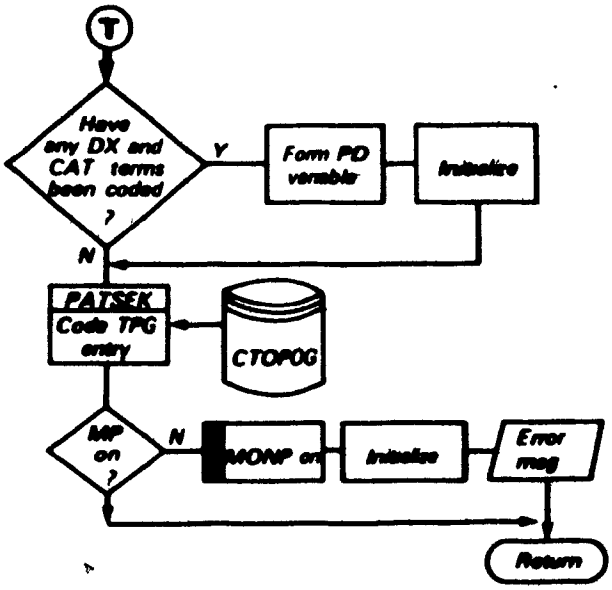


Fig. 6.16 a - System chart of module PAT

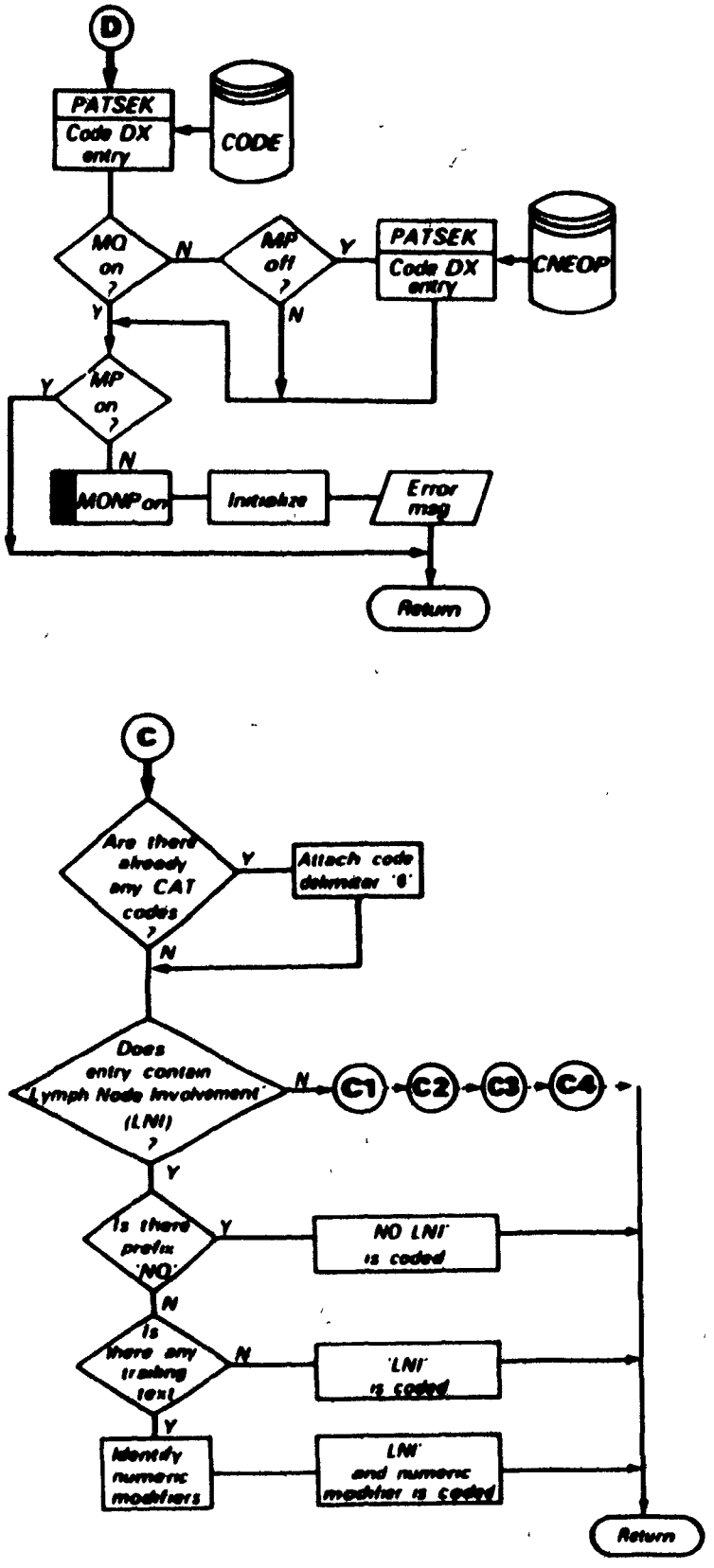
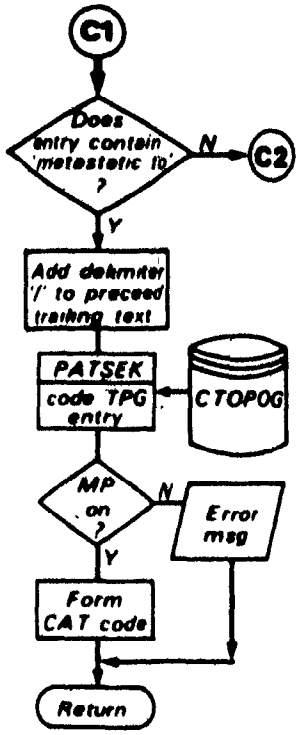


Fig. 6.16b - Module PAT



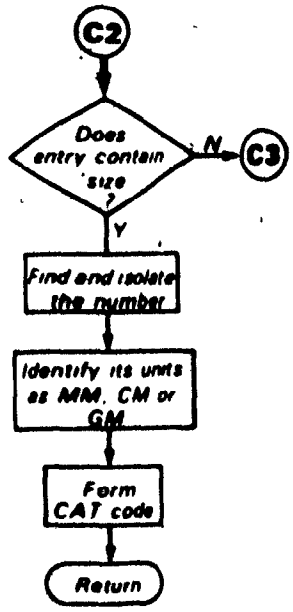


Fig. 6.16c - Module PAT

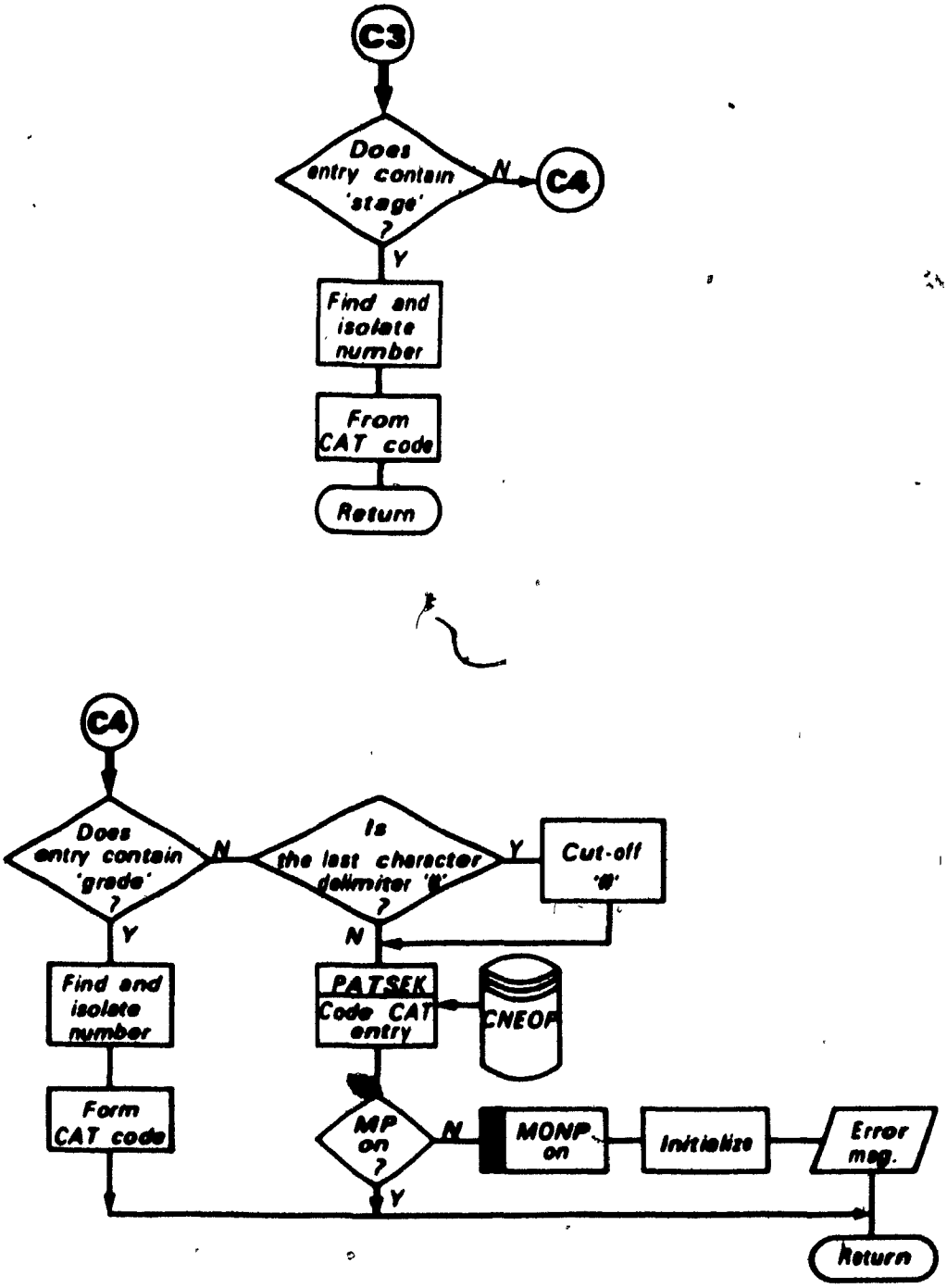


Fig. 6.16d - Module PAT

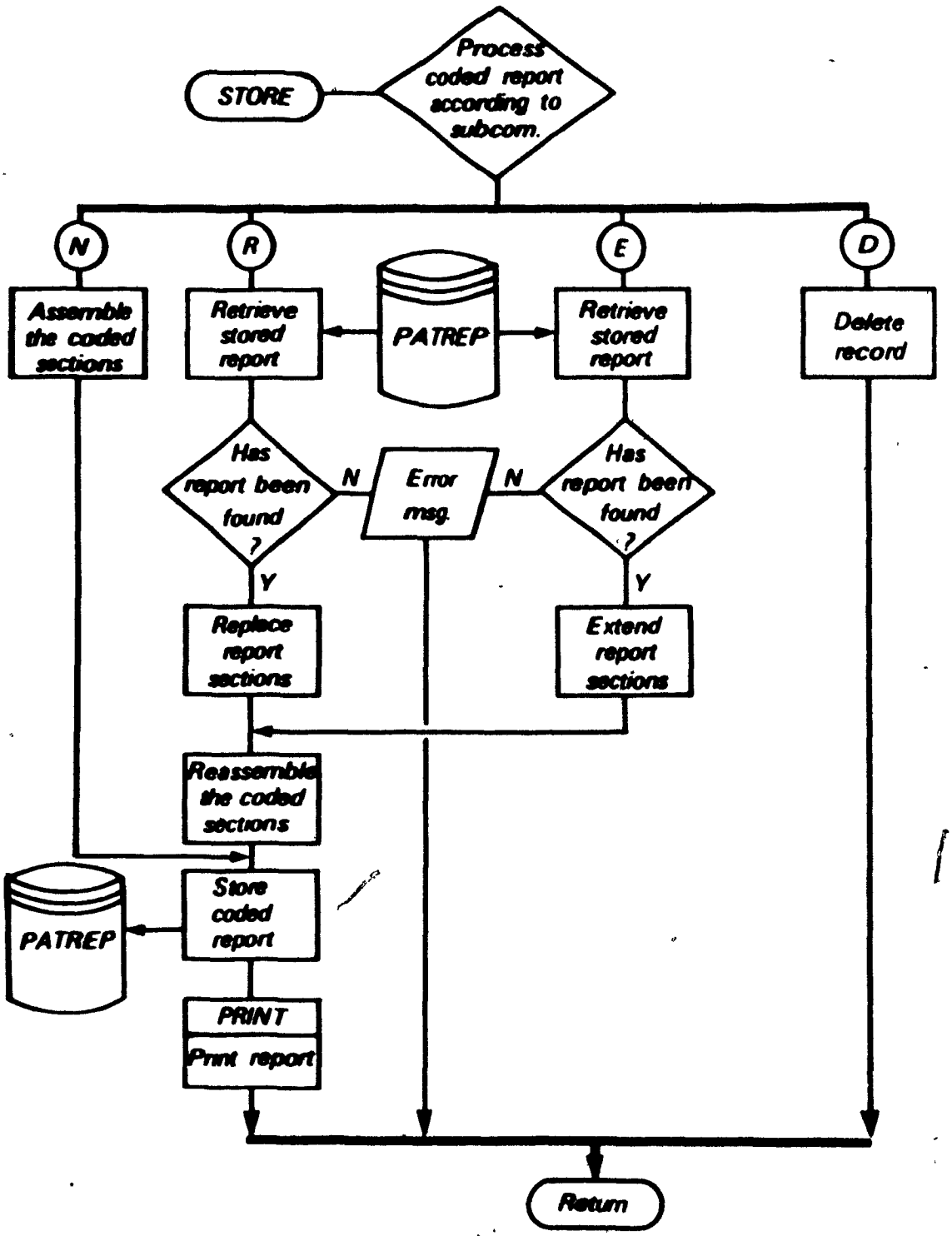


Fig. 6.17 - System chart of module STORE

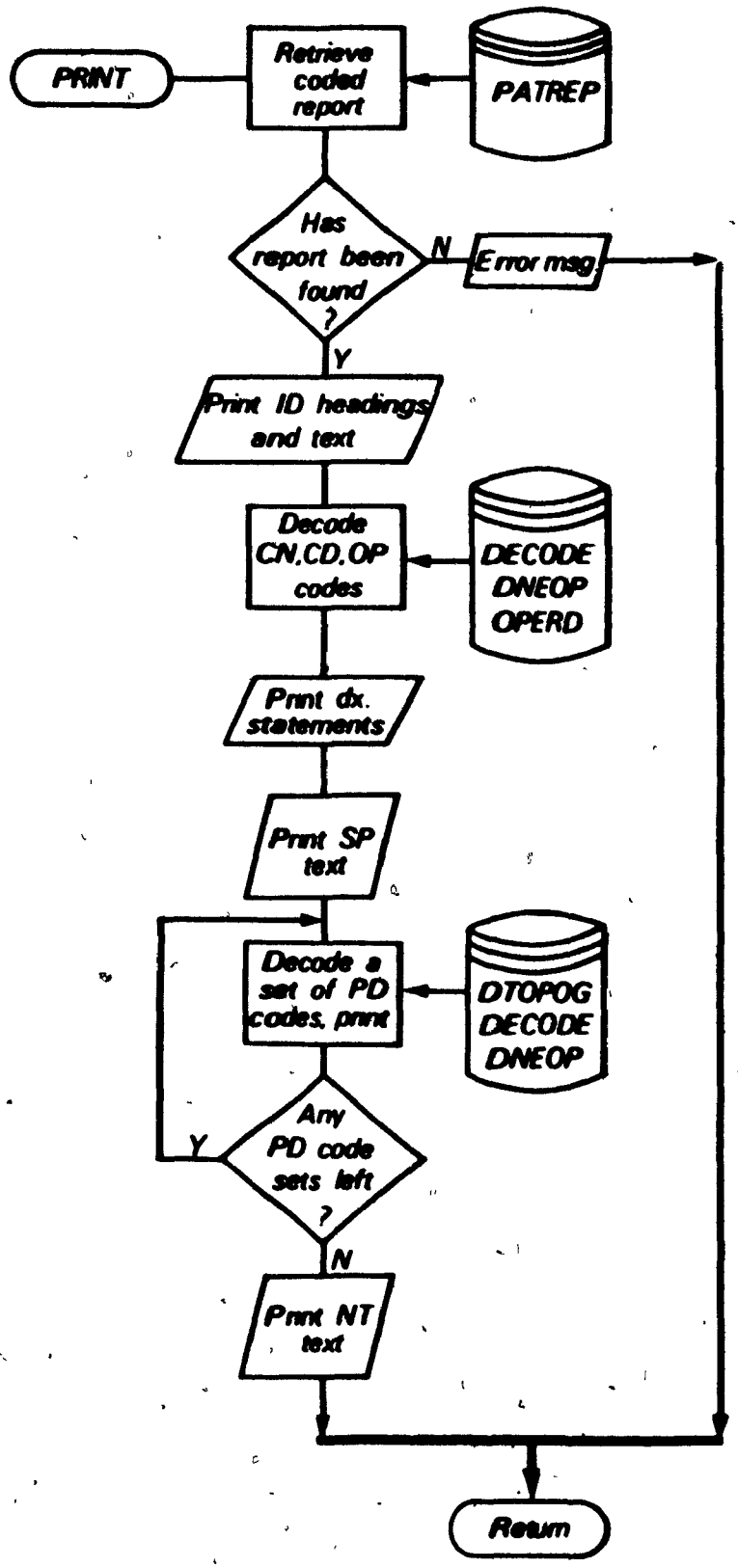


Fig. 6.18 - System chart of module PRINT

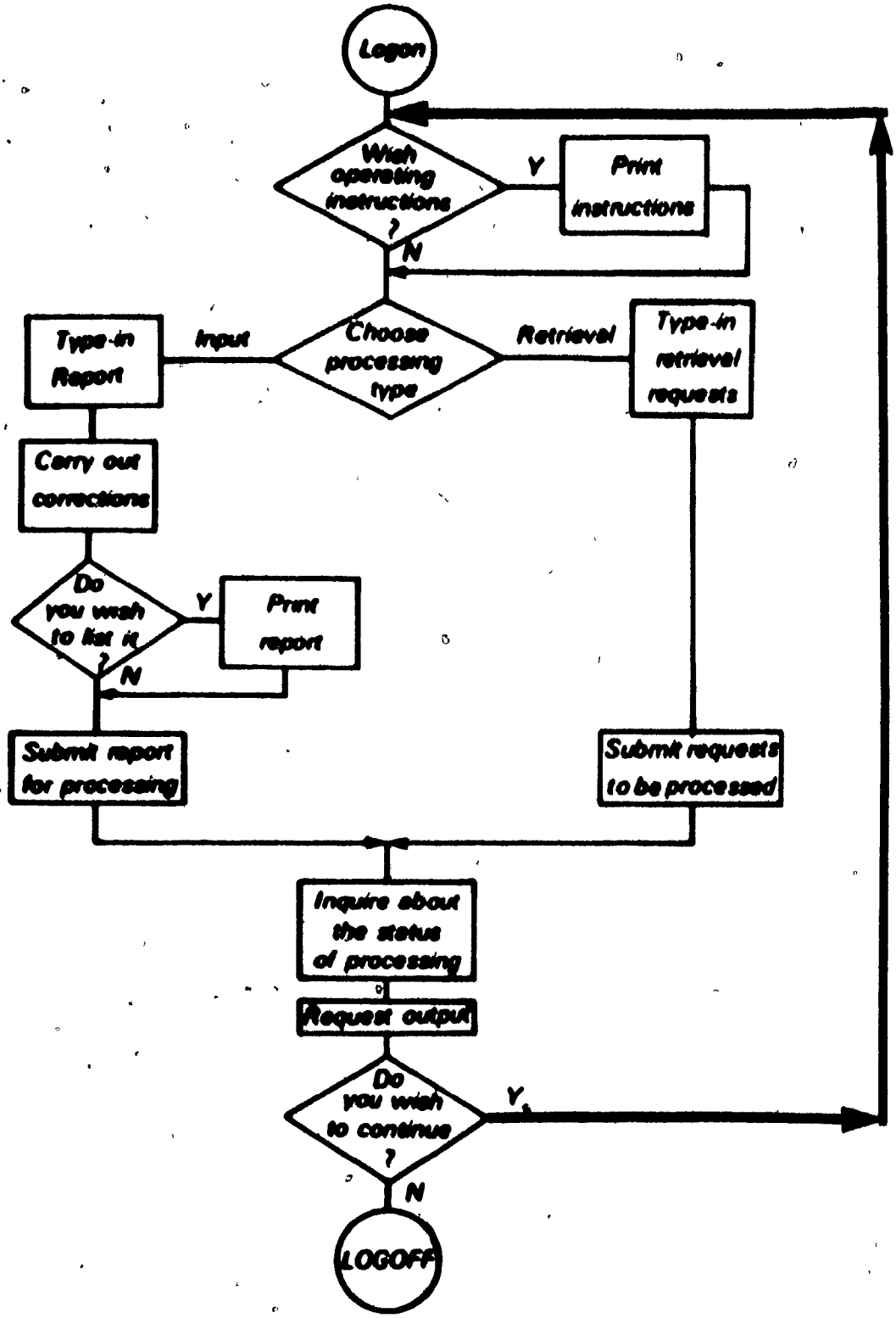


Fig. 6.19 - System chart of the interactive processing activities in CISP.

Chapter 7

EVALUATION OF THE PROTOTYPE CISP SYSTEM AND IMPROVEMENT RECOMMENDATIONS

7.1 Introduction

This chapter illustrates CISP's performance in coding pathology reports and retrieving them in decoded form. An example of an interactive work session with CISP is also shown. These illustrations are followed by an evaluation of the design and the performance of CISP.

Based on the trial-runs of this prototype system, a number of recommendations can already be made. However, the final improvements of this system can only be made after the in-hospital operational trials are completed.

7.2 Examples of Pathology Report Processing and Retrieval with CISP

Figures 7.1 (a,b,c,d) and 7.2 (a,b.c.d) each contain an example of a pathology report. The "a" figures contain the original document. The reports typed-in at the terminal are shown in "b"

figures. The coded version of the report as it is stored on disk is reproduced in the "c" figures. The "d" figures show the reports as they are printed out when the decoded version of the report is requested. This decoded version may be compared to the originals in the "a" figures.

Figure 7.3 shows an interactive work session in which a report is entered for coding and storage, is updated and then printed out in decoded form. This figure mainly illustrates the warning messages and error messages issued by CISP to the operator.

7.3 Evaluation of the Present Prototype

The present prototype CISP is operational and can be operated from a terminal situated in the hospital. The results obtained with CISP prove that it is possible to implement a system which stores a large ICDA-type dictionary for the coding of pathology reports. The present system contains all dictionary sections related to the digestive system. Thus reports of any patients suffering from ailments of the digestive system can be coded and stored. The operational trials have indicated that the text of the decoded pathology report is acceptable for clinical use.

The design and implementation of CISP was carried out over a period of 3 years. About 65% of this time was spent on designing CISP and experimenting with various approaches, while the balance was spent on creating the dictionaries and programming the modules.

THE MONTREAL GENERAL HOSPITAL

DEPARTMENT OF PATHOLOGY

Name: Smith, Mr. John Room No.: 937 Unit No.: 345296

Sex: M Age: 78 Service Dr.: Sored M.F. Serial No.: S.73-11234

Date Received: Nov. 13/73

CN: Upper Gastro-Intestinal bleeding

CD: Cancer of Stomach

Specimen: Stomach

The stomach measures 22 cm and 17 cm on the greater and lesser curvature respectively. The external surface is smooth and glistening with some induration along the lesser curvature. At the mucosal surface is a fairly large, irregular, superficial ulceration with an irregular and granular mucosa surrounding the ulcer. The ulcer measures 10 cm in its greatest dimension and is located mainly on the lesser curvature. It extends near the margins of resection. On section, the lesion appears to invade the entire thickness of the wall of the lesser curvature and at this region the wall measures 0.9 cm. The lesser omentum shows 5 lymph nodes two of which are partly replaced by tumor.
Representative sections are submitted.
Representative sections from the ulcer and nearby gastric mucosa submitted labelled no. 1 and no. 2,3,4 and 5. Representative section from the proximal and distal margins of resection are submitted as no. 6 and no. 7.

DIAGNOSIS:

Gastrectomy:

1. Poorly differentiated adenocarcinoma of lesser curvature with extensive perineural, perivascular invasion and massive involvement of the entire thickness of the gastric wall and with metastases to two of five lymph nodes.
2. Interstitial metaplasia in gastric mucosa.

Note: The tumor extends deep but not into the margins of resection

Prev. Op.: Gastroscopy biopsy sent Feb. 8, 1973.

Date: Nov. 14/73

Fig. 7.1a - The pathology report with Surgical No. S.73-11234

```
ID,NEW
NAME; SMITH,MR.JOHN K.
ROOM NO.: 937
UNIT NO   : 345296
SEX; M
AGE : 78
SERVICE DR. : SORED M. F.
SURGICAL NO.: S.73-11234
SPECIMEN DATE; NOV. 13/73

CN
/GASTROINTESTINAL TRACT HEMORRHAGE NOS/
CD
/152 (STOMACH)/
OP
/COMPLETE GASTRECTOMY/
SP
/SPECIMEN : STOMACH
      THE STOMACH MEASURES 22 CM AND 17 CM ON THE GREATER AND LESSER
CURVATURE RESPECTIVELY. THE EXTERNAL SURFACE IS SMOOTH AND
GLISTENING WITH SOME INDURATION ALONG THE LESSER CURVATURE.
AT THE MUCOSAL SURFACE IS A FAIRLY LARGE, IRREGULAR, SUPERFICIAL
ULCERATION WITH AN IRREGULAR AND GRANULAR MUCOSA SURROUNDING
THE ULCER. THE ULCER MEASURES 10 CM IN ITS GREATEST DIMENSION
AND IS LOCATED MAINLY ON THE LESSER CURVATURE. IT EXTENDS NEAR
THE MARGINS OF RESECTION. ON SECTION , THE LESION APPEARS TO
INVADE THE ENTIRE THICKNESS OF THE WALL OF THE LESSER CURVATURE
AND AT THIS REGION THE WALL MEASURES 0.9 CM. THE LESSER OMENTUM
SHOWS 5 LYMPH NODES TWO OF WHICH ARE PARTLY REPLACED BY TUMOR.
      REPRESENTATIVE SECTIONS ARE SUBMITTED.
      REPRESENTATIVE SECTIONS FROM ULCER AND NEARBY GASTRIC MUCOSA
SUBMITTED LABELLED NO.1 AND NO.2,3,4, AND 5 . REPRESENTATIVE SECTION
FROM THE PROXIMAL AND DISTAL MARGINS OF RESECTION ARE
SUBMITTED AS NO.6 AND NO.7./
NT
/PREVIOUS OP. GASTROSCOPY - FEB 8,1973/
PD
TOPOGRAPHY : /LESSER CURVATURE OF STOMACH/
DIAGNOSIS;
   /ADENOCARCINOMA/
CATEGORY; /POORLY DIFFERENTIATED/
/INVOLVEMENT OF NERVE/
/METASTATIC TO LYMPH NODE,NOS/
/EXTENSION TO SEROSA OR CAPSULE/
/LYMPH NODE INVOLVEMENT 2 OUT OF 5/
/RESECTION MARGINS FREE OF TUMOR/
```

Fig. 7.1b - The pre-edited pathology report

```
SNO *7311234*
NM  *2SMITH JOHN K              *
RNO *937   *
UNO * 349296*
SEX *M*
AGE *  78*
YR  *     *
SRV *SORED M F              *
SPD *111373*

*CN*7857*CD*152(STOMACH)*OP*444*PD*T<6321>D<1522>C<IC12#IC34#IC81
(8888)#IC25#IC88(2 5)#IC35>*NT*PREVIOUS OP. GASTROSCOPY - FEB 8 ,
1973*SP*SPECIMEN : STOMACH#       THE STOMACH MEASURES 22 CM AND
17 CM ON THE GREATER AND LESSER#CURVATURE RESPECTIVELY. THE EXTER
NAL SURFACE IS SMOOTH AND#GLISTENING WITH SOME INDURATION ALONG T
HE LESSER CURVATURE.#AT THE MUCOSAL SURFACE IS A FAIRLY LARGE, IR
REGULAR, SUPERFICIAL#ULCERATION WITH AN IRREGULAR AND GRANULAR MU
COSA SURROUNDING#THE ULCER. THE ULCER MEASURES 10 CM IN ITS GREAT
EST DIMENSION#AND IS LOCATED MAINLY ON THE LESSER CURVATURE. IT E
XTENDS NEAR#THE MARGINS OF RESECTION. ON SECTION , THE LESION AP*
EARS TO#INVADE THE ENTIRE THICKNESS OF THE WALL OF THE LESSER CUR
VATURE#AND AT THIS REGION THE WALL MEASURES 0.9 CM. THE LESSER OM
ENTUM#SHOWS 9 LYMPH NODES TWO OF WHICH ARE PARTLY REPLACED BY TUM
OR.#       REPRESENTATIVE SECTIONS ARE SUBMITTED.#       REPRESENTA
TIVE SECTIONS FROM ULCER AND NEARBY GASTRIC MUCOSA#SUBMITTED LABE
LLED NO.1 AND NO.2,3,4, AND 5 . REPRESENTATIVE SECTION#FROM THE P
ROXIMAL AND DISTAL MARGINS OF RESECTION ARE#SUBMITTED AS NO.6 AND
NO.7.
```

Fig. 7.1c - Transcript of the coded report stored on disk. The SP section (shown below the broken line) is deleted from the report once the patient leaves the hospital.

```
          THE MONTREAL GENERAL HOSPITAL
            DEPARTMENT OF PATHOLOGY

SURGICAL NO.: S.73-11234   UNIT NO.: 345296    ROOM NO.: 937
NAME: SMITH,MR. JOHN K                SEX: MALE      AGE:  78
SERVICE DR.: SORED M F
SPECIMEN DATE: NOVEMBER 13/73
CN:  1. SYMPTOMS REFERABLE TO ABDOMEN AND LOWER GASTROINTESTINAL TRACT -
       - GASTROINTESTINAL TRACT HEMORRHAGE NOS
CD:  1. CARCINOMA(STOMACH)
OP:  1. OPERATIONS ON STOMACH -- COMPLETE GASTRECTOMY
SP:  SPECIMEN : STOMACH
        THE STOMACH MEASURES 22 CM AND 17 CM ON THE GREATER AND LESSER
    CURVATURE RESPECTIVELY. THE EXTERNAL SURFACE IS SMOOTH AND
    GLISTENING WITH SOME INDURATION ALONG THE LESSER CURVATURE.
    AT THE MUCOSAL SURFACE IS A FAIRLY LARGE, IRREGULAR, SUPERFICIAL
    ULCERATION WITH AN IRREGULAR AND GRANULAR MUCOSA SURROUNDING
    THE ULCER. THE ULCER MEASURES 18 CM IN ITS GREATEST DIMENSION
    AND IS LOCATED MAINLY ON THE LESSER CURVATURE. IT EXTENDS NEAR
    THE MARGINS OF RESECTION. ON SECTION , THE LESION APPEARS TO
    INVADE THE ENTIRE THICKNESS OF THE WALL OF THE LESSER CURVATURE
    AND AT THIS REGION THE WALL MEASURES 0.9 CM. THE LESSER OMENTUM
    SHOWS 5 LYMPH NODES TWO OF WHICH ARE PARTLY REPLACED BY TUMOR.
        REPRESENTATIVE SECTIONS ARE SUBMITTED.
        REPRESENTATIVE SECTIONS FROM ULCER AND NEARBY GASTRIC MUCOSA
    SUBMITTED LABELLED NO.1 AND NO.2,3,4, AND 5 . REPRESENTATIVE SECTION
    FROM THE PROXIMAL AND DISTAL MARGINS OF RESECTION ARE
    SUBMITTED AS NO.6 AND NO.7.
PD:  1. LESSER CURVATURE OF STOMACH
       CARCINOMA -- ADENOCARCINOMA
         POORLY DIFFERENTIATED
         INVOLVEMENT OF NERVE
         METASTATIC TO LYMPH NODE , NOS
         EXTENSION TO SEROSA OR CAPSULE
         LYMPH NODE INVOLVEMENT 2  OUT OF 5
         RESECTION MARGINS FREE OF TUMOR
MT: PREVIOUS OP. GASTROSCOPY - FEB 8 , 1973
```

Fig. 7.1d - The decoded pathology report

THE MONTREAL GENERAL HOSPITAL.

DEPARTMENT OF PATHOLOGY

Name: Johnsav , Mrs. K. D. Room No.: 966 Unit No.: 345296

Sex: F Age: 59 Service Dr.: Doren F.K. Serial No.: S.73-5639

Date Received: May 23/73

CN: Vomiting - Duration 15 days
Constipation

CD: Bowel obstruction

OP: Exploratory Laparotomy

Specimen: Bx. gasto-colic ligament

FROZEN SECTION:

Malignant - sarcoma , probable fibrosarcoma

Special Remarks: No definitive OR evidence of Primary; possible pancreatic

GROSS:

The specimen is labelled "biopsy of gastro-colic ligament" and consists of 2 pieces of firm tissue , pinkish white in colour , measuring 1.5 x 0.7 c 1.5 x 0.8 cm . Cut section of the specimen shows it to be variegated.
Submitted in toto.

MICROSCOPY:

The tissue consists of several circumscribed fatty nodules which are subdivided by well developed bands of fibroblastic tissue. Individual fat cells are also separated by proliferative fibroblasts . There are occasional mitoses and atypical cells. Focal fat degeneration with chronic inflammatory reaction is also noted. This is an unusual pattern but Ibelieve it to be a malignant neoplasm although a reactive process can not be ruled out with complete certainty.

DIAGNOSIS: Fibrosarcoma

Note: See Autopsy A73-321 . The gross appearance suggests mesothelioma involving peritoneal cavity and right pleural cavity. Therefore this is a variant fibrosarcomatous mesothelioma.

Date: May 24/73

Fig. 7.2a - The pathology report.

```
ID.N
NM: JOHNSAY, MRS. K. D
AGE.:966
UNO   :  3278789
SEX : F
AGE : 59
SERVICE DR.: DOREN F. K.
SNO  : 73-5369
SPD: MAY 23/73
CN
/NAUSEA AND VOMITING(15 DAYS)/
/CONSTIPATION (15 DAYS)/
CD
/OBSTRUCTION DUE TO UNSPECIFIED CAUSE (BOWEL)/
OP
/EXPLORATORY LAPAROTOMY OR CELIOTOMY/
PD
IPS: /GASTROCOLIC LIGAMENT/
DX:
     /FIBROSARCOMA/
     /MESOTHELIOMA/
SP
/SPECIMEN : BX. GASTRO-COLIC LIGAMENT
FROZEN SECTION:
      MALIGNANT - SARCOMA, PROBABLE FIBROSARCOMA
SPECIAL REMARKS: NO DEFINITIVE OR EVIDENCE OF PRIMARY:
  POSSIBLY PANCREATIC.
GROSS :
      THE SPECIMEN IS LABELLED "BIOPSY OF GASTRO-COLIC LIGAMENT "
AND CONSISTS OF 2 PIECES OF FIRM TISSUE , PINKISH WHITE IN COLOUR ,
MEASURING 1.5 X 0.7 C 1.5 X 0.8 CM. SUBMITTED IN TOTO.
MICROSCOPY :
      THE TISSUE CONSISTS OF SEVERAL CIRCUMSCRIBED FATTY
NODULES WHICH ARE SUBDIVIDED BY WELL DEVELOPED BANDS OF FIBROBLASTIC
TISSUE. INDIVIDUAL FAT CELLS ARE ALSO SEPARATED BY PROLIFERATIVE
FIBROBLASTS. THERE ARE OCCASIONAL MITOSES AND ATYPICAL CELLS.
FOCAL FAT DEGENERATION WITH CHRONIC INFLAMMATORY REACTION IS ALSO
NOTED. THIS IS AN UNUSUAL PATTERN BUT I BELIEVE IT TO BE A MALIGNANT
NEOPLASM ALTHOUGH A REACTIVE PROCESS CAN NOT BE RULED OUT WITH
COMPLETE CERTAINTY ./
AT
/SEE AUTOPSY 73-321/
```

Fig. 7.2b - The pre-edited report

```
SNO *7385369*
NM  *1 JOHNSAY K D              *
RNO *966   *
UNO *3278789*
SEX *F*
AGE *  59*
YB  *      *
SRV *DOREN F K             *
SPD *052573*

*CN*7841(15 DAYS)#5640(15 DAYS)*CD*5609(BOWEL)*OP*551*PD*T<6381>D
<2258#288>*NT*SEE AUTOPSY A73-321*SP*SPECIMEN : BX. GASTRO-COLIC
LIGAMENT#FROZEN SECTION:#       MALIGNANT - SARCOMA, PROBABLE FIBR
OSARCOMA#SPECIAL REMARKS: NO DEFINITIVE OR EVIDENCE OF PRIMARY:#
 POSSIBLY PANCREATIC.#GROSS :#       THE SPECIMEN IS LABELLED "BIO
PSY OF GASTRO-COLIC LIGAMENT "#AND CONSISTS OF 2 PIECES OF FIRM T
ISSUE , PINKISH WHITE IN COLOUR ,#MEASURING 1.5 X 0.7 C 1.5 X 0.8
 CM. SUBMITTED IN TOTO.#MICROSCOPY :#       THE TISSUE CONSISTS OF
 SEVERAL CIRCUMSCRIBED FATTY#NODULES WHICH ARE SUBDIVIDED BY WELL
 DEVELOPED BANDS OF FIBROBLASTIC#TISSUE. INDIVIDUAL FAT CELLS ARE
 ALSO SEPARATED BY PROLIFERATIVE#FIBROBLASTS. THERE ARE OCCASIONA
L MITOSES AND ATYPICAL CELLS.#FOCAL FAT DEGENERATION WITH CHRONIC
 INFLAMMATORY REACTION IS ALSO#NOTED. THIS IS AN UNUSUAL PATTERN
BUT I BELIEVE IT TO BE A MALIGNANT#NEOPLASM ALTHOUGH A REACTIVE P
ROCESS CAN NOT BE RULED OUT WITH#COMPLETE CERTAINTY .
```

Fig. 7.2c - The coded report

THE MONTREAL GENERAL HOSPITAL
DEPARTMENT OF PATHOLOGY

SURGICAL NO.: S.73- 5369 UNIT NO.: 3278789 ROOM NO.: 966
NAME: JOHNSAY, MRS. K D SEX: FEMALE AGE: 59
SERVICE DR.: DOREN F K
SPECIMEN DATE: MAY 23/73
CN: 1. SYMPTOMS REFERABLE TO UPPER GASTROINTESTINAL TRACT -- NAUSEA AND VOMITING(15 DAYS)
2. FUNCTIONAL DISORDERS OF INTESTINE -- CONSTIPATION(15 DAYS)
CO: 1. INTESTINAL OBSTRUCTION WITHOUT MENTION OF HERNIA -- OBSTRUCTION DUE TO UNSPECIFIED CAUSE(BOWEL)
OP: 1. INCISION AND EXCISION OF ABDOMINAL WALL AND PERITONEUM -- EXPLORATORY LAPAROTOMY OR CELIOTOMY
SP: SPECIMEN : BX. GASTRO-COLIC LIGAMENT
FROZEN SECTION:
MALIGNANT - SARCOMA, PROBABLE FIBROSARCOMA
SPECIAL REMARKS: NO DEFINITIVE OR EVIDENCE OF PRIMARY:
POSSIBLY PANCREATIC.
GROSS :
THE SPECIMEN IS LABELLED "BIOPSY OF GASTRO-COLIC LIGAMENT " AND CONSISTS OF 2 PIECES OF FIRM TISSUE , PINKISH WHITE IN COLOUR , MEASURING 1.5 X 0.7 C 1.5 X 0.8 CM. SUBMITTED IN TOTO.
MICROSCOPY :
THE TISSUE CONSISTS OF SEVERAL CIRCUMSCRIBED FATTY NODULES WHICH ARE SUBDIVIDED BY WELL DEVELOPED BANDS OF FIBROBLASTIC TISSUE. INDIVIDUAL FAT CELLS ARE ALSO SEPARATED BY PROLIFERATIVE FIBROBLASTS. THERE ARE OCCASIONAL MITOSES AND ATYPICAL CELLS. FOCAL FAT DEGENERATION WITH CHRONIC INFLAMMATORY REACTION IS ALSO NOTED. THIS IS AN UNUSUAL PATTERN BUT I BELIEVE IT TO BE A MALIGNANT NEOPLASM ALTHOUGH A REACTIVE PROCESS CAN NOT BE RULED OUT WITH COMPLETE CERTAINTY .
PD: 1. GASTROCOLIC LIGAMENT
SARCOMA -- FIBROSARCOMA
MESOTHELIOMA
NT: SEE AUTOPSY A73-321

Fig. 7.2d - The decoded report

```
*Y ONE
      01   ID,N
      02   NAME;DAVIEL,J. K.
      03   SPD: MRCH 7/73
      04   UNIT :    793878
      05   CN
      06   /HEADACHE/
      07
      08   ID,N
      09   SNO:  73-9872
      10   NAME : JOSS,MR. M.
      11   AGE : 58
      12   UNO: 382311
      13   SPECIMEN DAY: NOVEMBER 2,73
      14   CN
      15   HEARTBURN/
      16   /DISPHAGIA/
      17   CD
      18   /ULCER/
      19

*S
*END
*

 X CODE1

REPORT(S) IN DATA SET "ONE" SUBMITTED FOR PROCESSING

*

 OUTPUT ONE
```

```
LINE:'NAME,DANIEL , J. K.';
!--ER2(UN)--  NO NAME PREFIX(MR,MRS...) APPEARS IN
   !--ER2(ID)--INCORRECT NAME "DANIEL , J. K."
NAME NOT ENTERED FOR THIS PATIENT
•
LINE:'SPD: MRCH 7/73';
!--ER1(SPD)--  MONTH COULD NOT BE IDENTIFIED IN SPECIMEN DATE
   "MRCH 7/73"
!--ER7(ID)--INCORRECT SPECIMEN DATE :MRCH 7/73
DATE NOT ENTERED
•
LINE:'UNIT : 793878';
•
!--ER2(PATREAD)--PATIENT IDENTIFICATION HEADINGS WERE ALL READ-IN YET S
RGICAL NO. WAS NOT ENCOUNTERED
!!!--GOING ON TO THE NEXT ID
•
LINE:'SNO: 73-98 72';
!--ER1(SNO)--  S MISSING IN SURGICAL NO :
          73-98 72
•
LINE:'NAME : JOSS , MR. M.';
•
LINE:'AGE : 58';
•
LINE:'UNO: 382311';
•
LINE:'SPECIMEN DAY: NOVEMBER 2 , 73';
!--ER3(HEAD)--   IN LINE "SPECIMEN DAY: NOVEMBER 2 , 73"
"SPECIMEN DATE" IS UNDERSTOOD
!--ER2(SPD)-- SPECIMEN DATE "NOVEMBER 2 , 73"
IS NOT OF THE FORM "MONTH DAY/YR."
!--ER7(ID)--INCORRECT SPECIMEN DATE :NOVEMBER 2 , 73
DATE NOT ENTERED
•
•
!--ER1(•CW•)--ENTRY FOR CW IN LINE :
HEARTBURN/
DOES NOT START WITH /. LINE SKIPPED.
--CW CODING FAILED
• /
•
                                                  51
!--LOOKUP OF CW ENTRY "DISPHAGIA" FAILED
--CW CODING FAILED
•
•
!--MORE THAN 1 DICTIONARY CODE WAS FOUND TO MATCH THE CD ENTRY :
ULCER
--AN ALTERNATIVE CODE FOUND IS:55810
•
•
PRINT: THE RECORD WITH SNO S.73-09872
 HAS BEEN SUCCESSFULLY STORED
••••••••••
TS.SNO='7309872';
TS.NM='2JOSS M
TS.UNO= 382311;
TS.AGE= 58;
•C)=52899
••••••••••
END OF DATA
•
```

```
> TWO

      01  ID,REPLACE
      02  SNO: S.73-9872
      03  CM
      04  /HEARTBURN/
      05  /DYSPHAGIA/
      06

>S
>END
>

> CODE2

oREPORT(S) IN DATA SET "TWO" SUBMITTED FOR PROCESSING

>

 OUTPUT TWO

o*
 *
 LINE:'SNO: S.73-9872';
 *
 *
 *
 *
 *
 *
 PRINT: THE RECORD WITH SNO S.73-09872
  HAD THE ABOVE ELEMENTS REPLACED SUCCESSFULLY
 **********
 IS.SNO:'7309872';
 *CM=7843078 44
 *CD=52889D
 **********
 END OF DATA
 >

 > PRINT
      01  SURGICAL NO.: S.73-9878
      02  SNO : S.73-9872
      03



oTHE PATHOLOGY REPORT RETRIEVAL REQUEST IS BEING PROCESSED

>

 OUTPUT REPORT



o1--ER7(P)--RECORD WITH SNO S.73-09878 COULD NOT BE FOUND!


            THE MONTREAL GENERAL HOSPITAL
              DEPARTMENT OF PATHOLOGY

 SURGICAL NO.: S.73- 9872   UNIT NO.:  382311   ROOM NO.:
 NAME: JOSS,MR. H                    SEX:          AGE:  58
 SERVICE DR.:
 SPECIMEN DATE:
 CM:  1. SYMPTOMS REFERABLE TO UPPER GASTROINTESTINAL TRACT -- HEARTBURN
      2. SYMPTOMS REFERABLE TOUPPER GASTROINTESTINAL TRACT -- DYSPHAGIA
 CD:  1. DISEASES OF ORAL SOFT TISSUES , EXCLUDING GINGIVA AND TONGUE --
         OTHER AND UNSPECIFIED -- ULCER
 OP:
 SP:
 PD:

 END OF DATA
```

Fig. 7.3 - Example of a work session with CISP.
> Text entered by operator
O Text printed by CISP

To gain an appreciation of the execution characteristics of CISP, the processing of a pathology report may be considered. The load module program which performs this function occupies about 230 Kilobytes of core storage. The elapsed time, during which this program resides in core under the multiprogramming arrangement of the computer system, varies from 30 sec. to 5 minutes. The cost of an average report processing is about $1.50 which can be equally divided between the CPU execution time (~ 3 sec.) and the Input/Output cost. One must add to this an overhead of about $1.00 for initiating the program execution. Therefore, by entering several reports in the same batch, the cost of processing is reduced.

The above costs were calculated on the basis of program execution in a 300K core partition under priority 2 (the normal priority used). By using overlaying in the report processing program, the core size requirement could be reduced to 200K yielding shorter queue-in time and less-expensive execution. Overlaying is a feature of the linkage editor which enables only those modules to reside in core which are actually needed at any time during the processing. Aside from PATREAD and SPLIT, one would only find one of these module sets in core:

a) ID, IDMED, NO, MOSNO, MOSPD, MONM

b) PATSEK

c) PAT; PATSEK

d) STORE, PRINT, RESET

With overlaying execution is slightly slower due to the time needed to read in the various modules. Only experiments using the overlay feature can indicate whether the advantages gained are not cancelled out by the drawbacks.

A realistic cost estimate should add the weekly data set storage cost ($1.25/cylinder), the CRJE connect time and the rental of the terminal to the above expenses.

Since CISP is only a prototype system, only a limited number of retrieval capabilities were programmed. It is feasible to retrieve reports by their surgical number and print them out either in the coded or in the decoded form. A larger collection of coded reports than the present few must be accummulated before other retrieval and correlation programs can be tested thoroughly. Nevertheless, a primary goal in the design of this sytem was to create a framework for a wide variety of correlative data manipulation procedures. With the present data base structure of CISP, correlations involving demographic data or diagnostic statements may be carried out. When involving PD statements, the correlations may be very specific by referring to the topography, diagnosis and category terms and to modifiers in the comment field of the category term. In conclusion, any data item that appears in a coded form in the stored report may be used to establish correlations. The complexity and variety of these correlationss are large because of the complexity of the dictionary used by CISP.

As described earlier the CISP programs are modular and therefore replacement of modules or expansion of the system can be easily carried out. The segmentation of the system into modules was done by identifying the basic tasks the system has to perform repeatedly. Thus a module exists to look up dictionaries for coding purposes (PATSEK), another to reorganize the entered text (SPLIT), and so on. This segmentation approach has successfully met the challenge of adding an additional dictionary (SNOP) to the system. Major alterations in CISP's processing can also be handled with reasonable effort. These alterations may occur if the form of the entered report is changed (redesign PATREAD), if the dictionary of the system is changed (redesign PATSEK), if the format of the report print-out is altered (redesign PRINT), if the number and type of demographic data is changed (redesign ID and IDHED), etc.

7.4 Improvements, Expansions and Recommendations

The single, most important developmental goal of CISP is to make it operational in the hospital's pathology department. To function within the hospital, CISP must allow the entering of any pathology report and the retrieval of the reports stored. Therefore the first step is to store the entire ICDA and the SNOP-Topography dictionaries. We estimate that in the new IBM

3330-type disk packs presently available, a total of 10 cylinders will be needed to store both the coding and the decoding dictionaries. While the present dictionary is extended, some of the dictionary sections may be concatenated (e.g. CODE and CNEOP). By reducing the number of dictionary sections, simpler decision making processes are needed, thereby reducing execution times.

Experience with CISP in the hospital will certainly yield indications on how to adapt the system for better interaction with the operator. This may necessitate some reprogramming of CISP's interactive phase, altering the input-text error detection and correction routines and changing the form and number of error detection and correction messages. The report pre-editing may be simplified by adding to the coding dictionary the synonyms and equivalents of the already stored diagnostic entries (e.g. "weight loss" and "loss of weight"). This will also simplify the querying of the system by somebody who does not have access to a copy of the system's dictionary.

To profit from the data base, a great variety of querying modules will be programmed. Conversational question-answering will greatly be facilitated by the Time Sharing Option (TSO) which at present is available on McGill's IBM 360/75 OS computer system. TSO allows the writing of programs which converse during execution.

With CRJE only minimal conversation is possible during report entry. The programming of a system to perform question-answer routines in English text would be extremely complex without the availability of the features of TSO.

Many psychological considerations must be accounted for in devising the proper question-answer routines. Extensive experimentation is needed to devise the proper series of questions which help the user to properly formulate his query. The entered question will be processed by the query processing module and the reply will be displayed at the user's terminal.

The query processing modules will be programmed first. While the conversational question-answer editor is still in development, queries may already be answered by relaying them to the system's operator who enters them in a pre-edited form. Accessory modules that help questioning of CISP's data base will also be developed. Among these we include those modules which display a small section of the stored dictionary to acquaint the user with its terminology and structuring. Also included are modules which store the question replies so that they may be used in a series of related exploratory questions.

The large storage capacity of CISP and the ease of report entering will undoubtedly encourage the extension of the content and type of reports entered. There are already plans to

increase the number of demographic data items entered by adding such items as place of birth, medicare number, etc. The OP section may be expanded to include the date of the operation and the name of the surgeon. New sections may be added, as for example one that lists the patient's previous operations. Discharge reports drawn up on all patients leaving the hospital, are not coded at present. These reports, whose annual number is at least twice the number of pathology reports may also be processed and stored using CISP.

And finally, it is hoped that other hospitals will also make use of CISP, or an adapted version of it, to store their own reports. CISP is transportable to any computer system similar to the system shown in Fig. 6.2.

7.4 Summary

A Computerized Information System for Pathology (CISP) was designed, programmed and its data base created. In the course of this work the following were achieved:

1) The unique information flow environment in which this sytem must function in the hospital was analysed. The results of this analysis were summarized in an information flow diagram. In addition, the retrieval and correlation capabilities required for this system were specified.

2) The literature on Pathology Information Systems was reviewed. This review indicated that at present it is not feasible to build a system with both narrative text input and high level correlative retrieval capability. Neither can one build a satisfactory system whose coding dictionary was assembled from terms found in the text of the entered reports. The review suggested that the use of a large medical diagnosis dictionary is most likely to yield a satisfactory system with the desired correlation capability.

3) CISP was implemented with the following features:

a) The system dictionary, consisting primarily of the ICDA dictionary and the SNOP-Topography dictionary, was created and used in the coding and decoding of pathology reports;

b) The pathology report can be entered in a pre-edited text form. The coding of the text is automated using the system's dictionary;

c) The coded reports are stored in a data set allowing direct access to any diagnostic item appearing in the coded form in the report;

d) The decoded report is acceptable for clinical use;

e) A comment field is present in both the entered and the coded report. This field helps overcome the rigidity of expression of the system's dictionary and in certain cases it

extends the correlative capability of the system;

f) The entering of a report for storage and the formulation of the retrieval request is carried out interactively. The actual processing of the entry/retrieval request is carried out in the batch mode;

g) The system is modular. Improvements and expansions may be carried out with reasonable effort;

h) CISP is operational. It is, however, incomplete due to the limitations in time and manpower that exist within the framework of this project; and

i) The system is portable and it may be tailored to the individual requirements of a pathology department.

The successful implementation of the above system has shown that:

a) It is cost-effective to store an ICDA-type dictionary in a computer and to use it for automated coding and decoding purposes in a PIS; and

b) Two or more such dictionaries may be used jointly without necessarily increasing the programming and/or operational complexity of the system. In CISP the same program module is used to search both ICDA and SNOP dictionaries.

Appendix A

PROGRAMS FOR DICTIONARY CREATION

(See section 6.2)

REC

```
01  REC: PROCEDURE OPTIONS(MAIN);
02  DCL (CARD,OL) CHAR(80) VAR;
03  DCL
04       (AK,OLDKEY) CHAR(6) VAR,
05        KEY CHAR(6),
06      1 REC CTL,
07        2 DUMMY BIT(8),
08        2 NO CHAR(6),
09        2 TEXT CHAR(V),
10      V FIXED DEC(2),
11  DECODE FILE RECORD KEYED ENVIRONMENT( INDEXED V(1648,96));
12
13  ON ENDFILE(SYSIN) GO TO PRINT;
14
15  OPEN FILE(DECODE) SEQUENTIAL OUTPUT;
16
17  READ : GET SKIP EDIT(OL) (A(80));
18         IF OL='' THEN GO TO READ;
19  CHOP: CALL SPLIT(OL,CARD);
20           CARD=CARD CAT ' ';
21            ENTRY=INDEX(CARD,' ');
22            AK=SUBSTR(CARD,1,ENTRY-1);
23            CARD=SUBSTR(CARD,ENTRY+1);
24  IF CARD=''  THEN PUT SKIP EDIT('KEY-TEXT OVERLAND IN: ',AK)(A,A);
26           IF AK LT '#' THEN KEY=OLDKEY CAT AK;
27       ELSE KEY,OLDKEY=AK;
28        PUT EDIT(KEY) ( SKIP,A(6));
29     /*   KEY CONTAINS RECORD-KEY;  CARD -- TEXT */
30
31  FORM:   V=LENGTH(K)+1;
32  ALLOCATE REC;
33  DUMMY=(8)'0'B;
34  NO=KEY;
35  TEXT=CARD;
36  WRITE FILE(DECODE) FROM(REC) KEYFROM(REC.NO);
37  FREE REC;
38  GO TO READ;
39
40  PRINT:  CLOSE FILE(DECODE);
41        PUT EDIT('*****END*****') (SKIP,X(3),A);
42  END REC;

01  /*
02  //LKED.SYSLIB DD
03  // DD
04  // DD
05  // DD DSN=B.BE15.PATLIB,DISP=OLD
06  //GO.SYSPRINT DD SYSOUT=A
07  //GO.DECODE DD UNIT=ONLN,DISP=(NEW,CATLG,DELETE),
08  //       DSN=B.BE15.DECODE,SPACE=(CYL,1,,CONTIG),
09  //          DCB=(DSORG=IS,KEYLEN=6,RKP=5,OPTCD=LY,CYLOFL=1,
10  //          RECFM=VB)
11  /*
```

SEQU

```
01  SEQU: PROCEDURE OPTIONS(MAIN);
02  DCL
03      LEAF CHAR(92) VAR,   TRN CHAR(92),
04    SORTIN FILE RECORD SEQUENTIAL ENV(CONSECUTIVE ),
05  DECODE FILE RECORD KEYED ENV(INDEXED );
06  ON ENDFILE(DECODE) GO TO ENDED;
07  OPEN FILE(SORTIN) SEQUENTIAL OUTPUT;
08  OPEN FILE(DECODE) SEQUENTIAL INPUT;
09     CYCLE: READ FILE(DECODE) INTO(LEAF);
10     TRN=LEAF;
11     WRITE FILE(SORTIN) FROM(TRN);
12      GO TO CYCLE;
13  ENDED: CLOSE FILE(DECODE), FILE(SORTIN);
14  PUT SKIP EDIT('FILE ON') (A);
15  END SEQU;
16  /*
17  //GO.SYSPRINT DD SYSOUT=A
18  //GO.SORTIN DD DSNAME=&.BEIS.SORTIN,UNIT=ONLN,
19  //      DCB=(RECFM=FB,LRECL=92,BLKSIZE=1656),
20  //       DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(20,5))
21  //GO.DECODE DD DSNAME=&.BEIS.DECODE,DISP=OLD
22  /*
```

SORTA

```
01  //STEP1 EXEC PL1LFCL,PARM.PL1L='SM=(1,80),NOL,NS,PE,SIZE=99999
,CAR,ST'
02  //PL1L.SYSIN DD *
03
04  SORTA: PROC OPTIONS(MAIN);
05  DCL INSERTA ENTRY(CHAR(361),CHAR(27),FIXED BIN(31,0),FIXED BIN(
31,0)),
06  RETURN_CODE FIXED BIN(31,0);
07
08  CALL INSERTA(' SORT FIELDS=(6,85,CH,4),SIZE=E9000
09    ' RECORD TYPE=F,LENGTH=(92) ',25*' ',RETURN_CODE);
10
11  IF RETURN_CODE=16 THEN PUT SKIP EDIT('SORT FAILED')(A);
12     ELSE IF RETURN_CODE=0 THEN PUT SKIP EDIT('SORT COMPLETED')
(A);
13     ELSE PUT SKIP EDIT('INVALID RETURN_CODE.    CODE=',RETURN_CODE
) (A,F(32));
14  END SORTA;
15  /*
16  //STEP2 EXEC PGM=*.STEP1.LKED.SYSLMOD
17  //SYSOUT DD SYSOUT=A
18  //SYSPRINT DD SYSOUT=A
19  //SORTLIB DD DISP=SHR,DSN=SYS1.SORTLIB
20  //SORTWK01 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
21  //SORTWK02 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
22  //SORTWK03 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
23  //SORTWK04 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
24  //SORTWK05 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
25  //SORTWK06 DD UNIT=ONLN,SPACE=(TRK,(60,20),,CONTIG)
26  //SORTOUT DD DSNAME=&.BEIS.SORTOUT,UNIT=ONLN,
27  //   DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(20,5)),
28  //     DCB=(RECFM=FB,LRECL=92,BLKSIZE=1656)
29  //SORTIN DD DSNAME=&.BEIS.SORTIN,DISP=OLD
30  /*
```

TOKA

```
01  TOKA : PROCEDURE OPTIONS(MAIN);
02  DCL
03      1 X CTL,
04        2 DUMMY BIT(8),
05        2 K CHAR(17),
06        2 NO CHAR(6),
07        2 TEXT CHAR(V),
08  LEAF CHAR(94) VAR,   OZ PIC'99',
09  CARD CHAR(90) VAR,    (1,V) FIXED DEC(2),
10  AK CHAR(6), OLDKEY CHAR(15),   KV CHAR(17) VAR,
11  CODE FILE RECORD KEYED ENVIRONMENT(INDEXED V(1640,99));
12  DCL SORTOUT FILE RECORD INPUT;
13  OLDKEY='';
14  OZ=0;
15  OPEN FILE(CODE) SEQUENTIAL OUTPUT;
16  OPEN FILE(SORTOUT);
17  ON ENDFILE(SORTOUT)  GO TO ENDED;
18  READ: READ FILE(SORTOUT) INTO(LEAF);
19    CARD=SUBSTR(LEAF,8); /* JUST TEXT */
20    AK=SUBSTR(LEAF,2,6);
21
22  DIVIDE: IF LENGTH(CARD) LE 15  THEN V=8;
23          ELSE V=LENGTH(SUBSTR(CARD,16));
24        ALLOCATE X;
25        KV=SUBSTR(CARD,1,15);
26        IF KV=OLDKEY  THEN DO ;
27              OZ=OZ+1;
28              KV=KV CAT OZ;
29              K=KV;
30           END;
31         ELSE DO;
32              OZ=0;
33             OLDKEY=KV;
34              KV=KV CAT OZ;
35              K=KV;
36              PUT SKIP EDIT(K) (A);
37              END;
38  NO=AK;
39  IF LENGTH(CARD) LE 15 THEN TEXT='';
40  ELSE TEXT=SUBSTR(CARD,16);
41  DUMMY='00'B;
42  WRITE FILE(CODE) FROM(V) KEYFROM(K);
43   FREE X;
44  GO TO READ;
45
46  ENDED: CLOSE FILE(CODE);
47      END TOKA;
01  /*
02  //GO.SYSPRINT DD SYSOUT=A
03  //GO.CODE DD UNIT=DWLN,DISP=(NEW,CATLG,DELETE),
04  //         DSN=S.BEIS.CODE,SPACE=(CYL,(1,,CONTIG),
05  //             DCB=(DSORG=IS,KEYLEN=17,RKP=5,OPTCD=LY,CYLOFL=1,
06  //             RECFM=VB)
07  //GO.SORTOUT DD DSN=S.BEIS.SORTOUT,DISP=OLD
08  /*
```

Appendix B

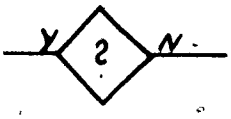
CONVENTIONS AND ABBREVIATIONS USED IN FLOW CHARTING

input, output

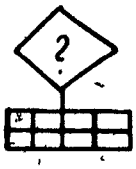
processing

processing with reference to program name

set monitors

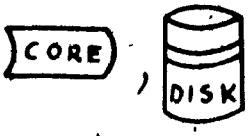


decision



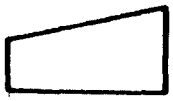
multiple decision

annotation

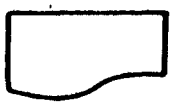


on-line storage devices

manual off-line operation



terminal



document

magnetic tape

punched card

assignment symbol

concatenate

symbol for blank character

digit 0 (zero)

char. — character

msg. — message

er. — error

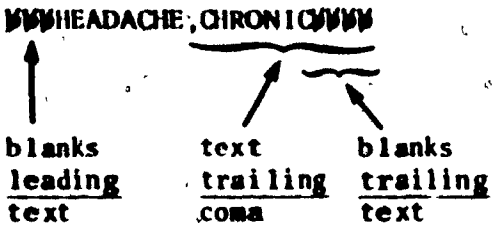
del. — delimiter

var. — variable

frag. — fragment

no. — number

com., subcom. — command, subcommand



Appendix C

CISP OPERATING INSTRUCTIONS

```
-- W E L C O M E  T O  C I S P II --
....COMPUTERIZED INFORMATION SYSTEM FOR PATHOLOGY

TO ENTER A PATHOLOGY REPORT FOR CODING AND STORAGE TYPE IN ANY OF:
     X ONE   OR   X TWO   OR   X THREE     , ETC.
AFTER YOUR REPORT IS TYPED IN CHANGES CAN BE MADE WITHIN THE REPORT.
E.G. :
     CHANGE 2 /OUT/OUT OF/
WILL CHANGE WORD "OUT"  TO "OUT OF" IN LINE 2 OF THE REPORT. AND THEN:
     LIST 2
WILL LIST LINE 2 TO CHECK THE CHANGES MADE.
WHEN ALL CHANGES ARE TERMINATED , TYPE IN:
     SAVE
     END
AND YOU ARE READY TO HAVE THE REPORT CODED AND STORED. FOR THIS TYPE IN
     X CODE!
.....IF YOU USED "X ONE" ORIGINALLY .
IF LATER ON YOU WISH TO CHANGE THIS REPORT AND RE ENTER IT FOR CODING
AND STORAGE , TYPE IN:
     X LIST!
AND AFTER CHANGES ARE MADE WITHIN THE REPORT , TYPE IN:
     SAVE
     END
     X CODE!

WHENEVER YOU TYPE IN:
     X ONE
THE PREVIOUS REPORT TYPED IN IS ERASED AND CISP IS READY TO ACCEPT
YOUR NEW REPORT .

TO OBTAIN YOUR OUTPUT SIMPLY TYPE IN :
     OUTPUT ONE

TO RETRIEVE A REPORT IN CODED FORM TYPE IN:
     X PRINTC
AND FOR THE DECODED FORM TYPE IN :
     X PRINT
FOLLOWED BY THE REPORT IDENTIFICATION. THE REPORT IS PRINTED
BY TYPING IN :
     OUTPUT REPORT

AND FINALLY, IN BETWEEN ANY INPUT OPERATIONS IF YOU WOULD LIKE TO
RECEIVE THESE INSTRUCTIONS AGAIN , TYPE IN :
     X STORY
```

In addition to the above, the system's operator must have an awareness of the form in which the pre-edited report is to be entered and the various delimiters that are to be used (section 6.3)

Appendix D

CISP MAINTENANCE PROGRAMS

TRANS: Creates and copies into B.BE15.DECODE2 the content of B.BE15.DECODE

```
01  TRANS:PROCEDURE OPTIONS(MAIN);
02  DCL
03  LEAF CHAR(2888) VAR,
04  DECODE2 FILE RECORD KEYED ENV(INDEXED V(1648,96)   ),
05  DECODE FILE RECORD KEYED ENV(INDEXED  ),
06  1 REC CTL,
07   2 DUMMY BIT(8),
08   2 NO CHAR(7),
09   2 TEXT CHAR(V),
10  V FIXED DEC(3);
11
12  ON ENDFILE(DECODE) GO TO ENDED;
13  OPEN FILE(DECODE) SEQUENTIAL INPUT;
14  OPEN FILE(DECODE2) SEQUENTIAL OUTPUT;
15  CYCLE:  READ FILE(DECODE) INTO(LEAF);
16          L:  V=LENGTH(SUBSTR(LEAF,9));
17          ALLOCATE REC;
18          DUMMY=(8)'0'B;
19          NO=SUBSTR(LEAF,2,7);
20          TEXT=SUBSTR(LEAF,*);
21          WRITE FILE(DECODE2) FROM(REC) KEYFROM(REC.NO);
22          FREE REC;
23          GO TO CYCLE;
24  ENDED: CLOSE FILE(DECODE),FILE(DECODE2); PUT SKIP EDIT('*END*'
) (A);
25       END TRANS;
26  /*
27  //GO.SYSPRINT DD SYSOUT=R
28  //GO.DECODE2 DD UNIT=ONLN,DISP=(NEW,CATLG,DELETE),
29  //       DSN=B.BE15.DECODE2,SPACE=(CYL,1,,CONTIG),
30  //          DCB=(DSORG=IS,KEYLEN=6,RKP=5,OPTCD=LY,CYLOFL=1,
31  //              RECFM=VB)
32  //GO.DECODE DD DSNAME=B.BE15.DECODE,DISP=OLD
33  /*
```

RENAME: Renames DECODE2 to DECODE

```
01  //GO EXEC PGM=IEHPROGM
02  //SYSPRINT DD SYSOUT=R
03  //SYSTEM DD DSNAME=SYS1.SVCLIB,DISP=OLD
04  //DD2 DD UNIT=ONLN,VOL=SER=4TS001,DISP=OLD
06  //GO.SYSIN DD *
07   RENAME DSNAME=B.BE15.DECODE2,VOL=2314=4TS001,
     X
08                      NEWNAME=B.BE15.DECODE
09   UNCATLG DSNAME=B.BE15.DECODE2
10   CATLG DSNAME=B.BE15.DECODE,VOL=2314=4TS001
11  /*
```

TOTRANS: Copies B,BE15.CODE into B,BE15.CODE2

```
01  TOTRANS: PROC OPTIONS(MAIN);
02  DCL
03  LEAF CHAR(182) VAR,
04  CODE2 FILE RECORD KEYED ENV(INDEXED V(1648,99)),
05  CODE FILE RECORD KEYED ENV(INDEXED ),
06  1 X CTL,
07   2 DUMMY BIT(8),
08   2 K CHAR(17),
09   2 TEXT CHAR(V),
10  AK CHAR(17) , V FIXED DEC(2);
11
12  ON ENDFILE(SYSIN) GO TO ENDED;
13  OPEN FILE(CODE) SEQL INPUT;
14  OPEN FILE(CODE2) SEQL OUTPUT;
15
16  CYCLE: READ FILE(CODE) INTO(LEAF);
17         V=LENGTH(SUBSTR(LEAF,19));
18         ALLOCATE X;
19         DUMMY=(8)'0'B;
20         K=SUBSTR(LEAF,2,17);
21         TEXT=SUBSTR(LEAF,19);
22         WRITE FILE(CODE2) FROM(X) KEYFROM(X.K);
23         FREE X;
24  GO TO CYCLE;
25  ENDED: CLOSE FILE(CODE),FILE(CODE2);
26       END TOTRANS;
27  /*
28  //GO.SYSPRINT DD SYSOUT=A
29  //GO.CODE2 DD UNIT=DISK,DISP=(NEW,CATLG,DELETE),
30  //       DSN=B.BE15.CODE2,SPACE=(CYL,1,,CONTIG),
31  //         DCB=(DSORG=IS,KEYLEN=17,RKP=5,OPTCD=LY,
32  //       CYLOFL=1,RECFM=VB)
33  //GO.CODE DD DSNAME=B.BE15.CODE,DISP=OLD
34  /*
```

BLOW: Prints dictionary DNEOP

```
01  BLOW: PROC OPTIONS(MAIN);
02  DCL     LEAF CHAR(182) VAR, P FIXED DEC(2),  AF CHAR(6),
03  DECODE FILE RECORD KEYED ENV(INDEXED);
04  ON ENDFILE(DECODE) GO TO ENDED;
05  OPEN FILE(DECODE) SEQL INPUT;
06  NEXT:    READ FILE(DECODE) INTO(LEAF) ;
07         PUT SKIP(2);
08  AF=SUBSTR(LEAF,2,6);
09  LEAF=SUBSTR(LEAF,8);   P=INDEX(AF,' ')-1  ;
10  IF P=3 THEN DO;
11        PUT EDIT(AF,LEAF)  (A(6),X(5),A);
12        END;
13  IF P=4 THEN DO;
14        PUT EDIT(AF,LEAF)  (X(1),A(6),X(5),A);
15        END;
16  IF P=5 THEN DO;
17        PUT EDIT(AF,LEAF)  (X(2),A(6),X(5),A);
18        END;
19  GO TO NEXT;
20  ENDED: CLOSE FILE(DECODE);
21  END BLOW;
22  /*
23  //GO.SYSPRINT DD SYSOUT=A
24  //GO.DECODE DD DSN=B.BE15.DNEOP,DISP=OLD
25  /*
```

TOBLOW: Prints CNEOP

```
01  TOBLOW: PROC OPTIONS(MAIN);
02  DCL    LEAF CHAR(128) VAR,     AK CHAR(6),
03  CODE FILE RECORD KEYED ENV(INDEXED);
04
05  ON ENDFILE(CODE) GO TO ENDED;
06  OPEN FILE(CODE) SEQL INPUT;
07  R: READ FILE(CODE) INTO(LEAF);
08        IF LENGTH(LEAF) GT 24 THEN DO ;
09          PUT SKIP(2);
10          PUT SKIP EDIT(SUBSTR(LEAF,2,15),SUBSTR(LEAF,17,2),SUBS
    TR(LEAF,19,6),SUBS
11  TR(LEAF,25))    (A(15),X(2),A(2),X(2),A(6),X(2),A);
12          END;
13  ELSE DO ;
14  PUT SKIP(2);
15          PUT SKIP EDIT(SUBSTR(LEAF,2,15),SUBSTR(LEAF,17,2),SUBS
    TR(LEAF,19,6))
16       (A(15),X(2),A(2),X(2),A(6));
17  END;
18  GO TO R;
19  ENDED: CLOSE FILE(CODE);  END TOBLOW;
20  /*
21  //GO.SYSPRINT DD SYSOUT=A
22  //GO.CODE DD DSN=B.BE15.CNEOP,DISP=OLD
23  /*
```

JCOPT

```
18  // EXEC PLILFCL,PARM.PLIL='SM=(1,63),NOL,OPT=2,PE,SIZE=999999,
    C48,ST',
29  // PARM.LKED='MAP,LIST,LET,NCAL'
30  //PLIL.SYSPRINT DD SYSOUT=A.
40  //PLIL.SYSIN DD *
```

LIB: Creates partitioned library B.BE15.PATLIB. Adds module SPLIT to this library.

```
81  /*
82  //LKED.SYSLMOD DD DSN=B.BE15.PATLIB(SPLIT),DISP=OLD
83  //    DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(1,2,2))
84  /*
```

LIN: Adds module PRT2 to library

```
81  /*
82  //LKED.SYSLMOD DD DSN=B.BE15.PATLIB(PRT2),DISP=OLD
83  /*
```

LINREP: Replaces module PRINT with a new version .

```
01  /*
02  //LKED.SYSLMOD DD DSN=B.BE15.PATLIB,
03  //   DISP=SHR,SPACE=
04  //LKED.SYSIN DD *
05    NAME PRINT(R)
06  /*
```

COMPRESS

```
10  // EXEC COMPRESS,NAME='B.BE15.PATLIB',O=R
20  //
```

RELEASE

```
01  // EXEC RELEASE,N='B.BE15.PATLIB',SPACE=CYL
02  /*
```

LOADCOPY: Copies library B.BE15.LIBLNK into a new library B.BE15.PATLIB

```
01  // EXEC PGM=IEBCOPY
02  //SYSPRINT DD SYSOUT=R
03  //A DD DSN=B.BE15.LIBLNK,DISP=SHR
04  //B DD DSN=B.BE15.PATLIB,SPACE=(TRK,(20,4,8)),
05  // DISP=(NEW,CATLG,DELETE),UNIT=DASD
06  //SYSIN DD *
07    C I=A,O=B
08  /*
```

SAVE

```
01  // EXEC NEWFILE,NAME='B.BE15.MODJ',O=R,P=1,B=1
02  //SYSUT1 DD DATA
```

LCTLG: Lists the names of data sets in the catalog

```
10  // EXEC LIST,O=A
20  //SO.SYSIN DD *
30      LISTCTLG MODE=B.BE15
40  /*
```

DELETE: Deletes a data set

```
01  //  EXEC PGM=IEFBR14
02  //SYSPRINT DD SYSOUT=R
03  //DD1 DD DSN=B.BE15.PATREP2,DISP=(OLD,DELETE)
```

GPD: Executes the program module that processes a pathology report.

```
01  // EXEC PLILFLG
02  //LKED.SYSLIB DD
03  // DD
04  // DD
05  // DD DSN=B.BE15.PATLIB,DISP=OLD
06  //LKED.SYSUT1 DD DISP=NEW,SPACE=(CYL,(10,2))
07  //LKED.SYSIN DD *
08    INCLUDE SYSLIB(PATREAD)
09    ENTRY INENTRY
10  //GO.SYSPRINT DD SYSOUT=R
11  //GO.PATREP DD DSN=B.BE15.PATREP,DISP=OLD
12  //GO.CODE DD DSN=B.BE15.CODE,DISP=OLD
13  //GO.OPERC DD DSN=B.BE15.OPERC,DISP=OLD
14  //GO.CTOPOG DD DSN=B.BE15.CTOPOG,DISP=OLD
15  //GO.CNEOP DD DSN=B.BE15.CNEOP,DISP=OLD
16  //GO.SYSIN DD *
17
```

GPRT: Executes the program module that prints out the pathology report in a decoded form.

```
01  // EXEC PLILFLG
02  //LKED.SYSLIB DD
03  // DD
04  // DD
05  // DD DSN=B.BE15.PATLIB,DISP=OLD
06  //LKED.SYSIN DD *
07    INCLUDE SYSLIB(PRT)
08    ENTRY INENTRY
09  //GO.SYSPRINT DD SYSOUT=R
10  //GO.PATREP DD DSN=B.BE15.PATREP,DISP=OLD
11  //GO.DECODE DD DSN=B.BE15.DECODE,DISP=OLD
12  //GO.OPERD DD DSN=B.BE15.OPERD,DISP=OLD
13  //GO.DTOPOG DD DSN=B.BE15.DTOPOG,DISP=OLD
14  //GO.DNEOP DD DSN=B.BE15.DNEOP,DISP=OLD
15  //GO.SYSIN DD *
16
```

References

Anonymous (1965)
Systematized Nomenclature of Pathology
College of American Pathologists, 1st ed., 1965

Anonymous (1966a)
Stedman's Medical Dictionary
21st ed., The Williams and Wilkins Company, Baltimore 1966

Anonymous (1966b)
Introduction to IBM Direct-Access Storage Devices and Organization Methods
IBM Corp. 1966, GC20-1649-5

Anonymous (1970a)
IBM System/360 Operating System,
Conversational Remote Job Entry
Terminal User's Guide
IBM Corp. 1970, GC30-2014-0

Anonymous (1970b)
IBM System/360 Operating System
PL/1 (F)
Language Reference Manual
IBM Corp., 1970, GC28-8201-3

Anonymous (1972a)
Hospital Adaptation of ICDA (Vol. 1 & 2)
Commission on Professional and Hospital Activities, 1st ed.
Ann Arbor, Michigan 1972

Anonymous (1972b)
IBM System/360 Operating System
PL/1 (F)
Programmer's Guide
IBM Corp., 1972, GC28-6594-8

Anonymous (1973)
PL/1: Where Are You Now ?
Datamation 19(1): 103-105

Barnhard, H.J., Long, J.M. (1966)
Computer Autocoding, Selecting and Correlating of Radiologic Diagnostic Cases: A Preliminary Report
The Am. J. of Roentgenology, Radium Therapy and Nuclear Medicine 96(4): 854-863

Barnhard, H.J., Long, J.M., Lang, L. (1968)
The Automatic Coding, Selecting and Correlation of Patient Data in Radiology: A Progress Report
in: Data Acquisition and Processing in Biology and Medicine (Vol.5) Proceedings of the 1966 Rochester Conference, K. Enstein (Ed.), pp. 15-20, Pergamon Press Inc. 1968

Bohrod, M.G. (1971)
What is Pathologic Diagnosis ? A Prelude to Computer Diagnosis
Pathology Annual 6: 197-208

Brown, G.D. (1970)
System/360 Job Control Language
John Wiley & Sons, Inc. 1970

Carville, M., Higgins, L.D., Smith, F.J. (1971)
Interactive Reference Retrieval in Large Files
Information Storage and Retrieval 2: 205-210

Chapin, N. (1970)
Flowcharting With the ANSI Standard: A Tutorial
Computing Surveys 2(2): 119-146

Chomsky, N. (1965)
Aspects of the Theory of Syntax
The MIT Press 1965

Corbato, F.J. (1969)
PL/1 as a Tool for System Programming
Datamation 15(5): 68-76

Dodd, G.G. (1969)
Elements of Data Management Systems
Computing Surveys 1(2): 117-133

Gaynon, P., Wong, R.L. (1972)
A Retrieval System for a Library of Pathology Reports, Slides and Kodachromes
Meth. of Info. in Med. 11(3): 152-162

Grams, R. (1971)
Pathology, Digital Computers and Planning in Coordinating Health Care Efforts (Part II)
Laboratory Medicine 2(12): 33-42

Hercz, L., Laszlo, C.A., Reesal, M., (1972)
System Analysis and Data Organization in a Computerized Pathology Information System
4th Can. Med & Biol. Engineering Conf., Sept. 1972, p.23

Jordan, E.P. (1947)
Standard Nomenclature of Diseases and Standard Nomenclature of Operations
3d. Ed., The Blakistan Company 1947

Knuth, D.E. (1972)
The History of Sorting
Datamation 18(12): 64-70

Korein, J., Tick, L.J., Woodbury, N.A., Cady, L., Goodgold, A., Randt, C.T. (1963)
Computer Processing of Medical Data by Variable-Field-Length Format
JAMA 186(2): 132-138

Krieg, A.F., Henry, J.B., Stratakos, S.M. (1968)
Analysis of Clinical Pathology Data by Means of a User-Oriented On-Line Data System
in: Data Acquisition and Processing in Biology and Medicine (Vol.5) Proceedings of the 1966 Rochester Conference, K. Enslein (Ed.), pp. 163-172
Pergamon Press Inc. 1968

Lamson, B.G. (1965)
Computer Assisted Data Processing in Laboratory Medicine
in: Computers in Biomedical Research (Vol.1), R.W. Stacy and B.D. Waxman (Eds.), pp. 353-376
Academic Press, N.Y. 1965

Lamson, B.G., Dimsdale, B. (1966)
A Natural Language Information Retrieval System
Proc. of the IEEE 54(2): 1636-1640

Lindberg, A.B. (1965)
Electronic Processing and Transmission of Clinical Laboratory Data
Missouri Medicine 62(4): 296-302

Martin, J. (1973)
Design of Man-Computer Dialogues
Prentice-Hall, Inc. 1973

Paplanus, S.H., Shepard, R.H., Zvargulis, J.E. (1969)
A Computer-Based System for Autopsy Diagnosis Storage and Retrieval Without Numerical Coding
Laboratory Investigation 20(2): 139-146

Pratt, A.W., Thomas, L.B. (1966)
An Information Processing System for Pathology Data
Pathology Annual 1:1-21

Reesal, M., Laszlo, C.A., Hercz, L. (1970)
Computerized Pathological Information System for the Department of Pathology, Montreal General Hospital,
Internal Report, 1970.

Rubey, R.R. (1968)
A Comparative Evaluation of PL/1
Datamation 14(12): 22-25

Smith, J., Melton, J. (1963)
Automated Retrieval of Autopsy Diagnoses by Computer Technique
Meth. of Info. in Med. 2(3): 85-90

Smith, J., Melton, J. (1964)
Manipulation of Autopsy Diagnosis by Computer Technique
JAMA 188(11): 958-962

Van der Esch, E.P. (Ed.) (1972)
Symposium on the Use of Computers for the Classification of Pathological Diagnosis
Path. Europ. 7(2): 177-200

Wong, R.L., Gaynon, P. (1971)
An Automated Parsing Routine for Diagnostic Statements of Surgical Pathology Reports
Meth. of Info. in Med. 10(3): 168-175