IRM-5231-05

U.S. Marine Corps

GENERAL DESIGN SPECIFICATION

PCN 186 523105 00

DEPARTMENT OF THE NAVY
HEADQUARTERS UNITED STATES MARINE CORPS
WASHINGTON, D.C. 20380

5231/05
CCIS-45
16 APR 1987

From: Commandant of the Marine Corps

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) GENERAL DESIGN SPECIFICATION

Ref: (a) MCO 1510.37
(b) MCO P5231.1
(c) MCO 5271.1
(d) MCO P5600.31

Encl: (1) IRM-5231-05

1. PURPOSE. To provide guidance and instructions on the development of General Design Specifications as required by references (a) and (b).

2. AUTHORITY. This publication is published under the auspices of reference (c).

3. APPLICABILITY. The guidance contained in this publication is applicable to all contractors and Marine Corps personnel responsible for the preparation of General Design Specifications. This standard is applicable to the Marine Corps Reserve.

4. DISTRIBUTION. This technical publication will be distributed as indicated. Appropriate activities will receive updated individual activity Table of Allowances for Publications. Requests for changes in allowance should be submitted in accordance with reference (d).

5. SCOPE

a. Compliance. Compliance with the provisions of this publication is required unless a specific waiver is authorized.

b. Waivers. Waivers to the provisions of this publication will be authorized only by CMC (CC) on a case by case basis.

6. RECOMMENDATIONS. Recommendations concerning the contents of this technical publication should be forwarded to CMC (CCI) via the appropriate chain of command. All recommended changes will be reviewed upon receipt and implemented if appropriate.

Subj: INFORMATION RESOURCES MANAGEMENT (IRM) GENERAL DESIGN SPECIFICATION

7. SPONSOR. The sponsor of the technical publication is CMC (CCI).

R. R. PORTER
Brigadier General, U.S. Marine Corps
Director, Command, Control, Communications
and Computer (C4) Systems Division

DISTRIBUTION: L1P
L1Q
L1R
7000006 (2)
7000007 (2)
7000012 (2)
7000014 (2)
7000015 (2)
7000017 (2)
7000045 (2)
7000050 (2)
7000142 (2)
7000148 (2)
7000071 (5)

Copy to: 8145001

UNITED STATES MARINE CORPS

Information Resources Management (IRM) Standards and Guidelines Progra

General Design Specification
IRM-5231-05

16 APR 1987

ENCL (1)

RECORD OF CHANGES

Log completed, change action as indicated.

Change Number	Date of Change	Date Received	Date Entered	Signature of Person Entering Change

PUBLICATION TABLE OF CONTENTS

LIST OF FIGURES

Chapter Table of Contents

Chapter 1

GENERAL

(This page intentionally left blank)

Chapter 1

GENERAL

1.1. INTRODUCTION.

1.1.1. Purpose. The objective of the General Design Specification Standard is to provide the documentational requirements and evaluation criteria necessary to ensure the production of a well-defined General Design Specification of the user's new requirements. Based on the definition of the user's functional requirements and the information regarding the environment in which the new system will operate, a clear, unambiguous General Design Specification is required before starting Detailed Design. The General Design Specification will detail the user's requirements for new, changed, or enhanced applications, and define the environment necessary to meet the proposed system's stated purpose.

1.1.2. Objectives. The specific objectives of this standard are:

a. Define all the required documentation produced in modeling the General Design.

b. Define the format for the identified documentation. (Refer to Appendix B for amplification of formats.)

c. Provide guidelines for the production of the documentation.

d. Provide the criteria through which the completeness, internal consistency, and acceptability of the documentation will be evaluated.

1.2. SCOPE.

These standards will govern the documentation produced during the Modeling of General Design process shown in Figure 1-01, "Structured System Development Activities."

1.2.1. The New Physical Model. Structured system analysis will produce a document called the New Physical Model. The New Physical Model defines the operational environment of the proposed system and equates to a structured specification composed of the following:

a. General information regarding system objective, scope of development activities, responsibilities of development team, and recommendations,

b. Context diagram, or in the case of a clearly defined subsystem, a Relative Context Diagram,

c. Leveled set of data flow diagrams consisting of a figure 0 and associated child diagrams to the functional primitive level,

d. Mini-specifications for all functional primitives declared on the lowest level child diagrams,

e. Data dictionary for all data declared on the data flow diagrams,

f. Supporting information such as performance characteristics, ADPE environment, and summaries of required changes.

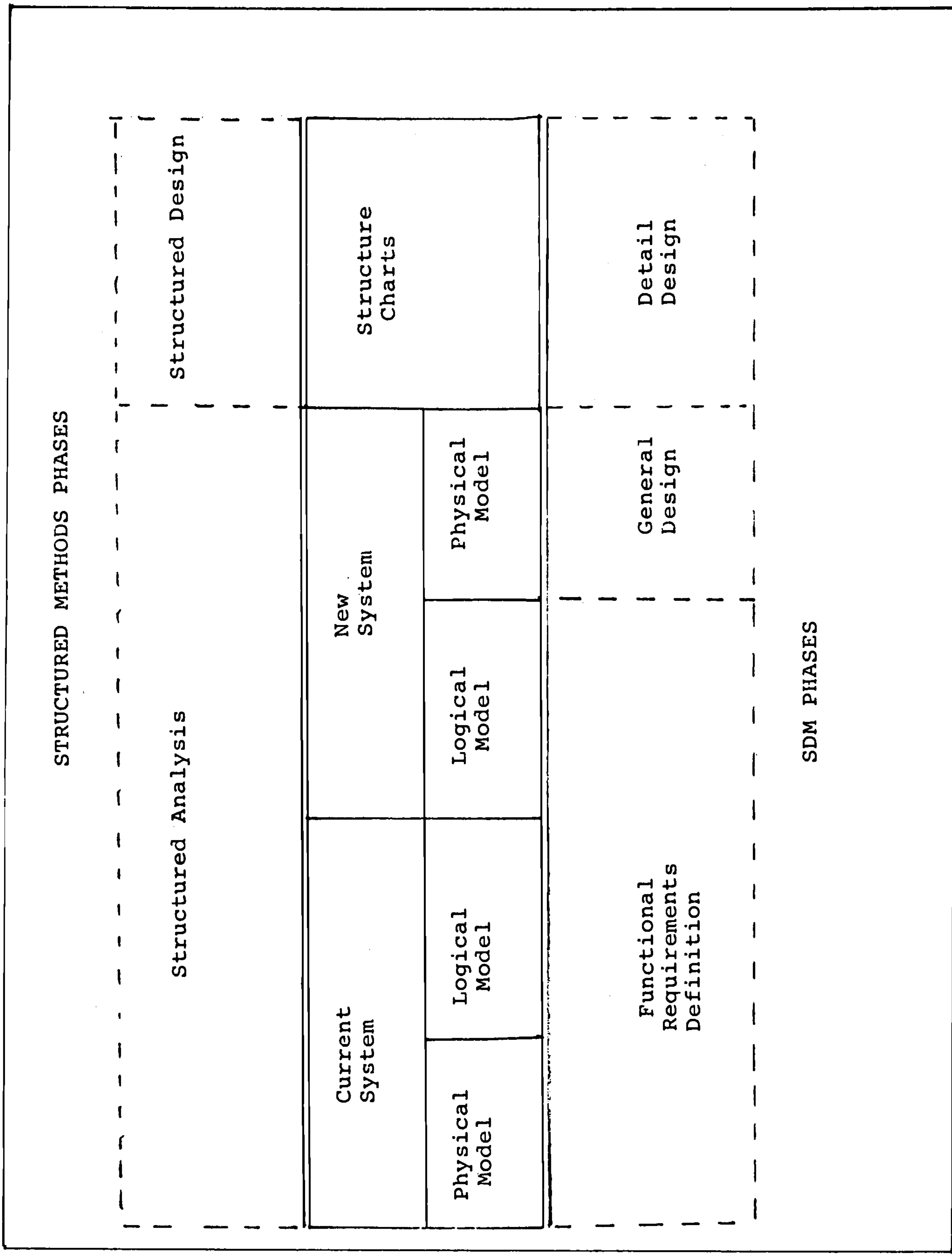


FIGURE 1-01
Structured Systems Development Activities

1.3. APPROACH. Modeling the General Design is the second step in structured systems analysis (Figure 1-01, "Structured System Development Activities"). This step requires the Functional Requirements Definition and knowledge of physical constraints of the proposed environment as input, and results in a specification of the proposed operational environment (New Physical Model).

1.3.1. Definition. The general design incorporates the physical constraints associated with the organizational and technological limitations of the environment in which the new system will be operating. Specifically:

a. Capacity and capability of the hardware and software environment,

b. Organizational responsibilities and job descriptions related to the implementation environment,

c. Geographic allocations required in the new environment such as a Central Design and Programming Activity (CDPA), Regional Automated Service Center (RASC), or Deployable Force Automated Service Center (DFASC).

d. Anticipated performance characteristics of the new system.

1.3.2. Procedures.

a. Initial Steps. During this activity an optimal boundary of automation is established and all of the essential and custodial functions are tailored and allocated to the appropriate physical processors based on an assessment of the implementation environment. Data requirements are modeled to meet the physical reporting and storage requirements of the new system. Implementation dependent functions are then added to the model to perform administrative tasks such as edits, audits, validates, and report formatting. For readers not familiar with the process of developing a physical model from an existing logical model, Appendix B of this standard outlines that relationship. Appendixes C and F provide additional instructional material.

b. Completion. General Design is considered to be complete with the production of the New Physical Model consisting of:

(1) All functions necessary to meet the new system's stated purpose when implemented,

(2) All data necessary to meet the new system's stated purpose when implemented,

(3) All key man-machine interfaces declared at the very top level.

c. Final Step. Once finalized, the General Design Specification is input to the process of Detailed Design.

Chapter Table of Contents

Chapter 2

CONTENT AND FORMAT

(This page intentionally left blank)

Chapter 2

CONTENT AND FORMAT

2.1. DOCUMENTATION STANDARDS. The General Design Specification should be documented in accordance with the guidelines described in the following paragraphs.

2.1.1 New Physical Model. In developing the New Physical Model the practitioner should use the table of contents described in Appendix D, and document the text according to the item descriptions in Appendix E.

2.1.2. Evaluation Criteria. The criteria to evaluate the acceptability of the New Physical Model will be as follows:

a. All sections and paragraphs detailed in Appendix D and E should be included in the document. Any sections or paragraph deemed not applicable to the model by the developing organization will appear with a justification for its exclusion.

b. Figure 0 of the data flow diagram (DFD) should show the organizations, departments, and/or machines that are identified to perform the functions of the new system.

c. Child diagram DFDs should declare the internal organization of these physical processes.

d. Process names should reflect the organization, person, task or task step doing the work.

e. Data flow names should reflect the actual package, form, or vehicle by which the data is transmitted.

f. Data dictionary definitions should include comments to define the physical characteristics of the data such as form number, report formats, and security.

g. Mini-specifications should reflect the new procedure by which work is performed. Sequence of tasks will be specified.

2.2. DOCUMENTATION DEPENDENCIES. The documentation governed by this standard may also rely on the content of other project deliverables and/or standards. Figure 2-01, "Precedence Relationship," shows those project deliverables and standards which impact the General Design Specification deliverables.

2.2.1. Preceding Documents. The boxes that precede the General Design Specification as shown by a connected line with an arrow, are those project deliverables that must be completed before the General Design Specification. The preceding documents for any one development effort are:

a. Functional Requirements Definition Deliverables

2.2.2. Consultation Documents. The boxes and bars that are in line vertically with the General Design Specification show the concurrent documents that may be consulted at that time. The boxes are other project deliverables governed by standards, and the bars are particular conventions described by standards. The deliverables and standards used for consultation are:

a. Data Base Plan (IRM-5231-11)
b. Project Deliverable Style Manual (IRM-5230-02)
c. Inspection and Acceptance (IRM-5231-17)
d. Data Dictionary (IRM-5235-01)
e. Library Management System (IRM-5233-06)
f. Man-Machine Dialogue (IRM-5234-02)
g. Programming Standard (IRM-5234-01)
h. Prototyping Standard (IRM-5231-18)

2.2.3. Change Requirements. Since the SDM is an integrated methodology, there exists a relationship between documents where preceding documents provide information to the follow-on documents. During the development of the General Design Specification new issues may arise that will require changes to preceding documents. These changes must be documented and approved in accordance with the quality assurance and configuration management procedures. Externally imposed milestones that are unrealistic to accomplish should not be used as an excuse to defer or eliminate the documentation requirements.

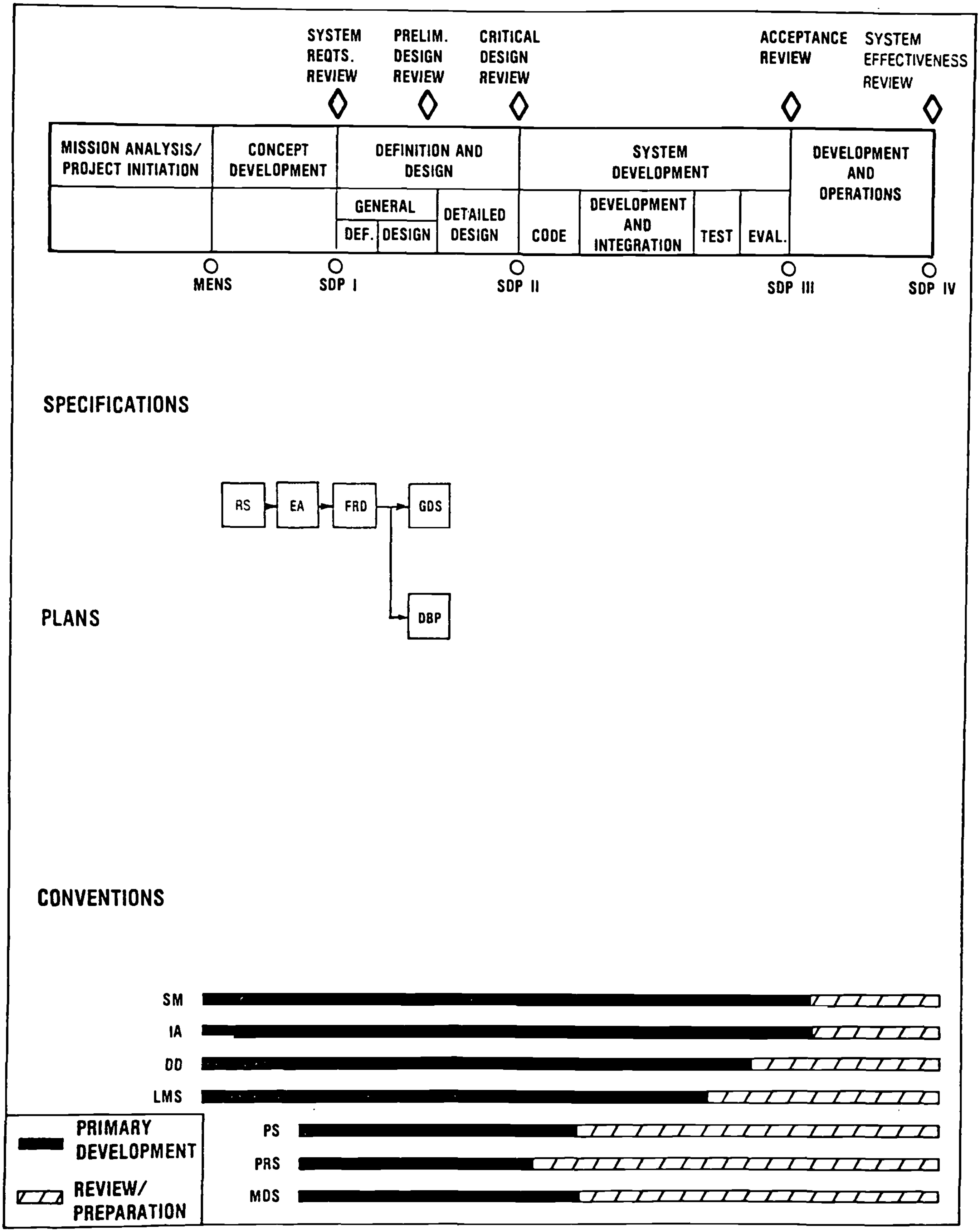


FIGURE 2-01
Precedence Relationship

(This page intentionally left blank)

Appendix A

GLOSSARY

DBP: DBP is an acronym for "Data Base Plan"

DD: DD is an acronym for "Data Dictionary"

EA: EA is an acronym for "Economic Analysis"

FRD: FRD is an acronym for "Functional Requirements Definition"

GDS: GDS is an acronym for "General Design Specification"

IA: IA is an acronym for "Inspection and Acceptance Standard"

LMS: LMS is an acronym for "Library Management System"

MDS: MDS is an acronym for "Man-Machine Dialogue Standard"

MENS: MENS is an acronym for "Mission Element Need Statement"

PRS: PRS is an acronym for "Prototyping Standard"

PS: PS is an acronym for "Programming Standard"

RS: RS is an acronym for "Requirements Statement"

SDP: SDP is an acronym for "System Decision Paper"

SM: SM is an acronym for "Project Deliverables Style Manual"

(This page intentionally left blank)

Appendix B

STRUCTURED ANALYSIS

Table of Contents

SECTION 1 INTRODUCTION

The Structured Specification (FRD and GDS) is the product of Structured Analysis. The Structured Specification is both a model of the system under investigation and the documentation of that system. For the system developer as well as the user, the Structured Specification is the vehicle through which system requirements are communicated.

This guide will describe the Structured Specification (or model), its organization, major components, supporting tools, and the criteria that ensures its internal consistency and completeness.

This guide will apply to the models developed during the Functional Requirements Definition (FRD) and General Design Specification (GDS). Specifically:

a. Current Physical Model - The model defining the operational environment of the existing system presented in the FRD.

b. Current Logical Model - The model defining the fundamental or essential functions which the existing system performs as well as the minimum stored data required to support those functions presented in the FRD.

c. New Logical Model - The model defining the fundamental functions which the new system must perform as well as the fundamental stored data to support those functions presented in the FRD.

d. New Physical Model - The model defining the operational environment of the proposed system presented in the GDS.

SECTION 2 CONCEPTUAL MODEL ORGANIZATION

The physical model developed in the GDS is a refinement of the logical model presented in the FRD. In both cases, the model is composed of three major components. One of these components, the data flow diagram, is a graphic tool. The other two components, the mini-specification and data dictionary, are primarily narrative tools. In the model, these tools are organized in such a way as to eliminate redundancy in the specification of system requirements. Underlying the model and its organization is the concept of controlling the complexity of the specification through dividing, or partitioning the model into small conceptually complete functions. The major components of the model are defined as follows:

a. Data Flow Diagram (DFD) - A network of related mini-systems and the data flows and data stores interfacing those mini-systems

b. Data Dictionary - A set of definitions of data flows and data stores declared on the DFD

c. Mini-specification (mini-spec) - Statement of the rules governing the transformation of input data flows into output data flows the mini-systems perform at the functional primitive level

To eliminate redundancy in the specification of system requirements each component of the model has a very limited charter. Specifically:

a. The DFD declares the mini-systems and the data interfacing those mini-systems. DFDs do not show:

(1) Data flow composition

(2) Data store composition

(3) Rules governing transformations

(4) Control or subordination

(5) Prompts, interrupts, flags

b. The data dictionary defines the composition of data flows and data stores. Data dictionaries do not show:

(1) How data was derived

(2) Source of data

(3) Destination of data

c. Mini-specs state the rules governing the transformation of data. Mini-specs do not show:

(1) Data flow composition

(2) Data store composition

(3) Source of data

(4) Destination of data

SECTION 3 DATA FLOW DIAGRAMS

This section describes and defines the components of a data flow diagram (DFD), a context diagram, a leveled set of DFDs, and the criteria to ensure the completeness of, and consistency between the diagrams.

Each diagram described and defined below will be produced on margined paper with title/heading information located in the lower right hand corner containing the following:

a. System Name - The name of the system or subsystem defined as the context of the study; for example:

(1) Transportation Management

(2) Weapons System Equipment Management

b. Specification Type - The name of the specification type being produced; for example:

(1) FRD (Functional Requirements Definition)

(2) GDS (General Design Specifications)

c. Version Type - The name of the version type being produced; for example:

(1) Current Physical

(2) Current Logical

(3) New Logical

(4) New Physical

d. Figure Name - The descriptive name of the figure being produced

e. Figure Number - The number of the figure being produced (except in the case of a context diagram which will be identified as "Context Diagram")

f. Responsible Organization - The name of the organization or person charged with producing the figure

g. Date - The date the figure was produced

See Figure B-01, "Standard Diagram Format," for an example.

Data Flow Diagram Components

A data flow diagram (DFD) is a graphic network of related mini-systems (or process) and the data flows and data stores interfacing those mini-systems. The components of a DFD are defined as follows:

a. Data Flow - A curved line, or arc, that represents the flow of information or data. Each arc will have an arrow indicating the direction of the data flow. The name of the data flow is written next to the arc and must summarize the content of the data flow.

b. Mini-system (or process) - A circle that represents a transformation of incoming data into outgoing data. The name of

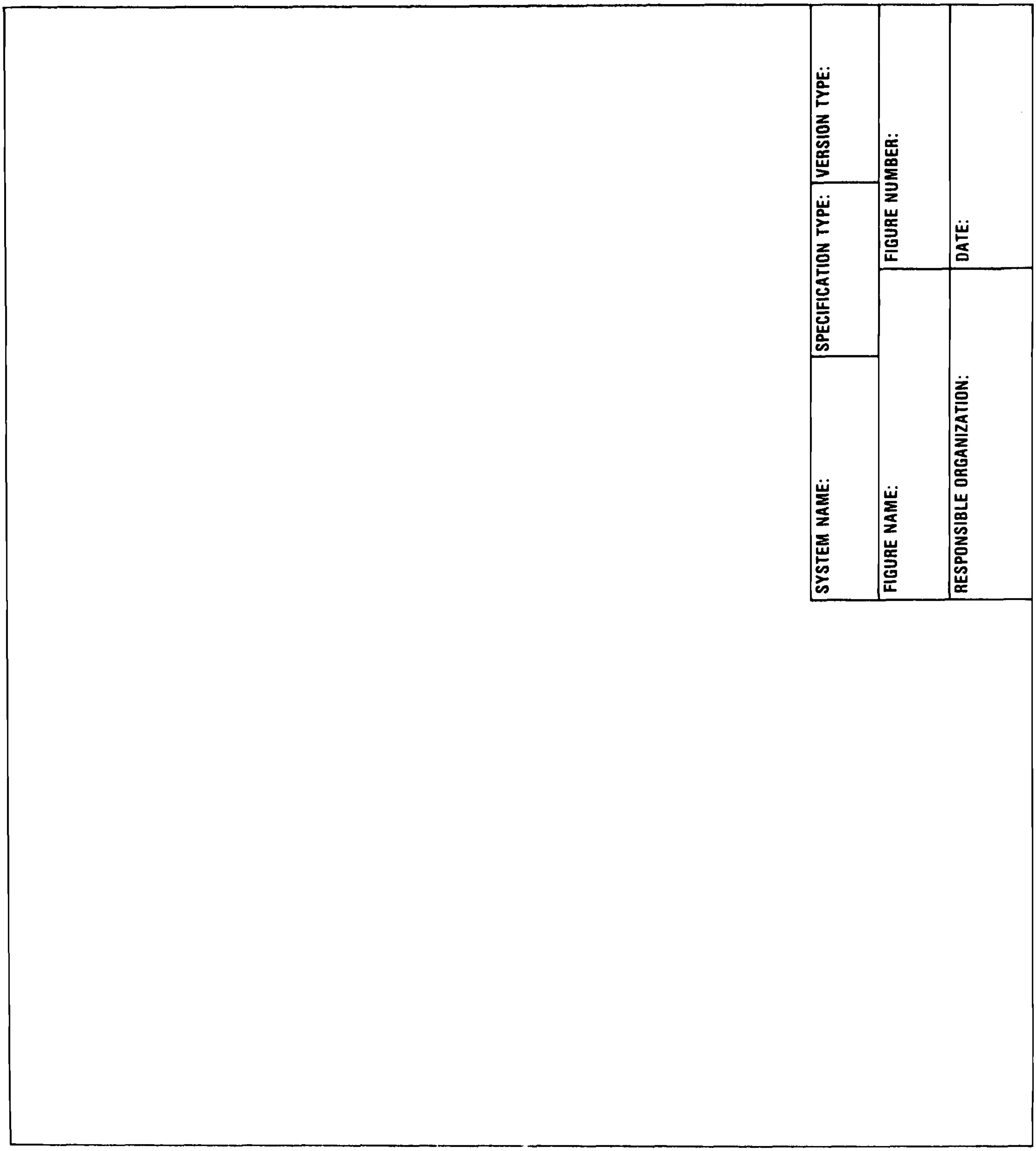


FIGURE B-01
Standard Diagram Format

the mini-system is written within the circle describing the function the mini-system is to perform, preferably as verb/noun combination.

c. Data Store - A set of parallel lines that represent a repository of data. This includes all data that is stored, or remembered, by the system and required by its mini-systems. The name of the data store is written between the parallel lines and must summarize the content of the data store.

d. Source/Destination - A rectangle that represents a source or destination for system inputs and outputs respectively. The source/destination name is written within the rectangle.

See Figure B-02, "Data Flow Diagram Components," for examples.

To ensure the completeness of the DFD, the following criteria must be met:

a. All title/heading information completed.

b. All data flows must be named.

c. All mini-systems or processes must be named.

d. All data stores must be named.

e. Output data flows from a mini-system or process must have been input in some form.

f. Input data flows to a mini-system or process must leave that mini-system in some form.

g. A DFD should not contain more than nine (9) mini-systems or processes to improve readability and control complexity.

Context Diagram Components

The context diagram, or relative context diagram, is the top-level graphic representation of the system or subsystem under study. It depicts all the net inputs to and outputs from the system, but shows no decomposition or partitioning of the system. The components of a context diagram are defined as follows:

a. Net Data Flow - A curved line, or arc, that represents the net flow of information or data into or out of the system. Each arc will have an arrow indicating the direction of the net data flow. The name of the net data flow is written next to the arc and must summarize the context of the net data flow.

b. System - A circle that represents the entire system or subsystem under study. The name of the system is written within the circle.

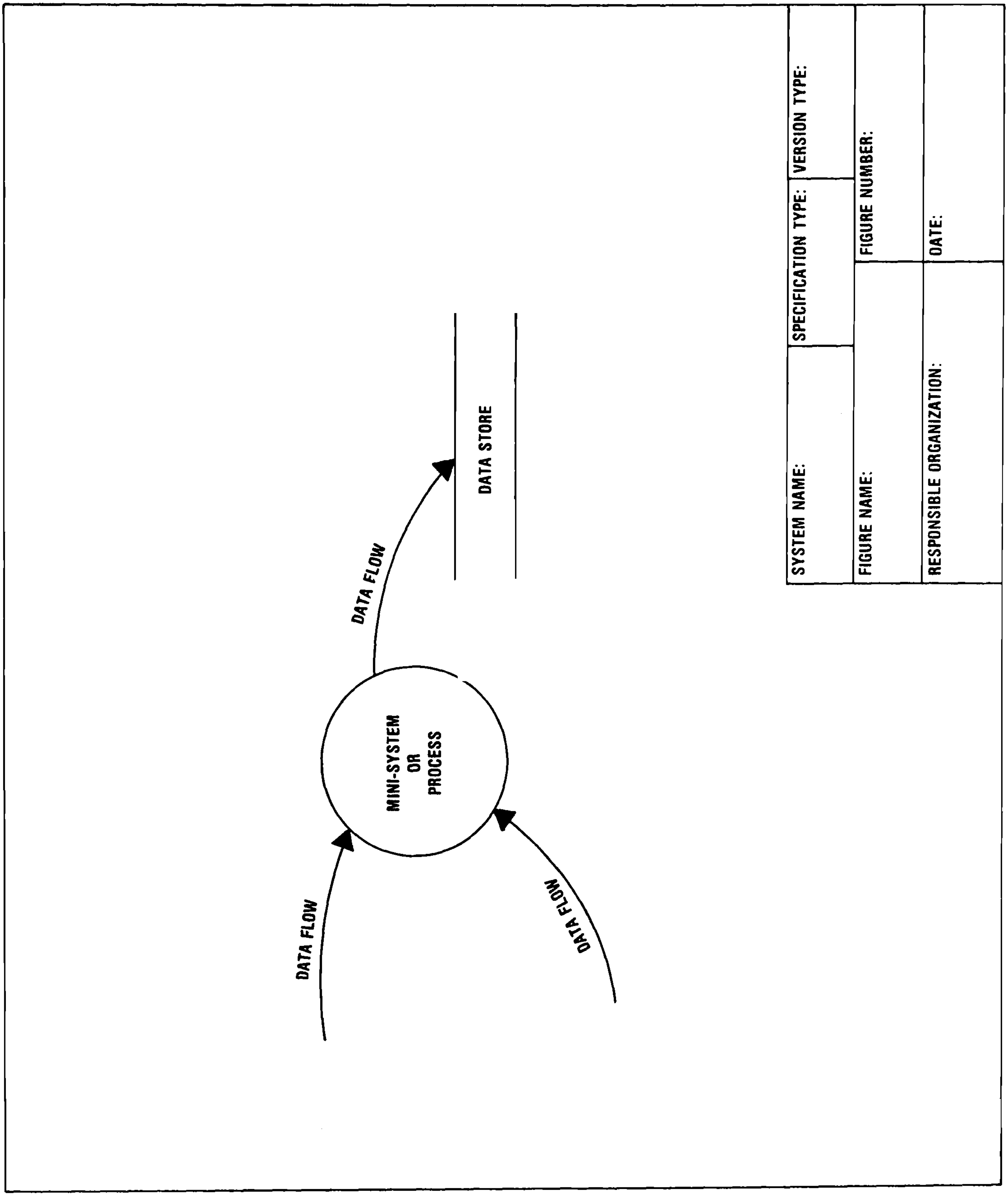


FIGURE B-02
Data Flow Diagram Components

c. Source/Destination - A rectangle that represents a source or destination for system inputs and outputs respectively. The source/destination name is written within the rectangle.

See Figure B-03, "Context Diagram Components," for an example.

To ensure the completeness of the context diagram the following criteria must be met:

a. All title/heading information completed.

b. All net data flows must be named.

c. The system, or subsystem, must be named.

d. All source/destinations must be named.

Leveled Set of DFD Components

A leveled set of DFDs graphically represents the entire system under study and the dividing, or partitioning, of that system into functionally primitive mini-systems or processes. A leveled set of data flow diagrams consist of the following:

a. Context Diagram or the Relative Context Diagram - The top-level graphic representation of the system, or subsystem under study.

b. Figure 0 - The next level graphic representation showing a high level partitioning of the system under study. Each mini-system or process declared on Figure 0 is assigned a number starting with 1. This number is for reference purposes only and in no way denotes sequence of processes.

c. Child Diagrams - A lower level graphic representing the further partitioning of a parent mini-system from the previous level. The child diagram figure name and figure number will be that of its parent mini-system and each mini-system declared on the child figure is assigned the number of the figure number, a decimal point, and a unique local number beginning with 1. A mini-system declared on a child figure could be further partitioned and require a child figure in turn.

See Figure B-04, "Leveled Set Of Data Flow Diagrams," for an example.

The completeness of the leveled set of data flow diagram will be ensured if the completeness criteria for both the DFDs and the context diagram is met.

To ensure the internal consistency of a leveled set of data flow diagrams, the following criteria must be met:

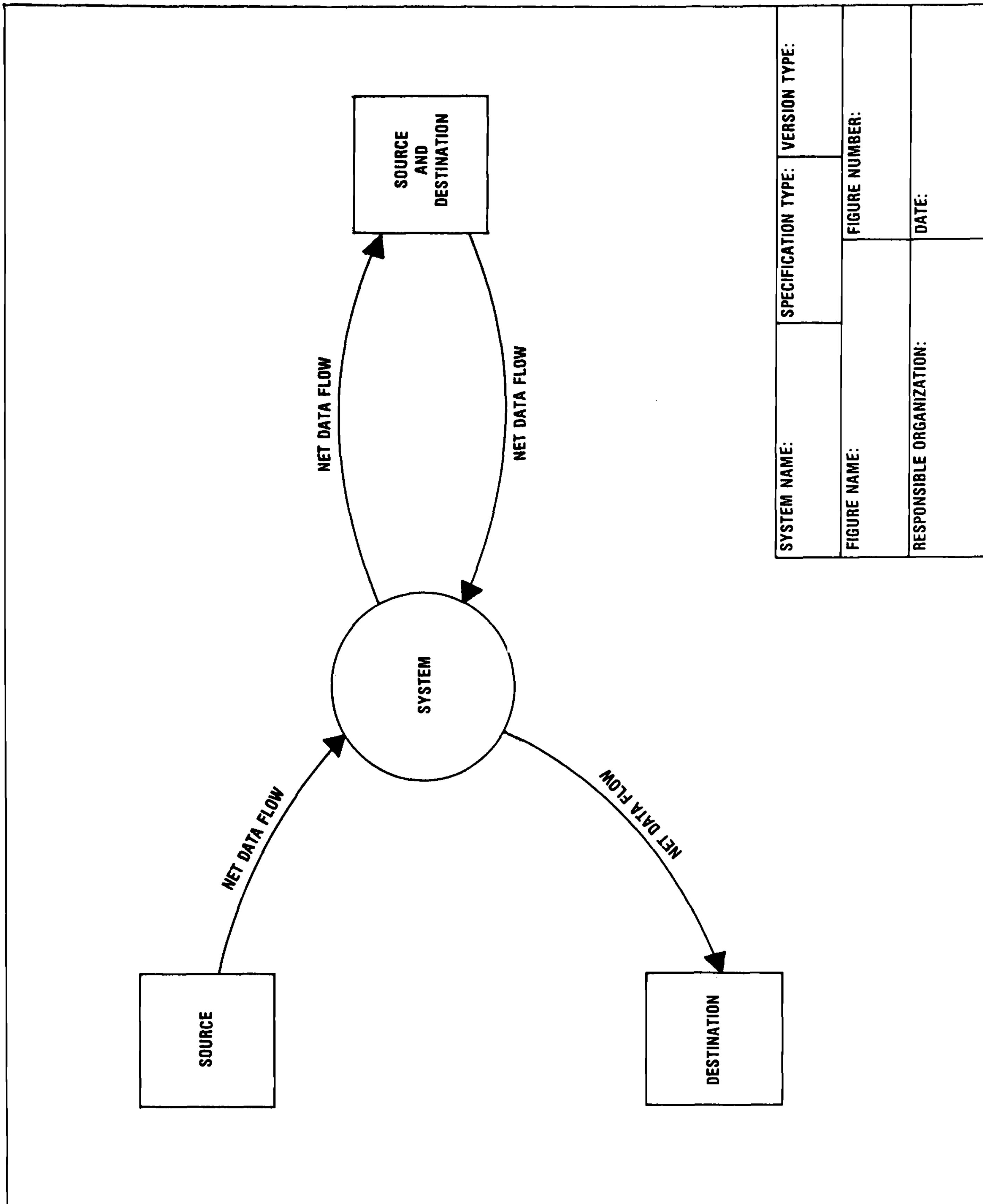


FIGURE B-03
Context Diagram Components

THE LEVELED SET

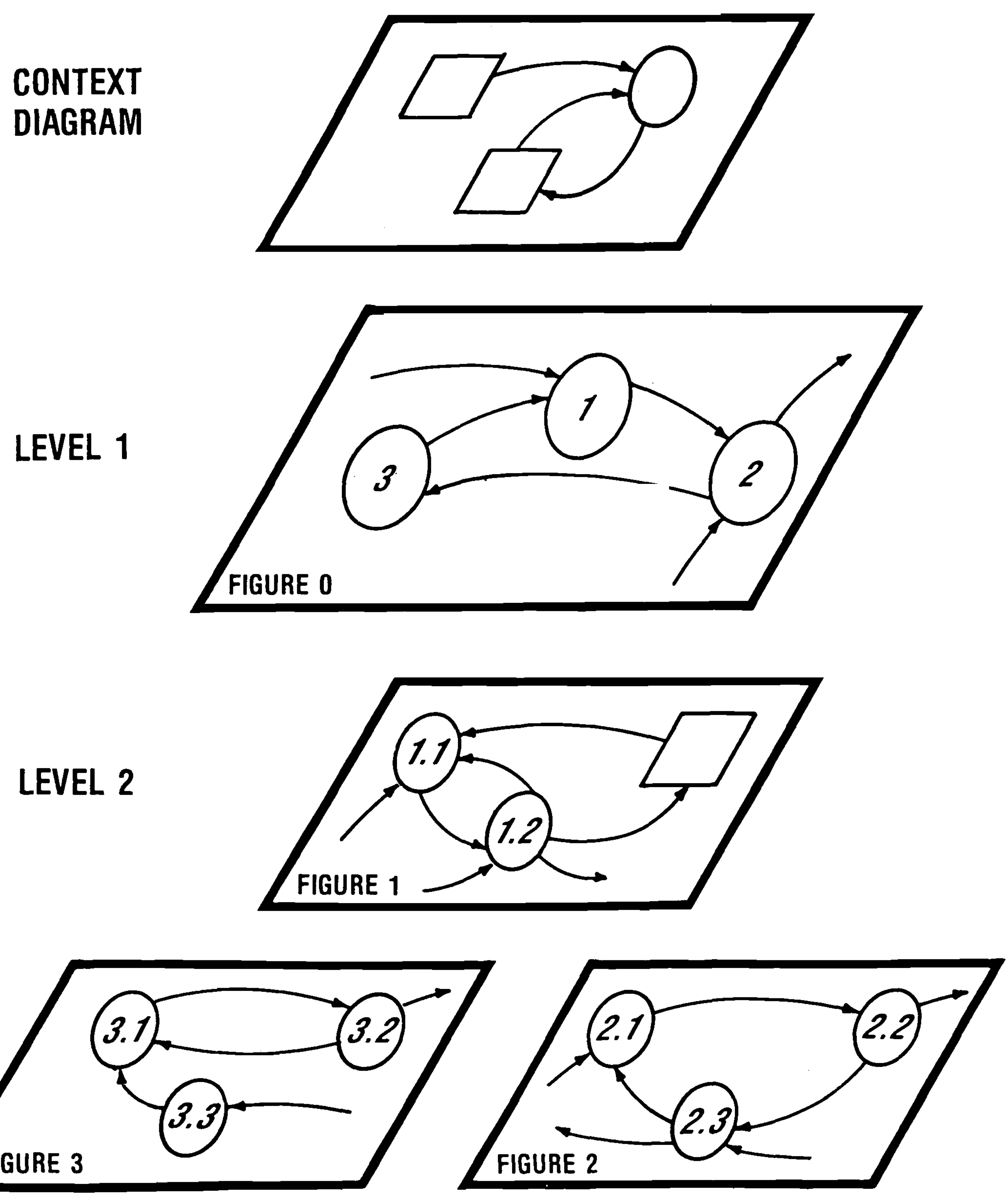


FIGURE B-04
Leveled Set of Data Flow Diagrams

a. Equivalence between a child figure and its parent mini-system or process must be maintained by:

(1) Matching or balancing the data flows entering and leaving a child figure to the data flows entering and leaving the parent mini-system or process,

(2) Matching or balancing references to data stores by declaring a data store only when it is shared or accessed by two or more mini-systems, and subsequently making references to those stores.

SECTION 4 THE DATA DICTIONARY

The Data Dictionary is a set of definitions for every data flow and data store declared in a leveled set of data flow diagrams. The composition of each data flow and data store will be defined as well as the definition of the components of the data flows and data stores to the lowest element level.

Data types can be either groups or elements. Groups may be composed of groups, groups and elements, and/or elements. Elements are the lowest level data item. Each data store or data flow must show its composition as either:

a. groups

b. groups and elements

c. elements

Each group and element defined in a data store or data flow definition must also be further defined by showing composition of the groups and definition of elements.

Each data dictionary entry will contain:

a. Data flow or data store name as declared on the DFD,

b. Composition,

c. The number of times a data component occurs (where applicable); for example, a "Monthly Total" could have at least 1 but not more than 12 occurrences,

d. Optional data components (where applicable),

e. Choice of one occurrence from a set of occurrences (where applicable).

To ensure the completeness and internal consistency of a data dictionary the following criteria must be met:

a. All data flows declared on the leveled DFDs must be defined.

b. All data stores declared on the leveled DFDs must be defined.

c. No redundant data component names will remain.

d. No redundant data component definitions will remain.

The use of a mechanized tool for creating and maintaining the data dictionary will be governed by IRM-5235-01.

SECTION 5 MINI-SPECIFICATION

The mini-specification (mini-spec) is a statement of the logical requirements governing the transformation of input data flows into output data flows at the functionally primitive mini-system, or process level. Mini-specs will be written only for those mini-systems declared at the lowest level. Each mini-spec will contain the following seven sections, formatted and described as follows:

1.0 Process Number

The number of the mini-system or process as declared on the DFD.

2.0 Process Name

The name of the mini-system or process as declared on the DFD.

3.0 Literal Name

The name of the mini-system or process with abbreviations and acronyms spelled out in full.

4.0 Logic

The rules governing the transformation of input data flows into output data flows. The transformation may be documented using Structured English, decision tables, or decision trees.

Structured English is a subset of the English language with limited syntax and grammar. The syntax of structured English is limited to:

° A single or sequenced group of simple declarative statements. COBOL verbs. (When Appropriate)

° Closed-end decision construction. One statement or conditional statement per line.

° Closed-end repetition construction.

See Figure B-05, "Structured English Constructs," for an example.

Decision Tables and/or Decision Trees are used in the specification of a mini-system or process if the nesting of conditions exceeds two levels.

See Figure B-06, "Decision Tables and Decision Trees," for an example.

5.0 Responsible Problem Definer

The analyst or organization responsible for the production of the mini-spec

6.0 Date Produced

The date the mini-spec was produced.

7.0 Date of Last Change

The date of the most recent change to the mini-spec.

The completeness of a mini-spec will be ensured if the following criteria are met:

a. Items 1 through 7 are completed.

b. All data referenced on the leveled DFD's is defined in the data dictionary.

c. Statements must be clear, imperative, and have only one meaning.

d. Algebraic equations follow mathematical, not programming, notation

e. Decision tables or decision trees

(1) Identify all relevant conditions.

(2) Identify all possible values of the conditions.

(3) Identify all possible combinations of the values.

(4) Identify the action for each combination.

The use of a mechanized tool for creating and maintaining the mini-specifications will be governed by IRM-5235-01 Data Dictionary."

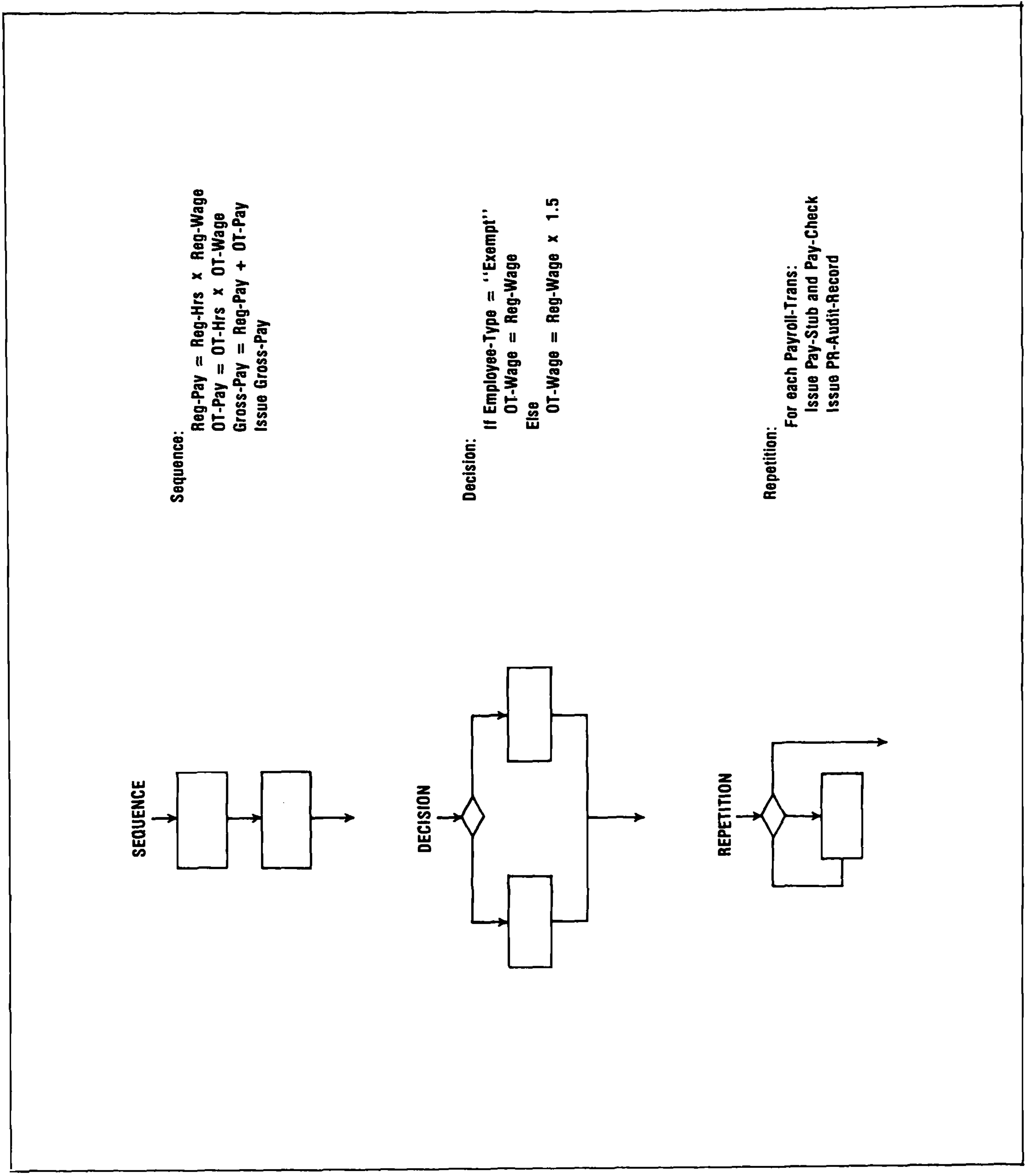
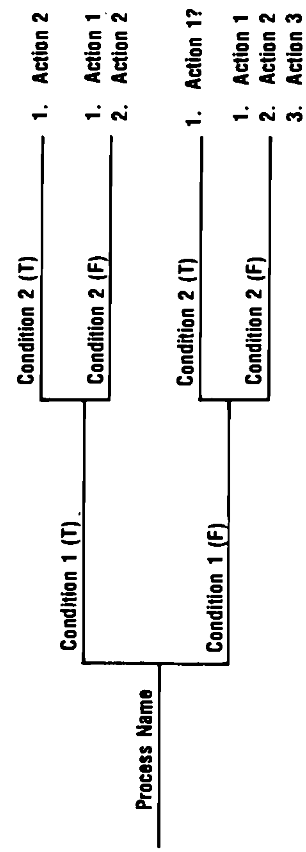


FIGURE B-05
Structured English Constructs

Process Example of a Decision Table

Process Name				
Condition 1	T	T	F	F
Condition 2	T	F	T	F
Action 1		X	?	X
Action 2	X	X		X
Action 3				X



Abstract example of a decision tree

FIGURE B-06
Decision Tables And Decision Trees

SECTION 6 INFORMATION MODELING

This section describes the product of an approach to modeling essential memory known as entity-relationship analysis. The specific product is the entity-relationship diagram (ERD). The ERD is a graphic tool used for modeling objects or entities in essential memory, and identifying the relationships between them. This material is supporting documentation presented with the FRD and/or the GDS.

The primary use of an ERD is to have a vehicle through which the analyst can identify the capabilities of the New Logical Model (Functional Requirements Definition) to meet the informational requirements of the users. Issues concerning stored data and how that data relates through identified processes can be raised using an ERD. Whereas a data flow diagram could be viewed as an "output" producing model, the ERD could be viewed as an "answer" producing model.

Entity-relationship analysis is an approach to modeling, or partitioning essential memory into objects or entities, and identifying the relationships between those entities. This analysis is performed during the Functional Requirements Definition, specifically when developing the New Logical Model. The analyst relies on two of the components of the New Logical Model for this analysis; the data dictionary and the mini-specifications.

The data dictionary contains information about essential memory, the total of all data elements remembered, or stored by the Logical Model and required by the essential functions that are fundamental to the system's stated purpose. The mini-specifications identify all essential accesses for both fundamental and custodial functions. The mini-specifications identify all essential accesses for both fundamental and custodial functions. It is from the mini-specifications that information regarding inter-object relationships is obtained.

In entity-relationship modeling the analyst identifies the data elements of essential memory in the data dictionary to begin the process of object partitioning. In object partitioning the analyst groups the elements of essential memory into objects, or stores, by simply keeping like or related elements together and keeping unlike or unrelated elements apart. Each object, or store, is given a name that describes the criteria for grouping the elements together. The analyst then organizes the elements within the object and selects an identifier element, or key, that completely describes all occurrences of that object. The objects are identified on the New Logical Model data flow diagrams as data stores used in the entity-relationship diagrams and the required access relationships from the mini-specifications as relationships.

An Information Model is composed of the following four components:

a. Entity Relationship Diagram - A network which describes the stored data layout of a system.

b. Data Dictionary - Extended definitions for all data elements identified in the Data Dictionary.

c. Object Specifications - Each object of an Information Model defines some aspect of business policy and requires a substantial policy definition.

d. Relationship Specifications - A relationship is a policy-based association among objects. The policy definition must be specified for each relationship.

The connection between the information and process models occurs at two points:

a. A data store in the essential version of the process model represents one or more interrelated objects in the information model.

b. A single Data Element Dictionary supports the major components of both models.

The entity-relationship diagram (ERD) is a graphic tool used to identify objects, or entities and the relationships between them. The ERD has two basic components:

a. Entity - An object, or store of data. The entity is represented on the ERD as a box.

b. Relationship - A set of connections that relate two or more entities. The relationship is represented on the ERD as a diamond.

In the case where a relationship is also a data store it is represented as a box with a superimposed diamond.

Figure B-07, "Entity-Relationship Diagram," is an example of the usage of the symbols.

For any identified relationship:

a. Select the object, from all the participating objects, on which the relationship can be logically based.

b. Define the relationship by stating the rules, laws and conditions that determine the association of the participating objects.

c. Define the other relationships that exist between the anchor object and the remaining objects.

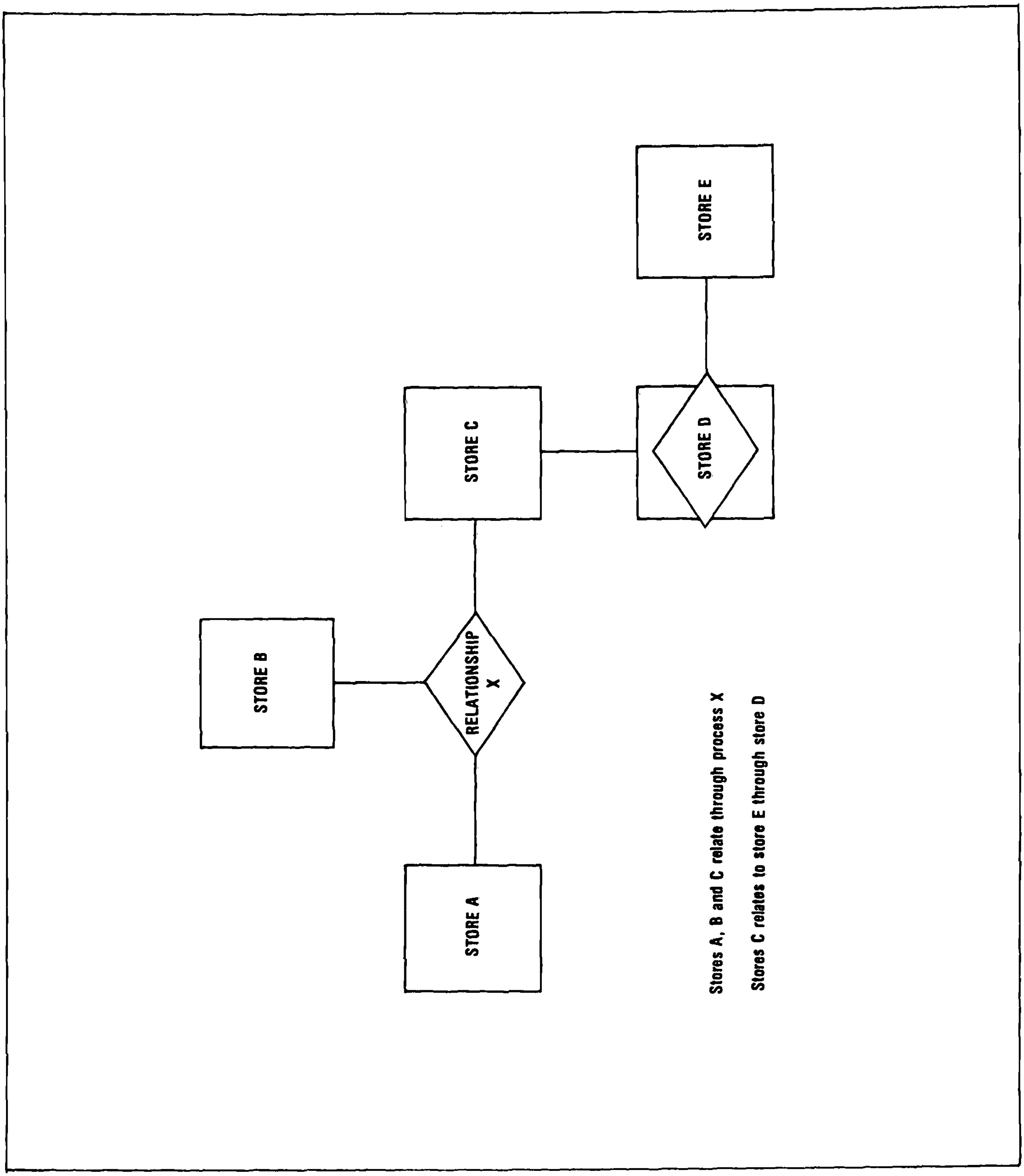


FIGURE B-07
Entity-Relationship Diagram

The criteria by which the completeness and internal consistency of an ERD can be evaluated consists of:

a. Every data store declared on the New Logical Model data flow diagrams after entity-relationship analysis must be declared on the ERD.

b. Every relationship on the ERD must be named.

c. There must be a process in the New Logical Model capable of creating and/or maintaining each relationship declared on the ERD.

(This page intentionally left blank)

Appendix C

DATA FLOW DIAGRAM CHECK LIST

This check list is to provide a guideline for developing and/or evaluating Data Flow Diagrams. While these guidelines are somewhat flexible, they will be followed unless individual deviations are justified.

A. General

1. Context Diagram

Definition: A "departitioned" version of Diagram O, it shows only the net inputs and outputs. A set of data flows that cross into and out of the domain (Example 1).

a. Have I captured the scope of the system (System definition: Includes a set of procedures, both automated and manual)?

b. Do my DFD's reflect both manual and automated procedures?

2. Can a manager read the top few levels of my DFD and get the big picture?

3. Have I partitioned into no more than nine pieces?

4. Have I worked my way from inputs to outputs, then checked by going backwards from outputs to inputs, and/or from the middle out?

B. Numbering

1. Have I observed the numbering conventions?

a. Numbering conventions are:

Example: Context Diagrams - RPPSII: Diagram O,

Level 1 - Allotments as 1.0, Survivor Annuities as 2.0, etc.: Diagram's 1,2,3, etc., Level 2 and would be numbered 1.1, 1.2; 2.1, 2.2, 2.3, etc.

(2) On large systems with many levels, long bubble numbers can be abbreviated. Example: The diagram is numbered - Diagram 5.8.4 and the child bubbles are number .1,.2,.3, etc. (Diagram 5.8.4 portion of the number would appear on the page corner.)

(3) You can tell what level a given figure is by counting the decimal points in the numbers given to its bubbles. The level is always one more than the number of decimal points in the bubble numbers.

C. Data Flows

Definition: A pipeline through which packets of information of know composition flow.

1. Have I concentrated first on data flows? Concentrating on the naming of processes after Data Flows have been named?

2. Have I labeled all the data flows properly (specific-uniquely)? Must apply to the whole data flow, not just to its major component.

3. Do all data flows show the direction of the arrow leading to and from a process, file, or terminator?

4. Have I chosen a name that applies to the whole data flow; (avoid names like 'data' and 'information')?

5. Have I grouped items together when they should not be treated as a whole (un-nameable data flows are particularly likely to be caused by a poor partitioning decision-start over!)?

6. Have I identified all net input and output data flows - have I left any out of the Context Diagram?

7. Have I hyphenated my Data Flow names?

D. Process Bubbles

Definition: A process is a transformation of incoming data flow(s) into outgoing data flow(s).

1. Have I used a descriptive name, (does it give the user a general idea; beware of words like 'process' and 'handle')?

2. Do the process bubbles show the transformation of incoming data flow into outgoing data flow?

E. Procedural Annotations

1. Have I used an asterisk (*) to denote duplicated terminators and data stores?

F. Files/Store

Definition: A temporary repository of data.

1. General

a. Are all files annotated?
b. Do my file names make sense?

2. Local Files - Definition: Relevant to only the insides of one parent bubble.

a. Are all local files shown on the DFD at the first level where they are used as an interface between two processes (e.g. a file used only by more than one process of a child diagram)?

G. Source, Sink or Terminator

Definition: A person or organization lying outside the context of a system; that is, a net originator or receiver or system data.

1. Have sources or sinks been annotated and are they all inclusive?

H. Leveling

1. Have my child relationships presented exactly the same transformation as the parent relationships - net inputs and outputs equivalent (rule: all data flows shown entering or exiting a child diagrams must be represented on the parent by the same data flow into the associated bubble - Example 2)?

2. When the equivalency of inputs or outputs rule is used (other than those shown as identical), these differences must be defined in the Data Dictionary. (See Example 3)

3. At any given level, are only files and data flows that are interfaces among DFD elements shown? Are files and data flows that are only relevant to the inside of some process concealed?

4. At the first level where a file is used as an interface between two processes, are all references to it shown?

I. Process Specs or Mini-Specifications

1. Are my mini-specs no more than one page in length? If they are longer, why are they?

2. Do I have one mini-spec for each DFD bubble which is not further decomposed?

3. Are the mini-specs marked with the bubble number of the related bubble?

4. Are all data elements consistent with those on my DFD and the Data Dictionary?

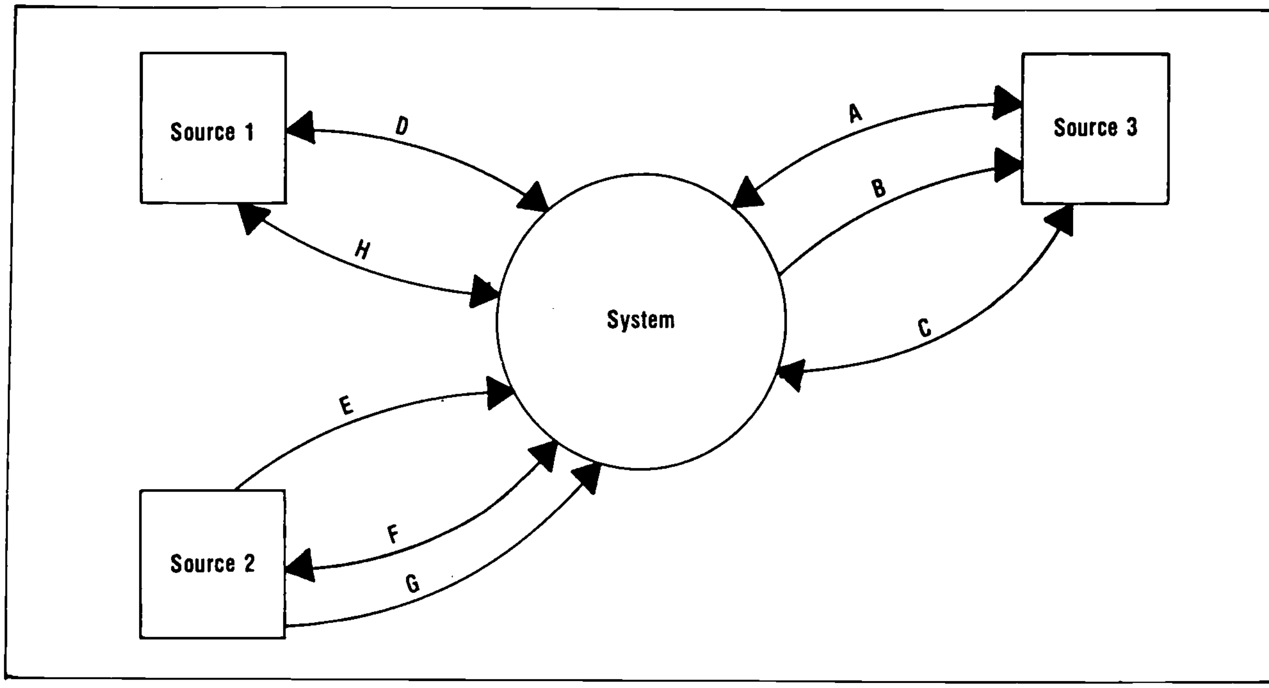
J. Data Dictionary

1. Are my definitions in alphabetical order? (Standards later).

2. Are all Data Flows and Data Stores defined?

EXAMPLE 1: CONTEXT DIAGRAM

Definition: A diagram that portrays the whole system without partitioning (it helps to define the boundaries of the study and to identify all of the interfaces the system has with the outside world).



Note 1: Source 1, 2, & 3—Represents a source, sink or terminator which are persons or organizations lying outside the context of a system.

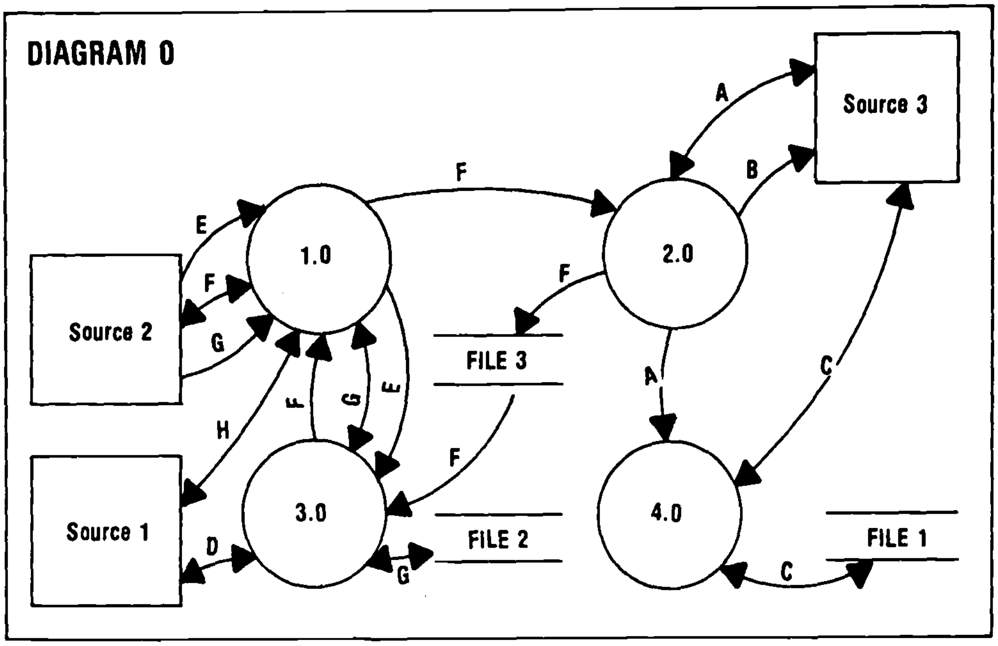
Note 2: Definition of System—A collection of activities and stored information that is organized to accomplish a specific purpose and responds to events (stimuli) in its environment in a pre-planned way.

EXAMPLE 2: Parent and Child Diagrams and the Balancing Rule.

BALANCING RULE: Each parent bubble must have exactly the equivalent input and output dataflows and data stores as the child diagram one level below it.

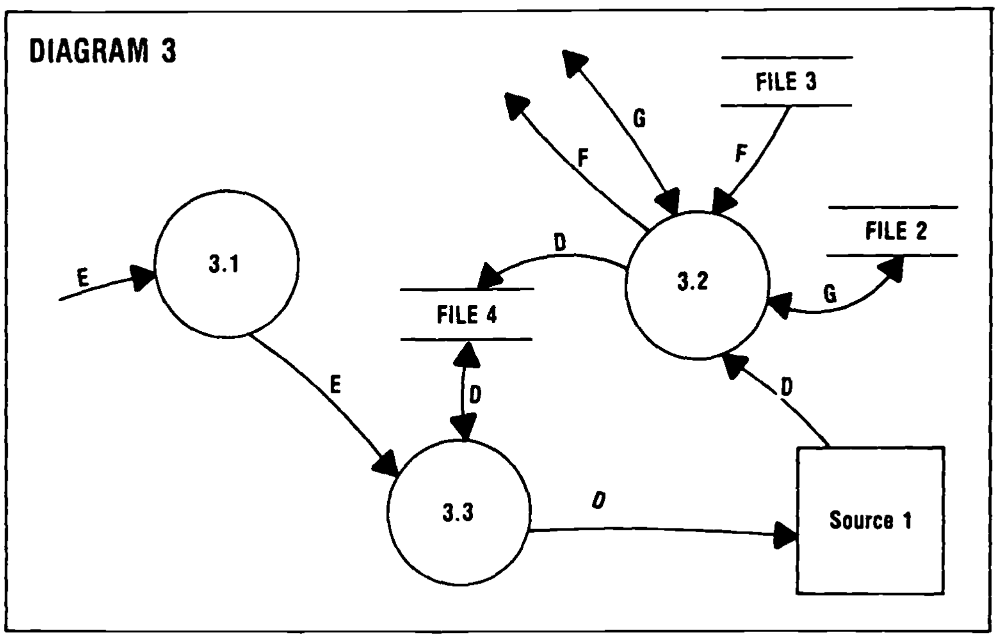
PARENT

LEVEL 1



CHILD

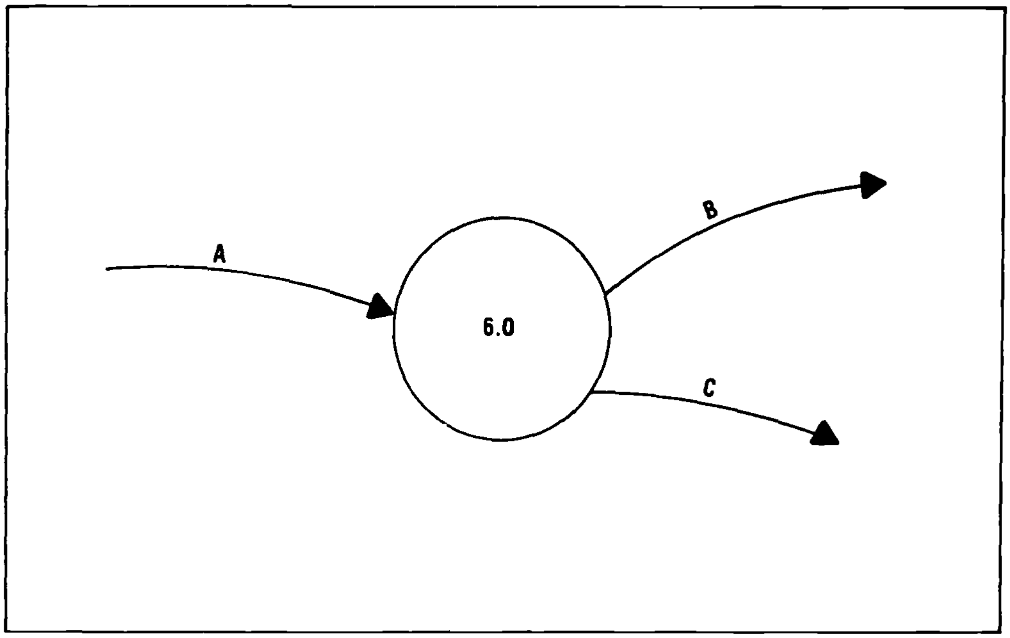
LEVEL 2



Note 1: A,B,C, etc. are used instead of names for the Data Flows.
Note 2: It is not necessary to show file 4 on Diagram 0 as it is between two processes of the Diagram 3.
Note 3: The Diagram number relates to the number in the bubble.

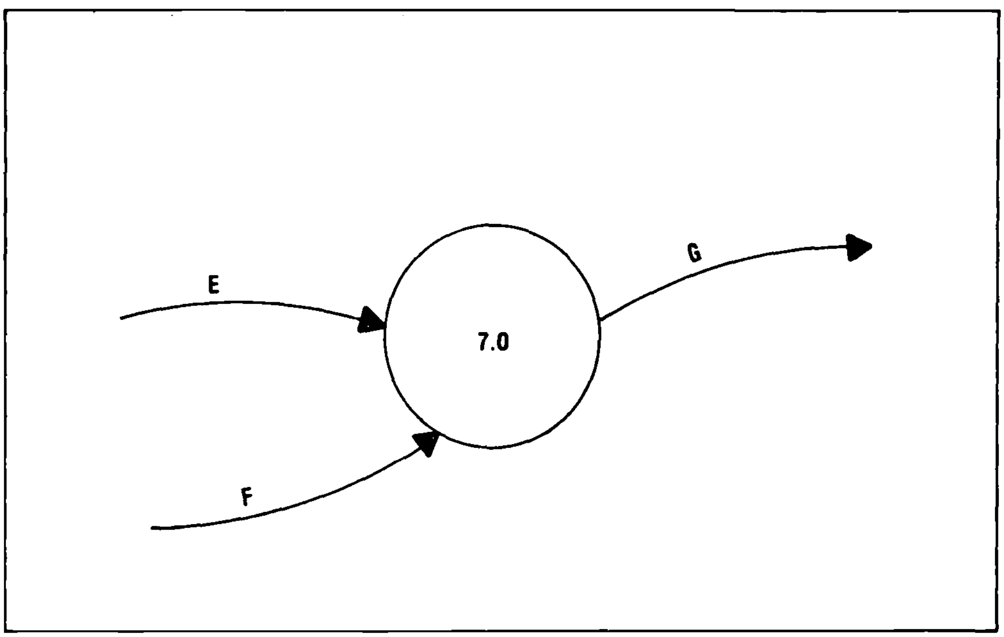
EXAMPLE 3: Equivalent Balancing.

FIGURE 1



OR

FIGURE 2



Note: Figure 1 shows an input that consists of two parts which is separated at this point and Figure 2 shows two inputs which are joined at this point.

These *must* be defined this way in the Data Dictionary.

(This page intentionally left blank)

Appendix D

GENERAL DESIGN SPECIFICATION TABLE OF CONTENTS

New Physical Model

(This page intentionally left blank)

Appendix E

GENERAL DESIGN SPECIFICATION CONTENT DESCRIPTION

SECTION 1 GENERAL

This section will document the objective, scope, and other general information required in the New Physical Model.

1.1 OBJECTIVE

This paragraph will describe the system objectives. It will include a general statement of the major purpose of the current system, identify the deficiencies of the current system in meeting the user's requirements, and state the specific objectives of the proposed system to rectify the deficiencies and meet the user's requirements. This will be in concert with the objective of the Functional Requirements Definition.

1.2 SCOPE

This paragraph will document the scope, or context, of the existing system. It will describe the existing environment affected by the deficiencies identified in 1.1, "Objective," above, with particular regard to organizational environment, site specific information, and the identification of the specific applications under study. This will be in concert with the scope as reported in the Functional Requirements Definition.

1.3 AUTHOR

This paragraph will identify the organization charged with producing the New Physical Model. In case of contracting efforts, it will include an organization chart of the producing organization with functional titles and a job description for those functional titles.

1.4 GENERAL DESIGN ACTION PLAN

This paragraph will document the findings and recommendations of the organization charged with producing the New Physical Model. The findings will present a qualitative measure of model acceptability with particular regard to:

a. Context appropriate to stated system objectives

b. Model packaged to meet performance objectives

c. Model a sound basis for starting the next activity (Detailed Design)

Based on the findings a recommended course of action will be stated whether to readdress issues raised in developing the model, through the use of prototypes or proceed with the Detailed Design.

SECTION 2 STRUCTURED SPECIFICATION

This section will document the New Physical Model. This section will be organized and formatted as documented in Appendix B, and must include the following:

- ° Data Flow Diagrams (see section 3)
- ° Data Dictionary (see section 4)
- ° Mini-specifications (see section 5)
- ° Information Model (see section 6)

SECTION 3 SUPPORTING DOCUMENTATION

This section will document information pertinent to the new system and operational environment not covered in the structured specification. Additional paragraphs may be included to document related topics as deemed necessary.

3.1. ORGANIZATIONAL CONTEXT

This paragraph will describe the organizational context within which the new system will operate. It will include organizational missions, job descriptions, and organization charts for those organizations within the context of this study.

3.2. ADPE ENVIRONMENT

This paragraph will document the targeted ADPE environment. It will include system architecture diagrams and information pertinent to hardware/software configurations.

3.3. NEW PERFORMANCE CHARACTERISTICS

This paragraph will document the anticipated performance characteristics of the new system. It will include all information pertinent to anticipated performance such as periodicity, cyclic processing requirements, response times, and throughput volume.

3.4. INFORMATION REQUIREMENTS

This paragraph should present information requirements of the New Physical Model in terms of transaction inputs and report outputs. At this point in the design, it is generally not possible to determine the physical structure and characteristics of these elements; hence a general description of content will suffice.

Once the character of data is determined in conjunction with detailed design, report formats, screen displays, and transaction layouts may be developed and presented in the Detailed Design

Specification. If the information requirements of the New Physical Model or portions thereof reflect those of the existing system, or Old Physical Model, existing formats and layouts may be presented in this document. It should be noted, however, that these examples are subject to alteration based on detailed design activities.

(This page intentionally left blank)

Appendix F

BIBLIOGRAPHY

DeMarco, Tom, Structured Analysis and System Specifications, New York, Yourdon Press, 1979.

Dickinson, Brian, Developing Structured Systems - A Methodology Using Structured Techniques, New York, Yourdon Press, 1980.

McMenamin, Stephen and Palmer, John, Essential Systems Analysis, New York, Yourdon Press, 1984.

Ward, Paul, Systems Development Without Pain: A User's Guide to Modeling Organizational Patterns, New York, Yourdon Press, 1984.

(This page intentionally left blank)

COMMENTS/REVISIONS

Technical publications under the Information Resources Management (IRM) Standards and Guidelines Program (MCO 5271.1) are reviewed annually. Your comments and/or recommendations are strongly encouraged.

IRM Tech Pub Name: ______________________________

IRM-_____-___ (Number) Date of Tech Pub: ____________________

COMMENTS/RECOMMENDATIONS: ______________________________

____

____

____

____

____

____

____

____

____

____

____

____

____

____

Name/Rank: ____ (optional)

Unit: ____ (optional)

Mail To: Commandant of the Marine Corps (Code CCID-40)
Headquarters, United States Marine Corps
Washington, D.C. 20380-0001