OCCASION

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

DISCLAIMER

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

FAIR USE POLICY

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

CONTACT

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

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

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

pl

D04556

Distr.
LIMITED

ID/WG.137/51
18 September 1972

ORIGINAL: ENGLISH

United Nations Industrial Development Organization

Symposium on the Development of the Plastics
Fabrication Industry in Latin America

Bogotà, Colombia, 20 November - 1 December 1972

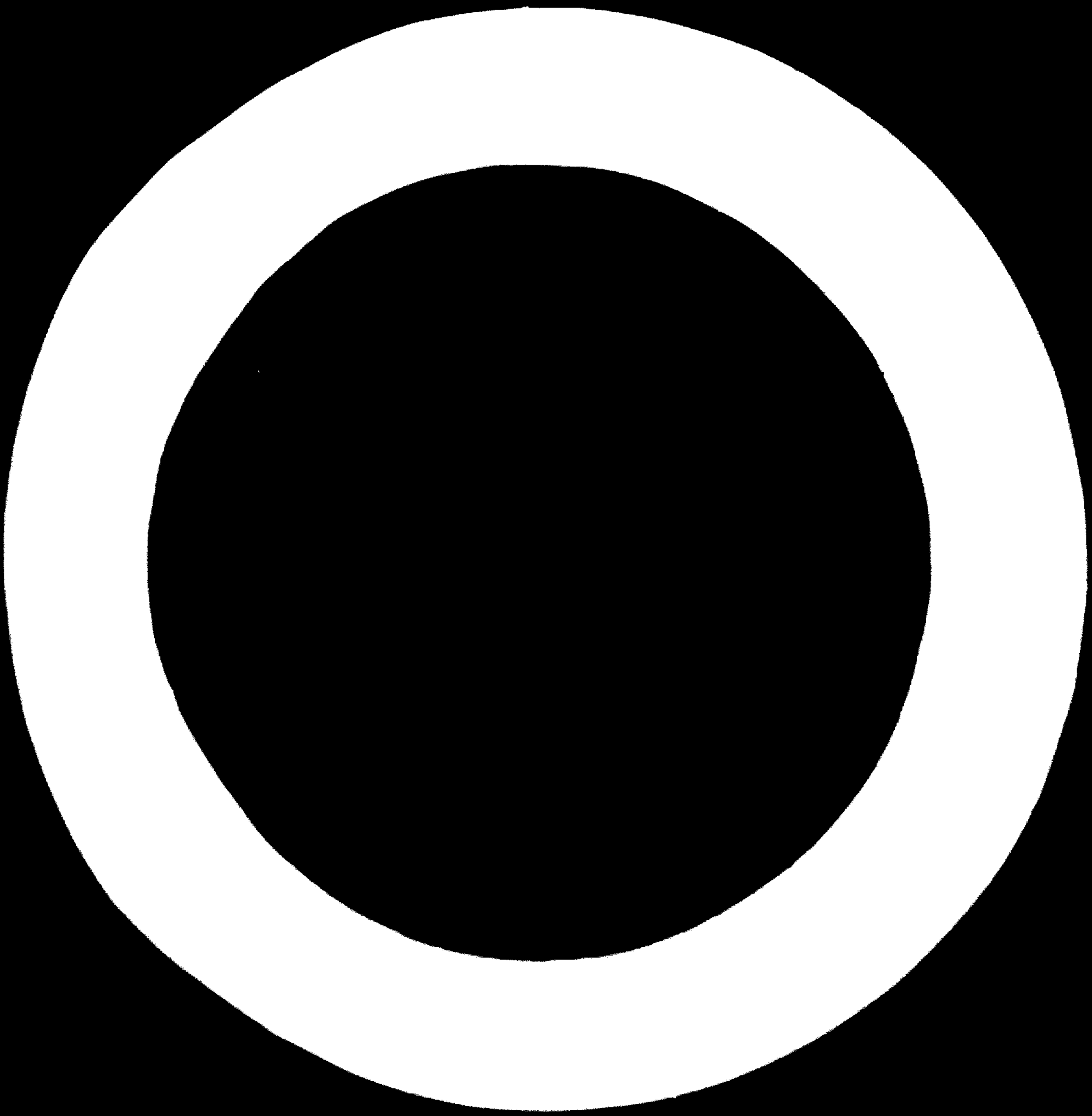
THE APPLICATION OF COMPUTERS TO THE PLASTICS INDUSTRY[1]

by

A. Schwab
AFCA
Geneva, Switzerland

[1] The views and opinions expressed in this paper are those of the author and do not necessarily reflect the views of the secretariat of UNIDO. This document has been reproduced without formal editing.

We regret that some of the pages in the microfiche copy of this report may not be up to the proper legibility standards, even though the best possible copy was used for preparing the master fiche.



THE APPLICATION OF COMPUTERS TO THE PLASTICS INDUSTRY

The title of this lecture is very challenging, and there is a vast complex of interesting questions which could be discussed in relation to the application of computer systems. However, a twenty-minute lecture means drastic limitation of the discussion for a few points of general interest.

I have therefore selected the following three main topics:

1. How to optimize computer systems
2. The important role of computer software
3. Improved communication with the computer

Before we go any further, it might be worth while to agree on the definition of some of the terms. When I speak of "the computer system", I refer to the overall system, comprising the human being, operational objectives, computer hardware and computer software. The term "brainware" is introduced to designate the written computer-oriented task requirements, as distinct from computer software. Finally, there is the term "optimization": this is the process of defining a computer system configuration which meets the established operational requirements at minimum cost, taking long-range requirements into account. This leads into our first subject:

(Chart No 1)

How to optimize computer systems

Effective computer systems have to be tailored to the individual needs of the user. In order to illustrate the multi-component management data flow within a given industrial organization, let us have a look at the following management pyramid. We recognize that this pyramid is built on the operational base directed by operational management, since it is operational management which determines the tactics. Strategy will be carried out by middle management, and corporate policy by top management.

Today, every company faces the problem of where to start using computers. In order to remain competitive, it is wise for every company to establish a long-term plan for a satisfactory computer system.

(Chart No 2)

For the past few years, everyone has been talking about management information systems, which are certainly the goal for optimum computer utilization. Since such a challenging goal will not be reached in one step, any long-term plan should define individual milestones for building up an optimum computer system step by step. This build-up has to start with data acquisition in the individual departments such as production, procurement, engineering, research and development, accounting and finance, as well as the personnel department. Advanced systems will use terminals for automatic data acquisition which are either directly connected with a main computer or linked with satellite or process computers which prepare the data for the main computer.

Very often, these steps to automatic data acquisition are too costly to begin with. If so, data acquisition may very well be started using punched cards, punched tapes or magnetic tapes, which are hand-carried to the central computer station. A growing computer system which is one day to become a management information system will require a data base which stores all the data needed for the various information formats, in such a way that the desired information format can be generated merely with random access to the individual data.

(Chart No 3)

The following chart shows a milestone plan which I now want to use to emphasize the importance of strong program management, which I think is the only way to achieve an optimized computer concept. Before any program team can start its work, the short- and long-range objectives have to be established by the user's management.

Once a program manager has been designated and the team set up, or when a competent sub-contractor has been selected, a system study phase may be carried out. After evaluation of the system study report, the user's management can either confirm or revise the main objectives and decide on the following phases. The same routine may be used after completion of the definition phase, which should lead to the go-ahead for the procurement phase. The last phase of such a computer introduction program is the installation and training phase. If these program phases are conducted carefully and the general rules of program management observed, there should be a good chance of ending up with a computer system which is optimized for an individual user.

(Chart No 4)

Before we turn to the topic "Computer software", let us have a look at the history of computers. As we all know, computer life started in the 1940s. The so-called first generation lasted for almost 15 years, up to 1960, and was equipped with vacuum tubes. The first computers were programmed in machine language. Later on, the first assemblers were developed.

The second generation lasted for only 5 years. It was equipped with transistors and discrete electronic components. The memory-access time was reduced to 10 microseconds. It was at this stage that the need for standardized computer languages, called compiler languages, was recognized. The first compiler languages and operation systems for internal computer control were introduced.

The third generation began to appear in 1965. This generation is equipped with integrated circuits. The memory-access time has been reduced right down to 0,5 microseconds, which is an improvement by the factor of 20. Other characteristics of this generation are multi-programming, time-sharing and remote data processing.

Since 1968, there is a so-called modified third generation which now employs medium- and large-scale integrated electronic circuits. Memory-access time has again been improved, this time by a factor of 2. Large computer systems of this generation are now capable of processing support programs in parallel.

The development of computer hardware in recent years has been fantastic. But what does it mean with regard to computer application by individual companies?

There are extremely powerful computers on the market, which are fast and have a very high data-processing capacity. But the efficient use of these powerful computers depends on the software packages available. It can be safely stated that the electronic computer hardware has left software development far behind.

(Chart No 5)

Let us recall how computer programming started.

The first level was the machine language itself. In the next echelon we find the so-called assembler languages, which were developed by each computer manufacturer for his own computer systems. These languages allow for programming with alpha-numerical instructions, which are automatically translated by the computer using the assembler. Users of a large number of different types of computers very soon demanded programs with a common and easily understandable structure for all types of computers. The compiler languages, such as COBOL, FORTRAN, ALGOL and others, were developed and introduced. The main advantage of the compiler languages, apart from standardization, is the fact that they are problem-oriented and easier to learn than assembler languages.

(Chart No 6)

What more can be done to improve the standard of software developments? There is quite a number of promising starts today, one of which I would like to bring to your attention.

Existing compilers can be greatly improved by the use of converters. Let me demonstrate this possibility by taking the example of a typical converter language designed on the basis of COBOL compiler language. The employment of this converter language has so greatly reduced the training time for programmers that whereas 20 days used to be required to train COBOL programmers, it is now possible to let an untrained person generate a COBOL program in the converter language after only 2 days' instruction. The programming cost for the generation of COBOL programs can be reduced by 50% to 95%, depending on the given task. In addition to this advantage, all the basic COBOL advantages are maintained.

(Chart No 7)

The following chart illustrates how it is possible to reduce the effort required for COBOL programming by using our converter language example. It should be noted that 10 frames of a converter language program will subsequently generate 200 frames of the corresponding COBOL program. On this chart you will find the typical converter instructions, such as Use, Give List, Containing, Sum of, which are complemented by the individual task parameters.

(Chart No 8)

Let me explain the working principle of our converter using the following charts. The converter program, stored on punched cards, is complemented by the file descriptions stored in the so-called file manager, and is subsequently converted into a COBOL source program by using the intelligence of the converter program. The conversion results can themselves be stored on a disk-file, on punched cards or printed out in list form. The development of our COBOL converter program required four man-years of work by highly qualified compiler experts.

(Chart No 9)

Now to our last subject: "Improved communication with the computer".

Let us use the following figure to discuss a desirable dialogue between a manager and his computer system. Assuming that the manager talks through a microphone, the instructions given or the requests made are fed to a decision logic. This logic then decides whether data retrieval can start immediately, that is, if the information is already available, or whether a dialogue is required, that is, if existing information formats have to be modified using existing programs.

The third and last possibility would be the generation of new programs to solve a given task.

The most important requirement for such a system is the data base already discussed. The information results and/or instructions given by the computer system will be displayed audio-visually to the manager. One of the fundamental and essential components in such a management information system will be the generation of brainware, which is a new term for computer-oriented task requirements. Obviously, there is a long way to go before we are able to implement such a system. However, it is important for every company to orient its long-range planning towards this challenging goal.

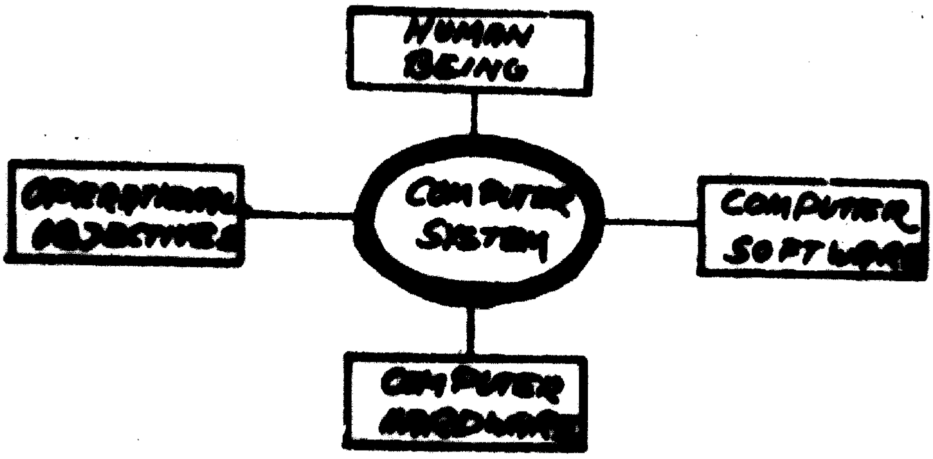
(Chart No 10)

• • • • • • •

Axel Schwab
98, Rue de Saint-Jean
CH - 1200 Genève 11
Tel. 31.82.50
Telex 23 446 Profi Ch.

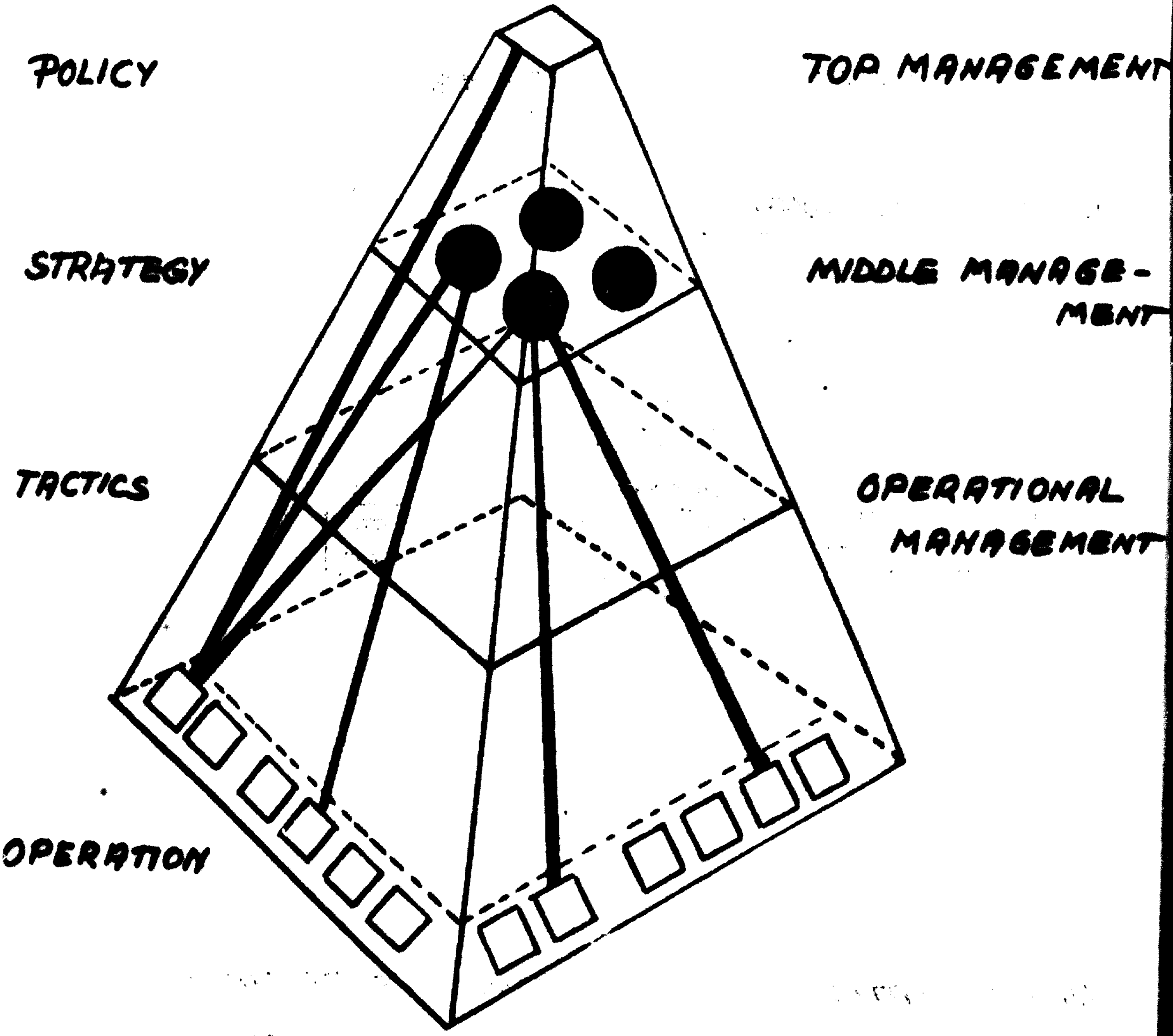
DEFINITION OF TERMS

COMPUTER SYSTEM



OPTIMIZATION - PROCESS OF DEFINING A COMPUTER SYSTEM CONFIGURATION WHICH MEETS THE ESTABLISHED OPERATIONAL REQUIREMENTS AT MINIMUM COST TAKING LONG RANGE REQUIREMENTS INTO ACCOUNT

MANAGEMENT PYRAMID



THE OPERATIONAL BASE IS ALSO THE BASE FOR EVERY MIS

HENCE

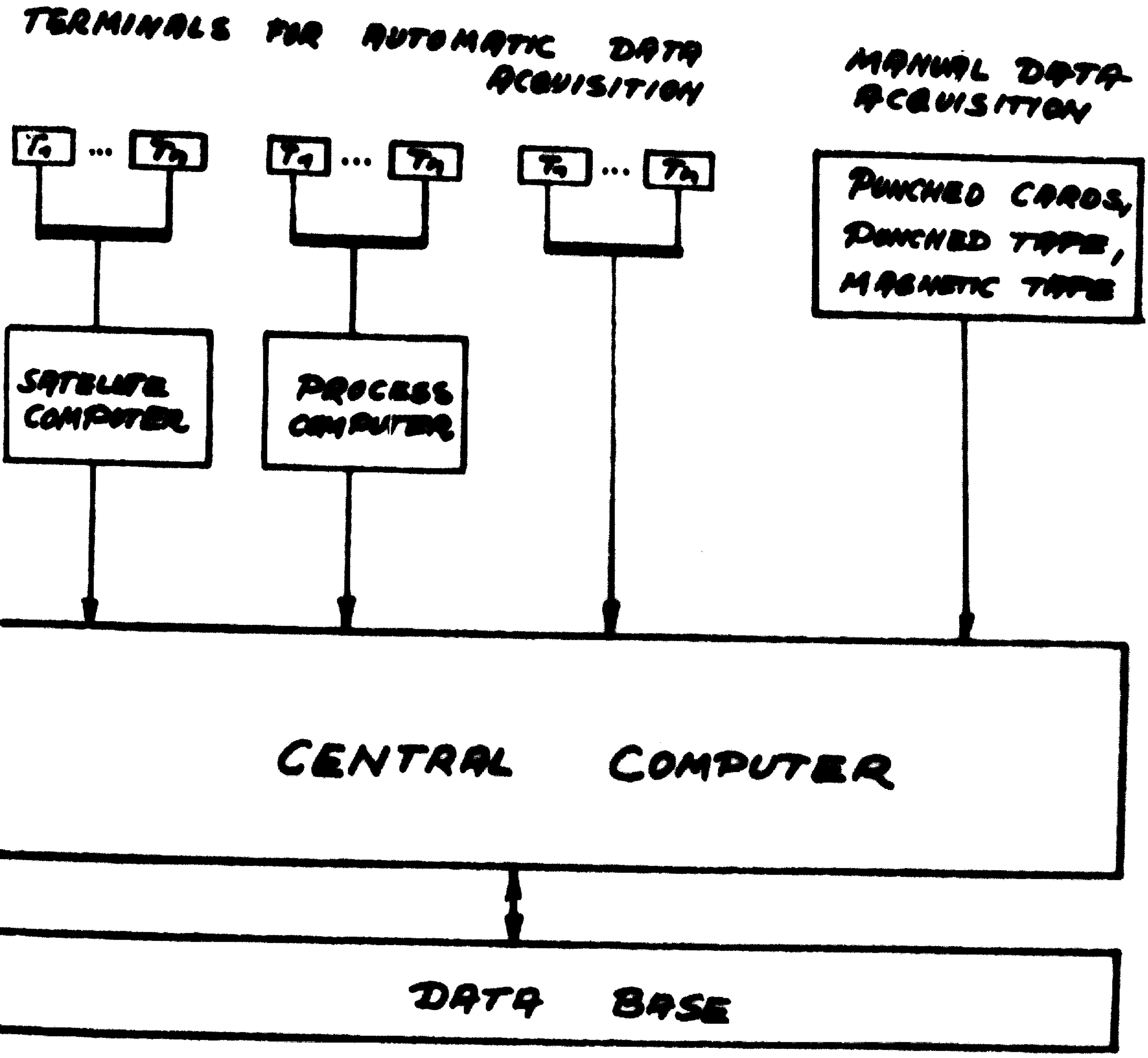
COMPUTERIZATION HAS TO BEGIN WITH DATA ACQUISITION IN THE OPERATIONAL DEPARTMENTS SUCH AS

PRODUCTION ENGINEERING ACCOUNTING
PROCUREMENT RESEARCH & DEVELOPMENT & FINANCE
PERSONNEL

WHICH SYSTEM MEETS THE REQUIREMENTS

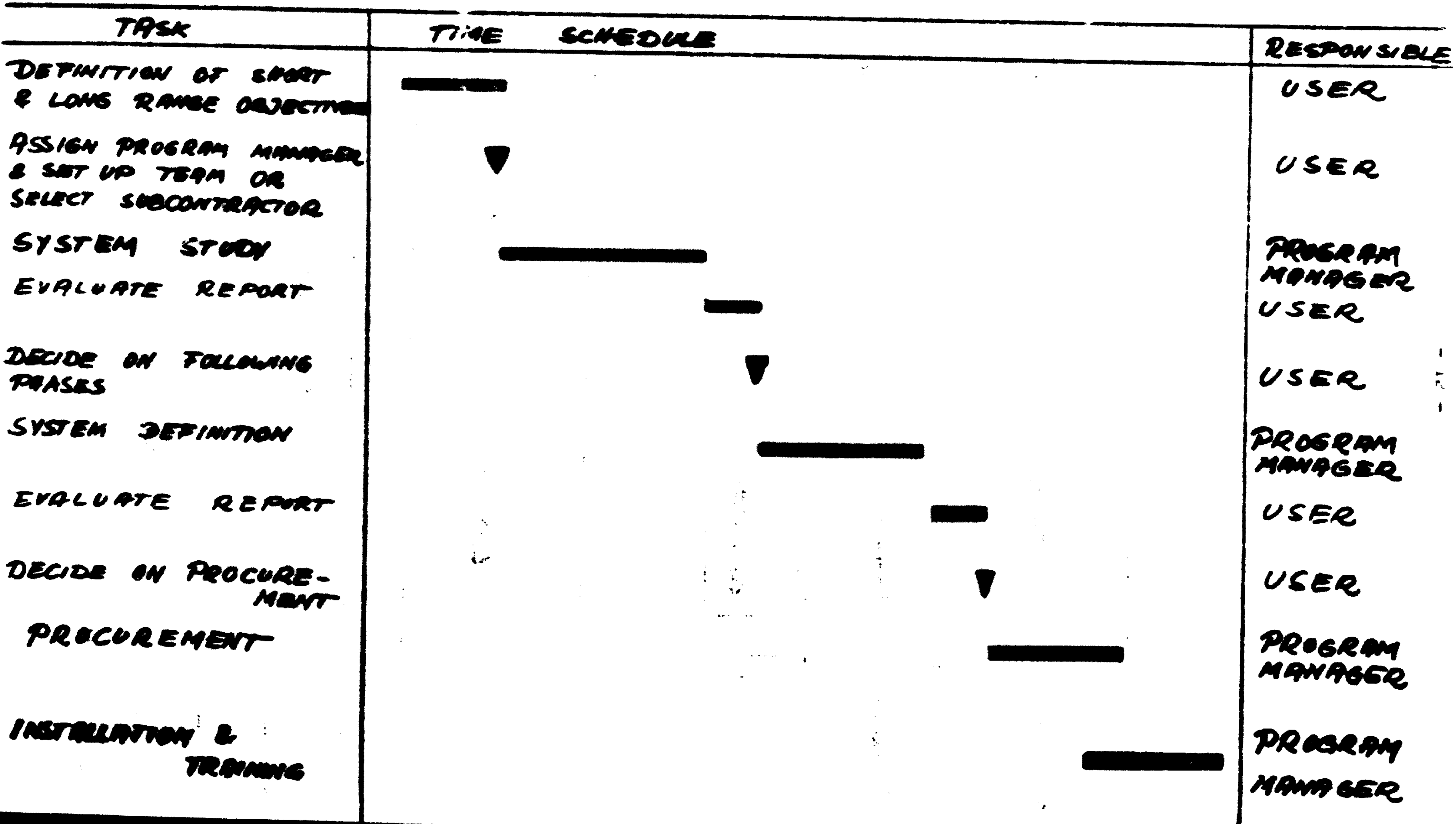
TODAY ?

TOMORROW ?



DATA STORAGE IN A DATA BASE FORMAT IS AN ABSOLUTE REQUIREMENT FOR POWERFUL INFORMATION SYSTEMS

STRONG PROGRAM MANAGEMENT LEADS TO AN OPTIMIZED COMPUTER CONCEPT



TASK	TIME SCHEDULE	RESPONSIBLE
DEFINITION OF SHORT & LONG RANGE OBJECTIVES		USER
ASSIGN PROGRAM MANAGER & SET UP TEAM OR SELECT SUBCONTRACTOR		USER
SYSTEM STUDY		PROGRAM MANAGER
EVALUATE REPORT		USER
DECIDE ON FOLLOWING PHASES		USER
SYSTEM DEFINITION		PROGRAM MANAGER
EVALUATE REPORT		USER
DECIDE ON PROCUREMENT		USER
PROCUREMENT		PROGRAM MANAGER
INSTALLATION & TRAINING		PROGRAM MANAGER

COMPUTER SYSTEM HISTORY

1ST GENERATION 1946 - 1960

- VACUUM TUBES
- MACHINE LANGUAGE
- START OF ASSEMBLER DEVELOPMENT

2ND GENERATION 1960 - 1965

- TRANSISTORS & DISCRETE COMPONENTS
- MEMORY ACCESS TIME 10 µs
- START OF COMPILER & OPERATION SYSTEM DEVELOPMENT

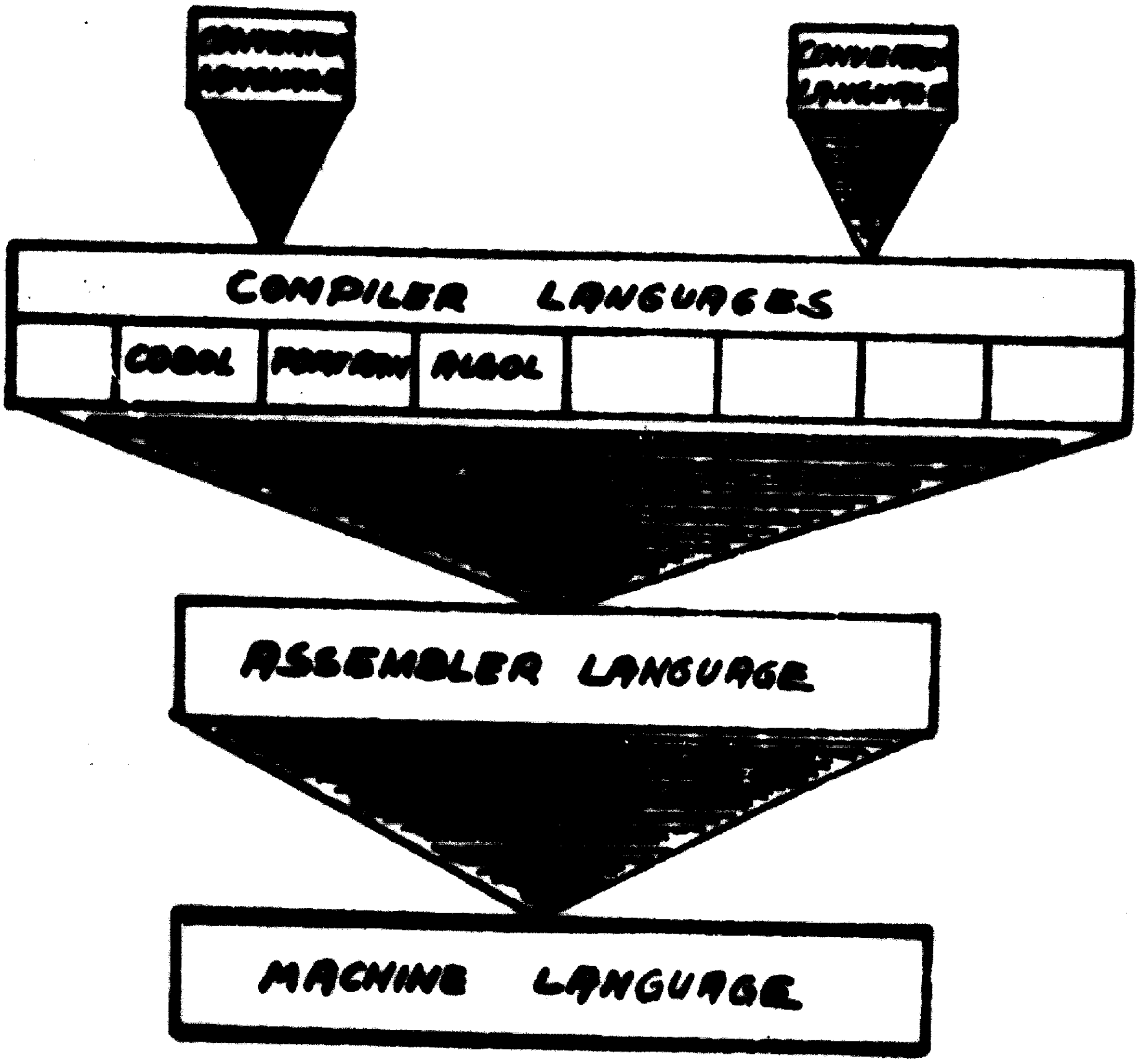
3RD GENERATION 1965 - 1968

- INTEGRATED CIRCUITS
- MEMORY ACCESS TIME 0,5 µs
- IMPROVED COMPILERS & OPERATION SYSTEMS
- MULTIPROGRAMMING, TIME SHARING, REMOTE DATA PROCESSING

MODIFIED 3RD GENERATION 1968

- MEDIUM & LARGE SCALE INTEGRATION
- MEMORY ACCESS TIME 0,2 µs
- IMPROVED SOFTWARE
- PARALLEL DATA PROCESSING

SOFTWARE DEVELOPMENT



ADVANTAGES OF A TYPICAL CONVERTER LANGUAGE

- REDUCED TRAINING FOR PROGRAMMERS

  20 DAYS REQUIRED FOR COBOL

  2 DAYS REQUIRED FOR THE CONVERTER LANGUAGE

- REDUCED PROGRAMMING EFFORT

  50 TO 95 % REDUCTION

- ALL COBOL ADVANTAGES SUCH AS

  STANDARDIZATION

  CLEAR PROGRAM STRUCTURE

  ETC.

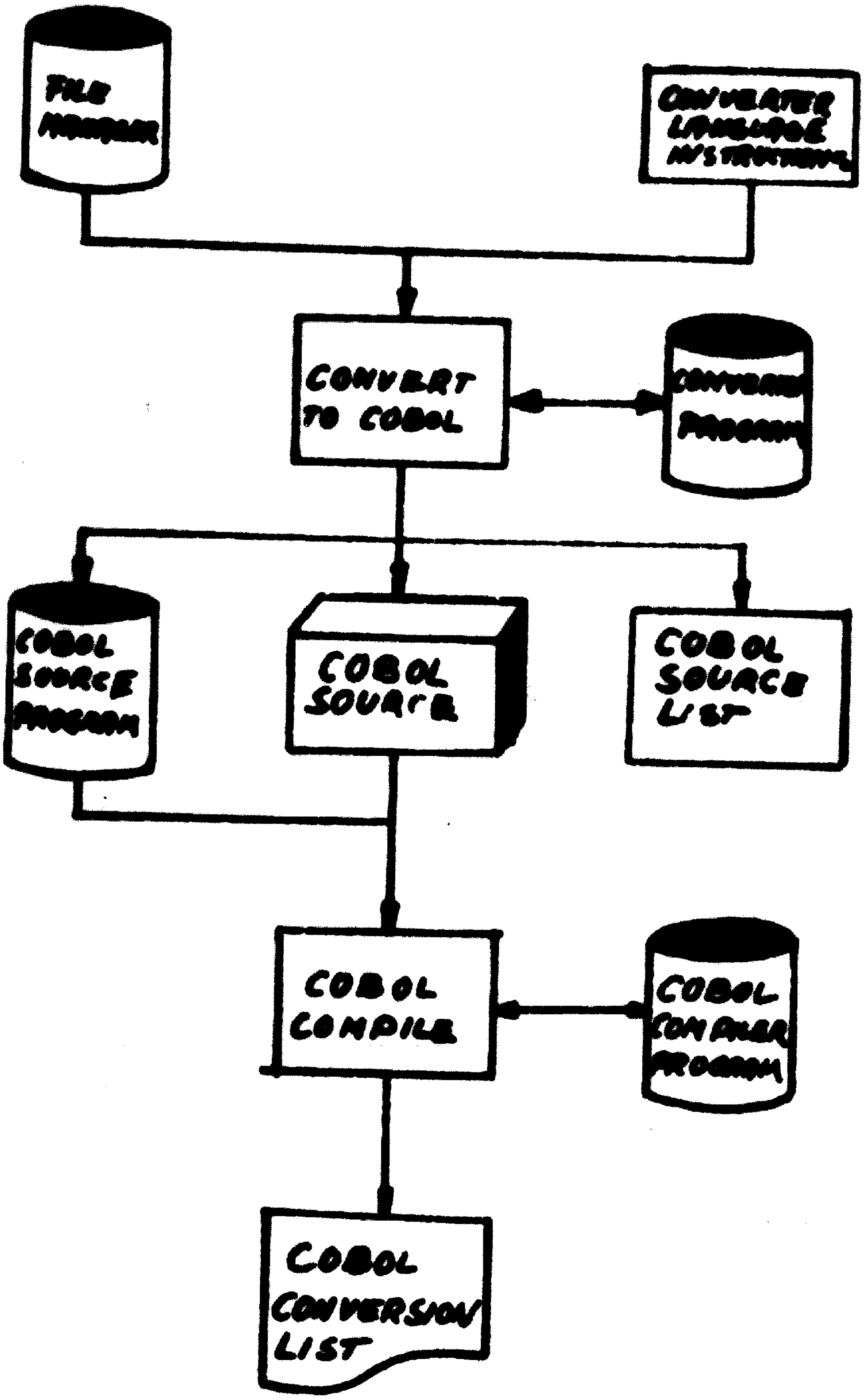
  ARE MAINTAINED

A TYPICAL CONVERTER PROGRAM :
" DATA RETRIEVAL IN A SPECIFIED LIST FORM "

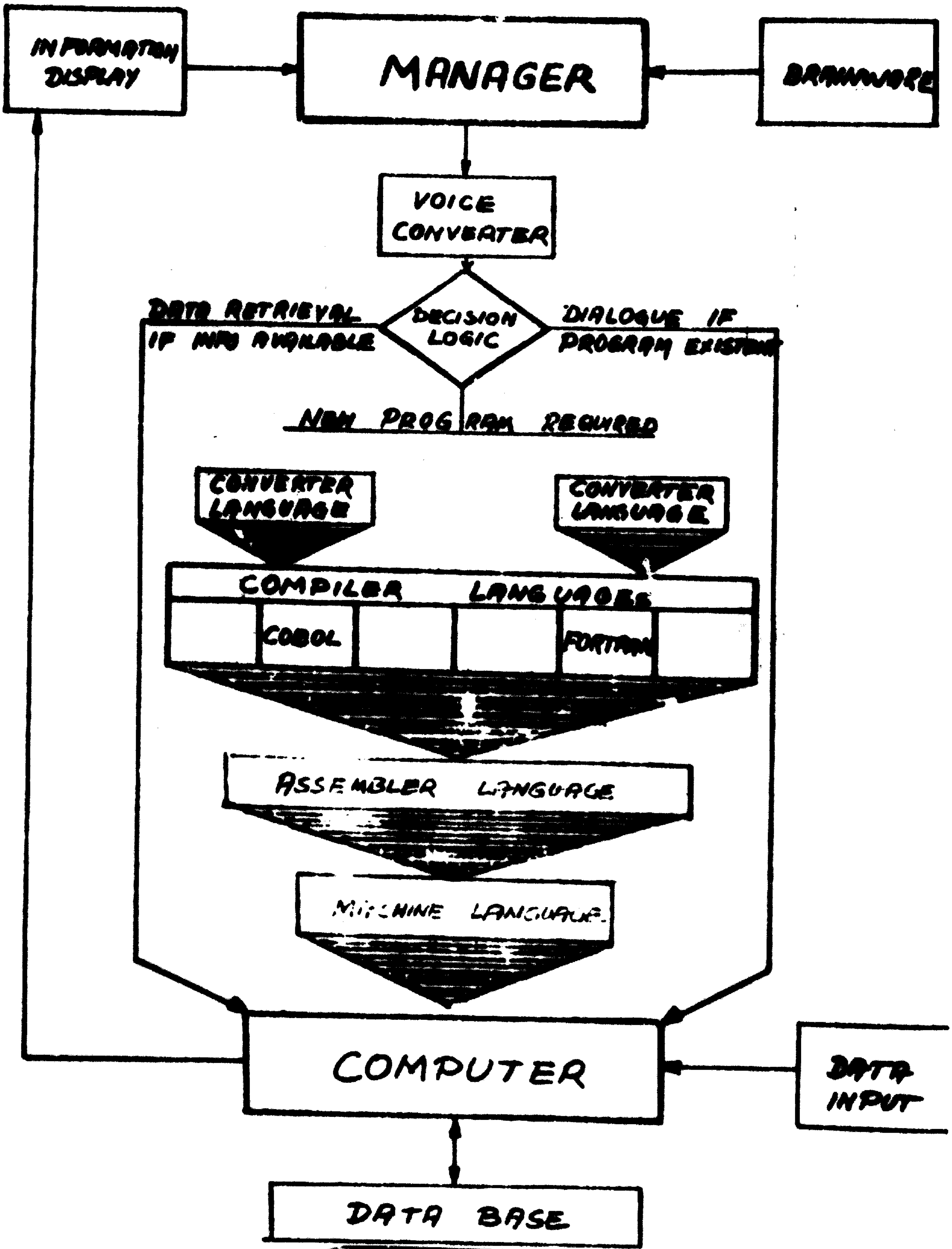
10 FRAMES OF A CONVERTER PROGRAM
GENERATE
200 FRAMES OF THE CORRESPONDING
COBOL PROGRAM

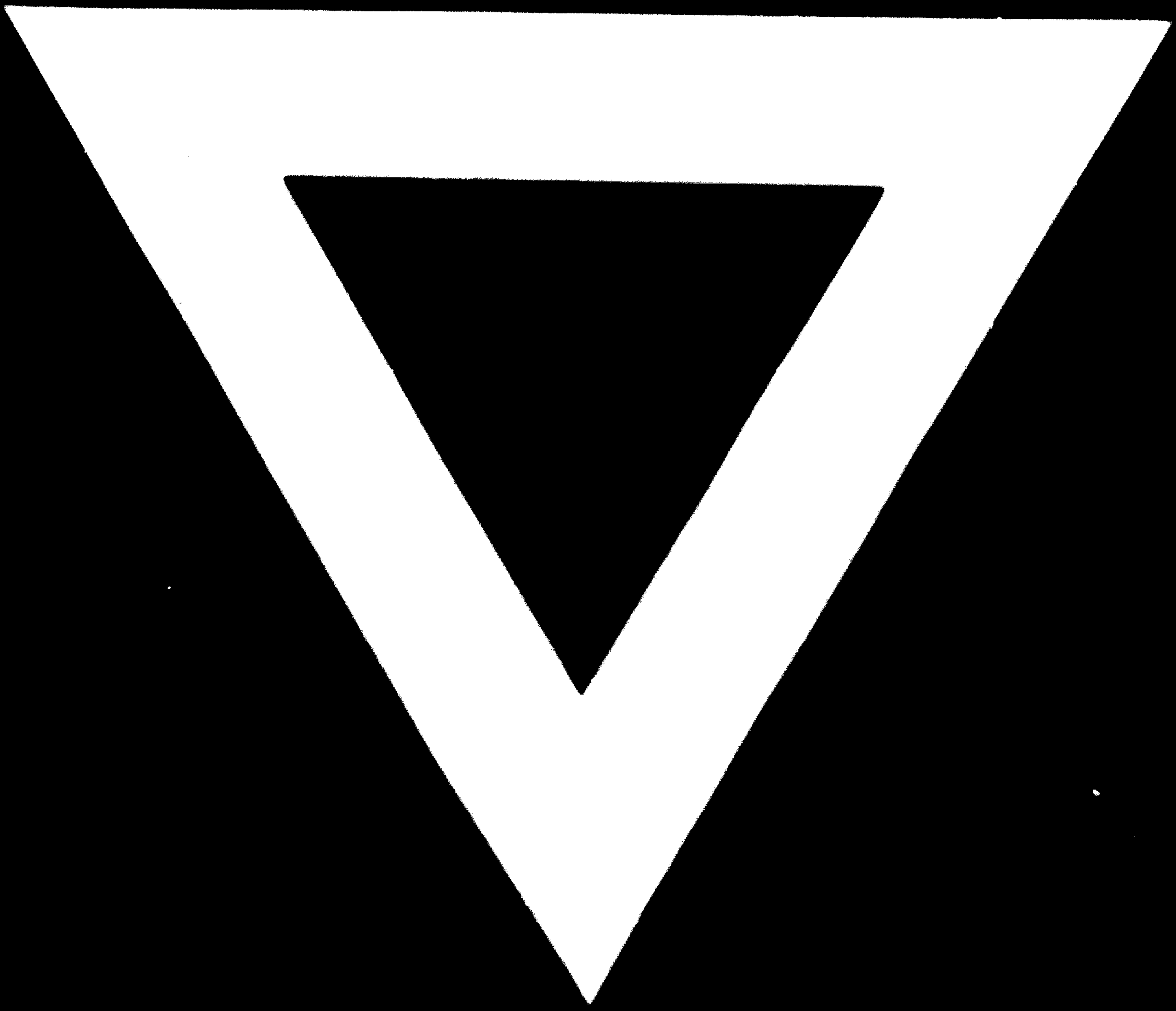
FRAME NO.	CONVERTER INSTRUCTIONS	TASK PARAMETERS
1		REQUESTED BY COMBYTE
2	USE	ARTICLE
3	[illegible] LIST	" ARTICLE MASTER FILE "
4		" STORAGE, GROUP, ARTICLE,
5		NAME, STOCK ON HAND, SALES "
6	[illegible]	STORAGE-NO, GROUP-NO, ARTICLE-NO,
7		ARTICLE-NAME, QTY-ON-STOCK, QTY-SOLD
8	SUM OF	QTY-ON-STOCK, QTY-SOLD,
9		GROUP-NO STORAGE-NO
10	END	& TOTAL

THE PRINCIPLE OF A CONVERTER LANGUAGE



MANAGER - COMPUTER DIALOGUE





2. 8. 74