1980

THE SYSTEMS DEVELOPMENT PROCESS: A DATA-THEORETIC APPROACH

Edward A. Stohr
New York University

THE SYSTEMS DEVELOPMENT PROCESS: A DATA-THEORETIC APPROACH

EDWARD A. STOHR

Graduate School of Business
New York University

ABSTRACT

This paper describes the logical design of a data base to support the software development process by analyzing the information content of common systems analysis techniques such as IBM's Business Systems Planning, structured analysis, structured design, and data base design. It is shown that these techniques can be represented in a single data base schema. The data base can be extended to allow for project control.

1. INTRODUCTION

The development of information systems that are truly responsive to user needs has proved to be a deceptively complex, error-prone, and expensive process. Much research has been directed to improving the process over the last 20 years and many different approaches and techniques have been tried. This paper provides a framework for systems development by proposing a conceptual data base model or schema of the information needed to specify a software project through several stages of its development. The model provides a basis for comparing and integrating different systems analysis techniques. Given the conceptual schema in network form, the software development process can be viewed from a data theoretic point of view as developing an instance of the database through iterative traversals of the network.

The schema is developed by integrating the data models underlying three well-known formal systems analysis techniques (1) IBM's Business Systems Plan (BSP) (12), (2) Structured Analysis (DeMarco (9), and Gane and Sarson (10)), and Structured Design (Myers (19), and Yourdon and Constantine (28)). With some exceptions the combined information requirements for these three techniques span those for many other systems analysis techniques such as ADS (20), the ISDOS group's PSL/PSA (23), Systematics (11), HIPO (13), and the ISAC approach (18).

The model covers the following stages of the 'structured life cycle' (DeMarco (9)): (1) Feasibility Study, (2) Structured Analysis, (3) Structured Design, (4) Coding, and (5) Testing. However, the coding and testing phases are not discussed in detail. A final section of the paper extends the data base model for the purpose of project control.

A data model of the systems development process can be useful in several different contexts. First, it can be used to design a coherent set of documentation standards and to specify alternative possible sequences for their development over time. The actual sequence in which the various pieces of information should be gathered and documented will depend on the particular project deadlines, scope, manpower allocation, etc. and the strategy chosen for the development process. Possible strategies in this sense range from relatively strict adherence to the stages of the development life cycle, with some looping back to earlier stages to prototyping, where the latter strategy is most useful in novel and unstructured situations; e.g., Decision Support Systems (Keen and Scott-Morton (14)).

Secondly, the schema can be implemented using any conventional Data Base Management System (DBMS). In this way it can serve as an extended Data Dictionary containing a repository of information about the systems being developed. The ultimate objective is to have the data base maintain the complete system specification in machine readable form using text-processing and interactive graphics for input and output. However, the viewpoint adopted in the illustrations in this paper is that much of the documentation will remain in hard-copy form in project diaries, designer's note books, etc. The database will contain key summary information and act as an index to the remaining documentation. Thus a mixture of manual and automated techniques is possible. The resulting system can be used to aid communication between analysts, to perform data transformations and analysis useful in the design process, for report generation, and for project management and control.

Finally, the schema can be used in a comparative analysis of the different systems analysis and design techniques in a manner similar to that in Couger (7). In this context, mapping the information requirements and outputs of the various techniques onto the conceptual schema shows their points of similarity and difference. As might be expected, each technique applies to different stages of the life cycle and emphasizes different aspects of the total information base, as well as different tabular and graphical conventions for the display and analysis of significant relationships.

2. FOUNDATIONS

A simple model of an organization due to Leavitt (16) will be used as a stepping stone for determining the scope of information requirements analysis. Leavitt pictured an organization as being composed of four interacting components: (1) the formal organization structure, (2) the people involved in the organization, (3) the tasks performed, and (4) the production technology employed. Figure 1 shows a modified version of this model which includes the environment and in which the organization's technology has been split into two components: production technology and information systems (IS). The arcs connecting the components represent interactions. In general, a change in the environment or a change in one or more of the other components. Thus an adequate system development methodology should take into account all six items shown in the figure.

However, this is a rather mechanistic overview of information analysis. The paradigm can be greatly improved by more in depth study of the structure, task, and people components. According to Churchman (4), any analysis of a system must first identify a number of role players and derive a set of system objectives. Namely, the systems analyst should identify: (1) Clients -- persons or groups who are to be served by the system, (2) Decision-makers -- persons or groups who can control the evolution of the system and whose goals may or may not be congruent with those of the clients, (3) System objectives -- formal goals which take into account (1) and (2) and to attain which the organization is willing to expend resources, (4) Performance criteria -- observable or computable measures of how well the system meets its objectives, and (5) System resources. Only by a continuing emphasis on these elements can the systems development effort avoid the common pitfall of building a technically correct system which serves the wrong purpose.

In this regard, several approaches have been advocated. King (15) develops the relationships between organization strategies and IS objectives, constraints, and design strategies. For executive information systems, Rockart (22) describes the development of "critical success factors" (CSF's). These are "key areas of the business in which high performance is necessary if the goals are to be met". The systems development effort is guided towards measuring and reporting alternative measures of the CFS's rather than towards producing voluminous reports summarizing operations. Lucas (17) urges active user involvement in the design process so that the clients and decision-makers, together with the systems analysts, can thoroughly explore alternatives and obtain a true picture of system requirements.

3. THE INFORMATION CONTENT OF SOME COMMON SYSTEMS ANALYSIS AND DESIGN TECHNIQUES

To indicate how these general information requirements are currently translated into system documentation, the information content of some well-known systems analysis techniques is now briefly described.

BUSINESS SYSTEMS PLANNING

All the formal (step-by-step) methodologies for systems analysis take users and their decision making requirements into account to some degree. Of these, BSP seems to concentrate more of its efforts on managing the collection, analysis, consolidation, and presentation of this less tangible data. BSP is used to produce a long-range plan for IS development. The approach is summarized graphically in Figure 2. Its major objectives are: (1) To decompose the organization's IS into a set of internally coherent, minimally connected sub-systems. Each subsystem consists of computer programs and data base elements. The resulting "Information Architecture" is depicted by means of a highly aggregated "data flow diagram" (Figure 2c); and, (2) To develop a set of priorities for IS development projects together with cost-benefit analyses and a time schedule or Gantt chart for development (Figure 2d).

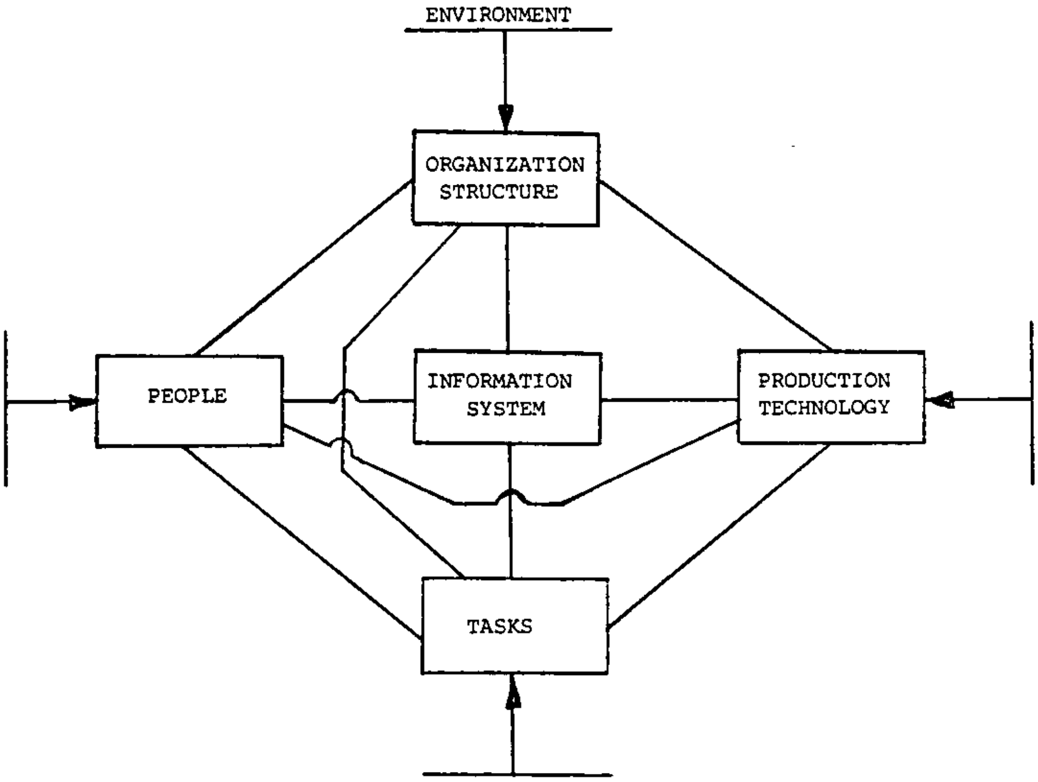
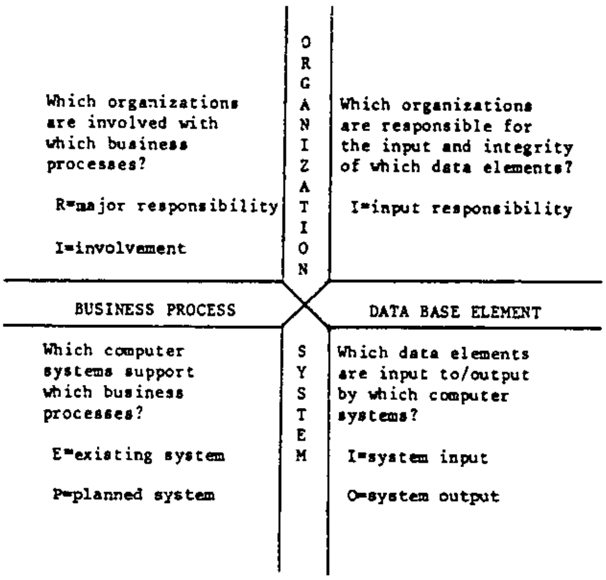


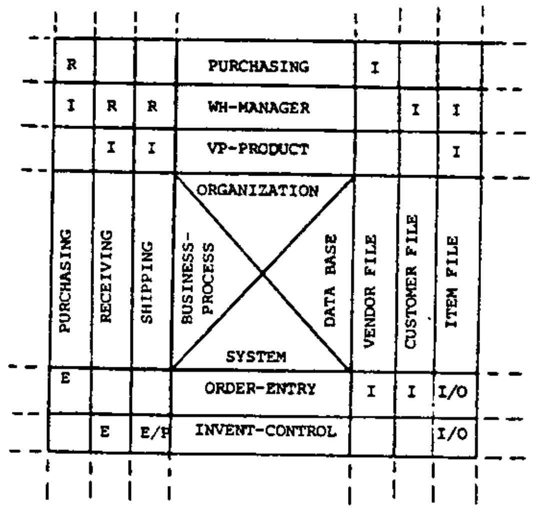
FIGURE 1

SIMPLE MODEL OF AN ORGANIZATION

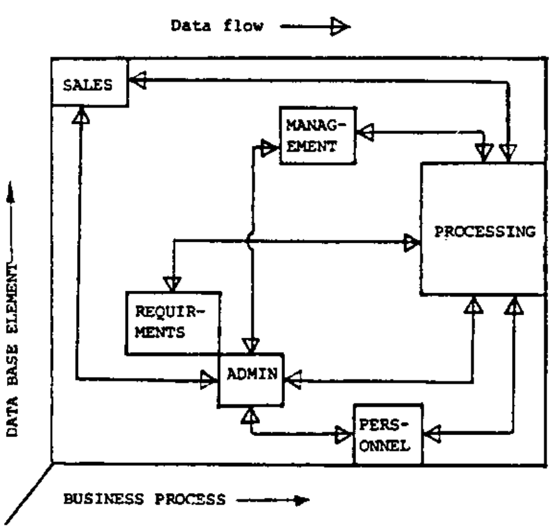
(a) INFORMATION CROSS



(b) INFORMATION CROSS EXAMPLE



(c) INFORMATION ARCHITECTURE



(d) OTHER COMPONENTS OF BSP

DATA GATHERING:

BUSINESS ENVIRONMENT
BUSINESS OBJECTIVE, PROBLEMS, CONSTRAINTS
FINANCIAL STATISTICS
ORGANIZATION CHARTS
PRODUCTS AND MARKETS
MANAGEMENT PLANNING & CONTROL PROCESSES
PRODUCT/SERVICE FLOWCHARTS
SUPPORTING RESOURCE FLOWCHARTS
ETC.

OUTPUTS:

REVIEW OF IS DEPARTMENT
COST BENEFIT STUDIES
INFORMATION ARCHITECTURE
PROJECT PRIORITIES
IS DEVELOPMENT PLAN
(GANTT OR PERT CHARTS)

FIGURE 2

CONCEPTS OF BSP

Details of both the current business and information system and the planned system are summarized and condensed by representing the existence of important relationships in a series of matrices. The most important of these are shown in Figure 2a. Comparing this with Figure 1, it can be seen that BSP recognizes the people, organization structure, and task and information systems components with the latter being split into the two sub-components of data and (computer) systems. The technology component is recognized separately via the development of Product/Service and Resource Flow charts. An appreciation of environmental factors is gained through management interviews. The "Business processes" (task component) of BSP are logically related sets of decisions and activities which are important to the functioning of the organization.

To anticipate future developments in this paper, note that each of the four matrices in Figure 2b represents a many-to-many relationship between the two "entities" on the axes. Thus, the WH Manager (an organization entity) is involved in both Receiving and Shipping (business processes entities). Conversely, the Shipping process involves both the WH Manager and the VP Production. BSP also records the nature of the "intersection data" e.g. the WH Manager has the major responsibility for Receiving, while the VP Production is "involved." We will greatly expand on these concepts in Section 3.

The information architecture shown in Figure 2c represents a partition of the business processes and a matching partition of the IS subsystems serving those tasks. The partition of data base elements does not exactly match the partition of subsystems. The data flows between the subsystems are necessary to compensate for this mismatch.

STRUCTURED ANALYSIS

While BSP is concerned with a long-range IS plan consisting of many projects, Structured Analysis can be used to develop the detailed logical specifications for a given project. The major outputs of structured analysis are shown in Figure 3. Based on a life cycle approach, the major emphasis in Structured Analysis is on the development of: (1) a comprehensive set of Data Flow Diagrams (DFD's), (2) a data dictionary, (3) a set of process specifications, and (4) a logical data base design.

DFD's (Figure 3a) are developed for both the existing and planned systems. They describe data flows (messages), files and processes within both the manual and automated parts of the system. The arcs (data flows) depict the movement of data on both electronic and hard-copy media. The bubbles (processes) depict manual or automated transformations performed on the data. The square boxes (external entities) are also called "sources and sinks." These represent organizational units (or other computer systems) which lie outside of the information system per se, in the sense that they are net receivers and transmitters of data.

The data dictionary contains definitions of all data structures, groups, and elementary items in the data flows and data files. The process specifications are expressed using such techniques as Structured English, and (where appropriate) decision tables. The data base is designed as a set of "normalized" relations (Codd (6)) to reduce redundancy and increase the precision with which multiple interrecord relationships are depicted. Data structure diagrams, in Figure 3b (Bachman (2)) and a "Census of Data Accesses" are developed to identify and document data access paths, their frequency of use, and the response time requirements. Other subsidiary information is developed as part of the approach as shown in Figure 3c.

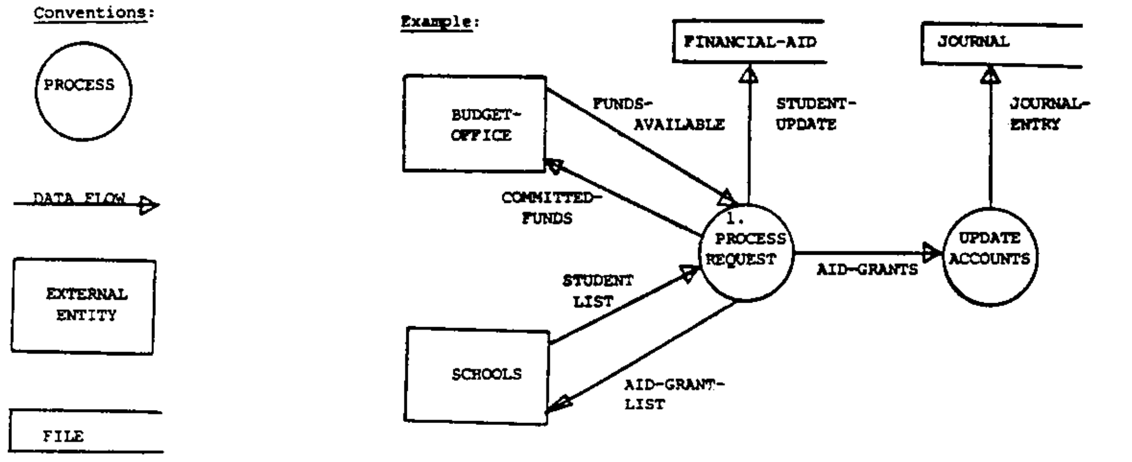
OTHER SYSTEMS ANALYSIS TECHNIQUES

The information content of the ISDOS approach (Figure 4) and Structured Analysis are essentially the same although the former is fully automated while the latter is manual. In ISDOS the definitions of data flows, files, processes and "real world entities" are input directly to the computer in increasingly fine detail using a top-down approach: logical consistency is checked and physical file design is attempted (Nunamaker, (21)). The ISAC approach to systems analysis is also informationally similar to Structured Analysis. The technology component is given more emphasis however; diagrams showing major material and product flows are developed prior to an abstraction of the information requirements in DFD's (Lundberg, (18)).

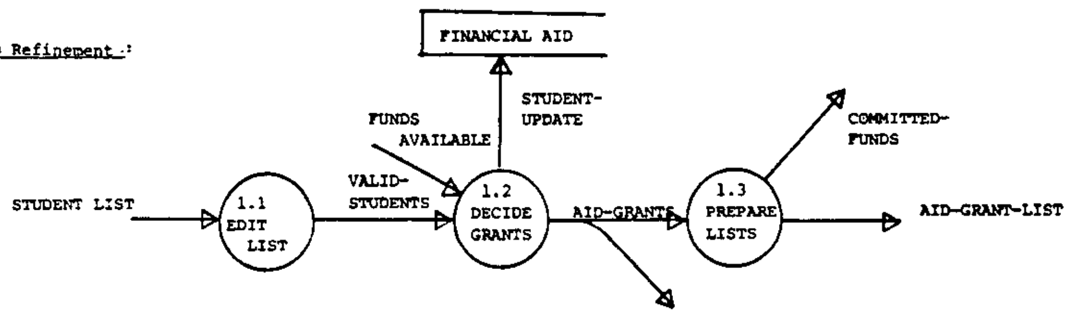
STRUCTURED DESIGN AND CODING

Techniques such as HIPO and Structured Design are most useful in the software design phase of the system's life cycle. Both are concerned with the structure or architecture of software in terms of the modules into which programs are decomposed, the flows of data into and out of the modules, and the form of their interconnection. Structured Design is

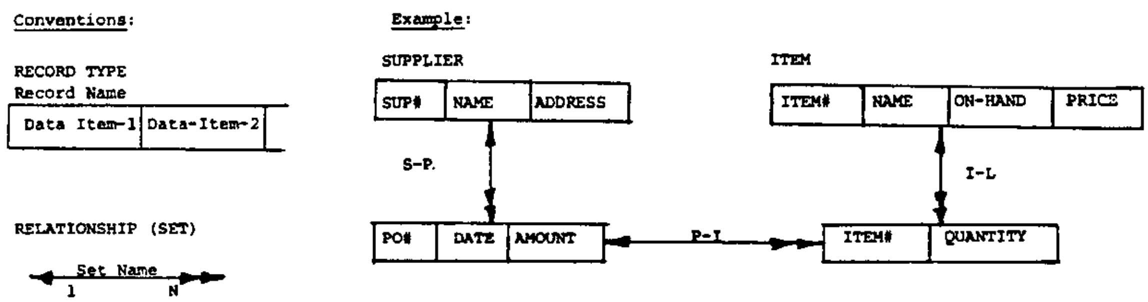
(a) DATA FLOW DIAGRAM



Top-Down Refinement:



(b) DATA STRUCTURE DIAGRAM



(c) COMPONENTS OF STRUCTURED ANALYSIS

SYSTEM ENVIRONMENT, PROBLEMS, CONSTRAINTS
SYSTEM OBJECTIVES
DATA FLOW DIAGRAMS
PROCESS LOGIC SPECIFICATIONS
DATA DICTIONARY

DATA BASE DESIGN:
- NORMALIZED RELATIONS
- DATA STRUCTURE DIAGRAMS
- DATA ACCESS & RESPONSE TIME REQUIREMENTS

FIGURE 3

CONCEPTS OF STRUCTURED ANALYSIS

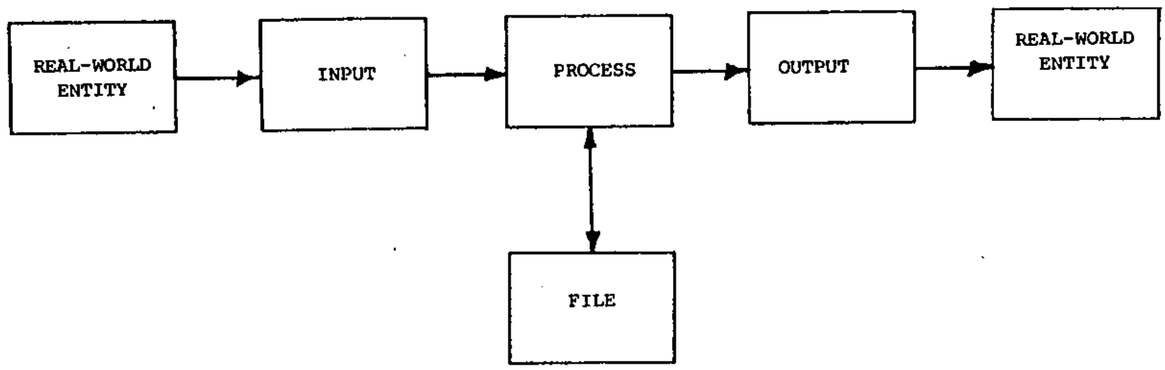


FIGURE 4

THE ISDOS APPROACH

noteworthy in its attempt to map directly from the DFD of a program to its realization as a hierarchy of modules or "Program Structure Chart" as shown in Figure 5a.

The coding phase of the system life cycle has received the most attention in both the literature and practice of software development. Perhaps this is because the code represents the final "product" of the systems development process and because it contains all the errors of omission and commission of the previous stages. Some documentation requirements are listed in Figures 5b and 5c. Fiqure 5d illustrates a "System Flow Chart." Advocates of top-down design, functional decomposition," and structured programming (Dahl, et al. (8)) assert that these techniques, together with internal program documentation and clarity of style, have reduced the need to produce and maintain program flow charts. Some external documentation of logically complex sections of the code in the form of program narratives, pseudocode, or decision tables may still be desirable however.

4. A DATA BASE FOR SYSTEMS ANALYSIS

The logical structure of a data base for recording the data collected and generated during systems analysis will now be described. Two "conceptual schemas" (or logical data models (ANSI-SPARC (1)) representing the different world views of BSP and Structured Analysis and Design will be developed separately and then combined. The result is a fairly comprehensive model of the "real world" objects and relationships which systems analysts seek to describe and quantify.

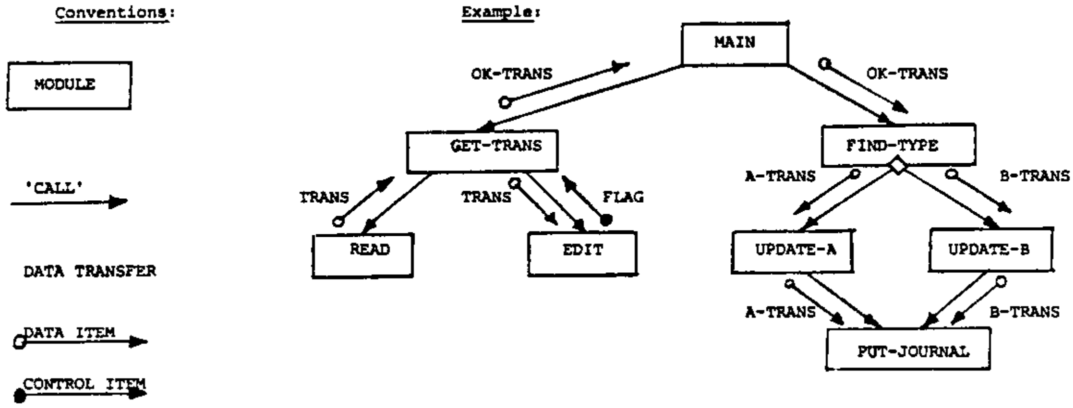
Although BSP is traditionally used for long-range IS planning, it collects much data which is useful for the development of specific projects. The information structure (entity and relationship types) remains valid. It is however, necessary to move to another level of resolution in terms of the detail of the information technology component. The contribution of BSP is that it is much more explicit than the other systems analysis techniques with regard to the people, task, and decision-making components of Figure 1.

It must be emphasized here that the data recorded in the data base model is subject to two further manipulations:

1. The data must be rearranged, edited and refined to produce the final products of the system analysis as shown in Figure 5c. Only one of these final documents (the Structured Specification) is represented fairly directly.
2. Additional entities and relationships must be added to the schema if it is to be integrated into a project control system.

The graphic conventions which will be used to display the conceptual schema are those of the Entity-Relationship (E-R) model, (Chen (3)) and are summarized in Figure 6. Briefly, entity types (boxes)

(a) PROGRAM STRUCTURE CHART



(b) PROGRAM DOCUMENTATION

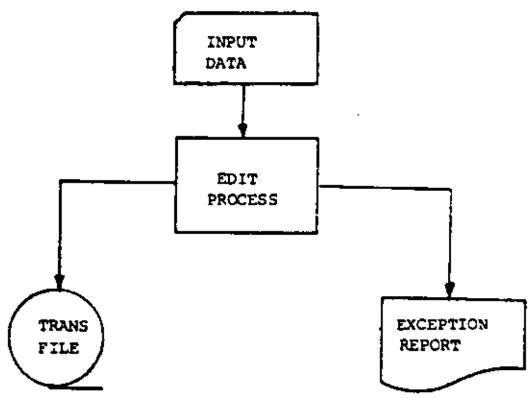
ENVIRONMENT, OBJECTIVES AND CONSTRAINTS
PROGRAM DESCRIPTION
LANGUAGE AND OPERATING ENVIRONMENT
SYSTEM FLOWCHART
PROGRAM USE STATISTICS, RUN-TIME
AND CORE REQUIREMENTS
OPERATING INSTRUCTIONS AND JCL
INPUTS AND OUTPUTS
- FILE LAYOUT DIAGRAMS
- OUTPUT REPORT LAYOUTS
- SCREEN LAYOUTS

PROGRAM STRUCTURE (OR HIERARCHY) CHARTS
MODULE DESCRIPTIONS
- INPUTS & OUTPUTS
- NARRATIVE/PSEUDO-CODE/DECISION TABLE/ FLOWCHART

PROGRAM TEST PROCEDURES
- TEST NO., DESCRIPTION, INPUT DATA, EXPECTED OUTPUT
- TEST FILES

RECOVERY PROCEDURES
MAINTENANCE INSTRUCTIONS
ERROR MESSAGES
GLOSSARY

(d) SYSTEM FLOW CHART



(c) FINAL SYSTEM DOCUMENTATION

USER MANUAL
STRUCTURED SPECIFICATION
PROGRAMMER MANUAL
PROCEDURES AND OPERATING INSTRUCTIONS
MAINTENANCE MANUAL
SYSTEM TEST MANUAL
PROJECT NOTEBOOK

FIGURE 5

CONCEPTS OF STRUCTURED DESIGN
AND
PROGRAM AND SYSTEM DOCUMENTATION REQUIREMENTS

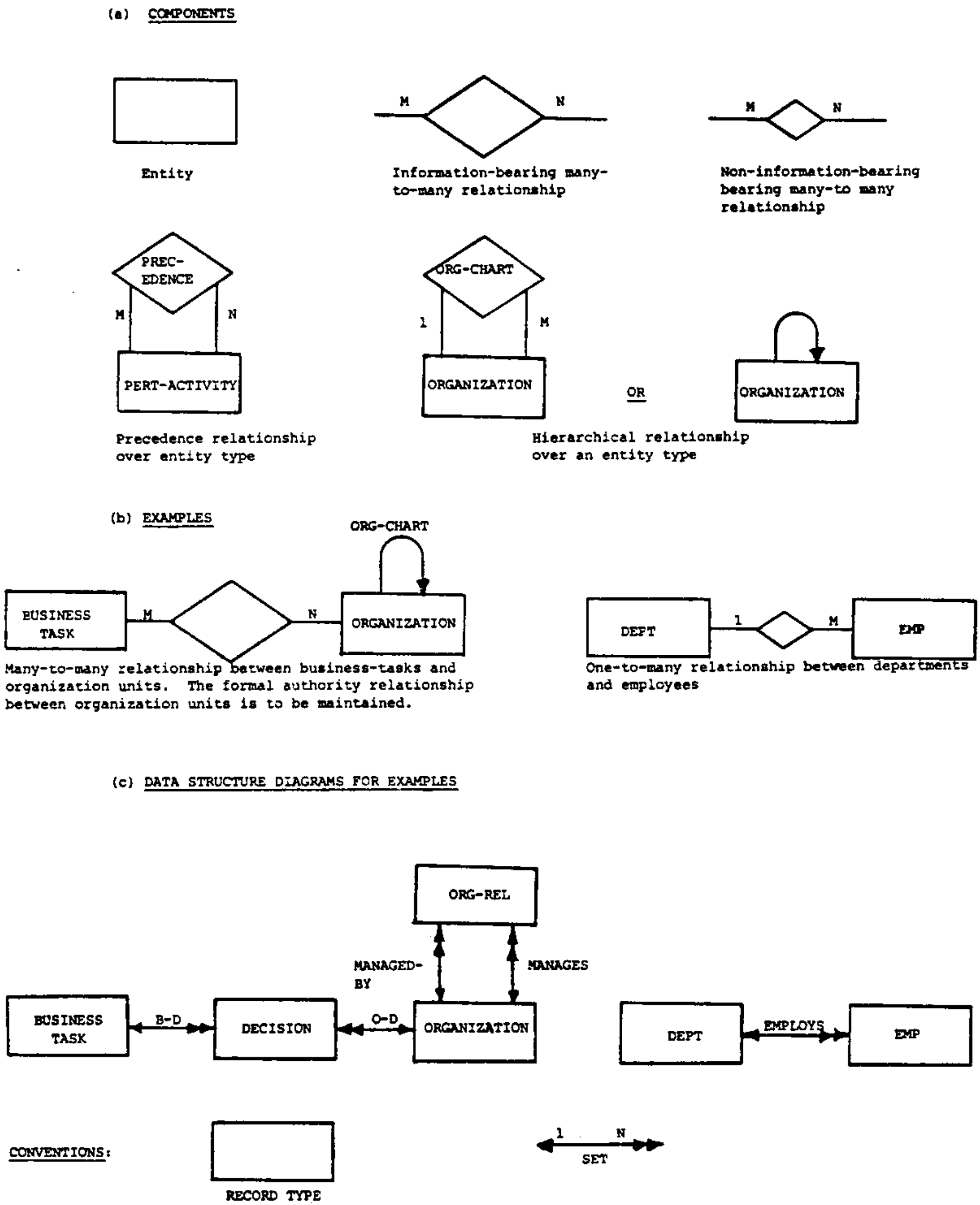


FIGURE 6

ENTITY-RELATIONSHIP DIAGRAMS

represent classes of objects which can be described by a common set of attributes. The entity type, Organization, for example might be described by the attributes, Name and Budget.. Particular instances of the entity type, Organization, might be Purchasing, Sales, and Production.

The relationships (diamond and arcs) are links between two or more entity types and represent the information that instances of these entity types are associated by some event or relation of interest. Thus in Figure 6b the entity types Organization and Business-Task are connected by the relationship, Decision, which records the information that certain entities of the type Organization have decision-making powers with regard to certain instances of the entity type Business-Task. Relationships can be one-as-to-one (1:1), one-as-to-many (1:m), or many-to-many (m:m). They relate two or more entities of different types or entities of the same type. In the latter case the relationship can be (1) hierarchical (1:n) or (2) network (m:n). An example of a hierarchical relationship is that relating organizational entities in an organization chart. An example of a network organization is that relating activities in a precedence network such as a PERT Chart.

The conceptual schema shows the logical information to be maintined independent of the particular DBMS to be employed. Further refinement is required to develop the actual schema for the DBMS and the physical aspects of storage (see Teory and Fry (24) for a description of the data base design task).

As a first approximation, entity instances can be thought of as records in the physical data base. A relationship may be information-bearing, in which case there will be a corresponding record type in the physical data base containing intersection data. If the relationship is non-information-bearing, it may be represented (in a network or hierarchical DBMS) merely by cross-referencing pointers in the records corresponding to the related entity types. In a relational DBMS, both entities and relationships are represented as "relations" (tables or "flat" files). The relationship relations contain the primary keys or identifiers of each related entity relation. Thus the association between entities is recorded by means of key values.

To save space in the diagrams: (1) non-information-bearing relationships will be represented by smaller diamonds than information-bearing relationships and (2) hierarchical relationships defined over the same entity type will be represented by loops without the diamond shape.

The hierarchical and network relationships to be described below can often be presented more clearly to human beings using tree-structures, network diagrams, and matrix-oriented techniques. E-R diagrams (and their close cousins Data Structure Diagrams) show these relationships in a more highly condensed and abstract form. However, it will be important in the sequel to use a common representation for all relationships. Another great advantage of using the Entity-Relationship approach is that it displays only logical relationships and can be translated quite readily into a specification for any of the three major DBMS models -- hierarchical, network, and relational, (Chen (3)). For example, the translation from an E-R diagram to a CODASYL type data struture diagram is straight forward as illustrated in Figure 6c.

5. CONCEPTUAL SCHEMA FOR LONG-RANGE IS PLANNING

The conceptual schema underlying the BSP approach is shown in Figure 7-1. Note that the BSP Business-Processing defined earlier are now called BUS-TASKS to distinguish them from the "processes" that will be introduced later. The entities and relaionships shown in dark outline have the same structure (informationally) as the information cross in Figure 2. However, the latter displays the value of only one attribute for each entity and relationship. In the data base approach, it is possible to record many attribute values. Some possibilities are shown in the Figure 7-2. These are meant to be suggestive only; many alternatives are possible depending on the implementation objectives.

The data recorded by the entities is straight forward. The information recorded by the relaionships in the BSP approach is now described.

DECISION is a many-to-many relationship recording the fact that a number of users may be associated with a given business task and a given user may be associated with a number of business tasks. As in BSP, the RESPONSIBILITY-CODE attribute records the degree of decision-making authority the organization unit has in the execution of the task. This relationship has been called "DECISION" however, to emphasize the necessity of the systems analyst to understand the user's goals and decision-making processes. Information about these is recorded during an interview and the REFERENCE field in the record gives the logical location of the paper-work within the project library.

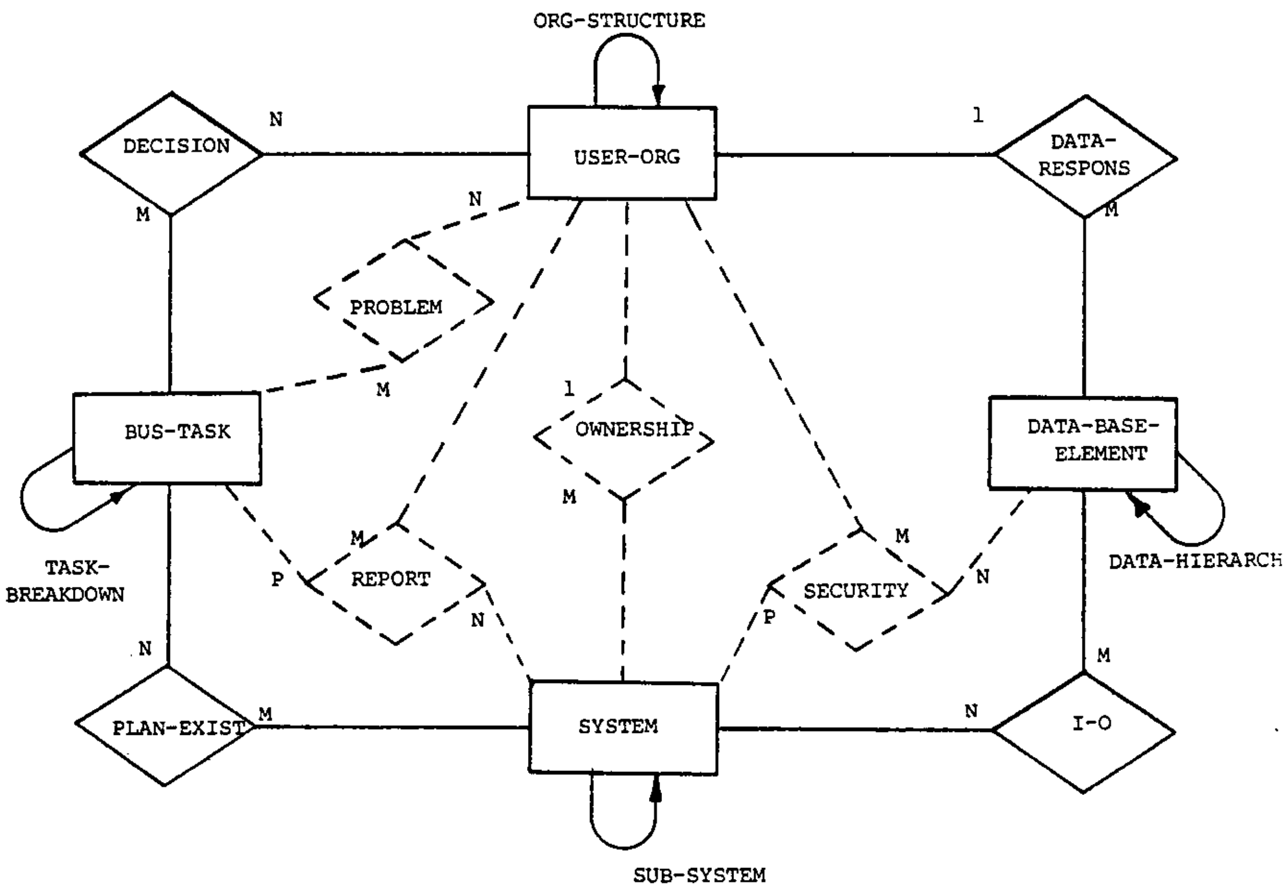


FIGURE 7-1

CONCEPTUAL SCHEMA FOR LONG-RANGE IS PLANNING

POSSIBLE ENTITY ATTRIBUTES	EXTERNAL DOCUMENT REFERENCES
USER-ORG (O-NAME, MANAGER, LIASON, IS-BUDGET, STCR)	Organization-Chart, Budget
BUS-TASK (B-NAME, SHORT-DESCR, PERFORMANCE-MEASURE, GOAL, STCR)	Interview Notes
SYSTEM (S-NAME, SHORT-DESCR, S-TYPE, S-SIZE, PRIORITY, STCR)	System Documentation
DATA-BASE-ELEMENT (D-NAME, D-TYPE, DEFINITION, D-SIZE, D-VOL, FILE-NAME, STCR)	File Layout Chart, Data Dictionary

POSSIBLE RELATIONSHIP ATTRIBUTES	
DECISION (INTERVIEWER, INTERVIEWEE, INTERVIEW-DATE, RESPONSIBILITY-CODE, SHORT-DESCR, STCR)	Interview Notes, Decision Table
PLAN-EXIST (P-OR-E, IMPLEMENTATION-DATE, SHORT-DESCR, STCR)	Project Notes
I-O (I-OR-O, PICTURE, FILE-NAME, FREQUENCY, RESPONSE-TIME, STCR)	File Lay-out Chart
DATA-RESPONSIBILITY (INPUT-MODE, PRIORITY, TIMING-REQUIREMENTS, DATA-VOL, STCR)	Systems and Procedures Manual
PROBLEM (INTERVIEWER, INTERVIEWEE, INTERVIEW-DATE, PROBLEM-STATEMENT-TEXT, STCR)	Interview Notes
REPORT (R-NAME, R-TYPE, SHORT-DESCR, RESPONSE-TIME, FREQUENCY, WHEN-REQUIRED, NO-OF-COPIES, STCR)	Report Layout Chart, Interview Notes
OWNERSHIP (USER-PRIORITY, DEVELOPMENT-BUDGET, IMPLEMENT-DATE, STCR)	Interview Notes, Data Base Manual
SECURITY (LEVEL-CODE, AUTHORIZING-OFFICER, STCR)	Interview Notes, Data Base Manual

NOTE: STCR Stands for four separate fields
SIGNATURE = signature code for person entering data
TIME = planned or actual completion date of related work
COMMENT = special comment field
REFERENCE = reference code to external documentation

FIGURE 7-2

CONCEPTUAL SCHEMA FOR LONG-RANGE IS PLANNING

As in BSP, the PLAN-EXIST relationship records the existence of computer systems supporting the business tasks and/or plans to provide such assistance in the future. The I-O relationship records the data inputs and outputs of each computer-system and the derivation and uses of each element of data. DATA-RESPONSIBILITY is a one-to-many relationship that records the user organization responsible for the input of each data item. The attributes of this relationship record statistics relating to the magnitude of the data entry task. The REFERENCE field refers to related manuals of operating instructions.

The four hierarchical relationships ORG-CHART, TASK-BREAK-DOWN, SUB-SYSTEM and DATA-HIERARCHY record the natural or imposed hierarchies associated with each entity type. Hierarchies such as these provide a basis for the management of complexity by allowing a top-down approach to design in which successively higher levels of detail are examined and analyzed. The highest level of aggregation in the data hierarchy might be a file or data base; the lowest an elementary data item. The highest level of aggregation in the program hierarchy might be a "system" of related programs; the lowest a module or sub-routine.

The relationships shown in dashed outline are implicit rather than explicit in the BSP approach. They assist in some essential systems analysis tasks. PROBLEM provides a means for recording, ranking, and listing the problems specific users associate with different business tasks. REPORT links users, business tasks, and computer programs; it can be used to generate a report distribution table. OWNERSHIP records the principle user organization served by the computer system together with the system development budget and contractual information if required. SECURITY links users, data base elements, and programs; it can be used to produce and "access matrix" of security codes for the data base administrator (Wood, et al. (27)).

Other relationships could be added if required. Assuming that a physical implementation will allow direct or indexed entry to each of the four entity records, it is clear that a combination of special purpose host language retrieval and update routines, together with a query language and report writing capability in the DBMS, could serve many useful management functions:

1. Project Library Maintenance

The database can serve as a directory or table of contents to all information gathered by project members. Some of this information is stored in the computer and some (for example, interview notes and existing software documentation) in hard-copy form in file folders and filing cabinets. As project members complete assignments, they inform the DBMS; data base records are added or updated and reference locations are assigned for hard-copy material. Obviously these functions are also useful during the maintenance phase of the system.

2. Project Management

The data base contains a record of all work performed to date by each project member, the SIGNATURE field. The ability to ascertain current status is useful in scheduling future work. Extensions to the data base schema to make it more useful in this regard will be mentioned later.

3. Analysis

The human analysis of any complex system can be greatly aided by mechanisms which can array, sort, and condense data in many different ways. For example, the data base can easily be used as a basis for automation of all the matrix-oriented computations in the BSP approach. The analytic capabilities of the computer can also be used to check the logical consistency of the design as in Nunamaker (21).

4. Report Generation

The requirements for project reporting, added to those for system documentation, are a heavy burden which could be considerably reduced by report generation and graphics programs acting on the data stored in the data base.

6. CONCEPTUAL SCHEMA FOR STRUCTURED ANALYSIS AND DESIGN

Structured Analysis is designed to accept the requirements and problem statement of a prior feasibility study, to expand upon these through more detailed analyses, and to develop the overall specifications for the manual and automated aspects of a new system. Its major outputs (see earlier) are: (1) a set of data flow diagrams (DFD's), (2) a data dictionary, (3) a set of process specifications, and (4) a logical data base design. These are used in the Structured Design phase to develop a detailed software specification for coding. Together these documents constitute the "Structured Specification" of DeMarco (9). A schema representing the world view of these two phases is shown in Figure 8-1. Suggested data items for the

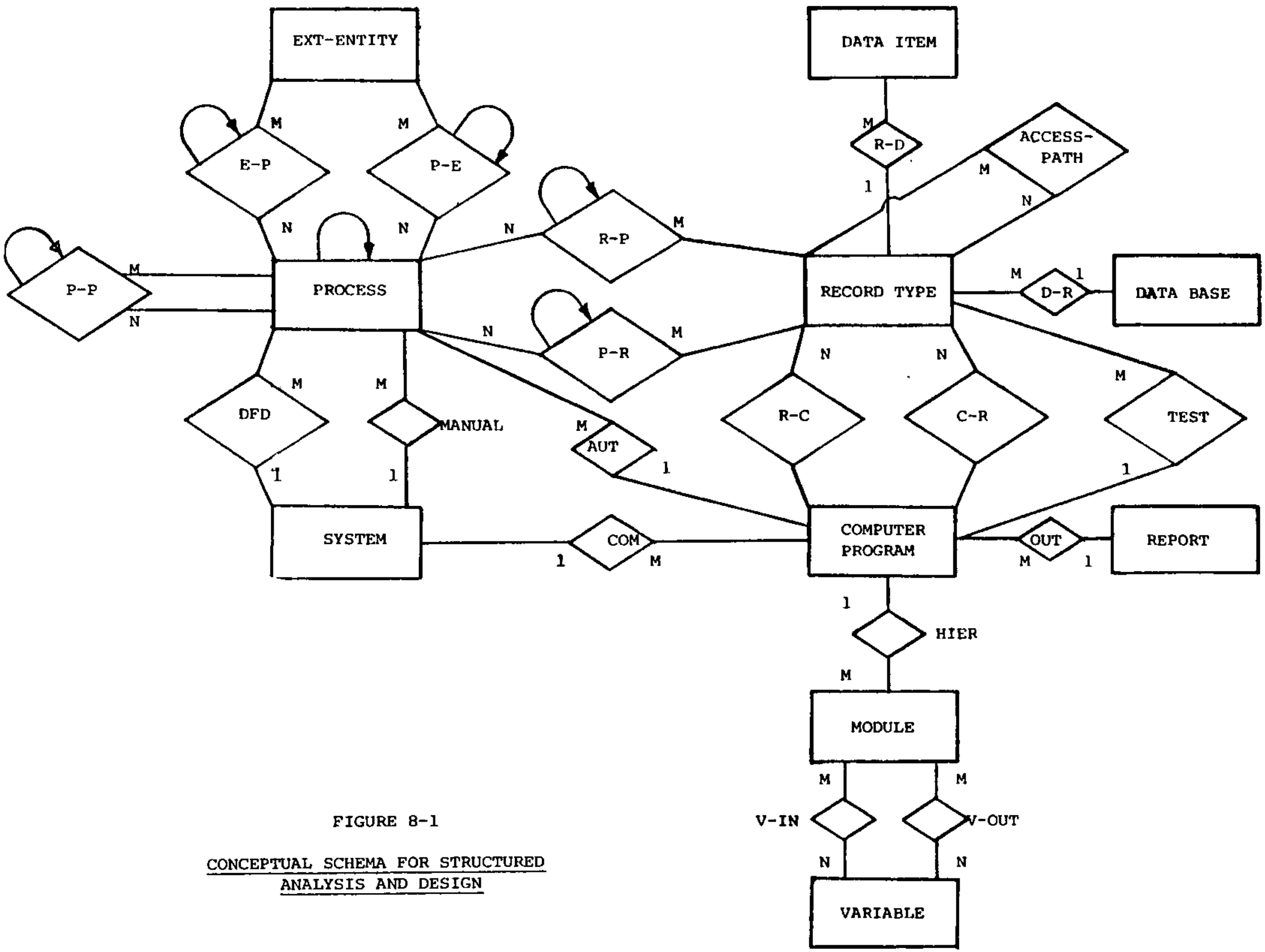


FIGURE 8-1

CONCEPTUAL SCHEMA FOR STRUCTURED ANALYSIS AND DESIGN

POSSIBLE ENTITY ATTRIBUTES	EXTERNAL DOCUMENT REFERENCES
SYSTEM (S-NAME, SHORT-DESCR, STCR)	Requirements Statement, Budget
EXT-ENTITY (E-NAME, E-TYPE, SHORT-DESCR, ROLE, STCR)	Organization Chart, Project Notes
PROCESS (P-NAME, P-NO, SHORT-DESCR, PERFORMED-BY, TRIGGER, STCR)	Process Logic Specification
COMPUTER-PROGRAM (CP-NAME, SHORT,DESCR, AUTHOR, LIB-NAME, SIZE, STCR)	Programmer, User & Maintenance Manuals
MODULE (M-NAME, SHORT-DESCR, AUTHOR, LIB-NAME, SIZE, STCR)	Module Narrative or Pseudo-code etc.
VARIABLE (V-NAME, V-TYPE, DEFINITION, STCR)	Program Manual
DATA-BASE (DB-NAME, SHORT-DESCR, DESIGNER, SIZE, STCR)	File Design
RECORD TYPE (R-NAME, SHORT-DESCR, LIB-NAME, SIZE, STORAGE-MEDIUM, ACCESS-METHOD, STCR)	File Lay-Out Chart, File Design Notes
DATA ITEM (D-NAME, ALIASES, DEFINITION, PICTURE, RANGE, CODE-VALUES-AND-MEANINGS, STCR)	File Design Notes
ACCESS-PATH (SET-NAME, USAGE, ORDERING, STCR)	File Design Notes
REPORT (R-NAME, R-TYPE, SHORT-DESCR, RESPONSE-TIME, FREQUENCY, WHEN-REQUIRED, PRIORITY, NO-OF-COPIES, STCR)	Report/Screen Layout Chart, Project Notes

POSSIBLE RELATIONSHIP ATTRIBUTES	
E-P (DF-NAME, ALIASES, SHORT-DESCR, COMPONENT-LIST, MEDIUM, RESPONSE-TIME, FREQUENCY, SIZE, STCR) (The other data-flows, E-P, P-E, P-P, P-F and F-P are defined similarly)	Document Format, Systems and Procedures Manual
DFD (DFD-NAME, SHORT-DESCR, STCR)	Structured Specification Manual
R-C (DATA-ITEM-LIST, SHORT-DESCR, SIZE, FREQUENCY, STCR) (C-R has a similar attribute list)	File Lay-out Chart, File Design Notes, System Flow Chart
TEST (TEST NO, PURPOSE, SHORT-DESCR, STCR)	Test Case Design, Test File Library

The remaining relationships are non-information bearing.

FIGURE 8-2

CONCEPTUAL SCHEMA FOR STRUCTURED ANALYSIS AND DESIGN records of the stored data base are shown in Figure 8-2. The data base represents components of the Structured Specification after completion of the two phases. This file structure would, however, only be achieved in stages. It has been assumed that the application will be developed using a CODASYL (network) DBMS (5). In a non-data base environment (or in the early stages of structured analysis prior to the data base design step) the RECORD-TYPE entity would be more aptly called FILE; in a relational data base framework it would be called RELATION.

Although it is possible to define data flows as an entity type, it is simpler to regard them as relationships between external entities, processes, and files. Thus the relationship E-P represents a data flow from an external entity to a process; the relationship P-E, a data flow from a process to an external entity. P-F and F-P are similarly defined while the relationship P-P represents the network of data flows between processes. The top-down refinement process (see Figure 3a) is represented by the hierarchiacal relationships (loops) on the data flow and PROCESS entities.

The relationship R-C and C-R represent the input and output flows of data between files and computer programs. Some of the computer program outputs -- printed documents and reports -- are treated separately via the OUT relationship and the REPORT entity. Note that this part of the schema is informationally equivalent to the common System Flow Chart or Run Chart (Figure 5d).

In Structured Analysis, the data flows and processes are uniquely named. A system is represented by its set of data flow diagrams which are developed early in the analysis. This is captured by the relationship, DFD, between SYSTEM and PROCESS. The DFD's encompass both manual and computerized processes. The AUT relationship relates computer programs to the processes they automate.

The top right-hand portion of the schema depicts the logical data base design phase of Structured Analysis assuming a network DBMS of the CODASYL type. A DATABASE consists of a number of RECORD-TYPES composed of DATA-ITEMS; the RECORD-TYPES are related in a network via named ACCESS-PATHS (implemented using the set construct of CODASYL).

The lower part of the diagram represents the Structured Design Phase. Here each program is decomposed into a hierarchy of sub-components (the HIER relation). The data flows between MODULE's are recorded (the V-IN and V-OUT relationships) to determine the level of modularity achieved and to increase the effectiveness of the documentation. This part of the schema is equivalent to a Program Structure Chart (see Figure 5a).

The TEST relationship relates programs/modules to test files and contains information related to the design and execution of test cases.

The schema in Figure 8 could be refined in a number of ways, but it does contain most of the data and logical relationships required to produce a "Structured Software Specification" (see also, Waters (26)). The correspondence between some of the elements of documentation and the data base entities and relationships is shown in Table 1. The order of presentation corresponds roughly to the sequence of steps in the analysis and design processes.

A data base of the type shown in Figure 8 could aid systems analysts and management in the four areas of Project Library Maintenance, Project Management, Analysis, and Report Generation as discussed for the BSP schema in the previous section. Interactive graphics software, terminals, and plotters could help produce and maintain the DFD's and other graphical aids listed in Table 1. Note that one section of the schema represents a conventional data dictionary (Uhrowzcik, (25)). Automated aids of this kind have been found to be extremely useful and are reaching high levels of sophistication. Sometimes the data dictionary is a stand-alone system and sometimes it is an integral part of the DBMS software.

7. A COMBINED SCHEMA

The broader managerial perspective of the BSP approach can now be combined with the more detailed approaches of Structured Analysis and Design. The resulting schema is shown in Figure 9. By this means a connection is established and maintained between the original motivation for the system building effort (CSF's, business tasks, decisions and problems) and the final system specification as developed through layer after layer of increasing detail and complexity.

The links uniting the two previous schemas are very simple. Essentially, the SYSTEM and DATABASE-ELEMENT entities of the longer range aggregate plan are each decomposed into a number of entities and relationships during the subsequent analysis and design phases. A BUS-TASK in the more strategic sense of BSP is related to the mechanical PROCESSes which execute

DOCUMENTATION	COMPONENTS OF SCHEMA
Data Flow Diagram	EXT-ENTITY, PROCESS, RECORD-TYPE, E-P, P-E, P-P, P-R, R-P
Data Dictionary	RECORD-TYPE, DATA-ITEM, E-P, P-E, P-P, P-R, R-P
Process Specifications	PROCESS, E-P, P-E, P-P, P-R, R-P
Man-Machine-Interface	COMPUTER-PROGRAM, AUT, PROCESS, E-P, P-E, EXT-ENTITY
Data Structure Diagram	RECORD-TYPE, DATA-ITEM, ACCESS-PATH
System Flow Chart	RECORD-TYPE, COMPUTER-PROGRAM, R-C, C-R
Program Hierarchy Chart (HIPO)	COMPUTER-PROGRAM, HIER, MODULE
Program Structure Chart (Structured Design)	COMPUTER-PROGRAM, HIER, MODULE, V-IN, V-OUT, VARIABLE
Program-Data Access Table*	COMPUTER-PROGRAM, R-C, C-R, RECORD-TYPE, DATA-ITEM
Program Test Library	COMPUTER-PROGRAM, TEST, RECORD-TYPE

* Displays data access requirements: response-time, frequency, volume of data transport etc.

TABLE 1

<u>CORRESPONDENCE BETWEEN ELEMENTS</u>

<u>OF DOCUMENTATION AND SCHEMA</u>

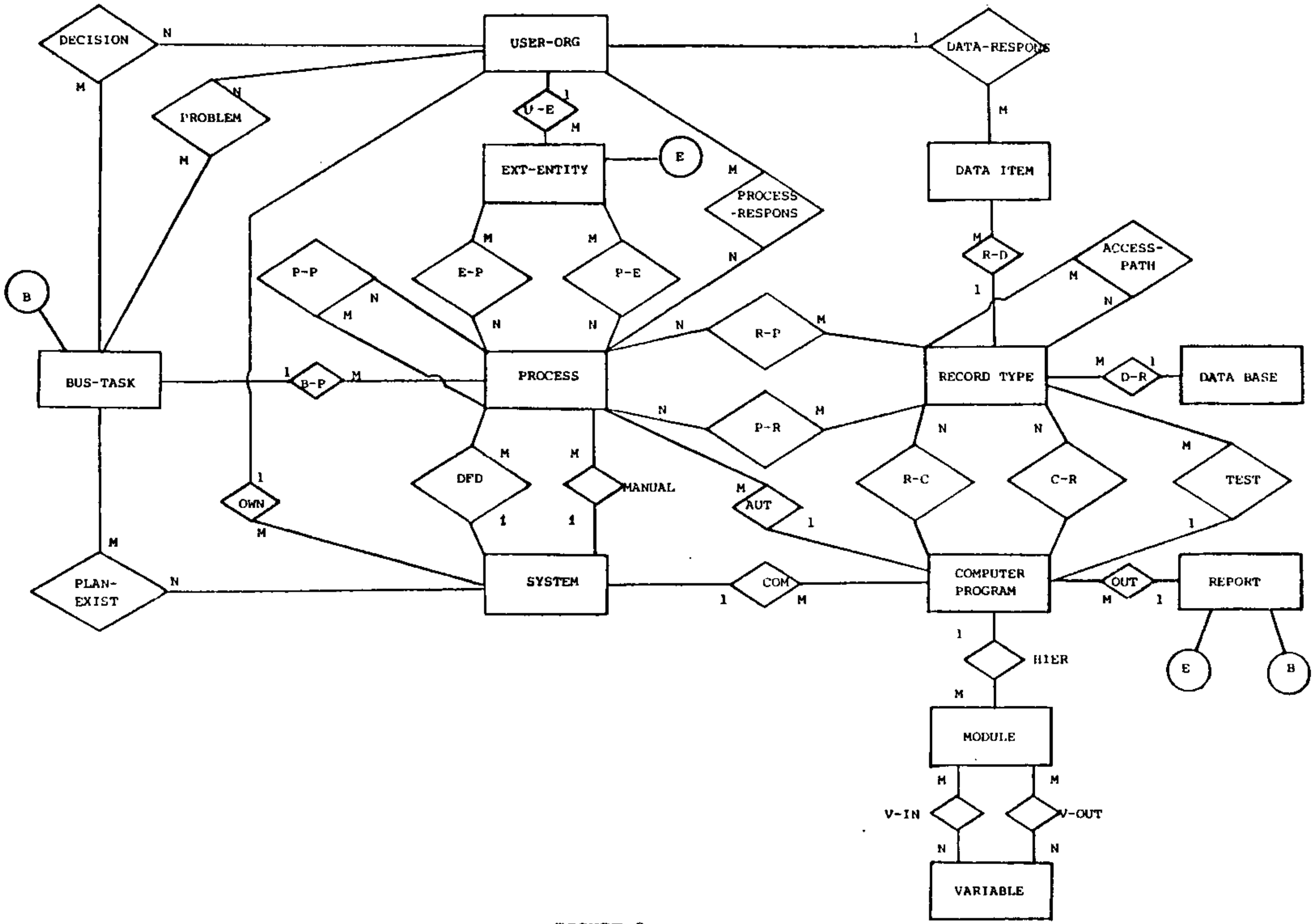


FIGURE 9

COMBINED SCHEMA

it by the one-to-many relationship B-P. User organizations per se are a subset of the External Entities of Structured Analysis. The relationship U-E between USE-ORG and EXT-ENTITY in Figure 9 captures the role of the user organization as a transmitter and receiver of information. The SYSTEM entity in BSP usually refers to a fully computerized system. In the combined schema, the BSP SYSTEM entity is replaced by the somewhat broader SYSTEM concept of Structured Analysis which encompasses both manual and automated elements. A new relationship PROCESS-RESPONS, is added in Figure 9 to record the responsibility of different organizational units for the execution of the PROCESSES. All other entities and relationships retain their original definitions.

8. ADDITIONAL PROJECT CONTROL ELEMENTS

The PLAN-EXIST relationship of BSP is a rudimentary device for recording project scheduling information. A simple project control system can be added to the schema by replacing PLAN-EXIST by the entities and relationships shown in Figure 10.

Before proceeding with a brief explanation of these added elements, notice that for the most part the Schema shown in Figure 9 is an analogue or replica of the real world system. The entities and relationships of the schema stand in one-to-one correspondence with their namesakes in the real system. One exception to this is the DFD relationship which represents an artificial documentation construct. Another exception is the TEST relationship. The project control components added to the schema in Figure 10 also have this artificial flavor.

Three new entities are introduced to record details of PROJECTS, responsibility units within the IS-Department (IS-ORG), and the break-down of the project work into discrete milestones (ACTIVITY). A fourth entity, BUDGET-ACTUAL, records budgeted and actual time and cost measures for each activity.

The REQUEST relationship relates USER-ORG, BUS-TASK, and PROJECT and records contractual information. At any time, each system may have a number of projects in progress as shown by the S-P relationship. The SCHEDULE relationship records the break-down of the project into smaller units of work or activities. The PRECEDENCE relationship records the time dependencies between activities and allows PERT or Critical path network management techniques to be applied. Relationships between the ACTIVITY entity type and elements of the information analogue of the target system (subsystem, computer-program, and data base) are indicated by the connectors. Thus updating the analogue system also serves to update the project control system.

The PERFORM relationship records the responsibility of IS personnel for the performance of specific project activities. The RESPONS relationship pinpoints overall project responsibility while ASSIGN records the assignment of personnel to the development and maintenance of specific systems.

8. CONCLUSION

As can be seen from Figure 10, it is possible to translate the information collection activities of a number of phases of the systems life cycle into a single coherent data base:

- User Requirements Analysis ("Business System Planning") - includes statements of problems and objectives.

- Structured Analysis - data flow diagrams.

- Data Base Design - data dictionary and schema definition.

- Software Design - program/module relationships.

In addition, as indicated in the Figure, the data base can contain information for project control.

The domain of each of these techniques with respect to the data stored in the schema is indicated on the diagram by the partitioning lines. This gives a clear picture of the relationship between the various techniques. In general, the systems development cycle should start with a review of the business, its problems, objectives, and decision making activities. This is the "Business Systems Planning" section of the diagram in the top left-hand corner. Project control mechanisms (lower left-hand corner) should then be put in place. The systems development data base would then be developed from left to right across the top of the diagram, finally ending with the software design activity shown in the lower right-hand corner.

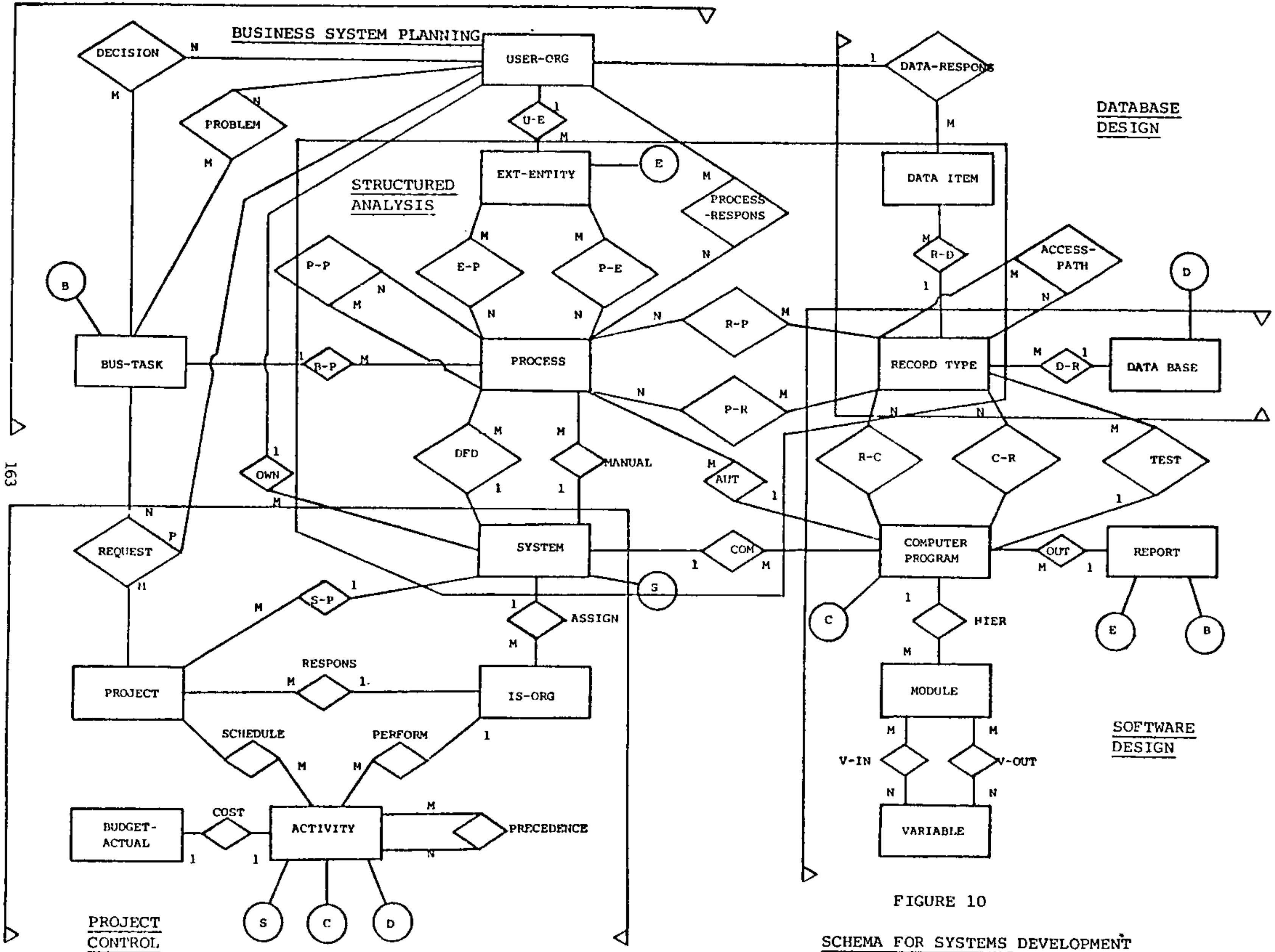


FIGURE 10

SCHEMA FOR SYSTEMS DEVELOPMENT

REFERENCES

1. ANSI/X3/SPARC, Interim Report ANSI/X3/SPARC Study Group on Data Base Management Systems' FDT, 7, 2.

2. Bachman, C.W. Data structure diagrams. Data Base, 1, 1 & 2, (Summer 1969).

3. Chen, P. Data Base Management. QED Monograph Series, No. 6, QED Information Sciences, Inc., Wellesley, Mass., (1978).

4. Churchman, C.W. The Design of Inquiring Systems, Internal Working Paper No. 107, Space Sciences Laboratory, University of California, Berkeley, (1969).

5. CODASYL, Data Base Task Group Report, ACM, New York, (1971).

6. Codd, E.F. A relational model of data for large shared data banks. Comm. ACM, 13, 6, (June 1970).

7. Couger, J.D. Evolution of systems analysis techniques. Couger, J.D. and Knapp, R.W. (Eds.) Systems Analysis Techniques, New York, John Wiley & Sons, (1974).

8. Dahl, O.J. Dijkdtra, E.W., and Hoare, C.A.R. Structured Programming, Academic Press, (1972).

9. DeMarco, T. Structured Analysis and System Specification, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1979).

10. Gane, C. and Sarson, T. Structured Systems Analysis: Tools and Techniques, Prentice-Hall, Englewood Cliffs, N.J., (1979).

11. Grindley, K. Systematics: A New Approach to Systems Analysis, McGraw-Hill, Inc. (UK), (1975).

12. IBM, Business Systems Planning: Information Systems Planning Guide, GE20-0527, (August 1975).

13. Katzan, H. Jr. Systems Design and Documentation: An Introduction to the HIPO Method, Van Nostrand-Reinhold, New York, (1976).

14. Keen, P.G.W. and Scott Morton, M.S. Decision Support Systems: An Organizational Perspective, Addison Wesley, (1978).

15. King, W.R. MIS strategic planning. MIS Quarterly, 2, 1, (March 1978).

16. Leavitt, H.J. Managerial Psychology, University of Chicago Press, (1964).

17. Lucas, H.C., Jr. The Analysis, Design and Implementation of Information Systems, McGraw-Hill, New York, (1976).

18. Lundeberg, M., Goldkuhl, G. and Nilsson, A. A systematic approach to information systems development C-I. introduction. Information Systems, 4, (January 1979).

19. Myers, G.J. Composite/Structured Design, Van Nostrand Reinhold, New York, (1978).

20. National Cash Register Company, A Study Guide for Accurately Defined Systems, Dayton, Ohio, (1968).

21. Nunamaker, J.F., Jr. A Methodology for the Design and Optimization of Information Processing Systems, Proceedings, AFIPS, FJCC Montdale, N.J., (1971).

22. Rockart, J.F. Chief executives define their own data needs. Harvard Business Review, (March-April 1979).

23. Teichroew, D. and Hershey, E. PSL/PSA: A computer-aided technique for structured documentation and analysis. IEEE Transactions of Software Engineering, SE-3, (January 1977).

24. Teory, T.J. and Fry, J.P. The logical record access approach to data base design. Computing Surveys, 12, 2, (June 1980).

25. Uhrowczik, P.P. Data dictionary/directories. IBM Systems Journal, 12, 4, (1973).

26. Waters, S.J. Toward comprehensive specifications. Computer Journal, 22, 3, (August 1979).

27. Wood, C., Fernandez, E.B., and Summers, R.C. Data base security: requirements, policies and models. IBM Systems Journal, 19, 2, (1980).

28. Yourdon, E. and Constantine, L.L. Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design, Yourdon Press, New York, (1977).