



ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE OF STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

A PROTOTYPE FUND ACCOUNTING SOFTWARE
FOR THE BUDGET AND FINANCE CONTROL DEPARTMENT
OF THE ADMINISTRATION FOR REFUGEE AND RETURNEE AFFAIRS

By

Mekonnen Habte

Name and Signature of Members of the Examining Board

Ato Getachew Birru, Chairman, Examining Board

A handwritten signature in black ink, appearing to read 'Getachew Birru', written over a horizontal line.

Dr. G.A Alabi, Advisor

A handwritten signature in black ink, appearing to read 'G.A Alabi', written over a horizontal line.

Ato Tesfaye Biru, Advisor

A handwritten signature in black ink, appearing to read 'Tesfaye Biru', written over a horizontal line.

Dr. G.G. Chowdhury, Internal Examiner

A handwritten signature in black ink, appearing to read 'G.G. Chowdhury', written over a horizontal line.

Prof. P. B. Mangla, External Examiner

A handwritten signature in black ink, appearing to read 'P. B. Mangla', written over a horizontal line.

ACKNOWLEDGEMENT

I owe a debt of gratitude to the accountants of the Administration for Refugee and Returnee Affairs for their cooperation in providing all information and documents I have asked for. I am also grateful to the officials of the Administration For Refugee and Returnee Affairs who supported my project by providing all types of assistance indispensable for finishing my work.

Sincere thanks to Dr. Alabi Gbade, my thesis supervisor, for reading, correcting the manuscript and for providing me valuable help. I am grateful to all faculty members of SISA. Heart felt debt of gratitude goes to Ato Hadera Admasu, deputy head of the Consular and Immigration Main Department, who provided me with a portable PC and printer which greatly facilitated the completion of the work.

I am indebted to Ato Hagos Haile Mariam, Head of the Department of Accounting at Addis Abeba Commercial College, who helped me get relief from the teaching work-load for the last two semesters. Special thanks to Betelheim Haile, for her unremitting care for me.

ACKNOWLEDGEMENT

I owe a debt of gratitude to the accountants of the Administration for Refugee and Returnee Affairs for their cooperation in providing all information and documents I have asked for. I am also grateful to the officials of the Administration For Refugee and Returnee Affairs who supported my project by providing all types of assistance indispensable for finishing my work.

Sincere thanks to Dr. Alabi Gbade, my thesis supervisor, for reading, correcting the manuscript and for providing me valuable help. I am grateful to all faculty members of SISA. Heart felt debt of gratitude goes to Ato Hadera Admasu, deputy head of the Consular and Immigration Main Department, who provided me with a portable PC and printer which greatly facilitated the completion of the work.

I am indebted to Ato Hagos Haile Mariam, Head of the Department of Accounting at Addis Abeba Commercial College, who helped me get relief from the teaching work-load for the last two semesters. Special thanks to Betelheim Haile, for her unremitting care for me.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
LIST OF DIAGRAMS	v
LIST OF APPENDICES	viii
ABSTRACT	ix

CHAPTER ONE

INTRODUCTION

1.1	STATEMENT OF THE PROBLEM AND JUSTIFICATION FOR THE STUDY.....	1
1.1.1	Project Background	1
1.1.2	Justification for the Study	5
1.1.3	Vastness of Volume of Operations	7
1.1.4	Weaknesses of the Existing Manual Methods	8
1.1.5	Structural Defects of the Existing System ...	10
1.1.6	Feasibility of the Financial System Automation and Its Expected Benefits	11
1.2	OBJECTIVES OF THE STUDY	13
1.3	LIMITATIONS OF THE STUDY	14
1.4	METHODOLOGY	14
1.4.1	DATA COLLECTIONS	14
1.4.2	SYSTEMS ANALYSIS AND DESIGN	15
	1.4.2.1 Structured Systems Development	

Methodologies.....	15
1.4.2.2 Object-Oriented Systems Development	
Methodologies	16
1.4.2.2.1 Object-Oriented Analysis ...	19
1.4.2.2.2 Object-Oriented Design	20
1.4.3 REQUIREMENT SPECIFICATION METHODS	20
1.4.4 TESTING	21

CHAPTER TWO

ANALYSIS AND DOCUMENTATION OF SYSTEM REQUIREMENTS

2.0 INTRODUCTION	23
2.1 DESCRIPTION OF THE EXISTING SYSTEM	24
2.2 SPECIFICATION OF FUNCTIONAL REQUIREMENTS OF THE NEW GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM ..	29
2.3 DATA FLOW ANALYSIS OF THE PROPOSED GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM	34
2.4 DATA DICTIONARY	38
2.5 PROCESS LOGIC	46

CHAPTER THREE

OBJECT-ORIENTED ANALYSIS OF THE FUND ACCOUNTING SYSTEM

3.0 INTRODUCTION	53
3.1 OBJECT STRUCTURE ANALYSIS	54
3.1.1 Categorizing Objects	56

3.1.2	Relationship among Objects	58
3.2	OBJECT BEHAVIOR ANALYSIS	68
3.2.1	Object States and Object State Changes	69
3.2.2	Events, Operations, and Rules	70

CHAPTER FOUR

OBJECT-ORIENTED DESIGN

4.0	INTRODUCTION	91
4.1	SPECIFICATION OF CLASSES	92
4.2	CLASS DATA STRUCTURE DESIGN	96
4.3	DESIGNING METHODS	105
4.3.1	Assigning Class Responsibilities	105
4.3.2	Creating Methods	115
4.4	DESIGNING USER INTERFACE	117
4.4.1	Print, Querying, and Updating the Database ..	118
4.4.2	End-user Direction	122
4.4.3	Maintaining Database Security and Integrity	122

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1	CONCLUSION	138
5.2	RECOMMENDATIONS	141
	BIBLIOGRAPHY	147

3.1.2 Relationship among Objects	58
3.2 OBJECT BEHAVIOR ANALYSIS	68
3.2.1 Object States and Object State Changes	69
3.2.2 Events, Operations, and Rules	70

CHAPTER FOUR

OBJECT-ORIENTED DESIGN

4.0 INTRODUCTION	91
4.1 SPECIFICATION OF CLASSES	92
4.2 CLASS DATA STRUCTURE DESIGN	96
4.3 DESIGNING METHODS	105
4.3.1 Assigning Class Responsibilities	105
4.3.2 Creating Methods	115
4.4 DESIGNING USER INTERFACE	117
4.4.1 Print, Querying, and Updating the Database	118
4.4.2 End-user Direction	122
4.4.3 Maintaining Database Security and Integrity	122

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION	138
5.2 RECOMMENDATIONS	141
BIBLIOGRAPHY	147

LIST OF DIAGRAMS

1. Context Diagram of the Budget and Finance Control System (Diagram NO 2.11)	26
2. Overview Diagram of the Budget and Finance Control System (Diagram No 2.12)	27
3. Level 1 Data Flow Diagram for the General Ledger and Financial Reporting Subsystem (Diagram No 2.31)	36
4. Level 2 Data Flow Diagram for Preparing Financial Reports (Diagram No 2.32)	37
5. Visual Table of Contents for the Proposed General Ledger and Financial Reporting Subsystem (Diagram No 2.51)	48
6. Input-Process-Output chart for Journalizing and Posting Financial Transactions (Diagram No 2.52 and 2.53)	49
7. Input-Process-Output chart for Preparing a Balance Sheet and a Cash Transfer and Expenditure Report (Diagram No 2.54 and 2.55)	50
8. Input-Process-Output chart for Preparing Site and Subproject Financial Reports (Diagram No 2.56 and 2.57)	51
9. Input-Process-Output chart for Preparing a Trial Balance and a Cash Flow Report (Diagram No 2.58 and 2.59)	52

10. Generalization Diagram for a Ledger Account Object Type (Diagram No 3.1.11)	59
11. Generalization Diagram for Financial Transaction Object Type (Diagram No 3.1.12)	60
12. Composed-of Relationship Diagrams (Diagram 3.1.21) ..	63
13. Object Relationship Diagrams (Diagram 3.1.22)	64
14. Object Relationship Diagrams (Diagram 3.1.23)	65
15. Object Relationship Diagrams (Diagram 3.1.24)	66
16. State Transition Diagrams (Diagram 3.2.11)	71
17. State Transition Diagrams (Diagram 3.2.12)	72
18. Operation Diagram for Journalizing Financial Transactions (Diagram No 3.2.21)	77
19. Operation Diagram for Posting Financial Transactions (Diagram No 3.2.22)	79
20. Operation Diagram for Generating a Balance sheet (Diagram No. 3.2.23)	81
21. Operation Diagram for Generating a Trial Balance (Diagram No 3.2.24)	83
22. Operation Diagram for Generating a Cash Transfer and Expenditure Report (Diagram No. 3.2.25)	85
23. Operation Diagram for Generating Cash Flow statement (Diagram No 3.2.26)	87
24. Operation Diagram for Generating a Project Site Financial Report (Diagram No 3.2.27)	89

10. Generalization Diagram for a Ledger Account Object Type (Diagram No 3.1.11)	59
11. Generalization Diagram for Financial Transaction Object Type (Diagram No 3.1.12)	60
12. Composed-of Relationship Diagrams (Diagram 3.1.21) ..	63
13. Object Relationship Diagrams (Diagram 3.1.22)	64
14. Object Relationship Diagrams (Diagram 3.1.23)	65
15. Object Relationship Diagrams (Diagram 3.1.24)	66
16. State Transition Diagrams (Diagram 3.2.11)	71
17. State Transition Diagrams (Diagram 3.2.12)	72
18. Operation Diagram for Journalizing Financial Transactions (Diagram No 3.2.21)	77
19. Operation Diagram for Posting Financial Transactions (Diagram No 3.2.22)	79
20. Operation Diagram for Generating a Balance sheet (Diagram No. 3.2.23)	81
21. Operation Diagram for Generating a Trial Balance (Diagram No 3.2.24)	83
22. Operation Diagram for Generating a Cash Transfer and Expenditure Report (Diagram No. 3.2.25)	85
23. Operation Diagram for Generating Cash Flow statement (Diagram No 3.2.26)	87
24. Operation Diagram for Generating a Project Site Financial Report (Diagram No 3.2.27)	89

25. Operation Diagram for Generating a Subproject	
Financial Report (Diagram No 3.2.28)	90
26. Class Hierarchies (Diagram No 4.11)	95
27. Class Communication Diagrams (Diagram No 4.31)	107
28. ARRA Fund Accounting Menus (Diagram No 4.41(A))	120
29. ARRA Fund Accounting Menus (Diagram No 4.41(B))	121
30. Cash Receipt Transaction Data Entry Form	
(Diagram No 4.42(A))	125
31. Budget Disbursement Data Entry Form	
(Diagram No 4.42(B))	126
32. Miscellaneous Disbursement Data Entry form	
(Diagram No 4.42(C))	127
33. Site Disbursement Data Entry Form	
(Diagram No 4.42(D))	128
34. General Transactions Data Entry Form	
(Diagram No 4.42(E))	129
35. Cash Transfer Data Entry Form	
(Diagram No 4.42(F))	130
36. Annual Subproject Data Entry Form	
(Diagram No 4.42(G))	131
37. Cash Flow Statement Report Form	
(Diagram No 4.43(A))	132
38. Trial Balance Report Form (Diagram No 4.43(B))	133
39. Balance Sheet Report Form (Diagram No 4.43(C))	134

40. Cash Transfer and Expenditure Report Form	
(Diagram No 4.43(D))	135
41. Project Site Financial Report Form	
(Diagram No 4.43(E))	136
42. Subproject Financial Report Form	
(Diagram No 4.43(F))	137

APPENDICES

Appendix I:	List of Persons Interviewed	151
Appendix II:	Interview Guide for ARRA's Officials	152
Appendix III :	Interview Guide for ARRA's Accountants ..	160
Appendix IV:	Interview Guide for Requirements Analysis	167
Appendix V:	Data Analysis Sheets	176

ABSTRACT

We are now living in an era known as the information age. Information has become the most important input for all types of systems established to attain certain objectives. The availability of timely, relevant, and accurate information is critical to operational efficiency and realization of organizational goals. The highly acclaimed role of information is always coupled with the use of advanced products of information technology, such as computers and computer software.

Computerization of manual information-intensive activities is now unavoidable regardless of the profession or the organization in which such tasks are performed. Accordingly, this study is undertaken to suggest a course of analysis and design activities that may be employed in computerizing the major activities of the budget and finance control department of the Administration for Refugee and Returnee Affairs office of the Ethiopian government.

The first chapter describes the project background, justifications, objectives, limitations, and methodology employed in the study. High volume of manually processed financial transactions and structural defects of the existing financial information system are the major problems that served as a justification for this study.

Fact finding tools used in this study include structured questioners, structured interviews, record examinations and observation of operations. A specification of system requirements is provided in chapter two. Data flow analysis and process analysis are employed to depict logical features of the system and to ensure the validity of functional specifications.

Object-oriented software engineering methodology is used for further analysis and design of the financial information system. Chapter three discusses object structure analysis and object behavior analysis as essential ingredients an object-oriented approach to systems analysis. Chapter four contains a description of object-oriented design methods employed in building the functional and nonfunctional structural framework of the financial application. The object-oriented design elements employed are class specifications, data structure design, methods design, and user interface design. Chapter five deals with conclusion and recommendations. Recommendations are made for acquisition of an object-oriented database for implementation of the design.

CHAPTER ONE

INTRODUCTION

1.1 STATEMENT OF THE PROBLEM AND JUSTIFICATION FOR THE STUDY

1.1.1 PROJECT BACKGROUND

Ethiopia is a signatory of the international and regional refugee conventions which demand that each government provide physical protection, material assistance, and medical care to all refugees within its territory (UNHCR 1990). The ethnic war in Somalia and the fighting between the Sudanese government and the Southern Sudanese rebels created massive influx of refugees since the late 1980's. Therefore, the Ethiopian government established ARRA to discharge its responsibilities in accordance with the international and regional refugee conventions. ARRA was officially constituted in 1989.

According to the head count carried out in March 1995, Ethiopia is a haven for 316,000 refugees. The refugees live in 15 geographically dispersed camps across the Eastern, North Eastern, and Western borders of the country. International relief organizations donate cash, food, medicines, and other materials to the refugees.

The rehabilitation of Ethiopian refugees who return to Ethiopia under the auspices of the United Nations Higher Commissioner for Refugees (UNHCR) repatriation program is another important aspect of ARRA's activity. Returnees are given a two-month food ration and transportation allowance.

According to ARRA's records, more than half a million Ethiopian refugees returned home from Somalia, Sudan, Kenya and other countries in the past five years. At present, a big program for the repatriation of an estimated 300,000 Ethiopian refugees from the Sudan is under operation .

Most of the resources donated to returnees and refugees are channelled to ARRA through the regional liaison office of UNHCR. ARRA is entrusted with the responsibility of coordinating the logistics activities and providing physical protection for the refugees. ARRA's activities are bound by a bilaterally drafted and ratified budget agreement. The budget agreement is divided into three independent sub-projects consisting of the Western zone (for Sudanese refugees), the Eastern zone (for Somali and Kenyan refugees) and the Returnee Program (for Ethiopians).

The discussion with ARRA's deputy director revealed that ARRA is fully financed by UNHCR, because Ethiopia is not financially capable of helping the refugees. See the discussion guide attached to Appendix II. The funds received by ARRA are spent exclusively for expenditure items authorized in the budget agreement. Therefore, ARRA is compelled to provide quarterly financial reports to UNHCR to explain how it has spent the donated funds. UNHCR releases funds to ARRA on quarterly basis. Funds are released for a new quarter only after receiving reports for the immediate past quarter. On top of this, ARRA submits a sub-project monitoring report for each sub-project twice a year.

ARRA is organized into four divisions: program formulation and monitoring, social services, health and nutrition, and support services. The program formulation and monitoring division drafts the annual work plan and financial plans for ARRA. The approved set of plans is then called the budget agreement. This division also prepares a semiannual project monitoring report that must be submitted to UNHCR. This report addresses all the activities performed and the problems encountered in the management of refugee programs.

The social services division is responsible for coordinating the administrative, logistics, health, and financial aspects of the field and head office operations. It is also involved in the provision of counselling and education of refugees. It encourages both refugees and returnees to engage in agriculture and other income generation activities by providing financial and material assistance.

The duties and responsibilities of the health and nutrition division include providing medical treatments to sick refugees, arranging and coordinating educational programs that promote the participation of refugees in camp sanitation activities, and supplying nourishments to refugees suffering from malnutrition problems. The security and legal protection and the inspection and auditing units are directly accountable to the director of ARRA. These units do not have subunits under them.

The support service division takes care of purchasing, warehousing, transportation, and technical matters. The budget and finance control department is also under control

of this division. The budget and finance control department within the head office of ARRA is responsible to keep ledgers and prepare reports for all financial transactions occurring at the camp level and at the head office. Therefore, the field offices send all transaction source documents to the head office. There are only two field finance offices in Jijiga and Gambella.

The project sites also have an organization chart similar to that of the head office. The refugee project sites administered by ARRA are the following:

Refugee Camps:

(a) Eastern zone

- Jijiga (zonal office)
- Hartishek
- Diredawa (coordination office)
- Derownaji
- Kebribeyah
- Teferiber
- Darror
- Rabasso
- Kamaboker
- Aisha

(b) Western zone

- Gambella (zonal office)
- Fugnido
- Dimma
- Bonga

(c) Returnee

- Addis Ababa (coordinating office)
- Country wide

1.1.2 JUSTIFICATION FOR THE STUDY

The purpose of this study is to develop and demonstrate a computer based solution for the massive manual operations of the budget and finance control department of the Administration for Refugee and Returnee Affairs (ARRA). The problem that spurred the development of a prototype fund accounting software for ARRA is the unmanageable size of manual data processing activities performed in the budget and finance control department of ARRA.

Nowadays, automation is the conventional solution to information systems afflicted with inefficient huge manual operations. As confirmed by the director's response to the interview questions (Appendix II), ARRA has decided to automate the major functions of its budget and finance control department. This study is, therefore, an attempt to assist put in place a prototype fund accounting system and pave the way for construction of a full fledged system. Any organization that contemplates automating its operations has to decide on software and hardware aspects of the new system. There is unlimited range of all types of computer hardware and peripherals from which an organization could select and purchase. But for the software aspect we have two choices: (1) buying off-the-shelf packages; or (2) contracting systems consultants or assigning company's system personnel to develop a customized software for the organization.

Accounting information systems used in business organizations and not-for-profit organizations, such as ARRA, are based on different principles. The accounting principles

and procedures employed in ARRA and other not-for-profit organizations are called fund accounting. Two classes of accounting software are available: business accounting software and fund accounting software. In general terms, business accounting software cannot be used for fund accounting; and fund accounting software can also not be used for business accounting. This is so because fund accounting and business accounting have different theoretical frameworks and reporting requirements.

Fund accounting practices are not standardized, because not-for-profit organizations do not prepare financial reports according to tax rules and generally accepted accounting principles. Record keeping and reporting methods are usually prescribed by donors. The result of absence of reporting standards is that it is truly difficult to select a fund accounting software suitable for a given type of nonbusiness organization. The examination of report formats and transaction processing procedures revealed that ARRA's reporting requirements are based on unique account numbering system and budget classifications. In addition to this, enquiries about major problems of ARRA's operations revealed that ARRA's accounting system is defective and inefficient. Therefore, prior to acquisition or development of any fund accounting software, the huge financial system need to be analyzed and designed. Because of the foregoing requirements, no fund accounting software, if at all available to meet ARRA's requirements, can be readily used to alleviate the problems. This thesis work is undertaken to produce system design specifications that will serve as the basis for evaluating the suitability of available software and customizing any chosen one for implementation. Developing a

prototype software is undertaken only to test the viability of recommendations. In the final analysis, the platform for this thesis problem is the opportunity provided by ARRA's automation requirements.

1.1.3 VASTNESS OF VOLUME OF OPERATIONS

The facts discovered in the course of exploring the relationship between ARRA and other organizations indicated that ARRA's activities are bound by a bilaterally drafted and ratified budget agreement. UNHCR releases funds on quarterly basis. Funds are released for a new quarter only after receiving reports for the immediate past quarter. The funds are to be spent exclusively for expenditure items authorized in the budget agreement.

The budget agreement is divided into three independent subprojects consisting of the Western zone (for Sudanese refugees), the Eastern zone (for Somali and Kenyan refugees) and the Returnee Program (for Ethiopians). The budget and finance control department of ARRA at the head office is fully responsible for keeping financial records and providing financial reports encompassing all head office and field level activities.

ARRA is essentially playing the role of an agency for UNHCR, because the budget agreement prescribes the budget record structure and financial report format of UNHCR to be used by ARRA. The budget agreement consists of detailed breakdown of expenditures classified by refugee subproject, by project site, by sector, and by sub sector.

The budget ledger structure and the report format are geared to the budget structure. Consequently, a separate ledger is maintained for each subproject and camp, likewise a separate financial report is prepared for each subproject and each camp.

The study of UNHCR's budget manual showed that UNHCR's standardized budget structure has two basic components: sectors and budget items. There are 17 major sectors which are divided into 168 sectoral activities. There are 19 budget items that have 250 subitems. Hence each budgetary expenditure account number indicates the sector, the subsector, the budget item, and the subitem codes, such as A.22.C.05010. A minimum of 3000 transactions occur in a month. The coded transactions are ordinarily posted manually to 1200 accounts and quarterly financial reports are manually prepared with the help of a spread sheet software.

1.1.4 WEAKNESSES OF THE EXISTING MANUAL METHODS

The extensive dependence of the existing system on manual procedures has resulted in numerous mistakes and contributed much to the delay of reports. According to the accountants, the department could not cope with the daily routine of budget disbursements transactions and budget enquiries. On the one hand, the transactions are usually fully processed and reports prepared for issuance on their due dates. On the other hand, the accountants have to take care of the administrative requirements of recurring disbursement transactions.

On the basis of discussions, observations, and responses to questioners, a summary of the shortcomings of the existing manual budget ledger maintenance and report generation operations is provided below:

- Manual posting to 1200 accounts takes too much time;
- Retrieving the account balances manually from the 1200 accounts has been awfully cumbersome;
- The manual sorting and summarizing of the retrieved account balances for generation of around 68 different types of reports has been a formidable task;
- The manual procedures have become so taxing that no account has been kept up-to-date and no readily usable budgetary expenditure information was provided to management for routine control of expenditures and performance evaluation;
- The inability to deliver timely reports has adversely affected the flow of funds from UNHCR to ARRA. Report delays resulted in delay of fund transfers. This in turn affected the assistance provided to the refugees. Therefore, viewed from another angle, report delays created problems for the refugees who depend exclusively on UNHCR assistance.

The preponderance of manual processing requirements of budgetary transactions is further compounded by a manual payroll and provident fund subsystem. The major function of this subsystem is to prepare a monthly payroll register and to maintain around 300 provident fund accounts for ARRA

employees. In describing the problems of the existing fund accounting system, the accountants also stated that provident fund balances have never been up-to-date because of the time required to make monthly manual posting to 300 employee accounts.

All these problems hindered ARRA's capability to meet the requirements for the provision of timely, accurate, relevant, and complete financial information for satisfying the needs of internal and external users. In conclusion, the replacement of the existing manual fund accounting information system by a properly designed computerized system is indispensable for solving the major operational problems of the budget and finance control department of ARRA.

1.1.5 STRUCTURAL DEFECTS OF THE EXISTING SYSTEM

The findings resulting from evaluation of the structure of the system and views expressed by all concerned parties confirmed that the existing system is deficient. For example, the whole financial record keeping and reporting effort in the system is narrowly focused on cash expenditure transactions. The other transactions that do not involve immediate cash collections or payments are totally neglected. The accounting system does not incorporate a formal system of records and methods for handling receivables and payables or noncash transactions. The consequence of the unattended parables is that creditors' claims have become hard to corroborate and settle on time. Many times payments are made only to appease the creditors who persistently send bills. As far as receivables are concerned no one knows how much money is owed to ARRA by external parties.

Because of the absence of a functional system framework for record keeping and reporting of noncash transaction events, ARRA's accountants are not able to summarize the financial records and clear them on annual basis. Consequently, ARRA's financial records have never been examined by independent auditors. The omission of periodic review of records has an unfavorable effect on the reliability of the financial system in preventing misuse of funds and safeguarding financial resources. Hence, such grave system defects deserve a systems analysis and design work of large scale.

1.1.6 FEASIBILITY OF THE FINANCIAL SYSTEM AUTOMATION AND ITS EXPECTED BENEFITS

This project is expected to be feasible and viable. Some of the operational, technical, and financial factors that contribute toward the feasibility of its implementation are the following:

- full support by management of ARRA;
- full support by users, especially, the accountants who were consistently asking for computerization to get some relief from tedious clerical jobs;
- the employees in the finance department are trained and experienced in the use of application software;
- the financing for the purchase of additional computers has already been taken care of and the cost of 14 computers and one server has been fully paid.
- the delivery of the computers has just been completed;

The successful completion of this project would benefit two parties: ARRA and UNHCR. The following benefits shall be realized when the completed application system is implemented by ARRA:

- (1) The submission of timely reports will solve the major complaints of UNHCR officials. This facilitates free flow of quarterly budget allotments.
- (2) Accuracy of reports and records shall be achieved, because the automated system eliminates almost all errors that appear in the manual system.
- (3) The addition of new system functions for controlling receivables and payables corrects structural problems of the existing system. The new functions facilitate the preparation of complete year-end financial reports and supply all necessary data for closing records on annual basis. In summary, financial records will be complete, properly organized, and fit for auditing.
- (4) The increase in processing speed will solve lack of up-to-date information that hampered budget monitoring activities. Hence, excessive expenditures will be controlled. The budget formulation and monitoring department of ARRA will be provided with monthly expenditure reports essential for evaluating field activities. This in turn will increase efficiency of operations and provision of more beneficial services to the refugees.

1.2 OBJECTIVES OF THE STUDY

(A) General Objectives

The general objective of this project are :

- (1) To design an automated transaction processing and report generation system for the UNHCR-fund based budget disbursement activities of ARRA;
- (2) To lay a structural and functional foundation for further development work on producing a complete operational fund accounting system;
- (3) To test the suitability of an object-oriented software development methodology for analysis and design of a fund accounting information system

(B) Specific Objectives;

In order to achieve the general objectives of this study, the project is focused on the following specific objectives:

- (1) to develop both functional and nonfunctional software specifications
- (2) to design the different kinds of databases to be maintained by the accounting information system;
- (3) to design program and module structures required to handle the functional and nonfunctional as well as database requirements;
- (4) to test cohesiveness and coupling of the program structure and evaluation of software modules;
- (5) to carry out testing runs for software evaluation;

1.3 LIMITATIONS OF THE STUDY

This study will not incorporate the following components due to time and know-how constraints:

- (1) installing the system at the users place and conducting acceptance testing
- (2) designing test cases
- (3) simulation modelling
- (4) user manual and training documents
- (5) software reliability tests
- (6) direct implementation of the designed system in any of object oriented languages such as Smalltalk, C++, Objective - C, Eiffel, or Simula.
- (7) pure object oriented program codes
- (8) implementation schedule

1.4 METHODOLOGY

In order to carry out the analysis and design of the prototype fund accounting information system, the following methodologies and tools were employed.

1.4.1 DATA COLLECTION

All popular fact finding methods for data collection were used in this study. Structured interviews, questionnaires, record review, and process observations were conducted to investigate the fund accounting system of ARRA. I have intensively used structured interviews for gathering information from the accountants. I have interviewed all the four accountants working at the head office and all the two

accountants working at Jijiga field office. All financial transactions are recorded and reports are produced at the head office. Three of the five officials of ARRA were interviewed to get information about ARRAs objectives and operations. The officials interviewed were the deputy director and the heads of two major divisions, namely program formulation and monitoring and social services. The interview guides, questionnaires, and the names of all persons interviewed, and the data analysis sheets are attached to appendices.

1.4.2 SYSTEMS ANALYSIS AND DESIGN

The systems analysis and design methodologies available for specifying requirements and for analyzing and designing information systems could be categorized into two groups: structured methodologies and object-oriented methodologies.

1.4.2.1 Structured Systems Development Methodologies:

Structured methodologies are composed of complementary methods, tools, and techniques applicable to both the analysis and design phases of information systems development. Hence, several structured systems development methodologies are composed of a cohesive set of structured analysis and structured design methods. All structured analysis and design methodologies follow a top-down decomposition approach during the analysis and design phases of the systems study.

The partitioning strategy improves comprehensibility of complex systems and simplifies the system modelling process. Structured methodologies give much attention to user

participation and employ graphical tools. The graphical tools used, especially during the analysis phase, such as data flow diagrams and HIPO (Hierarchy plus Input-Process-Output) charts are believed to be more understandable by the users' than the system flowcharts used during the nonstructured systems development era. Data flow diagrams and a variety of structure charts are used to analyze the movement of data and identify the system processes that manipulate the data. The data flow diagrams, structure charts, and HIPO charts represent abstract models of the problem domain under study. Refined versions of these diagrams produced in a long iterative drawing process are presented to the user as major requirements specification documents. These methodologies also use a data dictionary to document the results of data analysis. The design phase employs variants of data flow diagrams and structure charts that depict programming logic and modular structure of the system.

Some of the well known structured methodologies are the following:

- (a) Structured Systems Analysis and Design Method (SSADM)
- (b) DeMarco's Structured Analysis and Systems Specification
- (c) Jackson Systems Development
- (d) Structured Analysis and Design Technique (SADT)
- (e) Structured Analysis and Design: The Yourdon and Constantine Approach

1.4.2.2 Object-Oriented Systems Development Methodologies

Object-oriented analysis and design are fundamentally different from structured analysis and design. Object-oriented analysis and design methodologies emerged after the introduction of object oriented-programming in the early

80's. Object-oriented programming languages provided a built-in syntax to describe a module and the associated processing specifications as intrinsic ingredients. This necessitated a new analysis and design methodology compatible with object-oriented implementation.

The degree by of an information system model is closer to representation of system features from a user's perspective diminishes in the process of transition from requirements description to analysis, from analysis to design, and from design to implementation. This statement describes the characteristics of structured approaches to systems analysis and design. During analysis data flow diagrams are used to show both the processes and the data movements in one model; however, more than one model of the system is created during the design phase. An entity-relationship model is used to specify the data structure for the database.

On the other hand, processing requirements and control features are represented by structure charts which a user is not able to understand without expertise in programming. The user is, therefore, not in a position to prevent any misrepresentation of requirements that may be introduced during the transformation of the analysis models to design models and the design models to implementation models.

Object-oriented analysis and design methodologies describe system elements as objects at both the systems analysis and design phases. An object is anything that models things in the real world. These things could be physical entities such as production machines or abstract entities

such as expenditure accounts. In object-oriented analysis and design, a system model is built of direct abstractions of real world objects that have properties and behaviors. The objects are named using the problem domain language. For example, for the financial system under study, the objects are called by the names of the various accounts and reports. During programme coding, the software is divided into modules that represent those same objects used during analysis and design.

The fundamental achievement of the object-oriented approach is that it reduces the semantic gap between the models we use in computer programming languages and database management systems and the conceptual models of real world objects (Martin 1993). Object-oriented approach is chosen for this study, because it creates models of the application area in a way that is closer to reality than conventional structured analysis.

System models are built of objects during the analysis phase. These same models are used during design and implementation with additional refinements. In conventional structured design and programming approach, a paradigm shift occurs as we go from analysis to design and from design to implementation.

Object-oriented design results in the creation of software modules that incorporate both information that describe the object and processing applicable to the object. According to Pressman (1987), the unique nature of object-oriented design lies in its ability to build upon three important software design concepts: abstraction, information

hiding, and modularity. All structured design methodologies strive for software that exhibits these fundamental principles, but only object-oriented design provides a mechanism that enables the designer to achieve all three without complexity and compromise.

Several information system domain experts have evolved relatively different object-oriented analysis and design methodologies since the mid 1980's. The following authors are among well known experts that proposed their own methodologies:

- (a) Peter Coad and Edward Yourdon
- (b) James Martin
- (c) Ivar Jakobson
- (d) Mark Lorenze
- (e) Grady Booch

I have adopted an object-oriented analysis and design methodology suggested by James Martin because of the clarity of its modelling techniques and simplicity of diagramming methods.

1.4.2.2.1 Object-Oriented Analysis

Object-oriented analysis is composed of two major activities: object structure analysis and object behavior analysis. In an object-oriented software system objects are considered as actors that perform system functions, because each object incorporates a specification of operations or methods applicable to its own data structure. In order to study how system functions are carried out, investigation are made about the nature and type of relationships that link the

objects . Object-oriented modelling is employed to describe static functional relationships and dynamic relationships between objects of different types.

The behavior of an object refers to the range of operations an object performs in response to messages sent to it. This means the functionality of a system is analyzed in terms of what operations are performed by objects and what results are attained. The system results are normally described as object state changes - changes in values of object attributes. The events that mark object state changes are normally recorded by the object. The results of object-oriented analysis provide the basis for designing the data structures and methods of objects.

1.4.2.2.2 Object-Oriented Design

Object-oriented programming languages have relieved the designer from worrying much about coupling and cohesion of system modules. Object-oriented languages provide a software structure called class in which a programmer packs a data structure and operations. Therefore, an object-oriented system designer establishes a mechanism for the following: the representation of class data structure, the specification of class methods, and the invocation of methods. In summary, object-oriented design involves class structure design, methods design, and messages design.

1.4.3 REQUIREMENTS SPECIFICATION METHODS

A complete specification of software requirements is essential to the success of a software development effort. A software requirements specification is produced as a result

of the analysis task. Requirements are specified on the basis of the analysis of data collected from the accountants and higher officials of ARRA, and from the results of financial record reviews and process observations. A requirements specification furnishes narrative and graphical explanations about information flow, information content, information processing, and information structure of the proposed system. Therefore, chapter two of this study includes a narrative description of the input, processing, and output requirements of the current and proposed general ledger and financial reporting subsystem of ARRA's fund accounting system. A data dictionary is also used to describe the content of the information that flows between processes of the system being studied.

It is necessary to have complete knowledge about all aspects of a system under study prior to preparation of its requirements specification. Graphical tools are, therefore, indispensable for depicting important relationships among system components and for enhancing the analyst's understanding. In this respect, data flow diagrams and HIPO diagrams are useful supplementary graphical tools that can be used in the analysis phase for expediting the creation of an object-oriented requirements model of the system. Hence, data flow diagrams and HIPO diagrams are included in chapter two of this work as preliminary analysis tools. Data stores could indicate the appropriate objects and their respective data structures or attributes. Data flows and process descriptions provide a good platform for identifying methods and events.

Several writers of object-oriented software engineering recommended that a requirements model could be created by an

iterative process involving extraction of nouns, adjectives, and verbs from the English language descriptions of the system. It is presumed that nouns, adjectives, and verbs represent objects, their attributes, and operations respectively. This approach seems highly subjective, because terms and statements could carry more than one meaning.

1.4.4 Testing

The prototype fund accounting software was tested using the following elements of software testing:

- a. Module testing
- b. Bottom up integration testing
- c. Program inspection by peers
- d. Improvised test cases for module testing and integration testing,

CHAPTER TWO

ANALYSIS AND DOCUMENTATION OF SYSTEM REQUIREMENTS

2.0 INTRODUCTION

The purpose of requirements analysis is to determine the nature of the user's problem. The problem statements discussed in chapter one are formulated after long interviews, questioners, and record examinations have been used for discovering the users' problems.

The information obtained in course of the interviews, discussions, and observations could not be presented in conventional (tabular) form because of the small size of the population. However, the data gathered have been analysed and consolidated to reveal the state-of-the-art of the existing fund accounting system at ARRA.

Requirements are specific descriptions of functions or characteristics that address the more general purpose of the system. In other words, a requirement addresses the purpose of the system without regard for how the system is implemented. Hence, implementation specific descriptions are not considered to be requirements.

Based on analysis of the problems faced by the internal and external users of the fund accounting system, a draft requirements definition or specification document was prepared and presented to the chief accountant. The

specifications were then reviewed jointly with the chief accountant and the other members of the budget and financial control department of ARRA. Amendments had been made before the final specification of requirements were prepared.

Requirements describe systems behavior. Requirements are of two types: functional requirements, and nonfunctional requirements. Functional requirements focus on input, process, and output aspects of the system. This chapter describes functional requirement specifications for the general ledger and financial reporting subsystem of ARRA's fund accounting system. These requirements are implementation independent. The nonfunctional requirements describe a restriction on the system that limits ways and means of solving the problem and implementing the system. Hardware specification is an example of the non functional requirements. The user interface design presented in chapter four is a major example of the nonfunctional requirements.

2.1 DESCRIPTION OF THE EXISTING SYSTEM

Analysis of the existing manual system was made as a prerequisite for designing a computerized financial data processing system for ARRA's budget and finance control department. Computerized accounting systems are mirror reflections of their manual counterparts because similar logical processing steps are employed in both manual and computerized accounting systems.

The context diagram that portrays data sources, sinks, and boundaries of the existing budget and finance control

system of ARRA is depicted in diagram 2.11 on page 26. The functional breakdowns of the existing budget and finance control system are depicted in the overview diagram 2.12 on page 27. A brief description of input, processing, and output requirements of the existing system is provided below:

Major Classes of Inputs:

- Budget allocation data from UNHCR
- Budget disbursement transaction data
- Data about budget fund releases by UNHCR
- Data about internal movement of funds from head office to field finance offices
- Employee payroll and provident fund transaction details

Processing:

- verifying the accuracy and authenticity of all types of source documents and vouchers
- analyzing transactions of all types and specifying the various debit and credit components that make up each transaction
- recording the transactions in the appropriate journals and ledger accounts

Output:

- reconciliation of bank statement
- a quarterly subproject monitoring financial report for each project site and subproject.
- a monthly income tax deduction and payment report
- a monthly provident fund deposit report

DIAGRAM 2.11 CONTEXT DIAGRAM OF THE FUND ACCOUNTING FINANCIAL SYSTEM

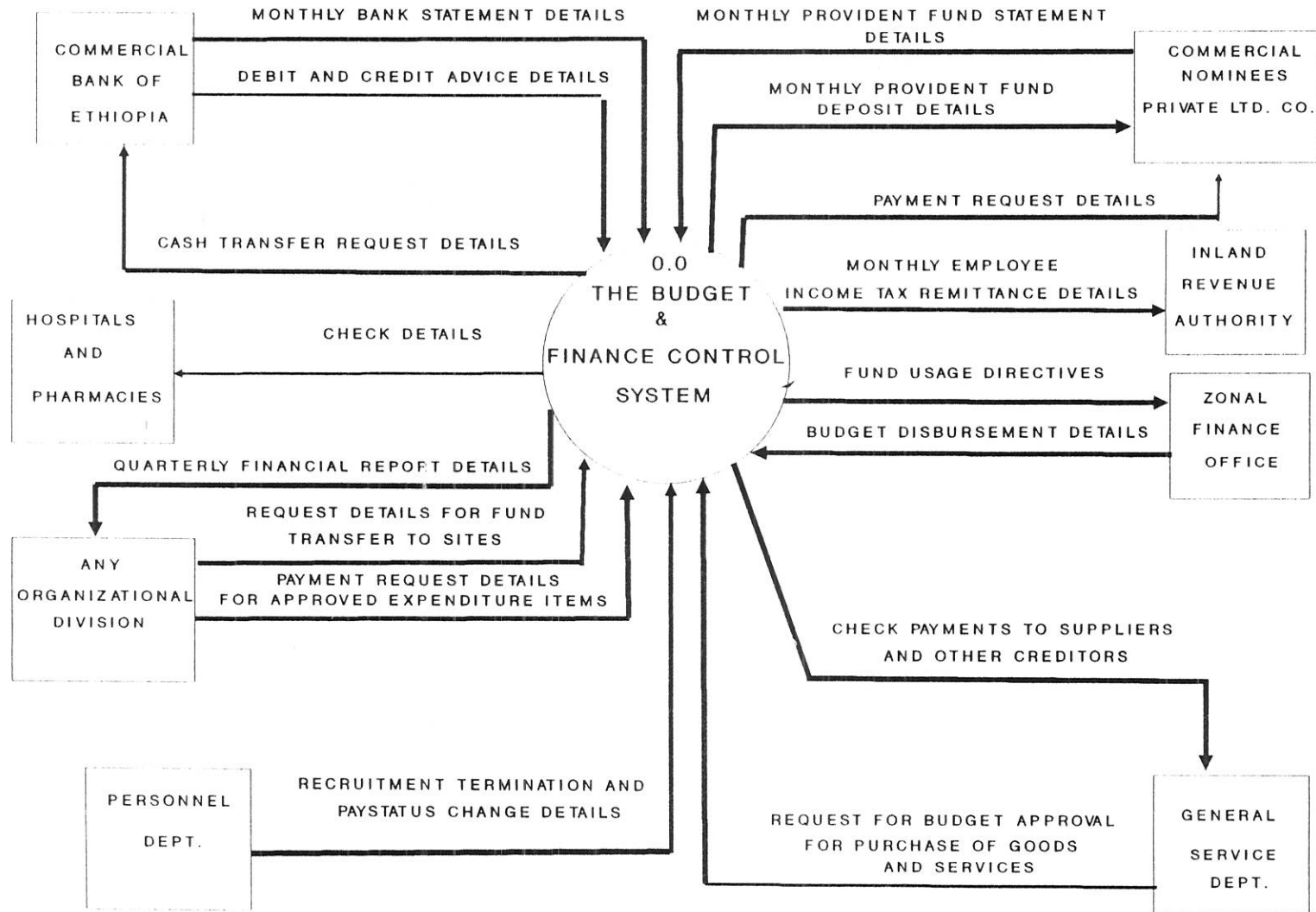
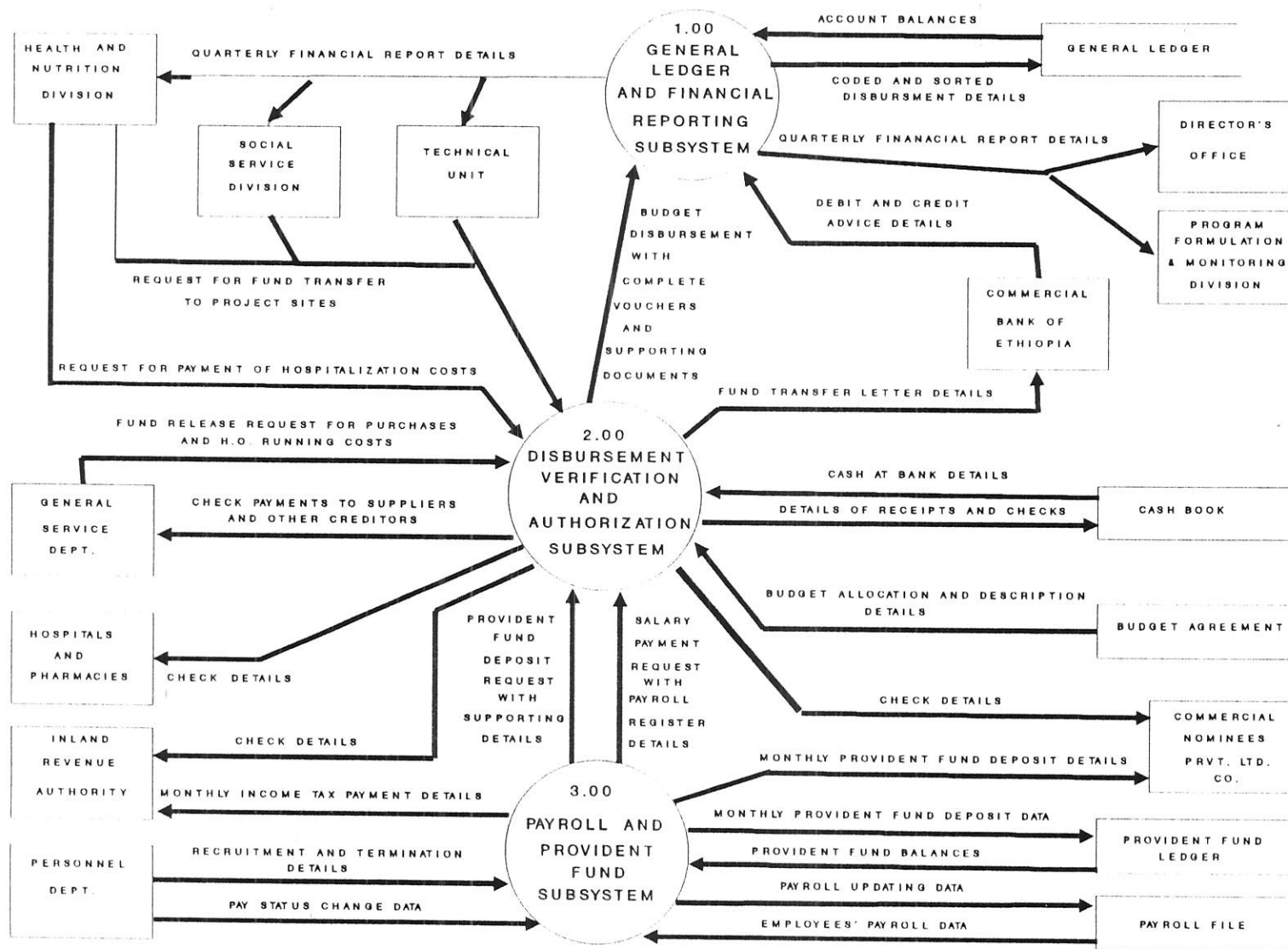


DIAGRAM 2.12 OVERVIEW OF THE EXISTING BUDGET AND FINANCE CONTROL SYSTEM



As the record keeping and reporting requirements are mainly directed towards the fulfillment of the demands imposed by the financiers of the refugee projects, the existing fund accounting system has been tuned to discharging only these requirements. The summary below shows these requirements.

Record keeping requirements:

<u>Subproject</u>	<u>No of refugee camps</u>	<u>No of ledgers</u>
Eastern Subproject	10	10
Western Subproject	4	4
Returnee Program	<u>1</u>	<u>1</u>
Total	<u>14</u>	<u>14</u>

Financial reporting requirement:

<u>Subproject</u>	<u>No of quarterly financial reports per year</u>
Eastern	44
Western	20
Returnee	<u>4</u>
Total	<u>68</u>

The process described as verification and authorization of budget disbursements involves manual administrative and control procedures for approving disbursement requests and verifying supporting documents. The other two major processes require record keeping and report generation. The chief accountants and the other members of the budget and finance control department have described the general ledger and financial reporting subsystem as a priority area to be studied for automation purposes. The scope of this thesis,

therefore, is limited to automation of only the general ledger and financial reports subsystem. Therefore, the requirements of the new general ledger and financial reports subsystem are presented in detail.

2.2 SPECIFICATION OF FUNCTIONAL REQUIREMENTS OF THE NEW GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM

The requirements of the new general ledger and financial reporting subsystem are produced after all types of documents handled by the budget and finance control department has been scrutinized and the operational personnel has been interview. The data analysis sheets are presented in Appendix V.

Input requirements:

The sources of input data for the general ledger subsystem originate from both internal and external sources. Ordinarily, transaction source documents are prepared by purveyors of goods and services. Therefore, basic financial raw data are drawn from source documents received from external parties. Additional transaction authorization documents are always internally generated for all transaction.

External source documents:

- utility bills
- payment receipts
- purchase invoices
- various contractual documents and related payment requests for services rendered
- debit and credit advices received from banks.

Internal source documents:

- authorized payment request
- receipt voucher
- cash disbursement voucher
- bank disbursement voucher
- cash transfer documents
- budget agreement
- balance sheet chart of accounts
- temporary payment slip
- travel advance clearance
- warehouse and purchases requisition
- goods receiving note
- journal voucher
- payroll register
- daily laborers wage payment list
- checks
- refugee payment list

The input data elements contained in the above source documents are defined in the data dictionary presented in the data analysis sheets in Appendix V.

Processing requirement:

Processing requirements consist of rules and procedures applied to the inputs.

- interactive data entry and query facilities are needed
- transactions must be classified into well recognized categories as described in the next section
- transactions are processed only when valid internal source documents are available

Transaction category	Principal source documents
cash receipts	cash receipt voucher
budget disbursements	cash payment voucher and bank disbursement voucher
liability incurrence	journal voucher
materials procurement	goods receiving note
fund transfer	fund transfer letter
purchases advance	purchases requisition

- each transaction is analyzed and decomposed into its debit and credit elements
- the debit and credit elements are assigned proper and descriptive account titles and numerical code numbers
- account title and number designations are always based on the balance sheet chart of accounts and the budget agreement descriptions
- each balance sheet account will have a five digit numerical code number
- each budget expenditure account will have a nine character alphanumeric code number
- for any transaction the total dollar amount of the debited accounts must equal the total dollar amount of the credited accounts
- a ledger account has a normal balance designated as a debit balance or a credit balance
- each account has a separate record that is used to store its up-to-date balance resulting from past transactions and events
- normal balances of accounts serve as a frame of reference for analytical decomposition of a transaction into its debit and credit ingredients

- no account is allowed to have an abnormal balance
- a chronologically arranged summary record of transactions should be kept for each class of transactions
- classified transaction summaries should be further sorted by account types and posted, as needed, to the general ledger accounts to determine current account balances
- a debit transaction amount of an account is an addition to the account balance if the account has a normal debit balance
- a credit transaction amount of an account is an addition to the account balance if the account has a normal credit balance
- the total of the balances of accounts with normal debit balances should always be equal to the total of the balances of accounts with normal credit balances
- a screen display or a hard copy of all transaction files (journals) and master file (general ledger) must be available for inspection at any time
- the system must contain detailed information at an expenditure account level about budget allocations and fund transfers to project sites
- the budget balance of an expenditure account must be computed as a difference between the account balance and the annual budget of the account and be reported in a site or subproject financial report
- a budget disbursement transaction is always associated with a particular project site and subproject.

- the budget expenditure accounts in the general ledger are sorted and summarized by budget sector order and the balance sheet accounts are arranged in numerical order for preparing a trial balance
- a subproject budget monitoring report (quarterly financial report) must be prepared by sorting expenditures by project site order, sectoral order, and account number order respectively.
- no new budget expenditure account can be used except those described in the budget agreement, but new asset or liability accounts can be added to the chart of accounts when new asset and liability entities emerge in the normal course of conducting financial activities
- payment and receipt transactions should be easily accessible to prepare a cash flow statement

Output requirements:

The output requirements include four types of new financial reports and the existing quarterly financial reports. The four new financial reports are needed to fulfil internal report requirements necessary for close control of fund usage and accuracy of the general ledger and financial reporting subsystem.

- a subproject financial performance report is prepared every quarter in 10 copies for distribution within and outside ARRA
- similarly, according to the requirements, project site financial performance report is prepared every quarter in 10 copies

- a balance sheet must be prepared at the end of each month and five copies of such balance sheet need to be generated for exclusive use by the accountants
- a trial balance of the general ledger is produced once every week and at least two hard copies should be produced
- two copies of a monthly statement of cash flows are needed for use by the chief accountant and person assigned for preparing a statement of bank reconciliation
- a statement that compares actual budget disbursements with funds dispatched to projects sites is also prepared on monthly basis for each project site in at least two copies
- the report formats for the new and existing reports are depicted in the user interface design section of chapter four.

2.3 DATA FLOW ANALYSIS OF THE PROPOSED GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM

The general ledger and financial reporting subsystem is the most important subsystem, because it incorporates all transaction analysis and report generation operations except payroll processing. Therefore, the budget and finance control department has recommended the aforementioned subsystem as a priority area for this short study. Data flow analysis of the general ledger and financial reporting subsystem is conducted to document the relationship between system processes and data. The logical descriptions of the system depicted by data flow diagrams and HIPO (Hierarchy Input

Process Output) charts furnish complete view of processing requirements of the proposed automated system. The context diagram of the budget and finance control department depicted in diagram 2.11 describes the data sources and sinks of the fund accounting system. In order to describe the major functions and relationships, an overview data flow diagram is shown in diagram 2.12. In this diagram, the boundaries of the general ledger and financial reporting subsystem and its relationship with other subsystems are shown.

Level 1 data flow diagram for the general ledger subsystem:

The major functions of the proposed automated general ledger and financial reporting subsystem are depicted in diagram 2.31 on page 36:

- 1.10 Journalizing financial transactions
- 1.20 Posting financial transactions
- 1.30 Preparing financial reports

Level 2 data flow diagram for process 1.30:

The largest of the four major processes of this subsystem, process 1.3, is decomposed further into six lowest level processes as shown in diagram 2.32 on page 37:

- 1.31 Preparing end of month balance sheet
- 1.32 Preparing a cash transfer and budget expenditure report
- 1.33 Preparing a site budget performance report
- 1.34 Preparing a subproject budget performance report
- 1.35 Preparing a cash flow statement
- 1.36 Preparing a weekly trial balance

DIAGRAM 2.31: LEVEL 1 DFD THE GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM

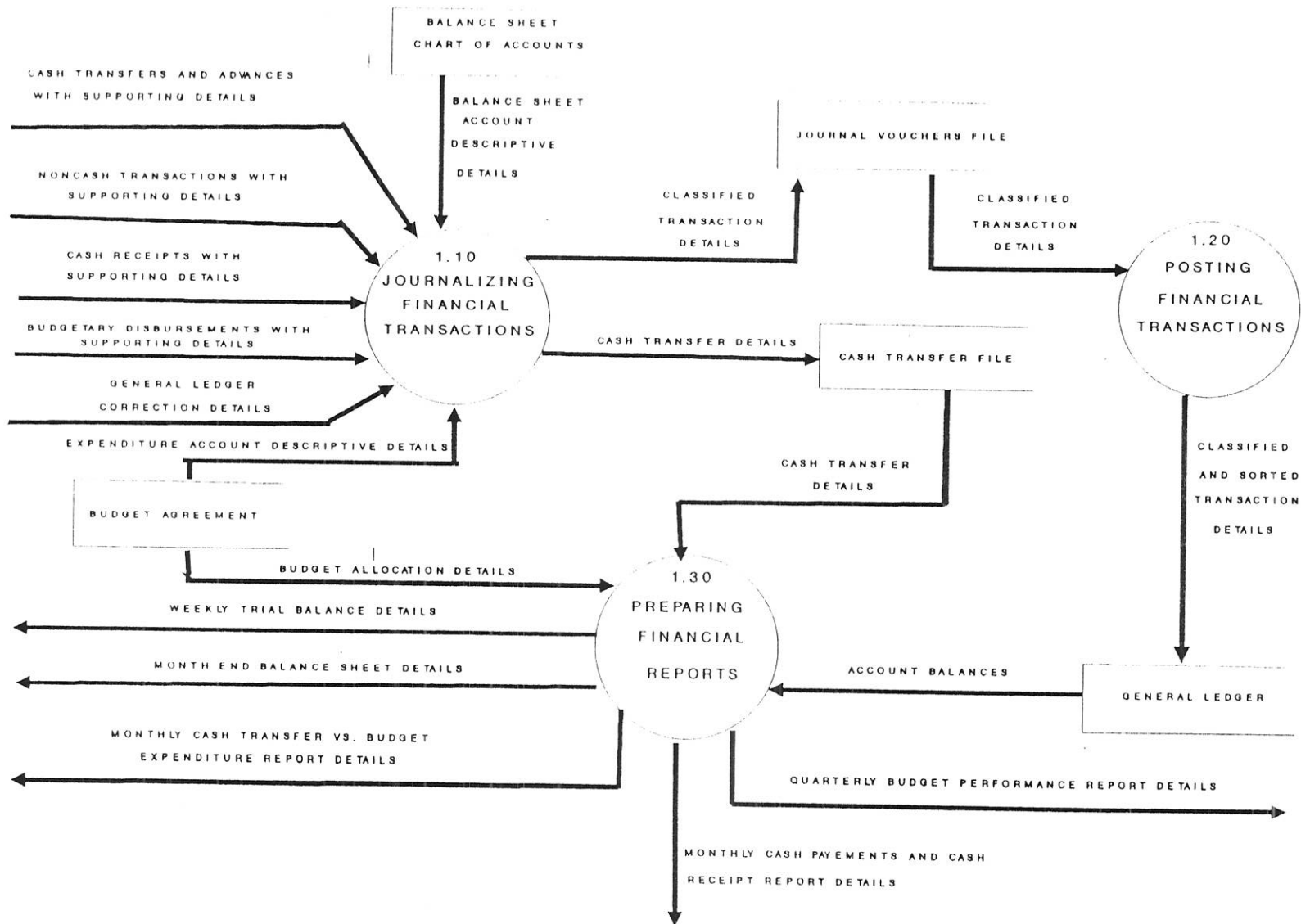
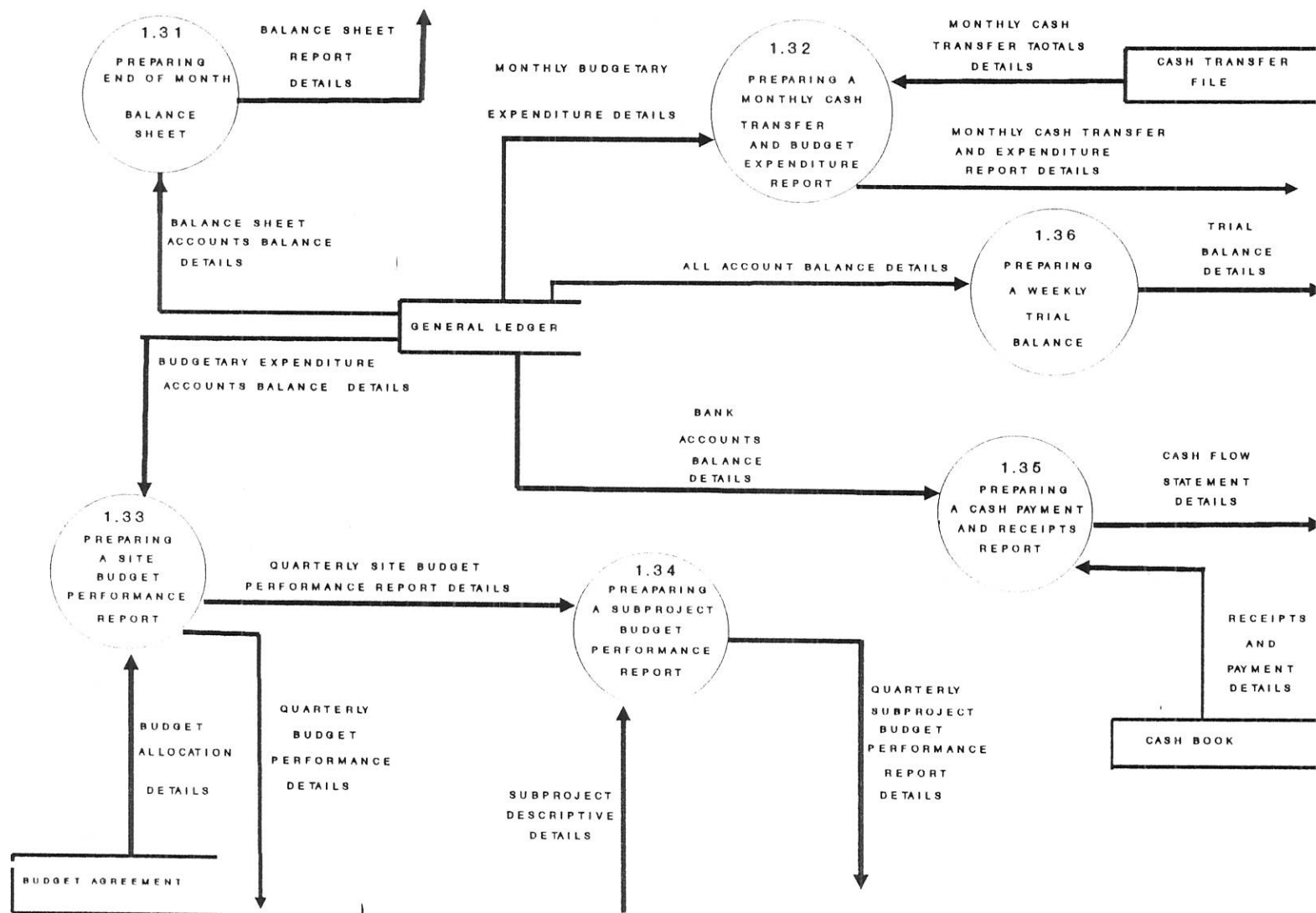


DIAGRAM 2.32: LEVEL 2 DFD FOR PREPARING FINANCIAL REPORTS



2.4 THE DATA DICTIONARY

The data dictionary is a list of all the data elements included in a set of data flow diagrams describing a system. The major elements in a system are data flow, data stores, and procedures. A data analysis of the whole budget and finance control system was carried out in order to accurately identify the data requirements of the subsystem selected for this study (See Appendix V). The data and process analysis of the proposed general ledger and financial reporting subsystem resulted in the identification of, the data elements to be captured, stored, processed, and reported as attributes of documents and reports. A complete description of data elements that will be used in the new automated general ledger and financial reporting subsystem of the fund accounting information system is provided below:

- (1) Subproject Symbol: Char 19

mask NN/XX/ETH/XX/NNN(X)

Where the first 'NN' are calendar year digits and the last 'NNN' are digits assigned for the project as a code number;

the two 'XX' symbols are alphabetical subproject identification codes.

- (2) Subproject Zone: Char 8

Description of geographic location of a subproject. It assumes only three values: East, West, and Returnee.

- (3) Subproject Description: Char 75

Subproject Title

- (4) Quarterly Reporting Period: Char 23
mask dd/mm/yyyy - dd/mm/yyyy
The span of time between the first and last date of the quarterly reporting period
- (5) Effective Commencement Date: Char 10
mask dd/mm/yyyy
The first date from which a subproject budget agreement begins binding upon the parties
- (6) Planned Termination Date: Char 10
mask dd/mm/yyyy
The last date of a budget fiscal period on which the budget agreement expires
- (7) Liquidation Date: Char 10
mask dd/mm/yyyy
A date on which the responsibility for payment of subproject liabilities expires
- (8) Annual Subproject Budget: Num 8
Budget assigned by UNHCR for a particular subproject. It is equal to the sum of budgeted subitem expenditures for all project sites in a zone.
- (9) Subproject Duration: Char 19
mask dd/mm/yyyy - dd/mm/yyyy
Time between the effective commencement date and the planned termination date of a subproject.
- (10) Subproject Cumulative Disbursement To Date: Num 8
Computed as a sum of the zonal project sites' cumulative disbursements to date
- (11) Subproject Budget Balance: Num 8
Computed as a difference between the annual subproject budget and subproject cumulative disbursement to date

- (12) Project Site Description: Char 20
Subproject location name
- (13) Annual Project Site Budget: Num 8
Budget assigned by UNHCR for a particular project site;
it is the sum of expenditure budgeted at account level
for a particular project site
- (14) Project Site Cumulative Disbursement To Date: Num 8
Computed as a sum of the budget t expenditure account
balances for a particular project site
- (15) Project Site Budget Balance: Num 8
The difference between project site cumulative
disbursement to date and project suite annual budget
- (16) Budget Sector Code: Char 1
A sectoral classification code that takes all letters
from 'A' to 'P'; and 'X'
- (17) Budget Sector Title: Char 40
It is the a short description given to a budget sector
classification or sectoral category
- (18) Sectoral Budget Cumulative Disbursement to date: Num 7
Computed as a sum of the up to date balances of budget
disbursement accounts that belong to a particular sector
in a project site ledger
- (19) Annual Sectoral Budget: Num 7
Project site annual budget is a consolidated figure that
represents the sum of annual budget of disbursement
accounts of a particular budget sector
- (20) Sectoral Budget Balance: Num 7
The difference between sectoral cumulative disbursement
to date and sectoral annual budget of particular
project site.

- (21) Budget Expenditure Account Code: Char 12
mask X:NN:X:NNNNN
Where the first 'X' indicates the budget sector code;
'NN' represents the sectoral budget breakdown numerical
code; the second 'X' designates an alphabetical budget
expenditure item category code; and 'NNNNN' is a
numerical code for a particular budget expenditure item
(also called subitem).
- (22) Budget Expenditure Account Description: Char 50
Budget Category Subitem Description.
- (23) Annual Budget of a Budget Expenditure Account: Num 7
Budget assigned to each budget item within a particular
budget sector and for a particular project site
- (24) Monthly Fund Transfer Per Expenditure Account: Num 7
Budget fund transferred to a project site on monthly
basis with detailed budget item breakdowns.
- (25) Budget Exp. Acct. Cumulative Disbursement To date: Num 8
Computed as a sum of all the amounts of individual
budgetary disbursement transactions affecting a
particular budget expenditure account of a project site
- (26) Budget Exp. Account Disbursement Total for the First
Month of the Quarter: Num 7
Computed as a sum of all the amounts of individual
disbursement transactions affecting a particular budget
expenditure account for first month of quarterly
reporting period
- (27) Budget Expenditure Account Disbursement Total for the
Second Month of the Quarter: Num 7
Computed as a sum of all the amounts of individual
disbursement transactions affecting a particular budget
expenditure account for second month of quarterly
reporting period

- (28) Budget Expenditure Account Disbursement Total for the Third Month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions affecting a particular budget expenditure account for third month of quarterly reporting period
- (29) Cash Transfer and Actual Disbursements Monthly Variance of a Budget Expenditure Account: Num 5
Computed as a difference between fund transfer and total actual disbursements of a budget expenditure account for the current month
- (30) Budget Expenditure Account Budget balance: Num 7
Computed as the difference between cumulative disbursement to date and annual budget amount of a budget expenditure account of a project site
- (31) Balance Sheet Account Number: Num 5
mask NNNNN
where the first 'N' is balance sheet account type classification code number; the second 'N' is code number of a balance sheet account type subdivision; the last 'NNNN' is the account number assigned to particular account within a given subdivision
- (32) Balance Sheet Account Description: Char 30
Account title of a balance sheet account
- (33) Transaction Date: Char 10
mask dd/mm/yyyy
A date of occurrence of a particular transaction
- (34) Transaction Reference Code: Char 3
An abbreviated code that designates the official principal source document of a transaction.

<u>Specific Values</u>	<u>Meaning</u>
CK	Check
RT	Receipt
LT	Letter
WPR	Warehouse and Purchase Requisition
GRN	Goods Receiving Note
JV	Journal Voucher
BDV	Bank Disbursement Voucher
CPV	Cash Disbursement Voucher

(35) Transaction Reference Code Number: Num 8

A sequential identification number assigned to official principal source documents of transactions

(36) Transaction Amount: Num 7

The monetary value of a particular transaction that affects two or more accounts

(37) Transaction Description: Char 15

A short description of the type of financial transaction that has occurred

(38) Amount Debited or credited to An account: Num 7

The amount that must be recorded as a debit to a particular account as a result of the occurrence of a particular financial transaction

(39) Total Debit Amount of a Transaction: Num 7

Computed as the sum of the individual debits related to two or more accounts as a result of a particular transaction

(40) Total Credit Amount of a Transaction: Num 7

Computed as the sum of the individual credits related to two or more accounts as a result of a particular transaction

- (41) Account Balance: Num 7
Computed as the difference between the total of the debit transaction amounts and the total of the credit transaction amounts posted to an account
- (42) Budget Sector Breakdown: Num 3
mask X.NN
Where 'X' indicates the alphabetical budget sector code; 'NN' indicates budget sector category breakdown code digits
- (43) Budget Sector Breakdown Description: Char 50
A description of activities related to sectoral breakdown
- (44) Budget Item Category Code: Char 1
It is alphabetical code for a budget item category. Its value takes any letter from 'a' to 'o' including 'z'.
- (45) Budget Item Category Description: Char 40
A short descriptive title given to a budget item category
- (46) Budget Subitem Code: Num 5
A numerical code given for a budget item category breakdown
- (47) Budget Subitem Description: Char
Budget expenditure account description
- (48) Subproject Budget Disbursement Total for the first month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of project sites in a particular subproject for first month of quarterly reporting period

- (49) Subproject Budget Disbursement Total for the second month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of project sites in a particular subproject for second month of quarterly reporting period
- (50) Subproject Budget Disbursement Total for the third month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of a project sites in a particular subproject for third month of quarterly reporting period
- (51) Project Site Budget Disbursement Total for the first month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of a project site for first month of quarterly reporting period
- (52) Project Site Budget Disbursement Total for the second month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of a project site for second month of quarterly reporting period
- (53) Project Site Budget Disbursement Total for the third month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions of a project site for third month of quarterly reporting period

- (54) Sectoral Budget Disbursement Total for the first month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions affecting budget disbursement accounts within a particular sector in a project site ledger for first month of quarterly reporting period
- (55) Sectoral Budget Disbursement Total for the second month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions affecting budget disbursement accounts within a particular sector in a project site ledger for second month of quarterly reporting period
- (56) Sectoral Budget Disbursement Total for the third month of the Quarter: Num 7
Computed as a sum of all the amounts of individual disbursement transactions affecting budget disbursement accounts within a particular sector in a project site ledger for third month of quarterly reporting period
- (57) User Name: Char 15
- (58) User Password: Char 8

2.5 Process Logic:

Hierarchy charts and HIPO

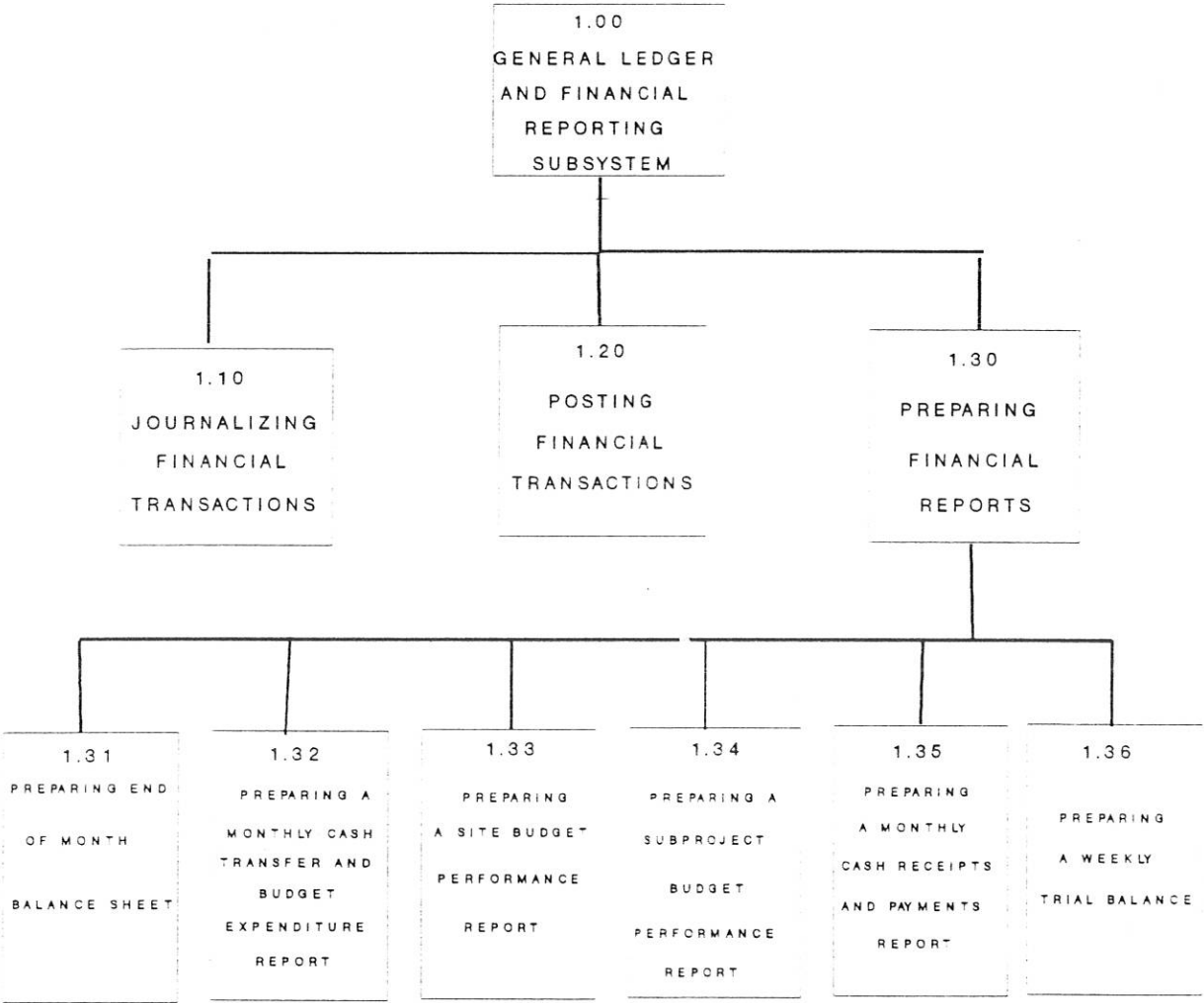
Although DFDs depict all the data flows and processes in a system, they do not explain exactly what detail procedures are carried out in each process in getting an output. This is accomplished by using another structured design tool called Hierarchy input output (HIPO) diagrams. A HIPO diagram shows how an input to a system is transformed to an output. The documentation of the process specification for the general

ledger subsystem of the fund accounting system is composed of a visual table of contents and detailed HIPO diagrams.

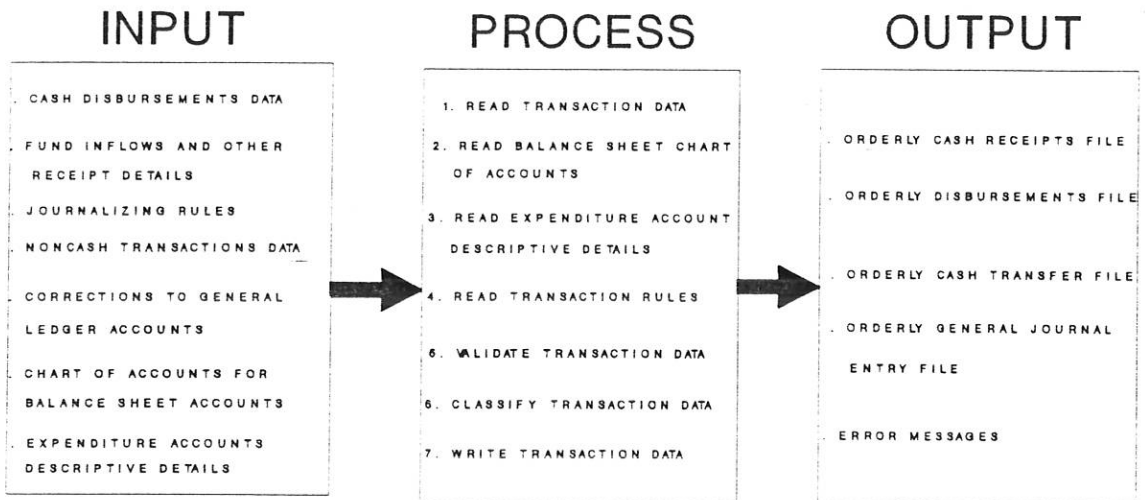
Visual table of contents (hierarchy charts): The VTOC that shows the structure of HIPO diagram package used in documenting the functions of the general ledger and financial reporting subsystem is presented in diagram 2.51 on page 40.

Detailed HIPO diagrams (IPO diagrams): These diagrams show actual inputs, outputs, and processes with detailed layout of data and processing logic. Eight detailed diagrams prepared for the VTOC of the general ledger and financial reporting subsystem are presented in the following pages.

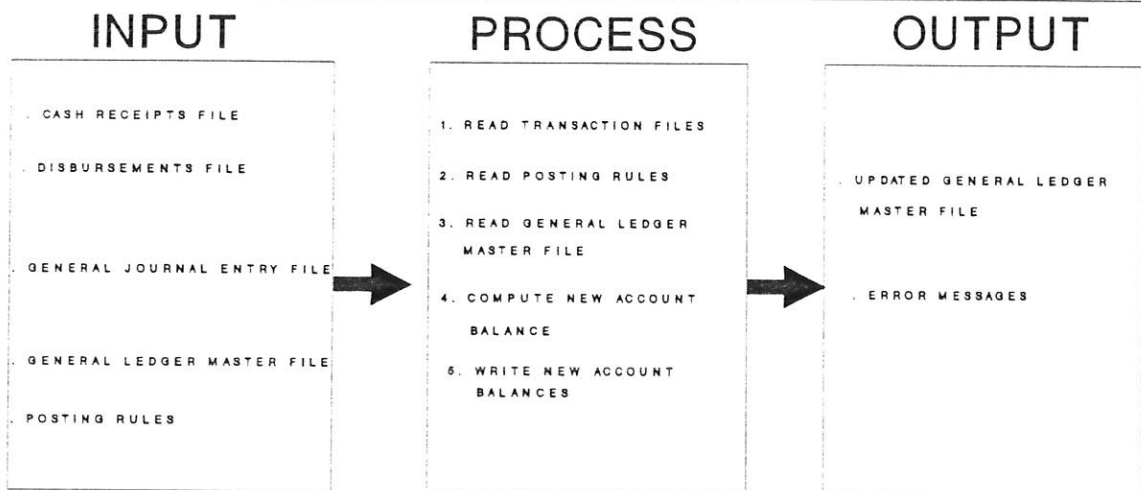
DIAGRAM 2.51 VTOC FOR THE PROPOSED GENERAL LEDGER
AND FINANCIAL REPORTING SUBSYSTEM



SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.52: IPO CHART 1.10
 DESCRIPTION: JOURNALIZING FINANCIAL TRANSACTIONS



SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.53: IPO CHART 1.20
 DESCRIPTION: POSTING FINANCIAL TRANSACTIONS

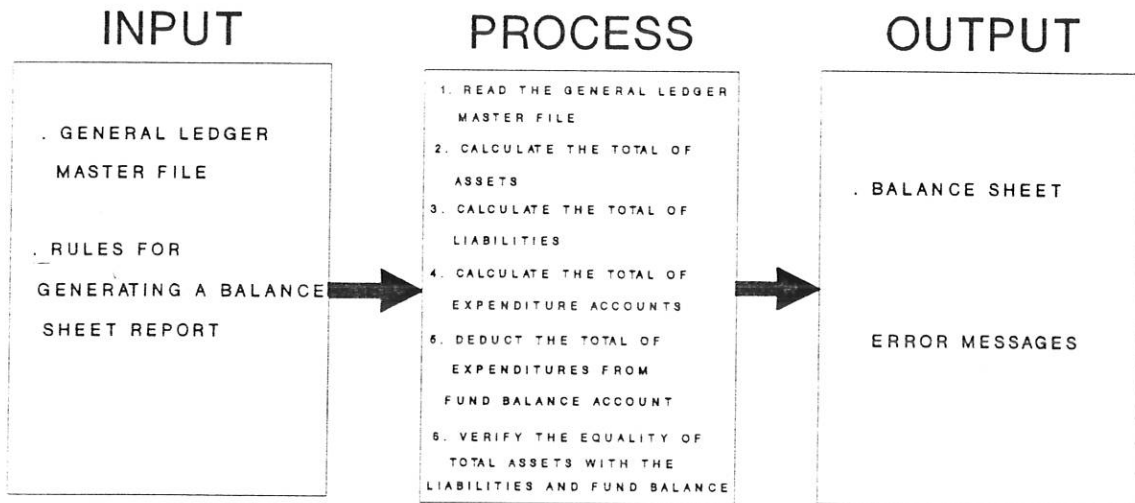


SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM

SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM

DIAGRAM 2.54: IPO CHART 1.31

DESCRIPTION: PREPARING END OF MONTH BALANCE SHEET

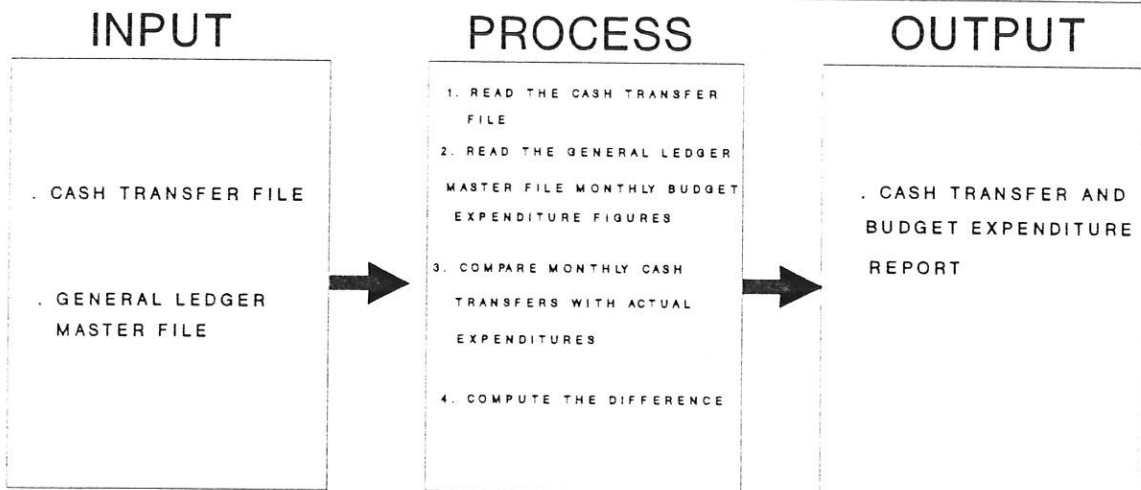


SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM

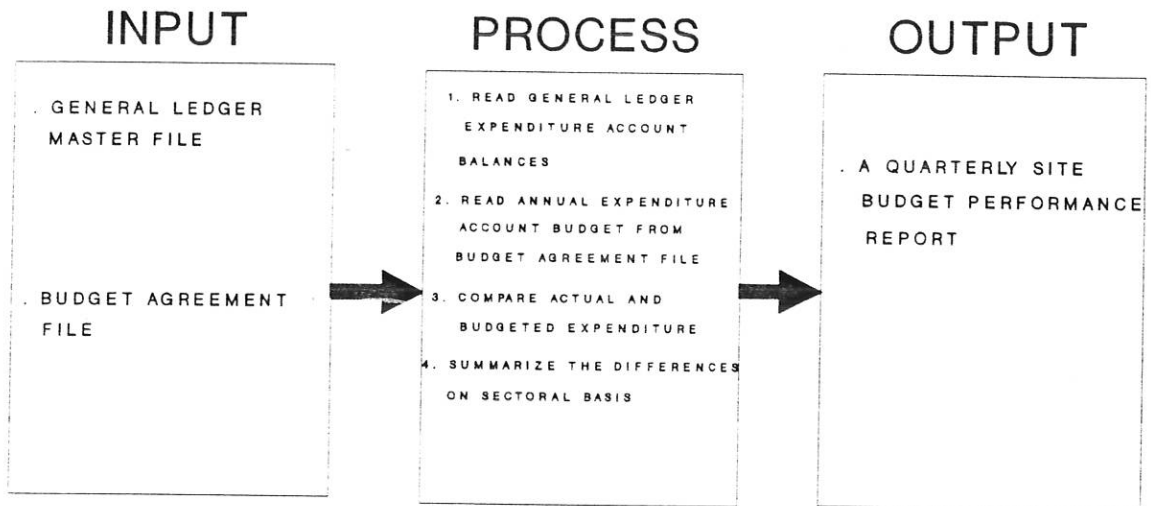
SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM

DIAGRAM 2.55: IPO CHART 1.32

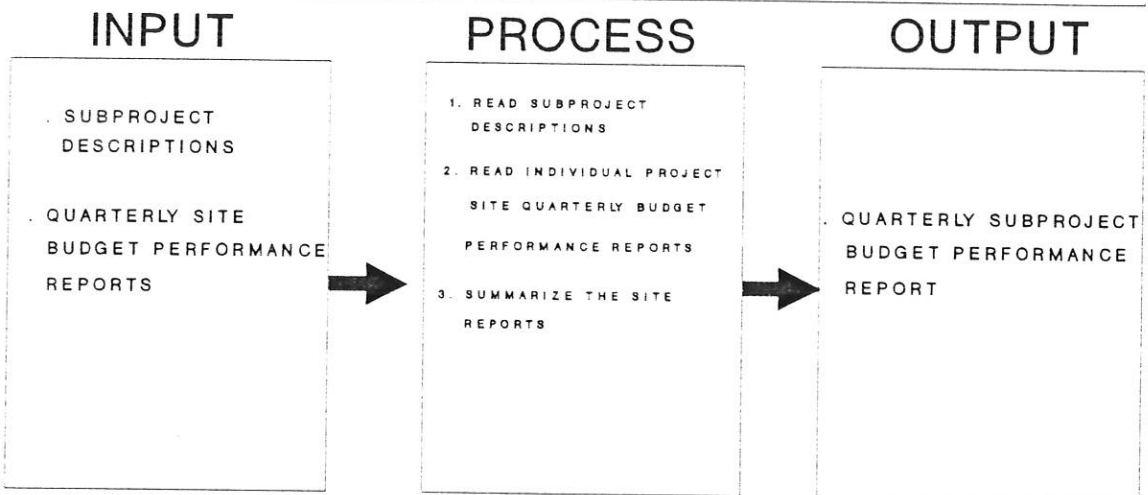
DESCRIPTION: PREPARING A MONTHLY CASH TRANSFER AND EXPENDITURE REPORT



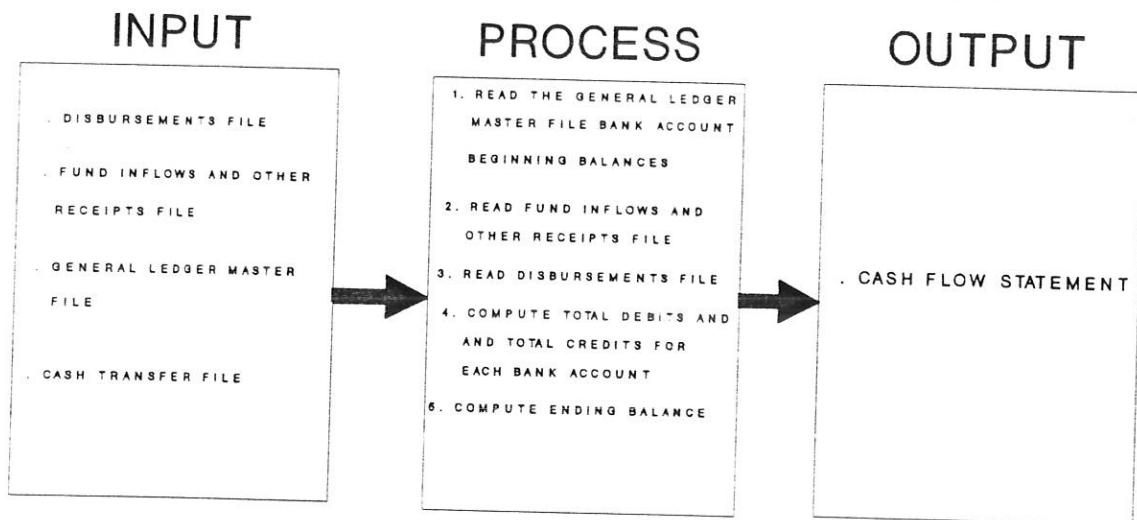
SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.56: IPO CHART 1.33
 DESCRIPTION: PREPARING A SITE BUDGET PERFORMANCE REPORT



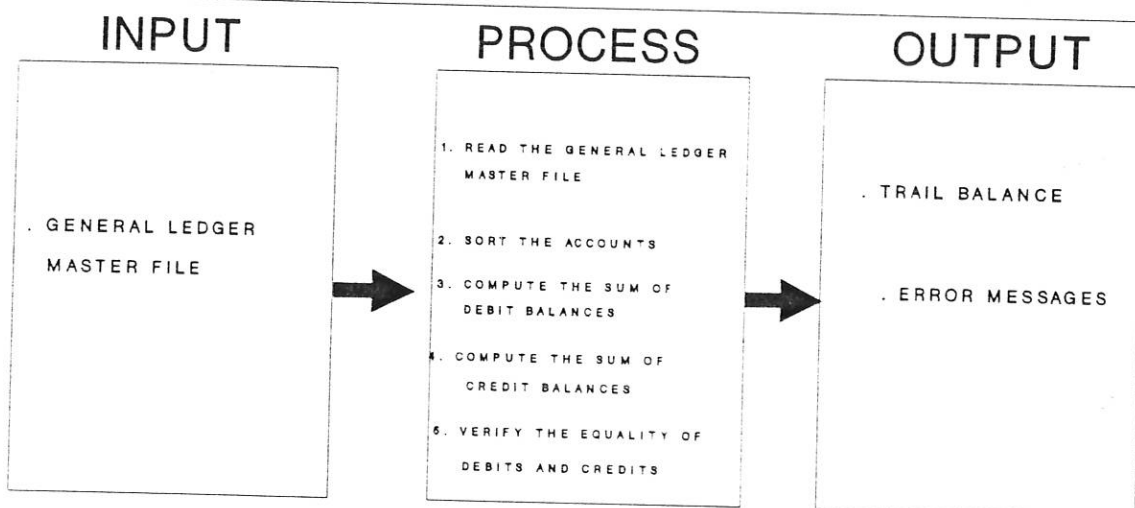
SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.57: IPO CHART 1.34
 DESCRIPTION: PREPARING A SUBPROJECT BUDGET PERFORMANCE REPORT



SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.58: IPO CHART 1.35
 DESCRIPTION: PREPARING A CASH FLOW REPORT



SYSTEM: ARRA FUND ACCOUNTING INFORMATION SYSTEM
 SUBSYSTEM: GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM
 DIAGRAM 2.58: IPO CHART 1.36
 DESCRIPTION: PREPARING A TRIAL BALANCE



CHAPTER THREE

OBJECT-ORIENTED ANALYSIS OF THE FUND ACCOUNTING SYSTEM

3. 0 INTRODUCTION

Object-oriented information systems began to appear in the late 80s. However, they became widely available and gained popularity as recently as the early 90s. Object-oriented software engineering is a relatively new software technology, therefore its development methodologies are not yet refined and standardized. A novice developer faces a problem in identifying and choosing a methodology and the associated techniques for developing object oriented applications. I have adopted, with some modification, analysis and design methodologies and diagramming techniques recommended by James Martin (1993).

Martin elucidates the principles of object-oriented analysis and design by dividing the development process into 'two equal halves'. The first half of the process involves analyzing the structure and behavior of objects. The second half of the process involves designing class structures and methods. The analysis phase is described in chapter three and the design phase is demonstrated in chapter four.

The object-oriented approach to software engineering is founded on the following key concepts: object, class, instance, message, method, inheritance, encapsulation, polymorphism, dynamic binding, and static binding. The descriptions of these concepts are elaborated in the relevant sections of this thesis.

CHAPTER THREE

OBJECT-ORIENTED ANALYSIS OF THE FUND ACCOUNTING SYSTEM

3. 0- INTRODUCTION

Object-oriented information systems began to appear in the late 80s. However, they became widely available and gained popularity as recently as the early 90s. Object-oriented software engineering is a relatively new software technology, therefore its development methodologies are not yet refined and standardized. A novice developer faces a problem in identifying and choosing a methodology and the associated techniques for developing object oriented applications. I have adopted, with some modification, analysis and design methodologies and diagramming techniques recommended by James Martin (1993).

Martin elucidates the principles of object-oriented analysis and design by dividing the development process into 'two equal halves'. The first half of the process involves analyzing the structure and behavior of objects. The second half of the process involves designing class structures and methods. The analysis phase is described in chapter three and the design phase is demonstrated in chapter four.

The object-oriented approach to software engineering is founded on the following key concepts: object, class, instance, message, method, inheritance, encapsulation, polymorphism, dynamic binding, and static binding. The descriptions of these concepts are elaborated in the relevant sections of this thesis.

The specification of systems requirements must precede any object oriented analysis and design activity. The conventional data flow diagrams and process specifications based on HIPO charts are still indispensable to capture functional requirements of a system. Without these traditional tools, a developer cannot have a clear and comprehensive view of the application domain. In addition, the relational database design is compulsory for an objected oriented application that will be implemented by a semi-object-oriented or pure relational data base management system. Consequently, the work presented in chapter two is believed to have laid a foundation for proceeding with the object oriented analysis and design of the fund accounting application.

3.1 OBJECT STRUCTURE ANALYSIS

Objects are real or abstract things and processes about which an information system is concerned in tracking down or generating information. This definition, however, equates the concept of objects with that of entities. From an object-oriented perspective, the concept of an object also embodies not only data types or attributes but also those operations that manipulate the stored data. Consequently, an object is a self-sustaining software module, a collection of data types and all necessary functions or procedures that work on the object's data structure. Software functions and procedures are designed and coded to manipulate a specific object's data exclusively. In an object oriented language, such exclusive subroutines or functions are called methods.

Because an object holds and processes its own data, other software modules of the system will not have direct access to its internal structure. The concept of encapsulation is used to describe the way software elements are structured as objects in an object oriented system. Such structuring of object-oriented software modules greatly reduces module coupling and enhances module cohesiveness. Program testing and maintenance are believed to be simplified as a result of encapsulation.

In the analysis stage of an object-oriented systems development project, object-oriented models of the application are created. These models portray a natural way of classifying and grouping objects within the application environment. Object structure analysis is done to identify and classify the objects of the application domain. Problem domain objects could be well recognized and differentiated from one another based on review of requirements specifications. Problem domain objects are objects that have a direct counterpart in real world problem domain under consideration.

The problem domain objects of the proposed general ledger fund accounting system are: subproject, project site, ledger account, cash transfer subsidiary ledger account, budget sector, budget sector category, budget item, budget subitem, subproject budget, project site budget, sectoral budget, expenditure account budget, financial transaction, balance sheet, trial balance, cash flow statement, cash transfer versus budget expenditure report, quarterly project site financial report, and quarterly subproject financial report.

3.1.1 CATEGORIZING OBJECTS

Object structure analysis entails investigating the similarities and differences among objects in terms of their data structures and methods, i.e., processing activities applied to these data structures. Such analysis enables the categorization of problem domain objects into various classes. The object structure analysis results in the creation of classes for similar objects that have common operations and data structures. A class represents a template for several objects and describes how these objects are structured internally (Ivar Jacobson 1992). A class is essentially an abstract representation of an object type. Each category or group of objects, such as a ledger account, is designated as a class, and may also be referred to as an object type. Each specific object such as a particular asset account is then an instance of the account object type or class. Extracting the common data and data handling processes of similar objects and designing and building a reusable modular software is the essence of object-oriented software engineering. Accordingly, the analysis of object structure begins with the identification of broad relationships among similar objects and proceeds to determine associations among classes of different objects.

In the fund accounting system, for instance, all transaction journal entries have a data structure composed of account number, transaction reference, and a debit or a credit amount. Some transactions are different from others in terms of the type of data they handle. Therefore, the transactions class may be further categorized into two or

more specialized subclasses depending on the type of data handled. The categorization process creates a hierarchy of classes. But the subtypes or subclasses of transaction objects still have the same data structure and transaction analysis methods declared for the parent transaction class. The point of the discussion is that once a data structure and methods are designed for a particular class, they are used by all classes down in the class hierarchy. This is called inheritance.

Inheritance is simply a way of reusing methods and data. Inheritance eliminates the need for analyzing and designing a complete new method or computer program for handling minor variations. This saves development time and increases productivity. Inheritance is one of the strengths of object oriented software engineering. Hence problem domain objects of a given type inherit operations and data structure of that class. The Inheritance relationships between object types and subtypes are depicted in a box diagram consisting of boxes within boxes. The box diagram is also called generalization diagram.

An object type is fundamentally different from an entity type in three ways. First, objects encapsulate both data and operations on data. Entities are stored without being linked to any particular application. Second, Objects contain data structures that may be very complex, because objects encapsulate data and operations on data. Entities are composed only of attributes. Third, objects permit inheritance relationships because object classes are reusable by newly declared subtypes.

The fund accounting system uses a ledger, a group of accounts, to store balances of accounts. The asset, liability, and fund balance accounts have similar data structure and functionality. The expenditure account, however, contains different attributes such as a different account number structure and additional codes for associating expenditures with project sites. Therefore, we have two subtypes of a ledger account, namely a balance sheet ledger account subtype and a budget expenditure report subtype (Diagram 3.1.11).

As discussed above, the data structures and methods of the financial transaction object type are inherited by its subtypes and this was shown in diagram 3.1.12. The generalization diagram could be referred to as an inheritance hierarchy or subtyping diagram.

3.1.2 RELATIONSHIP AMONG OBJECTS

Objects encapsulate data and methods. Each object is, therefore, responsible for performing some task. The interactions among objects in an object oriented environment must be modelled to portray how objects cooperate with each other in fulfilling the requirements of the system. Since object oriented techniques use the same conceptual model for analysis and design, the models developed in the analysis phase are also used as skeleton framework for the design phase. Therefore, these real world interactions need to be modelled to furnish a logical view of the system in an object oriented environment.

DIAGRAM 3.1.11 A GENERALIZATION DIAGRAM FOR
LEDGER ACCOUNT OBJECT TYPE

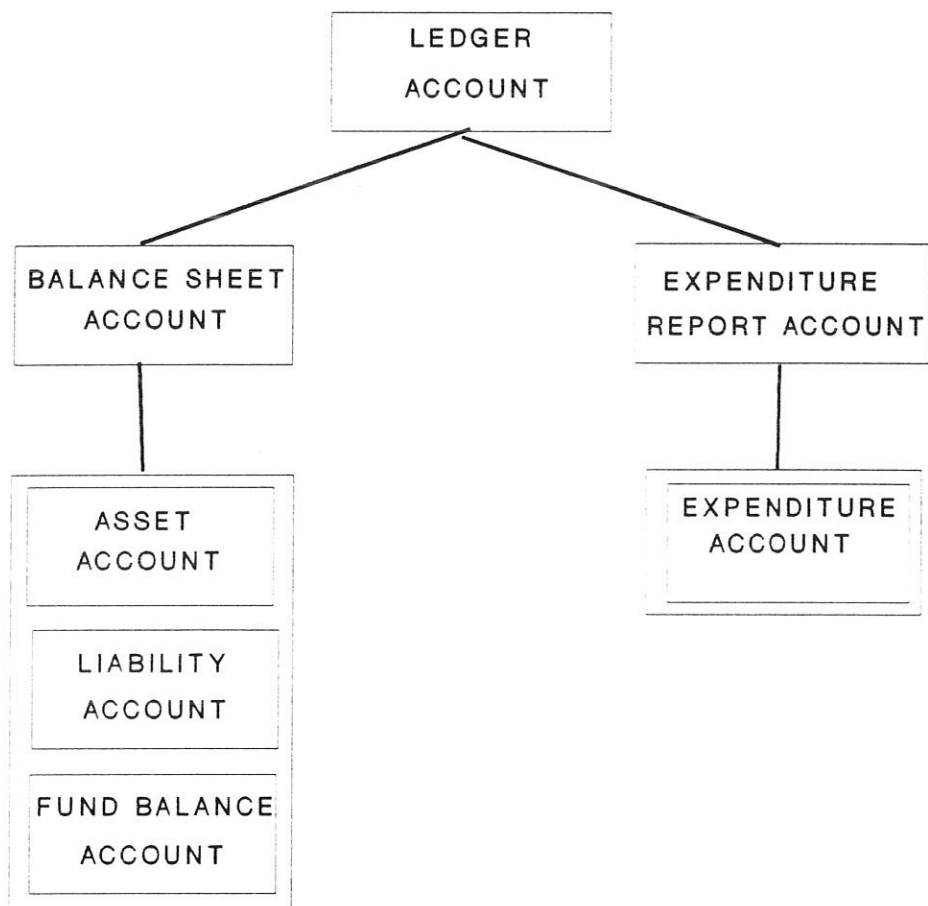
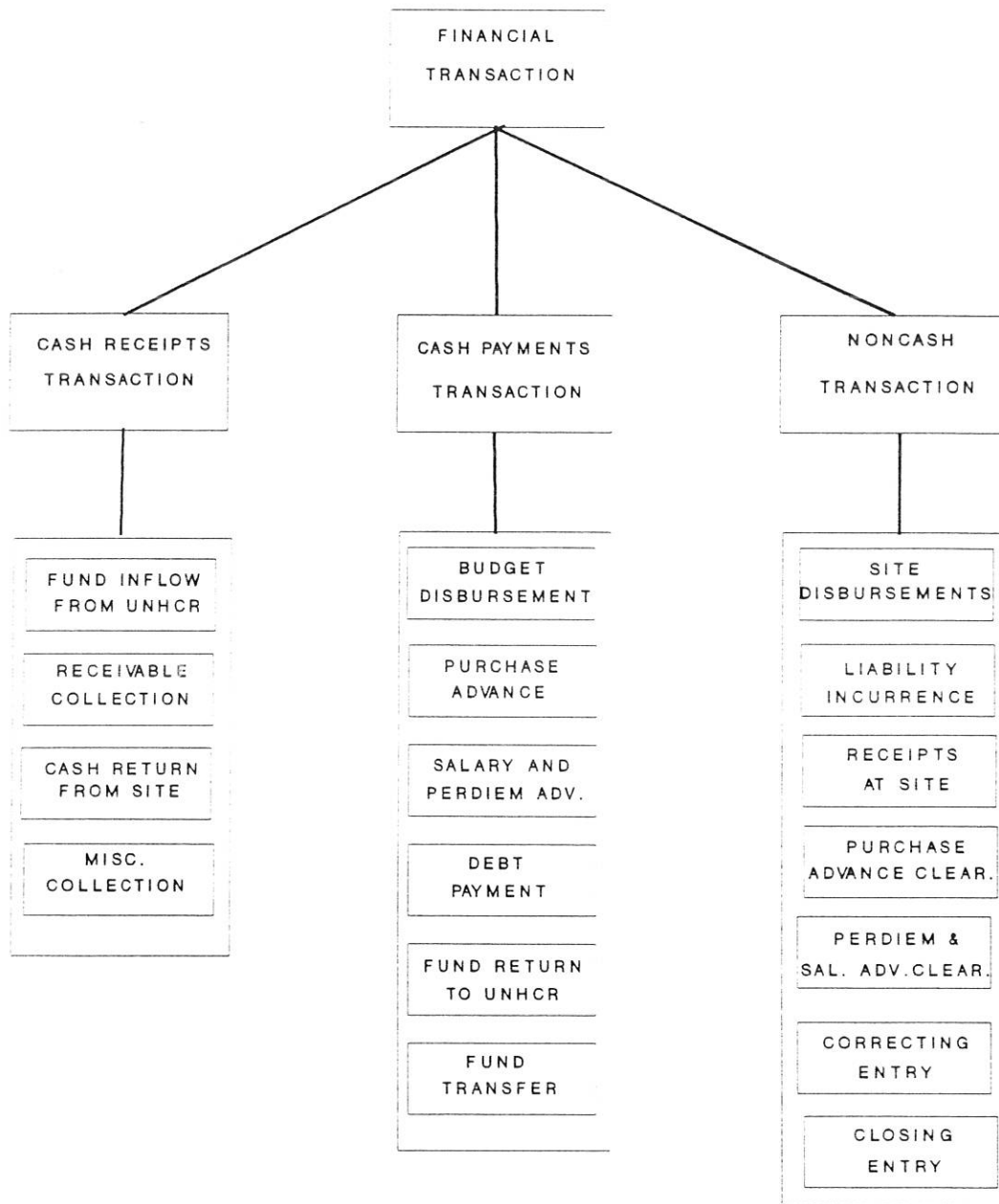


DIAGRAM 3.1.12 A GENERAIZATION DIAGRAM FOR
FINANCIAL TRANSACTION OBJECT TYPE



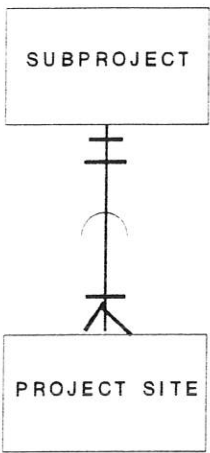
The traditional data modelling techniques are recommended for preparing object oriented models. This does not, however, imply that entities and objects represent equivalent concepts. There are four major types of relationships among object types: general relations, instantiation, subtypes and inneritance, composed-of relationships.

The subtypes, and inheritance relationship have been discussed in the previous section. Instantiation refers to an association between an actual object, such as a particular cash account, with its parent class which is the balance sheet account. Some object types of higher order are composed of different object types. Such relationships are known as "composed-of" relationships.

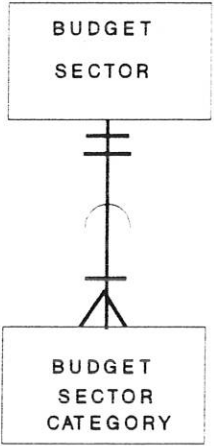
The lower order object types that participate in a composed-of relation do not inherit properties from the higher level items. These object types are merely essential building blocks of the larger object type. The lower order object types are not, therefore, subtypes of the higher level object type. Composed-of diagrams are used to show the relationships between aggregate objects and individual object types that make up the aggregate objects. This relationship is designated by an arc placed on the line connecting boxes that contain objects. Diagram 3.1.21 shows composed-of relationship between various object types or classes of the general ledger and financial reporting subsystem of the fund accounting system. The cardinality rules affecting object relationships are depicted in all object relationship diagrams.

The domain object data model includes diagrams that show general relationships among the objects. These relationships are essentially message flows or cooperations required for exchanging data and performing operations. No object need to be modelled and designed if it has not any relationship with other objects. Object relationship diagrams are used to show both static and dynamic relationships among objects. Diagrams 3.1.22, 3.1.23, 3.1.24, and 3.1.25 depict relationships among various objects. The objects in boxes represent instances of the object types.

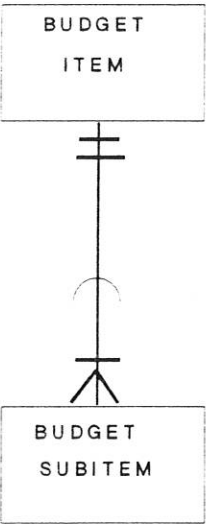
DIAGRAM 3.1.21 COMPOSED OF RELATIONSHIPS



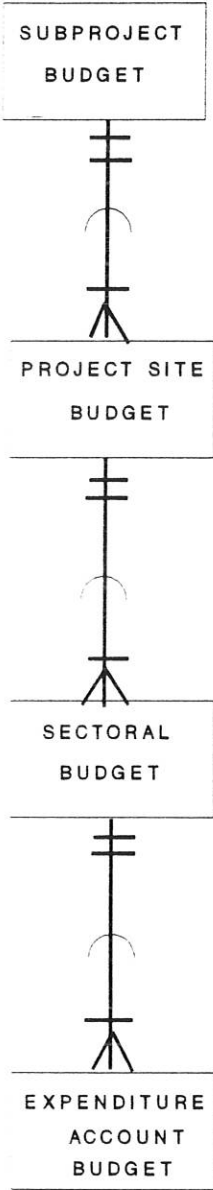
(A)



(B)



(C)



(D)

DIAGRAM 3.1.22 OBJECT RELATIONSHIP DIAGRAMS

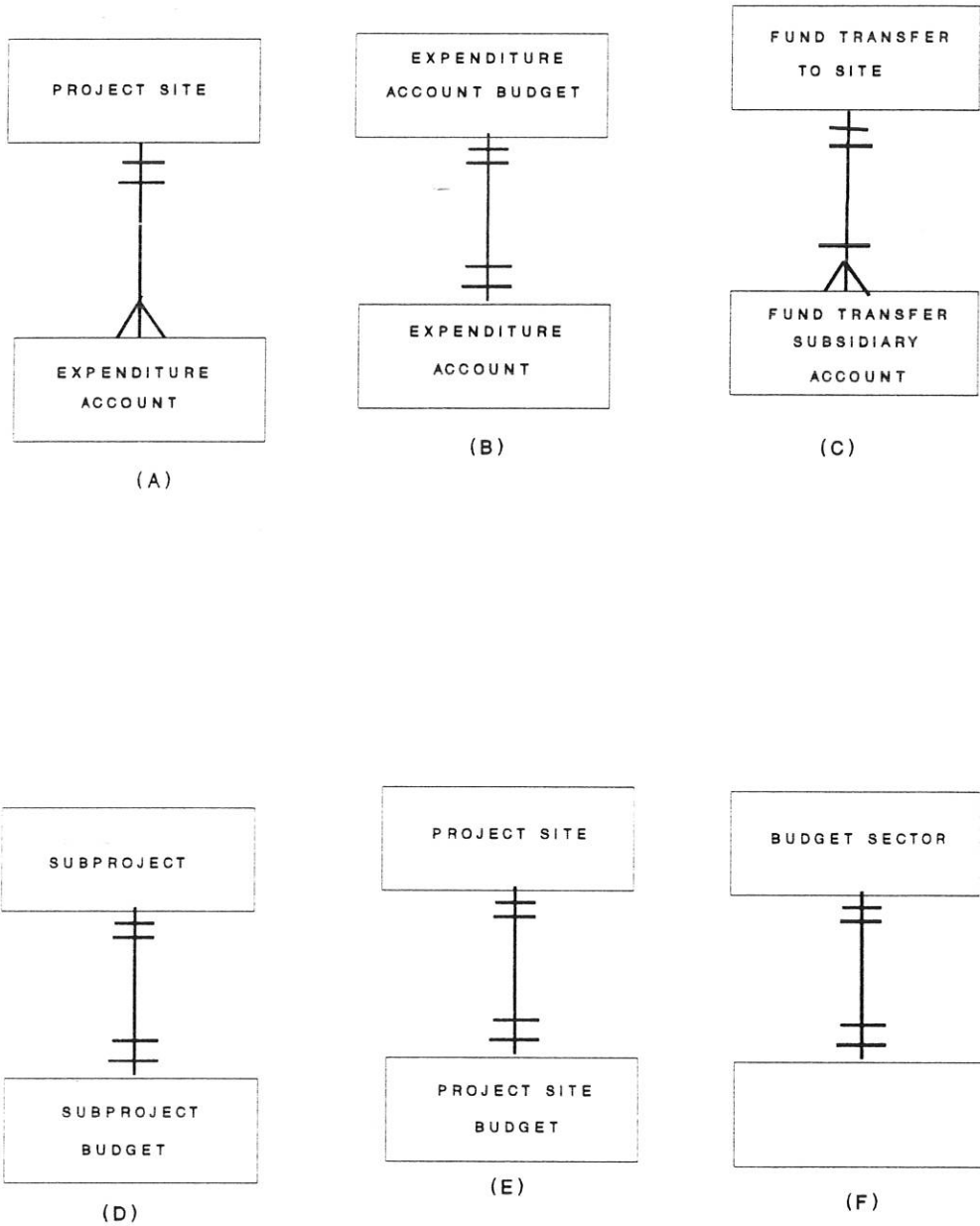


DIAGRAM 3.1.23 OBJECT REALTIONSHIP DIAGRAMS

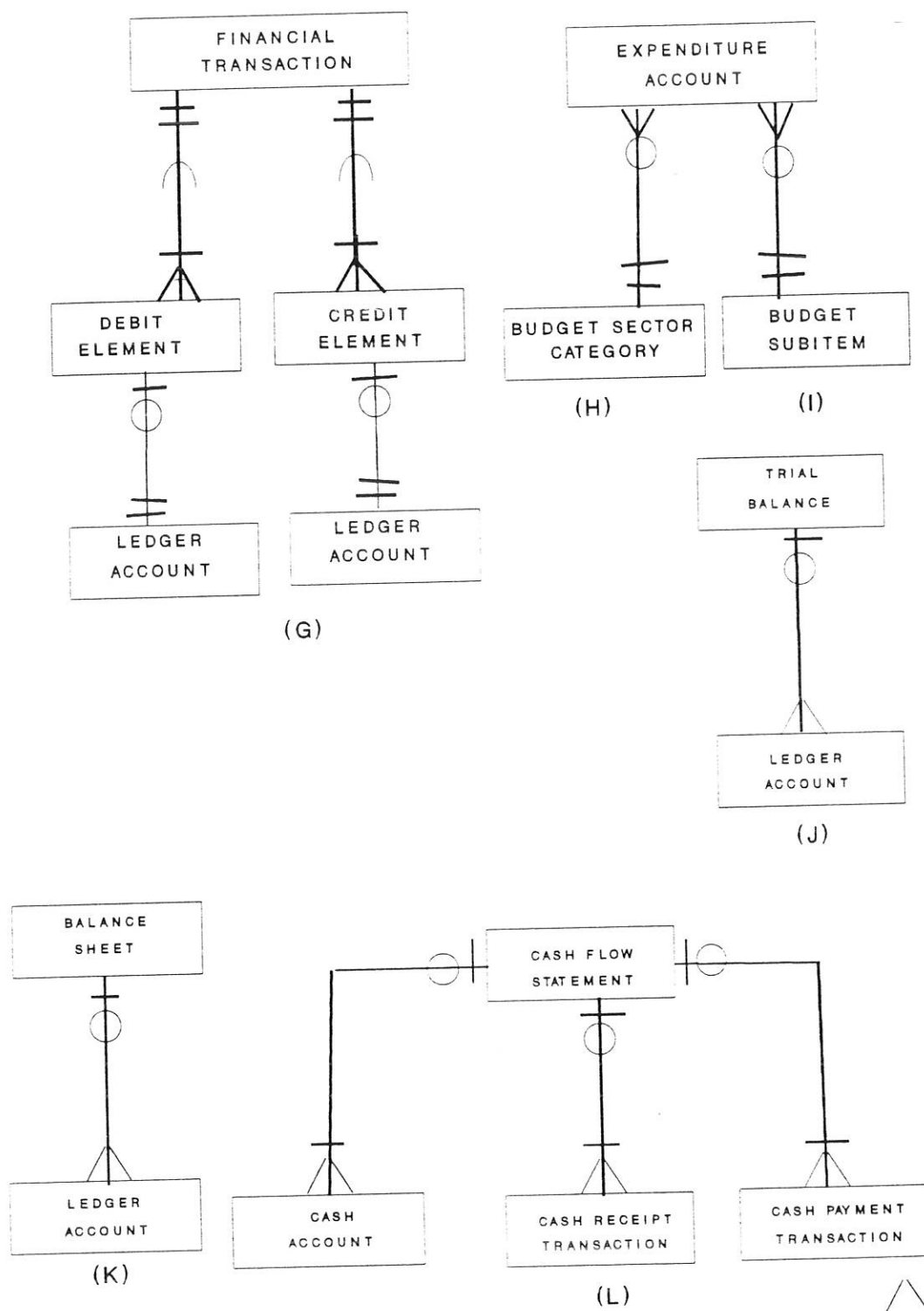


DIAGRAM 3.1.24 OBJECT RELATIONSHIP DIAGRAMS

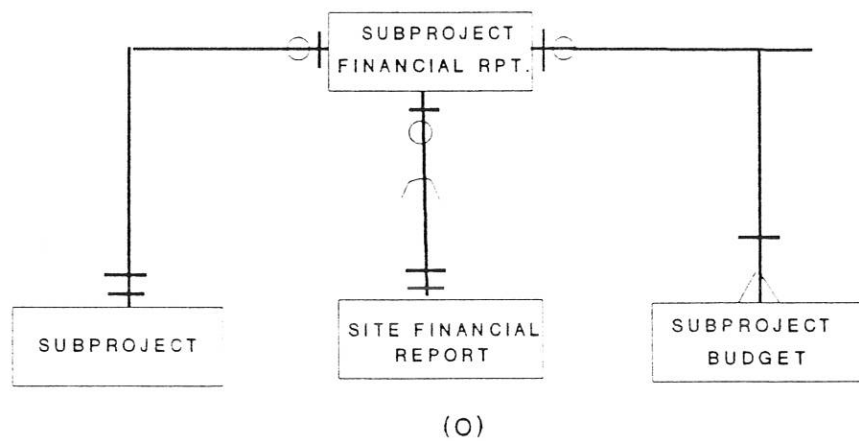
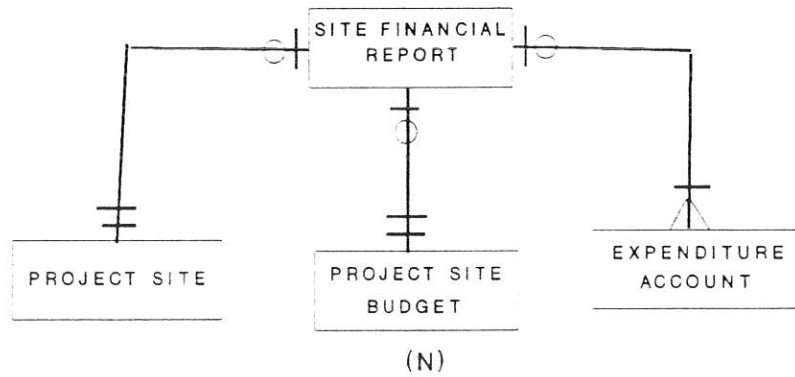
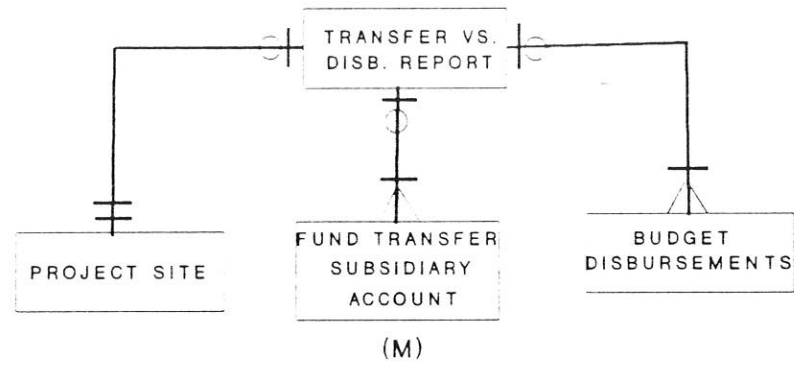
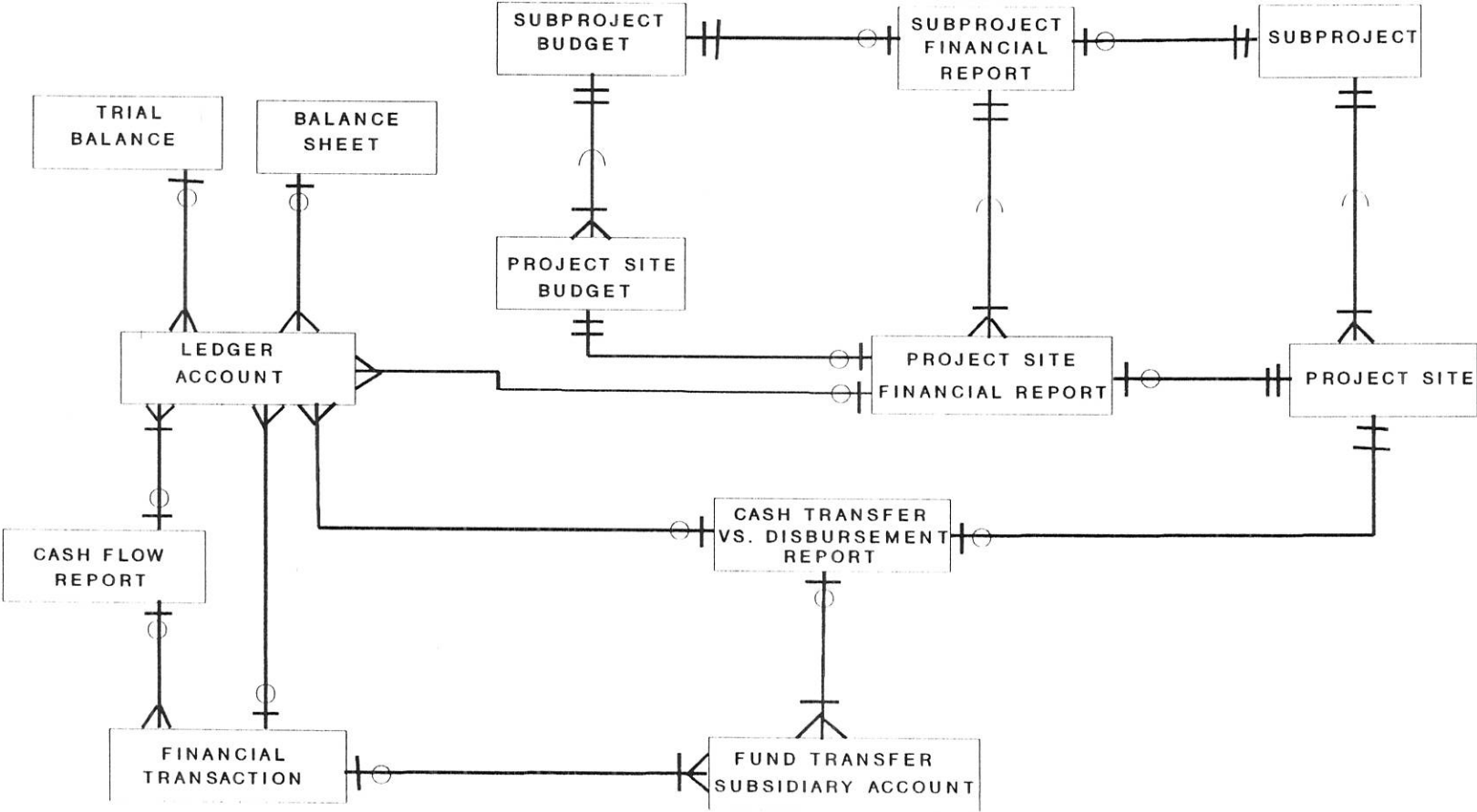


DIAGRAM 3.1.25 OBJECT RELATIONSHIP DIAGRAM FOR THE GENERAL
LEDGER AND FINANCIAL REPORTING SUBSYSTEM



3.2 OBJECT BEHAVIOR ANALYSIS

Object behavior analysis entails studying events and operations that cause changes in the states of objects. Object behavior analysis is similar to entity life history analysis. The analysis usually starts with an event in which the record of an object type is created, and terminates with an event in which the record of an object type is deleted or archived. The state of an object is the current value of the data objects held by a particular instance of an object type, such as the current balance of a cash account.

The behavior of an object is the association between the states and the methods appropriate for those states. Different methods are used to update the balance of the cash account depending on whether the transaction to be posted is a collection or a payment.

Objects are needed in a system because they store information and perform important operations vital to the fulfillment of the requirements of the system. Objects encapsulate private data structures and methods. In order to design appropriate methods and data structures for objects, it is indispensable to study what data they hold and what operations they perform with such data at a certain point in time. Object behavior changes result from occurrence of events. Events trigger operations that cause object state changes. In the financial system, an account balance changes when a transaction occurs. Financial transactions are, therefore, major event types.

In an object-oriented environment, events constitute requests sent to an object to activate its methods and cause changes in the state of an object. However, some operations performed by an object may not cause changes in its states. In object behavior analysis, we thoroughly examine only those operations and events that cause changes in the state of objects.

3.2.1 OBJECT STATES AND OBJECT STATE CHANGES

Objects in the real world may have infinite states. However, in dealing with problem domain objects, we assume that objects have discrete states. A state transition diagrams is used to depict the sequence of states an object assumes during its lifetime. The state transition diagram of each object type used in the general ledger and financial reporting subsystem is in effect an outline of operations that must be designed and implemented to attain system requirements.

The states are essentially operation results that represent the functionality of the fund accounting system. In a state transition diagram, horizontal lines represent states and vertical lines represent transition among states. Ordinarily, permissible states differ from one object to another. But certain fund accounting objects may have some similar states. These similar states are identified to design a reusable class of operations for the system. One of the salient features of an object oriented system is the inheritance of class behavior. For example, all general ledger accounts are debited and credited frequently.

Consequently debiting an account or crediting an account is an operation to be requested by all ledger account objects from the ledger account class.

Most of the objects in the fund accounting system have only two general states, namely creation and destruction. Some objects pass only through the creation state and remain in this state with their original data for a long period of time. Hence, the need for depicting object state changes in a state transition diagram is valid only for objects with more than two distinct states (Diagrams 3.2.11 and 3.2.12).

- (1) Object types with single states: budget sector, budget sector category, budget item, budget sub-item
- (2) Object types exhibiting two basic states: subproject, project site, subproject budget, project site budget, sectoral budget, expenditure account budget
- (3) Objects types with more than two basic states: ledger account, financial transaction, fund transfer subsidiary account, journal entry, balance sheet, trial balance, cash flow statement, transfer versus disbursement report for a project site, quarterly financial report for a project site, quarterly financial report for a subproject

3.2.2 EVENTS, OPERATIONS, and RULES

In order to design an object's methods, the sequence of operations related to each object are analyzed. Operations are invoked by events. An event diagram shows events and the operations triggered by events.

DIAGRAM 3.1.11 A GENERALIZATION DIAGRAM FOR
LEDGER ACCOUNT OBJECT TYPE

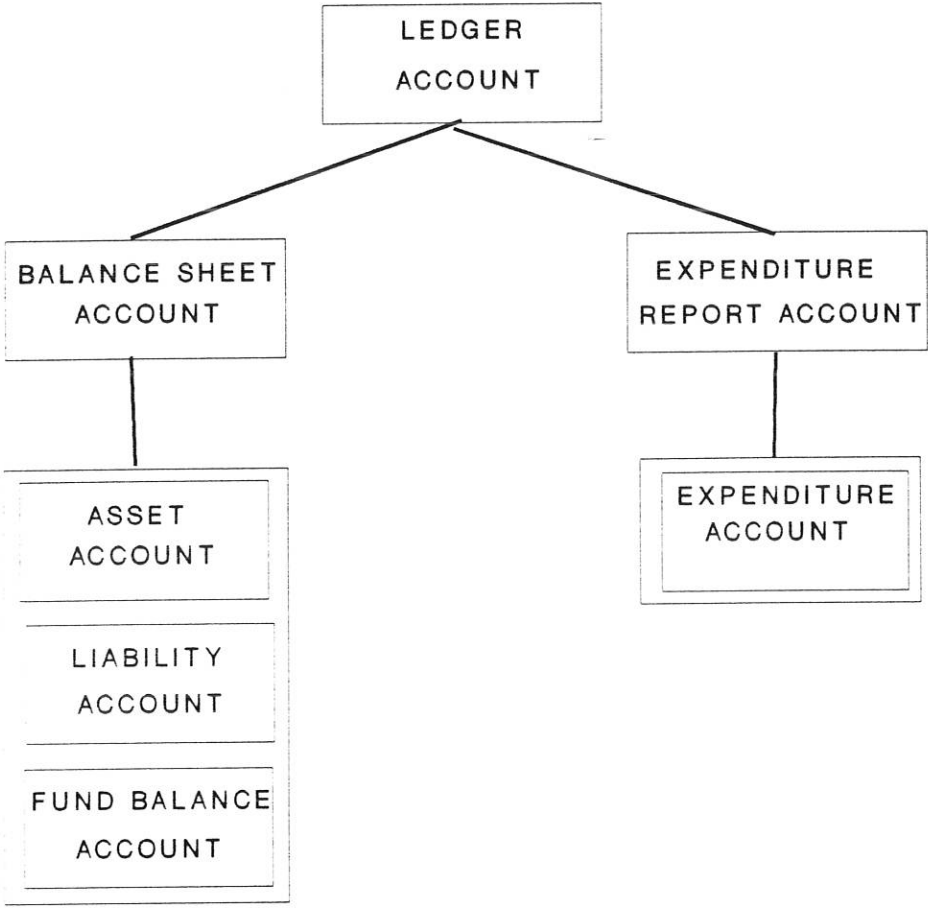
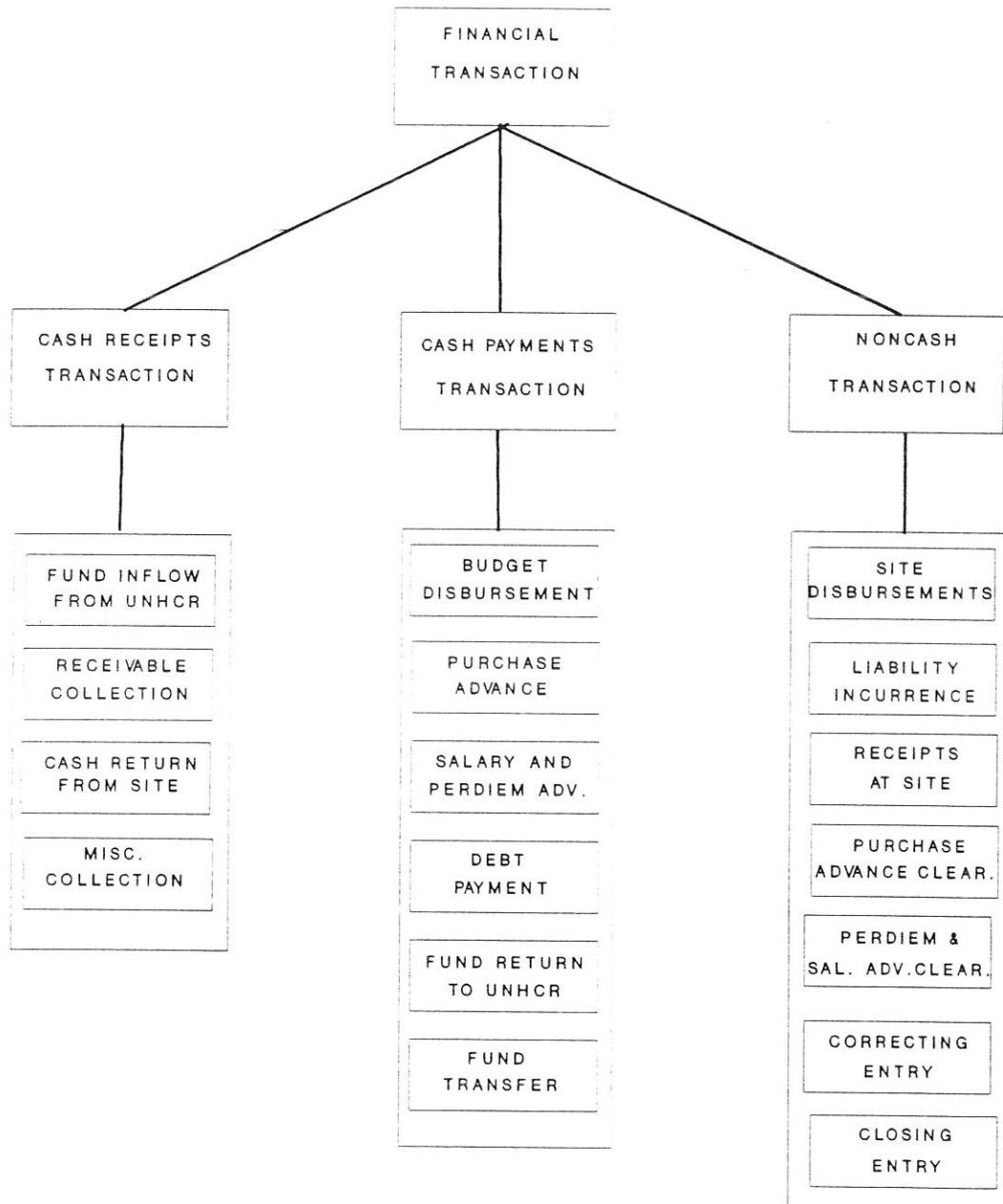


DIAGRAM 3.1.12 A GENERAIZATION DIAGRAM FOR
FINANCIAL TRANSACTION OBJECT TYPE



An event is a noteworthy change in the state of an object. Because of this, event diagrams must be tightly linked to state-transition diagrams. Operations are triggered only when certain conditions are fulfilled. These conditions are expressed as trigger rules and operation condition rules.

The primary processes to be performed by the newly proposed general ledger and financial reporting subsystem are:

1. journalizing financial transactions
2. posting financial transactions
3. preparing six types of financial reports
4. entering annual budget allocations
5. entering detailed fund transfer data in subsidiary fund transfer accounts.

After the primary processes are identified, it is necessary to examine the detailed sequences of activities that make up each of these processes. The availability of such details is a requisite for drawing event diagrams. As the motto of an object oriented approach is to maximize the creation of reusable classes, similar activities are extracted from the event diagrams. For example, both the preparation of site financial reports and subproject financial reports employ identical steps except that additional processes are performed for generating the latter. The design of methods is carried with the aim of eliminating duplication of process design.

The flow of events and operations is regulated by rules. Some rules express how operations are triggered. Some rules state that operations can not be performed unless certain

preconditions are fulfilled. Rules are also used to verify the results of operations. The relationships among objects and classes are also controlled by certain integrity rules.

Rules that govern the operations of the major functions of the general ledger and financial reporting subsystem of ARRA's fund accounting system are described in the following sections. A series of operation diagrams are prepared for documenting the operations and rules for the fund accounting objects. In operation diagrams round cornered boxes represent an operation or a method. An arrow connecting two operations indicates the event that marks the end of the preceding operation and the conditions and rules for triggering the next operation.

Rules attached to operation boxes describe operation precondition and operation post condition rules. Operation precondition rules are a means of testing the appropriate state an object must be in before a new operation or method is executed. The operation post condition rules check whether an operation has resulted in achieving the desired state of an object.

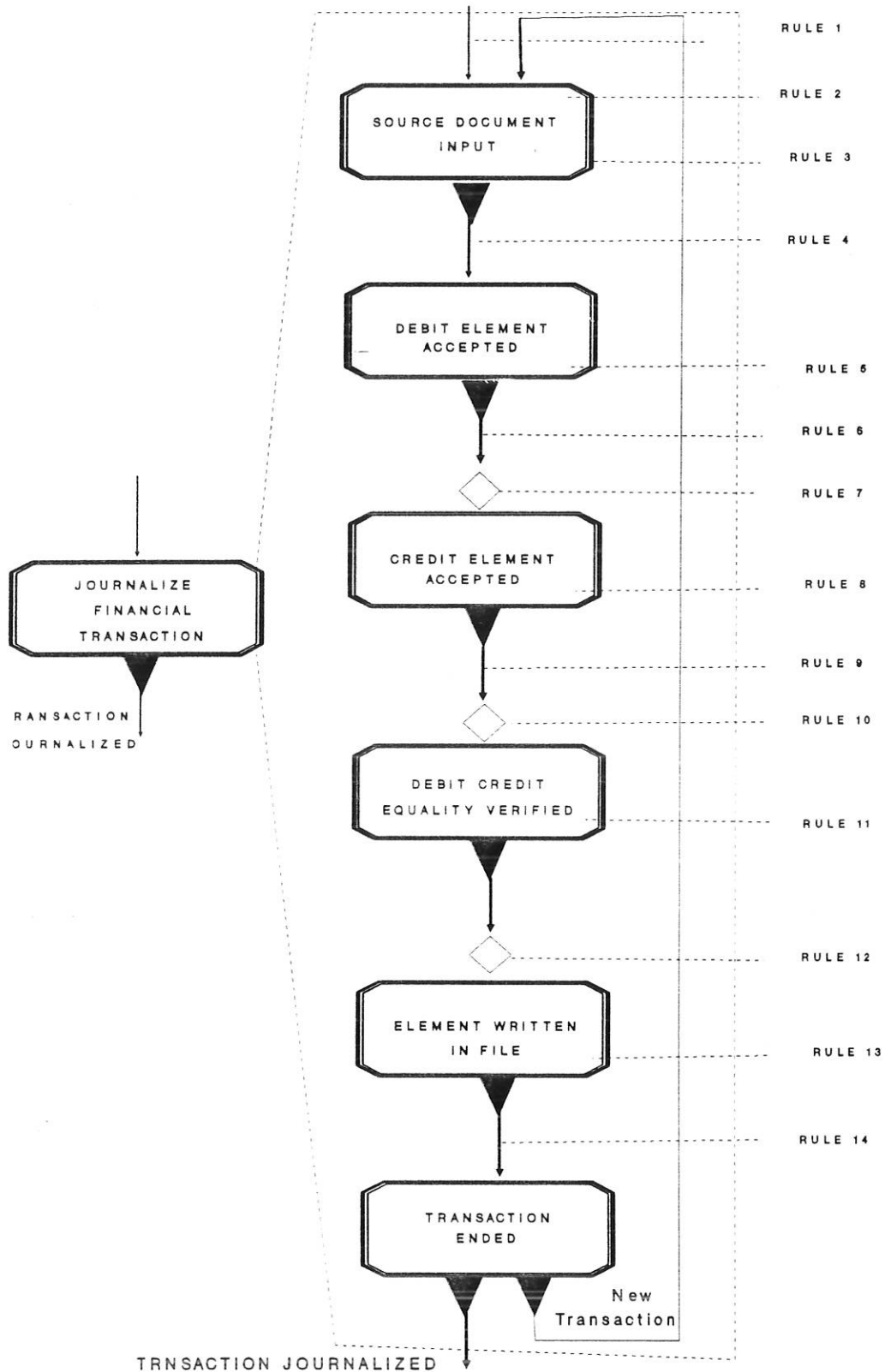
Diamond symbols represent special control conditions or rules to be satisfied before the next operation is invoked. Some operation blocks need to be decomposed further and be shown in a child operation diagram. However, such details are presented as class responsibilities in methods design in chapter four.

Process Description: Journalizing financial transaction
Diagram 3.2.21

- Rule 1: On Start Transaction
begin accepting source document input
- Rule 2: If the transaction is a cash receipt transactions
Then accept payer name, receipt number, and date
If the transaction is a cash payment transaction
Then accept payee name, check no or bank letter
no, and date
Begin
If it is a budget disbursement
Then accept the name of a project site
If it is a purchase advance
Then accept the Warehouse Purchase
Requisition No.
End
If the transaction is a noncash transaction
If it is related to disbursements at site
Then accept project site name, Bank
Disbursement Voucher No. or the Cash
Payment Voucher No.
If it is a purchase advance clearance
Then accept Goods Received Note No.
Else accept the Journal Voucher No.
- Rule 3: If source document received is not appropriate
Then send an error message
- Rule 4: If source document is correct and accepted
Then begin accepting debit elements (entries)

- Rule 5: When entering each debit element detail is completed
Then check that account number and amount are entered
If the entry is incomplete
Then send an error message
- Rule 6: When the last debit element is entered
Then begin accepting credit elements
- Rule 7: If the total of debit entries is not available
Then send an error message
- Rule 8: When entering each credit element detail is completed
Then check that account number and amount are entered
- Rule 9: When the last credit element is entered
Then begin checking the equality of debits and credits
- Rule 10: Check whether the totals of debit entries and credit entries are available before verifying their equality
- Rule 11: If the totals of debits and credits are different
Then send an error message
- Rule 12: If the totals of debits and credits are equal
Then begin recording the transaction elements
- Rule 13: Check that the record number of the transaction file
is increased by one after each element is written
- Rule 14: If the last entry is written
Then end the transaction

DIAGRAM 3.2.21 JOURNALIZING A FINANCIAL TRANSACTION



Process Description: Posting transaction files

Diagram 3.2.22

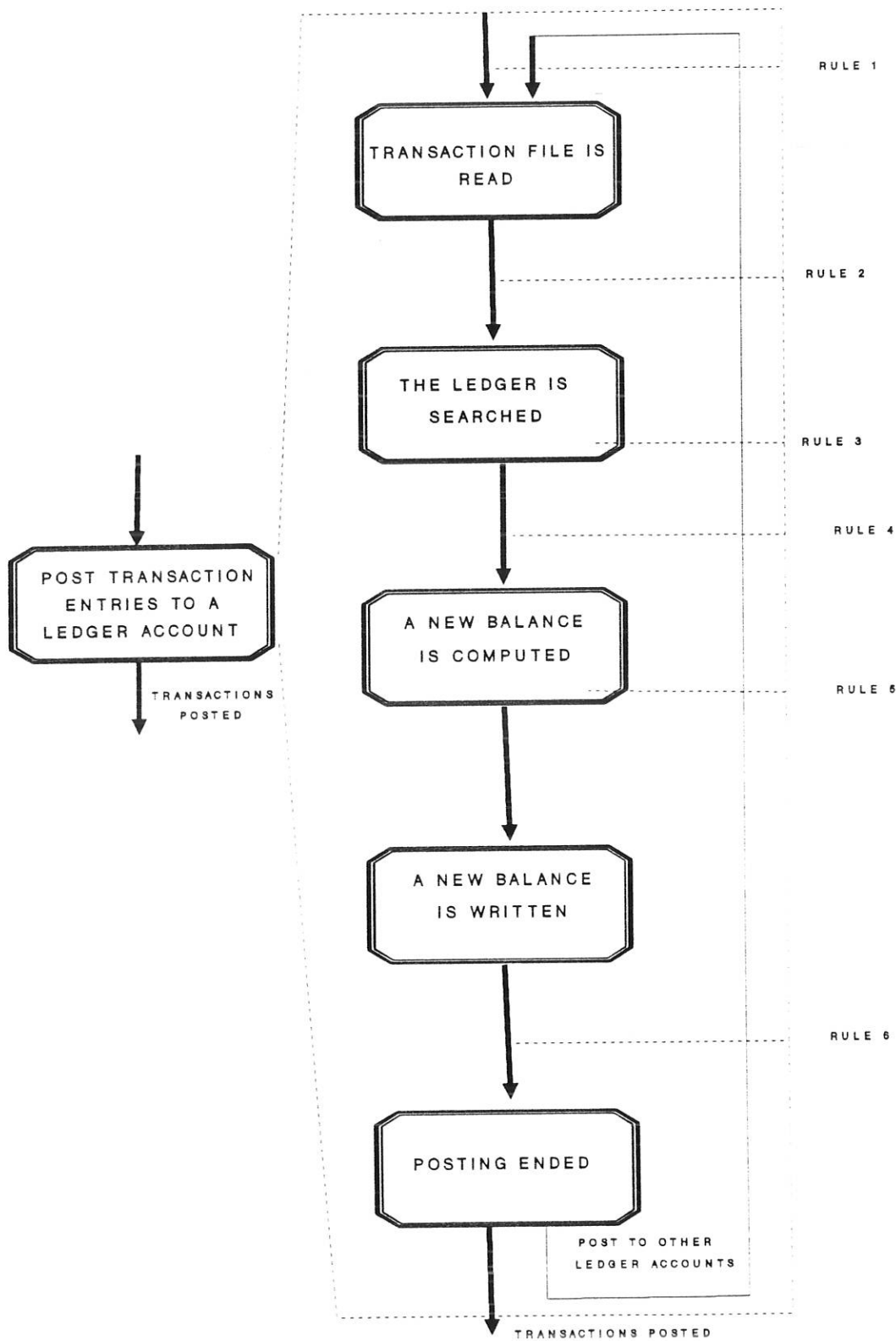
- Rule 1: On start posting
Begin reading a transaction file
- Rule 2: If the transaction file is cash debit or cash credit
Then look for cash accounts in the balance sheet account group
If the transaction file is budget disbursement debit
Then look for a budget disbursement account in the appropriate project site disbursement ledger
- Rule 3: If an account is not found in a ledger
Then send an error message
- Rule 4: If an account is found
Then begin processing balance computation
- Rule 5: If the new balance is the same as previous one
Then send an error message
- Rule 6: If the new balance is correctly computed
Then end the posting process

Process Description: Generating a balance sheet

Diagram 3.2.23

- Rule 1: When it is end of month or a statement is needed
Begin reading the ledger
- Rule 2: When the last ledger is read
Then begin computing the total of assets
- Rule 3: If the total of assets is zero or less than zero
then
Then send an error message
- Rule 4: When the total of assets is computed
Then begin computing the total of liabilities

DIAGRAM 3.2.22 POSTING FINANCIAL TRANSACTIONS



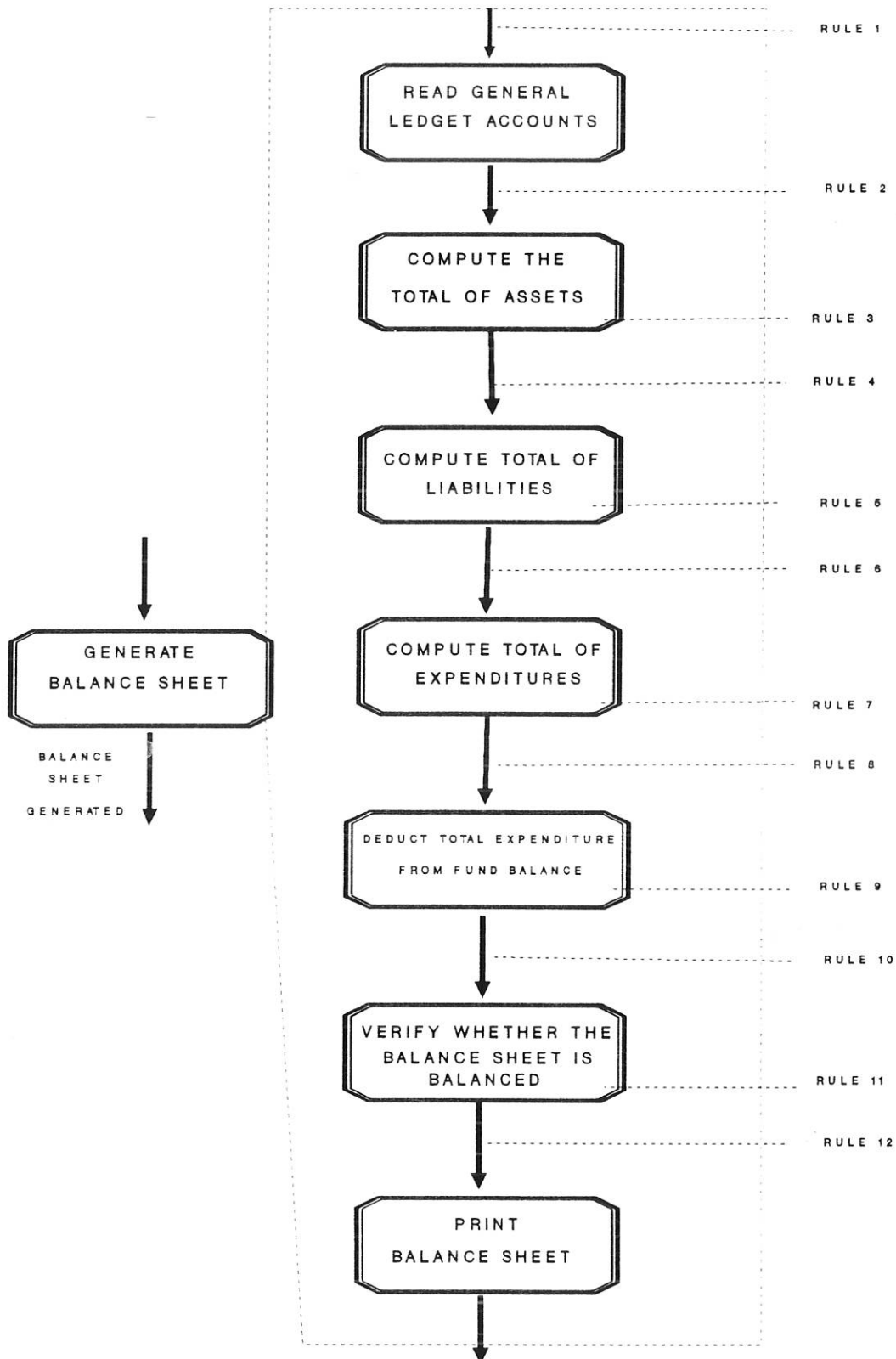
- Rule 5: If the total of liabilities is less than zero
Then send an error message
- Rule 6: When the total of liabilities is computed
Then begin computing the total of expenditures
- Rule 7: If the total of expenditures is zero or less than zero
Then send an error message
- Rule 8: When the total of expenditures is computed
Then begin computing the difference between
expenditures and fund balance
- Rule 9: If the difference is less than or equal to zero
Then send an error message
- Rule 10: When computing the difference is completed
Then begin verifying the balance sheet duality
- Rule 11: If the balance sheet total of assets is not equal
to the balance sheet total of liabilities and fund
balance
Then send an error message
- Rule 12: When a balance sheet is verified
Then begin printing the balance sheet

Process Description: Generating a trial balance

Diagram 3.2.24

- Rule 1: When it is end of week or a statement is needed
Then begin reading the ledger
- Rule 2: When the last ledger account is read
Then begin summarizing the expenditures
- Rule 3: If each sectoral total is less than or equal to zero
Then send an error message

DIAGRAM 3.2.23 GENERATING BALANCE SHEET



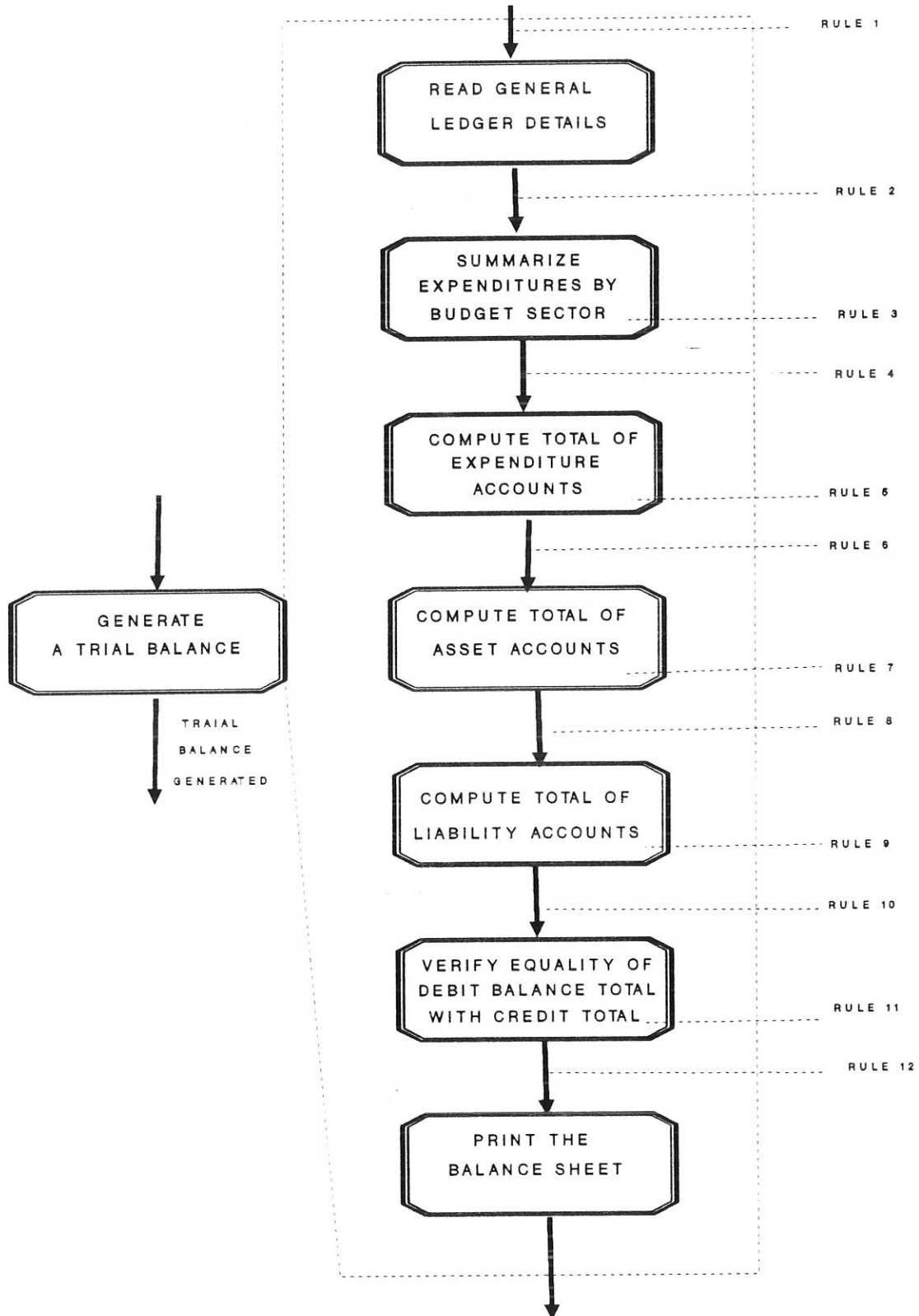
- Rule 4: When summarizing expenditures by sectors is completed
Then begin totaling expenditures
- Rule 5: If total of sectoral expenditures is less than or equal to zero
Then send an error message
- Rule 6: When totaling expenditures is completed
Then begin totaling the assets
- Rule 7: If the total of assets is less than or equal to zero
Then send an error message
- Rule 8: When totaling assets is completed
Then begin totaling liabilities
- Rule 9: If the total of liabilities is less than zero
Then send an error message
- Rule 10: When totaling liabilities is completed
Then begin checking duality of ledger balances
- Rule 11: If the total of debit balances is different from the total of credit balances
Then send an error message
- Rule 12: When the trial balance is proved
Then begin printing the trial balance

Process Description: Generating a cash transfer and expenditure

report. Diagram 3.2.25

- Rule 1: When it is end of month or report is needed
Then begin reading project site cash transfer file
- Rule 2: If project site cash transfer file is not found
Then send an error message

DIAGRAM 3.2.24 GENERATE A TRIAL BALANCE



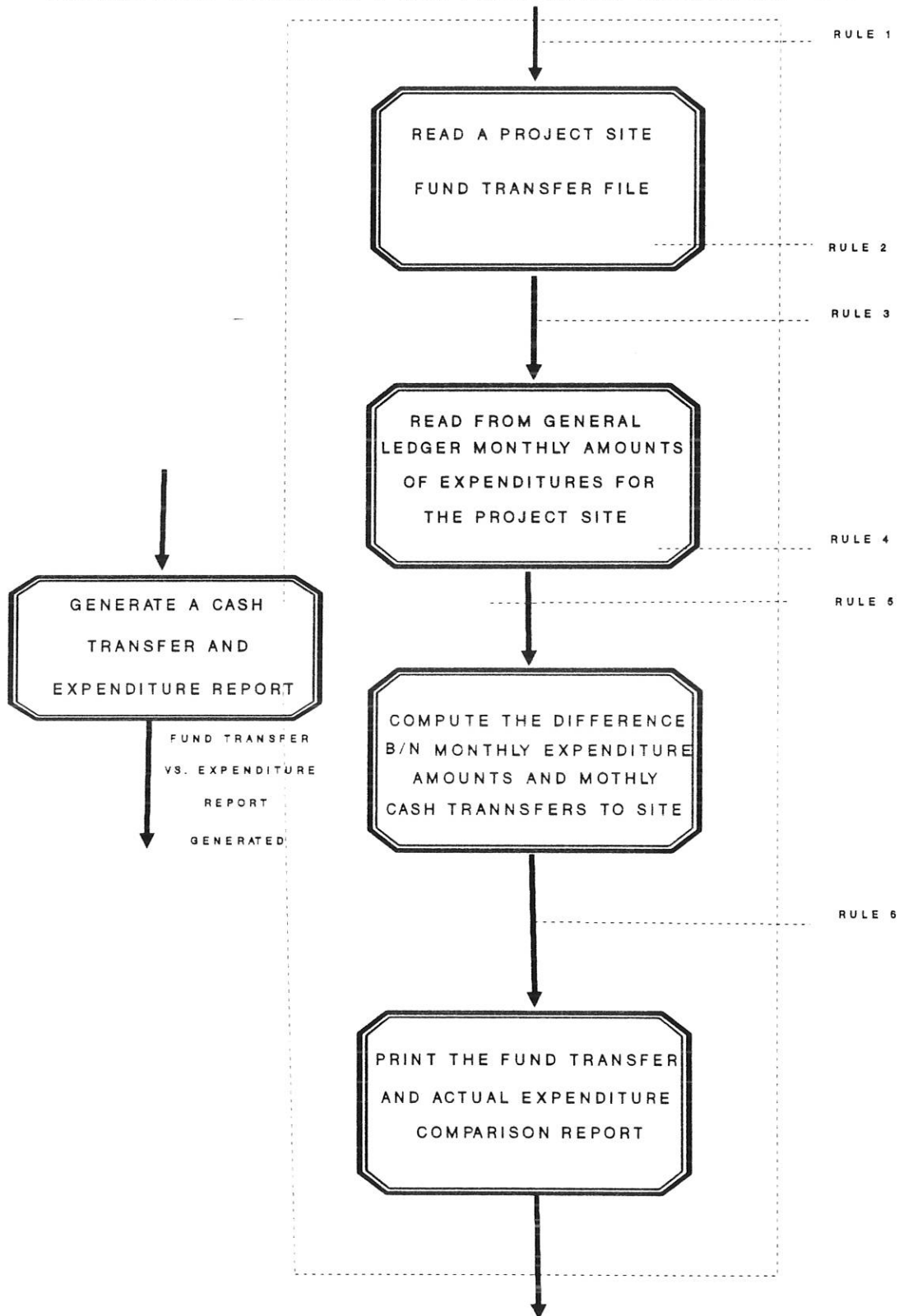
- Rule 3: When reading site fund transfer file is completed
Then begin reading general ledger expenditure accounts
- Rule 4: If monthly expenditure amounts for a project site are not available
Then send an error message
- Rule 5: When reading expenditure accounts is completed
then begin computing discrepancies between monthly transfers and expenditures
- Rule 6: When computing discrepancies is completed
Then begin printing the transfer and expenditure report

Process Description: Generating a cash flow report

Diagram 3.2.26

- Rule 1: When it is end of month or report is needed
Then begin reading general ledger cash account details
- Rule 2: If cash accounts are not found in the general ledger
Then send an error message
- Rule 3: When reading cash account details is completed
Then begin reading cash debit transaction file
- Rule 4: If a cash debit transaction file is not found
Then send an error message
- Rule 5: When reading cash debit transaction file is completed
Then begin reading cash credit transaction file
- Rule 6: If a cash credit transaction file is not found
Then send an error message

DIAGRAM 3.2.25 GENERATING A CASH TRANSFER AND EXPENDITURE REPORT

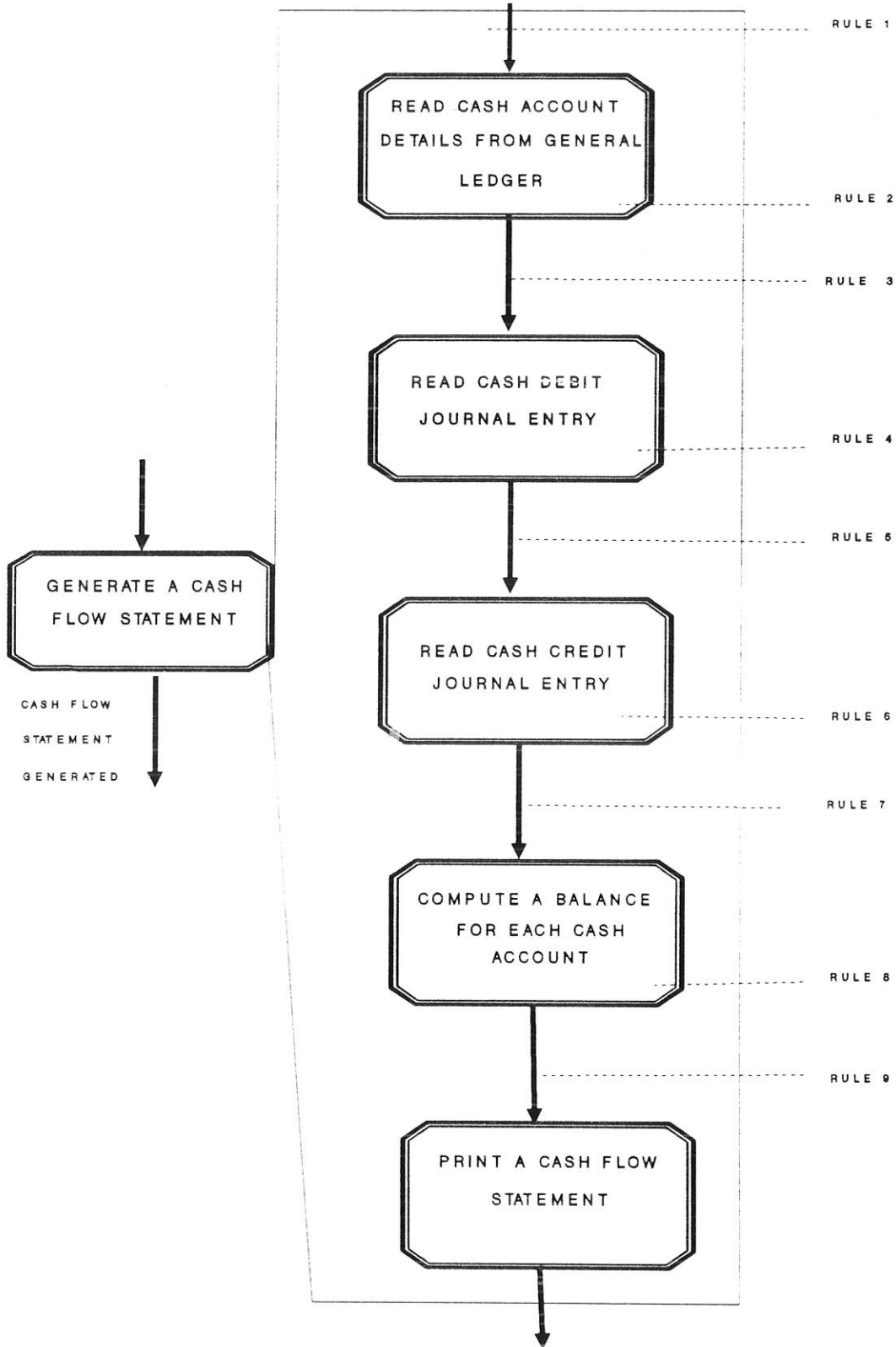


- Rule 7: When reading cash credit transaction file is completed
Then begin computing new account balance
- Rule 8: If new cash balance is less than or equal to zero
Then send an error message
- Rule 9: When computing new cash balances is completed
Then begin printing a cash flow report

Process Description: Generating a quarterly project site financial report. Diagram 3.2.27

- Rule 1: When it is the end of the quarter or a report is needed
Then begin reading a project site description file
- Rule 2: If a project site description file is not found
Then send an error message
- Rule 3: When finishing reading project description file
Then begin reading site expenditure accounts
- Rule 4: If project site expenditure accounts are not found
Then send an error message
- Rule 5: When reading expenditure accounts is completed
Then begin reading expenditure account budget file
- Rule 6: If a project site expenditure account budget file is not found
Then send an error message
- Rule 7: When reading the expenditure account budget file is completed
Then Begin computing discrepancies
- Rule 8: If grand totals cross checking does not indicate accuracy
Then send an error message

DIAGRAM 3.2.26 GENERATING A CASH FLOW STATEMENT



- Rule 9: When grand totals are correct
Then start categorizing the details
- Rule 10: When arranging budgeted and actual expenditures is done
Then start printing the balance sheet

Process description: Generating a quarterly subproject
Financial report. Diagram 3.2.28

- Rule 1: When it is the end of the quarter or a report is needed
Then start reading a subproject description file
- Rule 2: If a subproject description file is not found
Then send an error message
- Rule 3: When the subproject file is read
Then begin reading project site quarterly reports
- Rule 4: If a project site quarterly financial report is not found
Then send an error message
- Rule 5: when all site financial reports are read
Then start consolidating the site financial reports
- Rule 6: If grand total cross checking indicates a mistake
Then sen an error message
- Rule 7: If grand totals are proved
Then start printing a quarterly subproject financial report

DIAGRAM 3.2.27 GENERATING PROJECT SITE FINANACIAL REPORT

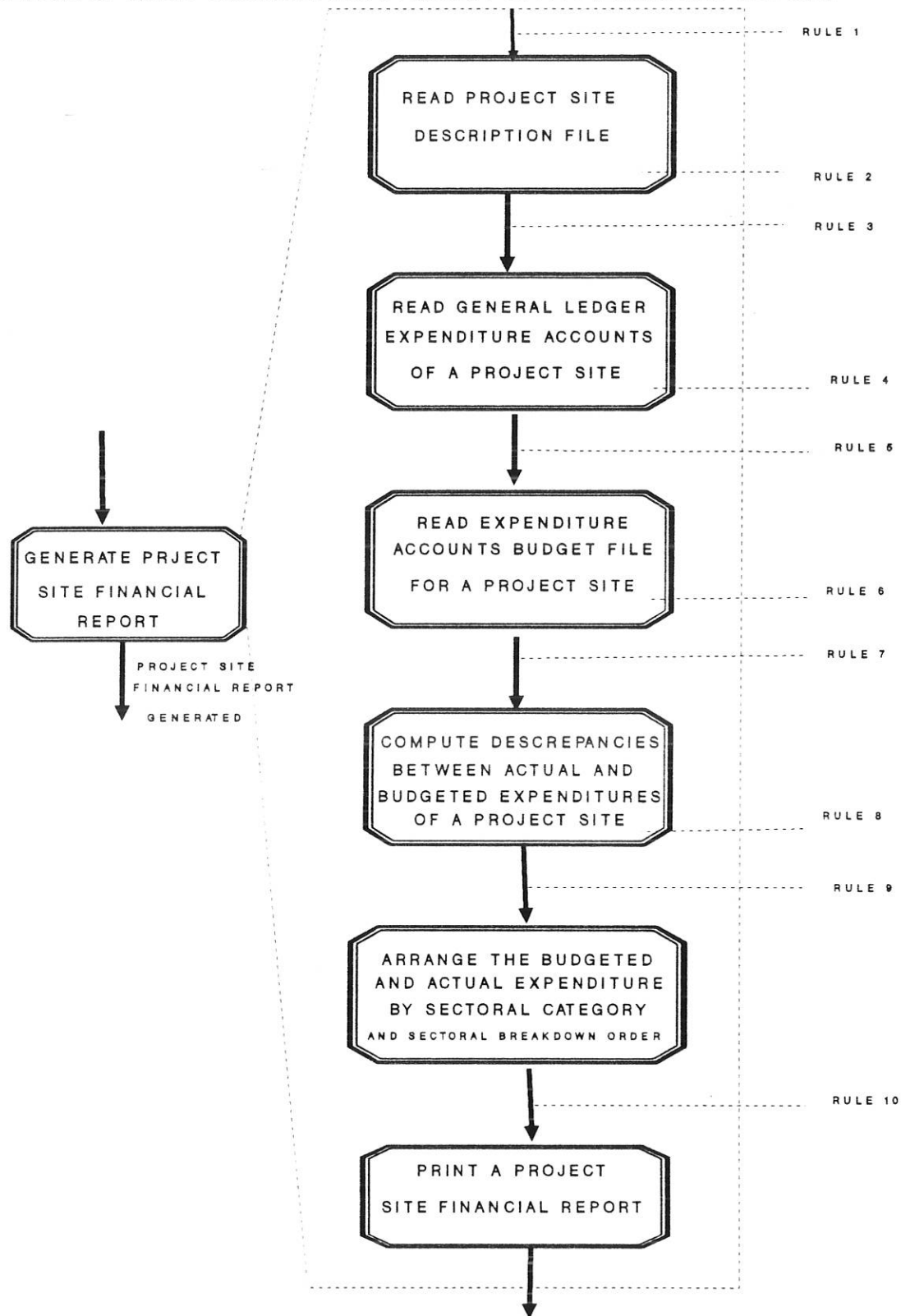
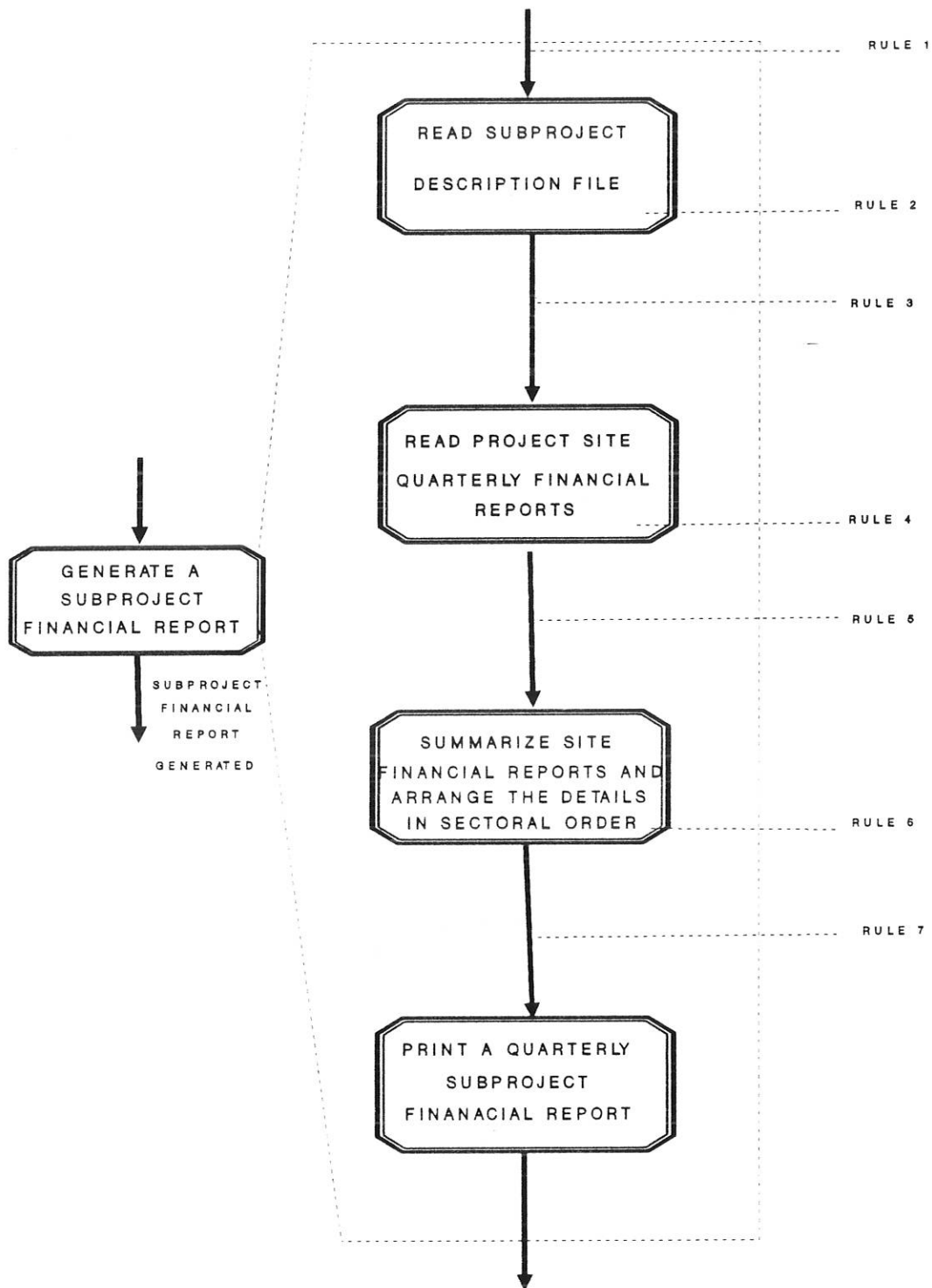


DIAGRAM 3.2.28 GENERATING A SUBPROJECT FINANACIAL REPORT



CHAPTER FOUR

OBJECT-ORIENTED DESIGN

4.0 INTRODUCTION

The design phase of an object-oriented application creates full specifications of classes selected for implementation. Object types and their relationships are mapped into class hierarchies and collaborations. The design activity fully preserves and builds on the results of the analysis stage. This smooth transition enhances the compatibility of the design course with the findings of the analysis stage and removes any abrupt change in the level of abstraction of system descriptions.

Object-oriented design encompasses the following tasks:

- (1) specifying classes and class hierarchies for implementing the problem domain objects
- (2) designing data structures for the classes
- (3) designing methods for operations to be performed by classes
- (4) designing user interfaces

The design of the user interface is to be tackled at last. The separation of the user interface design from class data and methods **design facilitates the design of interface-independent reusable classes.**

4.1 SPECIFICATION OF CLASSES

In object structure analysis, the system objects were identified. The analysis of the objects continues at a deeper level during design. Objects that have similar data structures and that perform similar operations are organized into a class.

The class serves as a module for source code and a type for the class instance. A class instance is each instance of the real world object of which a class is made of. A class defines the operations that can be performed on an instance. A class also defines the variables common to all of its instances, which are called class variables. However, the instance is responsible for storing the value of a variable that characterizes the instance's state.

Some classes containing similar objects are arranged in a hierarchy. Here some or all of the properties and methods of parent classes are inherited by the child classes. For example, balance sheet account and expenditure account classes have subclasses. Some objects, such as budget, could be of single type having no similarities with others. Nevertheless such objects may stand as classes by themselves as long as they offer useful service to other objects or to the system at large. For example, a budget class is specified to contain all budgetary data for the financial system and supply such data upon request by other classes.

In order to reduce internal complexity of some classes new subclasses are created. A journal entry contains a full description of financial transactions. However, different transactions have different data structures. Hence, a journal entry class is divided into three subtypes. Not all objects identified during analysis are good for implementation design. Proliferation of classes should be controlled.

Some objects may not have data structures except a few attributes which are supplied to other objects on request. Objects having no responsibilities for performing operations and whose structures are integral parts of other objects are better combined with other objects. Accordingly, the budget sub-item object and budget sector category objects are combined with the expenditure account object. An expenditure account is created from a combination of the attributes of budget sector, budget category, budget item, and budget sub-item objects. The financial transaction and journal entry objects are also merged together.

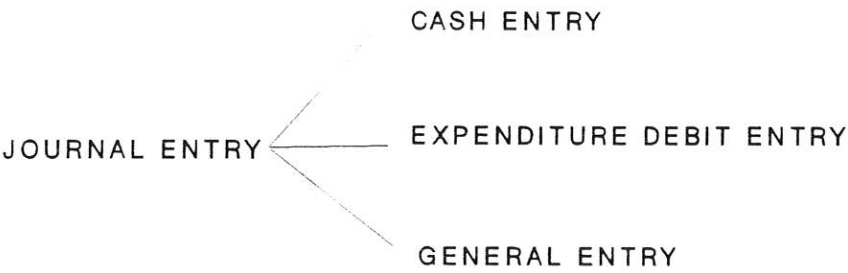
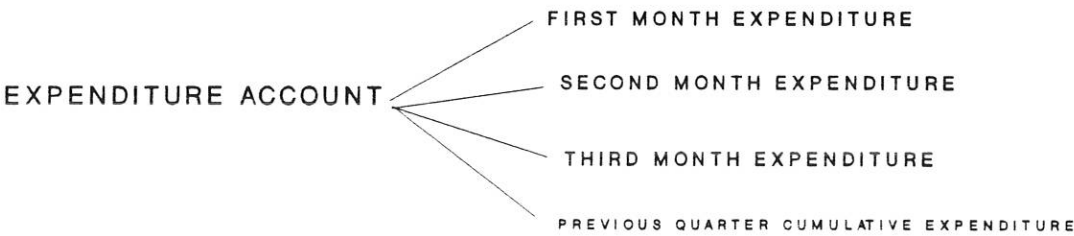
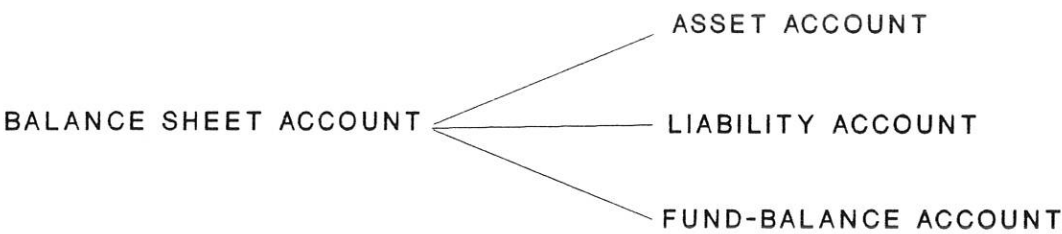
The object-oriented design of the general ledger and financial reporting application proceeds with the following list of key classes:

1. balance sheet account
2. asset account
3. liability account
4. fund balance account
5. expenditure account
6. first month expenditure account
7. second month expenditure account
8. third month expenditure account

9. previous quarter cumulative expenditure account
10. journal entry
11. cash entry
12. expenditure debit entry
13. general entry
14. budget
15. fund transfer
16. user interface
17. printer
18. user

Unlike former design methodologies, object-oriented design is geared to software implementation of classes, because one cannot create a true object oriented design by ignoring the details of object oriented implementations. Hence, during the design stage a profound investigation of object behavior is conducted to discover similarities among classes. The specification of class hierarchies and methods is directly related to object-oriented language implementations. The classes of the fund accounting general ledger subsystem that exhibit hierarchical inheritance relationships are shown in diagram 4.11. This diagram is known as a fern diagram.

DIAGRAM 4.11 CLASS HIERARCHIES



4.2 CLASS DATA STRUCTURE DESIGN

The persistence of the behavior of classes necessitate storing object state data. From object-oriented perspective, we say an object remembers its state. Object database management systems are now on the market. Such databases are believed to have all the necessary intelligence to automatically handle updating and redundancy problems. The classes of the fund accounting application are to be constructed for demonstration purposes using a semi-object-oriented data base management system known as dBASE for Windows. In true object-oriented applications there is no program data independence. The object data structure is bound to the object's methods, and retrieval is impossible without formal requests presented to the object. An object's internal structure is always hidden from other classes. This is what makes object-oriented programming more truly modular than earlier structured programming.

In the fund accounting application class structure design, each table represents a data structure of a particular class. The tables are to be accessed only by their class methods. However, financial report classes which acquire data only during run time and which do not have permanent data structures of their own are not assigned any tables. The journal entry class is directly related to the user interface. It is responsible to capture and verify transaction data. The journal entry class then takes the responsibility for recording the transactions in its three journal entry subclasses.

A class may require more than one table to describe its various aspects. That means, one object may have multiple normalized group of attributes. In the data modelling process conducted during object structure analysis, all object attributes have been identified. The attributes of each table must be related to the object identifier in a correctly normalized manner.

An attribute of a class is a variable representing the attribute of object instances that belong to the class. However a super class may not contain any attribute if the subclasses do not have identical data structure. The similarity of operations performed in the subclasses may be the justification for bringing them under a superclass. The balance sheet account class is an abstract class that contains the attributes or instance variables of the three subclasses. But the three subclasses of balance sheet will have different methods for updating their balances. The accounts in the expenditure class have similar data structure and operations such as posting and provide information to those classes responsible for generation of reports.

The requirement specification for transaction data collection varies with the type of account affected. More data must be captured for a cash transaction than for a noncash transaction. Hence the structure of each of the journal entry classes is different. Consequently, a journal entry class is a superclass of the three subclasses. In summary, the attributes of a class must be those data elements that will be used in the operations of the class.

The data elements described in group (b) for each class are variables to be computed by respective class methods. The description of data structures of the classes of the general ledger and financial reporting subsystem is provided below. The key fields contain the '#' symbol.

(1) CLASS: BALANCE SHEET ACCOUNT

Description: It is a superclass that controls updating of balance sheet accounts and that summarizes all types of balance sheet accounts. It generates two reports: balance sheet and trial balance

Subclasses: asset account, liability account, and fund-balance account

DATA : #Account No.
Account Description

(2) CLASS: ASSET ACCOUNT

Description: It manages the updating process for asset accounts

Superclass: Balance Sheet Account

Data (a) : #Asset Account No
Asset Account Balance

(b) : Total Assets

(3) CLASS: LIABILITY ACCOUNT

Description: It manages the updating process for liabilities

Superclass: Balance Sheet Account

DATA (a) : #Liability Account No
Liability Account Balance

(b) : Total liabilities

(4) CLASS: FUND-BALANCE ACCOUNT

Description: It manages the updating process for its balance by using methods inherited from the Balance Sheet Account Class and modified

Superclass: Balance sheet Account

DATA : Fund-Balance Account No
Fund-Balance Account Balance

(5) CLASS: EXPENDITURE ACCOUNT

Description: It is a superclass that updates and consolidates expenditure figures for reporting purposes

Subclasses: First Month Expenditure Account
Second Month Expenditure Account
Third Month Expenditure Account
Previous Quarter Cumulative Expenditure Account

DATA (a) : #Expenditure Account No
Expenditure Account Description
(b) : Expenditure Account Cumulative Disbursement to date for a project site
Sectoral Cumulative Expenditure Totals

(6) CLASS: FIRST MONTH EXPENDITURE ACCOUNT

Description: It is a repository of data for expenditures of 14 sites for the first month of the current quarter

Superclass: Expenditure Account Class

DATA (a) : #Expenditure Account Number
Jijiga
Hartishiek

Darror
 Rabasso
 Kamaboker
 Kebribeyah
 Derowonaji
 Teferiber
 Aysha
 Gambella
 Pugnido
 Dimma
 Bonga
 Addis abeba

- (b) : Total Expenditure for Project Site
 Expenditure Account Consolidated Total for
 all sites
 Sectoral Monthly Expenditure Total for Site

(7) CLASS: SECOND MONTH EXPENDITURE ACCOUNT

Description: It is a repository of data for expenditures
 of 14 sites for the second month of the
 current quarter

Superclass: Expenditure Account Class

DATA (a) : #Expenditure Account Number and 14 site names
 as described above

- (b) : Total Expenditure for Project Site
 Expenditure account consolidated total
 Sectoral Monthly Expenditure Total for Site

(8) CLASS: THIRD MONTH EXPENDITURE ACCOUNT

Description: It is a repository of data for expenditures
 of 14 sites for the third month of the
 current quarter

Superclass: Expenditure Account Class
DATA (a) : #Expenditure Account Number and 14 site names
as described for First Month Expenditure
Account Class
(b) : Total Expenditure for Project Site
Expenditure account consolidated total
Sectoral Monthly Expenditure Total for Site

(9) CLASS: PREVIOUS QUARTER CUMULATIVE EXPENDITURE ACCOUNT

Description: It is a repository of data for cumulative
expenditures of 14 sites for one or more past
quarterly periods

Superclass: Expenditure Account Class
DATA (a) : #Expenditure Account Number and 14 site names
as described for First Month Expenditure
Account Class
(b) Total Expenditure for Project Site
Expenditure account consolidated total
Sectoral Monthly Expenditure Total for Site

(10) CLASS: JOURNAL ENTRY

Description: It is a superclass that verifies the accuracy
of source documents and the debit credit
equality of financial transactions

Subclasses: Cash Debit Entry class, Cash Credit Entry
class, Expenditure Debit Entry Class, General
Debit entry Class, and General Credit Entry
class

DATA : Transaction Type
Source Document Type
Total Debit Amount of a Transaction
Total Credit Amount of a Transaction

(11) CLASS: CASH ENTRY

Description: A class that stores complete data about the cash receipt and payment transactions and generates a cash flow report on monthly basis

Superclass: Journal Entry Class

DATA (a): Transaction Date

#Check No or Receipt Number

Payer or Payee Name

Cash Account No Debited or Credited

Amount of cash Debited or Credited

(b): Total Monthly Debits to a Particular Cash Account

Total Monthly Credits to a Particular Cash Account

(12) CLASS: EXPENDITURE DEBIT ENTRY

Description: A journal entry that stores expenditure transaction details for all project sites

Superclass: Journal Entry Class

DATA (a): #Project Site Name

#Expenditure Account No

#Transaction Reference

Transaction date

Amount of Expenditure

(b): Total monthly debits to an expenditure account

(13) CLASS: GENERAL ENTRY

Description: A journal entry that stores transaction details of all account except cash receipts and budget disbursements

Superclass: Journal Entry Class
 DATA (A): #Transaction Reference
 #Account No Debited or Credited
 Transaction date
 Amount Debited or Credited
 (b): Total debited to an account during the month
 Total credited to an account during the month

(14) CLASS: EXPENDITURE BUDGET

Description: A class containing detailed budget data for all refugee projects managed by ARRA. It uses three tables for storing budget allocation and description details. It is responsible to prepare two reports that show the budget allocations, the actual expenditures, and the discrepancies at a project site level and at a subproject level.

Superclass: Expenditure Account Class
 DATA (a): 1. #Expenditure Account No
 14 columns for sites names
 2. #Budget Sector code
 Budget Sector Description
 3. #Project Site
 Subproject
 (b): Total Expenditure Budget for Project Site
 Total Expenditure Budget for Subproject
 Total Sectoral Expenditure Budget for Site
 Total Sectoral Expenditure Budget for Subproject

(15) CLASS: FUND TRANSFER ACCOUNT

Description: The Fund Transfer Account Class maintains detailed data for cash movements from head office to sites, and generates a monthly statement that shows the difference between monthly transfers and actual cash outlay

DATA (a): #Expenditure Account No

14 columns for project sites names

(b): Total monthly fund transfer for a project site

(16) CLASS: USER INTERFACE

Description: It is a collection of window objects that afford communication between the classes and the human users of the system

(17) CLASS: PRINTER

Description: A utility class that produces hard copies of reports; hardware component of the system

(18) CLASS: USER

Description: A class that controls access and allows only authorized users to run the system

DATA : Username

Password

4.3 DESIGNING METHODS

4.3.1 ASSIGNING CLASS RESPONSIBILITIES

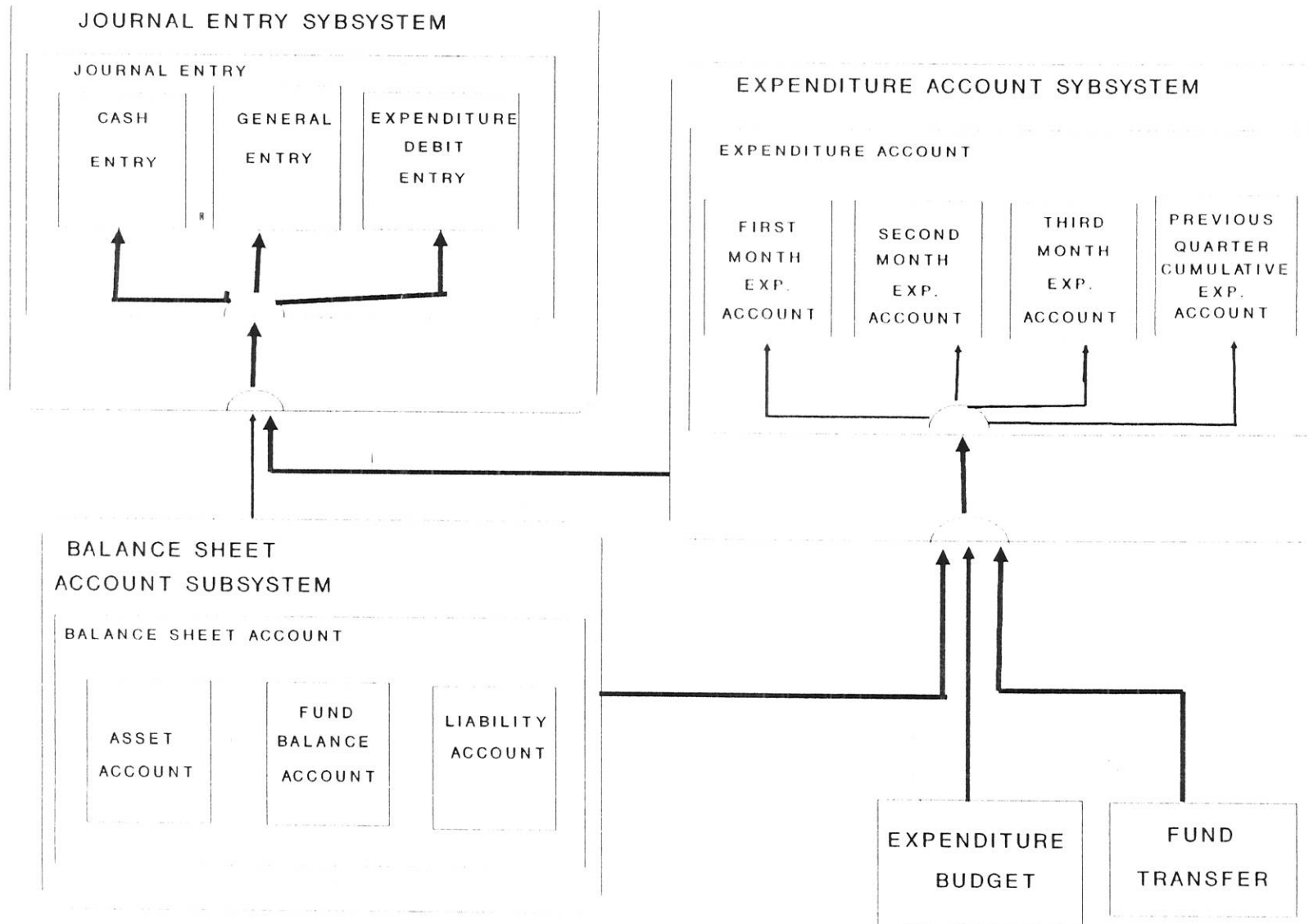
Methods are procedural specifications of an operation or functions of a class. Classes implement three basic types of methods: methods that perform system maintenance functions, such as initializers; methods that provide access to objects data or its services; and methods that implement functions specific to the object. The method that provides access to object data or its services is known as a public method, while the method that executes operations not available to other classes is called private method. An operation is a process that can be requested. This means a method is executed only when a server object receives a message from a client object. The operations provided in the form of methods are the appropriate means of fulfilling systems requirement in a system built of objects. Therefore, the detailed description of processes furnished in the previous chapter should now be assigned to the respective classes of the general ledger fund accounting system.

A method of a particular class is solely responsible for manipulating data in that class. For example, the account balances in asset accounts are updated only by a method that belongs to the asset account class or its superclass, the balance sheet account class. No class has direct access to the data structure of another class. If this requirement is not observed encapsulation is violated. However, in a fund accounting system financial reports must be produced by using data stored in different classes.

The need to collect data from different classes requires sending messages to each class so that each class supplies the data. Consequently, this client server kind of relationship is prevalent in object-oriented systems. Some of the responsibilities of the class can be grouped together. This grouping of responsibilities implies grouping of requests to which a server object must respond. The group of requests asking for the services of a class is referred to as a contract. For instance, the balance sheet class requests total expenditure figure from the expenditure account class to produce a balance sheet. The protocol description of the client object's request and the server object's contractual responsibility are the same so that messages flow easily between objects.

The contracts represent public responsibilities of an object, whereas the operations that can not be requested by other objects are its private responsibilities. Updating the balance of an expenditure account of a given class is a private responsibility of that class. Public responsibilities and contracts are formally incorporated in the design of the class methods. A collaboration diagram or a class communication diagram of the general ledger subsystem is depicted in the next page to show the built-in contractual relationships among classes and subsystems (Diagram 4.3.11).

DIAGRAM 4.31 CLASS COMMUNICATION DIAGRAM
FOR THE GENERAL LEDGER AND FINANCIAL REPORTING SUBSYSTEM



A complete description of the responsibilities, i.e. operations to be performed by classes or to be requested from others, is listed for all classes in the following section.

CLASS: USER	
RESPONSIBILITIES	COLLABORATORS
a.get authorized user name	USER INTERFACE
b.verify user name	
c.get user password	
d.verify user password	
e.allow user to proceed	

CLASS: BALANCE SHEET ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.get cash debit entry total by account	JOURNAL ENTRY
b.get cash credit entry total by account	JOURNAL ENTRY
c.get general debit entry total by account	JOURNAL ENTRY
d.get general credit entry total by account	JOURNAL ENTRY
e.get grand sectoral total of expenditures	
f.compute a new fund-balance account balance	EXPENDITURE ACCT
g.verify the quality of debits and credits	
h. print balance sheet	PRINTER
i. print the trial balance	PRINTER

CLASS: ASSET ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.update asset account balances b.compute the total of assets	Balance sheet account

CLASS: LIABILITY ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.update liability account balances b.compute the total of liabilities	BALANCE SHEET ACCOUNT

CLASS: FUND-BALANCE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a. Upcate fund-balance account balance	BALANCE SHEET ACCOUNT

CLASS: EXPENDITURE DEBIT ENTRY	
RESPONSIBILITIES	COLLABORATORS
a. compute the total amount debited to an expenditure account during the month b. compute the total amount of monthly expenditure recorded	JOURNAL ENTRY

CLASS: EXPENDITURE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.get expenditure debit entry total by account b.check month of expenditure c.update monthly expenditure account balances d.update previous quarter cumulative expenditure account balances (i)summarize the monthly amounts in three types of expenditure accounts by site (ii)add the above figure to cumulative previous quarter balances of site (iii) verify the updating process e. compute the total expenditure for site f.compute a combined sectoral expenditure total for site g. compute grand sectoral totals for site h. compute a combined sectoral expenditure total for subproject i. compute grand sectoral total for subproject j.supply monthly expenditure data for site k.supply previous quarter cumulative expenditure totals for sites and subprojects	JOURNAL ENTRY FIRST, SECOND, THIRD, AND PREV.QUARTER CUMULATIVE EXPENDITURE ACCOUNTS

CLASS: FIRST MONTH EXPENDITURE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.compute first month site total expenditure b.compute first month subproject total expenditure c.compute first month sectoral expenditure totals for a project site d.compute first month sectoral expenditure totals for a subproject	EXPENDITURE ACCOUNT

CLASS: SECOND MONTH EXPENDITURE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.compute second month site total expenditure b.compute second month subproject total expenditure c.compute second month sectoral expenditure totals for a project site d.compute second month sectoral expenditure totals for a subproject	EXPENDITURE ACCOUNT

CLASS: THIRD MONTH EXPENDITURE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.compute third month site total expenditure b.compute third month subproject total expenditure c.compute third month sectoral expenditure totals for a project site d.compute third month sectoral expenditure totals for a subproject	EXPENDITURE ACCOUNT

CLASS: PREVIOUS QUARTER CUMULATIVE EXPENDITURE ACCOUNT	
RESPONSIBILITIES	COLLABORATORS
a.compute previous quarter cumulative expenditure total for site b.compute previous quarter cumulative expenditure total for project site c.compute previous quarter cumulative expenditure sectoral totals for project site d.compute previous quarter cumulative expenditure sectoral totals for subproject	EXPENDITURE ACCOUNT

CLASS: JOURNAL ENTRY	
RESPONSIBILITIES	COLLABORATORS
a.get debit entry	USER INTERFACE
b.verify the debit entry details	
c.get credit entry	USER INTERFACE
d.verify credit entry details	
e.verify the equality of debits and credits	
f.write a debit entry	
g.write a credit entry	

CLASS: CASH ENTRY	
RESPONSIBILITIES	COLLABORATORS
a. compute the total amount debited to each cash account for the month	JOURNAL ENTRY
b. compute the total amount credited to each cash account for the month	
c. get beginning cash balances	ASSET ACCOUNT
d. compute total receipts and payments for the month	
e. compute ending cash balances	PRINTER
f. print cash flow statement	

CLASS: GENERAL ENTRY	
RESPONSIBILITIES	COLLABORATORS
a. compute the total monthly debits to any account different from cash and expenditure b. compute the total monthly credits to any account different from cash or expenditure c. compute a net debit or credit to an account	JOURNAL ENTRY

CLASS: BUDGET	
RESPONSIBILITIES	COLLABORATORS
a. get current quarter and previous quarter expenditure details for each site b. get current quarter and previous quarter expenditure details for each subproject c. compute the difference between annual budget and cumulative disbursement to date for each account d. compute intermediate sectoral totals e. compute grand columnar totals f. verify grand columnar totals g. print project site financial report h. print subproject financial report	EXPENDITURE ACCOUNT EXPENDITURE ACCOUNT PRINTER

CLASS: FUND TRANSFER	
RESPONSIBILITIES	COLLABORATORS
a. get expenditure fund transfer entry for site	USER INTERFACE
b. write expenditure fund transfer entry	
c. update expenditure fund transfer balance	
d. Compute total monthly site fund transfer	
f. get current month expenditure details for site	EXPENDITURE ACCT
g. compute the difference between fund transferred and actually expended	
h. verify the fund transfer and expenditure comparison report	
i. print the report	PRINTER

4.3.2 CREATING METHODS

The responsibilities of classes are accomplished by executing class methods. Thanks to computer-aided-software-engineering (CASE) tools, nowadays program codes could be automatically generated without hard-coding. The prototype software codes for this application are partially generated with the help of form designers and report generators.

According to Mark Lorenze(1993), the average number of methods per class should be less than 20. In the fund accounting application being designed, the maximum number of methods is twelve and these methods are contained in the

expenditure account class. The assignment of too many methods results in placing too much responsibility on a single class. To avoid further complication of the expenditure account, 4 other subclasses of expenditure account are discovered and established as per the reporting requirements. Generally, it is unnecessary to build complex classes since the object-oriented languages support small classes that provide only a single type of service.

To ensure that the methods design for this application are consistent with object-oriented , methods are designed to perform relatively simple and self-contained operations. This approach makes methods easier to test, debug, and update. It eliminates undesirable coupling among methods. Since the methods are designed to perform single tasks they represent highly cohesive module units. According to Marz Lorenze (1993), the average method size should be less than 8 lines of code for Smalltalk and 24 lines of code for C++.

The account balance updating methods for assets and liabilities are based on similar logic, but with different implementation procedures. A debit amount is an addition to an asset account balance but it is a deduction form a liability balance. The same method name could be used for updating both types of accounts. An updating message may be sent by the balance sheet account (super class) to the two subclasses without additional descriptions and the subclasses execute the methods correctly. This is what is known as polymorphism. Without writing new message codes, identical messages for activating methods could be executed in different ways by different class instances. This is one example of maximization of code reusability.

The methods for the general ledger and financial reporting classes are not coded in any object-oriented language, because studying any one of such languages and producing codes in a three month time is an undertaking that is difficult to accomplish.

4.4. DESIGNING USER INTERFACE

The user interface of an object-oriented application is event driven, where the user interacts with visible objects. User interface classes are part of the application. The separation of the user interface from the model is important for reusability. Ideally, an application provides an easy-to-use interface for authorized users to make authorized requests with valid and accurate data. A user friendly interface of an application provides informative and helpful error messages to authorized users who make unauthorized requests or mistakes, or who attempt to enter invalid data. Ideally, an application prevents unauthorized users from accessing the database.

Event-driven interface reflect the way people work in the real world. The multi tasking feature of a windows-based graphic user interface allows users to run different programs at the same time. A user need not perform all steps without break to finish a certain task. Instead of the application restricting the users pace of work, the user is given freedom in determining when to begin and when to stop performing each step in using the application. The user is not required to learn a complex hierarchy of menu choices. The user doesn't need to learn how to perform tasks with the change in menu screens.

An object-oriented application uses familiar objects such as push buttons and list boxes, which are consistent across all application windows. The object oriented windows interface is composed of controls or buttons that respond to users actions.

The controls recognize events such as mouse clicks or key presses. In response to events, event handlers are executed. Event handlers are the program codes that are activated by mouse clicks and key presses. In dBASE for windows the user interface objects are attached to form windows. The form windows contain data entry forms, dialog boxes and menus.

To be usable, an application must serve three functions:

1. prints, queries, and updates records
2. allows the user to direct and control the processing of the application;
3. maintains the security and integrity of the database at all times;

4.4.1 PRINT, QUERYING AND UPDATING THE DATABASE

The first function of an object-oriented financial database application is to print, query, and update information stored in the database records. When a user enters a query , the application must construct the answer from data stored one or more tables in the database. I have included query and print features compatible with the users needs. The users view of information stored in the database is materialized through screen outputs and printed forms.

Print features included in the user interface specification include the following:

1. Cash receipt journal for cash receipt transactions
2. Cash payment journal for cash payment transaction
3. Subproject Financial Report
4. Project Site Financial Report
4. Trial Balance
5. Balance Sheet
6. Cash transfer versus actual expenditure report
7. Cash flow statement

Query and edit features provided include:

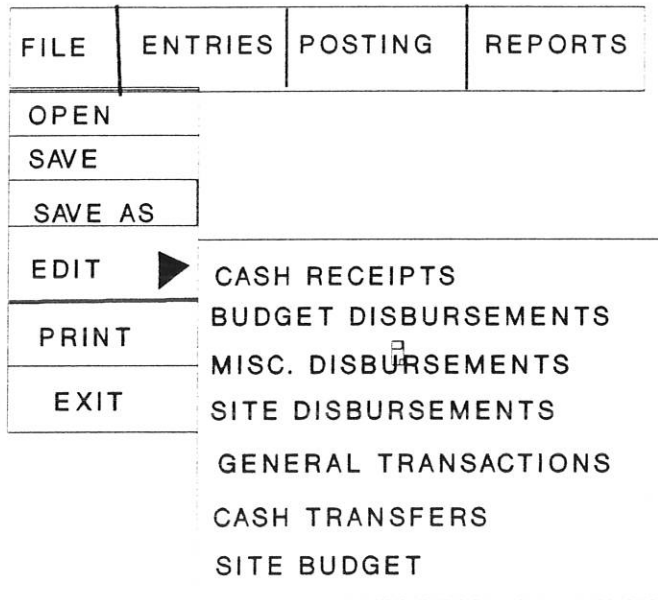
1. browsing the cash receipt transactions
2. browsing the cash payment transactions
3. browsing the fund allocations and expenditure account details
4. browsing the financial reports

Updating generally refers to changing the data stored in the database. The major updating function, however, is directly related to the addition of new data to the existing records in the database. To fulfil the updating requirements, a separate set of data entry form objects have been included for the following items:

1. budget allocations
2. Cash receipt transactions
3. cash payment transactions
4. budget disbursement transaction taking place at sites
5. other noncash (nonbank) transactions

DIAGRAM 4.41 ARRA FUND ACCOUNTING MENU

(A)



(B)

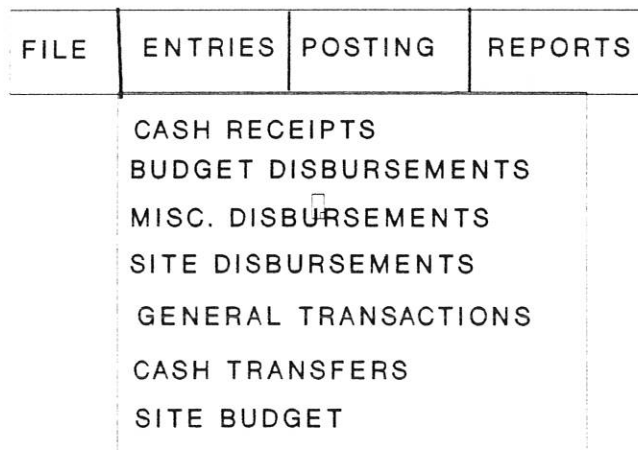
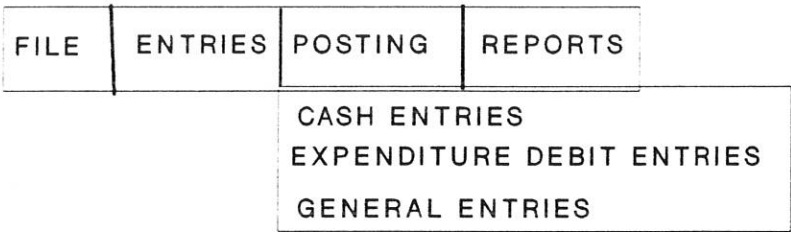
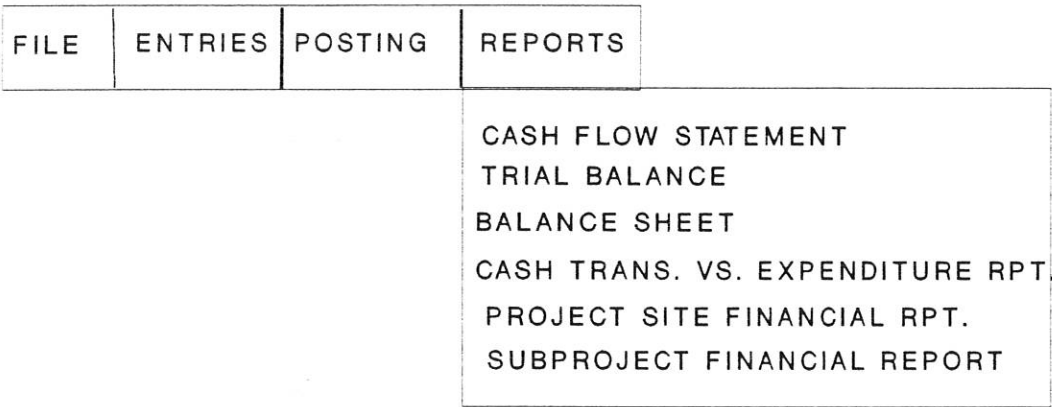


DIAGRAM 4.41 HIERARCHY OF MENUS FOR THE FUND
ACCOUNTING SYSTEM OF ARRA

(C)



(D)



4.4.2 ENDUSER DIRECTION

The second characteristics of an object oriented database application is that it allows the user to direct and control the application. An application normally include several features that the user can invoke. An application must be designed so it is easy for the user to conclude or suspend one feature and initiate another one. Therefore, this object oriented user interface design consists of menus with list of options that could be chosen by the users. The menu structure is designed in accordance with the users' view. For example the major menu contains separate processing options for cash transactions (bank transactions) and nonbank transactions. Ordinarily, transactions are classified into various types for ease of processing in manual systems. To this effect, some seven transaction input screen objects are provided as per the user's classification of transactions. The user is not required to remember any commands he simply presses a key that leads him to an input or an output screen with which he is already familiar. The users are familiar with the input and output forms because this forms are designed as per their requirements.

4.4.3 MAINTAINING THE SECURITY AND INTEGRITY OF THE DATABASE

An application should allow only authorized users to perform only authorized activities using only valid data. In the data dictionary the data type of data items and their storage size is described. The database has built-in control mechanism to allow only appropriate data types to be entered in the data entry forms. For example, if one enters letters

in the date column, the system automatically generates an error message. The data entry fields are adjusted to accept only letters where a user may wrongly add numerical data in a character field. Therefore, this feature partly prevents entering wrong information in the database. The account numbers to be entered during journalizing transactions are also well designed. A number must be within a given range to be acceptable as valid account number. A message that notifies the user to enter valid account number is displayed.

As the system in question is a financial system it is a system most susceptible to unauthorized access attempts. Therefore pass word class is included in the application. Unauthorized users will not be in a position to operate the system.

The use of transaction specific data entry screens with specific account types and source document designations protect the database from invalid data and ensure the integrity of stored data. For example, bank accounts are accessed only through two data entry screens for cash receipts and cash payments. The cash receipt screen needs the user to input receipt number before he input the cash account (bank account) number. By the same token, no account number data can be input for a cash payment before the user inputs a valid check number that must be verified from the stored information in the check number file in the database.

The type of accounts to be debited or credited for a particular transaction are most of the time specified in the data entry screens. Since the user has separate screens for different transactions, he or she has a minimal chance of

entering wrong entries. In summary, a journal entry will be recorded in transaction files only after the account numbers specified by the user in the input screens are found to be correct as per the application instructions.

The application will also check the equality of the debit and the credit amounts of a transaction before the data is saved on the files. This enables to maintain the integrity of database by ensuring duality of the overall debit and credit balances. The system will also give help messages and error messages to users to direct users.

The menus, data entry forms, and report forms designed for the general ledger financial reporting subsystem are shown from page 125-139.

DIAGRAM 4.42 DATA ENTRY FORMS

(A) CASH RECEIPT TRANSACTION

FILE	ENTRIES	POSTING	REPORTS
CASH RECEIPTS BUDGET DISBURSEMENTS MISC. DISBURSEMENTS SITE DISBURSEMENTS GENERAL TRANSACTIONS CASH TRANSFERS SITE BUDGET			

CASH RECEIPT TRANSACTION

TRANSACTION DATE: DD/MM/YYYY

RECEIVED FROM: XXXXXXXXXXXXXXXX

TRANSACTION REFERENCE: CRV/NNNNNN

CASH ACCOUNT NO DR: NNNNN AMOUNT: NNNNNNN.NN

ACCOUNT NO CR: NNNNN AMOUNT: NNNNNNN.NN

DIAGRAM 4.42 DATA ENTRY FORMS

(B) BUDGET DISBURSEMENT TRANSACTION

FILE	ENTRIES	POSTING	REPORTS
CASH RECEIPTS BUDGET DISBURSEMENTS MISC. DISBURSEMENTS SITE DISBURSEMENTS GENERAL TRANSACTIONS CASH TRANSFERS SITE BUDGET			

BUDGET DISBURSEMENTS TRANSACTION

TRANSACTION DATE: DD/MM/YYYY

PAYEE NAME: XXXXXXXXXXXXXXXX

TRANSACTION REFERENCE: CK/NNNNNN

CASH ACCOUNT NO. CR NNNNN AMOUNT NNNNNNN.NN

SITE NAME: XX

EXPENDITURE ACCOUNT NO. X.NN.X.NNNNN

AMOUNT: NNNNNNN.NN

DIAGRAM 4.42 DATA ENTRY FORMS

(C) MISCELLANEOUS DISBURSEMENTS

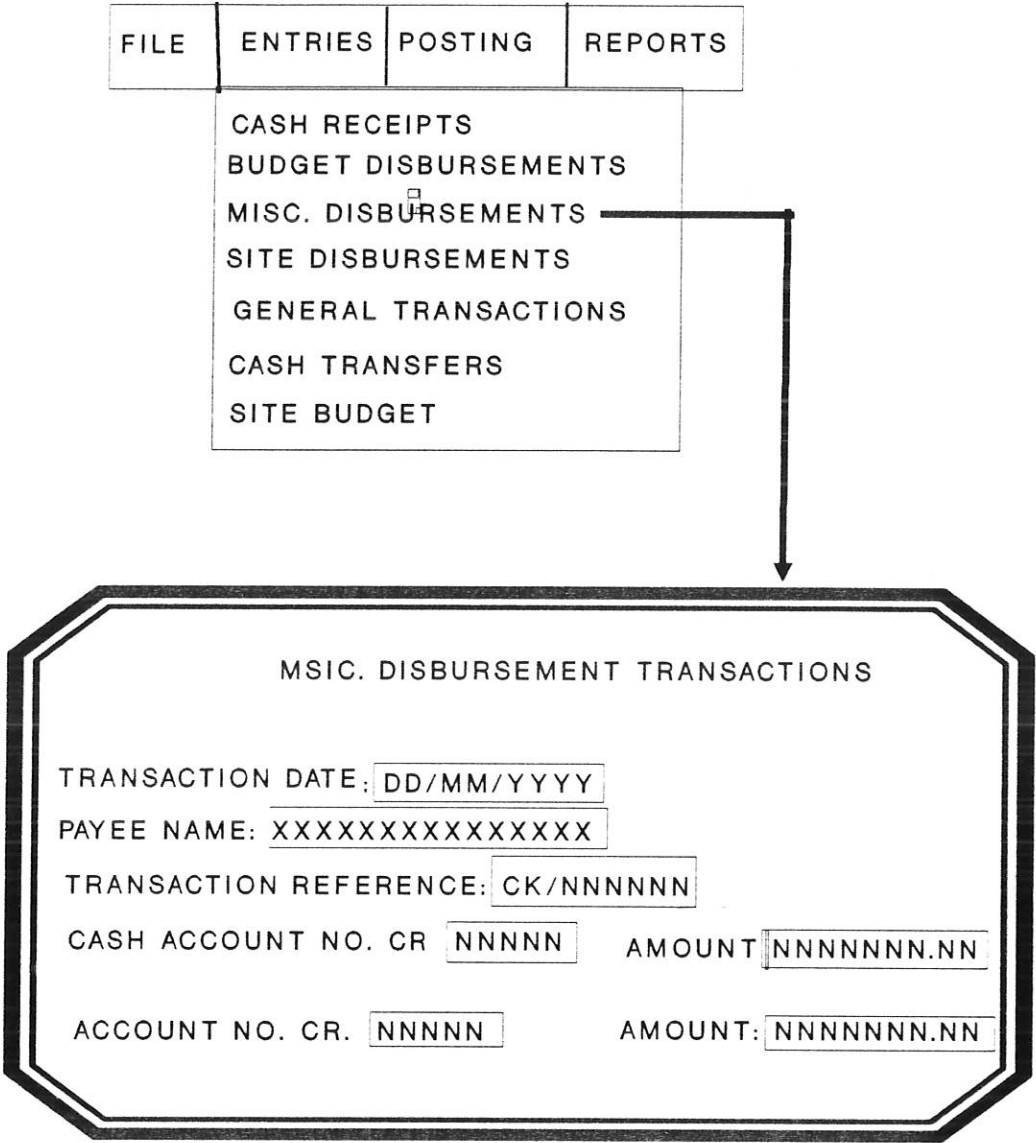


DIAGRAM 4.42 DATA ENTRY FORMS

(D) SITE DISBURSEMENT TRANSACTIONS

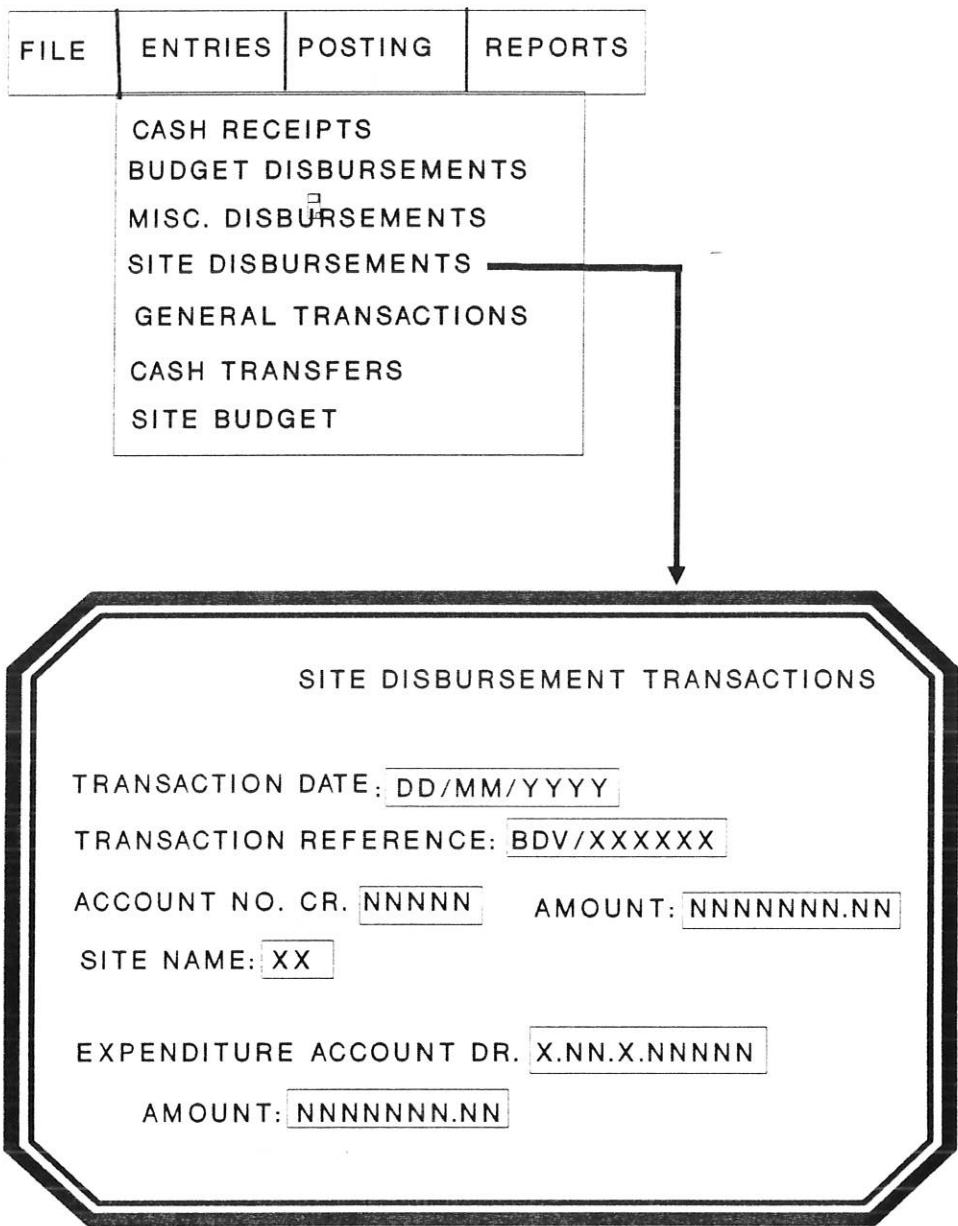


DIAGRAM 4.42 DATA ENTRY FORMS

(E) GENERAL TRANSACTIONS

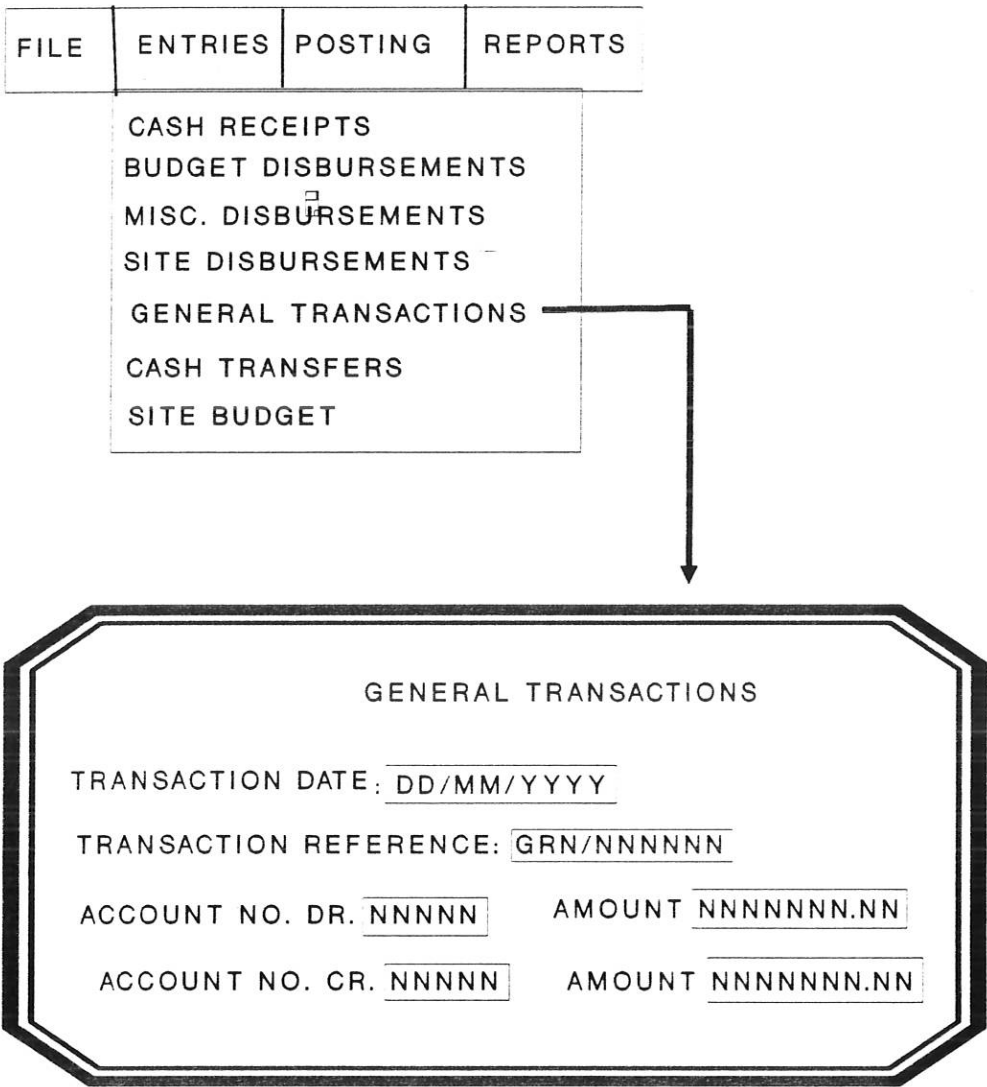


DIAGRAM 4.42 DATA ENTRY FORMS

(F) FUND TRANSFER TRANSACTION

FILE	ENTRIES	POSTING	REPORTS
	CASH RECEIPTS BUDGET DISBURSEMENTS MISC. DISBURSEMENTS SITE DISBURSEMENTS GENERAL TRANSACTIONS CASH TRANSFERS SITE BUDGET		

FUND TRANSFER TRANSACTION

ACCOUNT NO	SITE A - AMOUNT	SITE B - AMOUNT

(G) ANNUAL BUDGET ALLOCATION



4.43 FINANCIAL REPORT FORMS

(A)

FILE	ENTRIES	POSTING	REPORTS
------	---------	---------	---------

CASH FLOW STATEMENT
TRIAL BALANCE
BALANCE SHEET
CASH TRANS. VS. EXPENDITURE RPT.
PROJECT SITE FINANCIAL RPT.
SUBPROJECT FINANCIAL REPORT

CASH FLOW STATEMENT
FOR THE MONTH OF XXXXXX 19 NN

BANK ACCOUNT NO. NNNNN

BAL. AT BEG. OF MONTH: NNNNNNN.NN

DISBURSEMENTS

DEPOSITS

TOTAL NNNNNNN.NN NNNNNNN.NN

BALANCE AT THE END OF MONTH NNNNNNN.NN

4.43 FINANCIAL REPORT FORMS

(B)

FILE	ENTRIES	POSTING	REPORTS
------	---------	---------	---------

CASH FLOW STATEMENT
CASH TRANS. VS. EXPENDITURE RPT.
TRIAL BALANCE
BALANCE SHEET
PROJECT SITE FINANCIAL RPT.
SUBPROJECT FINANCIAL REPORT

CASH TRANSFER VERSUS ACTUAL EXPENDITURE REPORT

PROJRCT SITE NAME; MONTH: 19

EXPENDITURE ACCOUNT NO.	AMOUNT TRANSFERRED	ACTUAL EXPENDITURE	VARIANCE

DIAGRAM 4.43 FINANACIAL REPORTS

(C)

ARRA BALANCE SHEET END OF MONTH : 19

ASSETS

BANK ACCOUNTS:	
ACCOUNT NO.	AMOUNT
TOTAL CASH IN BANK	
PETTY CASH AND CASH TRANSFER ACCOUNTS:	
TOTAL CASH TRNSFER BALANCES:	
ADVANCES AND OTEHR RECEIVABLES:	
TOTAL RECEIVABLES FROM INTERNAL DEBTORS	
RECEIVABLES FROM EXTERNAL SOURCES:	
TOTAL RECEIVABLE FROM EXTERNAL DEBTORS	
TOTAL ASSETS	

LIABILITIES

INTERNAL CREDITORS	
TOTAL OWED TO INTERNAL CREDITORS	
EXTERNAL CREDITORS	
TOTAL OWED TO EXTERNAL CREDITORS	
TOTAL LIABILITIES	
FUND BALANCE	
TOTAL LIABILITIES AND FUND BALANCE.	

DIAGRAM 4.43 FINANCIAL REPORT MENUS
(D)

ARRA

TRAIL BALANCE

DATE

ACCOUNT NUMBER	DEBITS	CREDITS
ASSETS	XXXXXX.XX	
LIABILITIES		XXXXXX.XX
FUND BALANCE		XXXXXX.XX
SECTORAL EXPENDITURE TOTALS	XXXXXX.XX	
TOTAL	XXXXXX.XX	XXXXXX.XX

DIAGRAM 4.43 FINANCIAL REPORT MENUS

(E)

IMPLEMENTING AGENCY: ARRA

SUBPROJECT SYMBOL:

PROJECT LOCATION (ZONE) :

PROJECT SITE:

PROJECT DURATION: REPORTING PERIOD:

DATE:

BUDGET CODE	BUDGET DESCRIPTION	ANNUAL BUDGET	REMAINING BALANCE	CUMULATIVE DISBURSEMENT AND COMMITMENT TO DATE	EXPENDITURE			COMMIT MENTS	CUMULA TIVE DISBUR SEMENT LAST QUARTER
					3RD MONTH	2ND MONTH	1ST MONTH		

DIAGRAM 4.43 FINANCIAL REPORT MENUS

(F)

IMPLEMENTING AGENCY: ARRA

SUBPROJECT SYMBOL:

PROJECT LOCATION (ZONE) :

PROJECT DURATION:

REPORTING PERIOD:

DATE:

BUDGET CODE	BUDGET DESCRIPTION	ANNUAL BUDGET	REMAINING BALANCE	CUMULATIVE DISBURSEMENT AND COMMITMENT TO DATE	EXPENDITURE			COMMIT MENTS	CUMULA TIVE DISBUR SEMENT LAST QUARTER
					3RD MONTH	2ND MONTH	1ST MONTH		

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

This study has attempted to demonstrate how an object-oriented analysis and design of a computerized fund accounting system is carried out with particular reference to the budget and finance control department of ARRA. Organization background information was provided to describe the parent system of which budget and finance control department is a part. The problems that spurred this thesis project were described as existing system inefficiencies, structural and functional system defects, and organizational automation decisions. An evaluation of project feasibility was made by pointing out how ARRA and its employees are committed to the realization of the objectives of this project.

The general and specific objectives were outlined to show the scope of the project. Some tasks which are part and parcel of a software development project were not undertaken due to lack of time and expert support. Justifications were provided for methodologies used in requirements specification, further systems analysis, and complete design of the general ledger and financial reporting subsystem of ARRA's fund accounting system.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 CONCLUSION

This study has attempted to demonstrate how an object-oriented analysis and design of a computerized fund accounting system is carried out with particular reference to the budget and finance control department of ARRA. Organization background information was provided to describe the parent system of which budget and finance control department is a part. The problems that spurred this thesis project were described as existing system inefficiencies, structural and functional system defects, and organizational automation decisions. An evaluation of project feasibility was made by pointing out how ARRA and its employees are committed to the realization of the objectives of this project.

The general and specific objectives were outlined to show the scope of the project. Some tasks which are part and parcel of a software development project were not undertaken due to lack of time and expert support. Justifications were provided for methodologies used in requirements specification, further systems analysis, and complete design of the general ledger and financial reporting subsystem of ARRA's fund accounting system.

The detailed system requirements has been specified and depicted with the help of data flow diagrams and HIPO charts. Included in the requirement specification is a complete listing of sources and types of documents that furnish legitimate input data and detailed rules to be followed in processing the data. The output requirements specification is composed of two previously prepared and four newly added financial reports as per the users' needs.

The object-oriented approach has been used in describing the analysis and design aspects. Object-oriented analysis and design supports a highly structured and modular software design. Modularity is one of the major criteria used in appraising software quality. Therefore, the task of designing classes and class methods has been done by observing the object-oriented design guidelines.

In the object-oriented analysis phase object structures were examined and objects were classified into various categories. Hierarchy charts, composed-of diagrams, and object relationship diagrams were used to depict inheritance relationships, composed-of relationships, and general relationships respectively. The second half of the object-oriented analysis phase dealt with object behavior analysis.

Object behavior analysis required investigating those operations performed by objects, object state changes that resulted from the operations, and the events that trigger objects to perform operations. The relationships among operations, events, and state changes show how a functional requirement is fulfilled in an object-oriented system. In

summary, object behavior analysis was illustrated in chapter three with the help of state transition diagrams and operation diagrams.

In the object-oriented design phase, the object structure analysis continued with an implementation focus and selected object categories were designated as classes. Data structures were designed by identifying the attributes that are required for carrying out class methods or responsibilities.

The second aspect of the design phase proceeded with the description of methods for designated classes. Classes exchange messages to request services and to perform services according to valid requests. The grouping and flow of messages in the fund accounting system was depicted in a class communication diagram.

The last aspect of the object-oriented design entailed creating user interface structure. System users activate class methods and get things done by using interfaces of classes. Menus are designed to guide users and provide access to all class methods. Most of the interfaces to classes are implemented through form windows that allow updating, editing, and querying the class databases. Dialog boxes, pushbuttons, and other interface objects are included to enhance the user friendliness of the interface subsystem.

One of the findings of this study is that an object-oriented approach to systems development almost removes the communication gap between users and developers. Unlike former structured methodologies, an object-oriented approach names

objects using the problem domain language. Since the design descriptions of processing requirements are directly associated with the objects, users are able to understand and comment on systems design. The basic features of the design could be described fully without flowcharts, structure charts, and data flow diagrams. Early user sign-offs are now postponed upto a level where coding is started.

Objects in a financial system have three major features. A financial system is characterized by intensive data entry activities. A financial system is composed of permanent objects called accounts, and account balances are frequently increased and decreased. The third feature is that all or a portion of the attributes of one or more account objects are used to generate reports. These three features describe the less volatile and stable characteristics of any financial system. Hence, it is possible to design and code standard classes with customizable methods and put them in a class library. This library will be composed of data entry form classes, posting or updating function classes, and data summary and report creation classes. It is possible then to create a contractual client-server relationship between financial objects and the standard classes in the library. The use of class libraries facilitates reusability of codes and functions. The creation of reusable codes and the use of this standard elements through inheritance is one major advantage of object-oriented analysis and design.

5.2 RECOMMENDATIONS

Based on requirement specifications and the findings of object-oriented analysis and design exercise, the following recommendations are made to aid in further development and implementation of the fund accounting system.

- (1) The object-oriented approach is advised to be used for reviewing this work before any implementation attempt is made. The object oriented approach is also recommended for analysis, design, implementation of the remaining major subsystem, namely the provident fund and payroll subsystem. The object-oriented analysis and design approach to software engineering is very flexible. Classes in an object-oriented system are literally independent of one another. A class can be removed or modified without causing a rippling effect in the system. As system analysis and design is an iterative process, object-oriented design specifications are easily changeable and updatable. By the same token, future software maintenance and upgrading are simplified.
- (2) A team should be formed to undertake the review and implementation of the result of this thesis and the analysis and design of the partly manual payroll and provident fund subsystem. The team should be composed of the chief accountant and a programmer-analyst versed in object-oriented programming. Object-oriented languages allow easy communication among classes of objects and

encapsulation. Global concerns for control flow and data flow within the software system is no more a problem. The use of global variables contradicts with the very essence of object-oriented thinking. Only the protocol for requesting and providing a data or service need to be described in both the client and server classes. Because the methods are encapsulated along with the data attributes of the objects, the application scope of variables and level of complexity of programming is substantially minimized. Hence, the class structure design and methods specification presented in this paper could be easily translated into an object-oriented code in due course.

- (3) In a conventional relational databases system, data is stored in the database without being chained to a particular application. But in a pure object-oriented database management system, a particular data stored in a database is not accessible without using the method of the class to which the data structure belongs. This feature enhances the security of the financial database. Customarily, financial information is the most vulnerable type of information. The reliability of records is improved and tampering with records to conceal embezzlement of financial resources is minimized. Hence, the acquisition of pure object-oriented databases is recommended.
- (4) The budget and finance control department is responsible for preparing reports and custody of all transaction documents and financial reports. According to the requirements imposed by fund providers, all records and

all documents must be kept at the head office and be available for inspection. This requirement presupposes the establishment of a centralized financial system. Field finance offices will continue to send source documents directly to the head office. Therefore, to avoid duplication of work and comply with the aforementioned requirement, a local area accounting network need to be installed at the head office. A network is recommended so as to follow the existing plan of work distribution. Currently, the financial affair of each subproject is handled by a single accountant. This division of labor is satisfactory in that no common responsibilities are shared among the accountants.

BIBLIOGRAPHY

- Aktas, A. Ziya. 1987. ***Structured Analysis and Design of Information Systems***. London: Prentice-Hall.
- Bookholdt, J.L., and David H.Li. 1991. ***Accounting Information Systems: Transaction Processing and Controls***. Homewood:Irwin.
- Blowers, D.S. 1988. ***From Data to Databases***. Van Nos. Reinhold.
- Claire, Chris and Peri Loucopoulos. 1987. ***Business Information Systems***. London: Paradigm.
- Coad, Peter., and Edward Yourdon. 1990. ***Object-Oriented Analysis***. 2nd ed. Englewood Cliffs: Yourdon Press.
- C.Tsichritzis, Dionysios. 1982. ***Data Models***. Englewood Cliffs: Prentice-Hall.
- Davis, B. Gordon. 1988. ***Management Information Systems: Conceptual Foundations Structure and Development***.
- Downess, P.M. 1989. ***Practical Data Analysis***. Blenheim Online Publications.

BIBLIOGRAPHY

- Aktas, A. Ziya. 1987. ***Structured Analysis and Design of Information Systems***. London: Prentice-Hall.
- Bookholdt, J.L., and David H.Li. 1991. ***Accounting Information Systems: Transaction Processing and Controls***. Homewood:Irwin.
- Blowers, D.S. 1988. ***From Data to Databases***. Van Nos. Reinhold.
- Claire, Chris and Peri Loucopoulos. 1987. ***Business Information Systems***. London: Paradigm.
- Coad, Peter., and Edward Yourdon. 1990. ***Object-Oriented Analysis***. 2nd ed. Englewood Cliffs: Yourdon Press.
- C.Tsichritzis, Dionysios. 1982. ***Data Models***. Englewood Cliffs: Prentice-Hall.
- Davis, B. Gordon. 1988. ***Management Information Systems: Conceptual Foundations Structure and Development***.
- Downess, P.M. 1989. ***Practical Data Analysis***. Blenheim Online Publications.

- Eliason, Alan L. 1987. ***Online Business Computer Applications***.
2nd ed. New York: Macmillan.
- Faison, Ted. 1994. ***Borland C++ Object-Oriented Programming***.
3rd ed. Sams Publishing.
- Gilbert, Philip. 1983. ***Software Design and Development***. SRA, Inc.
- Hicks, James O. Jr. and Wayne E. Leininger. 1986. ***Accounting Information Systems***. 2nd ed. St. Paul: West Pub. Co.
- Jackobson, Ivar, et.al. 1992. ***Object-Oriented Software Engineering: A Usecase Driven Approach***. New York: McGraw-Hill.
- Korth, F. Henry and Abraham Silberschatz. 1986. ***Database Systems Concepts***. New York: McGraw-Hill.
- Kroenke, David M. and Kathleen A. Dolan. 1988.
Database Processing: Fundamentals, Design, Implementation.
3rd ed. New York: Macmillan.
- Larson, E. John and A. N. Mosich. 1989. ***Modern Advanced Accounting***. 4th ed. New York: McGraw Hill.
- Lorenz, Mark. 1993. ***Object-Oriented Software Development: A Practical Guide***. Englewood Cliffs: PTR Prentice Hall
- Martin, James. 1985. ***Diagramming Techniques for Analysts and Programmers***. Englewood Cliffs: Prentice Hall

- Martin, James. 1993. ***Principles of Object-Oriented Analysis and Design***. Englewood Cliffs: PTR Prentice Hall
- Model, Martin E. 1988. ***A Professions Guide to Systems Analysis***. New York: McGraw Hill.
- Mullin, Mark. 1990. ***Rapid Prototyping for Object-Oriented Systems***. Reading: Addison-Wesley Publishing Co, Inc.
- PC Computing. March 1995.
- Parkin, Andrew. 1991. ***Systems Analysis***. 2nd ed. London: Edward Arnold Publishers; 1991.
- Pfleeger, Shari Lawrence. 1991. ***Software Engineering: The Production of Quality Software***. 2nd ed. New York: McMillan.
- Philippakis, A.S., and Leonard J. Kazmier. 1983 ***Program Design Concepts With Applications in Cobol***. New York: McGraw_Hill.
- Pinson, Lewis J., and Richard S. Wiener, ed. 1990. ***Applications of Object-Oriented Programming***. Reading: Addison-Wesly Pub.
- Pressman, Roger S. 1987. ***Software Engineering: A Practitioners Approach***. 2nd ed. New York: McGraw-Hill.
- Rowley, Janifer. 1990. ***The Basics of Systems Analysis and Design for Information Managers***. London: Clive Bingley.

Senn, James A. 1984. ***Analysis and Design of Information systems.***

New York: Mc Graw Hill.

UNHCR. 1990. ***Collection of International Instruments Concerning***

Refugees. Geneva.

Vickery, Brian and Alina Vickery. 1989. ***Information Science in***

Theory and Practice. London: Butterworth Co.

APPENDIX I

LIST OF PERSONS INTERVIEWED

- | | |
|----------------------------|---|
| 1. Ato Kore Bawi | Deputy Director of ARRA |
| 2. Ato Daniel Ayele | Head, social services division |
| 3. Ato Zewde Serboro | Head, program formulation and
budget monitoring division |
| 4. Ato Kebede | Head, budget and finance control
dept. |
| 5. W/t Meseret Mengistu | Accountant, budget and fin. cont.
dept. |
| 6. W/o Eyerusalem Cherinet | Accountant, budget and fin.
cont.dept. |
| 7. W/t Beyenech Haile | Accountant, budget and fin. cont.
dept. |
| 8. Ato Getnet Bihonegne | Head, Jijiga zonal finance office |
| 9. Ato Firew Tefera | Accountant, Jijiga zonal finance
office |
| 10. Ato Tadesse Bezabih | Zonal Coordianator, Eastern zone
refugee program, jijiga |

APPENDIX II

INTERVIEW GUIDE FOR ARRA’S OFFICIALS

PART I General Information about the interview

- 1. Interviewee name:
- 2. Interviewee Title:
- 3. The interviewee’s immediate superior
.....
- 4. Date of interview.
Time of interview
Location of interview
- 5. Interview’s job or functional description
.....
- 6. The interviewee’s place in the organizational chart
- 7. Interview objective
.....
.....

PART II INTERVIEW QUESTIONS

1. When did this organization came into existence?

.....

2. What are the objectives of the organization?

.....

.....

3. List the names of organizations that collaborate with ARRA in helping refugees.

a.

b.

c.

d.

e.

f.

g.

h.

4. Describe the nature of relationships between ARRA each of the above mentioned organizations.

.....

.....

.....

5. What are the sources of funds for financing ARRA's activities and for covering its administrative office costs

.....
.....
.....
.....

6. a. Please list down the major types of assistance given
t o r e f u g e e s

.....
.....
.....

b. Please list down the major types of assistance given to
Ethiopian returnees

.....
.....
.....

c. What is the total number of Ethiopian and non Ethiopian
refugees and returnees currently assisted by ARRA?

.....

7. How do you judge the effectiveness of refugee project
management responsibilities undertaken by ARRA in relation
to the financial and human resources at its disposal.

a. The projects are hardly managed as they should be.

Yes [] No []

b. The projects are well managed taking into account the available financial and human resources

Yes [] No []

c. The projects could have been more successful in helping refugees provided that ARRA's managerial and administrative problems are investigated and solved

Yes [] No []

8. List the major departments of ARRA and the names of heads of the respective departments.

.....
.....
.....
.....
.....

9. What, if any, are the major operational problems of ARRA

i).....
ii).....
iii).....
iv).....
v).....

10. According to your observation, which department(s) are facing significant problems? list the names of the departments together with the respective activities or functions where the problems are observed.

.....
.....
.....

11. Is ARRA now ready to engage a consultant to have its illfunctioning departments studied?

Yes [] No []

12. If your answer to the above question is no, what are the reasons for not engaging a consultant now for system studies?

a. Financial (budgetary) problems

Yes [] No []

b. Lack of authorization from superior officials

Yes [] No []

c. The studies are deferred to future time

Yes [] No []

d. Lack of awareness of the necessity of studying the problems for finding solutions

Yes [] No []

13. Did you ever contemplate automating certain functions?

Yes [] No []

If your answer is yes, which activities need to be automated?

- i)
- ii)
- iii)
- iv)
- v)

14. In your capacity as a leader of ARRA you are entrusted the responsibility of ensuring the fulfilment of the organizational objectives of the organization. Do you believe that automation could be taken as one of the major solutions for your operational problems?

Yes [] No []

15. If your answer to the above question is yes, then list the major operations or activities together with the names of the departments in the order of priority for automation.

- 1
- 2
- 3

4

5

16. a) List the number and type of computers you own currently?

.....

.....

.....

b) describe the purposes for which you have committed your existing electronic data process equipment

.....

.....

.....

c) Will the existing data processing equipment be adequate for automating the higher priority areas mentioned above?

Yes [] No []

d) If your answer for the above question is no, do you have sufficient financial resources to acquire additional data processing equipment and software for automating top priority areas and training the employees?

Yes [] No []

18. Do you have qualified manpower that could undertake the analysis and design of a computer information system in the proposed areas?

Yes [] No [].

19. If answered Q.no. 18 no, would you officially authorize me to undertake the analysis and design of a computerized information system in the desired areas as a topic for my thesis?

Yes [] No []

20. If your answer to the above question is yes, would you allow me access to all relevant records and provide all necessary administrative assistance to create a conducive atmosphere for smooth conduct of my research?

Yes [] No []

Name

Signature

Address:

Organization

Tele Fax P.O. BOX.....

APPENDIX III

INTERVIEW GUIDE FOR ARRA'S ACCOUNTANTS

PART I General Information about the interview

1. Interviewee name:
 - b. Interviewee Educational qualification
 - University Education
 - Special training
2. Interviewee Title:
 - Length of experience in years in the current job, in ARRA and other organizations
3. The interviewee's immediate superior
.....
4. Date of interview.
 - Time of interview
 - Location of interview
5. Interviewee's job or functional description
.....
6. The interviewee's place in the organizational chart
7. Interview objective
.....

PART II INTERVIEW QUESTIONS

1. How long have you worked in ARRA ?

Please indicate the job description and the length of experience you have acquired in various posts in ARRA and other organizations.

Job Description

Time

.....

.....

.....

2. Please briefly describe the following questions

a. List down the names of the departments from which you receive source documents and with which you have work relationships.

.....

.....

b. List the names of a department along with the description of a source document(s).

Department

Source Document

.....

.....

.....

3. List down your daily activities?

.....

.....

.....

4. List down other activities you perform less often such as once a week, once a month etc.

.....

.....

.....

.....

.....

5. Please indicate the various permanent records you create and use in your department

.....

.....

.....

.....

.....

.....

6. List the type of journals and the types of transactions or entries you record in these journals, and the source documents of the entries.

[illegible]

7. List the various financial reports you prepare on daily, monthly, quarterly or annual basis.

.....

.....

.....

.....

8. List the names of the departments, if any, to whom you send various reports at regular intervals.

<u>Type of report</u>	<u>Department</u>	<u>No. of copies</u>
-----------------------	-------------------	----------------------

.....		
.....		
.....		
.....		

9. You are currently using one computer in your department.

a. List the type of jobs you do using your computer.

.....

.....

.....

.....

b. List the types of software you use for accounting jobs.

.....

.....

c. how many people are using this computer?

.....

10. What type of records are stored in the computer?

.....

.....

.....

.....

11. Are you satisfied with the ongoing work procedures?

Yes []

No []

12. Are you satisfied with the quality (depth and scope) of work you are performing now?

Yes []

No []

13. Do you use any accounting manuals?

Yes []

No []

14. What are the problems you are currently facing in your work?

.....

.....

.....

15. What improvements do you suggest regarding the quantity and quality of work you are currently performing?

.....

.....

16. According to your experience, what additional activities should be done by using computers?

.....
.....
.....
.....
.....

17. What additional reports or records are necessary taking into account the internal need for controlling efficiency of operations and safeguarding financial resources of ARRA.

.....
.....
.....
.....

APPENDIX IV

INTERVIEW GUIDE FOR

REQUIREMENT ANALYSIS

- A. Type or Description of a transaction or an activity.....
- B. Name of Person performing the activity.....
-
- 1) What other people or employees from another department or within this department receive , provide, or approve information regarding this particular activity or transaction?
-
-
- 2) How does this transaction or activity originate?
-
-
-
- 3) What details or types of record or data store is used in carrying out all the steps required to complete this particular transaction?
-
-

- 4) How long does it take to complete a particular transaction or activity such as report preparation?
.....
- 5) What other people are also directly involved in carrying out this transaction beside you within your department?
.....
.....
- 6) (a)How often do you handle the transaction? How often is the transaction repeated?
.....
(b)How many transactions of that type occur in a day, week, month, quarter semiannually, and annually?
.....
.....
.....
.....
- 7) Who uses the resulting information?
.....
.....
- 8) i) Is each transaction processed and reported individually?
Yes [] No []

ii) Is the transaction processed and reported in batches?

Yes []

No []

iii) In what manner, form, or method are the individually processed or batch processed transactions reported?

.....

9) What type of decisions are made using the transaction information or the reports generated by processing the transactions?

.....

.....

10) In what way or for what purposes are the individual or batch processed transactions used?

.....

.....

.....

11) What information or data is needed to begin performing the first step of the transaction or the activity?

.....

.....

12) i) Why does a transaction occur frequently?

.....

ii) Is the transaction processed and reported in batches?

Yes []

No []

iii) In what manner, form, or method are the individually processed or batch processed transactions reported?

.....

9) What type of decisions are made using the transaction information or the reports generated by processing the transactions?

.....

.....

10) In what way or for what purposes are the individual or batch processed transactions used?

.....

.....

.....

11) What information or data is needed to begin performing the first step of the transaction or the activity?

.....

.....

12) i) Why does a transaction occur frequently?

.....

ii) What causes or triggers the performance of an
a c t i v i t y ?

.....
.....

(a) Is it a request made by an outside party or
employees in another department?

Yes [] No []

(b) Is it triggered by events, i.e., the completion of
a particular transaction?

Yes [] No []

.....
.....

(c) Is it triggered by a passage of time?

Yes [] No []

.....

13) What is the volume of work involved in a single activity?

.....

14) (a) Are there specific performance standards for the
performance of the activity or the transaction?

.....
.....

(b) Who compares performance against standards?

.....

(c) How are processing mistakes caught, discovered, or brought into your attention?

.....

(d) How are errors handled?

.....

.....

(e) Are the errors that occur in the performance of the activity excessive?

.....

.....

15) What steps or procedures are carried out to accomplish a given transaction or task?

.....

.....

.....

16) Are there conditions that can affect how a transaction is handled or activities carried out?

.....

17) What information or report is generated upon completion of this particular transaction or activity?

.....

.....

- 18) What data is stored as a result of the completion of a
p a r t i c u l a r t r a n s a c t i o n ?
.....
.....
- 19) What areas or steps of an activity need specific
c o n t r o l ?
.....
.....
- 20) What factors govern the amount of time taken to carryout
a n a c t i v i t y ?
.....
.....
.....
- 21) How does interaction with outside elements occur?
.....
- 22) What is the cost of system operation or this particular
a c t i v i t y ?
.....
- 23) Are there specific management objectives to be
s a t i s f i e d ?
.....
.....

- 24) In what form is the system data received or stored?
.....
.....
- 25) What data is often missing?
.....
- 26) Is any data developed or used in an adhoc basis?
.....
.....
- 27) What reference tables, charts, or other data are used?
.....
.....
- 28) How are data or activities coded or abbreviated?
.....
.....
- 29) Who are the key persons in the system? Why are they
i m p o r t a n t ?
.....
.....
- 30) What obstacles or political influences affect system
e f f i c i e n c y ?
.....
.....

31) Are there any data which could be added to the document which would ease the processing burden?

yes [] No []

.....

32) Are there peak periods?

Yes[] No []

Are the documents received on continuous basis?

Yes[] No[]

32) What is the impact on the organization if this activity is not performed?

.....
.....

33) What is the impact if this activity is performed incorrectly?

.....

34) (a) What is the average backlog of documents waiting to be handled?

.....

(b) Are there any document priorities? Yes [] No []

.....

(c) Are there significant variations within the class of documents created and processed in relation to this particular transaction? Yes [] No []

35) How long are the processed documents kept in files ?

.....

36) What criteria determines when a document is disposed of?

.....

37) What are your problems concerning documents you receive
from others?

.....

.....

**APPENDIX V :
DATA ANALYSIS SHEETS**

**Administration for Refugee and Returnees Affairs
Budget and Finance Control Department
Existing System input forms and output forms summary sheet**

Form No	DESCRIPTION	No of Copies	SOURCE	DESTINATION (No of copies)	FREQUENCY	PURPOSE
1	Travel Advance Clearance	1	Finance	Finance	For any worker travelling for temporary duty	To compute the total per diem the worker is entitled to receive
2	Travel Declaration Sheet	1	Any department	Finance	As often as the above form	To provide officially approved information to the finance department for per diem computation
3	Temporary Payment Slip	1	Any department	Finance	As often as the above form	To authorize advance payments to travelling employees
4	Fuel issuing Form	1	Transport Section	Finance	Depends on travelling requirements	Used as a source document to compute cost of fuel payable to oil Co.
5	New Arrivals List	2	Social Services	Finance (1) Social Services(1)	Depends on the influx	Evidence of payments to new refugees who came to live in Addis Abeba
6	Hospital Examination Letter	2	Health & nutrition	Finance (1) Health & Nut. (1)	Depends on refugee health problems	Authorization for getting treatment in contracted hospitals
7	Pharmacy Letter	1	Health & nutrition	Finance	Depends on refugee health problems	Authorization for getting medicines from contracted pharmacist
8	Payment Request for hospitalization and medicines	2	Health & Nutrition	Finance (1) Health & Nut (1)	Depends on refugee health problems	shows the total hospitalization and medication cost to be paid
9	Subsistence allowance payment request for refugees sent for medical treatment to Addis	2	Health & Nutrition	Finance (1) Health & Nut (1)	depends on refugee health	evidence of daily subsistence allowance payments to refugees staying
10	Ledger Account	1	Finance	Finance	for each account	to keep sorted transaction data affecting a particular account
11	Journal Voucher	1	Finance	Finance	for each non cash transaction	to provide a historical evidence of a noncash transaction event

Form No	DESCRIPTION	No. of Copies	SOURCE	DESTINATION (No. of copies)	FREQUENCY	PURPOSE
12	Daily Labourer's Payment Request form	2	Site Finance Office	Head Office Fin (1) Site Finan. Off (1)	100 a year	To provide a documented evidence of wage payment
13	Payment Request					
14	Bank Disbursement Voucher	1	Head Office or Site Finance Office	Head office Finan	1000 per year	To document disbursements effected by check
15	Cash Payment Voucher	1	Head Office or Site Finance Office	Head Office Finan	20,000 per year	To document disbursements made in cash
16	Receipt Voucher	2	Head Office or Site Finance Office	Head Office Fin(1) Field Office F (1)	100 per year	To document cash receipt
17	Payroll Register	2	Head Office or Site	Head Office Fin(1) Field Office F (1)	180 per year	To document salary pay
18	Bin Card	1	Stores	Stores	For each item in the store	To provide ready reference data about the shelved items
19	Stock Card	1	Stores	Stores	For each item in the store	To accumulate all data about acquisition and use of a particular item
20	Warehouse and purchase requisition	4	Stores	Stores (2) Purchaser (2)	100 per year	To initiate a purchasing activity with official approval of the request
21	Store requisition	4	Stores	Stores(4)	300 per year	To provide a formal and objective evidence of request and issue of item
22	Dispatch Note	4	Stores	Stores (1) Driver (2) Recipient (1)	300 per year	To provide an evidence of issuances to field office and control the delivery of the items in transit
23	Goods Receiving Note	4	Stores	Stores (2) Purchaser (1) Finance (1)	95 per year	To acknowledge the receipt of purchased items and clear the purchaser's liability for purchase advances
24	Material Issue Voucher	4	Stores	Stores (3) Receiver dept.(1)	300 per year	To provide evidence of correct usage of items
25	Provident Fund ledger Account	1	Finance	Finance	One per empl.	record for contributions made by both parties
26	Cash Book	1	Finance	Finance	One for all bank accounts	Record of all checks issued for all bank acts

EXISTING DATA ITEMS SUMMARY SHEET

No	DATA ITEMS	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27
1	travellers Name	X	X																									
2	date of form	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X			X	X	X	X	X	X		
3	monthly gross salary	X										X		X	X	X	X	X			X	X	X	X	X	X		
4	reason for travel	X	X																									
5	date of departure	X	X																									
6	date of return	X	X																									
7	time of depart/return	X	X																									
8	no. of days on travel	X	X																									
9	per diem all per day	X	X																									
10	total amt. of per diem	X																										
11	breakfast. % of daily	X																										
12	lunch. % of daily per	X																										
13	dinner. % of daily per	X																										
14	room. % of daily per diem	X																										
15	calculated per diem rec	X																										
16	travellers section/dep	X	X																									
17	authorized/approved by	X	X	X	X	X				X		X		X	X			X			X	X						
18	checked/certified by	X	X	X	X	X						X	X		X			X			X	X						
19	remarks	X	X	X	X							X																
20	amt. of advance taken	X	X																									
21	pay mode cash/check	X	X																									
22	check no	X	X																									
23	prepared by	X																										
24	received by											X			X	X			X									
25	cpv/bdv no	X	X												X	X												
26	from (dept)	X	X								X				X	X										X		
27	to (dept)	X	X	X							X				X	X												
28	payee name	X	X											X	X	X	X										X	
29	advance/payment reason	X	X											X	X	X	X										X	
30	project no./symbol	X	X											X	X	X	X										X	
31	budget description	X	X											X	X	X	X										X	
32	budget code no.	X	X																								X	
33	requested by	X	X											X														
34	benzine in liters																				X	X			X			
35	disel in liters																											
36	oil in liters																											
37	full name of refugee						X	X	X	X	X																	
38	I D. No. of refugee						X	X	X	X	X																	
39	amt. received by refug						X	X	X	X	X																	
40	letter no.							X	X	X	X																	
41	pharmacy name							X	X	X	X																	
42	hospital name							X	X	X	X																	
43	description of medicin							X	X	X	X																	
44	purpose of message								X	X	X																	
45	refugee project site								X	X	X			X						X								
46	refugee medical cost								X	X	X																	

EXISTING DATA ITEMS SUMMARY SHEET

No	DATA ITEMS	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27
47	type of disease									X																		
48	no. of days for sub al									X																		
49	total amount paid									X				X	X	X												
50	debit amount										X	X			X	X												
51	credit amount										X				X	X										X		
52	account balance										X				X											X		
53	bal. carried forward										X															X		
54	bal. brought forward										X																	
55	account description										X	X			X	X												
56	JV no										X	X		X														
57	budget account no									X		X		X														
58	name of daily labourer																											
59	total no. of days work												X	X														
60	type of work												X	X														
61	rate per day												X	X														
62	total earning												X	X				X										
63	income tax deduction												X	X				X										
64	net pay												X	X				X										
65	paying office																											
66	received from													X														
67	amount received																X											
68	description of receipt																X											
69	receipt cash/check																X											
70	page no. form/record										X						X											
71	month of payment											X						X										
72	employee serial no.											X						X										
73	employee full name																	X										
74	job title																	X								X		
75	basic salary																	X								X		
76	hardship allowance																	X								X		
77	other allowance																	X										
78	overtime earning																	X										
79	provident fund deduct.																	X										
80	loan/advance deduction																	X										
81	family allot. deduction																	X										
82	total deduction																	X										
83	date of transaction																	X		X								
84	quantity received																	X		X								
85	quantity issued/dipate																	X		X								
86	quantity on hand/bal																	X		X								
87	bin card no																	X		X								
88	descrip. of store item																	X		X								
89	store trans. ref. no.																	X		X								
90	unit price																	X		X								
91	xPR. No																	X		X								
92	requesting section																	X		X								

EXISTING DATA ITEMS SUMMARY SHEET

No	DATA ITEMS	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27
93	quantity requested																				X	X						
94	total price																				X	X	X	X				
95	quan. to be purchased																				X	X	X	X				
96	unit of measurement																				X	X	X	X				
97	order no.																				X	X	X	X				
98	transported by/ driver name																				X	X	X	X				
99	dispatched by																					X	X					
100	received by																					X	X					
101	DN no.																					X	X	X				
102	goods dispatched from																					X	X	X				
103	goods dispatched to																					X	X	X				
104	type of carrier/transp																					X	X	X				
105	plate no.																					X	X	X				
106	GRN no.																					X	X	X				
107	supplier or donor																						X	X				
108	invoice no.																						X	X				
109	quantity received																						X	X				
110	goods delivered by																						X	X				
111	grand total																						X	X				
112	MIV no.																						X	X				
113	items issued to																						X	X				
114	SR. no.																						X	X				
115	date of employment																						X	X				
116	date of salary payment																							X	X			
117	description of entry																							X	X			
118	amount of check issued																								X			

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.



Mekonnen Habte Bekele

This thesis has been submitted for examination with my approval as a university advisor.

Dr. G. A. Alabi